



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



DEPARTAMENTO DE SISTEMAS
INFORMÁTICOS Y COMPUTACIÓN

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

Reingeniería de la Biblioteca Digital de SISTEDES

Trabajo Fin de Máster

**Máster Universitario en Ingeniería y Tecnología de
Sistemas Software**

Autor: Joan Chordà Blasco

Tutor: José Hilario Canós Cerdá

Curso Académico 2017-2018

Resumen

El presente Trabajo Final de Máster consiste en la reingeniería de una biblioteca digital llamada Biblioteca Digital de SISTEDES, pues su versión original ha sido desarrollada con una tecnología que presenta grandes limitaciones. El principal contenido de dicha biblioteca se basa en artículos relacionados con la ingeniería del software y las tecnologías que se pueden emplear para el desarrollo de dicho software.

Partiendo de las fases de análisis y diseño, se ha desarrollado dicha biblioteca con una tecnología distinta a la que se empleó en la versión original. En esta nueva reingeniería, se hace uso del framework Django y de tecnologías como Python, HTML, CSS y JavaScript

Palabras clave: biblioteca, biblioteca digital, desarrollo web, gestión de contenido, sistema de información

Resum

El present Treball Final de Màster consisteix en la reenginyeria d'una biblioteca digital anomenada Biblioteca Digital de SISTEDES, doncs la seva versió original ha estat desenvolupada amb una tecnologia que presenta grans limitacions. El principal contingut d'aquesta biblioteca es basa en articles relacionats amb l'enginyeria del programari i les tecnologies que es poden emprar per al desenvolupament d'aquest programari.

Partint de les fases d'anàlisi i disseny, s'ha desenvolupat aquesta biblioteca amb una tecnologia diferent de la que es va emprar en la versió original. En aquesta nova reenginyeria, es fa ús del framework Django i de tecnologies com Python, HTML, CSS i JavaScript.

Palabras clave: biblioteca, biblioteca digital, desenvolupament web, gestió de contingut, sistema de informació

Abstract

The present Final Master's Project consists of the reengineering of a digital library called Biblioteca Digital de SISTEDES, since its original version has been developed with a technology that presents great limitations. The main content of this library is based on articles related to software engineering and the technologies that can be used for the development of said software.

Starting from the analysis and design phases, the library has been developed with a technology different from the one used in the original version. In this new reengineering, Django framework and technologies such as Python, HTML, CSS and JavaScript are used.

Keywords: library, digital library, web development, content management, information system

Tabla de contenidos

1. Introducción.....	13
1.1 Motivación	13
1.2 Objetivos	14
1.3 Estructura del documento.....	14
2. Estado del arte.....	17
2.1 Concepto de bibliotecas digitales	17
2.2 Características de las bibliotecas digitales	17
2.2.1 Tipos de servicio	17
2.2.2 Modelo de referencia	18
2.3 Biblioteca Digital de SISTEDES	20
2.3.1 Propiedades de la Biblioteca Digital de SISTEDES.....	21
2.3.2 Estructuración de la Biblioteca Digital de SISTEDES.....	21
3. Reingeniería de la Biblioteca Digital de SISTEDES	23
3.1 Arquitectura.....	23
3.2 Metodología de Desarrollo	24
4. Análisis del sistema	27
4.1 Requisitos	27
4.1.1 Requisitos funcionales	27
4.1.2 Requisitos no funcionales	28
4.2 Modelado del sistema.....	28
4.2.1 Casos de uso.....	28
4.2.2 Diagrama de clases	33
5. Diseño	36
5.1 Arquitectura general del software	36
5.1.1 Arquitectura Modelo Vista Controlador.....	36
5.2. Framework Django	37
5.2.1 Estructura del proyecto Django.....	38
5.3 Capa de modelo.....	41
5.3.1 Modelo de datos	42
5.4 Capa de vista	45

5.4.1 Vistas.....	45
5.5 Capa de controlador.....	48
6. Implementación	50
6.1 Tecnología empleada.....	50
6.1.1 Python	50
6.1.2 HTML.....	51
6.1.3 JavaScript.....	52
6.2 Herramienta de desarrollo	53
6.3 Interfaz gráfica.....	53
6.3.1 Pantalla de Inicio.....	53
6.3.2 Pantalla de Inicio de sesión	54
6.3.3 Menú personal	54
6.3.4 Pantalla de creación de una conferencia	55
6.3.5 Pantalla del listado de conferencias.....	55
6.3.6 Pantalla de detalle de una conferencia	56
7. Conclusiones y trabajos futuros	57
7.1 Conclusiones.....	57
7.2 Trabajos futuros	58
REFERENCIAS.....	59
ANEXO A: ESPECIFICACIÓN DE CASOS DE USO	61

LISTA DE FIGURAS

Ilustración 1. Objeto digital (Framework Kahn/Wilensky)	18
Ilustración 2. Handle (Framework Kahn/Wilensky)	19
Ilustración 3. Protocolo de acceso al repositorio (Framework Kahn/Wilensky)	19
Ilustración 4. Diagrama del framework Kahn/Wilensky	20
Ilustración 5. Arquitectura de la Biblioteca Digital de SISTEDES	24
Ilustración 6. Ciclo metodología	25
Ilustración 7. Diagrama de contexto	28
Ilustración 8. Diagrama de casos de uso (Parte 1)	29
Ilustración 9. Diagrama de casos de uso (Parte 2)	30
Ilustración 10. <i>Diagrama de clases (UML)</i>	35
Ilustración 11. Arquitectura Modelo Vista Controlador.....	37
Ilustración 12. Arquitectura Modelo Vista Plantilla	38
Ilustración 13. Estructura del proyecto Django.	39
Ilustración 14. Configuración del proyecto Django	39
Ilustración 15. Cuerpo del proyecto Django	39
Ilustración 16. Elementos estáticos – Proyecto Django	40
Ilustración 17. Plantillas de la vista – Proyecto Django.....	40
Ilustración 18. Entorno virtual – Proyecto Django.....	41
Ilustración 19. Comunicación entre Django y PostgreSQL	42
Ilustración 20. Diagrama entidad-relación	43
Ilustración 21. Escenario de uso - Inicio de sesión	46
Ilustración 22. Escenario de uso - Submenú usuario.....	46
Ilustración 23. Escenario de uso – Vista listado	47
Ilustración 24. Escenario de uso – Vista del formulario.....	47
Ilustración 25. Escenario de uso – Ventana emergente	48
Ilustración 26. Fragmento de código Python.....	51
Ilustración 27. Fragmento de código de la vista.....	52
Ilustración 28. Pantalla de inicio -Vista.....	53

Ilustración 29. Pantalla de Inicio de sesión -Vista.....	54
Ilustración 30. Menú personal -Vista.....	54
Ilustración 31. Pantalla de creación de una conferencia -Vista	55
Ilustración 32. Pantalla del listado de conferencias -Vista	55
Ilustración 33. Pantalla de detalle de una conferencia -Vista	56

LISTA DE TABLAS

Tabla 1. Caso de uso: Crear artículo	31
Tabla 2. Caso de uso: Buscar artículo	31
Tabla 3. Caso de uso: Crear track	32
Tabla 4. Caso de uso: Crear edición	32
Tabla 5. Caso de uso: Crear conferencia	33
Tabla 6. Caso de uso: Crear conferencia	61
Tabla 7. Caso de uso: Crear artículo	61
Tabla 8. Caso de uso: Modificar artículo	61
Tabla 9. Caso de uso: Eliminar artículo	62
Tabla 10. Caso de uso: Buscar artículo	62
Tabla 11. Caso de uso: Iniciar sesión	63
Tabla 12. Caso de uso: Crear track	63
Tabla 13. Caso de uso: Modificar track	63
Tabla 14. Caso de uso: Eliminar track	64
Tabla 15. Caso de uso: Crear edición	64
Tabla 16. Caso de uso: Modificar edición	64
Tabla 17. Caso de uso: Eliminar edición	65
Tabla 18. Caso de uso: Asignar responsable de track	65
Tabla 19. Caso de uso: Crear conferencia	66
Tabla 20. Caso de uso: Modificar conferencia	66
Tabla 21. Caso de uso: Eliminar conferencia	67
Tabla 22. Caso de uso: Administrar usuarios	67
Tabla 23. Caso de uso: Buscar artículo	67
Tabla 24. Caso de uso: Ver conferencia	68
Tabla 25. Caso de uso: Ver edición	68
Tabla 26. Caso de uso: Ver track	68
Tabla 27. Caso de uso: Ver artículo	69
Tabla 28. Caso de uso: Descargar artículo	69

Tabla 29. Caso de uso: Ver conferencia	70
Tabla 30. Caso de uso: Ver edición	70
Tabla 31. Caso de uso: Ver track	70
Tabla 32. Caso de uso: Ver artículo	71
Tabla 33. Caso de uso: Listar conferencias	71
Tabla 34. Caso de uso: Listar ediciones	72
Tabla 35. Caso de uso: Listar tracks	72
Tabla 36. Caso de uso: Listar artículos	72
Tabla 37. Caso de uso: Crear autor	73
Tabla 38. Caso de uso: Modificar autor	73
Tabla 39. Caso de uso: Eliminar autor	73
Tabla 40. Caso de uso: Listar autores	74
Tabla 41. Caso de uso: Crear palabra clave	74
Tabla 42. Caso de uso: Modificar palabra clave.....	75
Tabla 43. Caso de uso: Eliminar palabra clave	75
Tabla 44. Caso de uso: Listar palabras clave	75

1. Introducción

1.1 Motivación

En términos históricos, el concepto de biblioteca ha hecho siempre referencia a la colección de información que se almacena y organiza de forma conjunta con el fin de conservar y preservar dicho contenido haciendo uso de mecanismos adecuados, además de ofrecerlo públicamente al alcance de cualquier usuario.

Con el paso del tiempo y gracias al avance de la tecnología y la expansión de internet, se permite que la distribución de dicha información, a través de las redes de comunicaciones, se pueda llevar a cabo de forma masiva.

Así pues, una biblioteca que cumple características como las nombradas anteriormente, se conoce como biblioteca digital [1] [2]. Es decir, un conjunto de recursos electrónicos y técnicos con capacidades para crear, buscar y usar información que se consideran como una extensión, una mejora del almacenamiento y una recuperación de información que manipulan datos digitales en cualquier medio. En definitiva, el concepto biblioteca digital destaca la importancia del flujo de información a través de las redes.

La Biblioteca Digital de SISTEDES [3] es una biblioteca digital cuyo contenido reside principalmente en el ámbito de la informática que aborda temas relacionados con la ingeniería del *software* y las tecnologías que se pueden emplear para el desarrollo de dicho *software*.

El principal objetivo de esta biblioteca digital es almacenar las actas que se presentan en cada edición de las Jornadas SISTEDES (<https://www.sistedes.es/jornadas>). En particular, artículos referentes a las actas de los congresos de Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Jornadas sobre Programación y Lenguajes (PROLE) y Jornadas de Ciencia e Ingeniería de Servicios (JCIS).

Actualmente, existe una versión de la Biblioteca Digital de SISTEDES implementada mediante un Sistema de Gestión de Contenido llamado WordPress que, aunque ofrece los servicios de forma correcta, presenta grandes limitaciones que imposibilitan que sea un servicio autosuficiente. Por lo tanto, se corre el riesgo de que, si se producen cambios en dicho sistema, puedan aparecer problemas en la Biblioteca Digital de SISTEDES. Es por esta misma razón por la que se pretende buscar una solución alternativa que lo evite y que ofrezca una implementación mucho más independiente.

1.2 Objetivos

El objetivo principal de este Trabajo Final de Máster (TFM) se centra en la reingeniería de la Biblioteca Digital de SISTEDES llevando a cabo una nueva versión de dicha biblioteca que implemente los aspectos y la funcionalidad de la versión original.

Así pues, este objetivo puede descomponerse en dos objetivos específicos que se detallan a continuación:

- Llevar a cabo una nueva implementación con una tecnología distinta de soporte a la Biblioteca Digital de SISTEDES.
- Migrar todo el contenido de la Biblioteca Digital de SISTEDES desde la plataforma original a la nueva.

Para llevar a cabo estos objetivos, se ha hecho uso de la metodología de desarrollo conocida como Proceso Unificado de Rational (RUP) [4] que posteriormente se detallará. En la parte de análisis se han definido todos los requisitos (funcionales y no funcionales) de la biblioteca digital, así como los casos de uso, la arquitectura y un prototipado que nos ayuda a orientar la posterior implementación. En cuanto al desarrollo, se ha utilizado un entorno de desarrollo web conocido como Django [5] basado en la arquitectura de software Modelo Vista Controlador [6] y desarrollado en Python [7].

1.3 Estructura del documento

Este documento está compuesto por siete capítulos que comprenden toda la elaboración de este Trabajo Final de Máster.

En el segundo capítulo se define el estado del arte en el campo de las bibliotecas digitales y la situación actual de estas.

En el tercer capítulo se contempla la descripción de todo el proceso de reingeniería de la Biblioteca Digital de SISTEDES. En particular, la arquitectura en la que se basa y la metodología de desarrollo empleada para dicha implementación llamada Rational Unified Process (RUP).

El cuarto capítulo trata toda la fase análisis, concretamente, se detallan todos los requisitos que se han establecido previamente al desarrollo de la herramienta. Esto implica la descripción de forma detallada de dichos requisitos, tanto los funcionales como los no funcionales. Además, se expone el diagrama de casos de uso [8] que refleja la funcionalidad de la herramienta, así como los actores involucrados, y el diagrama de clases [8] que refleja la estructura del sistema. Ambos diagramas forman parte del Lenguaje Unificado de Modelado (UML).

En el quinto capítulo se define la fase de diseño donde se plasman y se traducen los requisitos de la biblioteca en una representación de *software*. Esto incluye la

arquitectura general del software y el diseño de las capas de modelo, vista y controlador.

El sexto capítulo presenta la fase de implementación, donde se detallan los lenguajes de programación, herramientas y programas usados para el desarrollo, así como una demostración de la interfaz gráfica.

El séptimo capítulo hace referencia a las conclusiones de este Trabajo Final de Máster y los trabajos futuros que describen posibles ampliaciones que se podrían llevar a cabo.

Posteriormente, se indican todas las referencias bibliográficas que se han analizado y utilizado para la elaboración de este Trabajo Final de Máster.

Por último, se ubica un anexo con la especificación de todos los casos de uso pertenecientes a la herramienta.

2. Estado del arte

2.1 Concepto de bibliotecas digitales

En un contexto tecnológico y social, una biblioteca digital es un sistema que contiene un conjunto de materiales y servicios almacenados, administrados y accedidos a través de herramientas y redes de comunicaciones. Esto implica que, debido a que en ella coexisten materiales electrónicos que pueden ser proporcionados a través de la red, todos los usuarios pueden disfrutar de dichos recursos de información sin poseerlos físicamente

Por otra parte, también se consideran como una extensión y mejora de almacenamiento que manipulan datos digitales en cualquier medio (texto, imágenes, sonidos) y que existen en redes distribuidas cuyo contenido incluye datos, metadatos que describen dichos datos y metadatos que consisten en enlaces o relaciones con otros datos, ya sean internos o externos a la biblioteca.

2.2 Características de las bibliotecas digitales

En esta sección, se presentan algunos aspectos relevantes pertenecientes a la biblioteca digital.

2.2.1 Tipos de servicio

Como se ha mencionado anteriormente, el objetivo principal de una biblioteca digital es ofrecer acceso universal a la información permanentemente y sin limitaciones de tiempo ni de espacio. Las bibliotecas digitales deben ser capaces de guardar toda la información contenida en un sistema de almacenamiento de forma prolongada y así, poder ofrecer dicha información en cualquier lugar y momento. Con el paso del tiempo, un documento histórico o uno más actual, deben tener los mismos derechos para poder ser consultados.

En el momento en el que estos puntos anteriores se cumplen, las bibliotecas digitales deben proporcionar sus servicios a través de internet para poder exponerlo públicamente.

Por otra parte, existen múltiples servicios que debe contemplar una biblioteca digital para que ésta pueda considerarse como tal. Entre los diferentes servicios que puede ofrecer una biblioteca digital, se encuentran lo que se listan a continuación:

- **Consulta de catálogos:** Se trata del servicio más básico por excelencia dentro de una biblioteca. Éste consta de un caso particular de búsqueda por metadatos.

- **Referencias y servicios de respuesta:** En este servicio se comprenden aspectos como:
 - Anticipación a las preguntas para que nuevos usuarios puedan informarse al respecto.
 - Interacción asíncrona entre los usuarios y los bibliotecarios que ofrecen ayuda tanto a los usuarios nuevos como a los ya existentes.
- **Filtrado y diseminación selectiva de información:** Extrae de toda la información que se ofrece solo los datos de interés según el perfil de cada usuario. Mucho más sofisticado que la recuperación de la información.

En particular, de todos estos servicios que se acaban de mencionar, la Biblioteca Digital SISTEDES solo considera el requisito de consulta de catálogos.

2.2.2 Modelo de referencia

En este apartado se define de qué manera se estructura la información según el *framework* conocido como *Kahn/Wilensky Framework* [9] (en adelante, KWF). KWF presenta los diferentes conceptos clave comprendidos en una biblioteca digital.

A continuación, se detallan las entidades básicas que se encuentran en dicho sistema:

- **Originador:** Entidad que autoriza y valida un conjunto de objetos digitales. Se encarga de la disponibilidad en el sistema y del establecimiento de sus términos para su posterior uso.
- **Objeto digital:** Se trata de la unidad de intercambio en una biblioteca digital que contiene una estructura de datos concreta y una serie de características propia. Está compuesto tanto por lo datos en sí como por los metadatos que lo definen. Además, puede encontrarse en dos estados diferentes:
 - Almacenado: Tal y como el propio nombre indica, el objeto que se encuentra en este estado, indica que ha sido almacenado en un repositorio.
 - Registrado: Se trata de un objeto digital almacenado cuyo *handle* ha sido registrado en un sistema de *handle*.

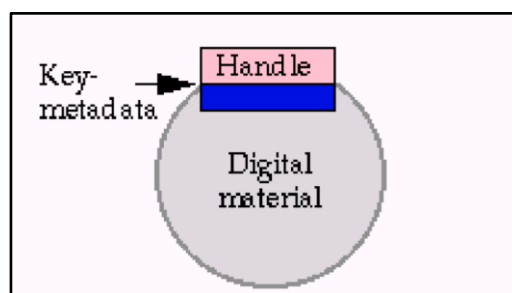


Ilustración 1. Objeto digital (Framework Kahn/Wilensky).

- **Handle:** Cada objeto digital registrado se identifica por un *handle* cuyo principal objetivo es establecer un nombre único e inmutable al propio objeto de forma persistente. Es decir, aunque el recurso se defina de diferente forma o cambie de repositorio, éste seguirá teniendo el mismo identificador.

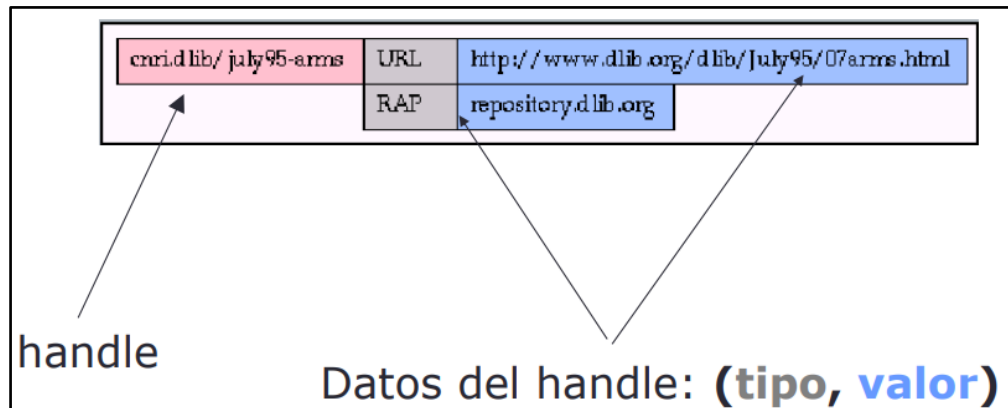


Ilustración 2. Handle (Framework Kahn/Wilensky).

- **Repositorio:** Se trata de un sistema de almacenamiento que permite el acceso a través de internet para poder consultar la información contenida a partir de un objeto digital. El repositorio tiene la capacidad de insertar nuevos objetos digitales y establecer diferentes permisos sobre éstos para hacer que estén disponibles. Estos dos últimos pasos pueden llevarse a cabo de forma conjunta (en un mismo proceso) o separada.
- **Protocolo de acceso al repositorio (RAP):** Permite acceder a un objeto digital o a sus metadatos proporcionando el *handle* establecido para éste, un tipo de solicitud de servicio y algunos parámetros adicionales. KWF define tres operaciones básicas: ACCESS_DO, DEPOSIT_DO y ACCESS_REF.

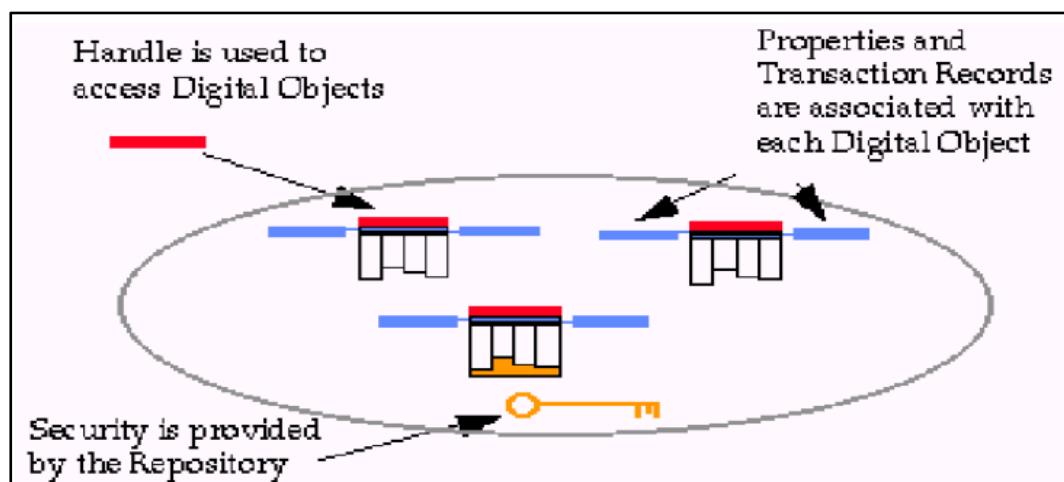


Ilustración 3. Protocolo de acceso al repositorio (Framework Kahn/Wilensky).

- **Servidor handle:** Registra identificadores únicos y persistentes a los recursos de información para poder localizar, acceder y hacer uso de los mismos. Dichos identificadores se consideran opacos y no codifican ninguna información sobre el recurso subyacente, ya que están vinculados solo a los metadatos del recurso.

KWF jerarquiza estos elementos como se refleja en la siguiente figura:

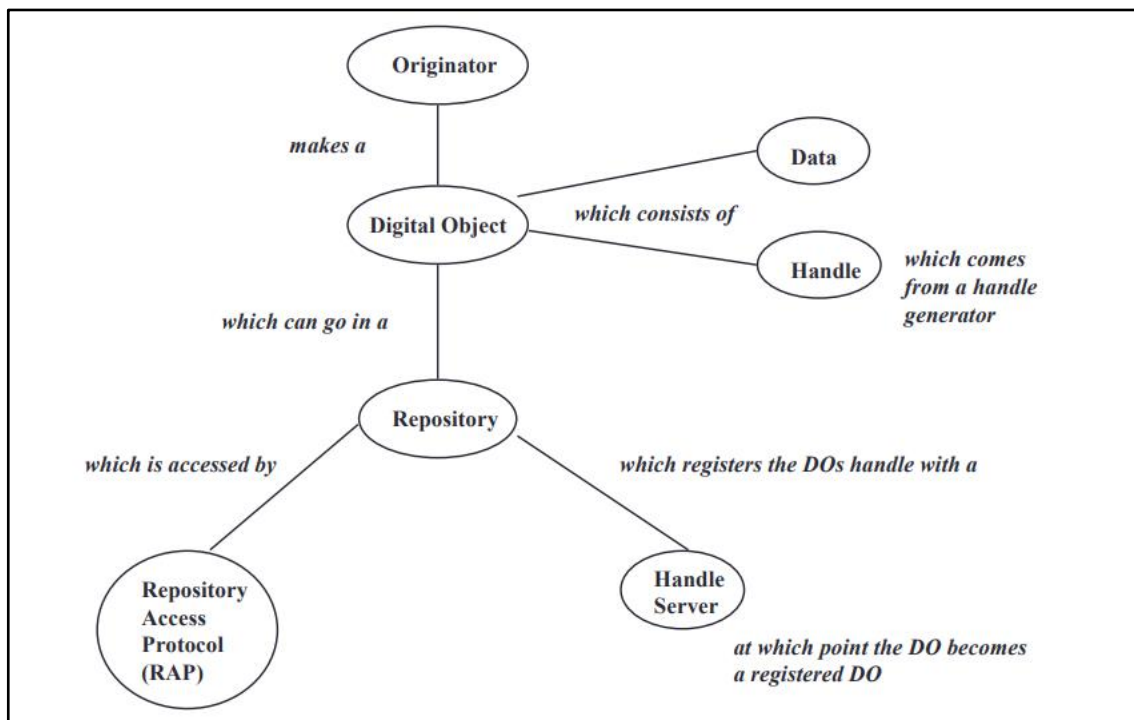


Ilustración 4. Diagrama del framework Kahn/Wilensky.

En resumen, un objeto digital es una unidad que representa un recurso de información de forma atómica. Éste está compuesto tanto por los propios datos, los metadatos y por su *handle* que se genera a partir de un servicio de *handles*. Además, es el propio repositorio el que, a partir del del servicio de *handle*, comprueba la disponibilidad de un nuevo *handle* para poder hacer uso de este y asignarlo a un objeto digital.

2.3 Biblioteca Digital de SISTEDES

La Biblioteca Digital de SISTEDES [3] es un sistema de información que se encarga principalmente de poner a disposición de sus socios y del público en general todas las actas de las jornadas nacionales que engloba.

2.3.1 Propiedades de la Biblioteca Digital de SISTEDES

En esta sección se dará paso a detallar todas las características que definen la Biblioteca Digital de SISTEDES y cuál es su comportamiento, así como todo el contenido e información que ofrece.

La sociedad SISTEDES es una asociación sin ánimo de lucro que fue fundada en el año 2005 con el fin de poder contribuir en las áreas científicas y tecnológicas de España, pero sobre todo en el ámbito de la Ingeniería del software y las Tecnologías de Desarrollo del Software. Además, también se define como una organización que intenta fomentar la investigación, la innovación y la transferencia de tecnología e ingeniería *software* entre todos los agentes pertenecientes a las áreas anteriormente mencionadas y los avances que estas disciplinas pueden promover.

Actualmente, existen tres jornadas distintas donde cada una de ellas aborda temas de distintas áreas de la informática. Dichas jornadas son la que se presentan a continuación:

- **Jornadas de Ciencia e Ingeniería de Servicios (JCIS):** Incluyen temas relacionados con los nuevos avances tecnológicos y con tecnologías que ya han sido desarrolladas en el ámbito de la computación orientada a los servicios y los procesos de negocio, ingeniería de servicios y los conocimientos que se asimilan.
- **Programación y Lenguajes (PROLE):** Sus participantes son investigadores que trabajan principalmente en el área de la programación y lenguajes de programación en general.

Al igual que en las jornadas detalladas en el punto anterior, éstas también se celebran cada año de forma que en cada evento se pretende promover tanto el intercambio de experiencias y conocimientos como la cooperación en grupo para poder destacar las ideas más relevantes y llegar a un acuerdo en común.

- **Ingeniería del Software y Bases de Datos (JISBD):** Componen nuevos foros y congregaciones cuyos temas principales se basan en las áreas de Ingeniería del Software y Bases de Datos.

2.3.2 Estructuración de la Biblioteca Digital de SISTEDES

El principal objetivo de la Biblioteca Digital de SISTEDES es almacenar y ofrecer a los visitantes todas las actas que se presentan en las jornadas anteriormente mencionadas. De esta manera, se consigue una mayor difusión de las contribuciones científico-tecnológicas llevadas a cabo por los miembros de la sociedad SISTEDES.

Además de contener información con respecto a las jornadas JCIS, JISBD y PROLE, también ofrece informes técnicos relacionados con la actividad de SISTEDES o cualquier otro documento de interés.

La forma en la que se organiza la Biblioteca Digital de SISTEDES se establece de la siguiente manera:

- **Conferencia:** Unidad principal y más general de agrupamiento de la cual se pueden crear ediciones.
- **Edición:** Cada edición refleja el año en el que se celebra una conferencia y el contenido se organiza en *tracks*.
- **Track:** Corresponden a las áreas temáticas en las que se organiza el contenido de las jornadas.
- **Artículo:** Documento que se presenta en las actas de las diferentes jornadas.

3. Reingeniería de la Biblioteca Digital de SISTEDES

En este capítulo se procede a describir todo el proceso de la reingeniería de la Biblioteca Digital de SISTEDES que ha incluido la arquitectura y la metodología de desarrollo.

3.1 Arquitectura

En capítulos anteriores se ha mencionado al *framework* Kahn/Wilensky [9] como el modelo de referencia para la construcción de la nueva biblioteca digital. A continuación, se procede a definir la arquitectura en la que se basa esta biblioteca siguiendo dicho *framework*.

En el contexto de la Biblioteca Digital de SISTEDES, las conferencias, las ediciones y los *tracks* hacen referencia a los metadatos de los artículos.

Las conferencias agrupan a las ediciones, y las ediciones agrupan a los *tracks*. Del mismo modo, los *tracks* agrupan a los artículos que constan de los objetos digitales y que se identifican con un *handle*. Dichos artículos hacen referencia al archivo de las actas de las Jornadas SISTEDES.

Por otra parte, también se establece un repositorio donde se ubican los objetos digitales mencionados que serán accedidos haciendo uso del protocolo RAP a través de los metadatos.

En la ilustración 5 se muestra cuál es la función de cada componente de esta arquitectura. En primer lugar, el usuario rellena el formulario de búsqueda en el catálogo solicitando algún artículo de la biblioteca. Posteriormente, a través de la interfaz del navegador, se le presenta al usuario una lista de los objetos digitales encontrados (en este caso, los artículos) que se ubican en el repositorio de la biblioteca digital.

En segundo lugar, el usuario selecciona el artículo en cuestión y se envía su respectivo *handle* al sistema de *handle* que, a su vez, devuelve la dirección del repositorio. Finalmente, el artículo seleccionado se ofrece desde el repositorio al usuario a través de la interfaz gráfica.

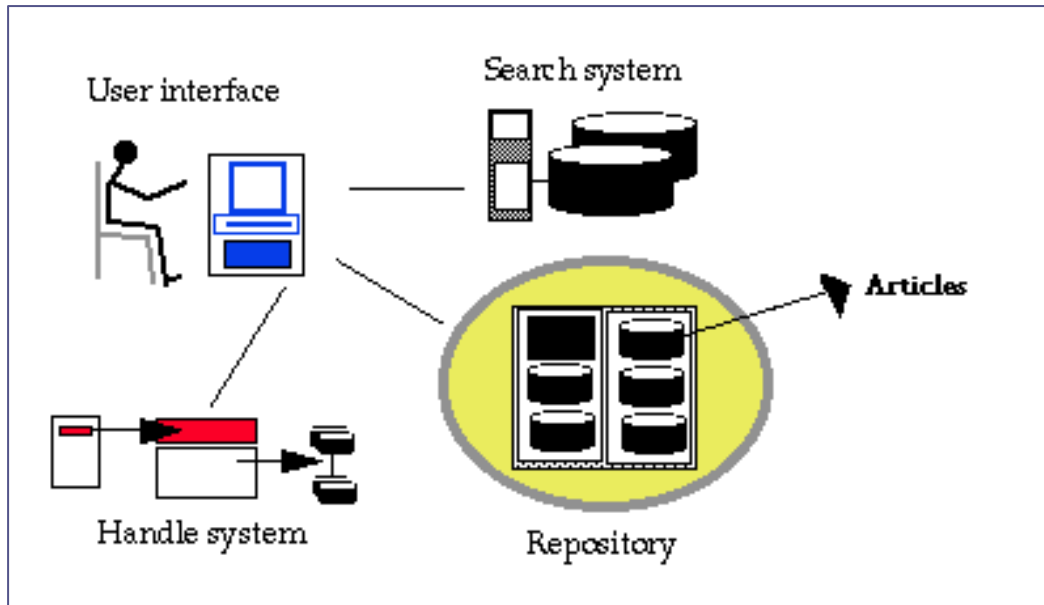


Ilustración 5. Arquitectura de la Biblioteca Digital de SISTEDES.

3.2 Metodología de Desarrollo

La metodología del desarrollo [10] es un marco de trabajo perteneciente a la Ingeniería del Software que ayuda a estructurar y a planificar todas las fases comprendidas en el proceso de desarrollo de cualquier sistema de información. Dicho de otra forma, lo que se pretende es formalizar todas las tareas que se deberían llevar a cabo para poder construir el producto final sin inconvenientes y las que definen las cuestiones: ¿Qué?, ¿Cómo? y ¿Cuándo?

Actualmente, son muchas las metodologías que pueden adaptarse a un desarrollo software de este tipo. Aun así, en este caso, se ha hecho uso de la metodología conocida como RUP [4].

RUP (*Rational Unified Process*) es una metodología de desarrollo iterativo que se orienta a partir de diagramas de los casos de uso y del comportamiento de la arquitectura empleada como tal. Además, ésta viene determinada por las siguientes fases:

- **Requisitos:** En este caso, los requisitos funcionales del sistema se convierten en casos de uso que posteriormente se detallarán. Esta fase ayuda a determinar si el proyecto en cuestión es factible para continuar el proyecto.
- **Análisis e Implementación:** Para poder pasar a la fase de la implementación, es imprescindible que la fase de los requisitos y análisis se haya planteado en su totalidad. Se trata de la fase del desarrollo físico del software.

- **Pruebas:** Una vez el *software* ha sido desarrollado, llega la fase pruebas. Es decir, determinar que no existen errores en el momento de ejecución de la aplicación. En caso contrario, se requiere ejecutar de manera iterativa toda la funcionalidad con el objetivo de refinar y disminuir las posibles excepciones.

En la figura 6 se muestra un esquema del comportamiento de esta metodología.



Ilustración 6. Ciclo metodología RUP.

En la fase de requerimiento, se han establecido los requisitos funcionales y no funcionales que debe ofrecer la Biblioteca Digital de SISTEDES. De esta forma es posible conocer qué aspectos debe cumplir la biblioteca digital para ser considerada como plenamente eficaz.

En la fase de análisis e implementación se han definido aspectos como el diagrama de casos de uso y el diagrama de clases UML; y la arquitectura del *software* que se pretende utilizar.

4. Análisis del sistema

Este capítulo comprende la fase de análisis que define el alcance del nuevo desarrollo de la Biblioteca Digital de SISTEDES para proponer una visión de alto nivel de abstracción.

4.1 Requisitos

En este apartado se especificarán, por una parte, los requisitos funcionales, que son aptos para estudiar determinados comportamientos y aspectos del sistema, y por otra parte los no funcionales, que son aquellos requisitos a los que se les establece una serie de criterios que pueden ser empleados para valorar las actuaciones del sistema.

4.1.1 Requisitos funcionales

A continuación, se nombrarán los diferentes requisitos funcionales:

- Establecer diferentes roles a los usuarios registrados en el sistema. Dichos roles son el *responsable de track*, el presidente de comité de programa y el administrador general.
- Autenticar y acreditar a los usuarios registrados en la plataforma para poder gestionar al contenido de la biblioteca.
- Dar la posibilidad al visitante de poder visualizar el contenido de la biblioteca.
- Dar la posibilidad de crear, modificar y/o eliminar el contenido de la biblioteca a los usuarios registrados. Concretamente, conferencias, ediciones, *tracks*, artículos, actores y palabras clave. Según qué rol, se le otorgarán más o menos privilegios de administración a dicho usuario. En particular:
 - Al administrador se le otorga la gestión total del contenido.
 - Al presidente del comité de programa se le otorgan los permisos de gestión de *tracks*, artículos, autores y palabras clave.
 - Al responsable de *track* se le otorgan tan solo los permisos de gestión de los artículos, los autores y las palabras clave.
- Asignar un responsable para cada *track*.
- Permitir a todos los visitantes, la búsqueda del contenido de forma sencilla y avanzada a través de metadatos.
- Enviar un correo electrónico cuando se solicite un nuevo registro en la aplicación.
- Capacitar la ejecución de las funciones concurrentemente.
- Usar hojas de estilo CSS [9] para el diseño de la web.
- Posibilitar el acceso múltiple a la plataforma de manera concurrente. Es decir, con control de sesiones.

4.1.2 Requisitos no funcionales

Por otra parte, se definirán una serie de requisitos no funcionales del entorno de soporte los cuales, como se ha definido anteriormente, están ligados a las pautas de la práctica de la herramienta que se aplican al sistema en su totalidad:

- Implementar un entorno cómodo y sencillo de utilizar.
- Poseer interfaces gráficas bien formadas.

4.2 Modelado del sistema

En este apartado se detalla el modelado del sistema que se ha definido y que se ha seguido durante todo el proceso de desarrollo de la biblioteca.

Los diagramas que se han llevado a cabo son, por un parte el diagrama de casos de uso, es decir, aquel que refleja la funcionalidad de la biblioteca y los actores implicados. Y, por otra parte, el diagrama de clases para plasmar todos los objetos que están involucrados.

4.2.1 Casos de uso

Como ya se ha mencionado anteriormente, el diagrama de casos de uso [8] refleja todos los actores que pueden intervenir en la biblioteca y sus respectivas acciones. A continuación, se definen dichos actores y todas las funciones que pueden llevar a cabo.

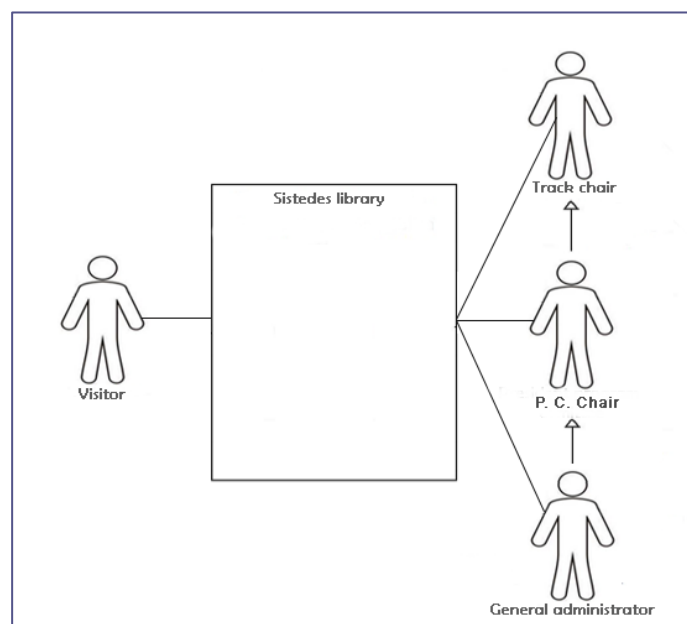


Ilustración 7. Diagrama de contexto.

A partir del diagrama de contexto que se muestra en la figura anterior, los actores que entran en juego son, por una parte, el visitante, que tan solo podrá visualizar y buscar el contenido de la biblioteca digital.

Entre dichos casos de uso se encuentran buscar los artículos existentes en el repositorio de la biblioteca mediante metadatos, como Autores, Palabras clave o Título. Por otra parte, éste podrá visualizar todas las conferencias, las ediciones y los *tracks*.

Por último, en el momento en el que se accede a la vista de un artículo en concreto, existe la opción de descargar desde la biblioteca digital el artículo en cuestión.

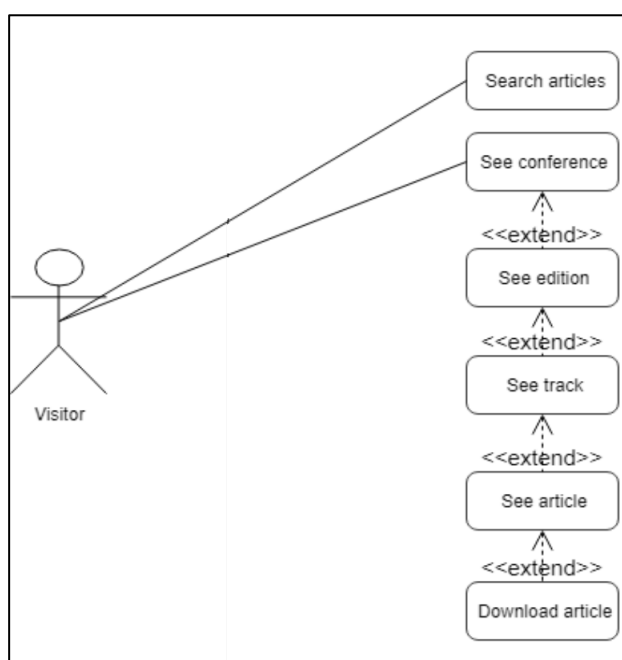


Ilustración 8. Diagrama de casos de uso (Parte 1).

Por otra parte, se presentan las acciones que pueden llevarse a cabo desde la parte privada de la biblioteca. Por este motivo, se requiere la autenticación del usuario para poder ejecutarlas.

Sin embargo, no a todos los usuarios se les otorgan los mismos permisos y puede que esto provoque la restricción de algunas funcionalidades. Es decir, en función del rol que se le establece al usuario, éste contemplará los privilegios que se determinen en dicho rol.

A continuación, se definen los distintos roles existentes en la biblioteca:

- **Responsable del *track*:** Sus privilegios se basan simplemente en la gestión de los artículos, metadatos como autores y palabras clave; y en el inicio de sesión. Se define como *Track chair*.

- **Presidente de comité de programa:** Además de heredar los privilegios del responsable del *track*, también se le da permiso para la gestión y control de los *tracks*. Se define como P. C. *Chair*.
- **Administrador general:** Puede llevar a cabo la gestión total del sistema. Es decir, además de heredar todas las funcionalidades que pueden ejecutar tanto el *track chair* como el P. C. *Chair*, también puede gestionar las conferencias, las ediciones y los usuarios de acceso a dicho sistema. Se define como Administrador general.

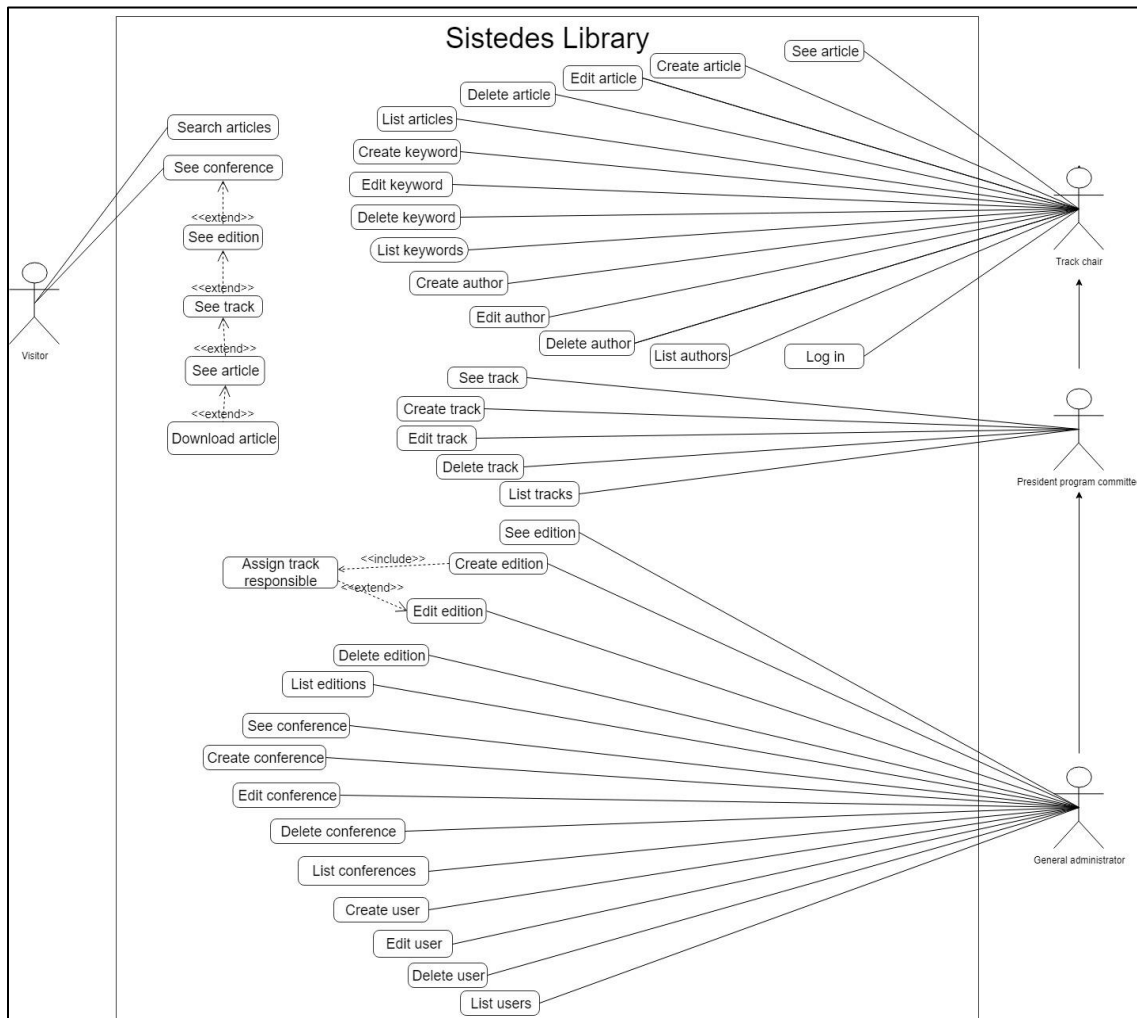


Ilustración 9. Diagrama de casos de uso (Parte 2).

En resumen, el visitante solo podrá buscar artículos mediante metadatos y ver todo el contenido público de la biblioteca. El responsable de *track* se le otorga el permiso de ejecutar acciones conocidas como CRUD (es decir, crear, leer, modificar y eliminar) sobre los artículos y algunos de sus metadatos como autores y palabras clave. Del mismo modo podrá actuar el P. C. *Chair*, aunque, también se le amplían los permisos para poder administrar los *tracks*. Finalmente, el resto de las funcionalidades que ofrece el sistema podrán llevarse a cabo por el administrador general. Dichas funcionalidades se basan en la administración de las conferencias, las ediciones, los

tracks, los usuarios, los autores de los artículos, las palabras clave y la asignación de un responsable a cada *track*.

Por consiguiente, se procede a detallar algunos de los casos de uso pertenecientes al diagrama anteriormente definido.

Tabla 1. Caso de uso: Crear artículo.

Caso de uso	Crear artículo
Actores	<i>Track chair</i>
Resumen	El usuario crea un nuevo artículo para la biblioteca.
Precondiciones	1. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	1. Hay un nuevo artículo creado para la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear un nuevo artículo. 3. El usuario establece los metadatos y guarda los cambios.	2. El sistema le indica al usuario todos los metadatos que tienen que establecer. 4. El sistema almacena en el repositorio el artículo creado.

Tabla 2. Caso de uso: Buscar artículo.

Caso de uso	Buscar artículo
Actores	Visitante
Resumen	El usuario busca un artículo en concreto.
Precondiciones	-
Postcondiciones	1. Se muestra por pantalla el artículo que se busca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de buscar un artículo. 3. El usuario escribe en el formulario de búsqueda el texto con el que quiere encontrar el artículo.	2. El sistema le muestra al usuario un formulario de búsqueda. 4. El sistema le muestra al usuario los artículos cuyos metadatos coinciden con el texto insertado en el formulario de búsqueda.

Tabla 3. Caso de uso: Crear track.

Caso de uso	Crear track
Actores	<i>P. C. Chair</i>
Resumen	El usuario crea un nuevo <i>track</i> para la biblioteca.
Precondiciones	1. El usuario está autenticado como <i>P. C. Chair</i>
Postcondiciones	1. Hay un nuevo <i>track</i> creado en la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear un nuevo <i>track</i> . 3. El usuario rellena los campos que se muestran en pantalla y guarda los cambios.	2. El sistema le muestra al usuario todos los campos que tienen que rellenar. 4. El sistema almacena en la biblioteca el <i>track</i> creado.

Tabla 4. Caso de uso: Crear edición.

Caso de uso	Crear edición
Actores	<i>Administrador general</i>
Resumen	El usuario crea una nueva edición para la biblioteca.
Precondiciones	1. El usuario está autenticado como Administrador general.
Postcondiciones	1. Hay una nueva edición creada para la biblioteca.
Incluye	Asignar responsable de <i>track</i>
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear una nueva edición. 3. El usuario rellena los campos que se muestran en pantalla y guarda los cambios.	2. El sistema le muestra al usuario todos los campos que tienen que rellenar. 4. El sistema almacena en la biblioteca la nueva edición.

Tabla 5. Caso de uso: Crear conferencia.

Caso de uso	Crear conferencia
Actores	<i>Administrador general</i>
Resumen	El usuario crea una nueva conferencia para la biblioteca.
Precondiciones	1. El usuario está autenticado como administrador general.
Postcondiciones	1. Hay una nueva conferencia creada para la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear una nueva conferencia. 3. El usuario rellena los campos que se muestran en pantalla y guarda los cambios.	2. El sistema le muestra al usuario todos los campos que tienen que rellenar. 4. El sistema almacena en la biblioteca la nueva conferencia.

4.2.2 Diagrama de clases

El diagrama de clases [8] representa de forma minuciosa la estructura y organización de todos aquellos objetos que se manifiestan en la Biblioteca Digital de SISTEDES. Con ello, se pretende plasmar todas las composiciones, relaciones, herencias y actuaciones de cada uno de los objetos.

A continuación, se detallan todas las entidades que forman parte del sistema y sus respectivos atributos:

- **Conference:** Representa a una conferencia de la biblioteca y contiene información como el identificador, el nombre, el acrónimo que servirá para identificarla y la descripción.
- **Edition:** Representa a una edición de la biblioteca y contiene información como el identificador, el nombre, el año en el que se celebra, el lugar donde se celebra, la información breve y la descripción.
- **Track:** Representa a un *track* de la biblioteca y contiene información como el identificador, nombre, la información breve y la descripción.
- **User:** Representa a un usuario registrado en la biblioteca y contiene información personal como el identificador, el nombre, los apellidos, el nombre de usuario, el correo electrónico, la contraseña para poder iniciar sesión y un rol que establece sus permisos.
- **Article:** Representa a un artículo de la biblioteca y contiene información como el identificador, el título, el *handle* y la ruta del documento indicando donde se ubica el documento en el repositorio.

- **Author:** Representa a el autor de un artículo de la biblioteca y contiene información personal como el identificador, el nombre, los apellidos, el correo electrónico, el país, la universidad y el departamento.
- **Keyword:** Representa a la palabra clave de un artículo de la biblioteca y contiene información como el identificador y el nombre.
- **Access Rights:** Representa a los derechos de acceso de un artículo de la biblioteca.

Tal y como se observa en el siguiente diagrama, una conferencia contiene múltiples ediciones. Además, cada edición contiene múltiples *tracks* que, a su vez, contienen diferentes artículos. Por otra parte, un artículo contiene metadatos como autores y palabras clave.

Finalmente, ya que las ediciones, los *tracks* y los artículos pueden implementarse por diferentes usuarios, existe una relación con el usuario para poder identificarlo.

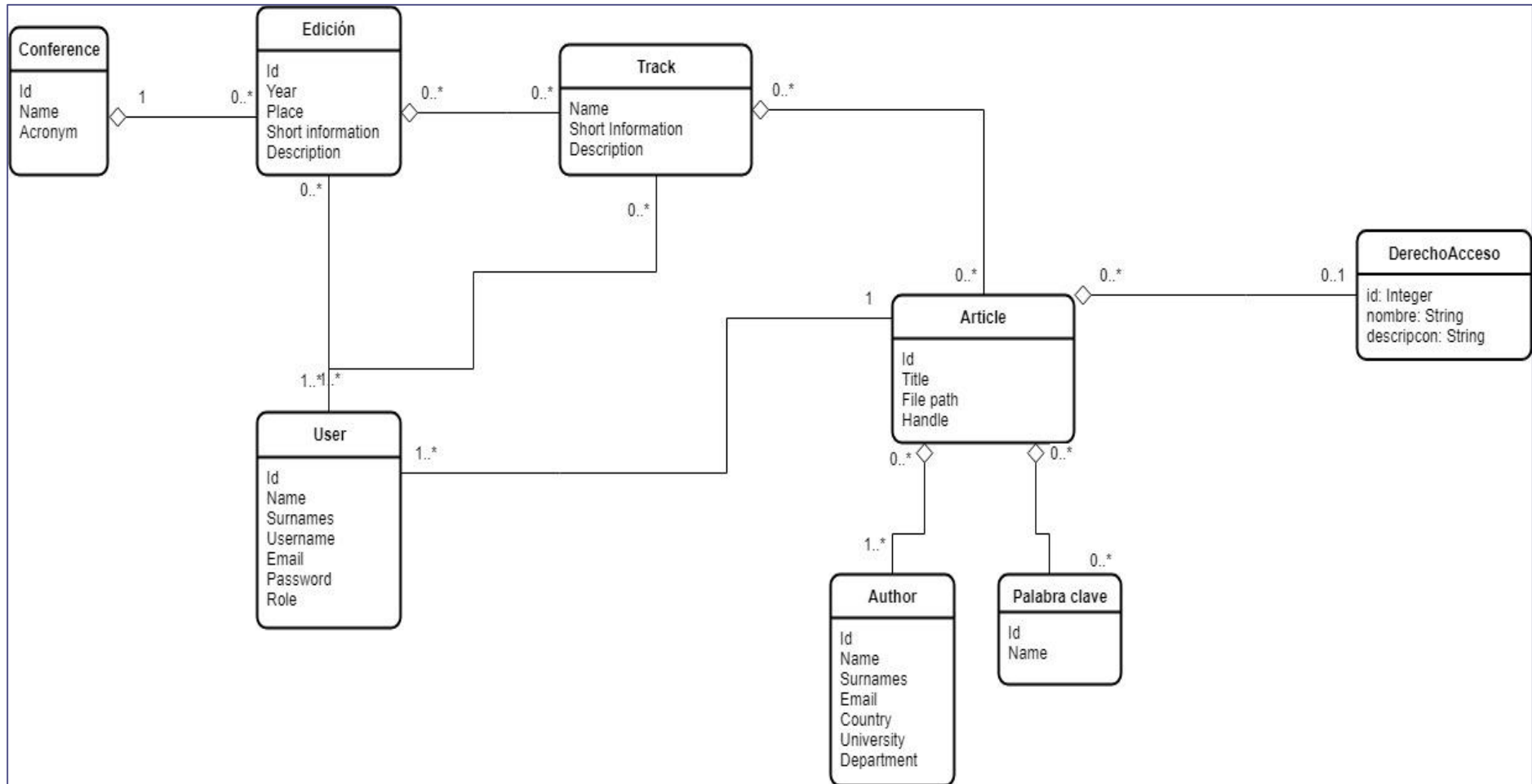


Ilustración 10. Diagrama de clases (UML).

5. Diseño

5.1 Arquitectura general del software

A continuación, se describe la arquitectura general del software que se ha utilizado para el desarrollo de la Biblioteca Digital de SISTEDES y que se conoce como arquitectura Modelo Vista Controlador.

5.1.1 Arquitectura Modelo Vista Controlador

La arquitectura Modelo Vista Controlador [6] define una forma de desarrollo *software* que pretende separar los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Dicha arquitectura se basa en los siguientes tres componentes:

- **Modelo:** Representación de los datos que maneja el sistema, los mecanismos de persistencia. Es el responsable de acceder a la capa de almacenamiento de los datos.
- **Vista:** Información que será percibida por el cliente a través de una interfaz gráfica. Establece cómo se verán los datos.
- **Controlador:** Este actúa como intermediario entre el Modelo y la Vista controlando todo el flujo de información. Responde a las solicitudes que el cliente envía y, posteriormente, genera peticiones al Modelo. Finalmente, el cliente observa los cambios.

En la siguiente figura, se muestra de forma gráfica cuál es el funcionamiento de esta arquitectura:

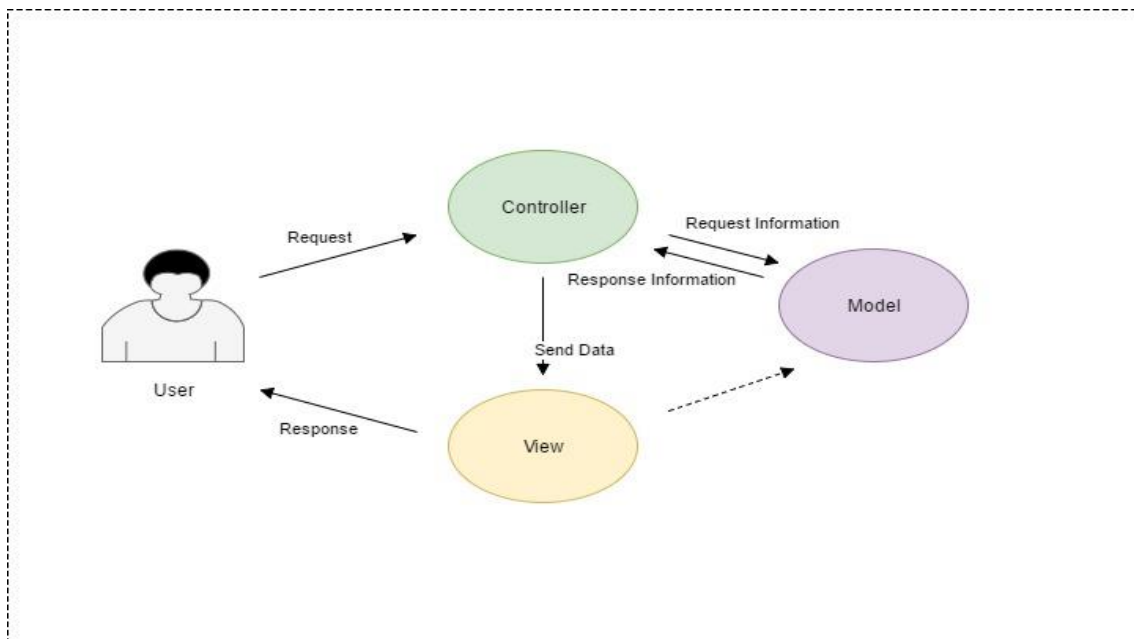


Ilustración 11. Arquitectura Modelo Vista Controlador.

Tal y como muestra en la anterior ilustración, cuando el usuario manda una petición al navegador, el controlador es el responsable de tomar una decisión ya que es el componente que se encarga de la lógica del sistema. Posteriormente, el controlador se comunica con el modelo para obtener información en relación a la solicitud. Finalmente, una vez el controlador tiene todos los datos que el modelo le ha facilitado, éste comunica dicha información a la vista, que es la que se encargará de organizar la información y mostrársela al usuario.

5.2. Framework Django

Para la construcción de esta biblioteca digital se ha hecho uso del *framework* Django [5] que, aunque posteriormente se detallará, hay que destacar que dicho *framework* define su propia arquitectura.

Django admite y sigue de cerca el patrón Modelo Vista Controlador. Sin embargo, como es el propio Django el que se ocupa de la parte del controlador, establece una arquitectura propia llamada Modelo Vista Plantilla.

La arquitectura Modelo Vista Plantilla (MVT) [11] es ligeramente diferente de la arquitectura Modelo Vista Controlador. Es decir, debido a que es Django el que se encarga de la parte del controlador, utiliza su propia lógica de implementación. En este caso, desaparece el controlador y se reemplaza por la vista que actúa como dicho controlador, además se añade un nuevo componente llamado Plantilla que gestiona la interfaz de usuario.

En la siguiente figura, se muestra cuál es el funcionamiento:

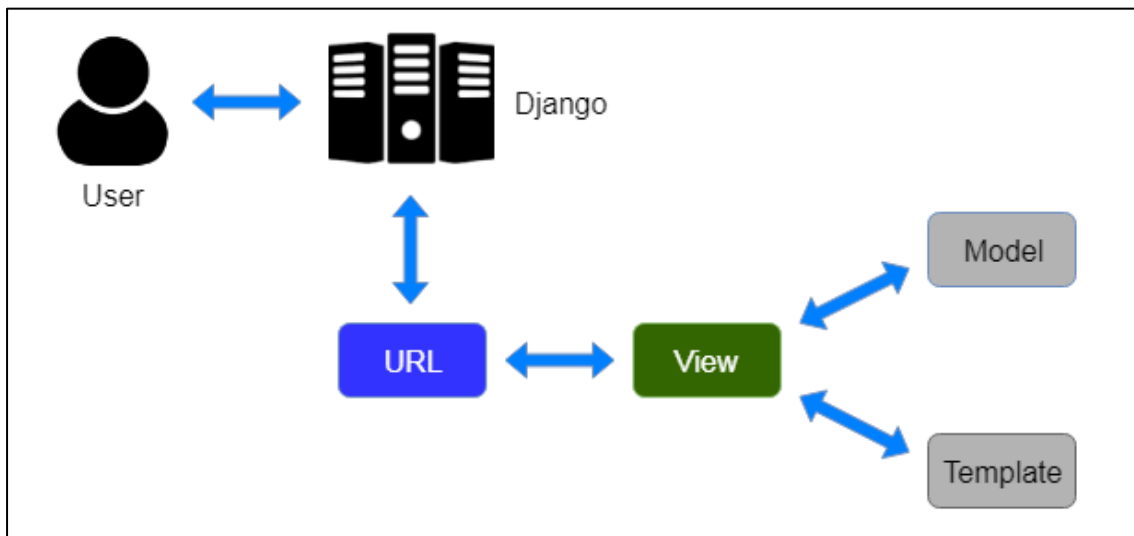


Ilustración 12. Arquitectura Modelo Vista Plantilla.

A diferencia de la arquitectura Modelo Vista Controlador, es la propia vista la que recibe la petición del cliente y la que decide qué cambios hacer en el sistema. Para ello, establece una comunicación con el modelo y la plantilla; y construye la respuesta al usuario.

En conclusión, el modelo sigue siendo el modelo, la vista ya no actúa como la vista que define la arquitectura Modelo Vista Controlador, es más bien un controlador. Finalmente, la Plantilla se encarga de las decisiones relacionadas a la presentación.

Tal y como se ha descrito anteriormente, Django es un *framework* web de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Pero, además, contiene características destacables que pueden inducir a emplearlo para desarrollos de *software* tipo *web*:

- Implementación de aplicaciones más potentes y eficientes.
- Una estructuración y arquitectura que te permiten la creación de aplicaciones dotadas de un mayor rendimiento.
- Integración de sistemas de seguridad.
- Integración de sistemas de plantillas.
- Integración de administración de sesiones.

5.2.1 Estructura del proyecto Django

A continuación, se presenta la estructura y organización del proyecto de Django que reflejan todos los componentes y repositorios pertenecientes al proyecto para poder llevar a cabo el correcto funcionamiento y que se comporte de forma adecuada:

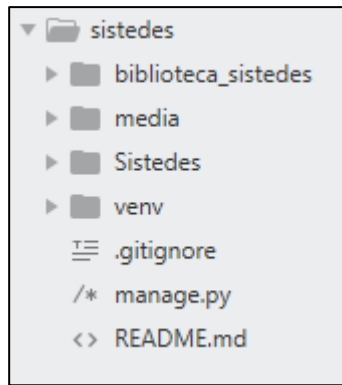


Ilustración 13. Estructura del proyecto Django.

Un proyecto de Django está compuesto por:

- **Configuración:** Valores predefinidos para poder arrancar la aplicación. Valores como propiedades de conexión a la base de datos, aplicaciones, enlaces estáticos, ...
- **Modelos:** Unidad que contempla atributos y características con información sobre sí mismo. Contiene campos y funcionalidades esenciales que reflejan su comportamiento.
- **Formularios:** Contiene todos los campos pertenecientes a un modelo para poder llevar a cabo una operación de crear, modificar y/o eliminar para posteriormente conservar los cambios en base de datos.
- **Enlaces:** Se basa en la definición de todas las posibles *URL* cuyo acceso devuelve una vista en concreto.
- **Vistas:** Fichero en el que se definen todas las vistas que podrán ser representadas y visibles de cara al usuario. Existe una vista por cada enlace.
- **Aplicaciones:** Listado de modelos y librerías externas las cuales se van a necesitar en la funcionalidad interna.

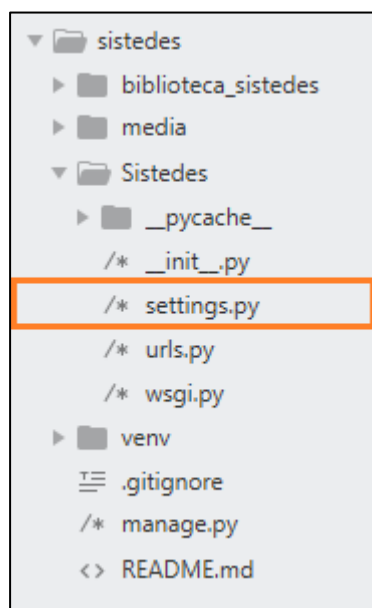


Ilustración 14. Configuración del proyecto Django.

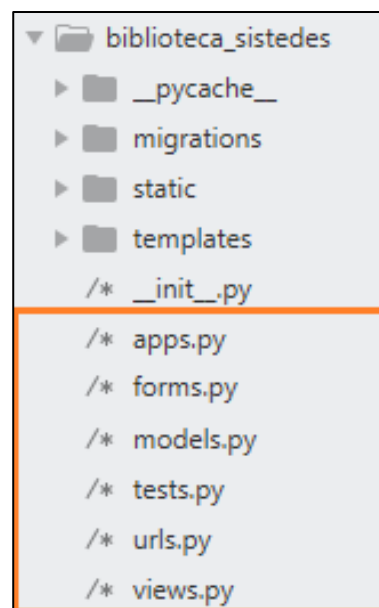


Ilustración 15. Cuerpo del proyecto Django.

- **Plantillas:** Esqueletos con simple texto basados en formatos como HTML, XML, CSV... Definen la composición de la vista solicitada por los usuarios. Pueden contener variables y etiquetas que representan estructuras condicionales.
- **Elementos estáticos:** Repositorio en el que se encuentran *scripts* que aportan funcionalidad extraordinaria a la vista. Por ejemplo, hojas de estilo CSS, ficheros JavaScript, iconos...

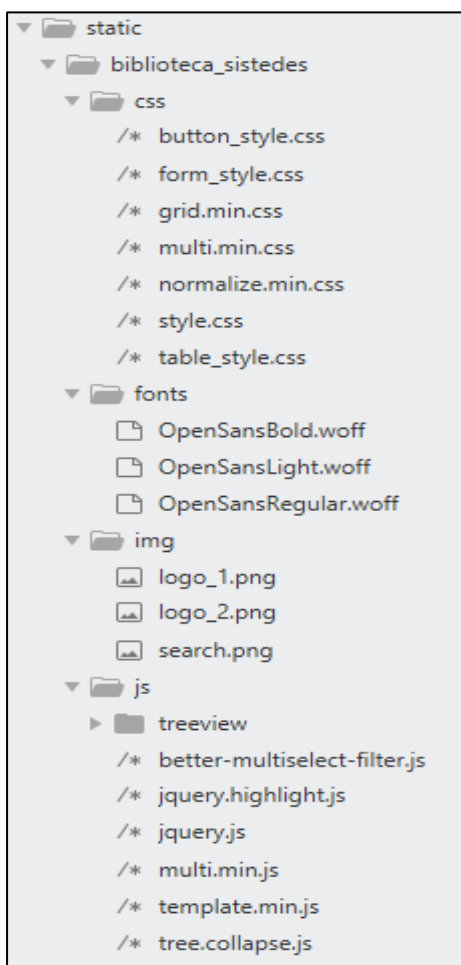


Ilustración 16. Elementos estáticos – Proyecto Django.

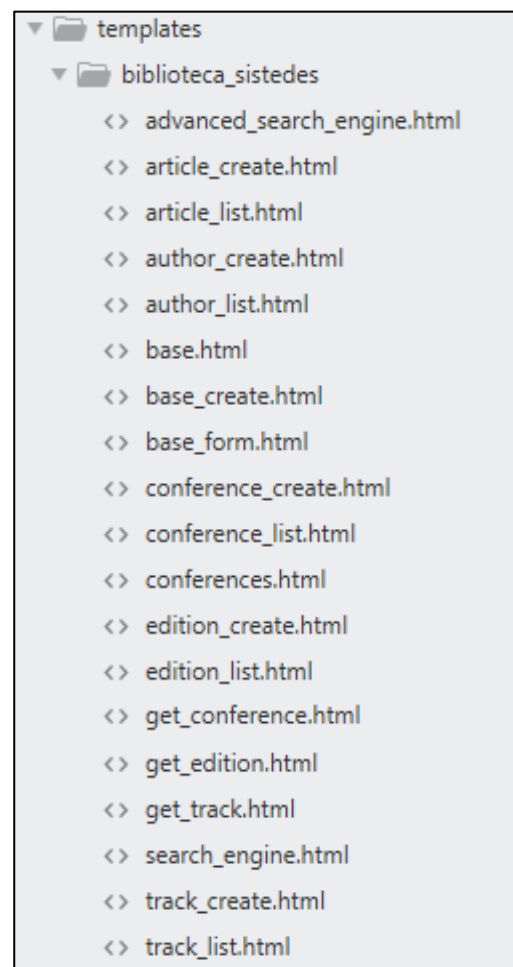


Ilustración 17. Plantillas de la vista – Proyecto Django.

- **Entorno virtual:** Repositorio opcional pero recomendado. Contiene todas las librerías que se necesitan en la lógica para poderse exportar a otro servidor sin necesidad de que éste incluya preinstaladas todas las dependencias.

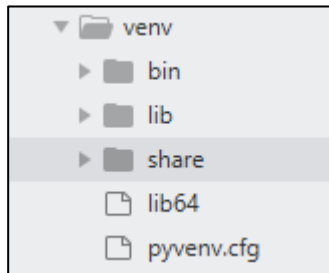


Ilustración 18. Entorno virtual
– Proyecto Django.

5.3 Capa de modelo

Para la Biblioteca Digital de SISTEDES, es indispensable el uso de una tecnología que facilite el almacenamiento de todo el contenido de forma segura y persistente. Existen diversos sistemas de gestión de base de datos relacionales que pueden cumplir los requerimientos que se solicitan en la Biblioteca Digital de SISTEDES. Sin embargo, el *framework* utilizado no ofrece soporte a todos los sistemas de almacenamiento y, por lo tanto, se presentan limitaciones en el momento de elegir un gestor de almacenamiento.

Tras el estudio de diversos sistemas de almacenamiento, se decide hacer uso de la herramienta PostgreSQL [12] por los puntos que se describen a continuación:

- Gestión de grandes cantidades de datos.
- Control de concurrencia para usuarios que acceden de forma simultánea.
- Soporte ACID (Atomicidad, Consistencia, Aislamiento, Persistencia).
- Copias de seguridad constantes.

Por este motivo, PostgreSQL se convierte en uno de los gestores de almacenamiento más potentes y robustos del mercado que ofrece estabilidad, facilidad de administración e implementación de estándares.

La forma en la que Django interactúa con PostgreSQL, es mediante su herramienta de mapeo objeto-relacional, que le permite comunicarse con la base de datos para poder consultar y manipular la información y obtener resultados de forma adaptativa al lenguaje Python. Esto es posible por medio de los *Querysets*, que constan de un listado de objetos de un determinado modelo que te permite leer datos de una base de datos, filtrarlo u ordenarlos. Estos constan de sentencias SQL que son construidas y generadas internamente para poder buscar, filtrar o acceder sin perjudicar la base de datos.

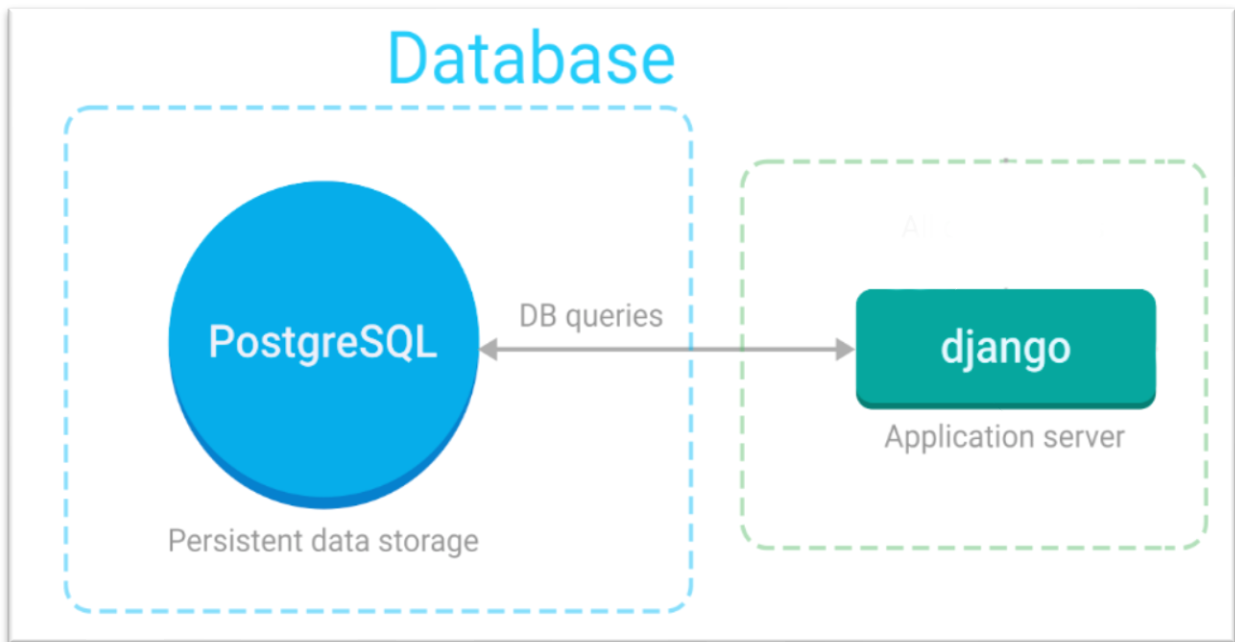


Ilustración 19. Comunicación entre Django y PostgreSQL.

5.3.1 Modelo de datos

La ilustración 23 muestra el diagrama entidad-relación resultante del modelado de la base de datos:

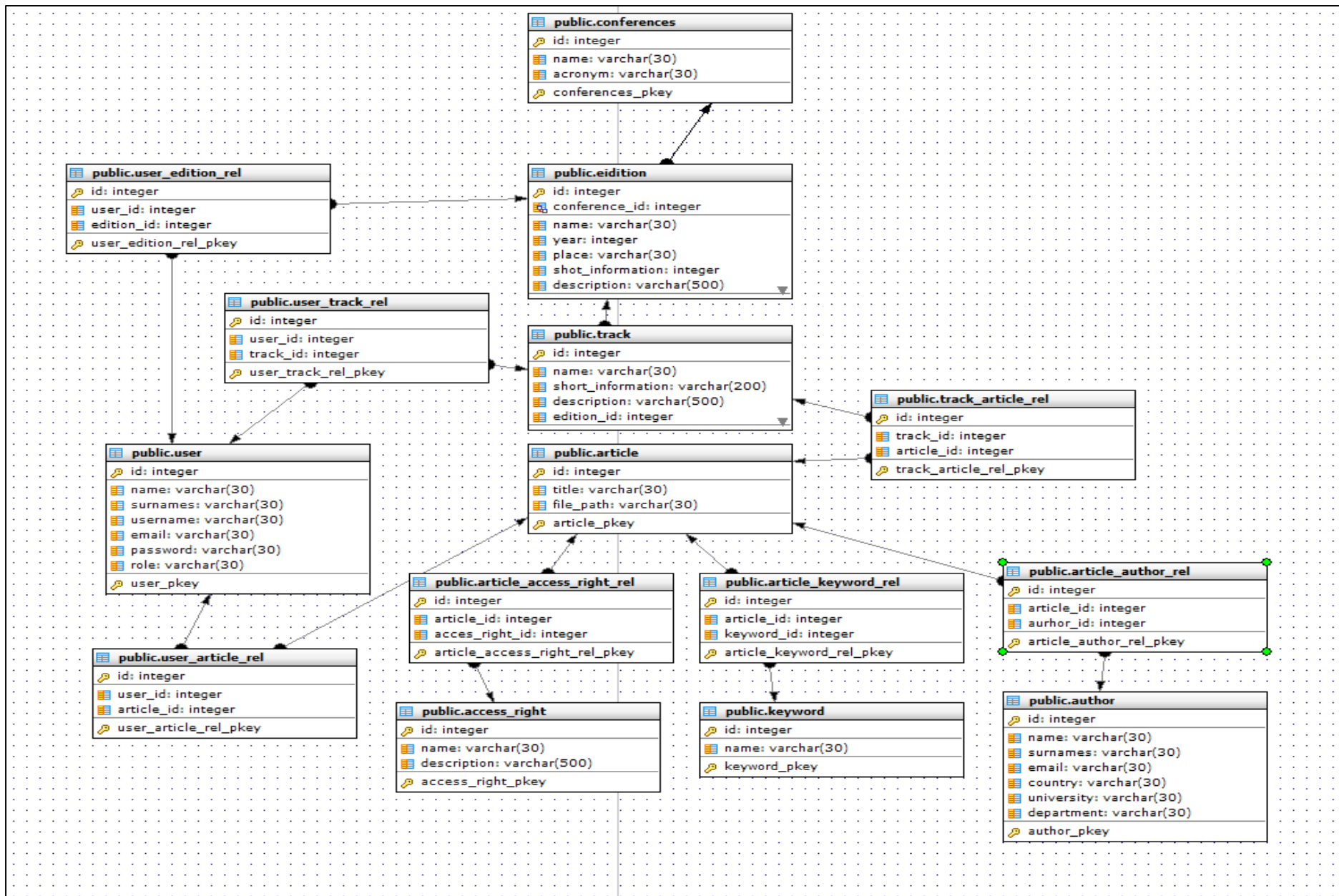


Ilustración 20. Diagrama entidad-relación.

Por último, se explica en detalle lo que representa cada columna:

- **Conference.**
 - **id.** Identificador numérico interno.
 - **name.** Nombre de la conferencia.
 - **acronym.** Acrónimo de la conferencia.
 - **description.** Descripción de la conferencia.
- **Edition.**
 - **id.** Identificador numérico interno.
 - **name.** Nombre de la edición.
 - **short_information.** Breve información del *track*.
 - **description.** Descripción del *track*.
- **Track.**
 - **id.** Identificador numérico interno.
 - **name.** Nombre del track.
 - **year.** Año en el que se celebra la edición.
 - **place.** Lugar donde se refleja la edición.
 - **short_information.** Breve información de la edición.
 - **password.** Descripción de la edición.
 - **role.** Descripción de la edición.
- **User:**
 - **id.** Identificador numérico interno.
 - **name.** Nombre del usuario registrado.
 - **surnames.** Apellidos del usuario registrado.
 - **username.** Nombre de usuario del usuario registrado.
 - **email** Correo electrónico del usuario registrado.
 - **description.** Descripción de la edición.
- **Article:**
 - **id.** Identificador numérico interno.
 - **title.** Título del artículo.
 - **file_path.** Refleja el *path* que refleja la ubicación del artículo.
 - **handle.** Identificador del artículo en el servicio de *handle*.
- **Author:**
 - **id.** Identificador numérico interno.
 - **name.** Nombre del autor.
 - **surnames.** Apellidos del autor.
 - **email.** Correo electrónico del autor.
 - **country** País del autor.
 - **university.** Universidad del autor.
 - **department.** Departamento del autor.
- **Keyword:**
 - **id.** Identificador numérico interno.
 - **name.** Nombre de la palabra clave.

- **Access Rights:**
 - **id.** Identificador numérico interno.
 - **name.** Nombre del derecho de acceso.
 - **description.** Descripción del derecho de acceso.

5.4 Capa de vista

La capa de la vista es aquel nivel en el que residen principalmente los componentes que serán visibles gráficamente para cualquier usuario. En otras palabras, es la que se encarga de representar la información almacenada en el sistema de almacenamiento de datos.

5.4.1 Vistas

Una de las etapas preliminar al desarrollo de una web o de una aplicación informática consta del modelo de representación con un alto detalle de lo que será el proyecto final. A través de la creación de los prototipos, se puede identificar fácilmente cómo será la interacción entre el usuario y la interfaz gráfica, así como clarificar todos los requerimientos.

Para el desarrollo del prototipo se ha hecho uso de la aplicación de prototipado Balsamiq, la cual permite bocetos muy realistas, con lo que estos serán muy similares a la interfaz que resulte de la implementación.

Al existir una primera versión de esta biblioteca digital, hay algunos escenarios que no requerían la creación del prototipo, pues seguirán manteniendo el mismo comportamiento y aspecto. Sin embargo, como sí se insertan nuevas funcionalidades adicionales en dicha biblioteca, sí se pudieron llevar a cabo los respectivos prototipos para poder tener una visualización previa de cuál sería la forma en la que se representarían gráficamente.

Dichas funcionalidades son:

- Iniciar sesión.
- Creación, edición y eliminación de conferencias, ediciones, *tracks* y artículos.
- Asignación de responsables a los diferentes artículos.
- Nueva sección en el menú principal.

A continuación, se muestran todos los *mockups* que se construyeron previamente a la fase del desarrollo de la aplicación:

1. Pantalla de inicio de sesión: Consiste en la pantalla que se usará para que los usuarios registrados en la Biblioteca Digital de SISTEDES puedan iniciar sesión a partir de sus credenciales y hacer uso de las funcionalidades que forman parte de la sección privada de dicha biblioteca.

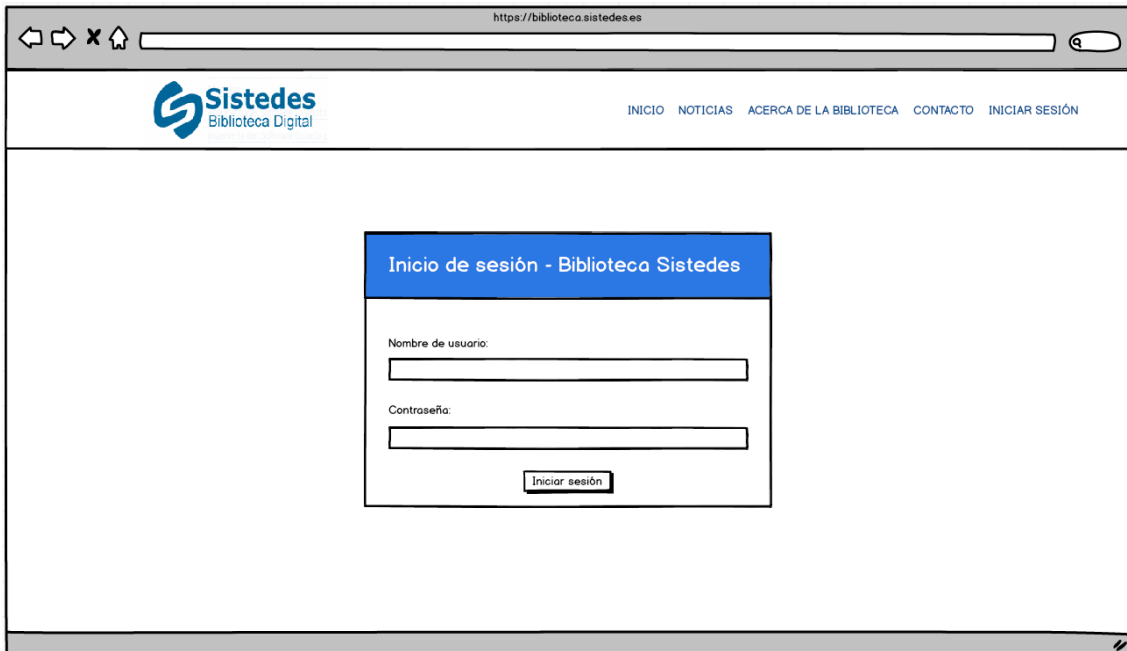


Ilustración 21. Escenario de uso - Inicio de sesión.

2. Menú personal: En el menú principal de la biblioteca, se pretende insertar un apartado que consiste en un desplegable con todas las secciones de la parte privada a las que el usuario que ha iniciado sesión tiene acceso.



Ilustración 22. Escenario de uso - Submenú usuario.

3. Vista general del listado: Consta de la vista de listado de todos los objetos digitales o metadatos que forman parte de la biblioteca con un poco de información más una columna adicional reservada para todas las acciones (ver, editar, eliminar, etc.) que se pueden llevar a cabo.

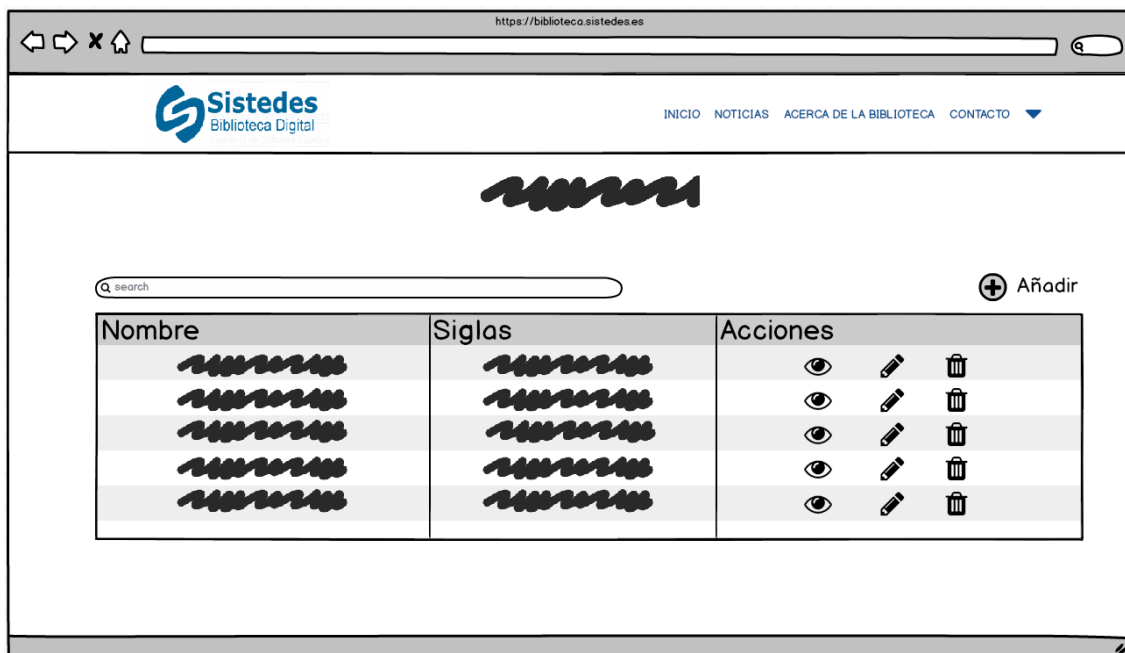


Ilustración 23. Escenario de uso - Vista listado.

4. Vista formulario general: Pantalla orientativa que ayuda a visualizar la estructuración del formulario de creación o edición de los objetos. Éste varía según la información que define a dichos objetos. En esta se engloban las funcionalidades de creación y edición.

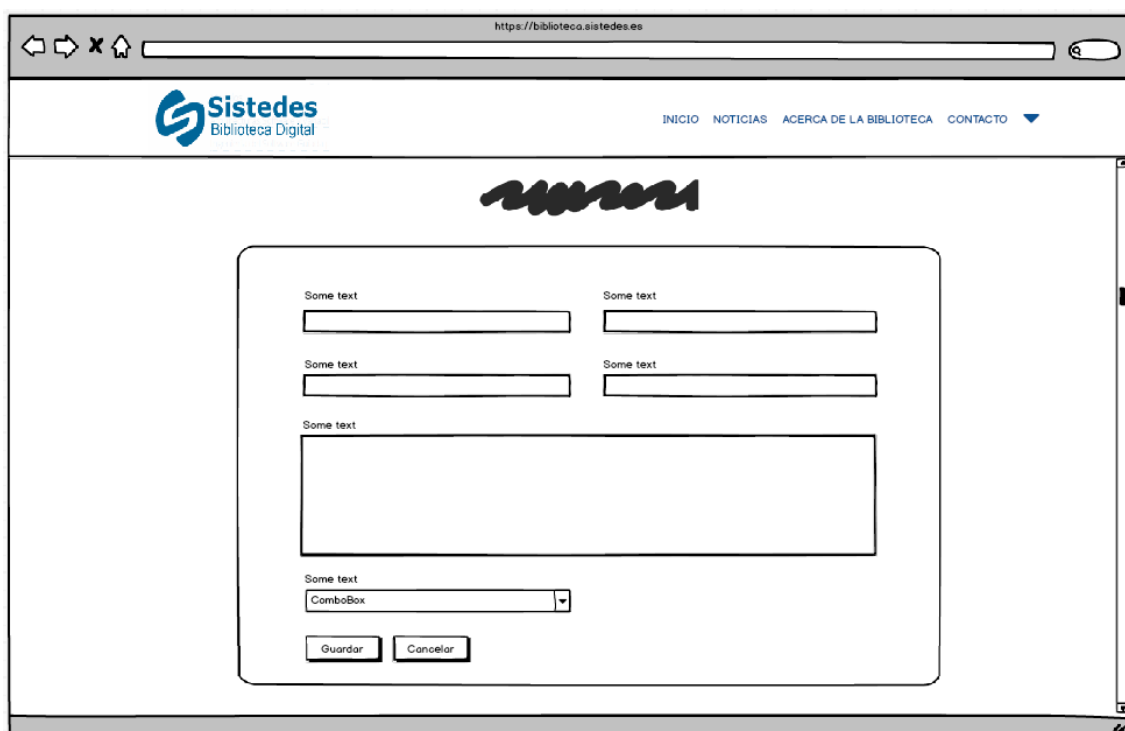


Ilustración 24. Escenario de uso - Vista del formulario.

5. Ventana emergente general: Algunas de las funciones adicionales requieren de una ventana emergente para poder insertar un número reducido de datos. En la siguiente figura, se puede visualizar cuál sería el aspecto de dicha ventana.

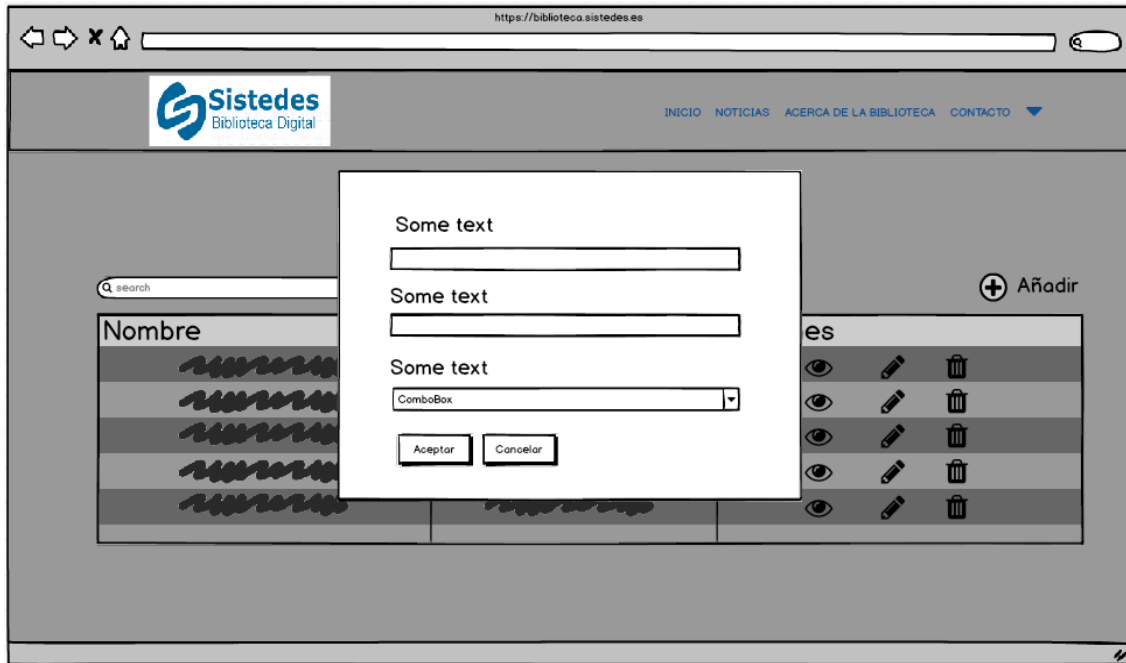


Ilustración 25. Escenario de uso - Ventana emergente.

5.5 Capa de controlador

Tal y como se ha descrito anteriormente, la capa del controlador es la encargada de procesar los datos de acuerdo a las reglas de la Biblioteca Digital de SISTEDES.

A partir de la herramienta de mapeo objeto-relacional (ORM) facilitada por Django, cada tabla de la base de datos corresponde a una clase de la capa de modelo. De esta forma los controladores pueden interactuar con los modelos y, sin necesidad de ninguna implementación adicional, estos cambios se ven reflejados de forma automática en la capa de modelo.

En Django, las clases UML son representadas por modelos. Dichos modelos creados coinciden con las tablas contenidas en la base de datos. Es decir, Conference, Edition, Track, User, Article, Keyword, Author y AccesRight.

6. Implementación

En este capítulo, se definirá la tecnología utilizada para la implementación de la Biblioteca Digital de SISTEDES, así como algunos ejemplos de las vistas de dicha biblioteca.

6.1 Tecnología empleada

En este apartado se procede a describir toda la tecnología de la que se ha hecho uso para el desarrollo de la Biblioteca Digital de SISTEDES.

6.1.1 Python

En concreto para este proyecto, el lenguaje de programación que protagoniza y se encarga de la lógica es Python [7] (en específico, la versión 3.6).

Python es un lenguaje de programación multiparadigma que da soporte a la programación orientada a objetos, imperativa y funcional con un gran abanico de propiedades ventajosas. Además de ser un lenguaje predeterminado y compatible con el *framework* Django, ofrece muchas peculiaridades que hacen que predomine sobre el resto:

- Estructuras de datos eficientes y efectivas.
- Tipado dinámico.
- Programación con una sintaxis fuerte y sencilla.
- Compatibilidad con funcionalidades implementadas con otros lenguajes de programación (por ejemplo, C o C++).
- Abundancia de bibliotecas.
- Escritura de código asíncrono.

Python es uno de los lenguajes de programación que más auge ha tenido en los últimos años, sobre todo en el desarrollo de aplicaciones web. Sin embargo, Python puede ser soportado por diversos *frameworks* que ayudan y facilitan la implementación de aplicaciones informáticas.

A continuación, se muestra un fragmento de código que consiste en el método que devuelve la primera plantilla HTML en el momento en el que se accede a la Biblioteca Digital de SISTEDES:

```

# PUBLIC PART

def index(request):
    # request.session['user'] = 'pepe'
    # request.session.set_expiry(30)
    key = request.GET.get('s') or False
    if key is not False:
        conference_list = search_article('b', key)
        context = {'conference_list': conference_list, 'key': key}
        context.update(global_context())
        return render(
            request,
            'biblioteca_sistedes/search_engine.html',
            context
        )
    else:
        return render(
            request,
            'biblioteca_sistedes/base.html',
            global_context()
        )

```

Ilustración 26. Fragmento de código Python.

La ilustración anterior refleja el método que devuelve la primera plantilla en el momento de acceder a la web llamada 'base.html'. En el caso en el que se utilice el formulario de búsqueda ofrecido por la biblioteca, este también redirige a la página índice pero esta vez con parámetros adicionales.

6.1.2 HTML

HTML (HyperText Markup Language) [13] es el lenguaje de marcado que se utiliza en la World Wide Web. A partir de su versión 5 se utiliza exclusivamente para definir la semántica de los documentos web, quedando cualquier aspecto visual a cargo de las hojas de estilo.

En este caso, Django necesita una forma conveniente de generar HTML dinámicamente. El enfoque más común se basa en plantillas. Una plantilla contiene las partes estáticas del resultado HTML deseado, así como también una sintaxis especial que describe cómo se insertará el contenido dinámico. Además, Django define una

API estándar para cargar y representar plantillas independientemente del *back-end*. Dicha carga consiste en encontrar la plantilla para un identificador dado y procesarla, usualmente compilando en una representación en memoria.

En la siguiente figura se muestra un pequeño fragmento de la vista perteneciente a este proyecto para observar el comportamiento de Django:

```

<ul class="children first-child">
  {% for conference in conferences %}
  <li class="expandable hit-{{ conference.domain }}-li">
    <div class="hitarea hit-{{ conference.domain }}-area"></div>
    <a href="/biblioteca/conferencias/{{ conference.domain }}" class="{{ conference.domain }}">{{ conference.name }}</a>
    <ul class="children child-{{ conference.domain }}">
      {% for edition in editions %}
      {% if edition.conference_id.id == conference.id %}
      <li class="expandable hit-{{ edition.conference_id.domain }}-{{ edition.year }}-li">
        <div class="hitarea hit-{{ edition.conference_id.domain }}-{{ edition.year }}-area" ></div>
        <a href="/biblioteca/conferencias/{{ edition.conference_id.domain }}/{{ edition.year }}"
        class="{{ edition.conference_id.domain }}-{{ edition.year }}">{{ edition.name }}</a>
        <ul class="children child-{{ edition.conference_id.domain }}-{{ edition.year }}">
          {% for track in tracks %}
          {% if track.edition_id.id == edition.id %}
          <li>
            <a href="/biblioteca/conferencias/
            {{ edition.conference_id.domain }}/{{ edition.year }}/{{ track.id }}">{{ track.name }}</a>
          </li>
          {% endif %}
          {% endfor %}
        </ul>
      </li>
      {% endif %}
      {% endfor %}
    </ul>
  </li>
  {% endfor %}
</ul>

```

{{ conference.name }} Variable de contexto
 Estructura condicional

Ilustración 27. Fragmento de código de la vista.

Una plantilla contiene **variables**, que se reemplazan con valores cuando se evalúa la plantilla, y **etiquetas**, que controlan la lógica de la plantilla.

6.1.3 JavaScript

Se trata de un lenguaje multiparadigma: funcional, imperativo y basado en prototipos (en lugar de clases).

Prácticamente la totalidad de navegadores modernos son capaces de interpretar código JavaScript [14].

6.2 Herramienta de desarrollo

En particular, este proyecto ha sido implementado por un entorno de desarrollo integrado conocido como PyCharm [15], un entorno diseñado principalmente para la construcción de aplicaciones cuya lógica se define con el lenguaje de programación Python.

6.3 Interfaz gráfica

A continuación, se muestran algunas capturas de pantalla que reflejan el estado final de la presentación.

6.3.1 Pantalla de Inicio

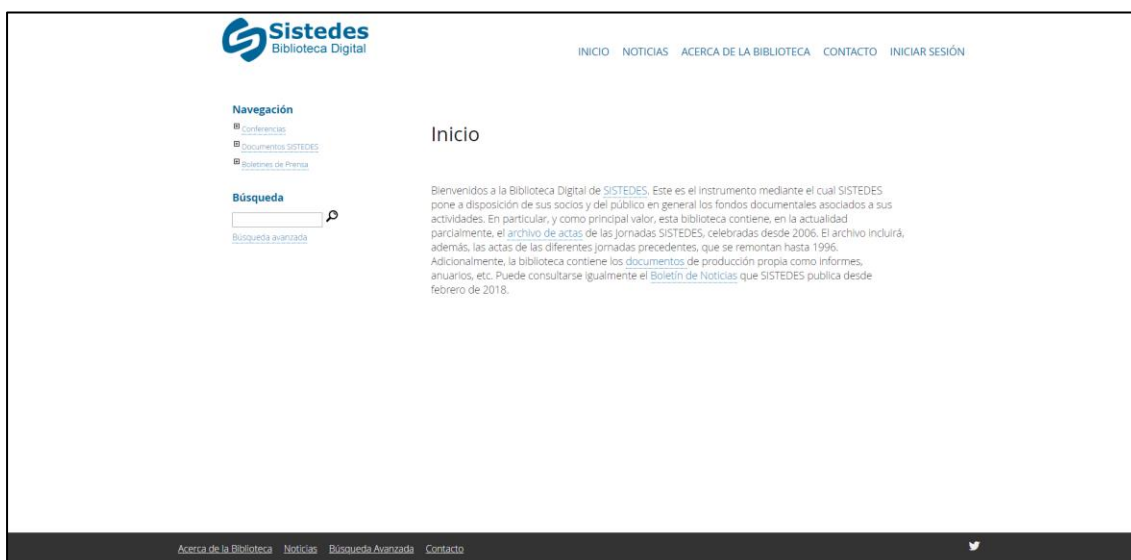


Ilustración 28. Pantalla de inicio -Vista.

6.3.2 Pantalla de Inicio de sesión

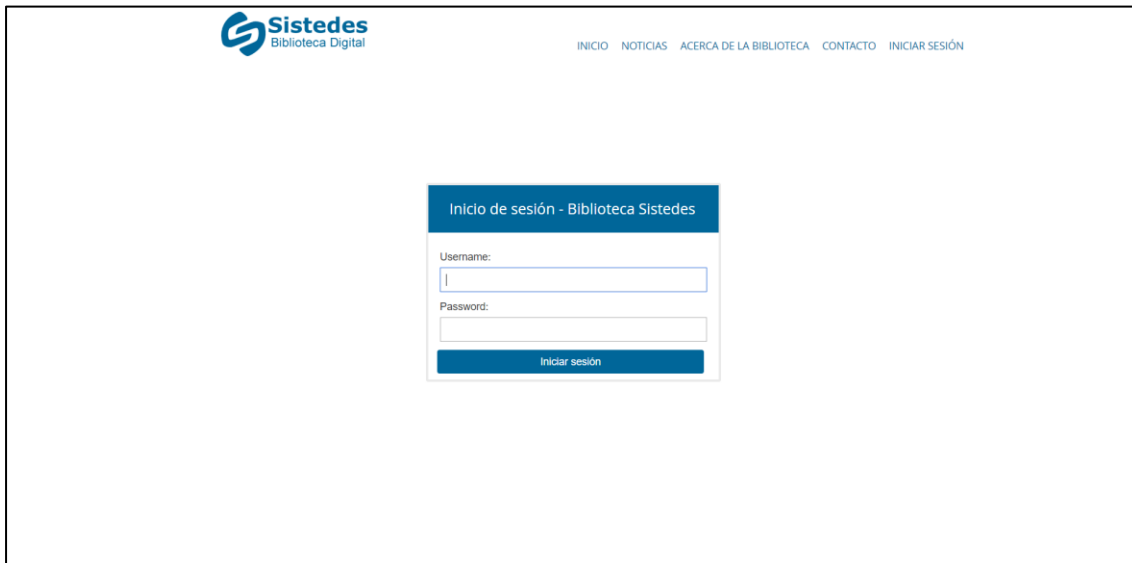


Ilustración 29. Pantalla de Inicio de sesión -Vista.

6.3.3 Menú personal

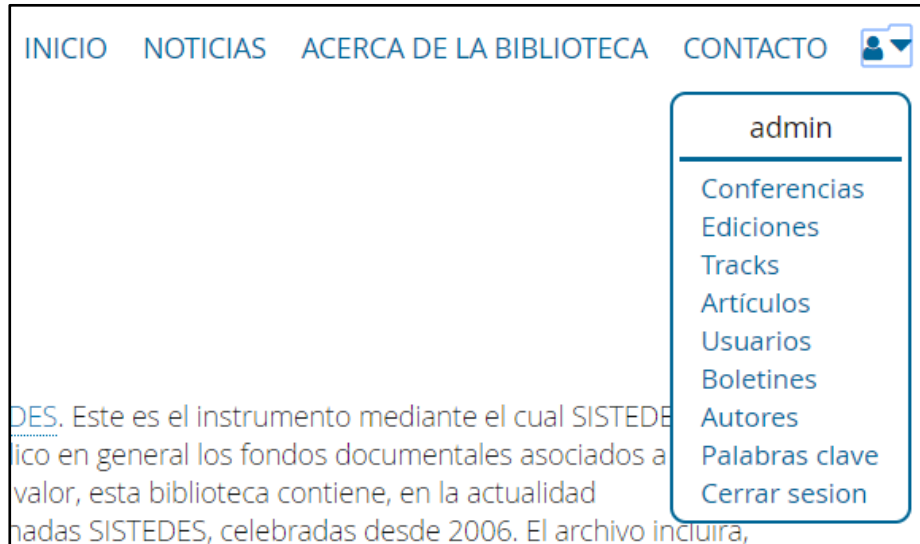


Ilustración 30. Menú personal -Vista.

6.3.4 Pantalla de creación de una conferencia

The screenshot shows the 'Crear conferencia' form in the Sistedes system. The form is titled 'Crear conferencia' and contains the following fields and elements:

- Nombre:** A text input field.
- Abreviatura:** A text input field.
- Descripción:** A large text area with a rich text editor toolbar above it.
- Buttons:** 'Guardar conferencia' (blue) and 'Cancelar' (red).

The page header includes the Sistedes logo and navigation links: INICIO, NOTICIAS, ACERCA DE LA BIBLIOTECA, CONTACTO.

Ilustración 31. Pantalla de creación de una conferencia -Vista.

6.3.5 Pantalla del listado de conferencias

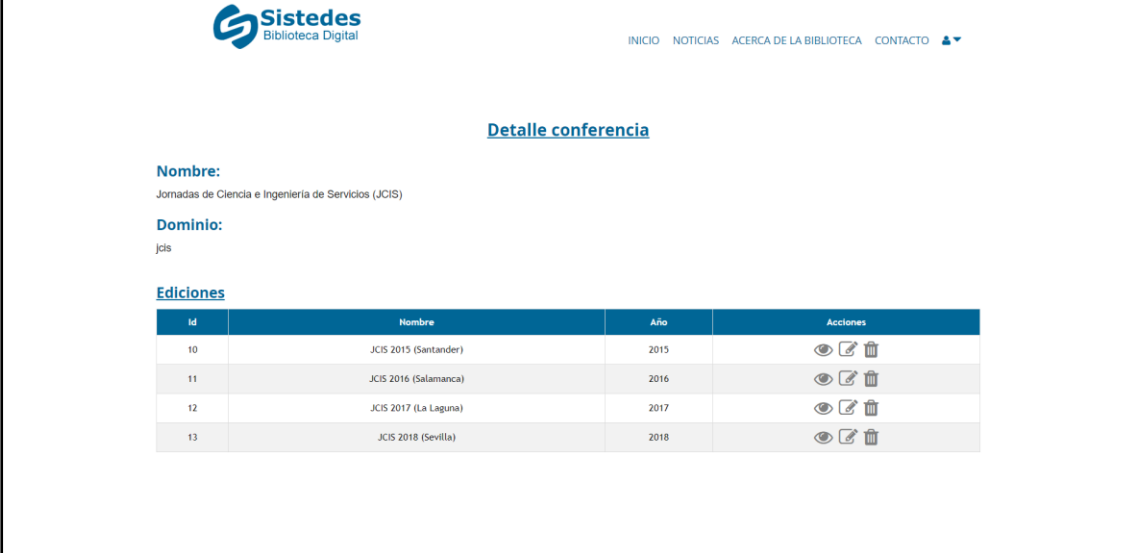
The screenshot shows the 'Conferencias' list view in the Sistedes system. The page features a search bar, an 'Añadir +' button, and a table with the following data:

Nombre	Dominio	Acciones
Jornadas de Ciencia e Ingeniería de Servicios (JCIS)	jcis	
Jornadas de Ingeniería del Software y Bases de Datos (JISBD)	JISBD	
Jornadas sobre Programación y Lenguajes (PROLE)	prole	

The page header includes the Sistedes logo and navigation links: INICIO, NOTICIAS, ACERCA DE LA BIBLIOTECA, CONTACTO.

Ilustración 32. Pantalla del listado de conferencias -Vista.

6.3.6 Pantalla de detalle de una conferencia



Sistedes
Biblioteca Digital

INICIO NOTICIAS ACERCA DE LA BIBLIOTECA CONTACTO

Detalle conferencia

Nombre:
Jornadas de Ciencia e Ingeniería de Servicios (JCIS)

Dominio:
jcis

Ediciones





Id	Nombre	Año	Acciones
10	JCIS 2015 (Santander)	2015	  
11	JCIS 2016 (Salamanca)	2016	  
12	JCIS 2017 (La Laguna)	2017	  
13	JCIS 2018 (Sevilla)	2018	  

Ilustración 33. Pantalla de detalle de una conferencia -Vista.

7. Conclusiones y trabajos futuros

En este capítulo se detallarán las conclusiones de este Trabajo Final de Máster, así como los posibles trabajos futuros que se pueden llevar a cabo.

7.1 Conclusiones

El objetivo principal de este Trabajo Final de Máster consistía en poder llevar a cabo una nueva implementación de la Biblioteca Digital de SISTEDES que ofreciera todas las funcionalidades que ya brindaba la versión original.

Dicha implementación ha sido desarrollada de acuerdo con la especificación de los requerimientos que se establecieron al principio. Como resultado se ha obtenido una nueva versión de la Biblioteca Digital de SISTEDES desarrollada con una tecnología distinta que cumple con dichos requisitos y que es capaz de contemplar todos los puntos que se marcaron en la fase de análisis.

En una primera toma, se planteó el comportamiento de esta biblioteca digital estableciendo los requisitos funcionales y los no funcionales, los diagramas de casos de uso y clases UML para poder visualizar provisionalmente cuál sería el comportamiento de la biblioteca. Esta fase se podría considerar la parte más decisiva de todo el proceso de desarrollo.

Posteriormente, se estableció el diseño de la biblioteca. Esta se basaba en la definición de la arquitectura principal del software Modelo Vista Controlador, del *framework* Django y finalmente, de los componentes de dicha arquitectura.

La planificación y la definición de estas fases han sido los aspectos que más han aportado a la implementación de la Biblioteca Digital de SISTEDES. El hecho de no empezar directamente con el desarrollo y poder establecer una planificación previa, ha permitido finalizar dicho desarrollo de forma más eficiente y eficaz.

Respecto a los conocimientos adquiridos, se ha podido llevar a la práctica algunos de los aspectos relacionados con la gestión de contenido digital. Concretamente, se ha podido definir una arquitectura basada en *Kahn/Wilensky framework* y organizar todo el contenido de la biblioteca digital adecuadamente.

Además, la utilización de la arquitectura Modelo Vista Controlador ha permitido entender mejor la utilidad de estos planteamientos en el campo de la arquitectura del *software* y sus implicaciones para el resto de la implementación de la biblioteca digital.

Por otra parte, en cuanto a la tecnología empleada, cabe destacar que Django es un *framework* bastante complejo que requiere de un proceso de formación previo y al que es necesario emplear mucho tiempo si se pretende conocer cómo actúa. Sin embargo, gracias al tiempo dedicado, se ha llegado a incrementar los conocimientos sobre dicho *framework*. Aunque la mayoría del tiempo de aprendizaje se ha utilizado en conocer Django, también se ha conseguido aumentar los conocimientos sobre el resto de la

tecnología empleada. Y, aunque dicha tecnología presenta limitaciones que pueden complicar la implementación de la biblioteca digital, no ha resultado un gran inconveniente para dicha implementación.

7.2 Trabajos futuros

Aunque la nueva versión de la Biblioteca Digital de SISTEDES contempla todos los requisitos establecidos en la fase de análisis, a continuación, se detallan posibles ampliaciones que podría ofrecer.

Actualmente, la biblioteca digital dispone de consulta del catálogo a través de metadatos. Una posible ampliación sería implementar una nueva funcionalidad que permitiese buscar por contenido los artículos de la Biblioteca Digital de SISTEDES.

Además, la Biblioteca Digital de SISTEDES ha sido desarrollada en español sin posibilidad de cambiar a otro idioma. Por lo tanto, ofrecer la funcionalidad de traducir el texto a otros idiomas se puede considerar posible ampliación.

Aunque, la Biblioteca Digital de SISTEDES ofrece una búsqueda avanzada en la cual puedes especificar algunos metadatos, se podría llevar a cabo la implementación de una búsqueda más extensa que permitiese consultar el catálogo con otro tipo de metadatos como fecha de creación, identificadores *handle*...

Por último, una de las actualizaciones que también podría llevarse a cabo es la adaptación de la biblioteca digital a cualquier dispositivo para que ésta pueda ser accesible desde ordenadores, dispositivos móviles...

REFERENCIAS

- [1] UNIVERSITAT OBERTA DE CATALUNYA. *La Biblioteca Digital*. <http://www.uoc.edu/web/esp/articles/La_biblioteca_digital.htm> [Consulta: 14 de junio de 2018]
- [2] WIKIPEDIA. *Biblioteca Digital*. <https://es.wikipedia.org/wiki/Biblioteca_digital> [Consulta: 14 de junio de 2018]
- [3] SISTEDES. *Sociedad de Ingeniería de Software y Tecnologías de Desarrollo de Software*. <<https://www.sistedes.es/>> [Consulta: 14 de junio de 2018]
- [4] WIKIPEDIA. *Proceso Unificado de Rational*. <https://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational> [Consulta: 23 de julio de 2018]
- [5] DJANGO SOFTWARE FOUNDATION. *Django: The Web framework for perfectionists with deadlines*. <<https://www.djangoproject.com/>> [Consulta: 04 de agosto de 2018]
- [6] UNIVERSIDAD DE ALICANTE. *Modelo Vista Controlador*. <<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>> [Consulta: 04 de agosto de 2018]
- [7] PYTHON SOFTWARE FOUNDATION. *Python*. <<https://www.python.org/>> [Consulta: 20 de julio de 2018]
- [8] WIKIPEDIA. *Lenguaje unificado de modelado*. <https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado> [Consulta: 27 de julio de 2018]
- [9] W3SCHOOLS. *CSS Tutorial*. <<https://www.w3schools.com/css/default.asp>> [Consulta: 21 de agosto de 2018]
- [10] WIKIPEDIA. *Metodología de desarrollo de software*. <https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software> [Consulta: 23 de julio de 2018]
- [11] TUTORIALSPPOINT. *Django Useful Resources*. <https://www.tutorialspoint.com/django/django_overview.htm> [Consulta: 04 de agosto de 2018]
- [12] POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL*. <<https://www.postgresql.org/>> [Consulta: 09 de agosto de 2018]
- [13] WIKIPEDIA. *HTML*. <<https://es.wikipedia.org/wiki/HTML>> [Consulta: 02 de agosto de 2018]
- [14] CORP. MOZILLA FOUNDATION. *JavaScript*. [Consulta: 14 de agosto de 2018]

[15] JETBRAINS. *PyCharm*. <<https://www.jetbrains.com/pycharm/>> [Consulta: 26 de julio de 2013]

[16] CORPORATION FOR NATIONAL RESEARCH INITIATIVES. *A Framework for Distributed Digital Object Services*. < <http://www.cnri.reston.va.us/k-w.html> > [Consulta: 10 de julio de 2013]

ANEXO A: ESPECIFICACIÓN DE CASOS DE USO

Tabla 6. Caso de uso: Crear conferencia.

Caso de uso	Crear conferencia
Actores	Administrador general
Resumen	El usuario crea una nueva conferencia para la biblioteca.
Precondiciones	1. El usuario está autenticado como Administrador general.
Postcondiciones	1. Hay una nueva conferencia creada para la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear una nueva conferencia. 3. El usuario establece los metadatos y guarda los cambios.	2. El sistema le muestra al usuario todos los metadatos que tienen que establecer. 4. El sistema almacena en la biblioteca la nueva conferencia.

Tabla 7. Caso de uso: Crear artículo.

Caso de uso	Crear artículo
Actores	<i>Track chair</i>
Resumen	El usuario crea un nuevo artículo para la biblioteca.
Precondiciones	1. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	1. Hay un nuevo artículo creado para la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear un nuevo artículo. 3. El usuario establece los metadatos y guarda los cambios.	2. El sistema le indica al usuario todos los metadatos que tienen que establecer. 4. El sistema almacena en el repositorio el artículo creado.

Tabla 8. Caso de uso: Modificar artículo.

Caso de uso	Modificar artículo
Actores	<i>Track chair</i>
Resumen	El usuario modifica un artículo ya existente en el repositorio.
Precondiciones	1. El artículo ya ha de estar creado. 2. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	1. El artículo ha sido modificado.

Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de modificar un nuevo artículo. 3. El usuario cambia los metadatos que considera y guarda los cambios.	2. El sistema le indica al usuario todos los metadatos que están disponibles para modificar dicho artículo. 4. El sistema publica el artículo modificado.

Tabla 9. Caso de uso: Eliminar artículo.

Caso de uso	Eliminar artículo
Actores	<i>Track chair</i>
Resumen	El usuario elimina un artículo ya existente en el repositorio.
Precondiciones	1. El artículo ya ha de estar creado. 2. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	1. El artículo ha sido eliminado.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de eliminar un nuevo artículo. 3. El usuario selecciona el artículo que quiere eliminar.	2. El sistema le muestra al usuario todos los artículos para que elija el artículo que se desea eliminar. 4. El sistema elimina del repositorio el artículo seleccionado.

Tabla 10. Caso de uso: Buscar artículo.

Caso de uso	Buscar artículo
Actores	Visitante
Resumen	El usuario busca un artículo en concreto.
Precondiciones	-
Postcondiciones	Se muestra por pantalla el artículo que se busca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de buscar un artículo. 3. El usuario escribe en el formulario de búsqueda el texto con el que quiere encontrar el artículo.	2. El sistema le muestra al usuario un formulario de búsqueda. 4. El sistema le muestra al usuario los artículos cuyos metadatos coinciden con el texto insertado en el formulario de búsqueda.

Tabla 11. Caso de uso: Iniciar sesión.

Caso de uso	Iniciar sesión
Actores	<i>Track chair</i>
Resumen	El usuario inicia sesión en el sistema con sus credenciales.
Precondiciones	1. El usuario debe estar registrado en el sistema previamente.
Postcondiciones	1. El usuario ha iniciado sesión en el sistema.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección para iniciar sesión en el sistema. 3. El usuario rellena el nombre de usuario y la contraseña que se muestran en pantalla.	2. El sistema le muestra al usuario un formulario para que inserte el nombre de usuario y la contraseña. 4. El sistema redirige al usuario a la pantalla de inicio si el usuario existe y ha introducido correctamente las credenciales.

Tabla 12. Caso de uso: Crear *track*.

Caso de uso	Crear <i>track</i>
Actores	<i>P. C. Chair</i>
Resumen	El usuario crea un nuevo <i>track</i> para la biblioteca.
Precondiciones	1. El usuario está autenticado como <i>P. C. Chair</i> .
Postcondiciones	1. Hay un nuevo <i>track</i> creado en la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear un nuevo <i>track</i> . 3. El usuario establece los metadatos y guarda los cambios.	2. El sistema le indica al usuario todos los metadatos que tienen que establecer. 4. El sistema almacena en la biblioteca el <i>track</i> creado.

Tabla 13. Caso de uso: Modificar *track*.

Caso de uso	Modificar <i>track</i>
Actores	<i>P. C. Chair</i>
Resumen	El usuario modifica un <i>track</i> ya existente.
Precondiciones	1. El <i>track</i> ya ha de estar creado. 2. El usuario está autenticado como <i>P. C. Chair</i> .
Postcondiciones	1. El <i>track</i> ha sido modificado.
Incluye	-
Extiende	-

Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de modificar un nuevo <i>track</i> . 3. El usuario establece los metadatos que considera y guarda los cambios.	2. El sistema le indica al usuario todos los metadatos que están disponibles para modificar dicho <i>track</i> . 4. El sistema almacena en la biblioteca el <i>track</i> modificado.

Tabla 14. Caso de uso: Eliminar *track*.

Caso de uso	Eliminar <i>track</i>
Actores	<i>P. C. Chair</i>
Resumen	El usuario elimina un <i>track</i> ya existente.
Precondiciones	1. El <i>track</i> ya ha de estar creado. 2. El usuario está autenticado como <i>P. C. Chair</i> .
Postcondiciones	1. El <i>track</i> ha sido eliminado.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de eliminar un nuevo <i>track</i> . 3. El usuario selecciona el <i>track</i> que quiere eliminar.	2. El sistema le muestra al usuario todos los <i>tracks</i> para que elija el aquel que desea eliminar. 4. El sistema elimina de la biblioteca el <i>track</i> seleccionado.

Tabla 15. Caso de uso: Crear edición.

Caso de uso	Crear edición
Actores	<i>Administrador general</i>
Resumen	El usuario crea una nueva edición para la biblioteca.
Precondiciones	1. El usuario está autenticado como Administrador general.
Postcondiciones	1. Hay una nueva edición creado para la biblioteca.
Incluye	Asignar responsable de <i>track</i>
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear una nueva edición. 3. El usuario establece los metadatos que se indican y guarda los cambios.	2. El sistema le indica al usuario todos los metadatos que tienen que rellenar. 4. El sistema almacena en la biblioteca la nueva edición.

Tabla 16. Caso de uso: Modificar edición.

Caso de uso	Modificar edición
Actores	Administrador general
Resumen	El usuario modifica una edición ya

	existente.
Precondiciones	1. La edición ya ha de estar creada.
Postcondiciones	1. La edición ha sido modificada. 2. El usuario está autenticado como Administrador general.
Incluye	-
Extiende	Asignar responsable de <i>track</i>
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de modificar una nueva edición. 3. El usuario cambia metadatos que considera y guarda los cambios.	2. El sistema le indica al usuario todos los metadatos que están disponibles para modificar dicha edición. 4. El sistema almacena en la biblioteca la edición modificada.

Tabla 17. Caso de uso: Eliminar edición.

Caso de uso	Eliminar edición
Actores	Administrador general
Resumen	El usuario elimina una edición ya existente.
Precondiciones	1. La edición ya ha de estar creada. 2. El usuario está autenticado como Administrador general.
Postcondiciones	1. La edición ha sido eliminada.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de eliminar una nueva edición. 3. El usuario selecciona la edición que quiere eliminar.	2. El sistema le muestra al usuario todas las ediciones para que elija la que se desea eliminar. 4. El sistema elimina de la biblioteca la edición seleccionada.

Tabla 18. Caso de uso: Asignar responsable de *track*.

Caso de uso	Asignar responsable de <i>track</i>
Actores	Administrador general
Resumen	El administrador general asigna un nuevo responsable de <i>track</i> a un artículo.
Precondiciones	1. El responsable de <i>track</i> debe estar previamente registrado. 2. Debe existir el artículo en cuestión. 3. El usuario está autenticado como administrador general.
Postcondiciones	1. El artículo tiene asignado un nuevo responsable de <i>track</i> .
Incluye	-
Extiende	-
Flujo de eventos	

Usuario	Sistema
1. El usuario se dirige a la sección de crear o modificar un artículo. 3. El usuario establece un nuevo responsable de track.	2. El sistema le muestra al usuario todos los metadatos para establecer. 4. Se guarda en el repositorio el artículo en cuestión

Tabla 19. Caso de uso: Crear conferencia.

Caso de uso	Crear conferencia
Actores	Administrador general
Resumen	El usuario crea una nueva conferencia para la biblioteca.
Precondiciones	1. El usuario está autenticado como Administrador general.
Postcondiciones	1. Hay una nueva conferencia creada para la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear una nueva conferencia. 3. El usuario establece los metadatos que se muestran en pantalla y guarda los cambios.	2. El sistema le muestra al usuario todos los metadatos que tienen que establecer. 4. El sistema almacena en la biblioteca la nueva conferencia.

Tabla 20. Caso de uso: Modificar conferencia.

Caso de uso	Modificar conferencia
Actores	Administrador general
Resumen	El usuario modifica una conferencia ya existente.
Precondiciones	1. La conferencia ya ha de estar creada. 2. El usuario está autenticado como Administrador general.
Postcondiciones	1. La conferencia ha sido modificada.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de modificar una nueva conferencia. 3. El usuario cambia los metadatos que considera y guarda los cambios.	2. El sistema le muestra al usuario todos los metadatos que están disponibles para modificar dicha conferencia. 4. El sistema almacena en la biblioteca la conferencia modificada.

Tabla 21. Caso de uso: Eliminar conferencia.

Caso de uso	Eliminar conferencia
Actores	Administrador general
Resumen	El usuario elimina una conferencia ya existente.
Precondiciones	<ol style="list-style-type: none"> 1. La conferencia ya ha de estar creada. 2. El usuario está autenticado como Administrador general.
Postcondiciones	<ol style="list-style-type: none"> 1. La conferencia ha sido eliminada.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
<ol style="list-style-type: none"> 1. El usuario se dirige a la sección de eliminar una nueva conferencia. 3. El usuario selecciona la conferencia que quiere eliminar. 	<ol style="list-style-type: none"> 2. El sistema le muestra al usuario todas las conferencias para que elija la que se desea eliminar. 4. El sistema elimina de la biblioteca la conferencia seleccionada.

Tabla 22. Caso de uso: Administrar usuarios.

Caso de uso	Administrar usuarios
Actores	Administrador general
Resumen	El administrador general gestiona los usuarios para la elaboración de las conferencias.
Precondiciones	<ol style="list-style-type: none"> 1. El usuario está autenticado como Administrador general.
Postcondiciones	<ol style="list-style-type: none"> 1. Un usuario es gestionado, es decir, creado, modificado o el eliminado.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
<ol style="list-style-type: none"> 1. El usuario se dirige a la sección de registrar un nuevo usuario o modificar/eliminar éste. 3. El usuario efectúa dichos pasos y guarda los cambios. 	<ol style="list-style-type: none"> 2. El sistema le muestra al usuario todos los pasos que debe de seguir. 4. El sistema almacena el usuario creado, modificado o eliminado.

Tabla 23. Caso de uso: Buscar artículo.

Caso de uso	Buscar artículo
Actores	Visitante
Resumen	El visitante busca un artículo a través de un formulario de búsqueda.
Precondiciones	-

Postcondiciones	1. El sistema muestra por pantalla el artículo que se busca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de buscar un artículo. 3. El usuario escribe en el formulario de búsqueda el texto que considere para buscar entre los metadatos del <i>track</i> .	2. El sistema le muestra al usuario un formulario de búsqueda. 4. El sistema muestra por pantalla los artículos que algunos de sus metadatos coinciden con el texto insertado en el formulario de búsqueda.

Tabla 24. Caso de uso: Ver conferencia.

Caso de uso	Ver conferencia
Actores	Visitante
Resumen	El visitante ve todas las conferencias que existen en la biblioteca.
Precondiciones	-
Postcondiciones	1. El sistema muestra por pantalla todas las conferencias existentes.
Incluye	-
Extiende	Ver edición
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de las conferencias.	2. El sistema le muestra al usuario todas las conferencias que existen en la biblioteca.

Tabla 25. Caso de uso: Ver edición.

Caso de uso	Ver edición
Actores	Visitante
Resumen	El visitante ve todas las ediciones que existen en la biblioteca.
Precondiciones	-
Postcondiciones	1. El sistema muestra por pantalla todas las ediciones existentes.
Incluye	-
Extiende	Ver <i>track</i>
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de las ediciones.	2. El sistema le muestra al usuario todas las ediciones que existen en la biblioteca.

Tabla 26. Caso de uso: Ver *track*.

Caso de uso	Ver <i>track</i>
Actores	Visitante
Resumen	El visitante ve todos los <i>tracks</i> que existen

	en la biblioteca.
Precondiciones	-
Postcondiciones	1. El sistema muestra por pantalla todos los <i>tracks</i> existentes.
Incluye	-
Extiende	Ver artículo
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de los <i>tracks</i> .	2. El sistema le muestra al usuario todos los <i>tracks</i> que existen en la biblioteca.

Tabla 27. Caso de uso: Ver artículo.

Caso de uso	Ver artículo
Actores	Visitante
Resumen	El visitante ve todos los artículos que existen en la biblioteca.
Precondiciones	-
Postcondiciones	1. El sistema muestra por pantalla todos los artículos existentes.
Incluye	-
Extiende	Descargar artículo
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de los artículos.	2. El sistema le muestra al usuario todos los artículos que existen en el repositorio.

Tabla 28. Caso de uso: Descargar artículo.

Caso de uso	Descargar artículo
Actores	Visitante
Resumen	El visitante descarga un artículo en concreto.
Precondiciones	1. Debe haber un artículo seleccionado.
Postcondiciones	1. El artículo en cuestión es descargado.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de los artículos. 3. El usuario accede al enlace que se le ofrece.	2. El sistema ofrece un enlace para descargar el artículo. 4. El sistema facilita dicho el fichero artículo.

Tabla 29. Caso de uso: Ver conferencia.

Caso de uso	Ver conferencia
Actores	Administrador general
Resumen	El usuario accede a la vista detalle a partir del listado de todas las conferencias.
Precondiciones	<ol style="list-style-type: none"> 1. La conferencia a visualizar es previamente creada. 2. El usuario está autenticado como Administrador general.
Postcondiciones	<ol style="list-style-type: none"> 1. Nueva vista con los detalles de la conferencia seleccionada.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
<ol style="list-style-type: none"> 1. El usuario se dirige a la sección del listado de todas las conferencias existentes y pulsa sobre la conferencia que desea visualizar. 	<ol style="list-style-type: none"> 2. El sistema le muestra al usuario todos los metadatos pertenecientes a la conferencia en cuestión con sus respectivos valores.

Tabla 30. Caso de uso: Ver edición.

Caso de uso	Ver edición
Actores	Administrador general
Resumen	El usuario accede a la vista detalle a partir del listado de todas las ediciones.
Precondiciones	<ol style="list-style-type: none"> 1. La edición a visualizar es previamente creada. 2. El usuario está autenticado como Administrador general.
Postcondiciones	<ol style="list-style-type: none"> 1. Nueva vista con los detalles de la edición seleccionada.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
<ol style="list-style-type: none"> 1. El usuario se dirige a la sección del listado de todas las ediciones existentes y pulsa sobre la edición que desea visualizar. 	<ol style="list-style-type: none"> 2. El sistema le muestra al usuario todos los metadatos pertenecientes a la edición en cuestión con sus respectivos valores.

Tabla 31. Caso de uso: Ver *track*.

Caso de uso	Ver <i>track</i>
Actores	P. C. Chair
Resumen	El usuario accede a la vista detalle a partir del listado de todos los <i>tracks</i> .
Precondiciones	<ol style="list-style-type: none"> 1. El <i>track</i> a visualizar es previamente creado. 2. El usuario está autenticado como P. C. Chair..

Postcondiciones	1. Nueva vista con los detalles del <i>track</i> seleccionado.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección del listado de todos los <i>tracks</i> existentes y pulsa sobre aquel que desea visualizar.	2. El sistema le muestra al usuario todos los metadatos pertenecientes al <i>track</i> en cuestión con sus respectivos valores.

Tabla 32. Caso de uso: Ver artículo.

Caso de uso	Ver artículo
Actores	<i>Track chair</i>
Resumen	El usuario accede a la vista detalle a partir del listado de todos los artículos.
Precondiciones	<ol style="list-style-type: none"> 1. El artículo a visualizar es previamente creado. 2. El usuario está autenticado como <i>Track chair</i>.
Postcondiciones	1. Nueva vista con los detalles del artículo seleccionado.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección del listado de todos los artículos existentes y pulsa sobre aquel que desea visualizar.	2. El sistema le muestra al usuario todos los metadatos pertenecientes al artículo en cuestión con sus respectivos valores.

Tabla 33. Caso de uso: Listar conferencias.

Caso de uso	Listar conferencias
Actores	Administrador general
Resumen	El usuario accede a la vista del listado de todas las conferencias.
Precondiciones	1. El usuario está autenticado como Administrador general.
Postcondiciones	1. Nueva vista con el listado de todas las conferencias existentes en la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección del listado de todas las conferencias existentes.	2. El sistema le muestra al usuario el listado de todas las conferencias con algunos de sus metadatos.

Tabla 34. Caso de uso: Listar ediciones.

Caso de uso	Listar ediciones
Actores	Administrador general
Resumen	El usuario accede a la vista del listado de todas las ediciones.
Precondiciones	1. El usuario está autenticado como Administrador general.
Postcondiciones	1. Nueva vista con el listado de todas las ediciones existentes en la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección del listado de todas las ediciones existentes.	2. El sistema le muestra al usuario el listado de todas las ediciones con algunos de sus metadatos.

Tabla 35. Caso de uso: Listar *tracks*.

Caso de uso	Listar <i>tracks</i>
Actores	<i>P. C. Chair</i>
Resumen	El usuario accede a la vista del listado de todos los <i>tracks</i> .
Precondiciones	1. El usuario está autenticado como <i>P. C. Chair</i> .
Postcondiciones	1. Nueva vista con el listado de todos los <i>tracks</i> existentes en la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección del listado de todos los <i>tracks</i> existentes.	2. El sistema le muestra al usuario el listado de todos los <i>tracks</i> con algunos de sus metadatos.

Tabla 36. Caso de uso: Listar artículos.

Caso de uso	Listar artículos
Actores	<i>Track chair</i>
Resumen	El usuario accede a la vista del listado de todos los artículos.
Precondiciones	1. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	1. Nueva vista con el listado de todos los artículos existentes en el repositorio.
Incluye	-

Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección del listado de todos los artículos existentes.	2. El sistema le muestra al usuario el listado de todos los artículos con algunos de sus metadatos.

Tabla 37. Caso de uso: Crear autor.

Caso de uso	Crear autor
Actores	<i>Track chair</i>
Resumen	El usuario crea un nuevo autor para la biblioteca.
Precondiciones	1. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	1. Hay un nuevo autor creada para la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear un nuevo autor. 3. El usuario establece los metadatos que se muestran en pantalla y guarda los cambios.	2. El sistema le muestra al usuario todos los metadatos que tienen que establecer. 4. El sistema almacena en la biblioteca el nuevo autor.

Tabla 38. Caso de uso: Modificar autor.

Caso de uso	Modificar autor
Actores	<i>Track chair</i>
Resumen	El usuario modifica un autorya existente.
Precondiciones	1. El autor ya ha de estar creado. 2. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	1. El autor ha sido modificada.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de modificar un nuevo autor. 3. El usuario cambia los metadatos que considera y guarda los cambios.	2. El sistema le muestra al usuario todos los metadatos que están disponibles para modificar dicho autor. 4. El sistema almacena en la biblioteca el autor modificado.

Tabla 39. Caso de uso: Eliminar autor.

Caso de uso	Eliminar autor
Actores	<i>Track Chair</i>
Resumen	El usuario elimina un autor ya existente.

Precondiciones	1. El autor ya ha de estar creado. 2. El usuario está autenticado como <i>track chair</i> .
Postcondiciones	2. El autor ha sido eliminado.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de eliminar un nuevo autor. 3. El usuario selecciona el autor que quiere eliminar.	2. El sistema le muestra al usuario todos los autores para que elija el aquel que desea eliminar. 4. El sistema elimina el autor seleccionado.

Tabla 40. Caso de uso: Listar autores.

Caso de uso	Listar autores
Actores	<i>Track chair</i>
Resumen	El usuario accede a la vista del listado de todos los autores.
Precondiciones	1. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	2. Nueva vista con el listado de todos los autores existentes en la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección del listado de todos los autores existentes.	2. El sistema le muestra al usuario el listado de todos los autores con algunos de sus metadatos.

Tabla 41. Caso de uso: Crear palabra clave.

Caso de uso	Crear palabra clave
Actores	<i>Track chair</i>
Resumen	El usuario crea una nueva palabra clave para la biblioteca.
Precondiciones	1. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	1. Hay una nueva palabra clave creada para la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección de crear una nueva palabra clave. 3. El usuario establece los metadatos que se muestran en pantalla y guarda los cambios.	2. El sistema le muestra al usuario todos los metadatos que tienen que establecer. 4. El sistema almacena en la biblioteca el nuevo autor.

Tabla 42. Caso de uso: Modificar palabra clave.

Caso de uso	Modificar palabra clave
Actores	<i>Track chair</i>
Resumen	El usuario modifica una palabra clave ya existente.
Precondiciones	<ol style="list-style-type: none"> 1. La palabra clave ya ha de estar creado. 2. El usuario está autenticado como <i>Track chair</i>.
Postcondiciones	<ol style="list-style-type: none"> 1. La palabra clave ha sido modificada.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
<ol style="list-style-type: none"> 1. El usuario se dirige a la sección de modificar una nueva palabra clave. 3. El usuario cambia los metadatos que considera y guarda los cambios. 	<ol style="list-style-type: none"> 2. El sistema le muestra al usuario todos los metadatos que están disponibles para modificar dicha palabra clave. 4. El sistema almacena en la biblioteca la nueva palabra clave modificada.

Tabla 43. Caso de uso: Eliminar palabra clave.

Caso de uso	Eliminar palabra clave
Actores	<i>Track Chair</i>
Resumen	El usuario elimina una palabra ya existente.
Precondiciones	<ol style="list-style-type: none"> 1. La palabra clave ya ha de estar creado. 2. El usuario está autenticado como <i>track chair</i>.
Postcondiciones	<ol style="list-style-type: none"> 1. La palabra clave ha sido eliminada.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
<ol style="list-style-type: none"> 1. El usuario se dirige a la sección de eliminar una nueva palabra clave. 3. El usuario selecciona la palabra clave que quiere eliminar. 	<ol style="list-style-type: none"> 2. El sistema le muestra al usuario todas las palabras clave para que elija aquella que desea eliminar. 4. El sistema elimina la palabra clave seleccionada.

Tabla 44. Caso de uso: Listar palabras clave.

Caso de uso	Listar palabras clave
Actores	<i>Track chair</i>
Resumen	El usuario accede a la vista del listado de todos las palabras clave.

Precondiciones	1. El usuario está autenticado como <i>Track chair</i> .
Postcondiciones	2. Nueva vista con el listado de todas las palabras clave existentes en la biblioteca.
Incluye	-
Extiende	-
Flujo de eventos	
Usuario	Sistema
1. El usuario se dirige a la sección del listado de todas las palabras clave existentes.	2. El sistema le muestra al usuario el listado de todas las palabras clave con algunos de sus metadatos.