



**Facultad de Ciencias y Tecnologías Computacionales**

“Repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa en la Universidad de las Ciencias Informáticas”

**Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas**

**Autor:** Yanety Milagros González Zayas

**Tutores:** Dr. C. Armando Leyva Pérez

Dr. C. Omar Mar Cornelio

La Habana, octubre de 2023

“Año 65 de la Revolución”



*“(...) porque para defender un país lo más importante es la disposición que los hijos de ese país han demostrado, el patriotismo, (...) la decisión de luchar hasta la última gota de sangre, la decisión de combatir (...)”<sup>1</sup>*

**Fidel Castro Ruz**

---

<sup>1</sup> Castro Ruz Fidel, “Sobre Temas Militares”, 1990, Tomo I, Pág. 95 (Palabras en el ascenso a oficiales”, 30.12.1962).

## Declaración de autoría

La autora del trabajo de diploma con título: ***“Repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa en la Universidad de las Ciencias Informáticas”***, concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como única autora de su contenido. Para que así conste firma la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_ .

---

Firma de la autora  
Yanety Milagros González Zayas

---

Firma del tutor  
Dr. C. Armando Leyva Pérez

---

Firma del tutor  
Dr. C. Omar Mar Cornelio

**Tutor:** Dr.C. Armando Leyva Pérez

**Título de pregrado:** Ingeniero Radiotécnico (1984); en el Instituto Técnico Militar “José Martí”

**Categoría científica:** Doctor en Ciencias Militares (2009)

**Categoría docente:** Profesor Titular (2011)

**Correo electrónico:** [leyvap@uci.cu](mailto:leyvap@uci.cu)

**Tutor:** Dr.C. Omar Mar Cornelio

**Título de pregrado:** Licenciado en pedagogía en la especialidad Computación

**Categoría científica:** Doctor en Ciencias Técnicas (2019)

**Categoría docente:** Profesor Titular (2023)

**Correo electrónico:** [omarmar@uci.cu](mailto:omarmar@uci.cu)

## Agradecimientos

*Le doy las gracias a Dios, a mi familia por ser tan especiales y siempre poder contar con ustedes en las buenas y en las malas, a mi madre por todo su apoyo y ayuda incondicional en todo momento sin importar las adversidades de la vida, por todo ese amor que solo una madre sabe dar, por ser la mejor madre del mundo. A mi padre, por ser un guerrero e inspirarme a ser cada día mejor, por ser mi guía, un gran padre, por su apoyo incondicional en todos estos años y en mi carrera, siempre pude contar con tu apoyo en todo momento, llevo presente cuando me sentía cansada que sentía que no podía más me decías que lo importante era que yo estuviera bien. A mis hermanos por su ayuda. A mi abuela por estar a mi lado en los peores momentos y en las mejores noticias y logros y siempre entenderme, eres la mejor Mima y única, al igual que mi bisabuela Enma por ver sido la mujer más fuerte y valiente que he conocido, por verme dado tu luz inspiradora y guía, y a mi abuelo Emilio por ser mi ángel guardián y luz guía por siempre, aunque me sentí muy mal ese día tan triste en que nos separamos físicamente, te prometí coger la UCI y ser la ingeniera en Ciencias Informáticas que soy hoy, un sueño cumplido, gracias al esfuerzo diario y al empeño realizado, el cual les dedico a todos ustedes, los que siempre han sido tan importantes en mi vida.*

*A Marta por ser la persona que me incitó a ser una profesional y a no darme por vencida cómo me decía yo quiero ver el título colgado en la pared, gracias por tus consejos únicos cuando me sentía triste, cansada y sin saber bien cuál era la mejor decisión que tomar, tú me guiaste y salía de tu casa sintiéndome bien y muy agradecida de por vida por tu luz protectora.*

*A mi prima Marily, que es mi madre le estoy eternamente agradecida por ser incondicional conmigo, aunque no puedas estar hoy físicamente, estas de corazón unida a mí, te digo que no tengo como pagarte lo que has hecho por mí en esta vida, por ver sido la persona que me proporcionó la laptop que tanto necesitaba y ser el medio que hoy formó parte de esta graduada, tú que nunca dudaste en ayudarme porque eres una persona especial, no tienes comparación, un gran ser humano con todos.*

*A Denis como le digo mi profesor, gracias por ser tan especial por inspirarme a ser cada día mejor profesional y por motivarme a superarme, a Pedro, a Yinet, a Claudia, a Yesenia, a Pablo, a Abdiel, a Glenda, a Dayelin, a Yosley, a Yunior, a Karel, a Daniel, a Danilo, a Dariel como le gusta que le digan mi cotutor, gracias por ser las amistades que me apoyaron, que no dudaron en ayudarme cuando más los necesité, gracias por sus consejos a Claudia, a Yesenia, a Yinet y a Glenda, que me decían tranquila que lo vas a lograr, por sus enseñanzas, a Denis que me inspiró a continuar ese día que sentí que el mundo se me venía encima y que todo estaba casi perdido, gracias por el apoyo y la ayuda, muchísimas gracias de corazón, gracias a todos, por todo lo que juntos vencimos en las adversidades de la vida estudiantil.*

*Todos mis amigos: muchas gracias por todo lo vivido juntos, los chistes compartidos cómo me dicen que mi risa es única, gracias por las emociones vividas, por su apoyo emocional y ayuda, nunca los olvidaré son mis amigos de por vida, mis hermanos.*

*A los profes Armando, Bárbara y su esposo Omar, Rodrigo, Leonardo, Bernardo, por su apoyo incondicional, les estoy agradecida de por vida, le doy gracias a Dios por permitirme conocerlos y siempre tuve su apoyo incondicional en todo momento, solo ustedes saben dar su amor, apoyo y ejemplo como profesional y ser humano, a todos los profesores que tuve en la vida estudiantil, por su ayuda y apoyo, por inspirarme a ser cada día mejor, me han ayudado a ser un mejor profesional, han hecho de mí una profesional.*

*Profes Armando y Omar: muchas gracias por todo su apoyo incondicional, por su ayuda en este proceso tan importante en mi vida la Tesis, por brindarse y aceptar ser mis tutores y al profe Rodrigo Desiderio que me atendió con mucho amor cómo una hija, cómo ya una profesional, cuando me dijo: "estoy contigo hasta el final"; a ustedes que sin conocerme me brindaron su más sincera ayuda y enseñanza, por esos momentos que compartimos juntos, por todos los consejos y lecciones dadas les estoy agradecida eternamente. Por sus enseñanzas quiero dedicarle estas frases: "Los grandes objetivos no se logran con dinero ni con talento, se logran con pasión, paciencia y perseverancia", " Si quieres llegar donde la mayoría no llega, debes hacer lo que la mayoría no hace" Profesores: muchas gracias por todas sus enseñanzas y lecciones, gracias a ustedes puedo ser la profesional que soy hoy.*

## Dedicatoria

*A mi madre Yanet, mi padre Lázaro, mi abuela Isabel, mi prima Maily, mis familiares y amigos y en especial a mi abuelo Emilio.*

## RESUMEN

Los repositorios son plataformas digitales de acceso abierto destinado a reunir, preservar y difundir la producción intelectual y académica. En tal sentido, el Repositorio Institucional Continental es un espacio digital que busca, reunir, preservar, difundir la producción científica, académica e institucional de la Corporación Continental. No existe una herramienta como un repositorio digital que recopile los contenidos educativos de la disciplina Preparación para la Defensa, por lo que se plantea como necesidad la creación de un repositorio informático que permita la gestión y visualización de los recursos educativos que posee la disciplina Enseñanza Militar. Para facilitar el trabajo y el uso de la misma se gestiona esta aplicación web. Para llevar a cabo este proyecto se utilizaron herramientas y tecnologías preestablecidas y se realizó un estudio de las características que presentan los sistemas existentes para el trabajo con aplicaciones web en el mundo y en Cuba. Siguiendo los pasos que propone la metodología AUP-UCI escenario 2 se elaboró la Documentación Técnica mediante la herramienta de modelado Visual Paradigm, así como la documentación complementaria requerida para describir el funcionamiento del sistema, quedando de este modo conformado para su acceso como repositorio web.

Palabras claves: gestión de contenidos, Preparación para la Defensa, recurso educativo, repositorio, tecnología.



## **ABSTRACT**

*Repositories are open access digital platforms intended to gather, preserve and disseminate intellectual and academic production. In this sense, the Continental Institutional Repository is a digital space that seeks to gather, preserve, and disseminate the scientific, academic and institutional production of the Continental Corporation. There is no tool such as a digital repository that compiles the educational content of the Defense Preparation discipline, which is why the creation of a computer repository that allows the management and visualization of the educational resources that the Military Education discipline has is proposed as a necessity. To facilitate the work and use of it, this web application is managed. To carry out this project, pre-established tools and technologies were used and a study was carried out of the characteristics of existing systems for working with web applications in the world and in Cuba. Following the steps proposed by the AUP-UCI scenario 2 methodology, the Technical Documentation was prepared using the Visual Paradigm modeling tool, as well as the complementary documentation required to describe the operation of the system, thus being formed for access as a web repository.*

*Keywords: content management, Defense Preparation, educational resource, repository, technology.*

**Tabla de contenidos**

**INTRODUCCIÓN** ..... 13

**CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN** ..... 18

**1.1. Introducción del capítulo**..... 18

**1.2. Recursos educativos digitales**..... 18

**1.2.1. Recursos educativos de preparación para la defensa** ..... 19

**1.3. Repositorios de recursos educativos digitales**..... 20

**1.4. Soluciones relacionadas con el objeto de estudio**..... 20

**1.4.1. Soluciones relacionadas con el objeto de estudio en Cuba**..... 20

**Repositorio de recursos educativos de la Universidad Virtual de Salud en Cuba** ..... 21

**Repositorio digital REPXOS.** ..... 21

**1.4.2. Soluciones relacionadas con el objeto de estudio en el mundo** ..... 22

**Agrega** ..... 22

**Sherpa Romeo**..... 23

**1.5. Necesidad de ejecutar la investigación** ..... 24

**1.6.1. Metodología de desarrollo de software** ..... 25

**1.6.2. Lenguaje de modelado** ..... 26

**1.6.3. Lenguajes de programación**..... 27

**1.6.4. Marcos de trabajo para el desarrollo**..... 29

**1.6.5. Entorno de Desarrollo Integrado** ..... 31

**1.6.6. Servidor Web** ..... 32

**1.6.7. Sistema Gestor de Bases de Datos** ..... 33

**1.7. Conclusiones del capítulo**..... 33

**CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA** ..... 34

**2.1. Introducción del capítulo**..... 34

**2.2. Modelado del contexto: modelo conceptual** ..... 34

**2.3. Requisitos de software** ..... 34

**2.3.1. Requisitos de software funcionales (RF)** ..... 35

**2.3.2. Requisitos de software no Funcionales (RNF)** ..... 37

**2.4. Modelo de casos de uso del sistema** ..... 39

**2.4.1. Actores del sistema** ..... 39

**2.4.2. Diagrama de casos de uso del sistema**..... 39

**2.4.3. Especificación de casos de uso del sistema** ..... 40

<b>2.5. Modelo del diseño</b> .....	48
<b>2.5.1. Diagrama de clases del diseño</b> .....	49
<b>2.5.2. Descripción de las clases del diseño</b> .....	49
<b>2.5.3. Patrón Arquitectónico Modelo - Plantilla – Vista</b> .....	50
<b>2.5.4. Patrones de diseño arquitectónico</b> .....	52
<b>Patrones para la asignación de responsabilidades (GRASP)</b> .....	52
<b>Patrones GOF</b> .....	53
Modelo de datos .....	53
<b>2.6. Conclusiones del capítulo</b> .....	54
<b>CAPÍTULO III: Implementación y Pruebas</b> .....	55
<b>3.1. Introducción del capítulo</b> .....	55
<b>3.2. Implementación del sistema</b> .....	55
<b>3.2.1. Diagrama de componentes</b> .....	55
<b>3.2.2. Código fuente</b> .....	56
<b>3.2.3. Estándares de codificación</b> .....	57
Nombres de clases y métodos.....	57
Estructura .....	57
<b>3.3. Diagrama de despliegue</b> .....	57
<b>3.4. Pruebas de software</b> .....	59
<b>3.4.1. Estrategia de pruebas</b> .....	59
<b>3.4.2. Nivel de prueba de sistema</b> .....	61
<b>3.4.3. Pruebas funcionales</b> .....	62
<b>3.4.3.1 Pruebas de Caja Negra</b> .....	63
<b>3.3.4 Pruebas de Seguridad</b> .....	66
<b>3.4.5 Nivel de prueba aceptación</b> .....	67
<b>3.4.5.1 Validación de las variables</b> .....	68
<b>3.5. Conclusiones del capítulo</b> .....	69
<b>CONCLUSIONES</b> .....	70
<b>RECOMENDACIONES</b> .....	71
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	72
<b>ANEXOS</b> .....	75
<b>Anexo 1. Requisitos funcionales agrupados en Casos de uso del sistema.</b> .....	75
<b>Anexo 2. Captura de Pantalla de SSRN</b> .....	76
<b>Anexo 3. Prototipos de interfaz de usuario</b> .....	77

<b>Anexo 4. Captura de pantalla de Agrega.....</b>	<b>82</b>
<b>Anexo 5. Aval del Cliente.....</b>	<b>83</b>
<b>Anexo 6. Prototipo de interfaz de portada.....</b>	<b>83</b>
<b>Anexo 7. Prototipo de interfaz de portada.....</b>	<b>85</b>
<b>Anexo 8: Captura de Pantalla de Sherpa Romeo.....</b>	<b>86</b>
<b>Anexo 9: Sección de la pantalla de inicio del Repositorio de Recursos Educativos de la Universidad Virtual de la Salud .....</b>	<b>87</b>
<b>Anexo 10: Página de inicio de CubaEduca.....</b>	<b>88</b>
<b>Anexo 11: Sección superior de la página de inicio del Repositorio Digital.....</b>	<b>88</b>

### Listado de tablas del documento

<b>Tabla 1.</b> Tipos de Recursos Educativos Digitales. <b>Fuente:</b> Elaboración propia.....	<b>19</b>
<b>Tabla 8.</b> Requisitos funcionales agrupados en Casos de uso del sistema.....	<b>75</b>

### Listado de figuras del documento

<b>Figura 1.</b> Modelo conceptual. ....	<b>34</b>
<b>Figura 2.</b> Diagrama de casos de uso del sistema. ....	<b>40</b>
<b>Figura 3.</b> PIU_ Interfaz Principal Profesor.....	<b>43</b>
<b>Figura 4.</b> PIU_ Realizar envío_ 1: Inicio.....	<b>44</b>
<b>Figura 5.</b> PIU_ Realizar envío_ 2: Cargar el envío.....	<b>44</b>
<b>Figura 6.</b> PIU_ Realizar envío_ 3: Introducir metadatos.....	<b>45</b>
<b>Figura 7.</b> PIU_ Realizar envío_ 4: Confirmación.....	<b>45</b>
<b>Figura 8.</b> PIU_ Gestionar Rol. ....	<b>48</b>
<b>Figura 9.</b> PIU_ Gestionar Rol.....	<b>48</b>
<b>Figura 10.</b> Diagrama de clases del diseño. ....	<b>49</b>
<b>Figura 11.</b> Flujo del patrón arquitectónico MTV de Django.....	<b>51</b>
<b>Figura 12.</b> Modelo de datos. ....	<b>54</b>
<b>Figura 13.</b> Diagrama de componentes.....	<b>¡Error! Marcador no definido.</b>
<b>Figura 14.</b> Diagrama de despliegue.....	<b>58</b>

## INTRODUCCIÓN

Las universidades cubanas han sido baluartes de la defensa de la Revolución, y los estudiantes sus protagonistas fundamentales. En 1975 se inició, como parte de la formación de los profesionales, la preparación militar. Con la creación del Ministerio de Educación Superior (MES) se extendió a todas las universidades y se impartió en concentrados militares, donde se formaron como oficiales decenas de miles de graduados hasta 1995 (Leyva et al., 2020).

Las condiciones económicas, la tensión a que fue sometido el país, los mandos militares y las propias universidades; las demandas y exigencias de formación del profesional, acorde a los nuevos escenarios, determinó que se emitiera la Directiva 29 del año 1995 del Ministro de las Fuerzas Armadas Revolucionarias, la cual estableció que la preparación para la defensa se impartiría por los propios profesores universitarios, mediante la disciplina Preparación para la Defensa (PPD) con un contenido teórico y práctico, vinculado con el perfil del profesional (Leyva et al., 2020).

Lo anterior ha conllevado a una permanente preparación integral del pueblo para contrarrestar las diferentes formas de agresión empleadas por el enemigo, lo cual se ha resumido en la Doctrina Militar de la Guerra de Todo el Pueblo, y en la instrumentación de las medidas de la Defensa Civil, donde el pueblo y sus dirigentes han jugado y juegan un papel protagónico. Los contenidos que se imparten a los estudiantes universitarios intentan un acercamiento científico, al sistema de conocimientos y habilidades que deben alcanzar los estudiantes, con el objetivo de garantizar la preparación básica en seguridad y defensa nacional, para que puedan cumplir con sus deberes, obligaciones y ejecutar las acciones dentro del sistema defensivo territorial a su nivel, tales como: la preparación de la economía para la defensa, los elementos de la lucha armada y la defensa civil, en tiempo de paz o en situaciones excepcionales (Quesada et al., 2014).

En la preparación del profesional cubano, la disciplina Preparación para la Defensa es una exigencia imprescindible para su futuro desempeño en las diferentes áreas del desarrollo económico, político y social, lo que se corresponde con la política económica y social del Partido y la Revolución, por su contribución en su formación revolucionaria, patriótica y el desarrollo de los valores que debe caracterizar a un graduado universitario.

Los especialistas en el área de Preparación para la Defensa, tienen una amplia producción científica sobre los principales sustentos de la preparación en Cuba. Asimismo, se ha constatado un gran número de Recursos Educativos Digitales (RED) en distintos formatos para ser empleados en la impartición de la asignatura (Beltrán & Urgellés, 2012). Sin embargo, estos RED muchas veces no son conocidos por todos los docentes de la disciplina, ni son accedidos por los estudiantes, puesto

que se almacenan en computadoras personales, limitando la accesibilidad y la publicación de los mismos.

A partir de entrevistas con profesores y especialistas de la disciplina Preparación para la Defensa, se pudo constatar la necesidad de contar con un repositorio para la gestión de contenidos digitales de esta disciplina. Los repositorios institucionales son una de las formas de publicación del Acceso Abierto, que surgen, desde la aparición de la web, con el objetivo de divulgar los resultados de las investigaciones científicas dentro de la comunidad de investigadores, así como facilitar el acceso a los RED creados por los docentes. El objetivo principal de esta herramienta es conservar, difundir y hacer accesibles a todos los usuarios los resultados científicos de las investigaciones realizadas en su institución y de los materiales de aprendizaje (Álvarez & López, 2008).

Cuba ha apostado por la soberanía tecnológica en este ámbito y ha desarrollado sus propios repositorios de gestión de recursos educativos como CubaEduca, que posee un amplio espacio de contenidos interactivos en formato digital, para educación desde preescolar hasta el duodécimo grado (Salabarría Cárdenas, 2022). En la Universidad de las Ciencias Informáticas (UCI) existen también repositorios que contienen los recursos educativos de la comunidad universitaria, entre estos el Repositorio Digital que brinda acceso a tesis y libros relacionados con las materias de la carrera Ingeniería en Ciencias Informáticas.

Un repositorio digital permite la gestión de recursos educativos en todos los formatos predefinidos, sean imágenes, videos, multimedia y el procesamiento de recursos en los formatos actuales (Delís et al., 2018). Además, no existe en la universidad un repositorio centralizado que contenga acceso a recursos digitales de contenido de preparación para la defensa que permitan la formación de los estudiantes de la Universidad de las Ciencias Informáticas. Las actividades referentes a las asignaturas de la disciplina Preparación para la Defensa contribuyen a formar un profesional con una visión integral, creativo; gestor de su propio aprendizaje, con el valor agregado de defender el proyecto socialista desde su profesión.

Existe una plétora de información sobre los RED, sin embargo, la práctica ha demostrado que no es sencillo acceder a ellos fácilmente, más allá de los sistemas de tecnologías institucionales y, en consecuencia, conducir a un uso y una aceptación potencialmente mayores entre los estudiantes y el personal, sigue siendo una idea, más que una práctica establecida. A pesar de un número cada vez mayor de iniciativas para establecer repositorios de recursos educativos digitales abiertos (Aparicio et al., 2019; Hew & Cheung, 2014; Margaryan et al., 2015), aún no es suficiente el acceso a estos, además de las complicaciones de compatibilidad, portabilidad y accesibilidad (Amén Mora et al., 2023).

El departamento de Enseñanza Militar no cuenta con un repositorio donde se publique documentos e información de las asignaturas vinculadas a la Preparación para la Defensa, lo cual ha debilitado la visibilidad de los contenidos para sus estudiantes en la Universidad y de personas jurídicas. Además, no cuenta con ninguna vía por la cual los profesores y estudiantes puedan intercambiar información ni consultar la misma de forma rápida y segura.

A partir de la problemática descrita anteriormente se obtiene el siguiente **problema a resolver**: ¿cómo contribuir a la gestión de recursos educativos digitales de la disciplina Preparación para la Defensa?

El **objeto de estudio** es: la gestión de contenidos educativos digitales. El **campo de acción**: repositorio informático para la gestión de recursos educativos digitales de la disciplina Preparación para la Defensa.

Para dar solución al problema científico declarado, esta investigación tiene como **objetivo general**: Desarrollar un repositorio para la gestión de recursos educativos digitales de la disciplina Preparación para la Defensa. A partir de este objetivo general se trazaron los siguientes **objetivos específicos**:

- Elaborar el marco teórico-conceptual relacionado a la gestión la gestión de recursos educativos digitales en los repositorios de contenidos digitales.
- Diseñar las funcionalidades del repositorio para la gestión de recursos educativos digitales de la disciplina Preparación para la Defensa.
- Implementar el repositorio de repositorio para la gestión de recursos educativos digitales de la disciplina Preparación para la Defensa.
- Realizar pruebas al repositorio desarrollado.

**Idea a defender**: con el desarrollo de un repositorio de contenidos se contribuirá a la gestión de los recursos educativos digitales de la disciplina Preparación para la Defensa.

Para llevar a cabo los objetivos propuestos se utilizaron los siguientes **métodos científicos**:

**Métodos teóricos.**

1. Analítico-sintético: el análisis de diversas metodologías y documentos relacionados con el objeto de estudio de los cuales se extraen los rasgos distintivos más importantes relacionados con la gestión de recursos educativos.
2. Modelación: este método posibilita modelar el funcionamiento de una aplicación para la gestión de recursos educativos, como parte de una propuesta para dar respuesta a las necesidades del proyecto de Informatización.

3. Histórico-lógico: se utiliza con el fin de estudiar los antecedentes y tendencias de los repositorios de recursos educativos existentes a nivel nacional e internacional.

#### **Métodos empíricos.**

1. Observación: debido a que se constataron los problemas que existían en el departamento de Enseñanza Militar con respecto al proceso de gestión de recursos educativos, motivándonos a realizar la investigación.

#### **Aportes prácticos esperados de la investigación:**

Se espera obtener un repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa que permita el acceso de cada usuario a la información necesaria para resolver problemas relacionados con la enseñanza y el aprendizaje de manera organizada. Con la realización de la investigación se espera contribuir a:

- La creación de entornos propicios para el uso de las TIC.
- Apoyo del aumento de capacidades para el desarrollo sostenible de materiales de aprendizaje de calidad en el Departamento de Enseñanza Militar.
- Promoción de la investigación sobre recursos educativos abiertos.

El presente documento se encuentra estructurado en 3 capítulos, los cuales resumen la siguiente información:

**Capítulo I: Fundamentación teórica de la investigación.** En este capítulo se presentan los elementos teóricos que sustentan al proceso de gestión de contenidos digitales y los objetivos del trabajo; así como un análisis de las soluciones existentes afines al objeto de estudio. Se demuestra la necesidad de implementar un repositorio para la gestión de contenidos digitales de la Preparación para la Defensa en la Universidad de las Ciencias Informáticas. Se selecciona la metodología, herramientas y tecnologías a utilizar en el desarrollo del sistema, justificando su selección.

**Capítulo II: Análisis y diseño de un repositorio para la gestión de contenidos de la disciplina Preparación para la Defensa en la UCI.** En este capítulo se efectúa el estudio detallado del problema, se describen los requisitos (funcionales y no funcionales) ingenieriles. Se define la arquitectura de implementación a utilizar para el desarrollo del repositorio de contenidos. Se realizan los diagramas de clases del diseño con la finalidad de describir gráficamente los datos asociados con el área de la aplicación y se aplican los patrones arquitectónicos y de diseño con la finalidad de cumplir el objetivo general.

**Capítulo III. Evaluación de la solución propuesta.** Se conforma el modelo de implementación del sistema como resultado del análisis y diseño. Se confeccionan los diagramas de componentes y de



despliegue. Es creada y aplicada la estrategia de pruebas del software, que consta de pruebas de unidad, de sistema y de aceptación; con el objetivo de verificar que el repositorio de contenidos cumple con los requisitos especificados por los docentes de la disciplina Preparación para la Defensa.

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

### 1.1. Introducción del capítulo

El aumento de los volúmenes de información en el auge de la nueva era digital y el surgimiento de nueva bibliografía actualizada sobre cada materia ocasiona que los espacios físicos constituidos por bibliotecas y almacenes de libros se vuelvan insuficientes ante la gran cantidad de títulos anuales relacionados con cada asignatura en las diversas enseñanzas (Hennessy et al., 2022). Es por eso que surge la necesidad de crear soluciones informáticas que gestionen esta ingente cantidad de recursos educativos con los que pueden contar en la actualidad los centros educacionales (Coban et al., 2022).

Estos recursos suelen estar constituidos por cursos, libros de texto, videos, exámenes, además pueden ser utilizados, compartidos, combinados, modificados y adaptados según las necesidades educativas. Debido a la diversidad de los contenidos que se crean, resulta imprescindible que los mismos cumplan con una estructura o formato común, de manera tal que puedan ser manejados por las diferentes herramientas de aprendizaje. Un ejemplo de herramienta lo constituyen los repositorios en particular, aplicaciones informáticas dedicadas al almacenamiento y manejo de recursos digitales, garantizando la accesibilidad, localización y reutilización de los mismos. Se han definido a estos como sistemas especializados en el almacenamiento de recursos digitales, accesibles y operables por usuarios a través de Internet/Intranet, con funciones encaminadas a la clasificación, localización, reutilización y mantenimiento de los mismos.

A lo largo de los años han surgido varias tecnologías para el desarrollo de productos *web*, destacando elementos como la productividad y velocidad en la construcción de productos, la seguridad, el rendimiento del producto y, obviamente, la reutilización de componentes en diferentes proyectos. Los productos *web* se dividen comúnmente en dos subproductos tecnológicos: *front-end* y *back-end*. Las tecnologías *back-end* están vinculadas a los lenguajes de programación; sin embargo, se ocupan de la programación lógica de los sistemas *web* e influyen en la forma en que se almacenan, acceden y sirven los datos desde los servidores. Por otro lado, las tecnologías *front-end* se centran en la estética y la visualización de contenidos, también conocido como desarrollo del lado del cliente. En este capítulo se describe la selección del lenguaje *Python* soportado por el marco de trabajo *Django* como la principal tecnología *back-end*; y el marco de trabajo *Bootstrap* como la principal tecnología *front-end*.

### 1.2. Recursos educativos digitales

Los Recursos Educativos Digitales (RED) son materiales (textos, imágenes, videos, libros de texto, módulos de cursos, pruebas, tareas, etc.) en formato digital que se utilizan en el contexto de la

enseñanza - aprendizaje. Al crear un RED, se tiene en cuenta la inclusión de una variedad de alumnos. Un RED permite satisfacer las necesidades de los estudiantes, relacionadas con sus limitaciones o diferentes estilos de aprendizaje, al tiempo que fomenta el aprendizaje.

Los RED constituyen un conjunto de materiales pedagógicos, de aprendizaje o de investigación que son de dominio público o pueden utilizarse bajo una licencia de propiedad intelectual que permite su reutilización o modificación (como una licencia *Creative Commons*). Específicamente, los Recursos Educativos Abiertos (REA) son materiales de enseñanza, aprendizaje e investigación en cualquier medio, digital o de otro tipo, que residen en el dominio público o se han publicado bajo una licencia abierta que permite el acceso, el uso, la adaptación y la redistribución sin costo por otros con sin restricciones limitadas.

Los RED también pueden ser (Nilashi et al., 2022):

- Contenidos formativos: cursos completos, software educativo, módulos de contenido, recopilaciones y publicaciones.
- Herramientas: software para poder desarrollar, utilizar, reutilizar y entregar el contenido formativo, incluidas la búsqueda y organización del contenido, los sistemas de gestión de contenido y formación, las herramientas de desarrollo de contenidos y las comunidades educativas en línea
- Recursos de implementación: licencias de propiedad intelectual para promover la publicación abierta de materiales, diseño de principios de buenas prácticas y de traducción de contenidos.

En esta clasificación se aprecia cómo una amplia variedad de objetos y recursos en línea se clasifican como REA. Se pretende que los futuros docentes, pedagogos y educadores conozcan la existencia de la localización de estos recursos en numerosos sitios webs y en múltiples formatos. De acuerdo con (Amén Mora et al., 2023), la literatura ha descrito distintos tipos de RED, a saber:

**Tabla 1.** Tipos de Recursos Educativos Digitales. **Fuente:** Elaboración propia.

• Fotografías	• Infografías	• Juegos de rol	• Modelos estáticos y dinámicos
• Gráficos	• Mapas	• Contenido interactivo	• Objetos de realidad virtual
• Video	• Objetos de aprendizaje	• Grabaciones sonoras	• Aplicaciones móviles (App)
• Software	• Redes sociales	• Documentos de texto	• Juegos interactivos
• Flash	• Herramientas autor	• Cursos	• Complementos Chrome
• Láminas	• Presentación	• Web	• WebQuest

**Fuente:** Tomado de (Amén Mora et al., 2023)

### 1.2.1. Recursos educativos de preparación para la defensa

Dentro de las competencias de los profesores de la disciplina Preparación para la Defensa se encuentra la creación de atractivos RED que resuman los aspectos básicos de las asignaturas

Seguridad Nacional y Defensa Nacional. El objetivo de estos recursos es proporcionarles a los estudiantes, la posibilidad de contar con un material ameno, asequible y resumido de los contenidos de las asignaturas Seguridad Nacional y Defensa Nacional de Cuba; así como, los fundamentos de la preparación del país para la defensa, lo que amplía el material bibliográfico para la universalización de estos contenidos en la Educación Superior Cubana.

El recurso debe:

- Ser creado y utilizado con fines pedagógicos.
- Proporcionar una referencia clara para todos los trabajos o extractos de trabajos.

### **1.3. Repositorios de recursos educativos digitales**

Los repositorios de recursos educativos digitales son bases de datos digitales que albergan contenido de aprendizaje, aplicaciones y herramientas tales como textos, documentos, videos, grabaciones de audio, aplicaciones multimedia y herramientas de redes sociales (Fernández et al., 2022). Por lo tanto, estos repositorios tienen como objetivo recopilar RED y sus metadatos para facilitar su búsqueda y hacerlos visibles, pero también presentan otros desafíos, por ejemplo, la falta de uso de metadatos educativos estandarizados (Aleixandre-Benavent et al., 2021).

A nivel internacional existe una variedad de repositorios para fines educacionales. Entre los más populares en la actualidad se encuentran Agrega, Sherpa Romeo y SSRN. Todos estos repositorios cumplen con los estándares de calidad actuales y poseen información verificada y actualizada. Los objetos de aprendizaje alojados en los servidores de estos repositorios cumplen con el modelo SCORM lo que garantiza su fiabilidad. Los servidores de estos repositorios se encuentran alojados en regiones localizadas en países de América del Norte o la Unión Europea.

### **1.4. Soluciones relacionadas con el objeto de estudio**

#### **1.4.1. Soluciones relacionadas con el objeto de estudio en Cuba**

En el contexto universitario cubano se han creado otras redes, como: la Red Universitaria de Datos (RedUniv), Red Nacional de Enfermería Informática (REDENFI), Cubarte, Cubaeduca, Ecured, Telemática de Salud, Cubafarma, entre otras.

RedUniv, es la segunda red en importancia en Educación Superior de Cuba. Es una red Nacional de Datos que incluye y enlaza a todas las redes de datos de todos los centros universitarios adscritos al Ministerio de Educación Superior (MES). Algunas de las principales iniciativas que facilitan la adopción y uso de RED en el sistema MES son:

- Biblioteca RedUniv en acceso abierto, que incluye un repositorio de tesis doctorales y presentaciones de ponencias en congresos internacionales en formato electrónico. Sin embargo, no siempre se cumple con los indicadores de calidad establecidos para su divulgación, publicación y uso.
- Las universidades cubanas del MES disponen de catálogos automatizados, no obstante, aún se trabaja en la creación de un catálogo que permita el acceso abierto a todos los repositorios disponibles en la Red Nacional Universitaria con decenas de miles de contenidos digitales.

En las Instituciones de Educación Superior de Cuba de manera general se evidencian iniciativas que pueden contribuir a la adopción y uso de recursos educativos digitales: se proyectan entornos propicios para el uso de estos materiales; se incrementan las capacidades para el desarrollo sostenible de recursos de aprendizaje de calidad y creación, uso y reutilización de recursos educativos digitales; se desarrollan y utilizan plataformas académicas interactivas para la gestión de contenidos de aprendizaje, que de forma paulatina se introducen en las universidades cubanas.

### **Repositorio de recursos educativos de la Universidad Virtual de Salud en Cuba**

El repositorio de recursos educativos de la Universidad Virtual de Salud (UVS) en Cuba, es un proyecto desarrollado por la red telemática Infomed. Este sistema tiene como objetivo garantizar la accesibilidad y la reutilización de recursos educativos producidos en el sistema nacional de salud, mediante el depósito, descripción y distribución de los mismos. Esta característica permite clasificarlo como repositorio temático. Esencialmente, garantiza los servicios básicos de búsquedas simples y avanzadas, además de poder filtrar las mismas por los metadatos de los contenidos tales como: autor, título, tipo de recurso y año de creación. Igualmente, permite descargar e imprimir los recursos, así como recomendar los mismos a otros usuarios. Los usuarios una vez autenticados en el repositorio califican y comentan los recursos.

Se constata mediante la observación que es el repositorio más simple de todos los investigados, no posee integración con redes sociales ni implementa el comercio electrónico. En el Anexo 9 se aprecia una sección de su página de inicio.

### **Repositorio digital REPXOS.**

Es un repositorio de la Universidad de Ciencias Informáticas que brinda acceso a tesis, libros y materiales de estudio relacionados con las Ciencias Informáticas, permite la adaptación a múltiples usos para la gestión y recuperación de documentos digitales en cualquier formato. Permite incorporar documentos digitales, buscar y navegar por documentos, administrar el sistema y realizar suscripciones. El repositorio no ha implementado la subida de recursos en

formato multimedia (imágenes, videos, audiolibros) ni una sección destinada al comercio electrónico. En el Anexo 11 se visualiza la sección superior de la página de inicio del repositorio digital.

#### **1.4.2. Soluciones relacionadas con el objeto de estudio en el mundo**

En el mundo, principalmente en Estados Unidos, América Latina y Europa, existen diferentes sistemas o herramientas que permiten dar solución a diversos problemas relacionados con la distribución de contenido.

Alrededor del mundo, las instituciones educacionales de diversos países incorporan repositorios como herramientas de apoyo al proceso de enseñanza y aprendizaje, incrementando la interacción de los estudiantes con los medios electrónicos y las nuevas tecnologías. Numerosos son los repositorios reconocidos a nivel internacional por la calidad de los servicios que brindan y los recursos que almacenan.

#### **Agrega**

Es un proyecto desarrollado entre el Ministerio de Educación, Cultura y Deporte, y las Comunidades Autónomas en España. El objetivo de Agrega es facilitar a la comunidad educativa una herramienta útil que integre las TIC en el aula y fuera de ella, permitiendo además acceder a cualquier contenido, desde cualquier sitio y sea cual sea el criterio de búsqueda empleado. La plataforma es capaz de gestionar todo tipo de objetos utilizables en entornos web, de diferentes formatos digitales, ya sean objetos simples (texto, imágenes, archivos de audio, videos y otros) o cualquier combinación de los anteriores.

Agrega utiliza la especificación LOM y el estándar SCORM para la catalogación y el empaquetamiento de los contenidos respectivamente. Los contenidos a su vez, están curricularmente organizados de forma que puedan ser utilizados en la enseñanza no universitaria. Como parte de las funciones que brinda el repositorio, el usuario puede realizar valoraciones y comentarios, pre visualizar el contenido de cualquier recurso almacenado sin tener que estar autenticado en el sistema. Los recursos pueden ser descargados y exportados a diferentes formatos, además pueden ser enviados por correo electrónico a otros usuarios y compartidos en redes sociales. Algunos recursos educativos contenidos en la plataforma Agrega no están disponibles para su descarga para Cuba. Se puede apreciar una captura de pantalla de este repositorio en el Anexo 4.

## Sherpa Romeo

Repositorio digital de recursos educativos con sede en Reino Unido. Es la plataforma por excelencia de la Unión Europea para la subida de tesis e investigaciones, contando con la colaboración de científicos que se encargan del proceso de revisión de los documentos. Cuenta con la posibilidad de subir el documento en múltiples formatos de texto y exportarlo al estándar .pdf. La página de entrada está disponible desde Cuba, pero mayoría de las bibliotecas de datos que contiene son inaccesibles y requieren de la utilización de software VPN. Sherpa Romeo permite la integración y autenticación de los usuarios mediante redes sociales. Se puede observar una captura de pantalla de este repositorio en el Anexo 8.

Tabla 2. Tabla de estudio de los sistemas homólogos. Fuente: Elaboración propia

<b>Reposit orios</b>	<b>Ubicación</b>	<b>Acceso desde Cuba</b>	<b>Tipo</b>	<b>Contiene Recursos para Educació n Superior</b>	<b>Tendencia s Actuales. (1, 2, 3)</b>
Agrega	Europa	Parcial	Centralizad o	Si	1, 3
SSRN	Estados Unidos	No permitido (Requiere VPN)	Centralizad o	Si	1, 2, 3
Sherpa Romeo	Europa	Parcial	Centralizad o	Si	1, 2, 3
Reposito rio UVS	Cuba	Si	Temático	Si	Ninguna
CubaEdu ca	Cuba	Si	Centralizad o	No	1, 2, 3
Repxos	Cuba	Si	Instituciona l	Si	Ninguna

Alrededor del mundo, las instituciones educacionales de diversos países incorporan repositorios como herramientas de apoyo al proceso de enseñanza y aprendizaje, incrementando la interacción de los estudiantes con los medios electrónicos y las nuevas tecnologías. Numerosos

son los repositorios reconocidos a nivel internacional por la calidad de los servicios que brindan y los recursos que almacenan.

### **1.5. Necesidad de ejecutar la investigación**

A pesar de que existe un gran número de investigaciones centradas en los méritos y desafíos pedagógicos de los RED, el aspecto técnico de su distribución y almacenamiento aún no se ha analizado a fondo, y mucho menos el establecimiento de una estrategia estandarizada para el desarrollo y alojamiento de repositorios de recursos educativos digitales.

De acuerdo con la investigación realizada, las entrevistas formuladas, y la práctica personal de la autora, se ha demostrado que no es sencillo acceder a los repositorios externos fácilmente, más allá de los sistemas de tecnologías institucionales y, en consecuencia, conducir a un uso y una aceptación potencialmente mayores entre los estudiantes y los profesores, aún no ha sido posible. A pesar de un número cada vez mayor de iniciativas para establecer repositorios de recursos educativos digitales (Aparicio et al., 2019; Hew & Cheung, 2014; Margaryan et al., 2015), aún no es suficiente el acceso a estos, además de las complicaciones de compatibilidad, portabilidad y accesibilidad.

Basado en esta problemática, la presente investigación tiene como objetivo desarrollar un repositorio para la gestión de contenido de la Preparación para la Defensa en la Universidad de las Ciencias Informáticas que permita la publicación, evaluación, alojamiento y acceso de los RED creados para la disciplina Preparación para la defensa. Como aspiración fundamental se espera la creación de un repositorio institucional de RED, basado en los principios de acceso abierto, igualdad, respeto a la diversidad, y el diseño universal. Este repositorio será diseñado con los metadatos estandarizados internacionalmente, de manera que pueda ser integrado a otras iniciativas, y no quede como un esfuerzo aislado e individual.

Cabe destacar que el sistema cuenta con los requisitos comparados anteriormente, donde se muestra que permite la adaptación a múltiples usos para la gestión y recuperación de documentos digitales en cualquier formato, permite incorporar documentos digitales, buscar y navegar por documentos. El repositorio ha implementado la subida de recursos en formato multimedia (imágenes, videos, audiolibros), se evidencia la importancia de la creación de este repositorio institucional de RED que le presta vital importancia a la disciplina Preparación para la Defensa.

### **1.6. Tecnologías, herramientas y metodologías a considerar**

Las tecnologías, herramientas y metodologías escogidas para el desarrollo del sistema se exponen en el siguiente subtema, para lo cual se analizaron las distintas alternativas y las necesidades del



proyecto. El principal elemento que se tuvo en consideración fue que las herramientas debían ser compatibles con el sistema que ya fue implementado, multiplataformas, sin restricciones de uso.

### 1.6.1. Metodología de desarrollo de software

Para lograr el desarrollo de sistemas informáticos es necesario el empleo de metodologías de desarrollo de *software*, que proponen una guía disciplinada que cumplir en la creación de la aplicación. El principal objetivo de estas es mejorar la calidad del *software* y mitigar los problemas que pueden acontecer en su proceso de creación.

Una metodología es un conjunto integrado de técnicas y métodos que permiten abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de *software* detallado y completo; un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito (Pantaleo & Rinaudo, 2015). El proceso de desarrollo llevaba asociado un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos. Este esquema tradicional para abordar el desarrollo de *software* ha demostrado ser efectivo y necesario en proyectos (Raya, 2014).

Las metodologías de desarrollo de *software* pueden ser: tradicionales o ágiles. Las metodologías tradicionales son orientadas por planeación. Estas inician el desarrollo de un proyecto con un riguroso proceso de elicitación de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario (Cadavid et al., 2013). Entre las metodologías tradicionales se encuentra *Rational Unified Proccess* (RUP, en sus siglas en inglés) (Akhunzada et al., 2015; Hui et al., 2015). Por otra parte las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto (Cadavid et al., 2013). Entre las metodologías ágiles se encuentran: *Agile Unified Process* (AUP, en sus siglas en inglés) (Jianwei & Xunan, 2010), Scrum (Koç & Aydos, 2017; Srivastava et al., 2017) y *Open Unified Process* (OpenUP) (Ardi & Shahmehri, 2008; Borg et al., 2007; Farinango et al., 2015). Para el desarrollo de la aplicación informática se utilizará una metodología de desarrollo de *software* ágil, ya que se ajusta al tipo de solución que se persigue.

En la presente investigación se decidió emplear la metodología AUP en su versión UCI (AUP-UCI). Esta es una adaptación de la metodología AUP propuesta para la actividad productiva de la UCI. Es seleccionada dicha metodología porque el equipo de desarrollo se encuentra familiarizado con ella, además de que propone un aumento de la calidad del producto, como resultado de sus buenas prácticas informáticas.

AUP-UCI es una metodología que encuentra un equilibrio entre las metodologías ágiles y las tradicionales. Se adapta, en gran medida, tanto al tipo de proyecto a desarrollar como a las condiciones para la elaboración de este trabajo. Tiene como ventaja que prioriza la gestión de riesgos lo que permite la detección y mitigación temprana de los mismos, se adecua rápidamente a los cambios en el entorno del sistema y proporciona un alto nivel de independencia a la hora de seleccionar las herramientas para el desarrollo del sistema (Duarte, 2016).

Esta metodología posee tres fases: Inicio, Ejecución y Cierre. Consta de siete disciplinas: Modelado de negocio; Requisitos; Análisis y diseño; Implementación; Pruebas internas; Pruebas de liberación; y Pruebas de aceptación. Entre sus roles se encuentran: arquitecto de información, desarrollador, administrador de la configuración, cliente/proveedor de requisitos, administrador de calidad, probador, arquitecto de *software* y administrador de BD.

AUP en su variante UCI propone 4 escenarios para modelar el sistema en los proyectos, a continuación, se exponen:

- **Escenario No. 1:** modelar el sistema con Caso de Uso (CUS) cuando el negocio se modele con Caso de Uso del Negocio (CUN) y Modelo Conceptual (MC).
- **Escenario No. 2:** modelar el sistema con CUS cuando el negocio se modele con MC.
- **Escenario No. 3:** modelar el sistema con Descripción de Requisitos por Proceso (DRP) cuando el negocio se modele con Descripción de Proceso de Negocio (DPN) y con MC.
- **Escenario No. 4:** modelar el sistema con Historias de Usuario (HU) cuando no se realice modelado de negocio.

Se seleccionó el Escenario No. 2 que requiere el modelado del negocio a través del Modelo Conceptual, y el modelado del sistema se realiza a partir de la especificación de casos de uso del sistema. Se considera, que la especificación de los Casos de Usos del Sistema, son una potente herramienta para la posterior implementación del sistema informático.

### 1.6.2. Lenguaje de modelado

Con la finalidad de analizar y modelar la solución será empleado el Lenguaje Unificado de Modelado v2.0 (UML, en sus siglas en inglés) (Larman, 2003). Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de *software* previo al proceso intensivo de escribir código. Dentro de las ventajas de la utilización de UML se tiene que permite visualizar, especificar, construir y documentar el sistema. Es necesario el empleo de UML v2.0 porque proporciona un lenguaje para expresar requisitos y pruebas, describe el comportamiento del

sistema tal y cómo es percibido por los usuarios y comprende los componentes y archivos que se utilizan para acoplar y hacer disponible el sistema físico.

Es especialmente adecuado cuando se trata la construcción de sistemas de software desde la perspectiva de la orientación a objetos. La especificación, visualización, construcción y documentación de cualquier sistema requiere que pueda ser estudiado desde diferentes puntos de vista, ya que un usuario final necesita una visión diferente del sistema de la que necesita un analista o un programador. UML incorpora toda una serie de diagramas y notaciones gráficas y textuales, destinadas a mostrar el sistema desde las diferentes perspectivas, que pueden utilizarse en las diferentes fases del ciclo de desarrollo del software.

### 1.6.3. Lenguajes de programación

**Lenguajes de programación para el *front-end*:** En el *front-end*, el desarrollador transforma la idea de diseño de la aplicación en realidad, centrándose en la maquetación, diseño e interactividad de la aplicación Web. Para ello se utilizan lenguajes como HTML (lenguaje de marcado), CSS (lenguaje de estilo) y *JavaScript* (lenguaje de scripting/ programación). A continuación, se muestra una breve descripción de cada uno de estos lenguajes que serán empleados para la implementación de la propuesta:

#### **Lenguaje de Marcado de Hipertexto v 5.0 (HTML, en sus siglas en inglés)**

HTML es el lenguaje de programación básico para el desarrollo web *front-end*. Proporciona la estructura para el contenido de un sitio web y define palabras, títulos, párrafos, imágenes, etc. HTML consta de un conjunto de etiquetas predefinido, que representa diferentes funciones que “ convierten ” el contenido en un formato legible en la pantalla (Duarte, 2016).

En el desarrollo del sistema se utiliza HTML v5.0, porque ofrece nuevas funcionalidades como: la aparición de elementos parecidos en funcionalidad al elemento *div*, ya que este era el único capaz de separar las sesiones de las páginas web, con esta versión puede ser utilizados otros elementos (*section*, *article*, *aside* y *nav*); con respecto a las tablas el elemento *input* ha sido ampliado permitiendo introducir varios datos (*datetime*, *date*, *month*, *week*, *number*, *rango*, *email*, *url*, *search* y *color*); además admite introducir audios y videos. Dentro de los beneficios de su empleo se encuentran que describe con mayor precisión el contenido de la página y permite a las aplicaciones almacenar datos localmente en el lado del cliente y operar sin conexión.

#### **Hojas de estilo en cascada v 3.0 (CSS, en sus siglas en inglés)**

CSS es una hoja de estilo que describe cómo aparecerán los elementos HTML. En una página *web* CSS se utiliza para controlar la presentación, el estilo y el formato de los elementos HTML en uno

sitio *web*, configuración de colores, bordes, imágenes de fondo, etc. Los archivos CSS declaran un conjunto de reglas que definen propiedades y sus respectivos valores para elementos HTML.

El empleo de este lenguaje está dado por la necesidad de separar el contenido de la página y la forma de su representación. Se selecciona CSS v3.0 debido a que ofrece nuevas técnicas de presentación y estructuración como permitir las esquinas redondeadas, sombras, transparencia de color, texto en varias columnas y admite crear animaciones sin necesidad de instalar *plugins*.

### JavaScript (JS)

Se refiere a controlar el comportamiento de una página *web*. Es uno de los lenguajes de programación más popular y difundido en el mundo. El uso de JavaScript crea sitios web interactivos manipulando las especificidades de los elementos HTML y CSS. Con JavaScript, el usuario puede hacer clic uno botón y disparar una acción, hacer que el sitio se desplace hacia abajo o mostrar fotos aleatorias. Ejemplos de las bibliotecas más utilizadas en el *front-end* son *Ember*, *React*, *Backbone*, *Vue*, *Angular* y *Bootstrap*.

Actualmente es el principal lenguaje para la programación cliente-servidor en los navegadores *web*, es capaz de actuar dentro de documentos HTML y dar carácter dinámico. Posee soporte a la programación funcional y presenta recursos como cierres y funciones de alto nivel comúnmente disponibles (Castillo, 2017). Se justifica el empleo de este lenguaje debido a las características que posee como: la evaluación en tiempos de ejecución y que está basado en objetos, además todos los navegadores *web* pueden interpretar su código.

**Lenguajes de programación para el *back-end*:** El *back-end* consta del servidor *web* que aloja el sitio *web*, una aplicación para ejecutarlo y una base de datos para almacenar la información. Como el nombre sugiere, el desarrollado *back-end* funciona en la parte “posterior” de la aplicación, escribiendo programas que aseguren que el servidor, la aplicación *web* y la base de datos funcionen bien juntos. Los desarrolladores necesitan analizar cuáles son las necesidades de las partes interesadas y brindar soluciones de programación eficientes y seguras. Para esto existe una variedad de lenguajes del lado del servidor como PHP, *Ruby*, *Java* o *Python*. Además, dada la complejidad de las tecnologías involucrados en el *back-end* y la cantidad de requisitos de calidad que deben cumplirse (por ejemplo, rendimiento, confiabilidad, seguridad y disponibilidad del servicio), generalmente se deben utilizar *frameworks* de *back-end* para ayudar al desarrollador en muchos estos aspectos. A continuación, se muestra una breve descripción de cada uno de estos lenguajes que serán empleados para la implementación de la propuesta:

### Pre-procesador de Hipertexto v7.2.9 (PHP)

PHP es un lenguaje de escritura de propósito general y de código abierto que está especialmente pensado para el desarrollo *web* y que puede ser embebido en páginas HTML. El objetivo principal de este lenguaje es permitir a los desarrolladores *web* escribir dinámica y rápidamente páginas *web* (Cowburn, 2018).

Es necesario utilizar PHP v7.2.9, porque permite la conexión a diferentes tipos de servidores de BD como a PostgreSQL, MySQL y Microsoft SQL Server; además de estar desplegado en la mayoría de los servidores *web* y ajustarse a las necesidades del problema.

## Python

*Python* es un lenguaje de programación ampliamente utilizado en las aplicaciones *web*, el desarrollo de *software*, la ciencia de datos y el *machine learning* (ML). Los desarrolladores utilizan *Python* porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software *Python* se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

Permite separar el programa en módulos que pueden reusarse en otros programas. Viene con una gran colección de módulos estándar que puedes usar como base de tus programas, o como ejemplos para empezar a aprender a programar en *Python*. Algunos de estos módulos proveen cosas como entrada/salida a archivos, llamadas al sistema, *sockets*, e incluso interfaces a sistemas de interfaz gráfica de usuario.

En esta investigación *Python* se utilizará para el desarrollo *web* del lado del servidor. El desarrollo *web* del lado del servidor incluye las funciones complejas de *backend* que los sitios *web* llevan a cabo para mostrar información al usuario. Por ejemplo, los sitios *web* deben interactuar con las bases de datos, comunicarse con otros sitios *web* y proteger los datos cuando se los envía a través de la red.

*Python* es útil para escribir código del lado del servidor debido a que ofrece muchas bibliotecas que constan de código preescrito para crear funciones de *backend* complejas. Los desarrolladores también utilizan un amplio rango de marcos de *Python* que proporcionan todas las herramientas necesarias para crear aplicaciones *web* con mayor rapidez.

### 1.6.4. Marcos de trabajo para el desarrollo

En el desarrollo de *software*, un marco de trabajo (*framework*), es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de *software* concretos, en base a la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje informático interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto (Gutiérrez, 2014).

Es interesante resaltar que un mismo sistema *web* puede incorporar varios *frameworks*, de manera que cada uno de ellos está diseñado para soportar el desarrollo de una parte de la aplicación. Normalmente, el desarrollo en la web implica uno *front-end* (lado del cliente), código que interactúa con el cliente a través de un navegador y un *back-end* (lado del servidor), código ejecutado en el servidor *web*, que encapsula la lógica del negocio y la interacción de la aplicación con la base de datos. Los *frameworks* se pueden utilizar tanto en el *front-end* como en el *back-end*.

## Bootstrap

*Bootstrap* es un *framework* o kit de herramientas de código abierto para desarrollos *web responsive* con HTML, CSS y *JavaScript*. Se trata de un marco que se ocupa solo de las aplicaciones *front-end* (formularios, componentes de la interfaz y complementos de *JavaScript*). *Bootstrap* un *frameworks front-end* creado para fomentar la coherencia entre las numerosas bibliotecas disponibles en el mundo del desarrollo *web*. La última iteración, *Bootstrap 4* fue lanzada en el año 2014. Sin embargo, nuevas características y aparecen a menudo y a medida que la comunidad hace uso de su licencia gratuita de *Creative Commons*.

*Bootstrap* proporciona definiciones de estilo a través de HTML para garantizar la coherencia, lo que permite a los desarrolladores crear compilaciones en múltiples plataformas.

## Django

Existen varios tipos de *frameworks* diseñados para aprovechar los beneficios de *Python* aplicados a necesidades específicas, por ejemplo, *Django* y *Flask* asisten en el desarrollo de aplicaciones web. En esta investigación se utilizará *Django*. *Django* utiliza la estructura MTV (Modelo - Plantilla -Vista), donde el marco gestiona la mayor parte de la comunicación entre solicitudes HTTP.

De esta forma, provee un alto nivel de abstracción de patrones comunes en el desarrollo *web*, atajos para tareas frecuentes de programación y convenciones claras sobre cómo solucionar problemas. *Django* es una colección de bibliotecas escritas en el lenguaje de programación *Python*, para desarrollar un sitio usando *Django* es necesario escribir el código *Python* que utiliza esas bibliotecas. *Django* fue diseñado para promover el acoplamiento débil y la estricta separación entre las piezas de una aplicación.

*Django* es el *framework* que se escoge para la realización del sistema porque presenta múltiples posibilidades en el trabajo con las validaciones, integración con otras herramientas y es muy factible para realizar pruebas y depuración del código.

### 1.6.5. Entorno de Desarrollo Integrado

Los Entornos de Desarrollo Integrado (IDE, en sus siglas en inglés) son aplicaciones informáticas que ofrecen servicios integrales para facilitarle al desarrollador o programador la creación de *software*. Consiste en un editor de código fuente, herramienta de construcción automática y un depurador; además de poseer autocompletados. Dado que el lenguaje de programación seleccionado es *Python*, los IDE a considerar son *PyCharm*, *IDLE*, *Spyder*, *Atom*.

#### PyCharm

Se selecciona como IDE de desarrollo *PyCharm*. Este IDE cuenta con una edición comunitaria gratuita que es adecuada para pequeñas aplicaciones de *Python*, así como con una edición profesional de pago que es adecuada para crear aplicaciones de *Python* a gran escala, con el siguiente conjunto completo de características:

- Compleción automática e inspección del código
- Gestión de errores y correcciones rápidas
- Limpieza del código sin necesidad de cambios de funcionalidad
- Compatibilidad con los marcos de aplicaciones *web*, como *Django* y *Flask*
- Compatibilidad con otros lenguajes de programación, como *JavaScript*, *CoffeeScript*, *TypeScript*, *AngularJS* y *Node*
- Herramientas y bibliotecas científicas, como *Matplotlib* and *NumPy*
- Posibilidad de ejecutar, depurar, probar e implementar aplicaciones en máquinas virtuales remotas
- Un depurador para encontrar errores en el código, un creador de perfiles para identificar problemas de rendimiento en el código y un ejecutor de pruebas para llevar a cabo pruebas de unidad
- Soporte para bases de datos

Para el desarrollo del repositorio se seleccionó *PyCharm*, por permitir escribir, compilar, ensamblar y desplegar aplicaciones, además de brindar soporte para varios lenguajes de programación mediante el empleo de *plugins*. Facilita el completamiento para los lenguajes de programación HTML, CSS, y *JavaScript*.

#### Herramientas de Ingeniería de Software Asistida por Computadora

Las Herramientas de Ingeniería de Software Asistida por Computadora (CASE), son diversas aplicaciones informáticas o programas informáticos destinados a aumentar el balance en el desarrollo de *software* reduciendo el costo de los mismos en términos de tiempo (Battaglia et al.,

2016). Ayudan con todo el ciclo de vida del desarrollo de *software* en áreas como: el diseño de proyectos, cálculos de costes, implementación de parte del código de forma automática, compilación automática y documentación o detección de errores.

### **Visual Paradigm for UML v8.0**

Se emplea esta herramienta para visualizar y diseñar elementos de *software*. Esta utiliza el UML como lenguaje de modelado y ofrece una gama de facilidades para la conformación de aplicaciones. Está orientada a la creación de diseños usando el paradigma de la Programación Orientada a Objetos. Provee soporte para la generación de código, tiene integración con diversos Entornos Integrados de Desarrollo (IDE's), así como la ventaja de la ingeniería inversa para aplicaciones ya conformadas. Está disponible para múltiples plataformas (Windows y Linux).

Se selecciona esta herramienta porque ofrece un conjunto de funcionalidades útiles que permiten llevar a cabo el proceso de desarrollo de *software* y a su correspondiente integración con UML. Permite el aumento de la calidad del *software*, además de la mejora de la productividad en el desarrollo y mantenimiento del *software*.

### **1.6.6. Servidor Web**

Son los encargados de procesar las aplicaciones *web* del lado del servidor y del respectivo envío de información a través de la red. Se encargan de generar el tráfico en Internet. Una de sus funcionalidades es realizar conexiones bidireccionales o unidireccionales generando una respuesta que es mostrada al usuario mediante la aplicación *web*.

#### **Apache**

Apache es un servidor *web* altamente configurable, robusto y estable. Es un sistema de código abierto para plataformas *Windows*, *Unix*, *Macintosh* y otras que implementa el protocolo HTTP. Apache es un servidor *web* flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP. Es multiplataforma y modular, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, para el desarrollo de módulos específicos. También permite personalizar la respuesta ante posibles errores y posibilita configurarlo para que ejecute un determinado *script* cuando ocurra un error en concreto. Permite la creación de ficheros de *log* facilitando, de este modo, el control de las acciones realizadas en el servidor.

Se seleccionó el servidor *web* Apache debido a que consume pocos recursos, es gratuito, es uno de los más usados y presenta una amplia documentación.



### 1.6.7. Sistema Gestor de Bases de Datos

Los Sistema Gestor de Bases de Datos (SGBD) permiten definir las Bases de Datos, además de la elección de las estructuras de datos que estos sistemas necesitan para el almacenamiento y búsqueda información. Estos se han convertido en una herramienta eficiente, que puede apoyar a los especialistas para lograr una mejor confidencialidad, seguridad, calidad e integridad de los datos que estos almacenan.

#### PostgreSQL v11.0

*PostgreSQL* es un Sistema de Gestión de Base de Datos Relacional de Objetos. Utiliza el Lenguaje de Consulta Estructurada (SQL, en sus siglas en inglés) y ofrece características como: consultas complejas, llaves extranjeras y vistas dinámicas, entre otras. Puede ser empleado de forma gratuita para cualquier fin, ya sea privado, comercial o académico (Lockhart, 1998).

Se selecciona este SGBD debido a sus múltiples bondades, como: alta concurrencia, amplia variedad de tipos nativos, posibilita el empleo de claves ajenas, entre otras. Es elegida la v11.0 porque permite participación y rendimiento mejorado y compilación sobre la marcha.

### 1.7. Conclusiones del capítulo

Una vez realizado el estudio de la literatura científica, se pudo identificar la falta de acceso y permisos de administración a los sistemas investigados, ya que las estructuras burocráticas de gestión de la información dificultan la publicación y descargas de los Recursos Educativos Digitales empleados y/o creados por los docentes del departamento de Enseñanza Militar, por lo que se concluyó que existe la necesidad de desarrollar un repositorio para la gestión de contenidos de la Preparación para la Defensa en la Universidad de las Ciencias Informáticas. Para guiar el proceso de desarrollo de *software* se seleccionó el Escenario No. 2 de la metodología AUP-UCI, que requiere el modelado del negocio a través del Modelo Conceptual, y el modelado del sistema se realiza a partir de la especificación de Casos de Usos del Sistema.

## CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA

### 2.1. Introducción del capítulo

### 2.2. Modelado del contexto: modelo conceptual

Dado que en esta investigación se definió como metodología de desarrollo de *software*, AUP-UCI Escenario 2, para modelar el negocio se empleará el modelo conceptual. Un modelo conceptual explica los conceptos significativos en un dominio de problema del mundo real. Un modelo del dominio o modelo conceptual es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Se realiza cuando no se logra determinar el proceso del negocio con fronteras bien establecidas y donde los flujos de información son difusos (múltiples orígenes, sólo eventos, sucesos), cuando existe solapamiento de responsabilidades, así como múltiples responsabilidades. Se representa con un conjunto de diagramas de clases UML en los que no se define ninguna operación.

El modelo realizado constituye una de las entradas fundamentales para la posterior identificación de requisitos. En la Figura 1 se muestra el modelo conceptual de la investigación en curso.

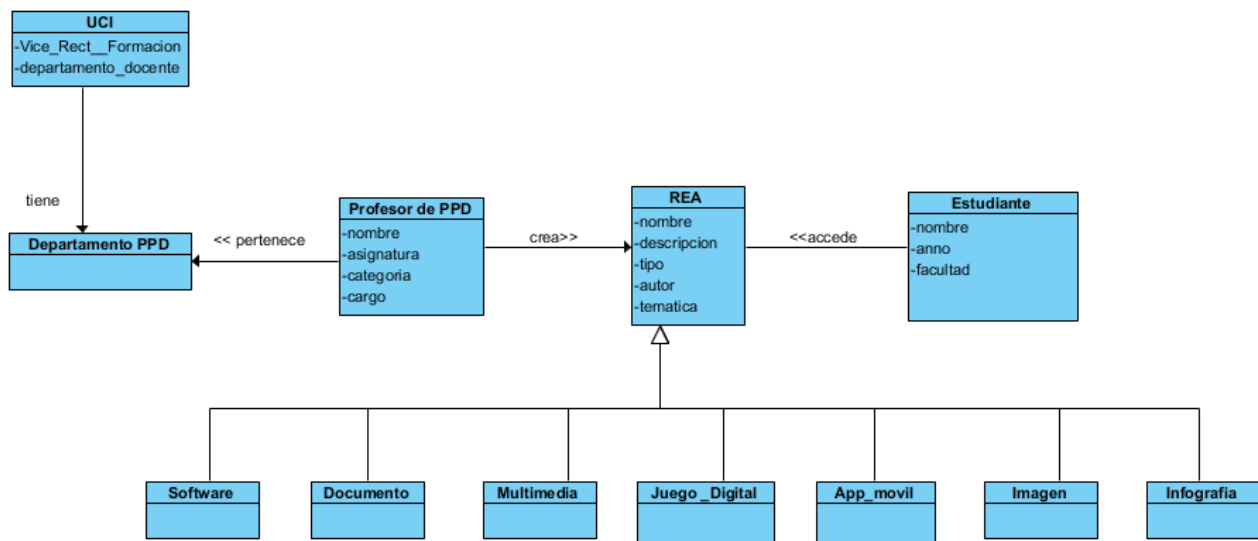


Figura 1. Modelo conceptual. Fuente: Elaboración propia.

### 2.3. Requisitos de software

Un punto crucial en la gestión de un proyecto de desarrollo informático es la determinación de los requisitos que debe cumplir para ser considerado una aplicación de calidad. En el contexto de los métodos de desarrollo de *software*, la ingeniería de requisitos es la piedra angular de todo proceso organizado que tenga por fin la obtención de sistemas informáticos que brinden solución verdadera a los problemas organizacionales. La delimitación de los requisitos debe contribuir a la obtención de

una primera descripción del problema que permita establecer las motivaciones para su solución mediante sistemas informáticos (Zapata & Arango Isaza, 2004).

### 2.3.1. Requisitos de software funcionales (RF)

Los RF son las declaraciones de las funcionalidades con que la aplicación debe contar, cómo debe reaccionar a la interacción del usuario, y cómo tiene que comportarse ante cada situación en particular. En algunos casos, los RF pueden declarar explícitamente lo que el sistema no debe hacer y lo que si debe permitir. Estos son normalmente descritos de una manera abstracta que puede entenderse por los usuarios (Sommerville, 2011). A continuación, se presentan los RF de la investigación.

#### Técnicas de obtención de requisitos

La obtención de requisitos es una de las cuatro actividades que define la Ingeniería de Requisitos, para ayudar a reconocer la importancia que tiene una especificación y administración adecuada de los requisitos funcionales. La obtención de requisitos es el proceso donde los interesados en un sistema de *software* descubren, revelan y entienden sus requerimientos. Existen diferentes técnicas para identificar los requisitos, dentro de las que se encuentran: sesiones de tormentas de ideas, entrevistas, observación de campo, revisión de la documentación técnica, análisis de sistemas existentes, ingeniería inversa, simulaciones y prototipos. Para la captura de requisitos de la solución propuesta se utilizaron las siguientes técnicas:

- **Sesiones de tormentas de ideas:** es una técnica de reuniones en grupo que permite generar ideas originales en un ambiente libre de críticas. Se realizaron reuniones con la participación de los tutores y profesores del departamento de Enseñanza Militar. También se contó con la participación de un especialista de la Editorial Ediciones Futuro de la UCI, con el objetivo de definir las funcionalidades y esclarecer las dudas en el proceso de gestión de contenidos de un repositorio institucional. A partir de las reuniones se encontraron algunas funcionalidades e interfaces que sirvieron de base para la propuesta de solución.
- **Entrevistas:** Se realizaron reuniones con las partes interesadas, donde se efectuaron una serie de preguntas para lograr una mayor comprensión de la problemática existente. Las entrevistas permitieron identificar varios de los requerimientos de la solución.
- **Prototipado:** un prototipo es una versión inicial de un sistema de *software* que se utiliza para demostrar los conceptos, probar las opciones de diseño y conocer más acerca del problema. Permitiendo formalizar la aceptación previa de los requisitos por los clientes. Los Prototipos de Interfaz de Usuario (PIU) se muestran en el Anexo 3.

La Tabla 3: muestra la especificación de requisitos funcionales.

**Tabla 3.** Especificación de requisitos funcionales. Fuente: Elaboración propia.

No.	Requisito Funcional	Descripción
RF 1.	Autenticar usuario	Permite que los usuarios se autenticquen en el sistema usando las credenciales institucionales del dominio UCI. El usuario se autentica por defecto con rol: Estudiante; excepto aquellos usuarios que el administrador le haya otorgado un rol especial (Profesor y/o administrador).
RF 2.	Adicionar usuario	Permite a un usuario con rol Administrador y/o Profesor, crear un usuario para cargar contenidos al sistema.
RF 3.	Modificar usuario	Permite modificar los datos de un usuario previamente creado.
RF 4.	Eliminar usuario	Permite a un usuario con rol Administrador eliminar un usuario del sistema.
RF 5.	Buscar usuario	Permite a un usuario con rol Administrador realiza la búsqueda de usuarios del sistema por: Correo electrónico; Nombre; y Rol.
RF 6.	Adicionar rol	Permite crear nuevos roles en el sistema. Cada rol tendrá asociado un conjunto de permisos dentro del repositorio. Los rolos básicos son: Administrador (Asignado por defecto al jefe de departamento); Profesor y Estudiante. Otros rolos pueden crearse como Invitado, investigador, revisor, entre otros.
RF 7.	Modificar rol	Permite modificar los datos y permisos asignados a cada rol.
RF 8.	Eliminar rol	Permite eliminar un rol del sistema. Importante: al eliminar un rol en específico, elimina todos los usuarios que tengan únicamente ese rol asignado.
RF 9.	Buscar rol	Permite buscar los roles creados en el sistema.
RF 10.	Asignar rol a usuario	Permite a un usuario con rol Administrador asignar roles especiales como: Profesor y Administrador.
RF 11.	Adicionar RED	Permite a un usuario con rol Administrador y/o Profesor enviar un RED al repositorio para ser almacenado
RF 12.	Modificar RED	Permite a un usuario con rol Administrador y/o Profesor modificar el envío, si el RED aún no ha sido publicado en el repositorio. Una vez publicado, solo el Administrador puede modificarlo.
RF 13.	Eliminar RED	Permite a un usuario con rol Administrador eliminar un RED del repositorio. Si el RED está publicado, primero se debe retirar y luego eliminar.
RF 14.	Publicar RED	Permite a un usuario con rol Administrador evaluar un recurso, y publicarlo en el repositorio en caso de que sea Aprobado.
RF 15.	Añadir autor del RED	Cada RED se agrega con el usuario autenticado como autor predefinido. Este requisito permite agregar otros autores al RED y eliminar/modificar el

		autor asignado por defecto. Un autor no es necesariamente un usuario. El usuario autenticado no es necesariamente el autor del RED.
RF 16.	Ver mi lista de envío	Muestra el listado de RED enviados por el usuario logueado en el repositorio y muestra el estado en que se encuentra: Enviado, Aprobado, Rechazado, Publicado.
RF 17.	Listar RED por autor	Permite a un usuario listar todos los RED dado el nombre del autor.
RF 18.	Listar RED por estado	Permite a un usuario con rol Administrador listar los RED según el estado. El estado puede ser: Enviado, Aprobado, Rechazado, Publicado.
RF 19.	Listar RED por tipo	Permite a un usuario listar todos los RED dado el tipo de recurso. Ejemplo: Video, Infografías, Presentación, Fotografías, Juegos, App, Audio, Documento.
RF 20.	Listar RED por temática	Permite a un usuario listar todos los RED dado la temática. Ejemplo: Seguridad Nacional, CiberGuerra, Defensa Civil, Derecho internacional.
RF 21.	Listar RED por colección	Lista todos los recursos que son asignados a una colección previamente creada.
RF 22.	Buscar RED	Permite buscar RED en el repositorio dado: nombre, autor, tipo, temática.
RF 23.	Adicionar colección	Permite adicionar una colección de RED. Una misma colección puede contener diferentes tipos de recursos y diferentes temáticas
RF 24.	Modificar colección	Permite modificar los datos de una colección y retirar los RED deseados. Importante: Eliminar un RED de una determinada colección, no lo elimina del repositorio.
RF 25.	Eliminar colección	Permite Eliminar una colección del repositorio. Eliminar una determinada colección, no lo elimina los RED del repositorio.
RF 26.	Buscar colección	Permite buscar colecciones dado el nombre de la misma.
RF 27.	Descargar RED	Permite a un usuario logueado en el sistema, descargar un RED.
RF 28.	Evaluar RED	Permite a un usuario logueado en el sistema emitir una evaluación sobre la calidad del RED Consultado.

### 2.3.2. Requisitos de software no Funcionales (RNF)

Los RNF describen atributos sólo del sistema o del ambiente del sistema que no están relacionados directamente con los RF. Los RNF incluyen restricciones cuantitativas, como el tiempo de respuesta o precisión, lenguajes de programación, etcétera (Sommerville, 2011). A continuación, se exponen los RNF del repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa en la Universidad de las Ciencias Informáticas.

De acuerdo a las características propias del repositorio que se desea implementar, y teniendo en cuenta la infraestructura tecnológica en la cual se prevé el despliegue del repositorio, así como temas

de seguridad e identidad institucional, fue posible definir un conjunto de requisitos no funcionales, tal como se muestra en la Tabla 4.

**Tabla 4.** Requisitos no funcionales. **Fuente:** Elaboración propia.

Requisito no Funcional	No.	Descripción
Usabilidad	RNF 1.	El Repositorio debe ofrecer una interfaz de usuario intuitiva, que permita la fluidez de la interacción con sus funcionalidades tanto para realizar un envío como para descargar los RED.
Rendimiento	RNF 2.	El Repositorio debe responder a las peticiones del usuario en un tiempo promedio no superior a 1500 milisegundos.
	RNF 3.	El sistema debe permitir que existan al menos 500 usuarios conectados de forma simultánea haciendo peticiones concurrentes.
Hardware	RNF 4.	La PC cliente debe contener como mínimo las siguientes propiedades: 4 GB de RAM, procesador 3.3 GB y una tarjeta de red.
	RNF 5.	La PC servidor de la aplicación web debe poseer como mínimo 4 GB de RAM, 1 TB de almacenamiento, un procesador de 3.3 GB y una tarjeta de red.
	RNF 6.	La PC servidor de la BD debe poseer como mínimo 4 GB de RAM, 1 TB de almacenamiento y un procesador de 3.3 GB.
Software	RNF 7.	La PC Cliente debe tener como Navegador Web alguno de los siguientes: <i>Chrome</i> , <i>Mozilla Firefox</i> , <i>Internet Explorer</i> , <i>Opera</i> u otros que cumpla con los estándares de la <i>Word Wide Web Consortium (W3C)</i> .
	RNF 8.	El servidor de la BD debe tener PostgreSQL v11.0.
	RNF 9.	El servidor de la aplicación web debe contar con el servidor web Apache.
Seguridad	RNF 10	Se define el acceso al repositorio de contenidos y sus funcionalidades mediante la asignación de permisos a través de roles de usuarios.
	RNF 11	Los mensajes de errores deben mostrar la menor cantidad de detalles posibles, para evitar ofrecer información que comprometa la seguridad e integridad del sistema.
	RNF 12	A los usuarios autorizados se les deberá garantizar el acceso a la información solicitada en todo momento.
Restricciones en el diseño y	RNF 13	Las funciones de cada interfaz del repositorio, serán habilitadas según el rol del usuario que esté logueado en el sistema.

la implementación		
----------------------	--	--

## 2.4. Modelo de casos de uso del sistema

El modelo de casos de uso del sistema permite que los desarrolladores de *software* y los clientes lleguen a un acuerdo sobre los requisitos, es decir sobre las condiciones y posibilidades que debe cumplir el sistema. Además, proporciona la entrada fundamental para el análisis, diseño y pruebas.

### 2.4.1. Actores del sistema

Se considera un actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Por lo general estimula al sistema con eventos de entrada o recibe algo de él. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Para el acceso a las funcionalidades que se desarrollarán se cuenta con varios actores, que se definen a continuación.

Tabla 5. Actores del sistema. Fuente: Elaboración propia.

Actor	Funciones del actor del sistema
Usuario	Es un usuario común que se autentica con las credenciales del dominio UCI. Todos los usuarios pueden descargar los RED publicados en el repositorio
Jefe de departamento	Revisa los RED enviados por los profesores para determinar qué recursos respaldan de manera efectiva los objetivos de aprendizaje y pueden publicarse en el repositorio. Tiene los permisos asociados al rol administrador.
Profesor	Es el encargado de enviar los RED al repositorio para que sean publicados.
Estudiante	Puede emitir una evaluación por cada RED descargado, en dependencia de la utilidad y facilidad de uso que el RED tuvo para su aprendizaje autónomo.

### 2.4.2. Diagrama de casos de uso del sistema

Un diagrama de casos de uso del sistema (DCUS) representa gráficamente a los procesos y su interacción con los actores. Además, facilita el entendimiento de los procesos realizados por el sistema para el desarrollador. La figura 3 muestra el DCUS diseñado para el Repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa en la UCI.

Los 28 Requisitos funcionales definidos para esta investigación, fueron agrupados en 13 Casos de Usos del Sistema, haciendo uso de los siguientes patrones de casos de uso:

- Concordancia (*Commonality*) por Adición: Representado por el caso de uso “Adicionar autor” que es una subsecuencia del caso de uso “Realizar envío”.

- CRUD (*Creating, Reading, Updating, Deleting*) Completo: Representado por los casos de uso “**Gestionar rol**” y “**Gestionar usuario**”.
- Múltiples Actores Roles Comunes: Representado por los casos de uso “**Autenticar usuario**” y “**Descargar RED**”.

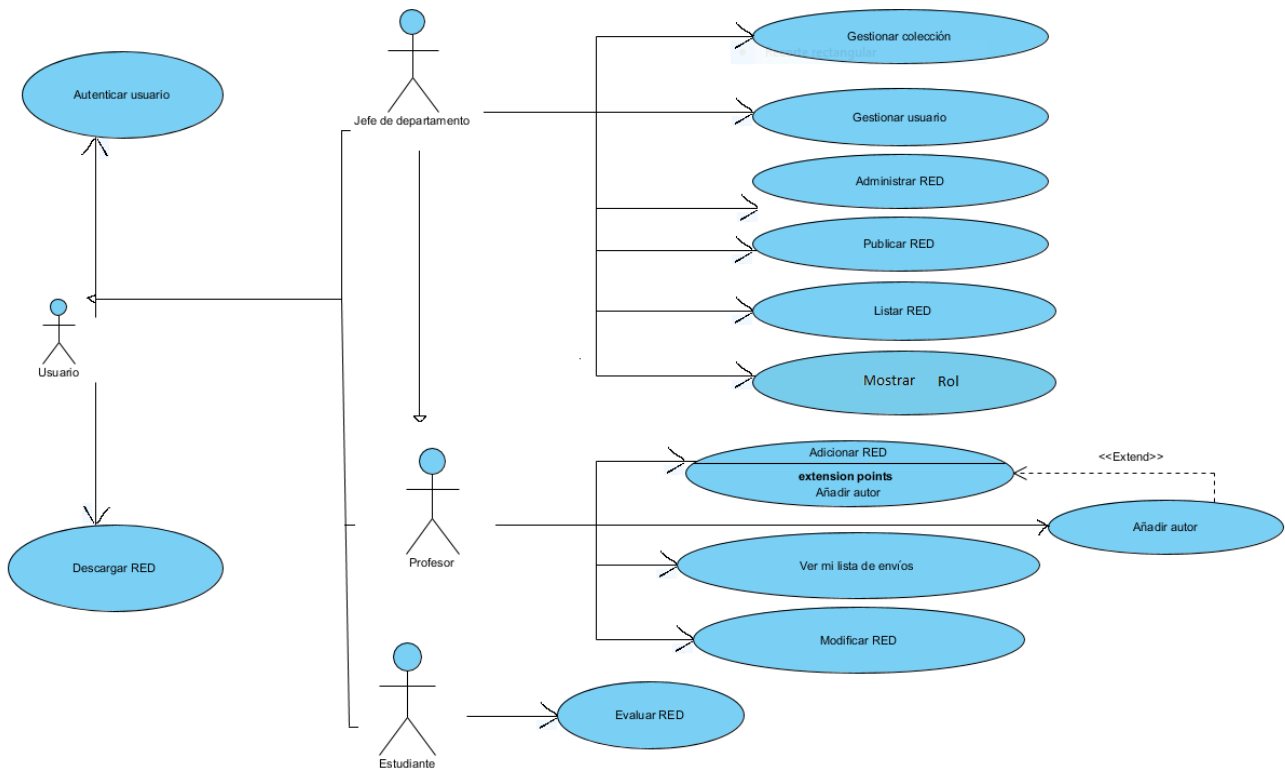


Figura 2. Diagrama de casos de uso del sistema. **Fuente:** Elaboración propia.

### 2.4.3. Especificación de casos de uso del sistema

La especificación de casos de uso del sistema es la declaración oficial de que deben implementar los desarrolladores del sistema. Debe incluir una especificación detallada de los requisitos del sistema. En este epígrafe se especifican los casos de uso Gestionar rol, Realizar envío y Añadir autor.

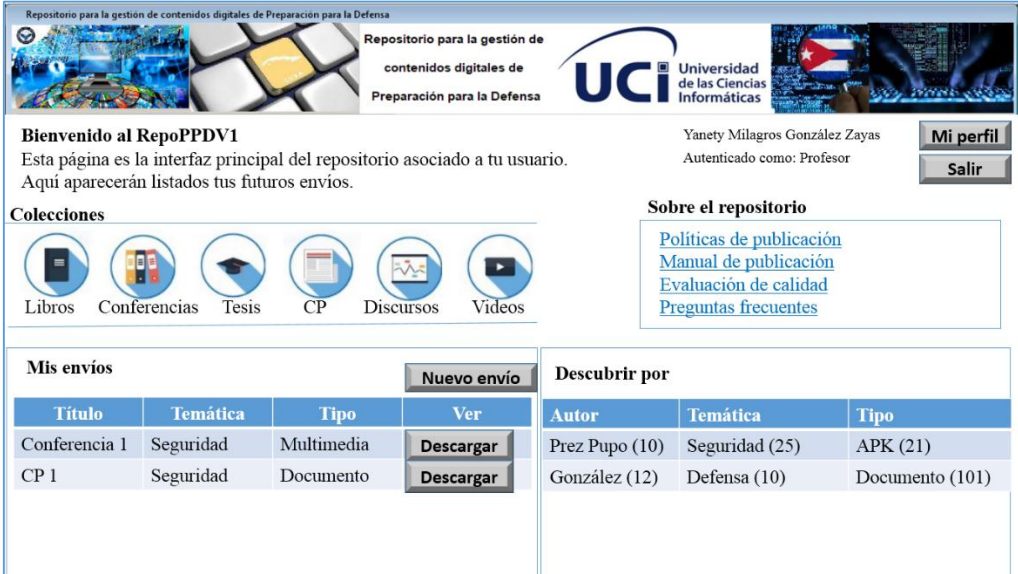
**Tabla 6.** Especificación del caso de uso: Realizar envío **Fuente:** Elaboración propia.

<b>Objetivo</b>	Realizar envío.
<b>Actores</b>	Jefe de departamento, Profesor.
<b>Resumen</b>	Los actores autenticados en el sistema escogen la opción “Nuevo envío” de la interfaz principal. Una vez abierta la interfaz “Realizar envío” introducen los metadatos correspondientes y cargan el RED que desean enviar al repositorio.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítica.
<b>Precondiciones</b>	Tiene que tener permiso y estar autenticado para realizar las acciones.



<b>Postcondiciones</b>	Se realiza el envío del RED y se guarda con estado “Enviado” por defecto. Se muestra en el listado “Mis envíos” (Ver PIU_Interfaz Principal Profesor), pero no aparece publicado en el repositorio hasta que su estado sea “Publicado” (Se ejecuta el caso de uso “Publicar RED”).	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción “Nuevo envío” en la interfaz principal del actor autenticado (Ver PIU_Figura 3).	
2.		<p>Muestra la interfaz realizar envío con la pestaña “<b>1. Inicio</b>” seleccionada por defecto, y muestra los siguientes campos que deben ser llenados:</p> <ul style="list-style-type: none"> <li>• <b>Tipo de archivo:</b> es componente de selección. Debe especificar un tipo de archivo: Imagen, Video, Juego, Gráfico, Mapa, Infografía, Aplicaciones móviles (<i>Apk</i>), Documentos, ect.</li> <li>• <b>Formato del archivo:</b> Se selecciona el formato, dependiendo del tipo definido anteriormente. Ejemplo, para una Imagen se selecciona formato png, jpg, gif, ect.</li> <li>• <b>Temática:</b> Se selecciona la temática a la cual está relacionada el RED: Seguridad Nacional, CiberGuerra, Defensa Civil, Derecho internacional, ect.</li> <li>• <b>Requisitos de envío:</b> Se selecciona uno o varios requisitos de envío según sea el caso.</li> </ul>
3	Se presiona la opción <b>Guardar y continuar.</b>	
4		Se muestra el formulario “ <b>2. Cargar el envío</b> ”. Este formulario permite arrastrar un archivo, o abrir el explorador al presionar el botón “ <b>Buscar archivo en la PC</b> ”
5	Se carga el archivo y se presiona la opción <b>Guardar y continuar.</b>	
6		<p>Se muestra el formulario “<b>3. Introducir metadatos</b>”, y muestra los siguientes campos que deben ser llenados:</p> <ul style="list-style-type: none"> <li>• <b>Título:</b> Campo de texto que admite letras, números, y otros símbolos. Debe especificarse el título oficial del RED.</li> <li>• <b>Título corto:</b> Campo de texto que admite letras, números, y otros símbolos. Se introduce una nomenclatura común, ejemplo: Conferencia 1.</li> <li>• <b>Resumen:</b> Se realiza un resumen de los principales elementos que caracterizan el RED. Debe especificarse, por ejemplo: Área del conocimiento, Institución productora, Audiencia a quien va dirigido, Palabras Clave, Competencias que promueve, Licenciamiento (derechos de autor), Idioma.</li> <li>• <b>Autores:</b> Aparece por defecto como autor el usuario que realiza el envío. El usuario autenticado no es necesariamente el autor del RED, por lo tanto, se puede eliminar y/o añadir otros autores. Ver caso de uso extendido: “<b>Añadir autor del RED</b>”</li> </ul>

		<b>Requisitos de envío:</b> Tiene que tener el Título y al menos un autor.
7	Se presiona la opción <b>Guardar y continuar</b>	
8		Se muestra la pestaña <b>"4. Confirmación"</b> , y se da la posibilidad de modificar metadatos a partir de la navegación por las pestañas anteriores
9	Una vez revisado el envío, se presiona la opción <b>"Enviar"</b>	
10		Se validan los campos, si hay un campo obligatorio vacío se muestra un mensaje: <i>"El campo &lt;&lt; nombre del campo &gt;&gt; es obligatorio"</i> .
11		Se realiza el envío del RED. El estado es <b>"Enviado"</b> por defecto. Si el usuario autenticado tiene rol Profesor, el envío se muestra en la interfaz de inicio (Ver PIU_Figura 6), en el listado <b>"Mis envíos"</b> . Si el rol autenticado es un Administrador, el envío aparece en la interfaz <b>"Administrar envío"</b> , del Caso de uso <b>"Administrar envío"</b> .
<b>Flujo alterno</b>		
<b>"Escoge la opción cancelar."</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1; 5.1; 7.1; 9.1	Selecciona la opción <b>"Cancelar"</b> , que se encuentra en el borde inferior derecho, de la interfaz.	
		El sistema muestra el mensaje <i>"Aún no ha finalizado el envío, si cancela perderá los metadatos introducidos"</i>
	El usuario selecciona la Opción <b>"Aceptar"</b> del mensaje de alerta	
		El sistema cierra la interfaz <b>"Realizar envío"</b>
		Finaliza el flujo alterno.
<b>Caso de uso extendido</b>		
<b>"Añadir autor del RED"</b>		
6.1	Selecciona la opción <b>"Añadir autor"</b> , del formulario <b>"3. Introducir metadatos"</b> (Ver PIU_Figura 6).	
6.2		Se muestra la ventana flotante <b>"Añadir autor"</b> con los siguientes campos: <ul style="list-style-type: none"> <li>• <b>Nombre*</b>: Especifica el nombre del autor. Es un campo obligatorio. Solo acepta letras.</li> <li>• <b>Apellidos</b>: Especifica los apellidos del autor. Solo acepta letras.</li> <li>• <b>Categoría Docente</b>: Permite seleccionar la categoría docente del profesor: Instructor, Asistente, Auxiliar, Titular.</li> <li>• <b>Departamento</b>: Especifica el departamento docente al cual pertenece el autor.</li> <li>• <b>Contacto*</b>: Especifica el correo de contacto del autor. Es un campo obligatorio. Acepta letras, números y caracteres especiales.</li> </ul>

6.3	Presiona la opción "Añadir"	
6.4		Se adiciona el autor al envío. El autor aparece en el listado de autores del formulario "3. Introducir metadatos" (Ver PIU_Figura 6).
<b>Flujo alternativo</b>		
<b>"Escoge la opción cancelar."</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la opción "Cancelar", que se encuentra en el borde inferior derecho, de la ventana flotante "Añadir autor".	
2		Finaliza el flujo alternativo.
<b>Requisitos funcionales</b>	no	Seguridad.
<b>Prototipos de Interfaz usuario</b>	de	 <p>The screenshot shows the main interface of the 'Repositorio para la gestión de contenidos digitales de Preparación para la Defensa'. It includes a header with the UCI logo and navigation links. A welcome message is displayed for user Yanety Milagros González Zayas, authenticated as 'Profesor'. Below this, there are sections for 'Colecciones' (Libros, Conferencias, Tesis, CP, Discursos, Videos) and 'Mis envíos' (Conferencia 1, CP 1) with a 'Nuevo envío' button. A 'Descubrir por' section shows a list of authors and their associated content counts.</p>
<b>Figura 3. PIU_Interfaz Principal Profesor.</b>		

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

Bienvenido al RepoPPDV1

Yanety Milagros González Zayas  
Autenticado como: Profesor

Mi perfil  
Salir

**Realizar Envío**

1. Inicio 2. Cargar el envío 3. Introducir metadatos 4. Confirmación

Tipo de archivo: Seleccione Formato del archivo: Seleccione Temática: Seleccione

Requisitos del envío:

- El RED fue creado y utilizado con fines pedagógicos.
- El RED solo contiene materiales digitales (ilustración, fotografía, diagrama, audiovisual) libres de restricciones sobre su uso; son de dominio público; posee la licencia para su uso.
- El RED cita y referencia correctamente todos los trabajos utilizados en su creación.
- Cuando el RED no es creado por usted. Referencia correctamente el autor.

Guardar y continuar Cancelar

Figura 4. PIU\_Realizar envío\_ 1: Inicio.

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

Bienvenido al RepoPPDV1

Yanety Milagros González Zayas  
Autenticado como: Profesor

Mi perfil  
Salir

**Realizar Envío**

1. Inicio 2. Cargar el envío 3. Introducir metadatos 4. Confirmación

Buscar archivo en la PC

Para cargar el archivo puede arrastrarlo a este cuadro de dialogo o buscar el archivo en la PC.  
Para subir el archivo presione: [Subir archivo](#).

Guardar y continuar Cancelar

Figura 5. PIU\_Realizar envío\_ 2: Cargar el envío.



Figura 6. PIU\_Realizar envío\_ 3: Introducir metadatos.

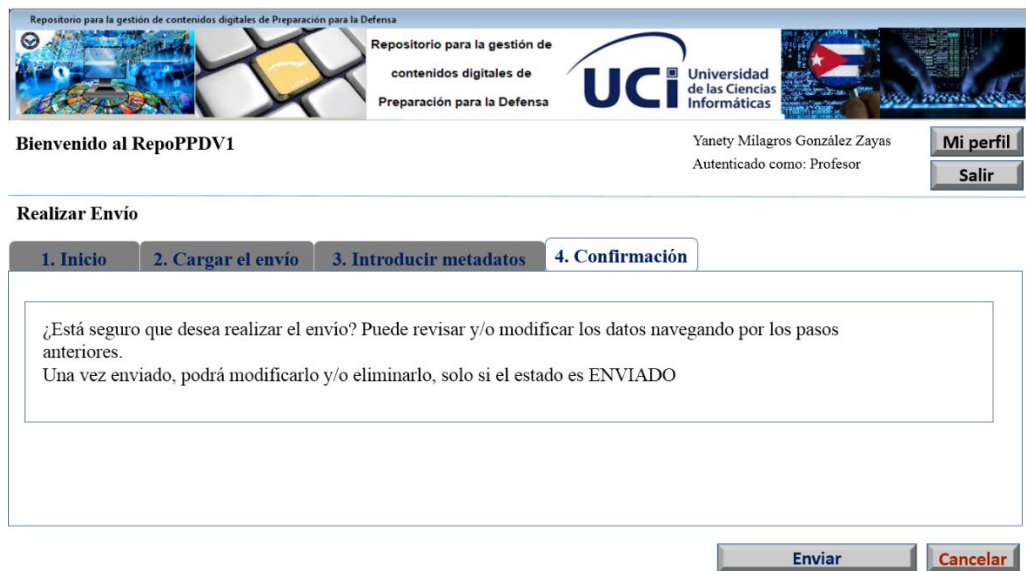


Figura 7. PIU\_Realizar envío\_ 4: Confirmación.

Tabla 7. Especificación del caso de uso: Gestionar rol **Fuente:** Elaboración propia.

<b>Objetivo</b>	Gestionar rol.
<b>Actores</b>	Jefe de departamento
<b>Resumen</b>	El administrador escoge en el sistema una opción “ <b>Gestionar roles</b> ” y se muestra la lista de roles adicionados en el sistema, con un botón de búsquedas y con la posibilidad de adicionar, modificar y eliminar el rol.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítica.
<b>Precondiciones</b>	Tiene que tener permiso y estar autenticado para realizar las acciones.

<b>Postcondicione s</b>	Se adiciona, modifica, elimina o busca uno o varios roles.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción " <b>Gestionar roles</b> "	
2.		Muestra un listado con los roles almacenados y posibilita al jefe de departamento ejecutar las siguientes acciones (Ver PIU_Figura 8): Adicionar rol. Ver <b>Sección 1</b> Modificar rol. Ver <b>Sección 2</b> Eliminar rol. Ver <b>sección 3.</b> Buscar rol. Ver <b>sección 4.</b>
3.	El jefe de departamento escoge una de las cuatro opciones.	
4.		Finaliza el caso de uso.
<b>Sección 1: "Adicionar rol"</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la opción " <b>Añadir rol</b> " de la interfaz <b>Gestionar roles.</b>	
2		Muestra un formulario de registro con los siguientes campos (Ver PIU_Figura 9). Se llenan los campos: <ul style="list-style-type: none"> <li>• <b>Nombre y apellidos:</b> solo acepta letra.</li> <li>• <b>Categoría docente:</b> solo admite: si es Titular o Auxiliar.</li> <li>• <b>Departamento:</b> solo acepta letra.</li> <li>• <b>Rol:</b> solo acepta letra.</li> </ul> <b>Contacto:</b> solo acepta letra.
3	Se presiona la opción <b>Adicionar.</b>	
4		Se validan los campos,  Si hay un campo vacío se muestra un mensaje "Rellene todos los campos".
5	Se presiona el botón <b>Aceptar</b>	
6		
<b>Flujo alternativo</b>		
<b>"Escoge la opción cancelar."</b>		
	<b>Actor</b>	<b>Sistema</b>
5.1	Selecciona la opción "Cancelar", que se encuentra en el borde inferior derecho, de la página principal de la sección "Adicionar persona".	
5.1.1		Finaliza el flujo alternativo.
<b>Sección 2: "Eliminar rol"</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>

1	Selecciona la opción "Eliminar" de la página principal de las evidencias.	
2		Se muestra una ventana con la lista de roles.
3	Seleccione el rol que desee y da clic en el botón "Eliminar".	
4		Se muestra un mensaje "Desea eliminar el rol".
5	El jefe de departamento da clic en el botón "sí".	
6		Se muestra un mensaje de notificación "Rol eliminado correctamente".
<b>Flujo alterno</b>		
<b>"Escoge la opción cancelar."</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la opción "Cancelar", que se encuentra en el borde inferior derecho, de la página principal de la sección "Eliminar criterio".	
2		Finaliza el flujo alterno.
<b>Sección 3: "Modificar rol"</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la opción "Actualizar" de la página principal de los criterios.	
2		Se muestra un formulario (nombre y descripción).
3	El jefe de departamento modifica los campos del formulario de criterios y da clic en el botón "Aceptar".	
4		Se muestra un mensaje de notificación "Criterio actualizado correctamente".
<b>Flujo alterno</b>		
<b>"Escoge la opción cancelar."</b>		
1	Seleccione la opción "Cancelar", que se encuentra en el borde inferior derecho, de la página principal de la sección "Actualizar criterio".	
2		Finaliza el flujo alterno.
<b>Sección 4: "Buscar rol"</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la opción "Buscar" de la página principal de los criterios.	
2		Permite la búsqueda del criterio por el nombre del mismo (nombre).
3	El jefe de departamento realiza la búsqueda del criterio y da clic en el botón "Buscar".	
4		Muestra el criterio buscado en caso de existir, sino muestra el mensaje "El criterio no se encuentra registrado".
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad.

**Prototipos de Interfaz de Usuario**

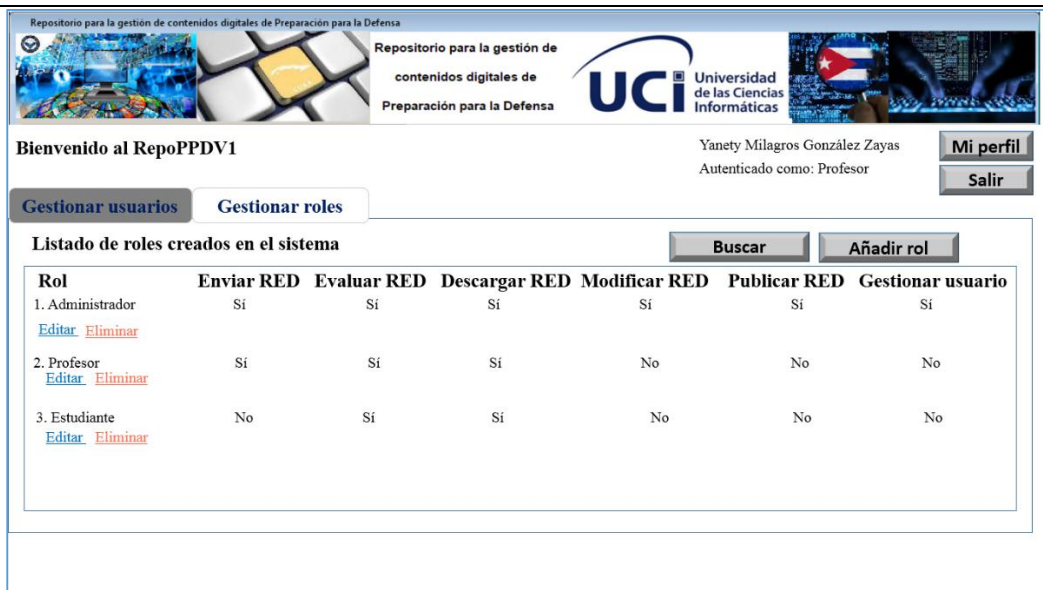


Figura 8. PIU\_ Gestionar Rol.

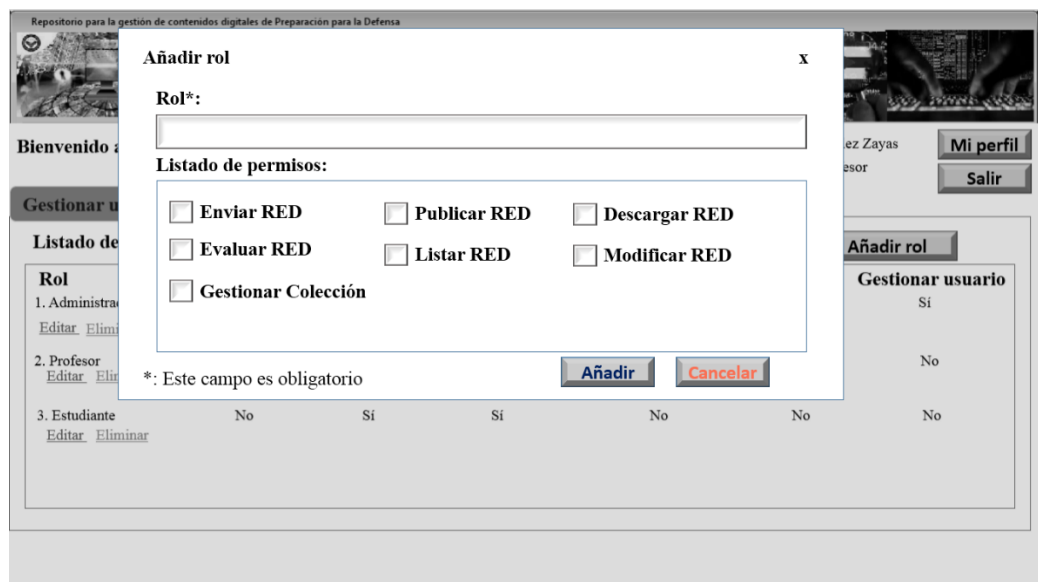


Figura 9. PIU\_ Gestionar Rol.

## 2.5. Modelo del diseño

La arquitectura de *software* se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Representa un diseño de alto nivel que tiene dos propósitos primarios: satisfacer los atributos de calidad y servir como guía en el desarrollo.

El modelo de diseño es un refinamiento y formalización del modelo de análisis, donde se toman en cuenta las consecuencias del ambiente de implementación. El resultado del modelo de diseño son especificaciones detalladas de todos los objetos, incluyendo sus operaciones y atributos. El Modelo



de Diseño es una disciplina que no se puede obviar en el proceso de desarrollo del *software*. Es imprescindible para comprender la forma en que va a funcionar el sistema en conjunto con los requisitos, lenguajes de programación, componentes reutilizables y tecnologías de interfaz de usuario que se eligieron para el desarrollo del mismo. Es una representación gráfica, mediante varios diagramas muy explícitos, de la implementación del sistema.

### 2.5.1. Diagrama de clases del diseño

El diagrama de clases del diseño muestra los bloques de construcción de cualquier sistema orientado a objetos. Permite representar gráficamente y de manera estática la estructura general de un sistema, mostrando cada una de las clases y sus interacciones (como herencias y asociaciones), representadas en forma de bloques, los cuales son unidos mediante líneas y arcos. A continuación, se representa el diagrama de clases del diseño correspondiente al caso de uso Administrar rol del repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa en la UCI.

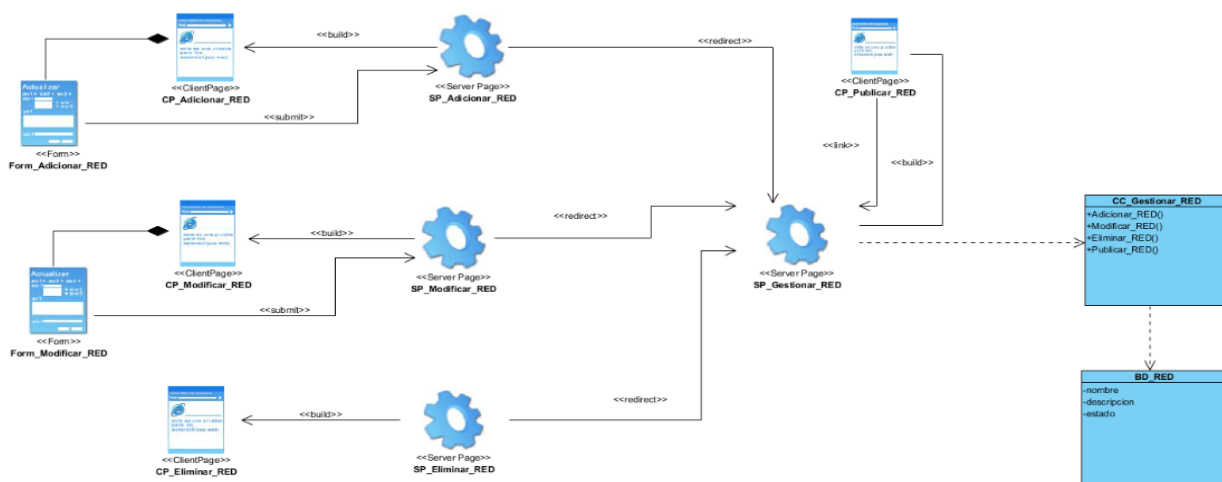


Figura 3. Diagrama de clases del diseño. Fuente: Elaboración propia.

### 2.5.2. Descripción de las clases del diseño.

Las clases de tipo `<<Client page>>` son las encargadas de recibir y procesar las peticiones del usuario. Estas tienen la responsabilidad de mostrar los formularios para realizar funcionalidades, tales como: adicionar rol y modificar rol.

La clase `DesignerController` tiene la responsabilidad de interpretar todas las peticiones del usuario. Esta es una clase servidora que se encarga de manejar el flujo de eventos e información.

La clase `LayersController` contiene todos los métodos que van a ser necesarios para crear o eliminar

capas y modificar o listar sus atributos. La clase *Layer* es la encargada de contener toda la información referente a una capa. Brinda un modelo para administrar esta información.

### 2.5.3. Patrón Arquitectónico Modelo - Plantilla – Vista

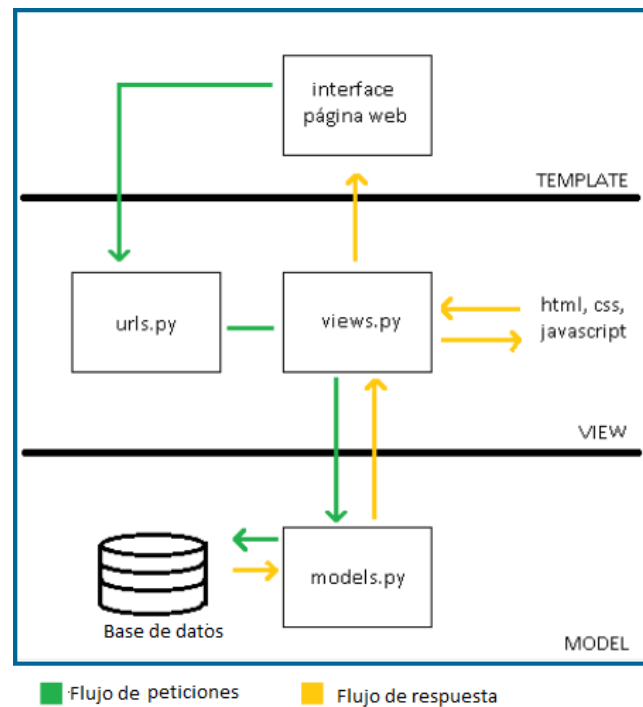
Según (Huaman, 2018) un patrón arquitectónico es una solución general y reutilizable a un problema común en la arquitectura de *software* dentro de un contexto dado. Son similares al patrón de diseño de *software*, pero tienen un alcance más amplio. Además, constituye una colección de decisiones de diseño arquitectónico, que tiene un nombre específico y son parametrizadas para tener en cuenta diferentes situaciones durante el desarrollo de *software*.

La mayoría de los *framework* para el desarrollo *web*, tienen su arquitectura basada en el patrón Modelo -Vista- Controlador (MVC) o en sus variaciones. Esto le permite separar las representaciones internas de los datos de la forma en que estos datos son presentados a los usuarios. Dado que el *framework* empleado en esta investigación es *Django*, se utilizará entonces la estructura Modelo - Plantilla -Vista (MTV), representada en la Figura 2, donde el *Django* gestiona la mayor parte de la comunicación entre solicitudes HTTP.

Cuando un proyecto se implementa en *Django*, se tiene un directorio de inicio de carpetas y archivos donde están los archivos *models.py*, *views.py* y *urls.py*, responsables de la transmisión de MTV. A continuación, se explican las capas de MTV y cómo funciona el flujo entre ellas:

- **Modelo (Model):** La capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos. *Django* implementa una solución para Mapeo Relacional de Objetos (ORM) en el que se describe el esquema de la base de datos en código *Python*. En este caso, las abstracciones en *Python* se pueden utilizar para crear consultas complejas sin que sea necesario realizar acciones directas a la base de datos.
- **Plantilla (Template):** Esta es la capa de presentación donde se almacena la información. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página *web* u otro tipo de documento. La plantilla consta de partes estáticas del archivo HTML de salida y partes con un sintaxis especial que describe cómo el contenido. *Django* tiene una ruta de búsqueda de plantillas, lo que permite minimizar la redundancia entre ellas. Generalmente, la vista recupera los datos en consecuencia con los parámetros de búsqueda, carga una plantilla y la renderiza con los datos recuperados.
- **Vista (View):** la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelo y las plantillas. Las vistas reciben la información y el tipo de solicitud (“POST” o “GET”)

del lado del cliente y luego formatean los datos para que sean almacenados en la base de datos a través de modelos en la capa Modelo. Las vistas también se comunican a través de modelos con la base de datos para recuperar datos que posteriormente se transfieren a las plantillas, para visualizarlas al usuario. Cada vista es responsable de hacer una función como devolver un objeto `HttpResponse` que contiene el contenido de la página solicitada, o levantar una excepción como `Http404`.



**Figura 4.** Flujo del patrón arquitectónico MTV de Django. Fuente: Tomado de Internet.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control en esta investigación, es el siguiente:

- El usuario interactúa con la interfaz de usuario, por ejemplo, al pulsar un enlace. Este punto corresponde en el patrón a la vista.
- El controlador recibe (a través de la interfaz) la notificación de la acción solicitada por el usuario. Es decir, el controlador gestiona el evento que llega desde la vista producida por un usuario.
- El controlador accede al modelo, ya sea con el fin de consultar datos o actualizarlos, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado de las películas que tal usuario tiene). El modelo no debe tener conocimiento directo sobre la vista. Por lo general, el

controlador no pasa objetos de dominio (el modelo) a la vista, aunque puede dar la orden a la vista para que se actualice. Sin embargo, en algunas implementaciones, la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.

- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

#### 2.5.4. Patrones de diseño arquitectónico

Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de *software*. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Propone que son plantillas para arquitecturas de *software* concretas, que especifican las propiedades estructurales de una y tienen un impacto en la arquitectura de subsistemas. La selección de un patrón arquitectónico es, por lo tanto, una decisión fundamental de diseño en el desarrollo de un sistema de *software*.

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de *software*. Se utilizan para crear propuestas de soluciones reusables y con documentación ya probada, partiendo de técnicas que se han aplicado en problemas similares. Estableciendo un lenguaje estándar entre todos los miembros de un equipo.

#### Patrones para la asignación de responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos. Son un método de enseñanza que ayuda a entender el Diseño de programación Orientada a objeto y aplica el análisis y diseño de objetos de un modo sistemático, racional y explicable. Para el desarrollo del módulo se hizo uso de los patrones siguientes:

- El patrón **Experto (Expert)** es el encargado de asignar responsabilidades a las clases que tiene la información necesaria para cumplir dicha responsabilidad. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Hay que tener en cuenta que esto es aplicable mientras estemos considerando los mismos aspectos del sistema. La utilización de dicho patrón se ve aplicada en clases como: *LayerModel* y *Layer*, debido que cada una de ellas es responsable de manejar su información.
- El patrón **Creador (Creator)** es quien guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Está aplicado en la clase *LayersController*, la cual utiliza la información de la clase *LayerModel* para crear objetos de ella.

- El patrón **Bajo Acoplamiento (*Low Coupling*)** estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento, tanto que produzca los resultados negativos propios de un alto acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecienten la oportunidad de una mayor productividad. Se ve empleado en todo el diseño del módulo ya que cada clase solo se relaciona con otras que necesite, para evitar que si se realiza algún cambio en algunas de ellas se vean afectadas las otras clases lo menos posible.
- El patrón **Alta Cohesión (*High Cohesion*)** es el encargado de que la información que contiene una clase debe de ser coherente, altamente relacionada con dicha clase y que la misma no realice una excesiva cantidad de trabajo. Posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande. Se refleja en las clases *Layer* y *LayerModel* ya que cada clase contiene la información y los métodos correspondientes exclusivamente a ellas, logrando que no se sobrecarguen con información de otras clases.

### Patrones GOF.

Los patrones GOF describen soluciones simples y elegantes a problemas específicos en el diseño de *software* orientado a objetos. Permiten enfrentarse a la programación de *software* propiciando reutilización y extensibilidad de soluciones que han funcionado en el pasado. Se clasifican según el propósito para el que han sido definidos en:

- **Singleton:** El patrón de diseño *Singleton* (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Se evidencia en la clase *DesignerController*, la cual define varias variables globales de instancia única que permiten la comunicación con el modelo y demás clases.
- **Decorator:** Permite añadir funcionalidad extra a un objeto (de forma dinámica o estática) sin modificar el comportamiento del resto de objetos del mismo tipo. Se ve reflejado en las clases cliente de la vista al heredar el código que es común en todas las páginas del sistema, para no tener que repetirlo en cada una de estas.
- **Observer:** Los objetos son capaces de suscribirse a una serie de eventos que otro objetivo va a emitir, y serán avisados cuando esto ocurra. Este patrón se utiliza en la clase *LayersController* para controlar las llamadas sincrónicas a la base de datos; así como en las clases del paquete vista para realizar las acciones deseadas.

### Modelo de datos

El Diagrama Entidad-Relación (DER) tiene como objetivo representar mediante una notación gráfica los objetos de datos y sus relaciones. Este define todos los datos que se introducen, se almacenan,

se transforman y se producen dentro de una aplicación. En la figura 12 se muestra el DER del Repositorio para la gestión de contenidos digitales de la Preparación para la Defensa en la Universidad de las Ciencias Informáticas.

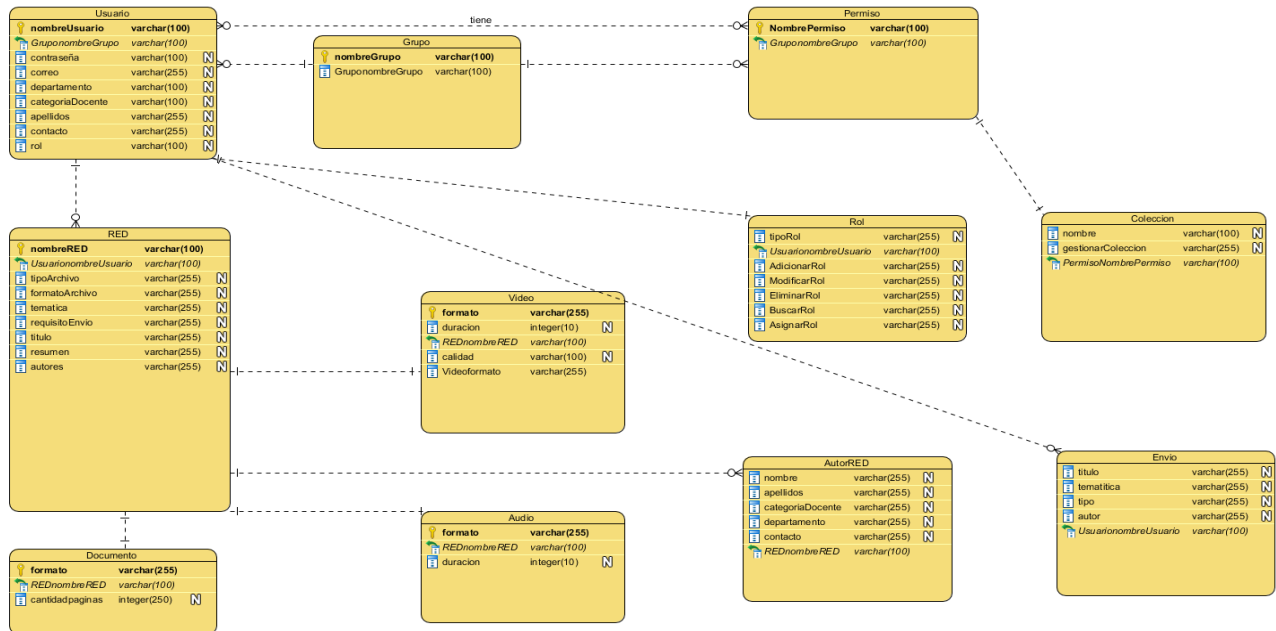


Figura 5. Modelo de datos. Fuente: Elaboración propia.

## 2.6. Conclusiones del capítulo

La revisión de la literatura científica, el estudio de los repositorios de contenidos digitales accesibles nacional e internacionalmente, el análisis del modelo del dominio, así como las entrevistas realizadas a los profesores del departamento de Enseñanza Militar, permitieron identificar los requisitos funcionales del repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa. La solución propuesta tiene un total de 28 requisitos funcionales.

Con el modelado del dominio, la obtención de los Requisitos Funcionales y No Funcionales y el modelado del sistema, se propicia el entendimiento entre el equipo de desarrollo y el cliente en función de lo que el sistema debe realizar y las características que debe poseer. Como resultado del presente capítulo se obtuvieron además los artefactos correspondientes al flujo de trabajo de diseño, el cual sirve como material de referencia para futuras ampliaciones y modificaciones del sistema como guía para comenzar su implementación.

## **CAPÍTULO III: Implementación y Pruebas**

### **3.1. Introducción del capítulo**

En este capítulo se detallan las fases finales de la construcción del sistema, abordando artefactos fundamentales dentro de la implementación, con el objetivo de obtener un producto que cumpla con los requisitos y que su aplicación sea de manera satisfactoria. Se hace necesaria la comprobación para asegurar que satisface todas las expectativas y cumple con las funcionalidades esperadas, con tal objetivo se introducen, un grupo de ideas fundamentales sobre el proceso de pruebas de la propuesta de solución, abordando las diferentes tipologías de las mismas, centradas en su aplicación sobre sistemas *web*.

### **3.2. Implementación del sistema**

La fase de implementación en el desarrollo de un producto de software, es el mecanismo donde se ponen en práctica todas las descripciones y arquitecturas propuestas en las fases de análisis y diseño, es el complemento del trabajo de las fases que lo preceden dentro del proceso de desarrollo de *software*. La implementación ofrece una materialización precisa de los requisitos.

#### **3.2.1. Diagrama de componentes**

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Se realizan con el objetivo de poseer una vista de forma general del sistema a partir de las dependencias e integraciones de los componentes y módulos.

Los componentes del diagrama pueden agruparse en paquetes; permitiéndose entre ellos relaciones (generalización, asociación, agregación y realización). A continuación, se representa el diagrama de componentes asociado al caso de uso: Realizar envío.

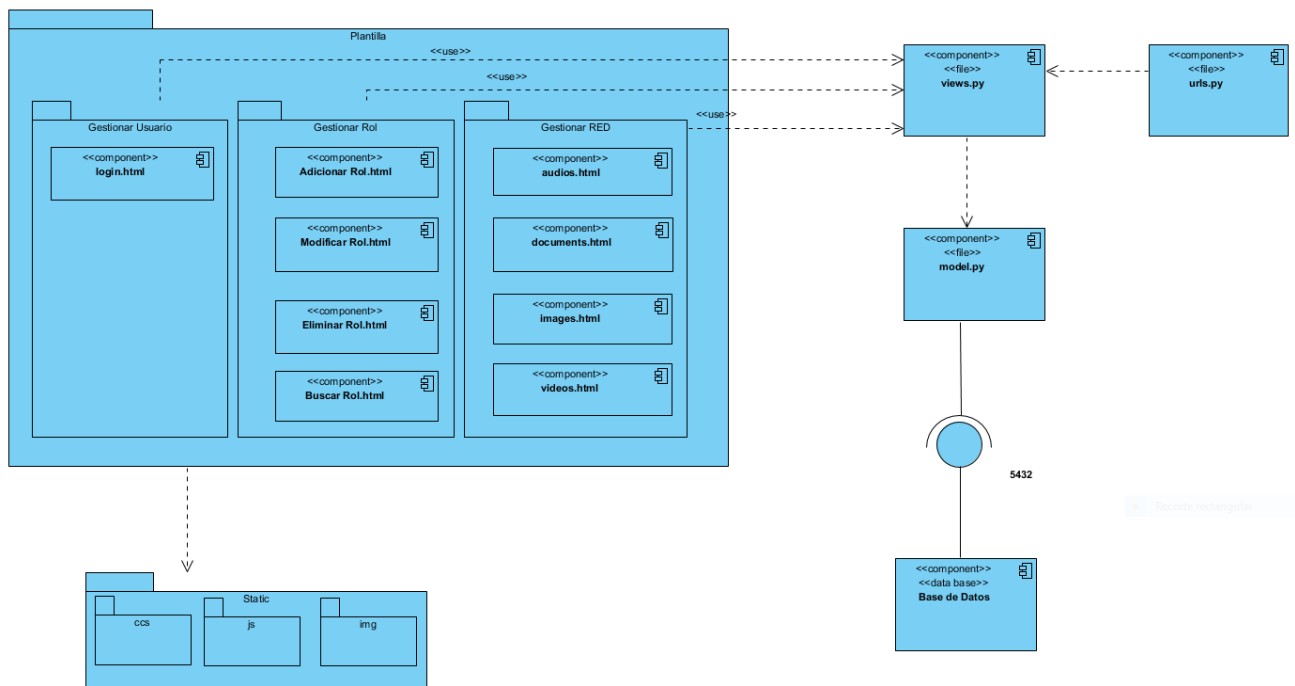


Figura 6. Diagrama de componentes. Fuente: Elaboración propia.

*Bootstrap* está constituido por una serie de archivos *CSS* y *JavaScript* responsables de asignar características específicas a los elementos de la página. Hay un archivo principal llamado *bootstrap.css*, que contiene una definición para todos los estilos utilizados. Básicamente, la estructura del *framework* se compone de dos directorios:

- **css:** contiene los archivos necesarios para la estilización de los elementos y una alternativa al tema original.
- **js:** contiene la parte posterior del archivo *bootstrap.js* (original y minificado), responsable de la ejecución de aplicaciones de estilo que requieren manipulación interactiva.

### 3.2.2. Código fuente

Para obtener una versión funcional de la aplicación se deben implementar los componentes que se han definido, como resultado se obtienen archivos que contienen el código fuente de la aplicación. El código fuente de un *software* es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está escrito su funcionamiento. Estas instrucciones son escritas en un lenguaje de programación que consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos (Lasso, Iván 2008).



### 3.2.3. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

Para facilitar el entendimiento del código y fijar un modelo a seguir, se establecieron estándares de codificación. A continuación, se muestran algunos de estos estándares utilizados en la implementación del sistema.

#### Nombres de clases y métodos

Para la definición de las clases y métodos en el código del sistema informático fueron utilizados los estándares de codificación *CamelCase* y *UpperCamelCase*. Estos son estilos de escritura que se aplica a frases o palabras compuestas.

#### Estructura

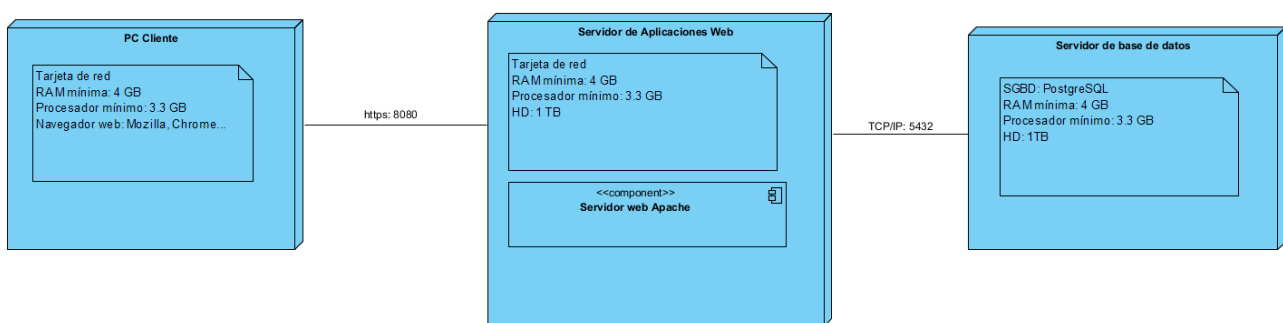
- El código debe usar cuatro espacios en vez de usar el tabulado. Esto minimiza problemas con otras herramientas de desarrollo.
- Las líneas deben tener 80 caracteres o menos, evitando tener más de 120 caracteres.
- Las llaves de apertura en las clases deben ir en la siguiente línea y la llave de cierre debe ir en la siguiente línea después del cuerpo.
- Las llaves de apertura, en los métodos de las estructuras de control debe ir en la misma línea y las llaves de cierre deben de ir después del cuerpo.
- Los paréntesis en las estructuras de control no deben usar espacios antes o después.
- Añadir un solo espacio después de cada limitador de coma.
- Añadir un solo espacio alrededor de los operadores (==, &&, ...).
- Usa llaves para indicar el control de la estructura sin tener en cuenta el número de declaraciones que el grupo pueda contener.
- Definir una clase por fichero.
- Declarar las propiedades de clase antes que los propios métodos de clase.

### 3.3. Diagrama de despliegue

Este Diagrama de despliegue representa las relaciones físicas de los distintos nodos que componen el sistema y el reparto de los componentes sobre estos. Define la topología del sistema, describiendo los elementos de *software* y *hardware* necesarios para la puesta en ejecución del sistema informático. El diagrama de despliegue es un tipo de diagrama UML que representa la arquitectura de tiempo de

ejecución de un sistema y los componentes de software que se ejecutan en esta arquitectura. Representa además los nodos como entornos de ejecución de *software* y *hardware*, y el *middleware* que los conecta. Un nodo es un recurso de ejecución tal como un procesador, un dispositivo o memoria (UML 2.1.2. (s. f.). Recuperado 11 de noviembre de 2022, de <http://www.omg.org/spec/UML/2.1.2/>).

Los nodos se conectan por asociaciones de comunicación como conexiones TCP/IP y HTTP. En este diagrama se representan las condiciones mínimas que debe garantizar la infraestructura tecnológica en la que se va a desplegar el sistema implementado. La figura 14 muestra el diagrama de despliegue del repositorio para la gestión de contenidos digitales de la Preparación para la Defensa en la Universidad de las Ciencias Informáticas.



**Figura 7.** Diagrama de despliegue Fuente: Elaboración propia

### Descripción del diagrama de despliegue:

- **PC Cliente:** su función es acceder e interactuar con el repositorio según sus necesidades. Al estar la aplicación desarrollada sobre la *web*, la máquina cliente necesita de muy pocas prestaciones; teniendo un navegador *web* (*Chrome, Mozilla Firefox, Internet Explorer, Opera,*), una RAM mínima de 4 GB, una tarjeta de red y un procesador mínimo de 3.3 GB podrá acceder al sistema y realizar las operaciones necesarias.
- **Servidor de la aplicación *web*:** Representa el servidor que se comunica con la PC Cliente mediante el protocolo HTTPS y además realiza peticiones al servidor de Bases de Datos mediante el protocolo TCP/IP, es el encargado de la presentación del repositorio, debe estar compuesto de 4 GB de RAM, 1 TB de almacenamiento, un procesador de 3.3 GB o superior, una tarjeta de red, servidor *web* Apache y como marco de trabajo *Bootstrap*.
- **Servidor de Base de Datos:** es el encargado de almacenar la información generada del sistema, para el correcto funcionamiento del repositorio es necesario que posea PostgreSQL, una RAM mínima de 4 GB, un procesador mínimo de 3.3 GB y un disco duro de 1 TB.

### 3.4. Pruebas de software

Las pruebas de *software* tienen como objetivo identificar aquellos defectos que pueda tener el *software* y una vez corregidos elevar el nivel de calidad de los sistemas. Las pruebas son un conjunto de actividades dentro del desarrollo de la aplicación, pudiendo ser implementadas en cualquier momento de dicho proceso del *software* (Barhate, 2015). En este epígrafe se pretende demostrar que el programa cumple con todos los requerimientos necesarios; además de descubrir imperfecciones del mismo para posterior corrección y despliegue.

#### 3.4.1. Estrategia de pruebas

La estrategia de pruebas de *software* proporciona una guía que describe los pasos que deben realizarse como parte de las pruebas. Esta debe incorporar la planificación de la prueba, el diseño de los casos de prueba, la ejecución de las pruebas y la recolección y evaluación de los resultados. Debe ser suficientemente flexible para promover un uso personalizado de la prueba y adecuadamente rígida para alentar la planificación razonable y el seguimiento de la gestión conforme avanza el proyecto (Pressman, 2010).

Para evaluar la calidad del repositorio desarrollado y verificar el cumplimiento de los objetivos trazados, se aplica un conjunto de pruebas. En la siguiente tabla se muestra la estrategia de pruebas diseñada que será aplicada al sistema.

**Tabla 8.** Estrategia de pruebas para el repositorio para la gestión de contenidos digitales de la Preparación para la Defensa en la Universidad de las Ciencias Informáticas.

Nivel de prueba	Tipo	Método	Herramienta
Unidad	Caja blanca	Camino básico	Manual
	Caja Negra	Partición de equivalencia	
Sistema	Seguridad		Owasp-Zap v2.7.0
	Usabilidad		Lista de chequeo
Aceptación	Beta		

##### 3.4.1.1 Pruebas unitarias

Las pruebas unitarias son pruebas de caja blanca, una técnica de prueba de *software* en la que se prueba la estructura interna, el diseño y la codificación del *software* para verificar el flujo de entrada y salida y para mejorar el diseño, la usabilidad y la seguridad. En una prueba de caja blanca, los probadores aparecen en el código, por lo que se denomina prueba de caja limpia, prueba de caja

abierta, prueba de caja transparente, prueba basada en código y prueba de caja de vidrio, se centran en los detalles procedimentales del *software*, por lo que su diseño está fuertemente ligado al código fuente. Este método intenta garantizar que se ejecutan al menos una vez todos los caminos independientes que presente el módulo y que todas las estructuras de datos internas serán usadas. (Pressman, R. S. (2013). Ingeniería de Software un enfoque práctico. Journal of Chemical Information and Modeling (7ma ed., Vol. 53). Mc Graw Hill.)

La prueba de caja blanca implica probar el código del *software* para lo siguiente:

- Agujeros de seguridad internos
- Rutas rotas o mal estructuradas en los procesos de codificación
- Las entradas específicas fluyen a través del código
- Rendimiento esperado
- Funcionalidad de bucle condicional
- Pruebe cada enunciado, objeto y función individualmente

Luego de realizar la prueba de caja blanca a la función *signin* (*request*) se determinó que la complejidad ciclomática era 3 por lo que el método es sencillo y de poco riesgo para el sistema.

Para calcularla, se tiene tres opciones:

- Restar las aristas menos los nodos y sumar 2:

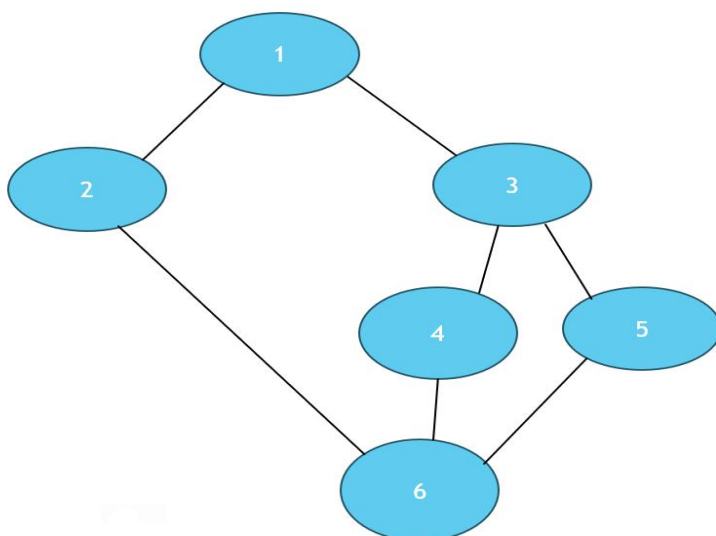
$$o V(G) = \text{Aristas} - \text{Nodos} + 2$$

- Sumar 1 al número de nodos predicados

$$o V(G) = \text{Nodos predicados} + 1$$

- Contar el número de regiones (espacios «encerrados entre nodos y aristas», también se tiene en cuenta el espacio «exterior» a todos los nodos y aristas).

$$o V(G) = \text{Regiones}$$



**Complejidad ciclomática: 3**  
 Por lo cual el método es sencillo y de poco riesgo

**Figura 8. Prueba de caja blanca: flujo de entrada y salida. Fuente: Elaboración propia.**

**Caminos:**

1,3,5,6

1,3,4,6

1,2,6

Camino	Usuario	Contraseña
1,3,5,6	Superusuario	Válida
1,3,4,6	Usuario	Válida
1,2,6	No está Registrado	Inválida

**Tabla 9. Juego de Datos. Fuente: Elaboración propia**

```

27
28 def signin(request):
29     if request.method == 'GET':
30         return render(request, 'files/authentication/login.html', {
31             'form': AuthenticationForm
32         })
33
34     else:
35         user = authenticate(request, username=request.POST['username'], password=request.POST['password'])
36
37         if user is None:
38             return render(request, 'files/authentication/login.html', {
39                 'form': AuthenticationForm,
40                 'error': 'Credenciales invalidas'
41             })
42         else:
43             login(request, user)
44             return redirect('/inicio/')
    
```

**Figura 9. Prueba de caja blanca a la función signin (request). Fuente: Elaboración propia.**

### 3.4.2. Nivel de prueba de sistema

Este nivel pretende demostrar que los componentes son compatibles, que interactúan correctamente y que transfieren los datos correctos en el momento adecuado a través de sus interfaces. Estas deben enfocarse en poner a prueba las interacciones entre los componentes y los objetos que constituyen el sistema. También se prueban componentes o sistemas reutilizables para acreditar que al integrarse nuevos componentes funcionen como se esperaba (Sommerville, 2011).

Existen dos tipos de pruebas en este nivel las funcionales y las no funcionales. Las funcionales se centran en comprobar que el sistema desarrollado funciona acorde a las especificaciones funcionales y requisitos de la aplicación. Ayudando a detectar posibles defectos derivados de errores en la fase de programación. Estas le prestan atención al comportamiento del sistema y a los datos de entrada y salida del entorno de pruebas (Escobar-Sánchez & Fuertes-Díaz, 2015). Las pruebas no funcionales están orientadas a validar los requerimientos no funcionales del sistema, es decir, la forma en que funciona el sistema y no por medio de comportamientos específicos. Su ejecución descubre los riesgos que el sistema posee, permitiendo comprobar si cumple con las expectativas del cliente. Entre sus beneficios se encuentran: mejora de la experiencia del usuario, determinación de cuellos de botella como la configuración de BD y permite conocer el rendimiento real de la aplicación en comparación con las expectativas esperadas (Morales, 2018).

### 3.4.3. Pruebas funcionales

Para la ejecución de las pruebas funcionales fue necesario la utilización del método de caja negra el cual se concentra en analizar los requisitos funcionales del *software*, llevándose a cabo sobre la interfaz de usuario, sin prestar atención al comportamiento interno y la estructura del programa. Su objetivo es detectar errores de interfaz, de estructura de datos, de inicialización o terminación. Para hacer dichas pruebas se confeccionan 11 Casos de prueba (si desea analizar dichos casos de pruebas remitirse al Expediente de proyecto y seleccionar la carpeta que lleva por nombre Casos de prueba). A continuación, se representa el caso de prueba correspondiente a la funcionalidad Autenticar usuario.

Tabla 10: Descripción de variables del Caso de Prueba del requisito Autenticar usuario. Fuente: Elaboración propia.

No.	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Usuario	Campo de texto	No	Se escribe el nombre del usuario que se quiere registrar en el sistema. El nombre de usuario solo debe contener letras minúsculas.
2	Contraseña	Campo de texto	No	Se escribe la contraseña del usuario que se quiere autenticar en el sistema.  Solo se aceptarán letras (minúsculas y mayúsculas), símbolos especiales y números. La longitud de la contraseña debe ser de 8 o más caracteres.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar en el repositorio el usuario y la contraseña.	Registrar correctamente en el repositorio el usuario y la contraseña.	Se registra correctamente el usuario y la contraseña.	Selecciona la opción aceptar.
EC 1.2 Los datos introducidos son incorrectos.	Se introducen datos incorrectos.	Identifica que los datos introducidos son incorrectos e impide el acceso al repositorio.	Selecciona la opción aceptar.
EC 1.3 Existen campos obligatorios vacíos.	Se dejan vacíos los campos usuario o contraseña.	Identifica los campos vacíos e impide el acceso al repositorio.	Selecciona la opción aceptar.
EC 1.4 La operación es cancelada.	Se cancela la operación de autenticar.	Cierra la interfaz de autenticación.	Selecciona la opción cancelar.

### 3.4.3.1 Pruebas de Caja Negra

Las pruebas de caja negra, conocidas también como *black box testing*, pueden definirse como una técnica donde se busca la verificación de las funcionalidades del *software* o aplicación analizada, sin tomar como referente la estructura del código interno, las rutas de tipo internas ni la información referente a la implementación. Esto quiere decir que la prueba se lleva a cabo con desconocimiento del funcionamiento del sistema interno, debido a que se enfoca en la entrada y salida de un *software*, tomando como base sus especificaciones y requisitos. De manera que se puede asegurar que el objetivo de las pruebas de caja negra está relacionado con la validación de los recursos funcionales del *software* o aplicación que se busca examinar. (KeepCoding, 2022)

Mientras más amplio sea el espectro de elementos de entrada para realizar la prueba, las probabilidades de encontrar problemas en el software aumentan y, por lo tanto, será más confiable la calidad del *software*.

Este tipo de pruebas permite encontrar:

- Funciones incorrectas o ausentes
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.

- Errores de rendimiento.
- Errores de inicialización y terminación.

Para preparar los casos de prueba es necesario un número de datos que ayuden a la ejecución de estos casos y que permitan que el sistema se ejecute en todas sus variantes. Estos datos pueden ser válidos o no para el *software* y son seleccionados atendiendo a las especificidades de la funcionalidad a probar. Para realiza estas pruebas existen varias técnicas:

- Partición de equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*. Esta técnica es muy efectiva puesto que permite examinar los valores válidos e inválidos de las entradas existentes en el producto, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la realización de muchos casos antes de detectar el error genérico. La partición de equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo de este modo el número de pruebas a realizar.

Se realizaron 6 casos de prueba a continuación se muestra un ejemplo de los mismos:

**Tabla 11.** Descripción de casos de prueba. **Fuente:** Elaboración propia.

Escenario	Nombre	Usuario	Rol	Descripción	Respuesta del sistema	Flujo Central
<b>EC 1.1</b> <b>Adicionar usuario correctamente aplicando el adicionar</b>	V(Karen)	V(kgzayas)	V (jefe de Departamento)	V (Se adiciona un usuario se presiona el botón aceptar)	Se muestra el mensaje " <i>El usuario ha sido adicionado satisfactoriamente</i> ". Se cierra el formulario.	1. Se selecciona la opción Componente usuario. 2. Seleccione la opción Usuario del menú de opciones. 3. Selecciona la opción Adicionar. 4. Se llenan los campos: Se presiona la opción Adicionar. Se validan los campos, Se adiciona usuario al sistema.
<b>EC 1.2</b> <b>Adicionar usuario correctamente</b>	V(Joanny)	V(jfdiaz)	V(Administrador)	V (Se aplica los cambios al usuario se presiona el botón Aplicar)	Se limpian los campos para adicionar un nuevo usuario.	1. Se selecciona la opción Componente usuario. 2. Seleccione la opción Usuario del menú de opciones.



aplicando el aplicar						<p>3. Selecciona la opción Adicionar.</p> <p>4. Se llenan los campos: Se presiona la opción Aplicar. Se validan los campos, <i>Se aplican los cambios del usuario al sistema.</i></p>
<p><b>EC1.3</b></p> <p><b>Adicionar usuario incorrectamente.</b></p>	V(Joan)	V(jghernandez)	V (jefe de Departamento)	Se muestra un mensaje según el error cometido.	Si ya existe un usuario con el número de expediente introducido se muestra el mensaje <i>"El número de expediente pertenece a otro usuario"</i> .	<p>1. Se selecciona la opción Componente usuario.</p> <p>2. Seleccione la opción Usuario del menú de opciones.</p> <p>3. Selecciona la opción Adicionar.</p> <p>4. Se llenan los campos: Se presiona la opción Adicionar. Se validan los campos.</p>
	V(Karen5)	V(kgallende)	V(Profesor)	Se muestra un mensaje según el error cometido.	Si el nombre contiene números no es válido se muestra el mensaje <i>"El nombre contiene número no es válido"</i> .	
		V(kgalconero)	V(Profesor)	Se muestra un mensaje según el error cometido.	Existen datos vacíos se muestra el mensaje <i>"Hay campos vacíos rellénelos"</i> .	
<p><b>EC1.4</b></p> <p><b>Adicionar usuario correctamente aplicando el Cancelar</b></p>	NA	NA	NA	NA (Se presiona el botón Cancelar)	Se cancela la persona y se cierra la ventana.	<p>1. Se selecciona la opción Componente usuario.</p> <p>2. Seleccione la opción Usuario del menú de opciones.</p> <p>3. Selecciona la opción Adicionar.</p>

						4. Se llenan los campos: Se presiona la opción Cancelar. Se cierra el formulario.
--	--	--	--	--	--	--

Las pruebas se efectuaron en tres iteraciones, en la primera se detectaron un total de 25 no conformidades; clasificadas como 13 de ortografía, cinco de funcionalidad, uno de interfaz y seis de validación. Antes de llevar a cabo la segunda iteración se corrigieron los errores encontrados en la primera; obteniéndose tres no conformidades ortográficas y una de interfaz, sumando cuatro. Tras resolver las imperfecciones anteriormente mencionadas se lleva a cabo la tercera iteración obteniéndose resultados satisfactorios (cero no conformidades). A continuación, se muestra los resultados de la aplicación de la prueba, teniendo en cuenta los tipos de no conformidades identificadas (ortográficas, validación, funcional e interfaz).

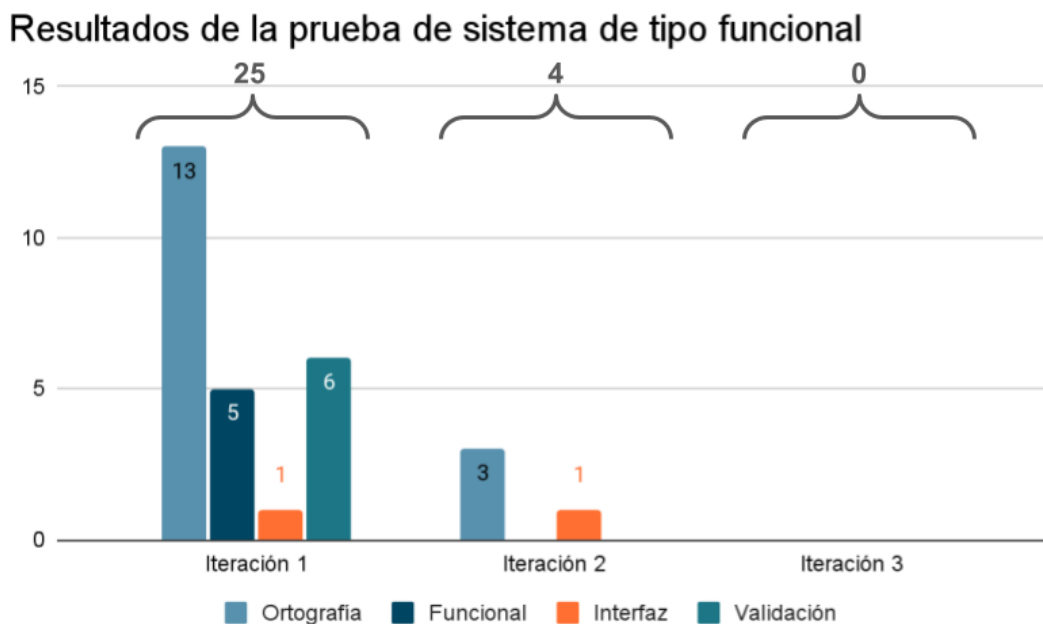


Figura 10. Gráfico de barras de los resultados de la prueba de sistema funcional.

**Fuente:** (Elaboración propia).

### 3.3.4 Pruebas de Seguridad

Las Pruebas de Seguridad son un tipo de prueba de software que revela vulnerabilidades, amenazas, riesgos en una aplicación de software y previene ataques maliciosos de intrusos. El propósito de las Pruebas de Seguridad es identificar todas las lagunas y debilidades potenciales en el sistema de

*software* que podrían resultar en la pérdida de información, ingresos, reputación de los empleados o personas ajenas a la Organización.

Para el desarrollo de las mismas se utilizó la herramienta *Netsparker* que encuentra e informa de las vulnerabilidades de una aplicación *web*, como la inyección de SQL y de *cross-site scripting* (XSS) en todos los tipos de aplicaciones *web*, independientemente de la plataforma y de la tecnología con la que están contruidos.

La tecnología única y absolutamente precisa *Proof-Based Scanning™* de *Netsparker* no sólo informa de las vulnerabilidades, también produce una prueba de concepto para confirmar que no sean falsos positivos. Lo que le libera de tener que comprobar las vulnerabilidades identificadas. («Netsparker Web Application Security Scanner \* QMA MSS», 2019)

Se realizó un análisis del sistema con la herramienta, los resultados del mismo están reflejados en el informe emitido por la misma herramienta.

### 3.4.5 Nivel de prueba aceptación

Están orientadas a determinar si el *software* cumple con las necesidades y requerimientos acordados, demostrando que está listo para ser desplegado. Estas pruebas se desarrollan para establecer el nivel de confianza del sistema, que está dado por su grado de adherencia a las necesidades, requerimientos y procesos de negocio solicitados por el usuario (Basit et al., 2018).

Las pruebas de aceptación a efectuar son de tipo Beta, esta incluye entregar el sistema a algunos de los clientes finales en un entorno real, con el objetivo de que, al interactuar con el software, se encuentren y reporten problemas o no conformidades presentes en el sistema que no fueron anticipadas por los desarrolladores. Después de esta retroalimentación, el sistema se modifica y libera, ya sea para más pruebas beta o para su despliegue (Sommerville, 2011).

Es mostrada la aplicación al cliente, con el objetivo de que sea evaluada; tras dicho análisis se obtuvieron siete no conformidades agrupadas en dos funcionales y cinco de interfaz culminándose así la primera iteración. Se corrigen los inconvenientes y se procede a la segunda iteración, dando como resultado una no conformidad de interfaz. Después de resolver las no conformidades es ejecutada la tercera iteración, en la que se mostró que la herramienta cumple con todos los requisitos del sistema. Culminadas las pruebas se evidencia que la aplicación es apta para el despliegue, constando la conformidad del cliente en la Carta de Aceptación de la aplicación. A continuación, se muestra el gráfico representativo con los resultados:

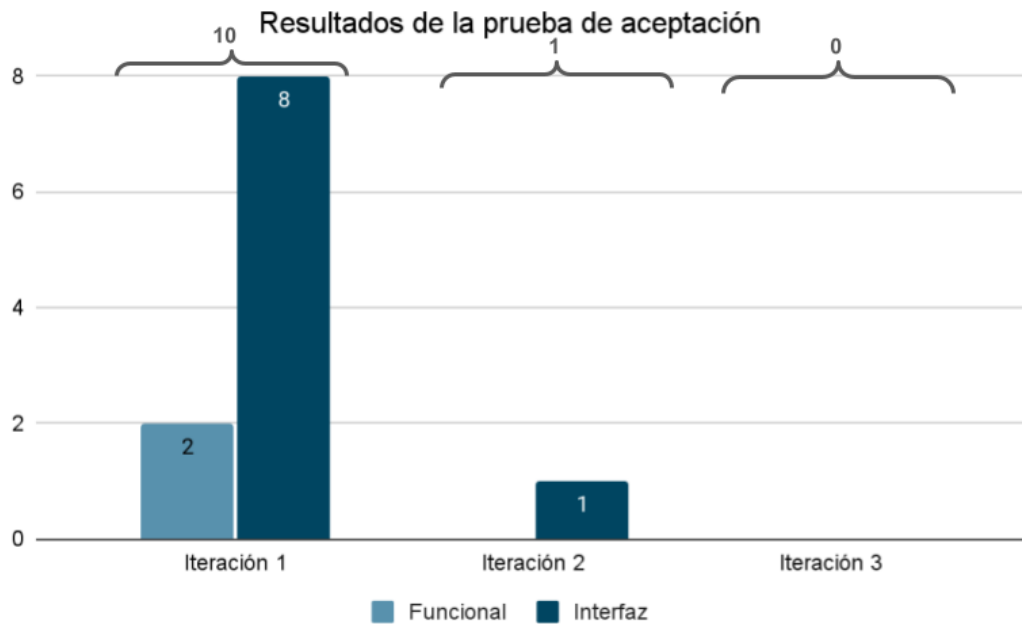


Figura 11. Gráfico de barras de los resultados de la prueba aceptación. **Fuente:** (Elaboración propia).

### 3.4.5.1 Validación de las variables

Luego del análisis, diseño e implementación de la solución se puede demostrar que:

Características	Antes de la solución	Después de la solución
Contenidos dispersos en diferentes sitios web	Cuba no contaba con un sistema donde se agrupará todo el contenido de la Preparación para la Defensa en la Educación Superior	Posee un sistema para la gestión de contenido de la Preparación para la Defensa en la Educación Superior
Intercambio entre profesores y alumnos a nivel nacional	Solo se hacía a nivel de Universidad	Ahora se hace a través del sistema
Posibilidad de dar noticias a nivel nacional	La forma en la que se divulgaban las noticias era de forma manual	Ahora se hace a nivel de país
Utilización de la conectividad para promoción y consumo de contenido	No se utilizaba	En Cuba existen más de 5 millones de líneas celulares, más de 2 millones de conexiones diarias, y todo lo expuesto está integrado en un solo sistema, dominio, permitiendo su rápida indexación, accesibilidad, tráfico

		y posicionamiento en motores de búsqueda.
--	--	---

Tabla 12. Validación de variables **Fuente:** Elaboración propia.

Por tanto, se puede afirmar que con el desarrollo del repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa en la Universidad de las Ciencias Informáticas contribuirá a la accesibilidad de los contenidos de la disciplina Preparación para la Defensa en la Educación Superior.

### 3.5. Conclusiones del capítulo

Se describen los elementos del modelo de diseño que se implementan en términos de componentes. El método de prueba seleccionado posibilitó detectar los errores existentes al concluir la implementación del repositorio, los cuales fueron revisados y corregidos para garantizar la calidad del mismo.

## CONCLUSIONES

- El estudio de repositorios de recursos educativos digitales a nivel nacional e internacional permitió definir las tendencias actuales en el desarrollo de este tipo de aplicaciones informáticas, demostrando que en el centro no existe una aplicación que cumpla con todas estas características.
- El estudio de tecnologías para la programación de este tipo de aplicaciones permitió escoger las herramientas más adecuadas para el desarrollo del repositorio.
- El modelado de los requisitos y la generación de los artefactos de la aplicación permitió el diseño de la solución del problema.
- Las pruebas realizadas permitieron identificar problemas en el funcionamiento del *software*, para proceder a su posterior eliminación.
- Se desarrolló un repositorio que brinda a los usuarios de la comunidad universitaria la posibilidad de gestionar, crear, modificar, visualizar y almacenar recursos educativos digitales de la disciplina Preparación para la Defensa. El repositorio permite que se publiquen documentos e información de las asignaturas vinculadas a la Preparación para la Defensa, el cual aporta una vital importancia a la visibilidad de los contenidos para sus estudiantes en la universidad y de personas jurídicas, siendo una vía por la cual los profesores y estudiantes pueden intercambiar información y consultar la misma de forma rápida y segura, contribuyendo a la gestión de recursos educativos digitales de la disciplina Preparación para la Defensa en la Universidad de las Ciencias Informáticas.

## RECOMENDACIONES

A partir de la investigación realizada se recomienda:

- Capacitar al personal en el trabajo con repositorios.
- Desplegar la solución propuesta en un ambiente estable de servicio.

## REFERENCIAS BIBLIOGRÁFICAS

- Akhunzada, A., Gani, A., Hussain, S., Khan, A. A., & Ashrafullah. (2015, 10-11 Nov. 2015). Towards experiencing the pair programming as a practice of the Rational Unified Process (RUP). 2015 SAI Intelligent Systems Conference (IntelliSys),
- Aleixandre-Benavent, R., Sapena, A. F., & Peset, F. (2021). Compartir los recursos útiles para la investigación: datos abiertos (open data). *Educación Médica*, 22, 208-215. <https://www.sciencedirect.com/science/article/pii/S1575181319301287>
- Álvarez, E. B., & López, Y. V. (2008). Un repositorio institucional en la Universidad de la Habana: ¿Necesidad o Capricho? *Biblios: Revista electrónica de bibliotecología, archivología y museología*(31), 8. <https://dialnet.unirioja.es/descarga/articulo/2663193.pdf>
- Amén Mora, P., Rodríguez Rodríguez, A., García Macías, V., & Rincón Zambrano, R. (2023). Estrategias didácticas en la era digital: diseño, implementación y evaluación de Recursos Educativos Digitales *Serie Científica de la Universidad de las Ciencias Informáticas*, 16(9), 78-94. <https://publicaciones.uci.cu/index.php/serie/article/view/1238>
- Aparicio, M., Oliveira, T., Bacao, F., & Painho, M. (2019). Gamification: A key determinant of massive open online course (MOOC) success. *Information & Management*, 56(1), 39-54. <https://www.sciencedirect.com/science/article/pii/S0378720616304062>
- Ardi, S., & Shahmehri, N. (2008, 4-7 March 2008). Integrating a Security Plug-in with the OpenUP/Basic Development Process. 2008 Third International Conference on Availability, Reliability and Security,
- Barhate, S. S. (2015, 28-30 May 2015). Effective test strategy for testing automotive software. 2015 International Conference on Industrial Instrumentation and Control (ICIC),
- Battaglia, N., Neil, C., De Vincenzi, M., & Martinez, R. (2016). UAICase: integración de un entorno académico con una herramienta CASE en una plataforma virtual colaborativa. XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016).
- Beltrán, R. D., & Urgellés, W. C. (2012). Aspectos sobre la seguridad nacional cubana en la asignatura Preparación para la Defensa para estudiantes de la salud. *Revista Información Científica*, 74(2). <http://revinfcientifica.sld.cu/index.php/ric/article/view/743>
- Borg, A., Sandahl, K., & Patel, M. (2007, 15-19 Oct. 2007). Extending the OpenUP/Basic Requirements Discipline to Specify Capacity Requirements. 15th IEEE International Requirements Engineering Conference (RE 2007),
- Cadavid, A. N., Martínez, J. D. F., & Vélez, J. M. J. P. (2013). Revisión de metodologías ágiles para el desarrollo de software. 11(2), 30-39.
- Castillo, A. A. (2017). *Curso de Programación Web: JavaScript, Ajax y jQuery*. IT Campus Academy.



- Coban, M., Bolat, Y. I., & Goksu, I. (2022). The potential of immersive virtual reality to enhance learning: A meta-analysis. *Educational Research Review*, 36, 100452. <https://www.sciencedirect.com/science/article/pii/S1747938X22000215>
- Cowburn, P. (2018). Manual de PHP.
- Delís, Y. M. R., Jiménez, R. L. C., & Arrufat, M. J. G. (2018). Iniciativas para la adopción y uso de recursos educativos abiertos en Instituciones de Educación Superior. *Revista Cubana de Educación Médica Superior*, 32(4), 273-285. <https://www.medigraphic.com/cgi-bin/new/resumen.cgi?IDARTICULO=86943>
- Duarte, R. H. G. y. O. M. (2016). *Módulo para la gestión de información del proceso diagnóstico sociopolítico de la Unión de Jóvenes Comunistas en la UCI* [Universidad de las Ciencias Informáticas]. jun-2016. <https://repositorio.uci.cu/jspui/handle/123456789/7741>
- Escobar-Sánchez, M. E., & Fuertes-Díaz, W. M. J. F. d. I. (2015). Modelo formal de pruebas funcionales de software para alcanzar el Nivel de Madurez Integrado 2. *24(39)*, 31-42.
- Farinango, C. D., Benavides, J. S., & Lopez, D. M. (2015). OpenUP/MMU-ISO 9241-210. Process for the Human Centered Development of Software Solutions. *IEEE Latin America Transactions*, 13(11), 3668-3675. <https://doi.org/10.1109/TLA.2015.7387947>
- Fernández, A. M., Reyes, M. J., & López, M. I. V. (2022). Tecnologías de la información y comunicación (TIC) en formación y docencia. *FMC-Formación Médica Continuada en Atención Primaria*, 29(3), 28-38. <https://www.sciencedirect.com/science/article/pii/S1134207222000810>
- Gutiérrez, J. J. (2014). ¿ Qué es un framework web. Available in: [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf) Accessed May, 12.
- Hennessy, S., D'Angelo, S., McIntyre, N., Koomar, S., Kreimeia, A., Cao, L., Brugha, M., & Zubairi, A. (2022). Technology use for teacher professional development in low-and middle-income countries: A systematic review. *Computers and Education Open*, 3, 100080. <https://www.sciencedirect.com/science/article/pii/S2666557322000088>
- Hew, K. F., & Cheung, W. S. (2014). Students' and instructors' use of massive open online courses (MOOCs): Motivations and challenges. *Educational Research Review*, 12, 45-58. <https://www.sciencedirect.com/science/article/pii/S1747938X14000128>
- Hui, Y., Yan, Y., Quanyu, W., & Zhiwen, C. (2015, 15-17 June 2015). Compare Essential Unified Process (EssUP) with Rational Unified Process (RUP). 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA),
- Jianwei, L., & Xunan, W. (2010, 3-5 Oct. 2010). Research and practice of agile Unified Process. 2010 2nd International Conference on Software Technology and Engineering,
- Koç, G., & Aydos, M. (2017, 5-8 Oct. 2017). Trustworthy scrum: Development of secure software with scrum. 2017 International Conference on Computer Science and Engineering (UBMK),

- Larman, C. (2003). *UML y Patrones*. Pearson Educación ^ eMadrid Madrid.
- Leyva, A., Gaubeca, F., & Alfonso, R. (2020). *Textos Especializados de la asignatura Defensa Nacional para los estudiantes de Ingeniería en Bioinformática* Universidad de las Ciencias Informáticas].
- Lockhart, T. (1998). Manual del usuario de PostgreSQL. *Australia*. Obtenido de <http://palomo.usach.cl/Docs/postgres/Postgres-Tutorial.pdf>.
- Margaryan, A., Bianco, M., & Littlejohn, A. (2015). Instructional quality of massive open online courses (MOOCs). *Computers & Education*, 80, 77-83. <https://www.sciencedirect.com/science/article/pii/S036013151400178X>
- Morales, V. M. S. (2018). *Conoce qué son las pruebas no funcionales de software*. Retrieved 26/03/2019 from [Conoce qué son las pruebas no funcionales de software](http://www.conocequesean.com)
- Nilashi, M., Minaei-Bidgoli, B., Alghamdi, A., Alrizq, M., Alghamdi, O., Nayer, F. K., Aljehane, N. O., Khosravi, A., & Mohd, S. (2022). Knowledge discovery for course choice decision in Massive Open Online Courses using machine learning approaches. *Expert Systems with Applications*, 199, 117092. <https://www.sciencedirect.com/science/article/pii/S0957417422004985>
- Pantaleo, G., & Rinaudo, L. (2015). *Ingeniería de software*. Alfaomega Grupo Editor.
- Pressman, R. S. (2010). *Ingeniería del software. Un enfoque práctico*. 7ma, 767.
- Quesada, R., Ceballos López, C., & Miranda Crespo, E. (2014). *Texto para la disciplina preparación para la defensa en seguridad nacional y defensa nacional para los estudiantes de la educación superior*.
- Raya, R. (2014). ¿ Qué son las metodologías ágiles. *Recuperado de: http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles*.
- Salabarría Cárdenas, F. L. (2022). *Repositorio para la gestión de recursos educativos* Universidad de las Ciencias Informáticas. FACULTAD 6. Facultad de Ciencias y ...]. <https://repositorio.uci.cu/jspui/handle/123456789/10502>
- Sommerville, I. (2011). *Software engineering*. Boston: Pearson.
- Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017, 5-6 May 2017). SCRUM model for agile methodology. 2017 International Conference on Computing, Communication and Automation (ICCCA),
- Zapata, C. M., & Arango Isaza, J. F. (2004). Alineación entre metas organizacionales y elicitación de requisitos del software. *Dyna*, 71(143).
- Basit, M. A., Baldwin, K. L., Kannan, V., Flahaven, E. L., Parks, C. J., Ott, J. M., & Willett, D. L. J. J. m. i. (2018). Agile Acceptance Test–Driven Development of Clinical Decision Support Advisories: Feasibility of Using Open Source Software. 6(2).  
Sommerville, I. (2011). *Software engineering*. Boston: Pearson.

**ANEXOS**

**Anexo 1.** Requisitos funcionales agrupados en Casos de uso del sistema.

**Tabla 2.** Requisitos funcionales agrupados en Casos de uso del sistema.

Caso de uso	Requisitos funcionales
Autenticar usuario	RF 1. Autenticar usuario
Gestionar usuario	RF 2. Adicionar usuario
	RF 3. Modificar usuario
	RF 4. Eliminar usuario
	RF 5. Buscar usuario
Administrar rol	RF 6. Adicionar rol
	RF 7. Modificar rol
	RF 8. Eliminar rol
	RF 9. Buscar rol
	RF 10. Asignar rol a usuario
Adicionar RED	RF 11. Adicionar RED
Modificar RED	RF 12. Modificar RED
Administrar RED	RF 13. Eliminar RED
	RF 14. Buscar RED
Publicar RED	RF 15. Publicar RED
Añadir autor	RF 16. Añadir autor del RED
Ver mi lista de envíos	RF 17. Ver mi lista de envío
Listar RED	RF 18. Listar RED por autor
	RF 19. Listar RED por estado
	RF 20. Listar RED por tipo
	RF 21. Listar RED por temática
	RF 22. Listar RED por colección
Gestionar colección	RF 23. Adicionar colección
	RF 24. Modificar colección
	RF 25. Eliminar colección
	RF 26. Buscar colección
Descargar RED	RF 27. Descargar RED
Evaluar RED	RF 28. Evaluar RED

## Anexo 2. Captura de Pantalla de SSRN



Product & Services   Subscribe   Submit a paper   Browse   Rankings   Blog ↗   Contact



Create account

Sign in

### Research Networks:

Accounting [ARN](#)

Agricultural Science [AgnSciRN](#)

Anthropology &  
Archaeology [AARN](#)

Architecture [ArchRN](#)

Biochemistry [BiochemRN](#)

Biology [BioRN](#)

Business History &  
Nonprofit  
Organizations [BHNP](#)

Chemistry [ChemRN](#)

Classics [CRN](#)



Feedback

### Extend the Exposure, Impact and Reach of Proceedings

SSRN helps conferences & meetings share the early-stage, high quality research presented at an event with a relevant global audience. Proceedings can be accessible online for participants before, during

### Anexo 3. Prototipos de interfaz de usuario

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

Bienvenido a

**Gestionar u**

Listado de

**Rol**

1. Administrador  
[Editar](#) [Eliminar](#)

2. Profesor  
[Editar](#) [Eliminar](#)

3. Estudiante  
[Editar](#) [Eliminar](#)

**Añadir rol** X

**Rol\*:**

**Listado de permisos:**

Enviar RED

Publicar RED

Descargar RED

Evaluar RED

Listar RED

Modificar RED

Gestionar Colección

\*: Este campo es obligatorio

ez Zayas  
esor

**Gestionar usuario**

Si

No

No

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

Bienvenido al Re

**Gestionar usuar**

Listado de usu

**Nombre**

1. Yanety Milagros C  
[Editar](#) [Eliminar](#)

2. Armando Leyva P  
[Editar](#) [Eliminar](#)

3. Omar Mar Cornejo  
[Editar](#) [Eliminar](#)

**Añadir usuario** X

**Nombre\*:**

**Apellidos:**

**Categoría docente:**

**Departamento:**

**Rol:**

**Contacto\*:**

\*: Este campo es obligatorio

as


**acto**

estudiantes.uci.cu


cu

omatimara@uci.cu


Repositorio para la gestión de contenidos digitales de Preparación para la Defensa



Repositorio para la gestión de contenidos digitales de Preparación para la Defensa



UCI Universidad de las Ciencias Informáticas



Bienvenido al RepoPPDV1

Yanety Milagros González Zayas  
Autenticado como: Profesor

**Mi perfil**  
**Salir**


**Gestionar usuarios**   **Gestionar roles**

Listado de roles creados en el sistema


**Buscar**   **Añadir rol**

Rol	Enviar RED	Evaluar RED	Descargar RED	Modificar RED	Publicar RED	Gestionar usuario
1. Administrador <a href="#">Editar</a> <a href="#">Eliminar</a>	Sí	Sí	Sí	Sí	Sí	Sí
2. Profesor <a href="#">Editar</a> <a href="#">Eliminar</a>	Si	Sí	Si	No	No	No
3. Estudiante <a href="#">Editar</a> <a href="#">Eliminar</a>	No	Si	Sí	No	No	No


Repositorio para la gestión de contenidos digitales de Preparación para la Defensa



Repositorio para la gestión de contenidos digitales de Preparación para la Defensa



UCI Universidad de las Ciencias Informáticas



Bienvenido al RepoPPDV1

Yanety Milagros González Zayas  
Autenticado como: Profesor

**Mi perfil**  
**Salir**

**Gestionar usuarios**   **Gestionar roles**

Listado de usuarios registrados en el sistema

**Buscar**   **Añadir usuario**

Nombre	Categoría Docente	Departamento	Rol	Contacto
1. Yanety Milagros González Zayas <a href="#">Editar</a> <a href="#">Eliminar</a>	-	-	Estudiante	milagros@estudiantes.uci.cu
2. Armando Leyva Pérez <a href="#">Editar</a> <a href="#">Eliminar</a>	Auxiliar	PPD	Jefe de Departamento	leyva@uci.cu
3. Omar Mar Cornelio <a href="#">Editar</a> <a href="#">Eliminar</a>	Titular	Matemática	Profesor	omarmara@uci.cu

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

**Añadir autor** X

**Nombre\*:**

**Apellidos:**

**Categoría docente:**


**Departamento:**

**Contacto\*:**


\*: Este campo es obligatorio

Nombre	Categoría Docente	Departamento	Contacto	Añadir autor
1. Yanety Milagros González Zayas	Instructor	PPD	milagros@uci.cu	<input type="button" value="Añadir"/>
<a href="#">Editar</a> <a href="#">Eliminar</a>				
2. Armando Leyva Pérez	Auxiliar	PPD	leyva@uci.cu	<input type="button" value="Añadir"/>
<a href="#">Editar</a> <a href="#">Eliminar</a>				


Repositorio para la gestión de contenidos digitales de Preparación para la Defensa



Repositorio para la gestión de contenidos digitales de Preparación para la Defensa




Universidad de las Ciencias Informáticas





**Bienvenido al RepoPPDV1** Yanety Milagros González Zayas


Esta página es la interfaz principal del repositorio asociado a tu usuario. Aquí aparecerán listados tus futuros envíos. Autenticado como: Profesor


**Colecciones**


  
Libros

  
Conferencias

  
Tesis

  
CP

  
Discursos

  
Videos

**Sobre el repositorio**

[Políticas de publicación](#)

[Manual de publicación](#)

[Evaluación de calidad](#)

[Preguntas frecuentes](#)

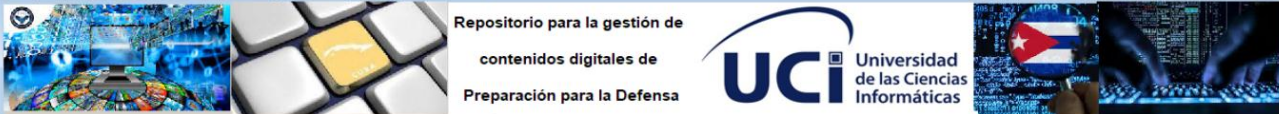
**Mis envíos**

Título	Temática	Tipo	Ver
Conferencia 1	Seguridad	Multimedia	<input type="button" value="Descargar"/>
CP 1	Seguridad	Documento	<input type="button" value="Descargar"/>

**Descubrir por**

Autor	Temática	Tipo
Prez Pupo (10)	Seguridad (25)	APK (21)
González (12)	Defensa (10)	Documento (101)

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa



Bienvenido al RepoPPDV1

Yanety Milagros González Zayas  
Autenticado como: Profesor

**Mi perfil**  
**Salir**

**Realizar Envío**


- 1. Inicio
- 2. Cargar el envío
- 3. Introducir metadatos
- 4. Confirmación

**Buscar archivo en la PC**

Para cargar el archivo puede arrastrarlo a este cuadro de dialogo o buscar el archivo en la PC.  
Para subir el archivo presione: [Subir archivo](#).

**Guardar y continuar** **Cancelar**

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa



Bienvenido al RepoPPDV1

Yanety Milagros González Zayas  
Autenticado como: Profesor

**Mi perfil**  
**Salir**

**Realizar envío**

- 1. Inicio
- 2. Cargar el envío
- 3. Introducir metadatos
- 4. Confirmación

**Título:**  **Título corto:**

**Resumen:**

Autores:	Nombre	Categoría Docente	Departamento	Contacto	<b>Añadir autor</b>
	1. Yanety Milagros González Zayas	Instructor	PPD	milagros@uci.cu	
	<a href="#">Editar</a> <a href="#">Eliminar</a>				
	2. Armando Leyva Pérez	Auxiliar	PPD	leyva@uci.cu	
	<a href="#">Editar</a> <a href="#">Eliminar</a>				

**Guardar y continuar** **Cancelar**



Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

**UCI** Universidad de las Ciencias Informáticas

Bienvenido al RepoPPDV1

Yanety Milagros González Zayas  
Autenticado como: Profesor

**Mi perfil**  
**Salir**

**Realizar Envío**

- 1. Inicio
- 2. Cargar el envío
- 3. Introducir metadatos
- 4. Confirmación

¿Está seguro que desea realizar el envío? Puede revisar y/o modificar los datos navegando por los pasos anteriores.  
Una vez enviado, podrá modificarlo y/o eliminarlo, solo si el estado es ENVIADO

**Enviar**

**Cancelar**

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

Repositorio para la gestión de contenidos digitales de Preparación para la Defensa

**UCI** Universidad de las Ciencias Informáticas

Bienvenido al RepoPPDV1

Yanety Milagros González Zayas  
Autenticado como: Profesor

**Mi perfil**  
**Salir**

**Realizar Envío**

- 1. Inicio
- 2. Cargar el envío
- 3. Introducir metadatos
- 4. Confirmación

Tipo del envío:  Temática:

**Requisitos del envío:**

- El RED fue creado y utilizado con fines pedagógicos.
- El RED solo contiene materiales digitales (ilustración, fotografía, diagrama, audiovisual) libres de restricciones sobre su uso; son de dominio público; posee la licencia para su uso.
- El RED cita y referencia correctamente todos los trabajos utilizados en su creación.
- Cuando el RED no es creado por usted. Referencia correctamente el autor.

**Guardar y continuar**

**Cancelar**

## Anexo 4. Captura de pantalla de Agrega



The screenshot shows the Agrega website interface. At the top, there is a navigation bar with links for 'Inicio de Agrega', 'Bienvenido', 'Búsqueda (C)', 'Búsqueda (V)', 'OngLEtar!', 'Búsqueda', and 'Welcome'. On the right, there is a link for '¿Tienes cuenta? Acceder'. The main header features the 'agrega' logo with a '2' in a circle. Below the logo are three buttons: 'Buscador Agrega', 'Buscador Externo', and 'Árbol Curricular'. The main content area has the heading 'Encuentra los mejores contenidos educativos' and a search bar with a dropdown menu for 'Idioma' and a 'Buscar' button. There is also a link for 'Avanzado'. Below the search bar, there are radio buttons for 'Todo Agrega' (selected) and 'INTEFP'. A light blue box contains the text: 'Visita el nuevo Espacio Procomún Educativo que integra un buscador mejorado de recursos educativos en Agrega y en el que podrás participar en la nueva Red Social Docente' with an 'Acceder' button. The page is divided into two columns. The left column has a 'Noticias' section with a news item titled 'Versión 1.5.4\_C de Agrega el nodo INTEFP' dated 04/05/2016, and a 'Descargas' section with a download link for 'Descargar Agrega Offline versión 3.0.2 - Instalador de Windows (65 MB)'. The right column has a 'Palabras Clave' section with a list of keywords: 'dicotiledóneo', 'árbol', 'edificio', 'animal', 'arquitectura', 'dicotiledóneas', 'fachada', 'árbol', 'arco', 'planta', 'árbol', 'fachada', 'fachada'. The footer contains logos for 'intef red.es', 'GOBIERNO DE ARAGÓN', 'UNION EUROPEA', and 'agrega' with a '2' in a circle.

### Anexo 5. Aval del Cliente

El resultado obtenido (“Repositorio para la gestión de contenidos digitales de la disciplina Preparación para la Defensa en la Universidad de las Ciencias Informáticas”) garantiza la solución del problema planteado de cómo contribuir a la gestión de recursos educativos digitales de la disciplina Preparación para la Defensa.

Se obtuvo un repositorio para la gestión de contenidos digitales de la Preparación para la Defensa que permite el acceso de cada usuario a la información necesaria para resolver problemas relacionados con la docencia y el aprendizaje de manera organizada. Con la realización de la investigación se contribuye a:

- La creación de entornos propicios para el uso de las TIC.
- Apoyo del aumento de capacidades para el desarrollo sostenible de materiales de aprendizaje de calidad en el departamento de Enseñanza Militar.
- Promoción de la investigación sobre recursos educativos abiertos.

MSc. Profesor Auxiliar Raúl Martín Mendoza

Metodólogo de Seguridad Nacional del Departamento de Enseñanza Militar – UCI

### Anexo 6. Prototipo de interfaz de portada.

Pantalla principal	
1	Presenta un identificador visual del producto como nombre de marca, ícono, nombre de producto, genérico.
2	Presenta la versión del producto.
3	La página de Bienvenida está diseñada profesionalmente y va a crear una primera impresión positiva.
Minimizar carga de usuario	
4	El sistema está libre de información irrelevante e innecesaria
5	El sistema evita que los usuarios se registren innecesariamente.
6	La información es presentada en un orden lógico, simple y natural.
7	El sistema requiere muy poco desplazamiento y uso de “clics”.
8	Los íconos se corresponderán con las tareas a realizar.
9	Los usuarios pueden completar rápidamente tareas comunes.
10	Si existe una imagen en un ícono o botón, debe ser relevante a la tarea a la que se refiere.
Interfaz visible y práctica	
11	La información estará libre de errores gramaticales, ortográficos y tipográficos.

12	La taxonomía de los títulos encima de los controles estará según la estipulada, primera letra con mayúscula, el resto con minúscula.
13	Los nombres de los botones de comandos no serán abreviaturas y tendrán un significado claro para los usuarios del sistema.
14	Los botones de comando serán de tamaño y forma similares.
15	La ventana que se emplea para adicionar un elemento siempre comenzará el título con la palabra “Adicionar seguida por el elemento específico”.
16	La ventana que se emplea para modificar un elemento siempre comenzará el título con la palabra “Modificar seguida por el elemento específico”.
17	La fecha siempre se utilizará en formato de d/m/y (04/11/2023).
18	Las instrucciones del menú, las preguntas y los mensajes aparecen en el mismo lugar en cada página.
Control y autonomía	
19	Si una acción tiene consecuencias, el sistema proporciona información y pide confirmación antes de continuar.
20	El usuario tiene la opción de “Salir” para abandonar el sistema en cualquier momento.
Consistencia y diseño gráfico	
21	Para tareas similares, los diálogos y formularios son similares.
22	Un mismo elemento aparece igual en todas las interfaces.
23	La relación entre los controles y sus acciones es obvia.
24	Las fuentes son legibles.
25	Existe una correcta combinación de colores y se evitan los fondos complicados.
Ayuda	
26	Los mensajes al usuario son breves y no ambiguos.
27	Los mensajes de confirmación son claros.
28	Existe suficiente espacio entre los elementos de acción (links, botones, etc.) para prevenir que el usuario haga clic en el elemento incorrecto.
29	Los mensajes de error son escritos en lenguaje sencillo con suficiente explicación del problema.
30	¿En la interfaz se utiliza el lenguaje de los usuarios y no uno propio relacionado con la tecnología?

## Anexo 7. Prototipo de interfaz de portada.

**REPDEF** Repositorio Digital [Iniciar sesión](#)

[Inicio](#) [Documentos](#) [Videos](#) [Audios](#) [Imágenes](#)

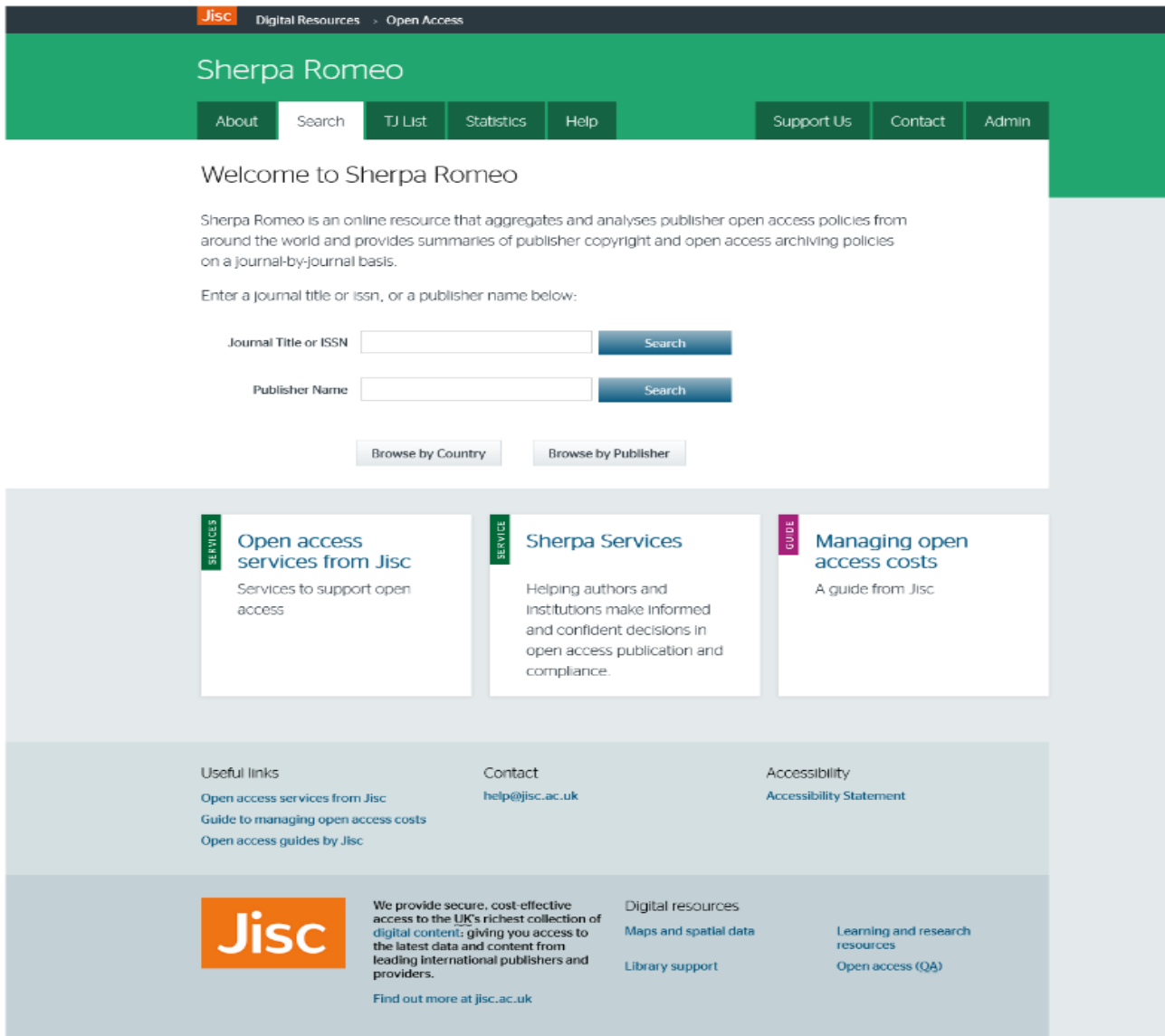
### Bienvenido a REPODEF v1.0

Repositorio digital del Departamento de Enseñanza Militar

Descubrir por:

Autor:	Temática:	Fecha:
Juana Maria Zuares 55	Ciencias 55	20-2-2020 55
Juana Maria Zuares 55	Tecnologías 55	20-2-2020 55
Juana Maria Zuares 55	Otros 55	20-2-2020 55
Juana Maria Zuares 55	Clases 55	20-2-2020 55
Juana Maria Zuares 55	JSociedad 55	20-2-2020 55

## Anexo 8: Captura de Pantalla de Sherpa Romeo



The screenshot shows the Sherpa Romeo website interface. At the top, there is a navigation bar with the Jisc logo and the text "Digital Resources > Open Access". Below this is a green header with the "Sherpa Romeo" title and a menu containing "About", "Search", "TJ List", "Statistics", "Help", "Support Us", "Contact", and "Admin".

The main content area features a "Welcome to Sherpa Romeo" section. It includes a brief description: "Sherpa Romeo is an online resource that aggregates and analyses publisher open access policies from around the world and provides summaries of publisher copyright and open access archiving policies on a journal-by-journal basis." Below this, there is a prompt: "Enter a journal title or issn, or a publisher name below:". This is followed by two search input fields: "Journal Title or ISSN" and "Publisher Name", each with a "Search" button. There are also two buttons: "Browse by Country" and "Browse by Publisher".

Below the search area, there are three service cards:

- Open access services from Jisc** (SERVICES): Services to support open access.
- Sherpa Services** (SERVICE): Helping authors and institutions make informed and confident decisions in open access publication and compliance.
- Managing open access costs** (GUIDE): A guide from Jisc.

The footer area is divided into three columns:

- Useful links:** Open access services from Jisc, Guide to managing open access costs, Open access guides by Jisc.
- Contact:** help@jisc.ac.uk
- Accessibility:** Accessibility Statement

At the bottom, there is a Jisc logo and a paragraph: "We provide secure, cost-effective access to the UK's richest collection of digital content: giving you access to the latest data and content from leading international publishers and providers. Find out more at jisc.ac.uk". To the right of this paragraph are three links: "Digital resources", "Maps and spatial data", and "Library support". Further right, there are two more links: "Learning and research resources" and "Open access (QA)".

## Anexo 9: Sección de la pantalla de inicio del Repositorio de Recursos Educativos de la Universidad Virtual de la Salud

The screenshot shows the homepage of the 'Repositorio de recursos educativos' (Educational Resources Repository) of the 'Universidad Virtual de la Salud' (Virtual University of Health). The header features the 'infomed' logo and navigation links for 'Contacto', 'BVS', and 'UVS'. Below the header, there is a search bar with the text 'Búsqueda avanzada' and a search icon. A navigation menu includes 'Inicio', 'Navegar por', 'Otros enlaces', 'Acerca de', and 'Ayuda'. The main content area is divided into two columns. The left column contains a graphic of a globe and a description of the repository: 'Repositorio de la Universidad Virtual de Salud' and 'Es una colección de recursos educativos y otros materiales útiles para el aprendizaje, estructurada como un banco o base de datos, con metadatos asociados y que permite la búsqueda en entornos web. Los recursos disponibles son, generalmente, archivos interoperables y accesibles.' The right column contains a 'CUENTA DE USUARIO:' section with input fields for 'Usuario:' and 'Contraseña:', a 'Registrarse' button, and a link for '¿Olvidó su contraseña?'. At the bottom, a dark blue bar displays 'Recursos' on the left and 'Total: 697 Recursos' on the right.

### Anexo 10: Página de inicio de CubaEduca.



[Inicio](#) [Quiénes somos](#) [Servicios](#) [Contactos](#) [Acceder](#)

## COMUNIDAD EDUCATIVA

Plataforma integradora e interactiva, que ofrece un conjunto de contenidos y servicios afines a la educación y que, mediante un proceso de co-creación, contribuye a una educación inclusiva, innovadora y de calidad.

Comencemos

Nuestros Servicios



### Anexo 11: Sección superior de la página de inicio del Repositorio Digital.



Servicios -

[Página de inicio](#) [Listar](#) [Ayuda](#)

Buscar en REPXOS

Repositorio Digital

Bienvenido a REPXOS 4.0

Esta es la página inicial de tu repositorio digital.

Sus futuros envíos aparecerán en esta página.