



CITEC

Repositorio para la gestión de recursos educativos

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

Autor(es): Fernando Lázaro Salabarría Cárdenas

Tutor(es): Dr. C. Yusniel Hidalgo Delgado

La Habana, Noviembre de 2022

Año 64 de la Revolución

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título "**Repositorio para la gestión de recursos educativos**", concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los __ días del mes de noviembre del año 2022.

Fernando Lázaro Salabarría Cárdenas

Firma del Autor

Dr. C. Yusniel Hidalgo Delgado

Firma del Tutor

DATOS DE CONTACTO

Yusniel Hidalgo Delgado. Profesor Auxiliar en el Departamento de Informática de la Facultad de Ciencias y Tecnologías Computacionales de la Universidad de las Ciencias Informáticas (UCI). Obtuvo el título de Ingeniero en Ciencias Informáticas, Máster en Informática Aplicada y Doctor en Informática por la UCI en el 2010, 2015 y 2021 respectivamente. Sus intereses de investigación son Web Semántica, Datos Enlazados e Ingeniería Ontológica. Es miembro de la Asociación Cubana de Reconocimiento de Patrones y de la Sociedad Cubana de Matemática y Computación. Posee varias publicaciones científicas en revistas y congresos internacionales. Actualmente se desempeña como Jefe del Grupo de investigación de Web Semántica y Jefe del Departamento de Informática de la CITEC de dicha universidad. Correo: yhdelgado@uci.cu.

AGRADECIMIENTOS

A todas las personas que de una forma u otra me ayudaron a llegar hasta aquí.

A mis padres, a mis compañeros de aula, a mis profesores, a mis amigos, al resto de los miembros de mi familia, a la Revolución.

DEDICATORIA

A mis padres por ser siempre la fuerza impulsora que me ha hecho superarme.

A mis profesores que han fortalecido mi interés por el estudio.

A mis alumnos del preuniversitario que me han hecho ver la educación de otra manera.

A mis amigos que siempre están ahí cuando hace falta.

A mis compañeros de aula que han sido faro y guía de mis proyectos.

RESUMEN

Existen muy pocas herramientas informáticas al alcance de los centros educacionales que permitan la obtención de multimedia educativa sin necesidad de conectarse directamente a buscarla en Internet. Por tanto se plantea como necesidad la creación de un repositorio informático que permita la gestión y visualización de los recursos educativos que posea una institución y su visualización. Para facilitar el trabajo y el uso de la misma se gestiona esta aplicación desde la 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 Ágil XP 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

Repositorio, gestión, recursos.

ABSTRACT

There are very few computer tools within the reach of educational centers that allow obtaining educational multimedia without the need to connect directly to search for it on the Internet. Therefore, the creation of a computer repository that allows the management and visualization of the educational resources that an institution has and its visualization is considered as a necessity. To facilitate the work and use of it, this application is managed from the web.

To carry out this project, pre-established tools and technologies were used and a study of the characteristics of existing systems for working with web applications in the world and in Cuba was carried out. Following the steps proposed by the Agile XP 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 conformed for its access as a web application.

KEYWORDS

Repository, management, resources.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	13
CAPÍTULO 1: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO.....	18
1.1 Recursos educativos.....	18
1.1.1 Objetos de Aprendizaje.....	19
1.2 Repositorios digitales.....	21
1.2.1 Repositorios de recursos educativos.....	21
1.2.2 Estudio de repositorios similares a nivel internacional y nacional.....	23
1.3 Gestión de recursos.....	26
1.4 Aplicación Web.....	27
1.5 Lenguajes, tecnologías y herramientas utilizadas.....	27
1.5.1 Lenguaje de programación.....	28

1.5.1.1 PHP.....	28
1.5.1.2 JavaScript.....	29
1.5.1.3 Ajax.....	30
1.5.2 Herramientas de modelado.....	31
1.5.2.1 Visual Paradigm.....	31
1.5.3 Sistema Gestor de Bases de Datos.....	32
1.5.3.1 MySQL.....	33
1.5.4 Servidores de Aplicaciones Web.....	33
1.5.4.1 Apache.....	34
1.5.5 Framework.....	34
1.5.5.1 WordPress.....	35
1.5.6 Entorno de Desarrollo Integrado (IDE).....	35
1.5.6.1 NetBeans.....	36
1.4.6.2 PHPStorm.....	36
1.6 Metodología de desarrollo de software.....	37
1.6.1. Metodologías tradicionales.....	37
1.6.2 Metodologías ágiles.....	38
Conclusiones del capítulo.....	40
CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO.....	41
2.1 Modelo de Dominio.....	41
2.2 Requerimientos.....	42
2.2.1 Requisitos funcionales del sistema.....	42
2.2.2 Requisitos no funcionales.....	44
2.3 Identificación de los actores y casos de uso del sistema.....	47
2.3.1 Casos de Uso del Sistema.....	47
2.4 Historias de Usuario.....	48
2.5 Plan de Entregas.....	56
2.6 Patrones de diseño.....	57
Conclusiones del capítulo.....	59
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	60
3.1 Estándar de codificación.....	60
3.1.1 General.....	60
3.1.2 Espacio de nombres y declaraciones.....	60
3.1.3 Clases, propiedades y métodos.....	61
3.1.4 Estructuras de Control.....	61
3.2 Implementación.....	61
3.2.1 Diagrama de despliegue.....	62
3.3 Pruebas.....	62
3.3.1 Pruebas de caja negra.....	63

3.3.2 Pruebas de caja blanca	65
3.3.3 Pruebas de usabilidad	65
Conclusiones del capítulo	66
CONCLUSIONES FINALES.....	67
RECOMENDACIONES.....	68
REFERENCIAS BIBLIOGRÁFICAS.....	69
ANEXOS.....	71
Anexo 1: Captura de pantalla de Agrega.....	71
Anexo 2: Captura de Pantalla de SSRN.....	72
Anexo 3: Captura de Pantalla de Sherpa Romeo.....	73
Anexo 4: Sección de la pantalla de inicio del Repositorio de Recursos Educativos de la Universidad Virtual de la Salud.....	74
Anexo 5: Página de inicio de CubaEduca.....	75
Anexo 6: Sección superior de la página de inicio del Repositorio Digital.....	75
Anexo 7: Curvas de relación Coste/Tiempo en metodologías tradicionales (Izquierda) y ágiles (Derecha).....	76

ÍNDICE DE TABLAS

Tabla 1: Tabla de estudio de homólogos.....	26
Tabla 2: Descripción de los actores del sistema.....	49
Tabla 3: Historia de Usuario Autenticar.....	51
Tabla 4: Historia de Usuario. Importar Recurso.....	51
Tabla 5. Historia de Usuario. Exportar Recurso.....	52
Tabla 6. Historia de Usuario. Descargar Recurso.....	52
Tabla 7: Historia de Usuario. Visualizar Recurso.....	52
Tabla 8: Historia de Usuario. Validar Recurso.....	53
Tabla 9: Historia de Usuario. Eliminar Recurso.....	53
Tabla 10: Historia de Usuario: Clasificar Recurso.....	53
Tabla 11: Historia de Usuario. Denunciar Recurso.....	54
Tabla 12: Historia de Usuario. Recomendar Recurso.....	54
Tabla 13: Historia de Usuario. Compartir Recurso.....	55
Tabla 14: Historia de Usuario. Adicionar Recurso a Favoritos.....	55
Tabla 15: Historia de Usuario. Listar Recursos Favoritos.....	55
Tabla 16: Historia de Usuario. Adicionar Comentario.....	56
Tabla 17: Historia de Usuario. Editar Comentario.....	56
Tabla 18: Historia de Usuario. Responder Comentario.....	56
Tabla 19: Historia de Usuario. Denunciar Comentario.....	57
Tabla 20: Historia de Usuario. Eliminar Comentario.....	57
Tabla 21: Historia de Usuario. Ver Comentario.....	58
Tabla 22: Historia de Usuario. Ver Ficha de un Recurso.....	58
Tabla 23: Plan de Entregas.....	59
Tabla 24: Descripción del Caso de Prueba del requisito Autenticar.....	64

ÍNDICE DE FIGURAS

Figura 1: El modelo tradicional para las aplicaciones Web (izq.) comparado con el modelo de AJAX (der.).....	31
Figura 2: Proceso de SCRUM.....	41
Figura 3: Capas de Extreme Programming (XP).....	42
Figura 4: Modelo de dominio.....	44
Figura 5: Diagrama de despliegue.....	64
Figura 6: Resultados de las iteraciones.....	66

OPINIÓN DEL(OS) TUTOR(ES)

<Contenido de la opinión de los tutores>

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada>

INTRODUCCIÓN

En los últimos años se ha venido produciendo a nivel mundial un vertiginoso desarrollo de las ciencias informáticas y dentro de ellas el proceso de desarrollo de software. La mayoría de las producciones, exportaciones y despliegue de productos de software se concentran en países como Estados Unidos, Japón, Rusia, Israel, China, Corea del Sur, los países de la Unión Europea, Australia y Nueva Zelanda, todos ellos países altamente industrializados que dedican buena parte de su Producto Interno Bruto a la investigación de técnicas que agilicen el desarrollo de aplicaciones informáticas. La globalización además ha permitido que otras naciones emergentes como India, Malasia, Indonesia, Singapur, Pakistán, Irán, Sudáfrica entre otros se hayan insertado a este emergente mercado del software.

En el contexto latinoamericano, naciones como Brasil, Argentina, Uruguay, Chile, Costa Rica y México han dado pasos agigantados hacia la construcción de industrias del software que se conviertan en referencias mundiales durante los próximos años con el consiguiente aumento de las exportaciones de sus productos y los ingresos por la elaboración de los mismos al alcanzar altas cotas de calidad.

Cuba, a pesar de sus limitantes como país subdesarrollado no es ajena al desarrollo de esta industria y el Estado se ha planteado como uno de sus objetivos el desarrollo de la Industria del Software para fomentar la informatización de la sociedad y la sustitución de las importaciones de soluciones informáticas extranjeras por productos nacionales de calidad que además se puedan insertar en el saturado mercado internacional. Una de las esferas en la que Cuba ha puesto más énfasis en informatizar es la educación, lo que ha traído como consecuencia diversas investigaciones en este campo para propiciar servicios informáticos que contribuyan a la agilización de los procesos educativos, al perfeccionamiento de la preparación de docentes y estudiantes y la creación de nuevos métodos de aprendizaje basados en la implementación de las nuevas tecnologías.

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. 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 educativos.

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 nivel internacional existe una variedad de repositorios para fines educativos. 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.

Debido a las restricciones impuestas a Cuba por el bloqueo económico y comercial de los Estados Unidos que tiene trascendencia en el ámbito de las ciencias informáticas, la comunidad universitaria cubana no puede acceder a la mayoría de estos valiosos recursos educativos sin utilizar medios alternativos (VPN, proxy, entre otros) y por la imposibilidad de acceder a contenidos de pago online existentes en estos repositorios.

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. En la 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.

El Repositorio Digital no permite la gestión de recursos educativos en todos los formatos predefinidos, sean imágenes, videos, multimedia ni el procesamiento de recursos en los nuevos formatos actuales. Además no existe en la universidad un repositorio centralizado que contenga acceso a recursos digitales que permitan la formación de los estudiantes y profesores de la Universidad de Ciencias Informáticas y a la vez permita la búsqueda de recursos e información actualizada a centros universitarios de disciplinas no afines a la carrera.

También se puede constatar que este repositorio no ofrece ningún tipo de integración con las redes sociales, una de las tendencias actuales observadas en repositorios educativos a nivel internacional.

A partir de la problemática descrita anteriormente se obtiene el siguiente **problema científico**:

¿Cómo desarrollar un repositorio que centralice la gestión de recursos educativos en la comunidad universitaria y cumpla con las tendencias actuales?

El **objeto de estudio** es la gestión de recursos educativos en repositorios de recursos educativos digitales y el **campo de acción**: repositorio informático para la gestión de recursos educativos digitales.

Para dar solución al problema científico declarado, esta investigación tiene como **objetivo general**:

Desarrollar un repositorio de recursos educativos mediante la utilización de tendencias y tecnologías actuales que facilite la gestión de recursos educativos a la UCI y el resto de la comunidad universitaria en Cuba. A partir de este objetivo general se trazaron los siguientes **objetivos específicos**:

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

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 centro educativo con respecto al proceso de Gestión de recursos educativos, motivándonos a de realizar la investigación.

Aportes prácticos esperados de la investigación:

Se espera obtener un repositorio para la gestión de recursos educativos que permita el acceso de cada usuario a la información necesaria para resolver problemas relacionados con la docencia y el aprendizaje de manera organizada.

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

Capítulo 1. “FUNDAMENTACIÓN TEÓRICA”: Describe el proceso de gestión de recursos educativos mediante el uso de un repositorio. Se manifiesta el estado del arte de

las tecnologías a usar en el desarrollo de la aplicación informática y se exponen las herramientas y metodologías a utilizar.

Capítulo 2. “CARACTERÍSTICAS DEL SISTEMA”: Presenta una descripción de las características del repositorio, sus requisitos funcionales y no funcionales y las historias de usuario en correspondencia con la metodología utilizada.

Capítulo 3. “VALIDACIÓN DE LA SOLUCIÓN PROPUESTA”: Se describe la solución que se propone a partir del estándar de codificación utilizado, el diagrama de despliegue del software y las pruebas realizadas a la aplicación.

CAPÍTULO 1: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO

Introducción:

Este capítulo es el respaldo teórico de los temas tratados en el informe, necesarios para el entendimiento correcto de la solución planteada. Se describen los conceptos fundamentales asociados al dominio del problema y el objeto de estudio, haciéndose un análisis de los aspectos relacionados con la gestión de recursos educativos. Por último, se presenta la fundamentación de las tecnologías utilizadas para el diseño de la aplicación.

1.1 Recursos educativos

Con el desarrollo de las TIC, el concepto de recurso educativo se ha visto enriquecido, vinculado al mismo existen diversas definiciones, incluso algunas bibliografías tratan este término como recursos digitales o recursos educativos abiertos.

El término Recursos Educativos Abiertos (REA), fue expuesto por la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) en el año 2002, con el objetivo de ofrecer de forma abierta recursos educativos provistos por medio de las TIC, para su consulta, uso y adaptación con fines no comerciales, son catalogados como: “materiales de enseñanza, aprendizaje o investigación que se encuentren en el dominio público o que han sido publicados con una licencia de propiedad intelectual que permite su utilización, adaptación y distribución gratuita” (1).

Otras fuentes consultadas definen los REA como: “recursos destinados para la enseñanza, el aprendizaje y la investigación que residen en el dominio público o que han sido liberados bajo un esquema de licenciamiento que protege la propiedad intelectual y permite su uso de forma pública y gratuita o permite la generación de obras derivadas por otros. Los Recursos Educativos Abiertos se identifican como cursos completos, materiales de cursos, módulos, libros, videos, exámenes, software y cualquier otra herramienta, materiales o técnicas empleadas para dar soporte al acceso de conocimiento” (2).

Otros estudiosos de la materia consideran los REA como: “materiales en formatos digitales que se ofrecen de manera gratuita y abierta para educadores, estudiantes y autodidactas para su uso y re-uso en la enseñanza, el aprendizaje y la investigación” (3).

“Los recursos educativos digitales son materiales compuestos por medios digitales y producidos con el fin de facilitar el desarrollo de las actividades de aprendizaje” (4).

Las definiciones anteriores coinciden en que los recursos educativos son materiales digitales que facilitan la enseñanza y el aprendizaje de manera que puedan ser accesibles y gratuitos, teniendo en cuenta esta observación, en su tesis doctoral Cañizares propone: “los recursos educativos están compuestos por contenidos educativos que pueden ser cursos completos, materiales para cursos, materiales multimedia, objetos de aprendizaje y cualquier otro recurso con fin educativo con una alta pertinencia y certificación de calidad. Son manejados por herramientas, para su accesibilidad y disponibilidad, y a su vez, pueden ser distribuidos de forma gratuita y libre de restricciones o bajo una licencia” (5). Después del análisis de los conceptos presentados, se adopta el enunciado anterior en la presente investigación teniendo en cuenta el nivel de generalización que se utiliza para identificar un recurso educativo.

1.1.1 Objetos de Aprendizaje

En el ámbito de la enseñanza, la corriente más destacada en cuanto al diseño de recursos educativos en formato digital es la de objetos de aprendizaje. Los OA son recursos educativos especiales, estos se caracterizan por cumplir determinadas pautas o requisitos.

Formalmente, no existe una definición única de OA, por el contrario, las definiciones son muy amplias, a continuación se analizan algunas de ellas:

En la bibliografía consultada, López Guzmán en su tesis doctoral en el año 2005, precisa como OA: “cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning ” (6).

Sin embargo, otros autores lo definen como: “cualquier recurso digital que puede ser reutilizado para apoyar el aprendizaje” (7).

También López Guzmán afirma que un OA es un “ recurso educativo digital relativamente pequeño y auto-contenido, con una marcada intención formativa, compuesto por uno o varios objetos de información, con un objetivo, descrito con metadatos y con un comportamiento secuenciado que asegure el correcto enlace entre los elementos de su estructura didáctica; concebido para diferentes poblaciones según sus contextos socioculturales para lograr su

reutilización e interoperabilidad en varios entornos e-learning, que contribuya a la transmisión de conocimiento y la formación de valores tanto en los profesores, diseñadores, como en los usuarios finales”. Partiendo del análisis de los conceptos anteriores, se decide hacer uso de este último, para la presente investigación, pues contiene todos los elementos fundamentales que identifican un OA.

Los OA tienen una serie de características específicas que los diferencian de los materiales de aprendizaje tradicionales. Estas se basan en el modelo SCORM (*Sharable Content Object Reference Model*), traducido como “Modelo de Referencia para Objetos de Contenido Compartibles” y son las siguientes:

-Accesibilidad: capacidad de acceder a los componentes de enseñanza desde un sitio a través de la tecnología web, y como distribuirlos a otros sitios.

-Adaptabilidad: capacidad de personalizar la formación en función de las necesidades de las personas y organizaciones.

-Durabilidad: capacidad de resistir a la evolución de la tecnología sin necesidad de una reconfiguración o una reescritura del código.

-Interoperabilidad: capacidad de utilizarse en otro emplazamiento y con otro conjunto de herramientas o sobre otra plataforma de componentes de enseñanza desarrolladas dentro de un sitio, con un cierto conjunto de herramientas o sobre una plataforma dada.

-Reusabilidad: flexibilidad que permite integrar componentes de enseñanza dentro de múltiples contextos y aplicaciones (8).

Los recursos educativos de manera general, representan un factor fundamental en el aprendizaje electrónico, por lo que resulta necesario prever que los mismos puedan ser consultados y accedidos en el momento necesario. Para garantizar estos aspectos, así como su reutilización, los recursos se almacenan y gestionan en contenedores llamados repositorios. Estos sistemas permiten además, visualizar, exportar, importar y reemplazar recursos, así como realizar comentarios y sugerencias a los mismos.

1.2 Repositorios digitales

Según el diccionario de la Real Academia Española (REA) el término repositorio significa: "Almacén o lugar donde se guardan ciertas cosas" (9). Con el surgimiento de la era digital algunos autores sostienen la definición de repositorios digitales como: "un sitio centralizado donde se almacena y mantiene información digital" (6).

Los repositorios digitales son básicamente sistemas en los que se guardan objetos digitales como vídeos, animaciones, imágenes, documentos y libros digitales. Estas aplicaciones organizan los objetos por categorías, áreas del conocimiento o contenidos. Sus objetivos esenciales son la búsqueda, acceso y visualización de materiales digitales, garantizando así la preservación y difusión de los mismos. En el mundo digital, los repositorios se clasifican en tres grandes grupos: institucionales, temáticos y centralizados (10).

-Repositorios institucionales: Conjunto de servicios prestados por las universidades o centros de investigación a su comunidad para recopilar, gestionar, difundir y preservar su producción científica digital a través de una colección organizada, de acceso abierto e interoperables. Son creados por las propias organizaciones para depositar, usar y preservar la producción científica y académica que generan (10).

-Repositorios temáticos: recogen la producción de determinadas áreas de conocimiento a nivel internacional (10).

-Repositorios centralizados: recopilan la producción científica depositada en abierto por investigadores de distintas instituciones y de distintas áreas temáticas (10).

Teniendo en cuenta las definiciones anteriores, el curso de la presente investigación, se remonta a los repositorios centralizados, que como su concepto lo indica, permiten el almacenamiento de los recursos creados en distintas instituciones.

1.2.1 Repositorios de recursos educativos

Los repositorios de recursos educativos constituyen una distinción especial de los repositorios digitales, definidos por las características de los recursos que almacenan.

Algunos autores definen a los repositorios de recursos educativos como: "un tipo de repositorio que permite el almacenamiento, la gestión y el uso (reutilización) de recursos educativos" (7).

Otros autores utilizan el término de Repositorio Institucional de Recursos Educativos Digitales Abiertos y lo definen como: “un sistema de información digital, definido, gestado, implementado y operado por una Institución de Educación Superior, cuya función principal es recopilar, almacenar, ordenar, localizar, preservar y redistribuir los recursos educativos digitales creados y producidos por la institución y los miembros de su comunidad, con la finalidad de disponerlos en un entorno web, de acceso público ” (11).

En la presente investigación se toma como referencia el concepto que brinda Cañizares donde expresa: “un sistema especializado en el almacenamiento de recursos educativos, accesibles y operables por usuarios a través de Internet/Intranet, con funciones encaminadas a la clasificación, localización, reutilización, recuperación y mantenimiento de los recursos, que permita compartir estos con otras herramientas de un entorno e-learning” (12).

Los Repositorios Institucionales de Recursos Educativos Digitales Abiertos deben proveer los siguientes servicios:

- Almacenamiento: capacidad del repositorio que permite la gestión de alojamiento y preservación de los archivos de los recursos educativos digitales.

- Búsqueda y recuperación: función que permite la consulta, exploración y recuperación de información sobre los recursos que contiene el repositorio.

- Navegación: función que permite explorar los diferentes servicios, secciones y colecciones que se encuentran en el repositorio.

- Visualización: función que se establece para presentar la respuesta que se obtiene de la búsqueda.

- Gestión de colecciones: servicio que facilita el control, manejo y organización de la oferta de los recursos educativos digitales.

- Gestión de usuarios: servicio que permite la identificación del acceso al repositorio de diferentes tipos de usuarios según los privilegios otorgados (12).

Otros autores consultados, refieren que dentro de las funcionalidades básicas de los repositorios de recursos educativos deben encontrarse las siguientes:

- Importar un recurso educativo creado previamente por una herramienta de autor.

- Publicar un recurso educativo después de pasar por un proceso de revisión.

- Descargar los recursos educativos (12).

En la actualidad, la mayor parte de los repositorios existentes contienen las funcionalidades anteriores, algunos incluso, comprenden la creación de OA. En el siguiente epígrafe, se muestra un estudio realizado de diferentes repositorios con el objetivo de identificar las funcionalidades comunes referentes a la gestión de los recursos que almacenan.

1.2.2 Estudio de repositorios similares a nivel internacional y nacional

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 (13). 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 1.

-Social Science Research Network (SSRN)

Es un repositorio digital de recursos educativos abiertos con sede en Estados Unidos, fundado en 1994, según el ranking de repositorios web, se ha convertido en el principal repositorio de acceso abierto a nivel mundial. SSRN ha crecido hasta convertirse en el servicio más interdisciplinario de su tipo, representando disciplinas en todo el espectro de investigación, incluidas las ciencias aplicadas, ciencias de la salud, humanidades, ciencias de la vida, ciencias físicas y sociales. Es fundamental tanto como punto de partida para estudiantes y profesores en el proceso de investigación, como para comunicar sus investigaciones en una etapa temprana, antes de su publicación en revistas académicas (14).

La mayoría de los documentos presentes en el repositorio se pueden descargar de manera gratuita, excepto aquellos cuyos derechos pertenecen a terceras personas, para los cuales se solicita una tarifa de descarga.

Se aprecia la implementación en el repositorio de un carrito de compras, representando una tendencia actual en los repositorios y en otras aplicaciones informáticas, el comercio electrónico.

El repositorio es accesible desde Cuba mediante un software VPN (Virtual Private Network) y los métodos de pago tampoco están disponibles para nuestro país. El Anexo 2 contiene una pantalla del repositorio SSRN.

-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 (15). 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 3.

-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 (16). 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 4 se aprecia una sección de su página de inicio.

-CubaEduca

CubaEduca es un repositorio de recursos educativos desarrollado por la empresa CINESOFT, que constituye una comunidad educativa en red. Ofrece un conjunto de contenidos, recursos y servicios afines a la educación. El sistema sirve además de host a varias páginas del MINED que facilitan la consulta de los contenidos de aprendizaje necesarios en cada nivel de enseñanza. Contiene múltiples colecciones de software de estudio desarrollado para distintos grados. Permite la creación, autenticación de usuarios y la descarga de recursos educativos. Los contenidos se clasifican por años y por materias. CubaEduca también se acoge a la nueva tendencia del comercio electrónico. Su repositorio no brinda contenido para estudiantes y docentes de la educación superior, se limita a las enseñanzas preescolar, primaria, media y media superior. En el Anexo 5 se observa la página de inicio de CubaEduca.

-Repositorio Digital REPXOS: 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 6 se visualiza la sección superior de la página de inicio del Repositorio Digital.

A modo de conclusión se tabuló el resultado de la investigación de los elementos presentes en los repositorios analizados en una tabla.

Tabla 1: Tabla de estudio de homólogos

Repositorios	Ubicación	Acceso desde Cuba	Tipo	Contiene Recursos para Educación Superior	Tendencias Actuales. (1, 2, 3)
Agrega	Europa	Parcial.	Centralizado	Si	1, 3
SSRN	Estados Unidos	No permitido (Requiere VPN)	Centralizado	SI	1, 2, 3
Sherpa Romeo	Europa	Parcial	Centralizado	Si	1, 2, 3
Repositorio UVS	Cuba	Si	Temático	Si	Ninguna
CubaEduca	Cuba	Si	Centralizado	No	1, 2, 3
Repxos	Cuba	Si	Institucional	Si	Ninguna

1.3 Gestión de recursos

Según el diccionario de la RAE la palabra gestión significa “acción y efecto de gestionar o administrar” (17). Partiendo de este concepto se llega a la visión empresarial de gestión de recursos, muy relacionada con la administración, concepto definido como: proceso de planificación y programación de los recursos y actividades de un equipo de trabajo (18). La gestión de recursos contiene muchos campos de impacto conocidos como la gestión de recursos humanos y la gestión de información.

La gestión de la información ha ganado en los últimos años mucha importancia por su impacto directo en todas las esferas del conocimiento y su importancia para el crecimiento de la economía de una nación. Woodman considera que “la gestión de información es todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado, para tomar la acción precisa” (19).

Los recursos educativos constituyen información de carácter útil para las instituciones que hacen uso de estos.

Tomando en cuenta los conceptos anteriores la gestión de recursos educativos digitales es el proceso por el que se generan, obtienen, procesan y administran recursos necesarios para el desarrollo de los procesos de una institución educativa que requieran para su despliegue el uso de las tecnologías de la información.

1.4 Aplicación Web

Las aplicaciones Web son aquellas herramientas donde los usuarios pueden acceder a un servidor Web a través de la red mediante un navegador determinado. Por lo tanto, se define como una aplicación que se accede mediante la Web por una red ya sea intranet o Internet. Por lo general se denomina aplicación Web a aquellos programas informáticos que son ejecutados a través del navegador (20).

Otros autores indican que de manera general: “las aplicaciones web no son más que las herramientas de ofimática (procesadores de texto, hojas de cálculo, entre otros), de la web que se manejan con una conexión a internet, y en estos casos cabe la opción de utilizar el ordenador solo como forma de acceso a la aplicación remota” (21).

1.5 Lenguajes, tecnologías y herramientas utilizadas

Para la selección del lenguaje de programación, así como la utilización de los diversos componentes; se consultaron diferentes características como portabilidad, multiplataforma y facilidad de uso, resultando seleccionados los que más se acercan a los requerimientos de la plataforma. La creciente informatización de los procesos productivos y sociales provoca que las organizaciones y empresas requieran cada vez más de aplicaciones de software confiables y de calidad tanto en su desarrollo como en su mantenimiento. Es por ello que en los últimos años se han venido publicando estándares, notaciones y procesos de desarrollo de Software que establecen las buenas prácticas para el desarrollo de aplicaciones de software.

1.5.1 Lenguaje de programación

Un **lenguaje de programación** es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana (22).

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se

escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.

También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).
- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.
- Desarrollo de la documentación.

1.5.1.1 PHP

PHP (acrónimo recursivo de "PHP: Hipertext Preprocessor", originado inicialmente del nombre PHP Tools, o Personal Home Page Tools) es un lenguaje del lado del servidor, que se utiliza básicamente para generar páginas dinámicas y que facilita de forma sencilla el acceso a diferentes bases de datos. También se utiliza para generar imágenes, generar pdf. PHP lo desarrollan programadores Web introduciendo nuevas funciones en las versiones mejoradas del lenguaje (23). La última versión estable del lenguaje de programación PHP es la 8.1.9.

Ventajas de PHP:

- Libre y Abierto (Código fuente disponible, es gratuito).
- Multiplataforma: inicialmente fue diseñado para entornos UNIX por lo que ofrece más prestaciones en este sistema operativo, pero es perfectamente compatible con Windows.
- Soporte para varios servidores Web.
- Fácil acceso a Bases de Datos.
- Mucha documentación (Manuales, tutoriales, cursos).

-Presenta una integración perfecta entre Apache-PHP-MySQL.

-Posee una sintaxis bastante clara.

-Fácil aprendizaje.

-Seguro.

-Orientado a objetos.

1.5.1.2 JavaScript

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con JavaScript podemos crear diferentes efectos e interactuar con nuestros usuarios. Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros (24).

JavaScript es soportado por la mayoría de los navegadores como Microsoft Edge, Safari, Opera, Mozilla Firefox y Google Chrome.

Con el surgimiento de lenguajes como PHP del lado del servidor y JavaScript del lado del cliente, surgió Ajax en acrónimo de (Asynchronous JavaScript And XML). El mismo es una técnica para crear aplicaciones web interactivas. Este lenguaje combina varias tecnologías:

HTML y hojas de estilos CSS.

XMLHttpRequest es una de las funciones más importantes que incluye, que permite intercambiar datos asincrónicamente con el servidor web, puede ser mediante PHP, ASP, entre otros. La última versión estable de JavaScript es ECMAScript 2021.

1.5.1.3 Ajax

Ajax no es una tecnología. Es realmente muchas tecnologías, cada una floreciendo por su propio mérito, uniéndose en poderosas nuevas formas. AJAX incorpora:

- Presentación basada en estándares usando XHTML y CSS.
- Exhibición dinámica usando el Document Object Model.
- Intercambio y manipulación de datos usando XML and

XSLT.

- Recuperación de datos usando JavaScript.

Una aplicación AJAX elimina la naturaleza arrancar-frenar- arrancar-frenar de la interacción en la Web introduciendo un intermediario -un motor AJAX- entre el usuario y el servidor. Parecería que sumar una capa a la aplicación la haría menos reactiva, pero la verdad es lo contrario (25).

En vez de cargar un página Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript y usualmente procesado en un frame oculto). Este motor es el responsable por renderizar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario. El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo.

Cada acción de un usuario que normalmente generaría un requerimiento HTTP toma la forma de un llamado JavaScript al motor AJAX en vez de ese requerimiento. Cualquier respuesta a una acción del usuario que no requiera un viaje de vuelta al servidor (como una simple validación de datos, edición de datos en memoria, incluso algo de navegación) es manejada por su cuenta. Si el motor necesita algo del servidor para responder (sea enviando datos para procesar, cargar código adicional, o recuperando nuevos datos) hace esos pedidos asincrónicamente, usualmente usando XML, sin frenar la interacción del usuario con la aplicación.

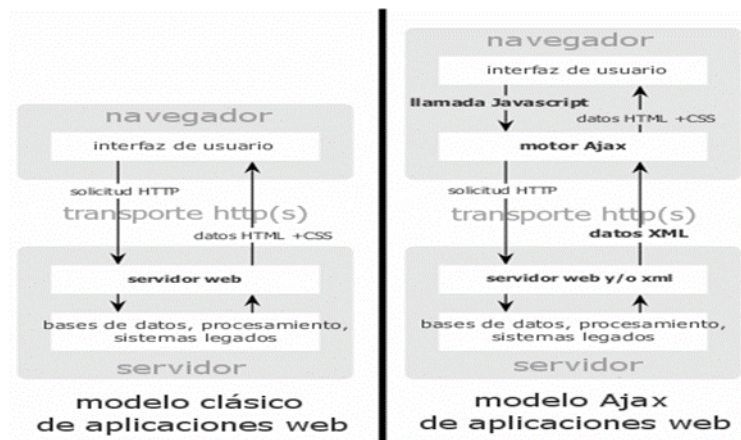


Figura 1: El modelo tradicional para las aplicaciones Web (izq.) comparado con el modelo de AJAX (der.)

1.5.2 Herramientas de modelado

El modelado de sistemas software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al ingeniero de software a "visualizar" el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente. Por último, las herramientas de modelado pueden ayudar a verificar la corrección del modelo.

Algunas de las herramientas CASE conocidas son el ArgoUML, Rational Rose, Visual Paradigm, Easy CASE, Xcase, CASE Studio 2, CASEWise entre otras. Dentro de las más utilizadas se encuentra el Visual Paradigm.

1.5.2.1 Visual Paradigm

Visual Paradigm, es una herramienta CASE que provee soporte para la generación de código, ingeniería inversa para Java. Se integra con Eclipse, Borland JBuilder y Oracle JDeveloper, para soportar las fases de implementación en el desarrollo de software. Se diferencia de otras aplicaciones por su portabilidad y gran facilidad de uso. (26)

Utiliza UML como lenguaje de modelado.

Características de **Visual Paradigm**.

- Es un producto de calidad.
- Soporta aplicaciones web.

-Se integra con las siguientes herramientas Java: Eclipse/IBM, WebSphere, JBuilder, NetBeans IDE, Oracle JDeveloper, BEA Weblogic.

-Es fácil de instalar y actualizar.

-Logra compatibilidad entre ediciones.

Teniendo en cuenta las características antes mencionadas del Visual Paradigm, especialmente su buena integración con PHP, lenguaje que se utilizará para desarrollar la aplicación Web y la ventaja de que presenta una interfaz de usuario de fácil uso y que permite realizar los diagramas y artefactos que se generan durante el desarrollo del software además de realizar un control de las versiones durante todo el ciclo de trabajo, es la herramienta escogida para el modelado del repositorio.

La última versión estable de Visual Paradigm es la 17.2022.

1.5.3 Sistema Gestor de Bases de Datos

Un **sistema gestor de base de datos (SGBD)** es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de consulta y de generación de informes, o bien mediante aplicaciones al efecto.

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

Generalmente se accede a los datos mediante lenguajes de consulta, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican las consultas y la presentación de la información. Un SGBD permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Las bases de datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con esmero.

1.5.3.1 MySQL

Existen diferentes arquitecturas para los sistemas de gestión de bases de datos, pero la más extendida, y la que más éxito ha tenido, es la arquitectura relacional MySQL es un servidor de bases de datos relacionales muy rápido y robusto. El mismo proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multihilos, multiusuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido (27).

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Este gestor se creó con la rapidez en mente, de modo que no tiene muchas de las características de los gestores comerciales más importantes, como Oracle, Sybase o SQL Server. No obstante, eso no ha impedido que sea el más indicado para aplicaciones que requieren muchas lecturas y pocas escrituras y no necesiten de características muy avanzadas, como es el caso de las aplicaciones Web y por tanto de la propuesta de solución que se ofrece con este trabajo debido a su gran rapidez y facilidad de uso.

Se decidió usar MySQL porque es una solución libre, ofrece facilidad, y posee una buena integración con PHP.

La última versión estable de MySQL es la 8.0.16.

1.5.4 Servidores de Aplicaciones Web

Un servidor de aplicaciones es un programa de servidor en un equipo en una red distribuida que proporciona la lógica de negocio para un programa de aplicación. El servidor de aplicaciones se ve frecuentemente como parte de una aplicación de tres niveles, que consta de un servidor gráfico de interfaz de usuario (GUI), un servidor de aplicaciones (lógica empresarial) y un servidor de bases de datos y transacciones (28).

1.5.4.1 Apache

El servidor Apache es un software (libre) servidor HTTP de código abierto para plataformas UNIX, Windows entre otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache es de acuerdo al estudio hecho por Netcraft, el servidor WWW más utilizado del momento. Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad. Apache ha demostrado ser sustancialmente más rápido que muchos otros servidores libres y compete de cerca con los mejores servidores comerciales, es el servidor Web por excelencia (29).

Entre sus características más sobresalientes están:

-Fiabilidad: Alrededor del 90% de los servidores con más alta disponibilidad funcionan con Apache.

-Gratuidad: Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.

-Extensibilidad: Se pueden añadir módulos para ampliar las ya de por sí amplias capacidades de Apache. Hay una amplia variedad de módulos, que permiten desde generar contenido dinámico (con PHP, Java, Perl, Python), monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, hasta crear servidores virtuales por IP o por nombre (varias direcciones Web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos módulos incluso pueden ser creados por cualquier persona con conocimientos de programación. La última versión estable de Apache es la 2.4.54.

1.5.5 Framework

Un **framework**, **entorno de trabajo** o **marco de trabajo** es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

1.5.5.1 WordPress.

WordPress es un sistema de gestión de contenidos web (conocido en inglés como CMS, siglas de Content Management System) para facilitar la publicación de contenido en la web de manera intuitiva y rápida. Desde hace años se ha convertido en el líder absoluto a nivel mundial en la creación de sitios web.

Es un software de código abierto, soportado por una poderosa comunidad, que trabaja constantemente en la mejora de la sintaxis del CMS.

Ventajas de WordPress:

1. Gratuito: Es un software libre de código, que no requiere pagar por su uso y ofrece poderosas herramientas para crear aplicaciones web profesionales sin realizar costosas inversiones.
2. Complementos: Su amplia comunidad de desarrolladores genera gran cantidad de plugins o complementos para mejorar la gestión y experiencia del usuario.
3. Plantillas de personalización de contenido: WordPress posee su propio repositorio de plantillas para satisfacer las necesidades del desarrollador.
4. Posibilidad de gestionar desde dispositivos móviles: Permite gestionar el desarrollo de la aplicación sin tener a mano el ordenador (30).

La última versión estable de WordPress es la 6.1.1.

1.5.6 Entorno de Desarrollo Integrado (IDE)

Es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen auto-completado inteligente de código (IntelliSense). Algunos IDE contienen un compilador, un intérprete, o ambos, tales como NetBeans y Eclipse; otros no, tales como SharpDevelop y Lazarus.

El límite entre un IDE y otras partes del entorno de desarrollo de software más amplio no está bien definido. Muchas veces, a los efectos de simplificar la construcción de la interfaz gráfica de usuario (GUI, por sus siglas en inglés) se integran un sistema controlador de versión y varias herramientas. Muchos IDE modernos también cuentan con un navegador de clases, un buscador de objetos y un diagrama de jerarquía de clases, para su uso con el desarrollo de software orientado a objetos.

1.5.6.1 NetBeans

NetBeans es una herramienta que permite a los desarrolladores de software escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede ser utilizado en cualquier otro lenguaje de programación. Posee un número importante de módulos que amplían y facilitan su utilización. Es un producto libre, gratuito y sin restricciones de uso. La versión a utilizar en la presente investigación es 8.0.1, que ofrece un rendimiento superior para desarrolladores de aplicaciones web, proporcionando editores y herramientas integrales con otros lenguajes como HTML, JavaScript y CSS. Posee un editor y depurador de código php e incluye además, la capacidad de integración con el framework WordPress y con sistemas de control de versiones (31). La última versión estable de NetBeans es la 15.0.

1.4.6.2 PHPStorm

Es un IDE moderno y eficiente optimizado para trabajar con Symfony, Laravel, Drupal, WordPress, Zend Framework, Magento, Joomla!, CakePHP, Yii y otros marcos de trabajo. El editor "capta" su código y comprende su estructura en profundidad, además de ser compatible con todas las funcionalidades del lenguaje PHP para proyectos tanto nuevos como heredados. Proporciona la mejor finalización de código, refactorizaciones, prevención de errores sobre la marcha y más. Posee otras características avanzadas como:

- Inclusión de tecnologías Front-End: Aprovecha al máximo las últimas tecnologías de front-end, tales como HTML 5, CSS, Sass, Less, Stylus, CoffeeScript, TypeScript, Emmet y JavaScript, con disponibilidad de refactorizaciones, depuración y pruebas de unidad.
- Herramientas de desarrollo integradas: Permite realizar tareas de rutina directamente desde el IDE gracias a la integración con el Sistema de control de versiones, compatibilidad con im-

plementación remota, bases de datos/SQL, herramientas de línea de comando, Docker, Composer, REST Client y muchas otras herramientas.

- Compatibilidad completa con PHP y SQL

- Asistencia inteligente a la codificación: Verificar su código a medida que se escribe, analizando el proyecto entero. La compatibilidad con PHPDoc, posibilita escribir un código limpio y fácil de mantener (32).

1.6 Metodología de desarrollo de software

Una metodología de desarrollo de software se define como un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la construcción de un software (33). Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software, con el fin de hacerlo más predecible y eficiente. Tienen como meta aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Existen numerosas metodologías de desarrollo, que inciden en distintas dimensiones del proceso de desarrollo. En el Anexo 7 se muestra una tabla comparativa de la relación coste/tiempo de los dos enfoques de las metodologías (tradicionales y ágiles).

1.6.1. Metodologías tradicionales

Pressman plantea en 2013 que las metodologías de desarrollo tradicionales o clásicas son también llamados modelos de proceso prescriptivo, y fueron planteadas originalmente para poner orden en el caos del desarrollo de software que existía cuando se empezó a generar masivamente. La historia revela que estos modelos tradicionales que fueron presentados en la década de los 60, dieron cierta estructura útil al trabajo de la Ingeniería de software y constituyen un mapa razonablemente eficaz para los equipos de desarrollo. En las metodologías tradicionales se concibe al proyecto como uno solo de grandes dimensiones y estructura definida; el proceso es de manera secuencial, en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta (33).

1.6.2 Metodologías ágiles

Las metodologías ágiles presentan como principal particularidad la flexibilidad, los proyectos en desarrollo son subdivididos en proyectos más pequeños, incluye una comunicación constante con el usuario, son altamente colaborativos y es mucho más adaptable a los cambios. De hecho, el cambio de requerimientos por parte del cliente es una característica especial, así como también las entregas, revisión y retroalimentación constante. (Cadavid, Fernández Martínez, & Morales Vélez, 2013) (34).

Entre las más notables metodologías de desarrollo ágil, se encuentran:

+Scrum: es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Requisitos para utilizar SCRUM.

- Cultura de empresa basada en trabajo en equipo, delegación, creatividad y mejora constante.
- Compromiso del cliente en la dirección de los resultados del proyecto.
- Compromiso de la Dirección de la organización para resolver problemas frecuentes y realizar cambios organizativos.
- Compromiso conjunto y colaboración de los miembros del equipo.
- Relación entre proveedor y cliente basada en la colaboración y transparencia.
- Facilidad para realizar cambios en el proyecto.
- Tamaño de cada equipo entre 5 y 9 personas (con técnicas específicas de planificación y coordinación cuando varios equipos trabajan en el mismo proyecto).
- Equipo trabajando en un mismo espacio común para maximizar la comunicación.
- Dedicación del equipo a tiempo completo.
- Estabilidad de los miembros del equipo. (35).

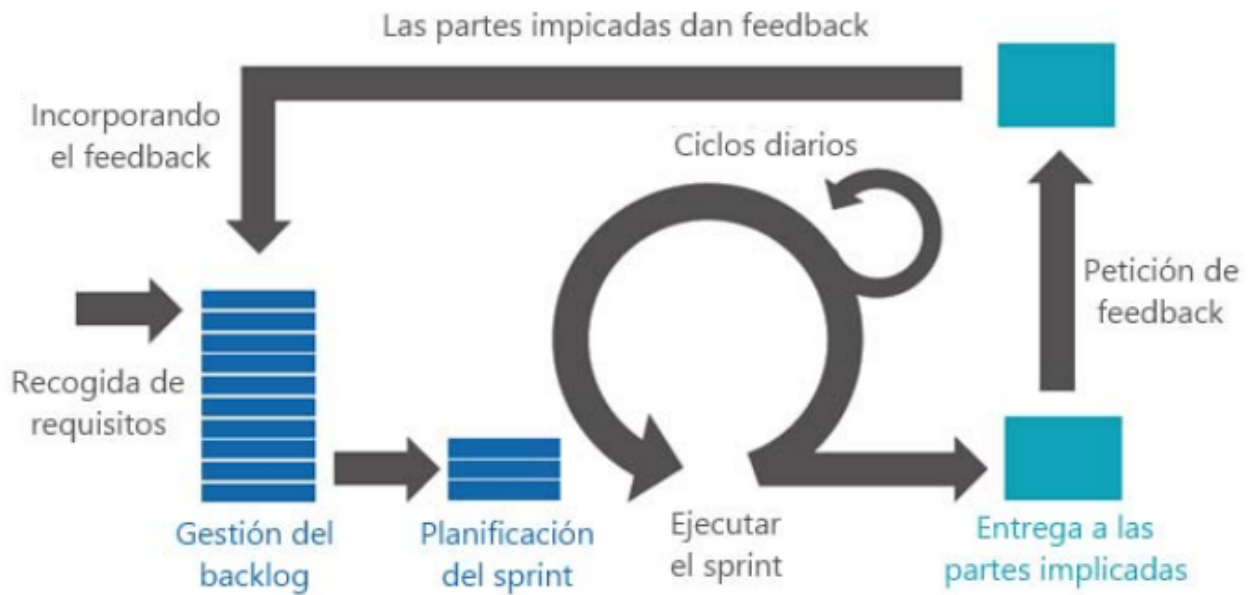


Figura 2: Proceso de SCRUM.

+Extreme Programming (XP): Es una metodología ágil basada en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Buenas prácticas de XP.

-Diseño simple: emplea el principio de utilizar el diseño más sencillo que consiga que todo funcione.

-Test: aplica una prueba tras cada cambio notable en el software.

-Estándar de codificación: el desarrollador debe seguir unas normas comunes para generar el código.

-Integración continua: a medida que se crean funcionalidades se ensamblan al sistema.

-Entregas frecuentes: desarrolla pequeñas versiones del sistema para que el usuario se familiarice con la versión final.



Figura 3: Capas de Extreme Programming (XP)

En la investigación actual se escoge la metodología XP por su viabilidad en proyectos pequeños que cuentan con pocos programadores, valoran la reducción de costos y se entregan al usuario final en un espacio de tiempo relativamente corto.

Conclusiones del capítulo

En este capítulo se realizó un estudio del estado del arte sobre herramientas existentes para el trabajo con un repositorio. También se estudiaron las metodologías de desarrollo de software. Se seleccionó la metodología, herramientas y tecnologías a utilizar para el desarrollo del repositorio, la cual mostrará la dinámica de la aplicación y permitirá que los usuarios puedan realizar sus diversas solicitudes a través de una interfaz amigable.

CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

Introducción

El siguiente capítulo plantea una descripción del sistema que se pretende desarrollar. Se definen los Requerimientos del mismo, las descripciones de Historias de Usuarios, y los Actores del Sistema. Este proceso de desarrollo se realiza aplicando la metodología ágil XP.

2.1 Modelo de Dominio

El modelo del dominio captura, comprende y describe los objetos más importantes dentro del contexto de un sistema (36). Se describe mediante diagramas de UML, especialmente mediante diagramas de clases. Los objetos del dominio o clases pueden obtenerse a partir de una especificación de requisitos o mediante la entrevista con los expertos del tema. Las clases que lo componen suelen aparecer en tres formas típicas (36):

- Objetos del negocio que representan los elementos que se manipulan en el negocio.
- Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento.
- Sucesos que ocurrirán o que han ocurrido.

El modelo de dominio que a continuación se describe constituye una representación gráfica de los principales conceptos que forman parte del repositorio para la gestión de recursos educativos.

Definición de las clases.

- Gestora: repositorio que almacena recursos educativos.
- Usuario: persona registrada en el sistema.
- Recurso educativo: recurso compuesto por contenidos educativos que pueden ser cursos completos, materiales para cursos, materiales multimedia, objetos de aprendizaje y cualquier otro recurso con fin educativo con una alta pertinencia y certificación de calidad.
- Autor: persona registrada en el sistema que posee los permisos para subir recursos al repositorio, a través de una herramienta de Autor y editar las propiedades del mismo.

-Administrador: persona registrada en el sistema que cuenta con todos los privilegios, es el encargado de todas las configuraciones del repositorio.

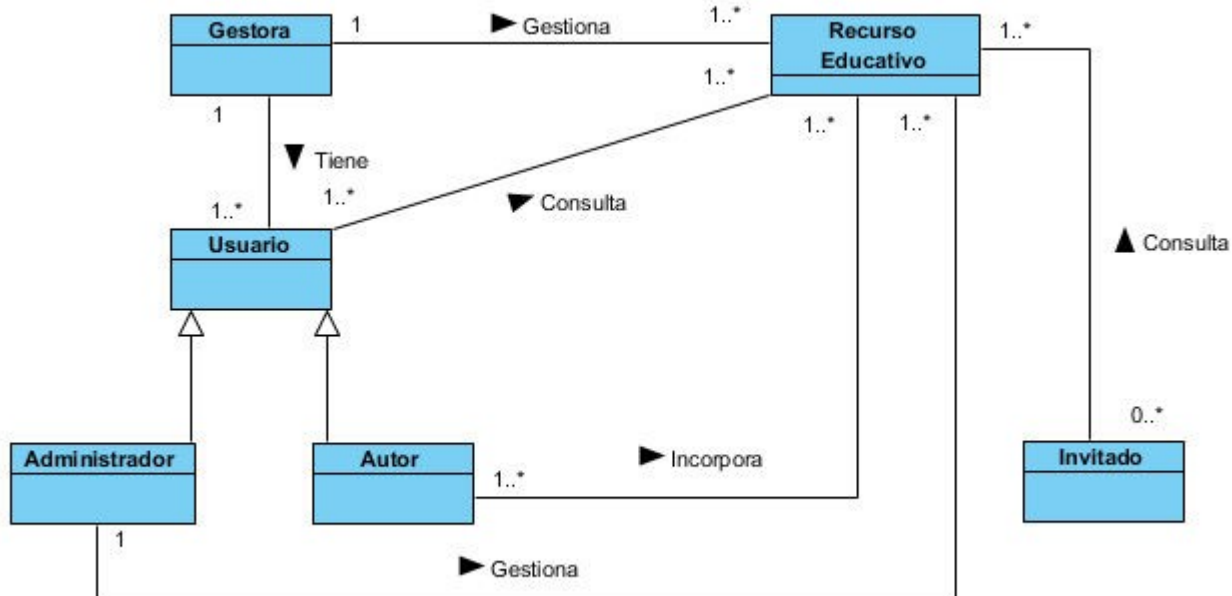


Figura 4: Modelo de dominio

2.2 Requerimientos

Una colección de requerimientos describe las características o atributos del sistema deseado. Se omite el cómo debe lograrse su implementación, ya que esto debe ser decidido en la etapa de diseño por los diseñadores.

2.2.1 Requisitos funcionales del sistema

Los requisitos funcionales especifican comportamientos particulares de un sistema. La base del requisito es la descripción del comportamiento requerido, que debe ser claro y legible. El comportamiento descrito puede venir de reglas de organización o de

negocio, o puede ser descubierto con sesiones de interacción con los usuarios, inversores, y otros expertos dentro de la organización.

El repositorio debe permitir:

- RF1: Autenticar. Permite a los usuarios del repositorio acceder a este mediante contraseña o crearse una cuenta en caso que no posean.
- RF2. Importar recurso educativo. Permite al autor y al administrador del repositorio importar un recurso educativo al repositorio.
- RF3. Exportar recurso educativo. Permite a los usuarios del repositorio incluyendo al invitado, exportar los recursos en formato de texto (.doc, .txt, .docx) a formato pdf.
- RF4. Descargar recurso educativo. Permite a los usuarios del repositorio incluyendo al invitado, descargar un recurso educativo.
- RF5. Visualizar recurso educativo. Permite a los usuarios del repositorio incluyendo al invitado, visualizar el contenido de un recurso educativo.
- RF6: Validar recurso educativo. Permite al administrador del repositorio validar si el recurso educativo estará disponible para su visualización.
- RF7. Eliminar recurso educativo. Permite al administrador del repositorio eliminar un recurso.
- RF8. Clasificar un recurso educativo. Permite al administrador del repositorio clasificar los recursos por temáticas.
- RF9. Denunciar un recurso educativo. Permite a los usuarios del repositorio denunciar un recurso educativo.
- RF10. Recomendar recurso educativo. Permite a los usuarios escribir una recomendación sobre un recurso educativo y compartir la recomendación y el enlace a este recurso en redes sociales.
- RF11. Compartir recurso educativo. Permite a los usuarios compartir en la red social Facebook recursos educativos en formato de imagen o enlaces a recursos educativos en formato de video.
- RF12. Adicionar recurso educativo a Favoritos. Permite a los usuarios adicionar recursos educativos a su lista de Favoritos.

- RF13. Listar recursos educativos Favoritos. Permite a los usuarios desplegar la lista de Favoritos.
- RF14. Adicionar comentario a un recurso educativo. Permite a los usuarios seleccionar un recurso e introducir un comentario sobre este.
- RF15. Editar comentarios de un recurso educativo. Permite a los usuarios editar sus comentarios sobre un recurso.
- RF16. Responder comentario de un recurso educativo. Permite a los usuarios introducir la valoración personal de un comentario a un recurso educativo.
- RF17. Denunciar comentario de un recurso educativo. Permite a los usuarios denunciar un comentario que se ha realizado a un recurso.
- RF18. Eliminar comentarios de un recurso educativo. Permite al administrador o al usuario que lo escribió, eliminar un comentario realizado a un recurso.
- RF19. Ver comentario de un recurso educativo. Permite a los usuarios del sistema incluyendo al invitado ver los comentarios de un recurso educativo.
- RF20. Ver ficha de un recurso educativo. Permite a los usuarios del sistema incluyendo al invitado ver la ficha de un recurso educativo.

2.2.2 Requisitos no funcionales

Característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo. (36)

RNF1. Apariencia o interfaz externa

La interfaz contiene imágenes ligeras para no demorar los tiempos de carga. El diseño de la interfaz es sencillo, con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones. Es formal, serio, con una navegación sugerente y con una gama cromática respetando pautas, todo esto teniendo en cuenta el fin con el que se desarrolla la aplicación.

RNF2. Usabilidad

La aplicación puede ser usada por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

RNF3. Rendimiento

La disponibilidad de trabajo en red contra el servidor es constante. Se garantiza que la respuesta a solicitudes de los usuarios, sea en un período de tiempo breve (de segundos) para evitar la acumulación de trabajo por parte de los responsables. La aplicación deberá ser lo más estable y confiable posible.

RNF4. Soporte

Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra, para lograr esto la aplicación constará con una documentación apropiada para agilizar su mantenimiento y configuración, de forma tal que el tiempo de mantenimiento sea mínimo en caso de que sea necesario cambiar el equipo de desarrollo.

RNF5. Portabilidad

El producto es multiplataforma, corre sobre una plataforma web, los servicios en PHP 5 y su sistema de bases de datos en MySQL. Es de fácil mantenimiento.

RNF6. Seguridad

El sistema se encarga de controlar los diferentes accesos y funcionalidades, identifica al usuario antes de que pueda realizar cualquier acción sobre el sistema. La información será almacenada en bases de datos, dejando registro de toda operación realizada. Garantiza que la información sea vista únicamente por quien tiene derecho a verla. Se usan mecanismos de encriptación (MD5) de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas. Se hacen validaciones de la información tanto en el cliente como en el servidor, no obstante, los usuarios acceden de manera rápida y operativa a la aplicación sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

RNF7. Restricciones en el diseño y la implementación

Es una aplicación web desarrollada con la tecnología para creación de páginas web dinámicas PHP 7.4.1 y base de datos en MySQL 8.0.16.

RNF8. Legales

El producto es una aplicación para la gestión de los recursos educativos, se rige por las normas establecidas por la DIN. Las tecnologías escogidas para el desarrollo de la aplicación, están basadas en la licencia GNU/GPL y la herramienta Visual Paradigm que cuenta con licencia.

RNF9. Confiabilidad

La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores. En caso de alguna dificultad con el funcionamiento del sistema, el tiempo medio de reparación debe ser menor de 24 horas.

La aplicación estará funcionando las 24 horas del día; de esta forma es posible que los usuarios tengan acceso (según sus permisos) en todo momento a la información solicitada.

RNF10. Software

En los clientes:

- Sistema operativo Linux, Windows 7 o superior, Android 4.1 o superior.
- Cualquier navegador Web.

En los servidores:

- Sistema Operativo Linux o Unix.
- Sistema Operativo Windows 98 o superior.
- Servidor Web Apache 2.0
- Servidor de Bases de Datos

MySQL.

- PHP 7 con servicios web.
- Servidor de Aplicaciones Apache.

Hardware

Para el Servidor:

-Se necesitan como requerimientos mínimos una PC con procesador Intel Core i3 o superior. --Microprocesador con una velocidad de 3.0 GHz o superior.

-Memoria RAM de 2 GB o superior.

-HDD de 80 GB o superior.

Para el cliente:

-Se necesitan como requerimientos mínimos una PC con procesador Pentium IV o superior.

-Microprocesador con una velocidad a 1 GHz o superior.

-Memoria RAM de 1 GB o superior. HDD de 20 GB o superior.

RnF11. Lenguaje de Programación

El lenguaje de programación utilizado es PHP en su versión 7.4.1.

RnF12. Entorno de Desarrollo Integrado.

Se utilizaron como Entornos de Desarrollo Integrado Netbeans 8.0.1 y PHP Storm 2017.1.3.

RnF13. Framework de desarrollo.

El framework de desarrollo empleado es WordPress 6.1.1.

RnF14. Estilo arquitectónico.

El estilo arquitectónico utilizado es Modelo Vista Controlador.

2.3 Identificación de los actores y casos de uso del sistema

Los actores representan personas u otro software que interactúan con el sistema. Estos se comunican con el mismo a través del envío y recepción de mensajes según este lleve a cabo los casos de uso. Con el objetivo de especificar privilegios, nivel de acceso a la información y funciones en el módulo de gestión de recursos, se identifican los siguientes actores:

Tabla 2. Descripción de los actores del sistema.

Actores	Descripción
Invitado	Persona que no posee un nombre de usuario en el sistema y que por tanto solo tendrá acceso a sus funciones básicas.
Usuario	Persona autenticada en el sistema con nombre y contraseña.
Autor	Persona registrada en el sistema que posee los privilegios suficientes para subir recursos al repositorio.
Administrador	Persona registrada en el sistema que cuenta con todos los privilegios.

2.3.1 Casos de Uso del Sistema

CU_1: Autenticar.

CU_2: Importar recurso educativo.

CU_3: Exportar recurso educativo.

- CU_4:** Descargar recurso educativo.
- CU_5:** Visualizar recurso educativo.
- CU_6:** Validar recurso educativo.
- CU_7:** Eliminar recurso educativo.
- CU_8:** Clasificar recurso educativo.
- CU_9:** Denunciar recurso educativo.
- CU_10:** Recomendar recurso educativo.
- CU_11:** Compartir recurso educativo.
- CU_12:** Adicionar recurso educativo.
- CU_13:** Listar recursos educativos favoritos.
- CU_14:** Adicionar comentario a recurso educativo.
- CU_15:** Editar comentario de recurso educativo.
- CU_16:** Responder comentario de recurso educativo.
- CU_17:** Denunciar comentario de recurso educativo.
- CU_18:** Eliminar comentario de recurso educativo.
- CU_19:** Ver comentario de recurso educativo.
- CU_20:** Ver ficha de recurso educativo.

2.4 Historias de Usuario

Las Historias de Usuario son la técnica utilizada en XP para especificar los requisitos del software. Estas describen brevemente las características que el sistema debe poseer. Cada Historia de Usuario debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla. Estas se usan para estimar el tiempo y el Plan de entregas los cuales dirigen la creación de las pruebas de aceptación. Se caracterizan por ser independientes una de otras, negociables, valoradas por los clientes o usuarios, estimables y verificables. Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores, utilizando como medida el punto.

Tabla 3. Historia de Usuario Autenticar.

Número: 01	Nombre de Historia de Usuario: Autenticar.
Modificación de Historia de Usuario Número: ninguna.	
Usuario:	Iteración Asignada : 1
Prioridad: Alta.	Puntos Estimados: 1 día.
Riesgo en Desarrollo: Alto.	Puntos Reales: 1
Descripción: El usuario inserta sus datos (usuario y contraseña o cuenta de Facebook), se verifica en la base de datos que sean correctos y otorga el acceso al sistema con los permisos correspondientes a dicho usuario. Si no está registrado debe crearse una cuenta.	
Observaciones: Debe notificarse al usuario si su contraseña no cumple los requisitos de seguridad. (8 o más caracteres y presencia de mayúsculas o signos especiales).	

Tabla 4. Historia de Usuario: Importar recurso.

Número: 02	Nombre de Historia de Usuario: Importar re-
Modificación de Historia de Usuario Número: ninguna.	
Usuario:	Iteración Asignada: 2
Prioridad: Alta	Puntos Estimados: 1 semana.
Riesgo en Desarrollo: Alto.	Puntos Reales: 1
Descripción: El Autor sube un recurso educativo al repositorio, le asigna nombre e	
Observaciones: El recurso no estará disponible para su visualización hasta que el administrador no lo valide.	

Tabla 5. Historia de Usuario. Exportar Recurso.

Número: 02	Nombre de Historia de Usuario: Exportar recurso
Modificación de Historia de Usuario Número: ninguna.	
Usuario:	Iteración Asignada: 3
Prioridad: Alta	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Alta.	Puntos Reales: .1
Descripción: El usuario convierte un recurso en formato .doc a formato .pdf y lo vi-	
Observaciones: Lanza una excepción si se exporta un recurso en formato incom- patible.	

Tabla 6. Historia de Usuario. Descargar Recurso.

Número: 04	Nombre de Historia de Usuario: Descargar
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 4
Prioridad: Alta	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Alta.	Puntos Reales: 1
Descripción: El usuario selecciona y guarda el recurso educativo en el dispositivo.	
Observaciones: La aplicación debe permitir al usuario seleccionar la ubicación don- de desea descargar el recurso.	

Tabla 7: Historia de Usuario. Visualizar Recurso.

Número: 05	Nombre de Historia de Usuario: Visualizar Re-
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 5

Prioridad: Alta	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Alta.	Puntos Reales: 1
Descripción: El usuario selecciona el recurso y este se visualiza como texto o mul-	
Observaciones:	

Tabla 8: Historia de Usuario. Validar Recurso.

Número: 06	Nombre de Historia de Usuario: Validar Re-
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 6
Prioridad: Alta	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Alta.	Puntos Reales: 1
Descripción: El administrador revisa los recursos no validados, comprueba que cumplen las normas y los hace visibles al resto de usuarios del repositorio.	
Observaciones: La aplicación debe permitir al administrador eliminar recursos que no cumplan con los requisitos.	

Tabla 9: Historia de Usuario. Eliminar Recurso.

Número: 07	Nombre de Historia de Usuario: Eliminar Re-
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 7
Prioridad: Alta	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Alta.	Puntos Reales: 1
Descripción: El administrador selecciona un recurso y lo elimina del repositorio.	
Observaciones:	

Tabla 10: Historia de Usuario: Clasificar Recurso.

Número: 08	Nombre de Historia de Usuario: Clasificar Re-
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 8
Prioridad: Alta	Puntos Estimados: 1 semana.

Riesgo en desarrollo: Alta.	Puntos Reales: 1
Descripción: El administrador selecciona un recurso y lo clasifica según su temática.	
Observaciones: La aplicación debe mostrar al administrador un menú desplegable para la selección de temáticas.	

Tabla 11: Historia de Usuario: Denunciar Recurso.

Número: 09	Nombre de Historia de Usuario: Denunciar
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 9
Prioridad: Media	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario denuncia un recurso que no cumpla con los requisitos o contenga material inapropiado.	
Observaciones:	

Tabla 12: Historia de Usuario. Recomendar Recurso.

Número: 10	Nombre de Historia de Usuario: Recomendar Recurso
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 10
Prioridad: Media	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario selecciona un recurso y comparte una recomendación de este en redes sociales.	
Observaciones: El usuario debe escribir una recomendación sobre el recurso. Se genera un enlace para acceder a este.	

Tabla 13: Historia de Usuario. Compartir Recurso.

Número: 11	Nombre de Historia de Usuario: Compartir
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 10
Prioridad: Media	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Me-	Puntos Reales: 1
Descripción: El usuario selecciona un recurso y comparte en redes sociales.	
Observaciones: Sólo se pueden compartir recursos en formato de foto. En caso que el recurso sea un video se comparte un enlace a este.	

Tabla 14: Historia de Usuario. Adicionar Recurso a Favoritos.

Número: 12	Nombre de Historia de Usuario: Adicionar Re-Favoritos.
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 11
Prioridad: Media	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario selecciona un recurso y lo coloca en su lista de favoritos.	
Observaciones:	

Tabla 15: Historia de Usuario. Listar Recursos Favoritos.

Número: 13	Nombre de Historia de Usuario: Favoritos.
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 13
Prioridad: Media	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Me-	Puntos Reales: 1
Descripción: El usuario selecciona el menú Favoritos.	

Observaciones: Al seleccionar la opción Favoritos, la aplicación devuelve una lista con los recursos que se encuentran en esta.

Tabla 16: Adicionar Comentario.

Número: 14	Nombre de Historia de Usuario: Adicionar Comentario
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 14
Prioridad: Bajo	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario selecciona el recurso y escribe un comentario sobre este.	
Observaciones:	

Tabla 17: Editar comentario.

Número: 15	Nombre de Historia de Usuario: Editar Co-
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 15
Prioridad: Bajo	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario modifica un comentario escrito anteriormente.	
Observaciones: Los comentarios solo pueden ser editados por el usuario que los	

Tabla 18: Responder Comentario.

Número: 16	Nombre de Historia de Usuario: Responder Comentario
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 16
Prioridad: Bajo	Puntos Estimados: 1 semana.

Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario responde un comentario escrito anteriormente.	
Observaciones:	

Tabla 19: Denunciar Comentario.

Número: 17	Nombre de Historia de Usuario: Denunciar Comentario
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 17
Prioridad: Bajo	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario denuncia un comentario escrito anteriormente.	
Observaciones: La aplicación debe notificar al administrador que un recurso ha sido denunciado.	

Tabla 20: Eliminar Comentario.

Número: 18	Nombre de Historia de Usuario: Eliminar Comentario
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 18
Prioridad: Bajo	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario elimina un comentario escrito anteriormente.	
Observaciones: El usuario solo puede eliminar sus propios comentarios. El administrador puede eliminar comentarios de todos los usuarios.	

Tabla 21: Ver Comentario.

Número: 19	Nombre de Historia de Usuario: Ver Comenta-
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 19
Prioridad: Bajo	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario visualiza la lista de comentarios de un recurso.	
Observaciones:	

Tabla 22: Ver Ficha de un Recurso.

Número: 20	Nombre de Historia de Usuario: Ver Ficha
Modificación de Historia de Usuario Número: ninguna	
Usuario:	Iteración Asignada: 20
Prioridad: Bajo	Puntos Estimados: 1 semana.
Riesgo en desarrollo: Bajo.	Puntos Reales: 1
Descripción: El usuario despliega la ficha del recurso educativo.	
Observaciones: La aplicación muestra una pequeña ficha del recurso seleccionado.	

2.5 Plan de Entregas.

El Plan de Entregas tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha Historia. Como resultado de la priorización de Historias se llegó a la siguiente planificación.

Tabla 23. Plan de Entregas.

Entregas	Iteración	Tiempo Estimado (días)
Autenticar.		1
Importar recurso educativo.		3
Exportar recurso educativo.		1
Descargar recurso educativo.		1
Visualizar recurso educativo.		3
Validar recurso educativo.		1
Eliminar recurso educativo.		1
Clasificar recurso educativo.		1
Denunciar recurso educativo.		1
Recomendar recurso educativo a otro usuario.		1
Compartir recurso educativo en redes sociales.		1
Adicionar recurso educativo a favoritos.		1
Listar recursos educativos favoritos.		1
Adicionar comentario a un recurso educativo.		1
Editar comentario de un recurso educativo.		1
Responder comentario de un recurso educativo.		1
Denunciar comentario de un recurso educativo.		1
Eliminar comentario de un recurso educativo.		1
Ver comentarios de un recurso educativo.		1
Ver ficha de un recurso educativo.		1

2.6 Patrones de diseño.

Los patrones del diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. El diseño de la propuesta de solución se realizó aplicando los patrones GRASP (por sus siglas en inglés, General Responsibility Assignment Software Patterns), los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos y GOF (por sus siglas en inglés, Gang of Four), los cuales definen estructuras de clases que tratan una situación en particular. (37)

Los patrones GRASP utilizados fueron:

-Experto (Expert): Se utiliza para la asignación de responsabilidades a las clases de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica, o sea, el controlador frontal le asigna responsabilidades a las clases controladoras para realizar acciones de acuerdo a las peticiones que recibe.

-Creador (Creator): Se utiliza para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la

información necesaria para ello.

-Alta cohesión (High Cohesion): Se utiliza para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas. Se diseñaron las clases de forma tal que contengan las mínimas responsabilidades necesarias y colaboren con otras para llevar a cabo una tarea.

-Bajo Acoplamiento (Low Coupling): El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases.

-Controlador: Asignar la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

Los patrones GOF utilizados fueron:

-Observador (Observer): Es el encargado de definir una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y se actualicen automáticamente todos los objetos que dependen de él.

-Decorador (Decorator): WordPress presenta una plantilla global, en la que convergen todos los elementos comunes a cada una de las páginas del sistema en construcción. Este fichero se complementa con las plantillas, decorándolas y obteniéndose la interfaz final que será mostrada al usuario.

-Controlador frontal (Front Controller): Es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse.

Conclusiones del capítulo.

La identificación de los requisitos tanto funcionales como no funcionales, permitió elaborar un modelo del sistema que sirvió de acuerdo entre los desarrolladores y el cliente, constituyendo además la entrada fundamental para las fases posteriores del desarrollo. Las descripciones de los CU permitieron conocer el funcionamiento general del sistema, estableciendo para ellos un conjunto de pasos lógicos, flujos de información y respuestas del sistema al usuario, logrando obtener una comprensión general del negocio.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Introducción

El siguiente capítulo plantea una descripción del estándar de codificación empleado en el código del repositorio. Se define la fase de implementación y se elabora el Diagrama de Despliegue. Finalmente se realizan pruebas de validación para verificar el funcionamiento correcto del software.

3.1 Estándar de codificación.

Los estándares de codificación son las reglas que comprenden todo lo relacionado al código fuente de cualquier software, por lo que resulta importante establecer su utilización para asegurar que durante el desarrollo del mismo cualquier programador pueda entender el código y trabajar de forma coordinada.

Para el desarrollo en cuestión se utiliza el estándar de codificación básica PSR-1, que consta de una serie de reglas y expresiones para dar formato al código en PHP. (38)

3.1.1 General.

- Etiquetas PHP: El código PHP debe utilizar las etiquetas largas (<?php ?>) o las etiquetas cortas (<?= ?>) para imprimir salida de información.
- Codificación de caracteres: El código PHP debe utilizar codificación UTF-8 sin BOM.
- Archivos: Todos los archivos PHP deben terminar con una línea en blanco. La etiqueta de cierre (?>) debe ser omitida en los archivos que sólo contengan código PHP.
- Líneas: Las líneas no deberían ser más largas de 80 caracteres.
- Palabras claves y constantes: Las palabras claves de PHP deben estar en minúsculas. Las constantes de PHP, *null*, *false* y *true* deben estar en minúsculas.

3.1.2 Espacio de nombres y declaraciones.

- Cuando esté presente, debe existir una línea en blanco después de la declaración del *namespace*.
- Cuando estén presentes, todas las declaraciones *use* deben ir después de la declaración del *namespace*.
- Debe haber un *use* por declaración.

-Debe escribirse una línea en blanco después del bloque de declaraciones *use*.

3.1.3 Clases, propiedades y métodos.

-Las palabras clave *extends* e *implements* deben declararse en la misma línea del nombre de la clase.

-La llave de apertura de la clase debe ir en la línea siguiente; la llave de cierre debe ir en la línea siguiente al cuerpo de la clase.

-La visibilidad debe ser declarada en todas las propiedades.

-La palabra clave *var* no debe ser usada para declarar una propiedad.

-No debe declararse más de una propiedad por sentencia.

-Los nombres de métodos no deben estar declarados con un espacio después del nombre del método.

3.1.4 Estructuras de Control.

- Debe haber un espacio después de una palabra clave de estructura de control.

- No debe haber espacios después del paréntesis de apertura.

- No debe haber espacios antes del paréntesis de cierre.

- Debe haber un espacio entre paréntesis de cierre y la llave de apertura.

- La llave de cierre debe estar en la línea siguiente al final del cuerpo.

3.2 Implementación.

Etapa del ciclo de vida de un sistema de información en la que se procede a la instalación del hardware y el software elegidos. Esta fase del proceso de desarrollo de software se nutre de los resultados obtenidos en la fase de diseño. En el modelo de implementación está contenido el diagrama de despliegue, el cual describe la dependencia entre los nodos físicos en los que funcionará el sistema.

3.2.1 Diagrama de despliegue.

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. (39) El diagrama de despliegue del repositorio, está compuesto por:

-Dispositivo_Cliente: Representa una computadora o teléfono móvil mediante la cual el usuario accede a la aplicación.

-Servidor_Web (Apache): Representa el servidor que se comunica con el Dispositivo_Cliente mediante el protocolo HTTPS y además realiza peticiones al servidor de Bases de Datos mediante el protocolo TCP/IP.

-Servidor_Base_Datos: Representa el servidor que aloja el Sistema Gestor der Bases de Datos que responde a las peticiones realizadas por la aplicación.

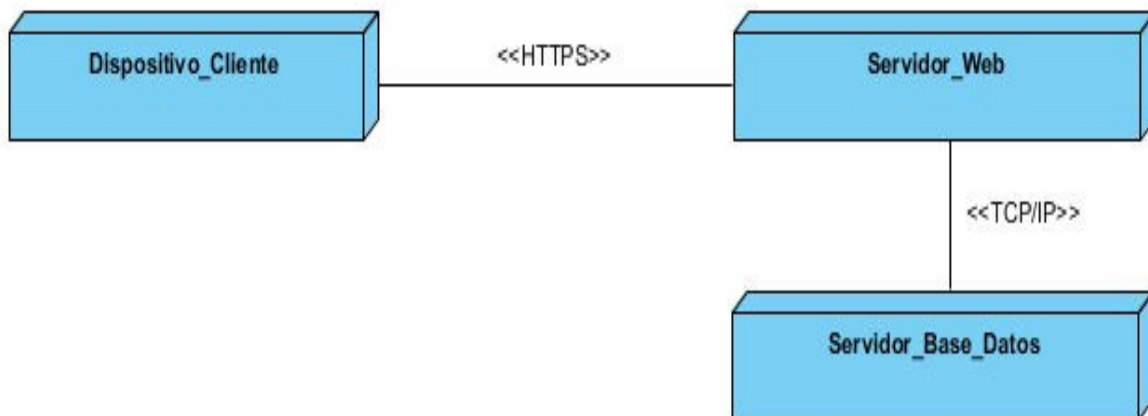


Figura 5: Diagrama de despliegue

3.3 Pruebas.

Con el objetivo de controlar que los productos cumplan con los requisitos funcionales y determinar la calidad de los mismos, resulta importante realizar pruebas. Estas pruebas constituyen una actividad en la cual un sistema o unos de sus componentes se ejecutan en circunstancias previamente especificadas, donde los resultados se observan, se registran y se realiza una evaluación de algún aspecto del sistema. Dentro de los procedimientos para la realización de pruebas al software, se encuentran las pruebas de caja blanca y las pruebas de caja negra.

3.3.1 Pruebas de caja negra.

Es un método de prueba de software que analiza la funcionalidad de un software sin saber mucho sobre la estructura interna del elemento que se está probando y compara el valor de entrada con el valor de salida. El enfoque principal de esta prueba es la funcionalidad del sistema en su conjunto. También se utiliza el término pruebas de comportamiento. Se encargan de encontrar errores de: funciones incorrectas o faltantes, de interfaz, en estructuras de datos o el acceso a base de datos externas, errores de comportamiento y errores de inicialización. Esta prueba se produce durante todo el ciclo de vida de prueba y desarrollo de software, es decir, en las etapas de prueba de unidad, integración, sistema, aceptación y regresión. Existen varios métodos para realizar las pruebas de caja negra, entre los que se encuentran:

- Método de partición equivalente: divide el dominio de entradas de un programa en clases de datos, a partir de las cuales pueden derivarse casos de prueba.
- Método del análisis de valores límites: se encarga del análisis de los valores límites en los campo de entrada.
- Método de prueba de la tabla ortogonal: se aplica en problemas donde el dominio de entrada es relativamente pequeño, el método resulta útil para encontrar fallas de región (categoría de error asociado con los defectos de la lógica en un componente de software). (33)

A continuación se muestra un ejemplo de caso de prueba aplicados al requisito Autenticar:

Tabla 24: Descripción del Caso de Prueba del requisito Autenticar.

No	Nombre de 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.

3.3.2 Pruebas de caja blanca.

Las pruebas de caja blanca, también conocidas como pruebas de caja transparente o pruebas estructurales, 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. (33)

3.3.3 Pruebas de usabilidad.

Es un método para evaluar la experiencia del usuario de un producto o sitio web. Al probar la usabilidad con un grupo representativo de usuarios o clientes, el evaluador sabe si el usuario real puede manejar el software de manera fácil e intuitiva. Esto permite descubrir cualquier problema con el producto a partir de la experiencia del usuario, decidir cómo solucionarlo y, en última instancia, determinar si el producto es realmente útil. Durante un estudio de usabilidad, el moderador pide a los participantes en sus sesiones individuales que completen una serie de tareas, mientras que el resto del equipo observa y toma notas. Al observar el manejo

de la aplicación en tiempo real puede constatar que la aplicación cumple con los estándares de calidad. (33)

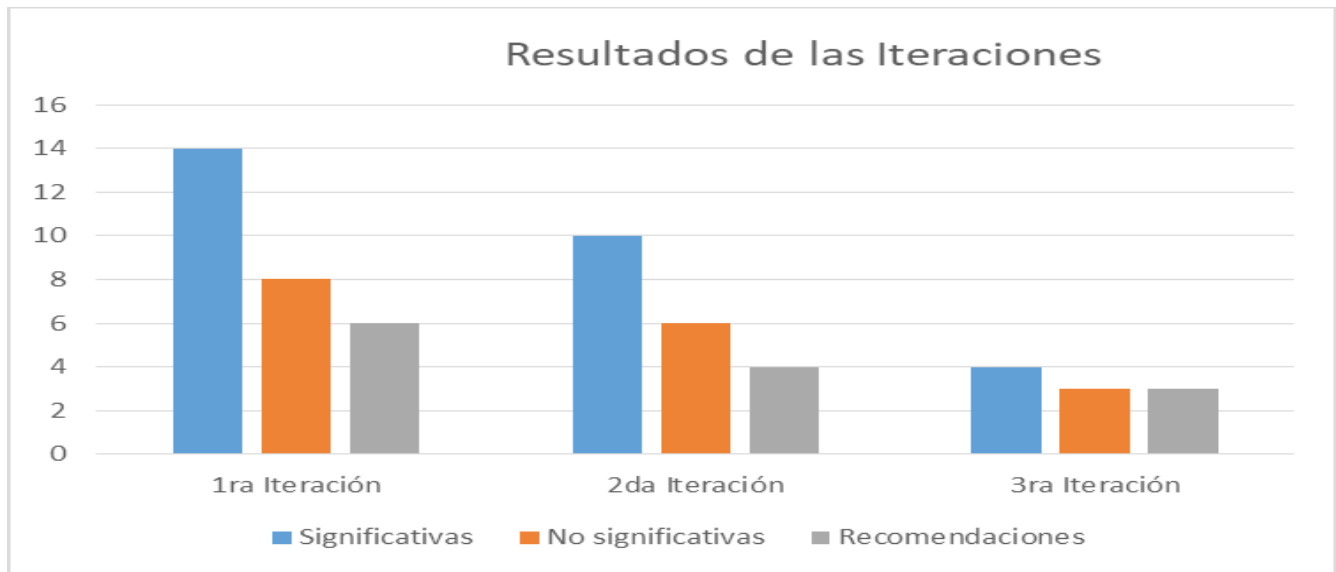


Figura 6: Resultados de las Iteraciones.

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 FINALES

- El estudio de repositorios de recursos educativos 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.

RECOMENDACIONES

A partir de la investigación realizada se recomienda:

- Capacitar al personal en el trabajo con repositorios.
- Designar como administradores del repositorio a personas con conocimientos en el área de gestión de recursos educativos.
- Aumentar en el futuro el espacio de almacenamiento destinado al repositorio.

REFERENCIAS BIBLIOGRÁFICAS

1. Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. (23 de mayo de 2013). Recuperado de <http://www.unesco.org/new/es/communication-and-information/access-to-knowledge/open-educational-resources>.
2. Atkins, Daniel (2007) A Review of the Open Educational Resources (OER) Movement: Achievements, Challenges, and New Opportunities.
3. Maurizi, María Rosa. (Julio de 2012) Que son los Recursos Educativos Abiertos. Recuperado de <http://www.slideshare.net/mariarosamaurizi/que-son-los-recursos-educativos-abiertos>.
4. Zapata, Marta. Programa Integración de Tecnologías a la Docencia. (Septiembre de 2012). <http://aprendeonline.udea.edu.co/boa/contenidos/>.
5. Recursos Educativos Abiertos (REA) gratis para todos (Julio de 2018). Recuperado de <http://www.educaciontrespuntocero.com/recursos-educativos-gratis-para-todos>.
6. López Guzmán, C. (2005). *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno E-learning*. Recuperado de <https://gredos.usal.es/handle/10366/56649>.
7. Fuente, Gema Bueno de la. (2010) Modelo de repositorio institucional de contenido educativo (RICE):
La gestión de materiales digitales de docencia y aprendizaje en la biblioteca universitaria.
8. *The benefits of adding SCORM to your product or platform*. (s. f.). SCORM.com. Recuperado 18 de noviembre de 2022, de <https://scorm.com/scorm-explained/business-of-scorm/benefits-of-scorm/>.
9. *Repositorio | Definición | Diccionario de la lengua española | RAE - ASALE*. (s. f.). Recuperado 18 de noviembre de 2022, de <https://dle.rae.es/repositorio?m=form>.
10. Investigación, S. de A. a la D. e. (s. f.). *Biblioguías UCM: Acceso abierto: Repositorios*. Recuperado 18 de noviembre de 2022, de <https://biblioguias.ucm.es/acceso-abierto/repositorios>.
11. Méndez, Anala. (s. f.). *Beneficios de un Repositorio de Recursos Educativos Abiertos*.
12. Cañizarez González, R. (2013). *REPOSITORIO DE RECURSOS EDUCATIVOS PARA LAS INSTITUCIONES DE EDUCACIÓN SUPERIOR*. [Thesis]. <https://repositorio.uci.cu/jspui/handle/ident/7913>
13. *Agrega—Portada*. (s. f.). Recuperado 10 de noviembre de 2022, de <http://agrega.educacion.es/visualizadorcontenidos2/AcercaDeAgrega/AcercaDeAgrega.do>
14. SSRN. (s. f.). Recuperado 10 de noviembre de 2022, de <https://blogs.sld.cu/marionod/2020/04/21/ssrn/>
15. Ruipérez, M. E. (s. f.). *Biblioguías: Sherpa Romeo (política de copyright de revistas): Inicio*. Recuperado 10 de noviembre de 2022, de <https://biblioguias.upct.es/sherpa-romeo/inicio>.
16. *Repositorio de Recursos Educativos—About Repositorio de Recursos Educativos*. (s. f.). Recuperado 10 de noviembre de 2022, de <http://recursosuvs.sld.cu/index.php?P=About>.
17. ASALE, R.-, & RAE. (s. f.). *Gestión | Diccionario de la lengua española*. «Diccionario de la lengua española» - Edición del Tricentenario. Recuperado 9 de noviembre de 2022, de <https://dle.rae.es/gestión>.

18. *Qué es la gestión de recursos y cómo comenzar • Asana*. (s. f.). Recuperado 19 de noviembre de 2022, de <https://asana.com/es/resources/resource-management-plan>.
19. Woodman, L. (1985) *Information Management in Large Organizations*. London: ASLIB.
20. *Aplicaciones Web 2.0—Google Docs*. (s. f.). Recuperado 11 de noviembre de 2022, de <https://www.digitaliapublishing.com/a/14467/aplicaciones-web-2.0---google-docs>.
21. Definición de aplicación web. (2015, enero 20). *Miguel Ángel López WEB*. <http://mialtoweb.es/definicion-de-aplicacion-web/>.
22. Lutz, Mark (2010). *Learning Python*. O'Reilly Media, Inc.
23. Vázquez Mariño, Carlos (2008). *Programación en PHP5*.
24. Gauchat, Juan Diego (2012) *El gran libro de HTML, CSS3 y Javascript*. Ed. Marcombo.
25. *IBM Documentation*. (s.f.). Recuperado 10 de noviembre de 2022, de <https://www.ibm.com/docs/es/rational-soft-arch/9.6.1?topic=page-asynchronous-javascript-xml-ajax-overview>.
26. Sommerville, I. (2005) *Ingeniería del Software*. Séptima edición. Madrid: Pearson Addison Wesley.
27. *Qué es MySQL: Características y ventajas | OpenWebinars*. (s. f.). Recuperado 10 de noviembre de 2022, de <https://openwebinars.net/blog/que-es-mysql/>.
28. *¿Qué es Servidor de aplicaciones y proveedor de servicios de aplicaciones? - Definición en Whats.com*. (s. f.). ComputerWeekly.es. Recuperado 10 de noviembre de 2022, de <https://www.computerweekly.com/es/definicion/Servidor-de-aplicaciones-y-proveedor-de-servicios-de-aplicaciones>.
29. *About the Apache HTTP Server Project—The Apache HTTP Server Project*. (s. f.). Recuperado 10 de noviembre de 2022, de https://httpd.apache.org/ABOUT_APACHE.html.
30. Pursell, S. (s. f.). *WordPress: Ventajas y desventajas de este CMS*. Recuperado 10 de noviembre de 2022, de <https://blog.hubspot.es/website/wordpress-ventajas-desventajas>.
31. *NetBeans IDE | Oracle México*. (s. f.). Recuperado 11 de noviembre de 2022, de <https://www.oracle.com/mx/tools/technologies/netbeans-ide.html>.
32. *PhpStorm: IDE de PHP y editor de código de JetBrains*. (s. f.). Recuperado 11 de noviembre de 2022, de <https://www.jetbrains.com/es-es/phpstorm/>.
33. Pressman, R. S. (2013). *Ingeniería de Software un enfoque práctico*. Journal of Chemical Information and Modeling (7ma ed., Vol. 53). Mc Graw Hill.
34. Cadavid, A. N., Fernández Martínez, J. D., & Morales Vélez, J. (2013). *Revisión de metodologías ágiles para el desarrollo de software*.
35. Maida, Esteban Gabriel, Pacienza Julian (2015). *Metodologías de Desarrollo de Software* (Tesis). Universidad Católica Argentina.
36. JACOBSON, Ivar, BOOCH, Grady and RUMBAUGH, James (2000). *El Proceso Unificado de Desarrollo de Software*. Madrid.
37. *Resumen Principios SOLID, Patrones GRASP y GoF - StuDocu*. (s. f.). Recuperado 10 de noviembre de 2022, de <https://www.studocu.com/latam/document/universidad-ort-uruguay/disenio-de-aplicaciones-1/resumen-principios-solid-patrones-grasp-y-gof/9259537>
38. *PHP - Estándar de programación Coppel para PHP*. (s. f.). Recuperado 10 de noviembre de 2022, de <http://coppeldev.github.io/php/standards/coppel.html>
39. *UML 2.1.2*. (s. f.). Recuperado 11 de noviembre de 2022, de <http://www.omg.org/spec/UML/2.1.2/>

ANEXOS

Anexo 1: Captura de pantalla de Agrega



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 [AgriSciRN](#)

Anthropology & Archaeology [AARN](#)

Architecture [ArchRN](#)

Biochemistry [BiochemRN](#)

Biology [BioRN](#)

Business History & Nonprofit Organizations [BHNP](#)

Chemistry [ChemRN](#)

Classics [CRN](#)

Cognitive Science [CSN](#)



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 and after the conference. Creating a permanent archive of the proceedings presented at an event extends the life of the research, expands exposure for the event and researchers, and ensures the

Anexo 3: Captura de Pantalla de Sherpa Romeo

The screenshot shows the Sherpa Romeo website. 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 with links: About, Search, TJ List, Statistics, Help, Support Us, Contact, and Admin.

The main content area starts with a "Welcome to Sherpa Romeo" heading, followed by 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 is a search prompt: "Enter a journal title or issn, or a publisher name below:". There are two search input fields: "Journal Title or ISSN" and "Publisher Name", each with a "Search" button. Below the search fields are two buttons: "Browse by Country" and "Browse by Publisher".

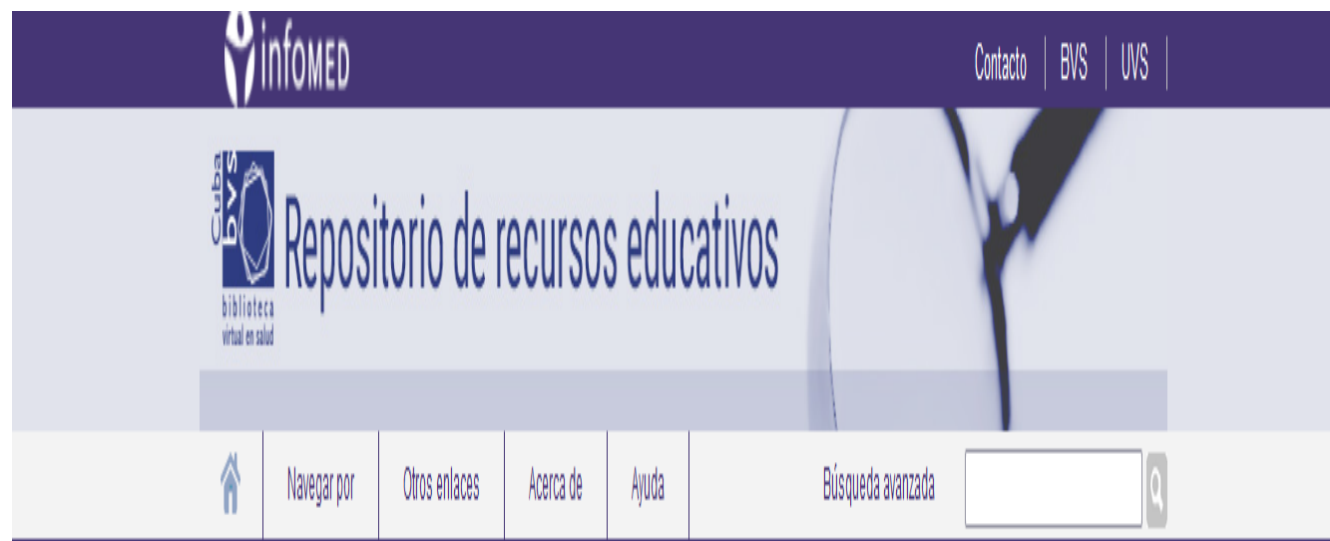
The lower section features three service cards:

- 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".

At the bottom, there is a footer section with "Useful links" (Open access services from Jisc, Guide to managing open access costs, Open access guides by Jisc), "Contact" (help@jisc.ac.uk), and "Accessibility" (Accessibility Statement). Below this is the Jisc logo and a description: "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." with a link "Find out more at jisc.ac.uk". To the right are links for "Digital resources" (Maps and spatial data, Library support) and "Learning and research resources" (Open access (OA)).

The footer also includes a Creative Commons license icon (CC BY-NC-ND 4.0) and the text: "This work is licensed under the [CC BY-NC-ND 4.0](#) About using our content".

Anexo 4: Sección de la pantalla de inicio del Repositorio de Recursos Educativos de la Universidad Virtual de la Salud.



Repositorio de la Universidad Virtual de Salud

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.

CUENTA DE USUARIO:

Usuario:

Contraseña:

Registrarse

¿Olvidó su contraseña?

Recursos

Total: 697 Recursos

Anexo 5: 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 6: Sección superior de la página de inicio del Repositorio Digital.

REPXOS
Repositorio Digital

Servicios -

[Página de inicio](#)

[Listar](#) -

[Ayuda](#)

Repositorio Digital

Bienvenido a REPXOS 4.0

Esta es la página inicial de tu repositorio digital.

Sus futuros envíos aparecerán en esta página.

Anexo 7: Curvas de relación Coste/Tiempo en metodologías tradicionales (Izquierda) y ágiles (Derecha).

