



Facultad 1

Módulo de integración de XABAL REPXOS con Drupal

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

Autor: Gian Michel Rodríguez Pérez

Tutor: Ing. Yunior Chacón Sorio

La Habana, mayo de 2022

“Año 64 de la Revolución”

A black and white photograph of Richard P. Feynman. He is shown from the waist up, wearing a light-colored, long-sleeved button-down shirt and dark trousers with a belt. He is standing in a wooded area with trees in the background. He is looking slightly to his right and has his hands raised in a gesturing motion, as if explaining something. The lighting is natural, suggesting an outdoor setting.

Pensamientos

“La prueba final del conocimiento es tu capacidad de poder transmitirlo a otra persona”

Richard P. Feynman

DECLARACIÓN DE AUTORÍA

El autor **Gian Michel Rodríguez Pérez** del trabajo de diploma con título: **Módulo para facilitar la visibilidad de documentos en XABAL REPXOS**, 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 <día> días del mes de <mes> del año <año>.

<nombre del autor>

Firma del Autor

<nombre del tutor>

Firma del Tutor

<nombre del autor>

Firma del Autor

<nombre del tutor>

Firma del Tutor

DATOS DE CONTACTO

Contacto del tutor: Ing. Yunior Chacón Sorio

Correo: ychacon@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba

AGRADECIMIENTOS

Quiero agradecer, en un principio, a todas aquellas personas que de una forma u otra me apoyaron incondicionalmente en estos cinco años de carrera.

A mi tutor Yunion Chacón Sorio, por ofrecer sus conocimientos y perfeccionar los míos, por su comprensión, dedicación, profesionalidad, sabiduría y amistad.

A mis abuelos, mis tíos y mis primos por su apoyo en mis éxitos y momentos difíciles.

Al colectivo de profesores que a lo largo de 5 años de mi carrera se han esforzado en mi formación como Ingeniero en Ciencias Informáticas.

Agradezco a mi tribunal de tesis y oponente pues sus sugerencias me sirvieron de guía para lograr un buen trabajo de diploma.

A todas aquellas personas, amigos y compañeros de aula que de una manera u otra me apoyaron y estuvieron en mis andanzas que han colaborado de una u otra forma para que se dé este momento tan especial, ya sea con un consejo, una ayuda, acompañando, incluso regañando y guiando mis pasos.

A mis abuelas por siempre apoyarme y estar ahí siempre dándome ánimos, por su amor incondicional y creer en mí.

A mi madre, que siempre estuviste para decirme no desistas, levántate y brilla. Por el gran amor y la devoción que me tienes, por el apoyo ilimitado e incondicional que siempre me has dado, por tener siempre la fortaleza de salir adelante sin importar los obstáculos, por haberme formado como un hombre de bien, por ser mi mejor amiga, mi confidente, mi todo y por ser la mujer que me dio la vida y me enseñó a vivirla, no hay palabras en este mundo para agradecerte, mamá.

A todos ustedes, con todo mi corazón.

DEDICATORIA

A mi madre por ser mi motor impulsor, gracias por confiar y creer siempre en mí y en mis expectativas.

A mis abuelos y mis tíos por su apoyo en cada momento.

A todos ustedes, con todo mi corazón.

RESUMEN

La presente investigación consistió en desarrollar un módulo para Drupal que facilita la visibilidad de documentos desde la plataforma XABAL Repxos, dado que no existe un módulo que permita mostrar en un sitio web creado con Drupal, documentos publicados desde la plataforma XABAL Repxos de forma sencilla. XABAL Repxos es un sistema para repositorios digitales desarrollado en la Universidad de las Ciencias Informáticas en el Centro de Informatización de la Gestión Documental (CIGED), es de gran importancia pues permite a los clientes el acceso a la información, garantizando un camino para una mayor propagación del conocimiento. Brinda varios beneficios entre ellos garantizar la visibilidad de los autores, posibilitar la relación entre ellos, auspiciar el debate de los trabajos depositados, colaborar al incremento de las citas y al impacto de los trabajos en la comunidad científica de la institución donde se instale («REPXOS 4.0 | Universidad de las Ciencias Informáticas» [En línea]). Esta solución permite a administradores y editores de un sitio web desarrollado en Drupal, visualizar publicaciones científicas con una mayor facilidad, logrando una mejor utilización de los contenidos que se publican en la Universidad.

PALABRAS CLAVE: repositorio digital, visibilidad de documentos, gestión del conocimiento, REPXOS

ABSTRACT

The present investigation consisted of developing a module for Drupal that facilitates the visibility of documents from the XABAL Repxos platform, since there is no module that allows to display on a website created with Drupal, documents published from the XABAL Repxos platform in a simple way. XABAL Repxos is a system for digital repositories developed at the University of Informatics Sciences in the Document Management Computerization Center (CIGED), it is of great importance because it allows clients access to information, guaranteeing a path for greater spread of knowledge. It provides several benefits, among them guaranteeing the visibility of the authors, enabling the relationship between them, sponsoring the debate of the deposited works, collaborating to increase citations and the impact of the works in the scientific community of the institution where it is installed («REPXOS 4.0 | Universidad de las Ciencias Informáticas» [En línea]). This solution allows administrators and editors of a website developed in Drupal to view scientific publications more easily, making better use of the content published at the University.

KEYWORDS: digital repository, document visibility, knowledge management, REPXOS

TABLA DE CONTENIDOS

Índice General

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL.....	<u>76</u>
I.1 Conceptos asociados	<u>76</u>
I.2 Situación existente	<u>98</u>
I.3 Metodología.....	<u>1140</u>
I.4 Herramientas.....	<u>1312</u>
Conclusiones del capítulo.....	<u>15</u>
CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL.....	<u>1746</u>
II.1 Descripción de la propuesta de solución	<u>1746</u>
II.2 Ingeniería de Requisitos	<u>1817</u>
II.3 Análisis y diseño.....	<u>2422</u>
II.4 Modelo de despliegue	<u>3028</u>
Conclusiones del capítulo.....	<u>3230</u>
CAPÍTULO III: VALIDACIÓN DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL.....	<u>3334</u>
III.1 Diagrama de componentes.....	<u>3334</u>
III.2 Estándares de codificación.....	<u>3533</u>
III.3 Validación de la propuesta de solución.....	<u>3936</u>
Conclusiones del capítulo.....	<u>5045</u>
CONCLUSIONES FINALES	<u>5146</u>
RECOMENDACIONES.....	<u>5247</u>
REFERENCIAS BIBLIOGRÁFICAS.....	<u>5348</u>
ANEXOS.....	<u>5853</u>

ÍNDICE DE TABLAS

TABLA 1. ESTUDIO DE HOMÓLOGOS	11
TABLA 2: FASES DE LA VARIACIÓN DE AUP PARA LA UCI	12
TABLA 3. LISTADO DE LOS REQUISITOS FUNCIONALES.....	20
TABLA 4. HISTORIA DE USUARIO DEL RF6 MOSTRAR METADATOS.....	22
FIGURA 3. FUNCIONAMIENTO DEL ESTILO ARQUITECTÓNICO REST	27
FIGURA 5. DIAGRAMA DE CLASE DEL DISEÑO MOSTRAR METADATOS	30
FIGURA 6. MODELO DE DESPLIEGUE	31
FIGURA 7. DIAGRAMA DE COMPONENTES.....	34
TABLA 5: VARIABLES EMPLEADAS EN EL CASO DE PRUEBA BASADO EN LA HU MOSTRAR METADATOS	<u>4039</u>
TABLA 6: CASO DE PRUEBA CORRESPONDIENTE A LA HU MOSTRAR METADATOS	<u>4039</u>
TABLA 7: RESULTADO DE LAS PRUEBAS DE INTEGRACIÓN	<u>4342</u>
FIGURA 9. REPRESENTACIÓN DE LA CONFIGURACIÓN DE LAS PRUEBAS DE RENDIMIENTO. PARTE 1.....	<u>4544</u>
FIGURA 11. REPRESENTACIÓN DE LA CONFIGURACIÓN DE LAS PRUEBAS DE RENDIMIENTO. PARTE 3.....	<u>4645</u>
TABLA 8. CUADRO LÓGICO DE IADOV	48
TABLA 9. RESULTADOS DE LAS ESCALAS DE SATISFACCIÓN.....	49

ÍNDICE DE FIGURAS

FIGURA 1. MODELO CONCEPTUAL.....	<u>1729</u>
FIGURA 2. ARQUITECTURA DEL MÓDULO	<u>2629</u>
FIGURA 3. FUNCIONAMIENTO DEL ESTILO ARQUITECTÓNICO REST	<u>2739</u>
FIGURA 4. DIAGRAMA DE CLASE DEL DISEÑO DE DRUPAL	<u>2932</u>
FIGURA 5. DIAGRAMA DE CLASE DEL DISEÑO MOSTRAR METADATOS	<u>3033</u>
FIGURA 6. MODELO DE DESPLIEGUE	<u>3134</u>
FIGURA 7. DIAGRAMA DE COMPONENTES.....	<u>3437</u>
FIGURA 8. INDENTACIÓN DE LA PROPUESTA DE SOLUCIÓN.....	<u>3639</u>
FIGURA 9. OPERADORES DE LA PROPUESTA DE SOLUCIÓN	<u>3639</u>
FIGURA 10. USO DE COMILLAS DE LA PROPUESTA DE SOLUCIÓN.....	<u>3639</u>
FIGURA 11. USO DE PUNTO Y COMA (;) EN CÓDIGO PHP DE LA PROPUESTA DE SOLUCIÓN	<u>3639</u>
FIGURA 12. ESTRUCTURAS DE CONTROL DE LA PROPUESTA DE SOLUCIÓN	<u>3740</u>
FIGURA 13. FUNCIONES DE LA PROPUESTA DE SOLUCIÓN	<u>3844</u>
FIGURA 14. ARRAYS DE LA PROPUESTA DE SOLUCIÓN.....	<u>3844</u>
FIGURA 15. COMENTAR EL CÓDIGO DE LA PROPUESTA DE SOLUCIÓN	<u>3942</u>
FIGURA 16. COMPORTAMIENTO DE LAS NO CONFORMIDADES POR ITERACIONES	<u>4245</u>
FIGURA 17. REPRESENTACIÓN DE LA CONFIGURACIÓN DE LAS PRUEBAS DE RENDIMIENTO. PARTE 1.....	<u>4548</u>
FIGURA 19. REPRESENTACIÓN DE LA CONFIGURACIÓN DE LAS PRUEBAS DE RENDIMIENTO. PARTE 3.....	<u>4649</u>

INTRODUCCIÓN

A lo largo de la historia las instituciones y personas han logrado informatizar los servicios y el control de documentos, debido a la continua evolución de la información. Actualmente una de las principales formas de almacenar la información de grandes volúmenes de documentos son los sistemas de gestión de documentos. Estos sistemas de gestión documental proporcionan principalmente seguridad, almacenamiento y la capacidad de recuperar archivos, como ejemplo de ello tenemos los repositorios digitales.

Los repositorios componen sistemas de información que tienen como objetivo ordenar, salvaguardar y propagar en la modalidad acceso abierto (*Open Access*) recursos científicos y académicos de las instituciones. Cuba, ha favorecido la creación de estos sistemas con la intención de asegurar un acceso fácil, verificado y supervisado de determinados recursos informativos (Duperet Cabrera et al., 2015).

El movimiento de acceso abierto a la literatura científica, como método para favorecer la mayor cantidad de los documentos científicos y académicos, promueve el crecimiento de los repositorios en todo el mundo. Según *Barton* y *Waters* "un repositorio institucional no se define únicamente por el software y la base de datos que contiene sus colecciones digitales. Es un conjunto de servicios para aquellos que almacenan contenidos, tanto las comunidades académicas y de investigación que abarca, como para los usuarios finales" (Piñeiro, Sera y Puente, 2015). Se reconoce así a los repositorios como el medio ideal para conservar, compartir y reutilizar conjuntos de datos de investigaciones científicas que se desarrollan en las universidades como instituciones productoras de conocimiento.

En el transcurso de un estudio o una investigación se encuentran muchas publicaciones, que contienen información relevante que posteriormente se va a necesitar. Para identificarlas fácilmente, se necesita registrar los detalles (Piñeiro, Sera y Puente, 2015). A partir de estos detalles se facilitará la visibilidad de los documentos almacenados en la plataforma.

La Universidad de las Ciencias Informáticas (UCI), pilar universitario que sirve de soporte a la industria cubana de la informática, surgida el 23 de septiembre de 2002 con la visión de futuro del Comandante en Jefe Fidel Castro Ruz tiene como misión: Formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática, producir aplicaciones y servicios informáticos a partir del vínculo estudio–trabajo como modelo de formación– investigación–producción, sirviendo de soporte a la industria cubana de la Informática («Universidad de las Ciencias Informáticas», 2022).

Estas características hacen que la Universidad forme parte indispensable en el proceso de gestión de la información y el conocimiento en función de las prioridades nacionales para las empresas y entidades. Es por ello que el Centro de Informatización de la Gestión Documental (CIGED por sus siglas en español) desarrolló XABAL REPXOS (Sistema para Repositorios Institucionales). Este sistema administra todas las publicaciones científicas e investigativas producidas en la Universidad con el fin de mantenerlas agrupadas y asequibles en todo momento.

Proporciona almacenamiento, facilita la visualización y concede la obtención de ítems (documento digital con sus metadatos asociados) mediante búsquedas simples y avanzadas. Repxos es un sistema web que permite agrupar y propagar la producción intelectual digital desarrollada por una institución. Se basa en la herramienta DSpace (programa de código abierto que provee herramientas para la gestión de colecciones digitales) es comúnmente usada como base para el desarrollo de repositorios como es el caso de Acceda, Aquatic Commons y Social Science Research Network (Trujillo Pérez, 2019).

El Centro de Innovación y Desarrollo para Internet (CIDI), perteneciente a la Facultad 1 de la UCI, encargado de proveer soluciones integrales, productos y servicios relacionados con las tecnologías de internet, en función del desarrollo de la Universidad, tiene como una de sus misiones, el desarrollo web y utiliza el CMS Drupal para la gestión de los contenidos de estos portales.

A diferencia de otros CMS, Drupal es un potente Content Management Framework (CMF) que permite crear no solamente soluciones en buena medida basadas en web, sino también, que puede ser usado como un Application Programming Interfaces (API) backend o sistema de gestión de contenidos omnicanal para conectar con otros sistemas, o para alimentar otras aplicaciones con los datos que almacena.

Drupal, está construido sobre componentes Open Source, su core está basado en Symfony, lo cual quiere decir que, el producto evolucionará siempre igual que lo hacen las librerías sobre las que está desarrollado su core.

La API de migraciones que viene con el core permite automatizar la importación del contenido hacia Drupal desde los distintos orígenes de datos, transformar y adaptar la información y guardarla en entidades y campos de la base de datos de Drupal. Así es posible utilizar Drupal como un frontend para mostrar los contenidos traídos de otras plataformas.

La necesidad de acceder a un recurso de XABAL Repxos desde Drupal hace necesario la integración de los sistemas pues actualmente no existe interoperabilidad entre estos y no es posible acceder a un elemento del repositorio desde Drupal por lo que no se puede compartir datos entre estos de manera fluida entorpeciendo el trabajo de algunas áreas. Con las nuevas tendencias en la organización del contenido y las nuevas tecnologías para la enseñanza-aprendizaje sería muy útil la comunicación de estas plataformas ya que son muy utilizadas por los especialistas de CIDI, esto les permitiría utilizar el contenido publicado en el repositorio de la Universidad como fuente generadora de conocimiento para la creación de nuevos portales web y módulos. Provee además un servicio que permite el intercambio de información automático. Algunos sitios no poseen consumo de documentos a través de API lo que hace que el mismo se cargue de información y ralentice el trabajo y el tiempo de respuesta de este provocando la pérdida de interés del usuario, la instalación de este módulo va a permitir la utilización de los documentos del repositorio sin necesidad de guardar la información en el sitio y manteniendo el control de las publicaciones.

A partir de la situación problemática antes mencionada se define como **problema a resolver**: ¿Cómo consumir los documentos de la plataforma XABAL Repxos a través de un servicio web?

Determinando para la investigación como **Objeto de estudio**: el proceso de visualización de documentos en repositorios digitales a través de un servicio web; dando paso **al campo de acción**: el proceso de visualización de documentos del repositorio institucional Repxos a través de una API REST.

Para dar solución al problema se plantea como **objetivo general**: desarrollar un módulo para Drupal para consumir los documentos en XABAL Repxos a través de un servicio web REST.

Preguntas científicas:

1. ¿Cuáles son las tendencias y tecnologías existentes en el mundo, las herramientas, metodología de desarrollo de software y lenguajes de programación a tener en cuenta para abordar la solución del problema planteado?
2. ¿Cómo precisar las características con que contará el módulo de integración y cómo realizar su análisis y diseño?
3. ¿Cómo consumir los documentos de la plataforma XABAL Repxos mediante un servicio web?
4. ¿Qué resultados tendrá la aplicación del Módulo de integración con XABAL Repxos?

Para dar cumplimiento al objetivo general propuesto se llevarán a cabo las siguientes **tareas de investigación**:

1. Definición de fundamentos teóricos y metodológicos relevantes a la visibilidad de documentos en un servicio web.
2. Análisis y descripción de la propuesta de solución para determinar los requisitos funcionales y no funcionales que darán paso al diseño del módulo para la integración de XABAL Repxos.
3. Desarrollo del módulo de integración de XABAL Repxos que permita consumir los documentos de la plataforma mediante un servicio web.
4. Valoración de los resultados de la aplicación del Módulo de integración de XABAL Repxos mediante la aplicación de pruebas de software y técnicas de validación.

Los **métodos científicos** para dar cumplimiento a las tareas de investigación previstas y que estuvieron determinados por el objetivo general se describen a continuación.

Métodos teóricos:

Histórico-lógico: se empleó este método para analizar la evolución histórica de los repositorios digitales, su clasificación y los estándares utilizados, sobre todo en sitios web de universidades e instituciones nacionales e internacionales.

Inductivo-deductivo: se empleó para analizar una muestra representativa de trabajos asociados al desarrollo de software vinculado a XABAL Repxos, buscando elementos en común para encontrar una propuesta para darle solución a la problemática planteada.

Analítico-Sintético: para realizar un análisis de las teorías y conceptos sobre la extracción y conversión de metadatos para la generación de los datos necesarios para el repositorio, permitiendo la valoración de los elementos más importantes que definen la necesidad de la investigación.

Modelación: se utilizó para representar la solución a través de esquemas y diagramas para tener una mejor perspectiva del módulo en cuestión a desarrollar.

Métodos Empíricos:

Observación: para comparar los resultados obtenidos con los esperados en el desarrollo del módulo y tener una visión clara de los elementos presentes en la investigación.

Entrevista: se realiza una entrevista a los especialistas del proyecto Repositorio Institucional, mediante la cual se obtiene toda la información posible para el desarrollo del Módulo de integración con la plataforma XABAL Repxos, procesos del negocio, requisitos funcionales y no funcionales.

El presente documento de tesis está integrado por: Resumen, Introducción, tres Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía, Anexos y Glosario de términos, para una mejor estructuración y comprensión de la investigación.

Capítulo 1: FUNDAMENTACIÓN TEÓRICA DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL. Trata sobre los elementos teóricos que sustentan la investigación, incluye el estudio del arte del tema tratado, se describen los conceptos fundamentales referentes a la investigación, se realiza una búsqueda de las soluciones similares y se analizan las herramientas, metodologías y tecnologías a utilizar justificando su selección y uso.

Capítulo 2: ANÁLISIS Y DISEÑO DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL. Se especifican los requerimientos funcionales y no funcionales, a partir de los cuales se representan las historias de usuario de cada uno de ellos. Se describe la propuesta de solución y se muestran los prototipos del sistema. Se presenta el patrón arquitectónico, los patrones de diseño y el modelado de diseño. Se presentan los principales artefactos generados en las primeras fases del desarrollo.

Capítulo 3: VALIDACIÓN DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL. Se presenta el diagrama de componentes para un mejor entendimiento de la solución a desarrollar y los estándares de codificación utilizados. Describe todo lo relacionado a las pruebas realizadas al módulo desarrollado, en aras de verificar su calidad y efectividad. Se muestran los artefactos generados en el diseño del módulo.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL.

En el presente capítulo se analizan los elementos teóricos que sustentan la investigación, se estudian las tendencias y tecnologías existentes en el mundo, relacionadas con los repositorios institucionales, además se definen los conceptos básicos a utilizar para una mejor comprensión del dominio del problema. Se fundamentan las herramientas, tecnologías, lenguajes de programación y metodología de desarrollo de software que se utilizan en la implementación del sistema.

I.1 Conceptos asociados

Es necesario plantear algunos conceptos asociados al contexto del problema para una mejor comprensión.

I.1.1 Repositorio

Un repositorio es un sitio para la preservación de la información digital de la historia institucional. Los datos almacenados en el repositorio son de Acceso Abierto (Ciencia Abierta), suelen tener sistemas de respaldo y se le aplican mantenimientos preventivos y correctivos, lo que hace que la información esté protegida y se pueda recuperar en casos de fallas en el sistema (De Giusti, Marisa Raquel; Lira, Ariel Jorge; Villarreal, Gonzalo Luján; Texier, José, 2012). Proporcionan organización y permite identificar los derechos de autor de las publicaciones depositadas, permitiendo mayor cantidad de citas de fuentes confiables. Permite a la comunidad un control sobre sus publicaciones y compartir experiencias con otros autores (Manola, 2022). Es un elemento tecnológico unido con diferentes aplicaciones web que permiten la visibilidad de la identidad digital de la comunidad académica de las universidades. La definición de “acceso abierto” se transformó al añadirse elementos, componentes y características y es conocido como “Ciencia Abierta” que significa el acceso libre y gratuito a través de internet a los contenidos científicos aportados con los fondos públicos de los gobiernos y que son almacenados en repositorios institucionales. Es por ello que instituciones educativas como Massachusetts Institute of Technology (MIT), comenzaron a brindar sus programas de estudio de forma libre a través de internet y le denominaron OpenCourseWare (OCW) (González-Pérez, 2019).

I.1.2 Metadato

El concepto de metadatos existe desde antes de la aparición de la internet, pero en la actualidad se ha propagado debido a la necesidad de estructurar la información de la Web. Estos son un conjunto de elementos necesarios para explicar las características de un recurso determinado, funciona como

identificador de los materiales digitales diseñados. Para ello existen estándares que deben seguirse. Se le da la acepción de dato sobre el dato pues proporciona información necesaria para identificar un recurso, tiene como objetivos identificar, describir y definir un recurso para filtrar e informar sobre la autenticación y evaluación de este (Agudelo Benjumea, 2020).

I.1.3 XABAL Repxos

Es un repositorio institucional basado en DSpace. Tiene como objetivo almacenar, salvaguardar y propagar la información científica y académica de la comunidad universitaria. Permite publicar documentos resultantes de investigaciones científicas, otorga mayor visibilidad e impacto permitiendo mayor cantidad de citas garantizando una correcta gestión de los derechos de autor. Esto beneficia a la sociedad pues da acceso al conocimiento y permite la reutilización del contenido publicado en beneficio de todos (Trujillo Pérez, 2019).

I.1.4 API (Interfaz de Programación de Aplicaciones)

Es un grupo de definiciones o métodos utilizados por los programas informáticos para diseñar e integrar el software de las aplicaciones. Hacen posibles que sus servicios se comuniquen con otros, ofrecen oportunidades de innovación y permite ahorrar tiempo y dinero, resultando imprescindible para crear herramientas y productos nuevos. Se clasifican en SOAP y REST (Santos Hernández y Serrano Parreño, 2017).

I.1.5 Servicio REST

Es una forma fácil de enviar y recibir datos entre el cliente y el servidor, estos pueden ser JSON, XML e incluso texto sin formato posibilitando que puedan ser utilizadas por cualquier dispositivo que entienda HTTP (Santos Hernández y Serrano Parreño, 2017).

Características de REST:

Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información que necesita para ejecutarla, por lo que ni cliente ni servidor tienen que recordar ningún estado previo para satisfacerla.

Utiliza los métodos:

GET: Para consultar y leer recursos

POST: Para crear recursos

PUT: Para editar recursos

DELETE: Para eliminar recursos

PATCH: Para editar partes concretas de un recurso

Sistema de capas: arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.

I.1.6 Drupal

Si preguntas a los expertos en tecnología qué es un sistema de gestión de contenidos, lo más probable es que citen a Drupal como ejemplo. Eso es porque es el CMS que probablemente requiere más conocimientos técnicos.

Drupal es un sistema de gestión de contenidos de código abierto que sirve para administrar recursos web. Es un sistema multiusuario, multiplataforma, multilenguaje, extensible y modular. Introduce el concepto de nodo como sinónimo de tipos de contenido, cualquier recurso que se ingrese al sistema pasa a ser un nodo, que puede ser variable e incluir artículos, historias, posts, encuestas, imágenes, libros colaborativos, reseñas, recetas y más.

Este nuevo concepto permite estandarizar la información asignándoles las mismas características a distintos tipos de objetos y la posibilidad de tener toda la información centralizada y a la vez catalogada. La clasificación de los nodos es un tanto arbitraria y está al servicio de los usuarios/administradores de contenidos que pueden construir los que necesite (García García, 2017).

Drupal presenta un sistema de taxonomías robusto y potente, el contenido se organiza en categorías que se arman a través del módulo taxonomías, el cual permite generar vocabularios controlados con términos que pueden ordenarse jerárquicamente y asociarse a un tipo de nodo en particular si hiciera falta.

Se puede configurar que un nodo pueda ser clasificado bajo uno o múltiples términos de un vocabulario, flexibilidad que aprecian los administradores de sistemas de contenidos rígidos (García García, 2017).

Desde un avanzado panel de administración, ayuda a controlar totalmente la presentación del contenido, así como la gestión de usuarios y permisos. De entrada, carga rápido, es seguro y soporta más de 70 idiomas. Las soluciones preparadas para diferentes segmentos de mercado, como el turismo o el comercio electrónico, facilitan a los usuarios la creación de sitios web.

I.2 Situación existente

En el Centro de Informatización de la Gestión Documental (CIGED) de la Universidad de las Ciencias Informáticas (UCI) se encuentra el proyecto Repositorio Institucional con el sistema XABAL REPXOS. Este cuenta con un conjunto de módulos como son el Módulo de Referencias Bibliográficas que permite

la extracción de metadatos de los documentos consultados, el Módulo de Recomendaciones que permite la recomendación de artículos a otros investigadores, el Módulo Página de Autor que facilita obtener por cada autor del sistema las publicaciones realizadas en cualquier tipo de evento, entre otros (Labrada Delgado, 2019). La necesidad de acceder a un recurso de XABAL Repxos desde Drupal hace necesario la integración de los sistemas pues actualmente no existe interoperabilidad entre estos y no es posible acceder a un elemento del repositorio desde el CMS por lo que no se puede compartir datos entre estos. Esta solución permite a especialistas de un sitio web desarrollado en Drupal, visualizar publicaciones científicas con una mayor facilidad, logrando una mejor utilización de los contenidos que se publican en la Universidad para la creación de nuevos portales web y módulos. Existen varios ejemplos de módulos de integración y creación de servicios web sobre los que se harán alusión a continuación.

1.2.1 Solución internacional

Creación de una API REST en Drupal e integración con Angular/IONIC. IONIC es un framework de código abierto que se usa para crear aplicaciones para los distintos clientes móviles, creada a través de HTML5 y una única interfaz para luego ser compilada a través de todas las plataformas. Podemos aprovechar esta solución pues utiliza una API REST en Drupal y servirá de guía para el desarrollo de este trabajo (Montenegro, 2020).

1.2.2 Solución nacional

Creación de la API REST de servicios web de las actividades colaborativas (wiki, foro, taller, glosario, diálogo, diario) de Moodle para la versión 1.8.x, donde estas puedan ser visualizadas y administradas desde otros sistemas y propiciar de esta forma el aprendizaje colaborativo. Su creación no está integrada con Drupal, pero es un ejemplo práctico de lo que se va a realizar (Cabo Díaz *et al.*, 2011).

El “Módulo para la integración de la plataforma Moodle con el repositorio institucional DSpace” establece la conexión mediante una API REST del repositorio DSpace con la plataforma Moodle, accediendo de forma simple a la información almacenada en este. Este repositorio cuenta con plugins que hacen posible añadir a la lista de actividades de un curso un nuevo recurso con un enlace al repositorio, posibilitando navegar en las diferentes comunidades y colecciones del DSpace para seleccionar un ítem (Bravo del Río, 2016).

1.2.3 Conclusiones del análisis de sistemas similares

Luego del análisis realizado se concluye que la integración de las plataformas permitirá una mejor visualización de los contenidos del repositorio en otras aplicaciones. Permitirá la evolución de la

plataforma Drupal posibilitando el acceso a los datos del repositorio para una utilización más precisa de sus documentos. Al no encontrar un módulo para Drupal que diera solución al problema planteado, se escogieron los elementos fundamentales del estudio de homólogos el desarrollo del módulo a implementar, como son: la flexibilidad de los sitios, la interoperabilidad y la gestión del conocimiento.

Tabla 1. Estudio de homólogos

Sistema	Funcionalidad	Uso de Drupal 9
Angular Ionic/Drupal	-Flexibilidad -Interoperabilidad	(-)
Alfresco/Drupal	-Flexibilidad -Interoperabilidad -Gestión de conocimiento	(-)
API Rest Moodle	-Interoperabilidad -Gestión de conocimiento	(-)
Moodle/Dspace	-Interoperabilidad -Gestión de conocimiento	(-)

I.3 Metodología

Se utilizó la siguiente metodología para el modelado de la propuesta de solución.

I.3.1 Metodología de desarrollo de software

Una metodología de desarrollo de software es un marco de trabajo que se utiliza para organizar, planear y controlar el desarrollo de un sistema informático. Son todas las herramientas, pautas, técnicas y artefactos que sirven de orientación a los desarrolladores en la realización del software en cuestión. Para este trabajo utilizaremos la Variación de AUP para la UCI (Muñoz-González, 2018).

1.3.2 Metodología Variación de AUP para la UCI

La metodología Variación de AUP para la UCI es una variante de la metodología AUP (Agile Unified Process, Proceso Unificado Ágil). La UCI desarrolló esta versión para orientar el proceso de desarrollo de software con el fin de que esta se adapte al ciclo de vida definido por la actividad productiva de la institución (Muñoz-González, 2018).

Esta versión a contiene tres fases: Inicio, Ejecución y Cierre.

Tabla 2: Fases de la variación de AUP para la UCI

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Actividades relacionadas con la planeación del proyecto. Estudio inicial para recopilar información acerca del alcance del proyecto, realizar gestión de tiempo, esfuerzo y costo y decidir si es rentable la realización del proyecto.
Elaboración	Ejecución	Se realizan las actividades necesarias para la ejecución del proyecto, se ajustan los planes del proyecto considerando los requisitos y la arquitectura. Se modela el negocio, se obtienen los requisitos, la arquitectura y el diseño. Implementación y liberación del producto.
Construcción		
Transición		
	Cierre	Para analizar los resultados, y el funcionamiento y se realizan las actividades formales de cierre de proyecto.

1.3.3 Escenarios para la disciplina requisitos que presenta AUP-UCI

La disciplina Modelado de negocio propone tres variantes a utilizar en los proyectos (casos de uso del negocio, descripción de proceso de negocio y modelo conceptual) y existen tres formas de encapsular los requisitos (casos de uso del sistema, historias de usuario y descripción de requisitos por proceso), a partir de esto surgen cuatro escenarios para modelar el sistema en los proyectos estos son:

Escenario No 1: Proyectos que modelen el negocio con casos de uso del negocio solo pueden modelar el sistema con casos de uso del sistema.

Escenario No 2: Proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con casos de uso del sistema.

Escenario No 3: Proyectos que modelen el negocio con descripción de procesos de negocio solo pueden modelar el sistema con descripción de requisitos por procesos.

Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con historias de usuario.

Para llevar a cabo la solución propuesta, se emplea el **escenario 4** de metodología, se modela el negocio con historias de usuario para la descripción de los requisitos ya que el proyecto tiene el negocio bien definido, se consulta todo el tiempo al cliente, especialistas del Centro de Innovación y Desarrollo para Internet (CIDI), para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos, el proyecto no es extenso y el equipo de desarrollo está conformado por un integrante.

I.4 Herramientas

Se utilizaron las siguientes herramientas para el modelado de la propuesta de solución.

I.4.1 Lenguaje Unificado de Modelado (UML)

El UML define la abstracción de un lenguaje para expresar otros modelos. Es una herramienta para dar soporte al diseño y el modelado de objetos para representarlo en términos concretos. Para ello se debe contar con la documentación adecuada para su desarrollo (Silva Avila *et al.*, 2018).

I.4.2 Herramienta Visual Paradigm 15.1 para UML

Visual Paradigm es un proveedor líder y mundialmente reconocido de gestión empresarial y desarrollo de software de las Tecnologías de la Información y las Comunicaciones (TIC). Mejora la agilidad empresarial y fomenta innovación a través de estándares abiertos populares.

Ofrece la gestión de proyectos y permite que los desarrolladores se adapten con flexibilidad, consiste en un gran conjunto de herramientas ágiles para satisfacer las necesidades de todo el proceso de desarrollo de software para su equipo («Visual Paradigm», 2022).

I.4.3 Marco de trabajo (Drupal 9.3.4)

Software de gestión de contenidos gratuito y de código abierto. Utilizado para crear sitios y aplicaciones web. Permite la creación de contenido de manera fácil con una seguridad confiable y un gran rendimiento. Sus módulos amplían sus funcionalidades. Puede integrarse con otros servicios externos mejorando las capacidades básicas de Drupal. Tiene gran aceptación lo que favorece el desarrollo de actualizaciones continuas («Drupal», 2018).

I.4.4 Entorno de Desarrollo Integrado (Visual Studio Code 1.67.2)

Visual Studio Code es un editor de código. Propicia las herramientas que un desarrollador necesita para un ciclo rápido de compilación y depuración de código, resalta las palabras clave en su código en diferentes colores para ayudarlo a identificar fácilmente los patrones de codificación y aprender más rápido. Brinda sugerencias a medida que se codifica para completar las líneas de código y soluciones rápidas para errores comunes («Visual Studio Code», 2022).

I.4.5 Gestor de Referencias (Zotero 6.0.8)

Zotero es una herramienta de investigación gratuita y de código abierto que permite juntar, ordenar y analizar investigaciones y compartirlas de diversas maneras. Incluye la capacidad de almacenar campos de autor, título y publicación y exportar esa información como referencias formateadas. Detecta que está viendo un libro, artículo u otro objeto en la web y permite extraer y guardar automáticamente las referencias bibliográficas completas. Zotero se ejecuta tanto como un servicio web como fuera de línea en los dispositivos personales («Zotero», 2022).

1.4.6 Servidor WEB (Apache 2.4.51)

Es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es potente, flexible y compatible con HTTP, es altamente configurable y extensible con módulos de terceros, fomenta la retroalimentación de los usuarios a través de nuevas ideas, informes de errores y parches («Apache HTTP Server Project», 2022).

I.4.7 Servicio CURL

Curl es una herramienta de línea de comandos “orientado a la transferencia de archivos” con sintaxis de URL, lo que la hace muy útil para interactuar con API REST y otros recursos web. Automatiza transferencia de archivos o secuencias de operaciones no supervisadas. Fue agregado en Windows 10 en diciembre de 2017 y se puede acceder a él a través de la línea de comandos sin software adicional (Stenberg, 2015).

I.4.8 Apache JMeter 5.5

Es un software de código abierto, una aplicación de Java 100 % pura diseñada para cargar, probar el comportamiento funcional y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba (Apache JMeter, 2022).

I.5 Lenguajes

Se utilizaron los siguientes lenguajes para el modelado de la propuesta de solución.

1.5.1 Lenguaje de marcas de hipertexto (HTML 5)

Es el lenguaje de marcado estándar para crear páginas web. Describe la estructura de una página Web. Los elementos HTML le dicen al navegador cómo mostrar el contenido. HTML 5 contiene un conjunto amplio de tecnologías que permiten a los sitios web y a las aplicaciones ser más diversas y de gran alcance. Presentan nuevas características que proporcionan un buen soporte multimedia y también mejoran el apoyo para la creación de aplicaciones web («Introduction to HTML» [Online]).

1.5.2 Hojas de estilo en cascada (CSS 3)

Es el lenguaje que usamos para diseñar una página web. Describe cómo se deben mostrar los elementos HTML en la pantalla, el papel o en otros medios CSS, ahorra mucho trabajo. Puede controlar el diseño de varias páginas web a la vez. Las hojas de estilo externas se almacenan en archivos CSS. En la actualidad, CSS3 brinda la facilidad de producir diseños más flexibles que permiten optimizar la experiencia del usuario, el uso de herramientas permite mejorar el rendimiento y la visualización en base al dispositivo de navegación que utilice el usuario. Con los nuevos selectores se pueden realizar efectos de animación sin el uso de JavaScript (Álvarez Jarel, 2018).

1.5.3 Java Script

Es el lenguaje de programación más popular del mundo. Es el lenguaje de programación de la Web. Es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia y animar imágenes («¿Qué es JavaScript?» [Online]).

1.5.4 PHP 7

Es un lenguaje de código abierto muy utilizado especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. El código es ejecutado del lado del servidor. Lo mejor de utilizar PHP es su simpleza para los principiantes, y a su vez presenta muchas características avanzadas para los programadores profesionales («PHP: Documentation» (Online)).

Conclusiones del capítulo

Como resultado del análisis realizado en este capítulo, se puede concluir que:

- El estudio de los referentes teóricos permitió identificar las funcionalidades básicas del módulo, entre las que se encuentran mayor flexibilidad permitiendo que las plataformas se adapten al nuevo entorno integrado, interoperabilidad permitiendo el uso de los dos sistemas como un

conjunto y la gestión del conocimiento permitiendo transferir el conocimiento desde donde se genera hasta donde se emplea.

- La identificación de las principales herramientas, tecnologías y metodología permitió facilitar el desarrollo del software.
- El análisis realizado en el Centro de Informatización de la Gestión Documental (CIGED) con el tutor asignado y los especialistas, la entrevista realizada a los desarrolladores del proyecto XABAL REPXOS; permitieron definir los detalles más importantes para la realización del módulo que se propone implementar, entre estos la planificación del tiempo a emplear para la realización del proyecto y los sitios de interés para recopilar la información necesaria.
- El estudio de los sistemas similares no cuenta con todas las características requeridas, como son la utilización de Drupal 9 o la utilización de DSpace para la publicación de contenido por lo que se hace necesario el desarrollo de un módulo para la integración del repositorio institucional de la Universidad con el CMS Drupal.
- El CMS Drupal tiene muchos beneficios para los especialistas de CIDI como son: la gran facilidad para desarrollar una página y mantenerla actualizada, tiene una amplia gama de características adicionales, se le pueden adicionar complementos y herramientas, es gratuito, permite la autoadministración de contenido y la creación de distintos tipos de contenido y de múltiples campos.

CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL.

El presente capítulo tiene como objetivo fundamental describir la propuesta de solución para el sistema a desarrollar, sus características y funcionalidades; así como mostrar los requisitos funcionales y no funcionales y los diagramas que requiere el módulo a implementar.

II.1 Descripción del sistema a través de un modelo conceptual

Se elaboró un modelo conceptual en la descripción de la propuesta de solución. El cual permitió identificar los conceptos existentes en el proceso a informatizar, así como sus atributos y relaciones, lo que permitió conocer las características del entorno en el que se desarrolla la solución. El Centro de Innovación y Desarrollo para Internet (CIDI) de la Universidad de las Ciencias Informáticas (UCI) actualmente no cuenta con un módulo para Drupal que facilite la visibilidad de los documentos publicados en la plataforma Repxos, por lo que se hace necesario la creación de este pues el CMS es muy utilizado por el centro y permitiría la utilización del contenido científico publicado en la Universidad para una mayor difusión de su contenido y aprovechamiento en la creación de portales web y módulos.

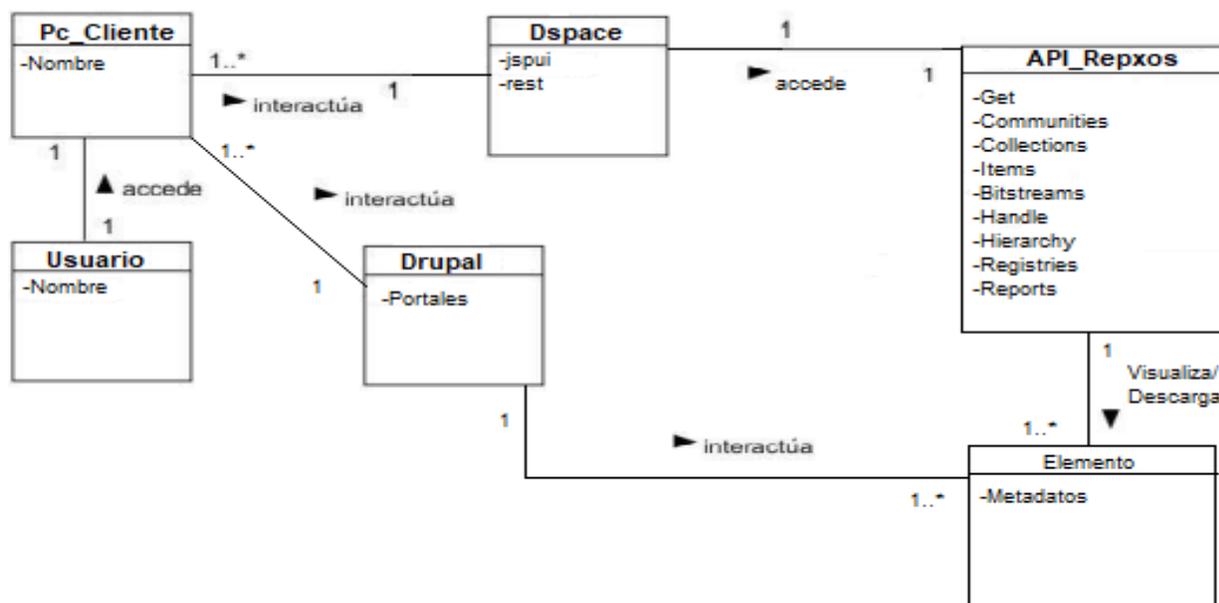


Figura 1. Modelo Conceptual

II.1.1 Descripción de los elementos del modelo conceptual

Usuario: es la persona que trabaja con la herramienta y que su interacción con el sistema depende de sus responsabilidades dentro del negocio.

PC_Cliente: es la máquina que interactúa con la herramienta y a su vez con el usuario.

DSpace: software de código abierto que provee herramientas para la administración de colecciones digitales, y comúnmente es usada como solución de repositorio bibliográfico institucional.

API_Repxos: es un API que permite la indexación de contenidos, la regeneración del índice y realizar búsquedas en el contenido, comunidades y colecciones.

Drupal: es un sistema de gestión de contenidos o CMS libre, modular, multipropósito y muy configurable.

En el diagrama modelo conceptual (figura 1) se inicia con un usuario accediendo a la Pc que contiene el sistema diseñado, este interactúa con DSpace a través de la dirección correspondiente y accede a la API_Repxos posibilitando obtener los documentos depositados en esta para visualizarlos o descargarlos. El usuario accede a la Pc cliente que contiene el CMS Drupal e interactúa con el elemento obtenido de la interacción con la API_Repxos ya sea desde el repositorio o descargado.

II.1.2 Descripción de la propuesta de solución

El usuario accede al CMS Drupal este contiene el módulo XABAL Repxos el cual permite la visualización de los documentos científicos e investigativos producidos por la Universidad de las Ciencias Informáticas mediante una tabla con el listado de elementos depositados en el repositorio y la posibilidad de acceder a sus metadatos y al elemento. Esta solución permite a especialistas y desarrolladores de Drupal, acceder a publicaciones científicas con una mayor facilidad, logrando una mejor utilización de los contenidos que se publican en la Universidad para la creación de nuevos portales web y módulos. Provee un servicio que permite el intercambio de información automático. La instalación de este módulo va a permitir la utilización de los documentos del repositorio sin necesidad de guardar la información en los sitios y manteniendo el control de las publicaciones brindando agilización a los servidores de los sitios y permitiendo que estos no se sobrecarguen.

II.2 Ingeniería de Requisitos

La ingeniería de requisitos constituye un elemento fundamental en todo proceso de desarrollo de software ya que se encarga de entender las necesidades exactas de los usuarios de un sistema, y traducir éstas a precisas funciones y/o acciones que serán usadas en el desarrollo del sistema, o bien, serán realizadas por el sistema al terminarse satisfactoriamente (Hernández Reinoza *et al.*, 2021).

II.2.1 Obtención de requisitos

La obtención de requisitos permite a los interesados el descubrimiento y la articulación de los requisitos del sistema. Facilitará la incorporación de cambios posteriores en el sistema; por lo tanto, se hace necesario que los analistas empleen técnicas que les permitan establecer una buena comunicación con los interesados y su equipo de trabajo, permitiendo descubrir el problema en cuestión que se debe resolver (Lenzano Ramírez, 2019).

A continuación, se describe la obtención de requisitos de la propuesta de solución mediante la entrevista y la observación.

Entrevista

Consiste en reuniones en las cuales se realizan preguntas y respuestas para extraer el dominio de la aplicación, se debe escoger y preparar el sitio adecuadamente, prepararse correctamente, crear las preguntas previamente y documentarse (Castañeda Bueno, 2010).

La entrevista (anexo 1) se realizó al especialista en la plataforma XABAL Repxos del Centro de Informatización de la Gestión Documental (CIGED), encargado de trabajar directamente con la plataforma; y a los especialistas en Drupal del Centro de Innovación y Desarrollo para Internet (CIDI). Esta permitió obtener información sobre la plataforma Repxos y el CMS Drupal para desarrollar el módulo en cuestión, así como requisitos funcionales y no funcionales de la propuesta de solución.

Observación

Esta permite la recopilación de información sobre la forma en que se realizan los procesos. Los ingenieros aprenden sobre las tareas del usuario observando como un especialista actúa y se relaciona con su software (Cardona Duque y Castaño Marín, 2015).

La observación realizada (anexo 2) al uso de la plataforma XABAL Repxos en el Centro de Informatización de la Gestión Documental (CIGED) le permitió conocer al autor de la investigación: la estructura de la plataforma, el funcionamiento y como obtener los datos correspondientes a las publicaciones.

II.2.2 Especificación de requisitos de software

Posibilita reflejar las propiedades o restricciones que debe cumplir el módulo, a fin de satisfacer las necesidades del cliente. A continuación, se proponen los siguientes requisitos funcionales y no funcionales.

II.2.3 Requisitos funcionales

Los requisitos funcionales (RF) son las descripciones específicas de las funciones que debe realizar el sistema, refleja cómo se debe comportar el sistema con entradas detalladas y contiene las propiedades esenciales de este, a fin de cumplir las expectativas de los usuarios finales (Feijoo Ortiz y García Portal, 2019). A continuación, se listan los requisitos funcionales que responden a la propuesta de solución:

Tabla 3. Listado de los requisitos funcionales

Número	Requisito funcional	Descripción	Prioridad
RF.1	Cargar elementos	El módulo debe cargar los elementos que va a mostrar.	Alta
RF.2	Mostrar listado de elementos	El módulo debe mostrar un listado con los elementos que va a mostrar.	Alta
RF.3	Mostrar Nombre elemento	El módulo debe permitir visualizar el nombre de los elemento cargados.	Alta
RF.4	Mostrar Autor	El módulo debe permitir mostrar el autor del elemento.	Alta
RF.5	Mostrar Fecha de subida	El módulo debe permitir mostrar la fecha de subida del elemento.	Alta
RF.6	Mostrar Metadatos	El módulo debe permitir visualizar los metadatos asociado al elemento.	Alta
RF.7	Descargar elemento	El módulo debe permitir descargar el elemento en el formato correspondiente.	Media

II.2.4 Requisitos no funcionales

Los requisitos no funcionales especifican o restringen las propiedades de los servicios. Incluyen restricciones tanto de tiempo y del desarrollo, como impuestas por estándares. Son las características que hacen al producto confiable, atractivo, rápido y usable, aportando un valor y utilidad para el cliente final (Lenzano Ramírez, 2019).

Para el desarrollo del módulo se definen los siguientes requisitos no funcionales (RNF)

Usabilidad

RNF1: El sistema está creado para ser utilizado por personas con conocimientos básicos de informática y el idioma de la interfaz será en español.

RNF2: Se utilizará colores y letras apropiadas para las personas con discapacidades visuales.

Disponibilidad

RNF3: El sistema debe estar disponible todo el tiempo para los usuarios autorizados.

RNF4: El período entre fallos recuperables, como por ejemplo fallos en el servidor no debe exceder las 24 horas.

Mantenimiento y soporte

RNF5: El módulo funcionará en varios sistemas operativos como Windows 7, 8 y 10, Linux en sus distribuciones basadas en Debian y Ubuntu.

Interfaz

RNF6: La aplicación debe ser visible en navegadores como Mozilla Firefox versión superior a la 50, Chrome versión superior a la 50 y Opera 11 o superior.

Licencia

RNF7: Se debe utilizar la licencia PHP License.

RNF8: Se debe utilizar la licencia de PostgreSQL.

RNF9: Se debe utilizar la licencia de MySQL.

Operacionales

RNF10: Para la distribución del producto y su instalación se necesitará un entorno de trabajo compuesto por:

- Servidor Web: Apache 2.4.51
- Base de Datos: MySQL 5.6
- Sistema gestor de contenido: Drupal 9
- Lenguaje de programación: PHP 7.4

II.2.5 Historia de Usuario (HU)

Las Historias de Usuario (HU) sirven para describir las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Es lo suficientemente comprensible y delimitada para que los programadores puedan implementarlos sin dificultad (Acosta Guzmán, 2019). A continuación, se muestra la historia de usuario del RF6 que preside de una prioridad alta, las restantes historias de usuario se mostrarán en el (anexo 3).

La prioridad en el negocio:

- Alta: Cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
- Media: Cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.

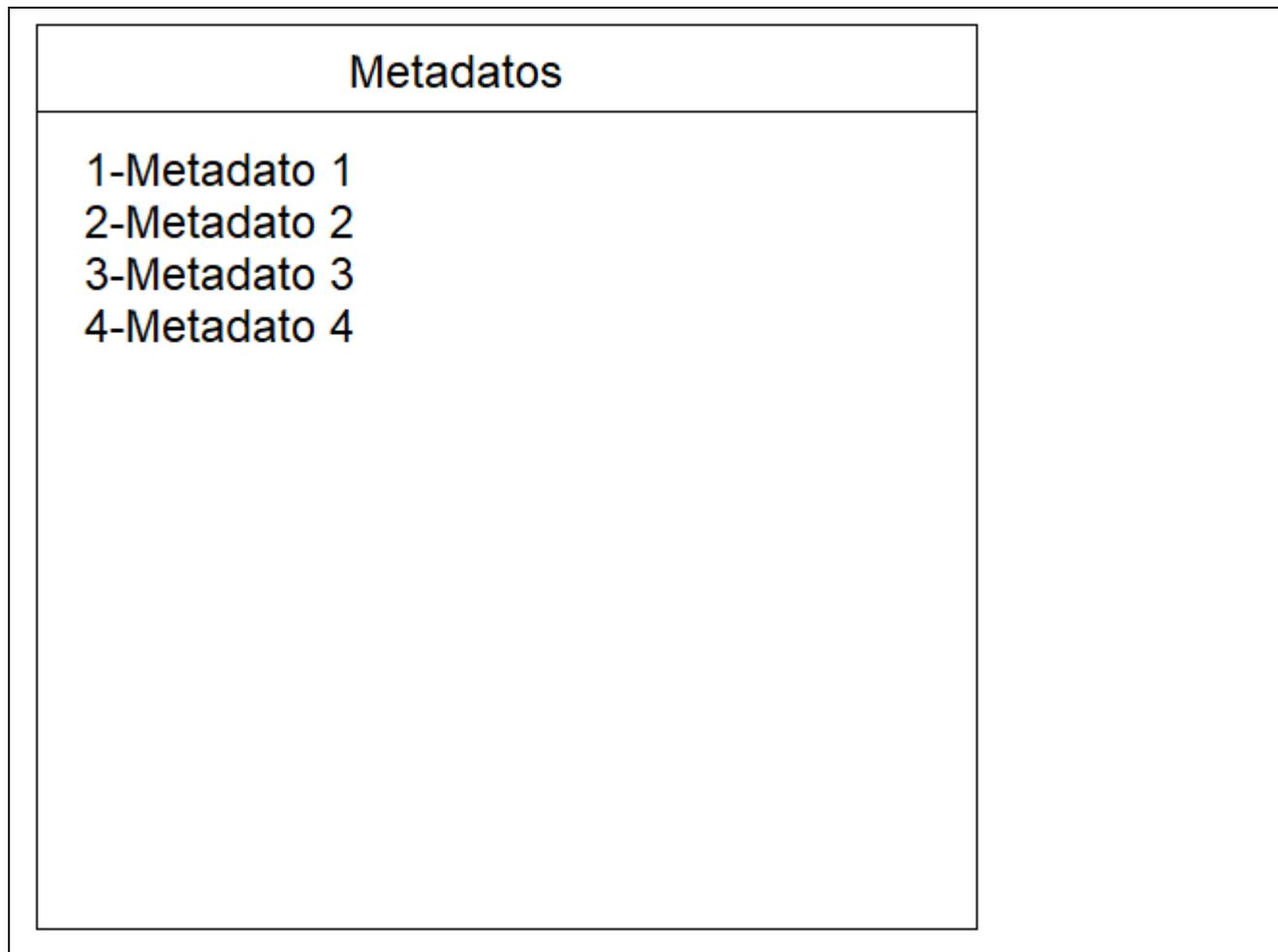
- Baja: Cuando constituyen procesos que se deben tener en cuenta, pero su ausencia no perjudica el flujo principal del negocio.

El riesgo en desarrollo:

- Alto: Cuando en la implementación de las HU pueden surgir errores que lleven a la inoperatividad del código.
- Medio: Cuando en la implementación de las HU pueden existir errores que retrasen la entrega del producto.
- Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Tabla 4. Historia de usuario del RF6 Mostrar metadatos

Historia de usuario	
Número: 4	Nombre del requisito: Mostrar metadatos
Programador: Gian Michel Rodríguez Pérez	Iteración Asignada: 1 iteración
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: Medio	Tiempo Real: 1 semanas
Descripción: La aplicación muestra un elemento con sus metadatos ordenados, a partir de la interacción con el nombre del elemento. Inicialmente el usuario da click en el nombre del elemento y este muestra una nueva pestaña con los metadatos asociados al elemento ya sean: Nombre, tipo, Fecha de subida, Autor, lenguaje, descripción, entre otros.	
Observaciones: El módulo mostrará los metadatos del elemento correspondiente en una nueva pestaña de forma ordenada.	
Prototipo de Interfaz:	



II.2.6 Validación de requisitos de software

La validación de requerimientos es un proceso continuo en el proyecto de desarrollo de software con el fin de garantizar que los requisitos hayan sido representados de manera exacta a las necesidades y expectativas de los usuarios. Esto contribuye a mejorar la calidad de los requisitos, a reducir tiempos, costos y riesgos en el desarrollo de software (Santana *et al.*, 2020).

A continuación, se describen las técnicas de validación de requisitos utilizadas en la propuesta de solución:

Diseño de casos de pruebas

Los diseños de casos de pruebas posibilitan crear un conjunto de entradas y salidas esperadas que sean positivos para encontrar fallas o ineficiencias en los programas e indiquen que el sistema cumple con sus requerimientos. Para su diseño, se elige una característica del sistema o componente que se está probando, un conjunto de entradas que ejecutan dicha característica, se documentan las salidas

esperadas o rasgos de salida y donde sea posible se diseña una prueba automatizada que demuestre que las salidas reales y las esperadas son las mismas (Muñoz González, 2018).

En el epígrafe 3.5.1 se evidencia un ejemplo de los diseños de casos de pruebas elaborados para la validación de los requisitos funcionales de la propuesta de solución.

Revisión Técnica Formal

Una revisión técnica formal (RTF) es una técnica de control de calidad del software realizada por ingenieros de software. Los objetivos de la revisión técnica formal son encontrar errores en la función, la lógica o la implementación del producto a desarrollar, verificar que satisface sus requerimientos, garantizar que cumple con los estándares establecidos, señalando las posibles desviaciones detectadas. Es un proceso de revisión riguroso que posibilita llegar a detectar lo antes posible, los posibles defectos en los productos que se van generando a lo largo del desarrollo (Muñoz González, 2018).

II.3 Análisis y diseño

En el análisis y diseño se transforman los requisitos al diseño del futuro sistema, se desarrolla una arquitectura para el sistema y se adapta el diseño para que sea consistente con el entorno de implementación. En el presente epígrafe se describe el diseño de clases, datos y los patrones de diseño.

II.3.1 Diseño arquitectónico

El diseño arquitectónico nos permite definir cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global de este. Es el vínculo entre el diseño y la ingeniería de requisitos, que reconoce los principales componentes estructurales en un sistema y la relación entre ellos (Muñoz González, 2018).

Arquitectura de Drupal

El módulo al ser implementado para el CMS Drupal, la arquitectura y los patrones son heredados del mismo. Drupal presenta una arquitectura n-capas lo que le da flexibilidad y facilidad en la creación de sitios web, este permite separar los diferentes aspectos del desarrollo mediante capas y permitir intercambiar porciones de la aplicación sin necesidad de modificarla completamente. Una de sus características principales es su adaptabilidad mediante la instalación de módulos, esto permite modificar porciones de la aplicación sin realizar cambios en el resto de la aplicación. Esta arquitectura posee específicamente 5 capas que son descritas a continuación (García García, 2017).

Nodos (Datos): en esta capa se encuentran los elementos básicos en que Drupal almacena la información, los contenidos. A medida que aumenta el sitio web, el número de nodos aumenta los cuales van formando un “Repositorio de Nodos”. Se evidencia en el contenido que muestra la tabla con las publicaciones científicas de la plataforma XABAL Repxos

Módulos: son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal, permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web. En esta capa se encuentra el módulo desarrollado XABAL Repxos Module.

Bloques y menús: permiten estructurar y organizar los contenidos en la página web, es decir, que son los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos. Se evidencia en los menús que genera, como ejemplo en el apartado Extend se ve XABAL Repxos Module con la información que genera.

Control de usuarios y permisos: Drupal dispone de un registro de usuarios y de roles que permiten especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario. Las operaciones realizadas sobre los elementos provenientes de las capas inferiores (lectura, modificación, creación, entre otras) se encuentran limitadas por la capa de control de usuarios y permisos de Drupal. En este caso todos los usuarios tienen acceso al módulo desarrollado.

Temas: establece la apariencia gráfica o estilo de la información mostrada al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño o apariencia del sitio web sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que se requiere es renovar la apariencia de un sitio web. Esta capa permite estructurar la información del módulo. Tema Plantilla Repxos que muestra el logo del sitio seguido por la tabla que contiene las publicaciones científicas que genera la plataforma.

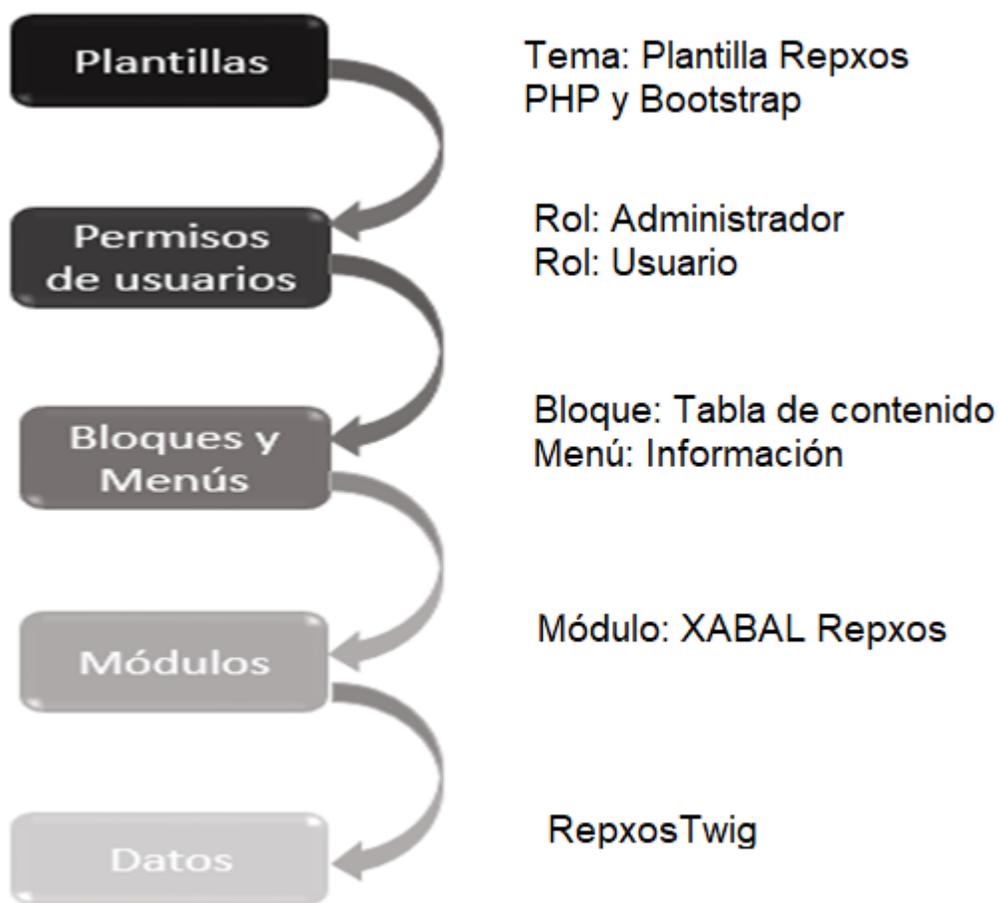
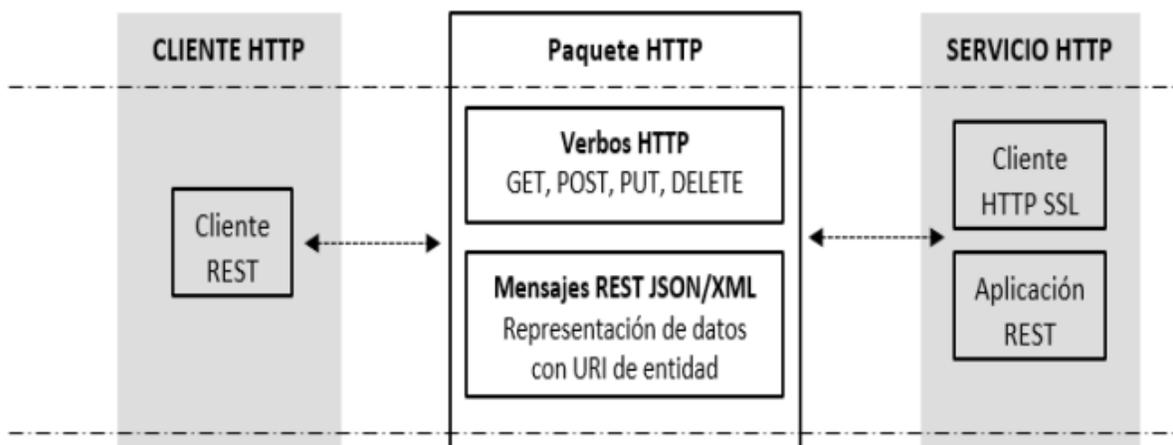


Figura 2. Arquitectura del módulo

Estilo arquitectónico REST

REST es un estilo de arquitectura orientado a los recursos, puede proporcionar grandes comodidades para construir aplicaciones a escala de internet permitiendo que las comunicaciones entre productor y consumidor sean más ligeras y escalables (Rea-Peñañiel *et al.*, 2020). Se usa para el desarrollo API REST utilizando el protocolo HTTP, garantizando que pueda ser utilizada prácticamente por cualquier lenguaje de programación y de manera fácil (Muñoz González, 2018)



```
//establecer NOBODY en true para hacer una solicitud tipo HEAD
curl_setopt($ch, CURLOPT_NOBODY, true);
//Permitir seguir redireccionamientos
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
//recibir la respuesta como string, no output
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

$data = curl_exec($ch);

//Obtener el código de respuesta
$httpcode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
//cerrar conexión
curl_close($ch);
```

Figura 3. Funcionamiento del estilo arquitectónico REST

II.3.2 Patrones de Diseños

Un patrón de diseño es una descripción del problema y la esencia de su solución, que se puede reutilizar en casos distintos, sin tener que pensar en cómo de válida es la respuesta o si puede haber una forma de solución mejor (del Pino Acosta, 2019). El objetivo de los patrones es crear un lenguaje común para los desarrolladores con el objetivo de transmitir experiencia sobre los problemas y soluciones (Gómez Montiel, 2019).

Patrones GOF:

Los patrones GoF (Gang of Four, en español Pandilla de los Cuatro, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides) se clasifican en 3 categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Muñoz González, 2018).

Cadena de responsabilidades (Chain of responsibility)

El sistema de menús de Drupal es la evidencia del patrón Cadena de responsabilidades. En cada petición de una página, el sistema de menús de Drupal determina si hay algún módulo para manejar la petición, si el usuario tiene acceso al recurso solicitado y que la función se llame para hacer el trabajo. En este proceso se transmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos (Arango Fernández, 2017). La implementación del hook_menu() es un ejemplo claro que genera un menú de configuración del módulo.

Puente (Bridge)

La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón Puente. Cada módulo debe ser programado de manera que sea independiente del motor de base de datos que utiliza el sistema. Esto se logra por la capa de abstracción de base de datos, sobre la que se pueden desarrollar nuevas funcionalidades siguiendo la API definida, para añadir soporte para nuevos sistemas gestores de base de datos, sin necesidad de modificar el código de los módulos (Álvarez Jarel, 2018). Este patrón se evidencia en el módulo Repxos que fue programado logrando independencia del motor de BD que se utilice.

Sencillo (Singleton)

Es un patrón de creación que se utiliza para garantizar la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Los módulos en Drupal pueden considerarse como objetos que no encapsulan datos, lo que separa a un módulo a otro es el conjunto de funciones que contiene. La propuesta de solución utiliza este patrón al ser un módulo para Drupal 9 (Muñoz González, 2018).

Acción (Command)

Es un patrón de comportamiento a nivel de objetos. Su propósito es encapsular en un objeto la acción que satisface una petición, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma. Muchos de los hook implementados en la propuesta de solución utilizan el patrón Acción para reducir el número de funciones que son necesarias para su funcionamiento, pasando la operación como un parámetro, junto con los argumentos. De hecho, el propio sistema de hook utiliza este patrón,

por lo que los módulos no tienen que definir cada hook, sino más bien sólo los que se necesiten implementar (Arango Fernández, 2017). Se evidencia en el Hook_Help.

II.3.2.1 Descripción de hook:

Son funciones PHP que aumentan la funcionalidad de Drupal permitiendo a los módulos interactuar con el core. De esta manera se puede alterar los comportamientos default de Drupal, y agregar nuevas funcionalidades, si son necesarias para nuestro sitio.

II.3.3 Diagrama de clases del diseño

El Diagrama de Clases es uno de los diagramas principales de análisis y diseño para un sistema. Muestra la representación del sistema y la relación que existe entre él; incluye clases, operaciones y los atributos de las clases (Hernández, Jiménez y Godoy, 2018).

Para alcanzar un mayor grado de comprensión del módulo a desarrollar, es indispensable tener en cuenta el funcionamiento del CMS Drupal. A continuación, se muestra el diagrama de clases del diseño de Drupal:

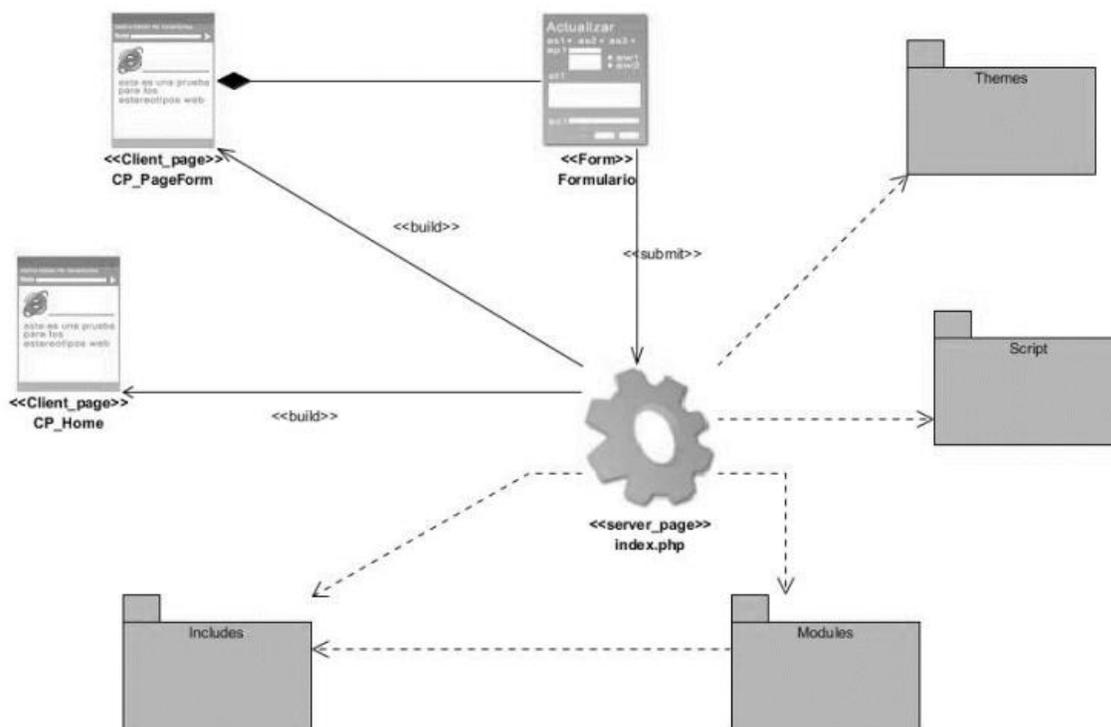


Figura 4. Diagrama de Clase del Diseño de Drupal

Includes: En este paquete se encuentran los ficheros de conexión a la base de datos del sistema.

Modules: Espacio en el que se encuentran todos los módulos que forman parte del funcionamiento del CMS. En este contenido se encuentra la propuesta de solución.

Themes: Agrupa las plantillas que forman parte del tema de la interfaz del sistema. Cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.

Scripts: Almacena un grupo de ficheros que contienen código para el correcto funcionamiento del sistema, permitiendo que se visualicen los datos correctamente.

A partir del análisis del Diagrama de clase de Diseño de Drupal, se modelan los diagramas de clase del diseño para cada escenario como se muestra a continuación para el requisito funcional Mostrar Metadatos:

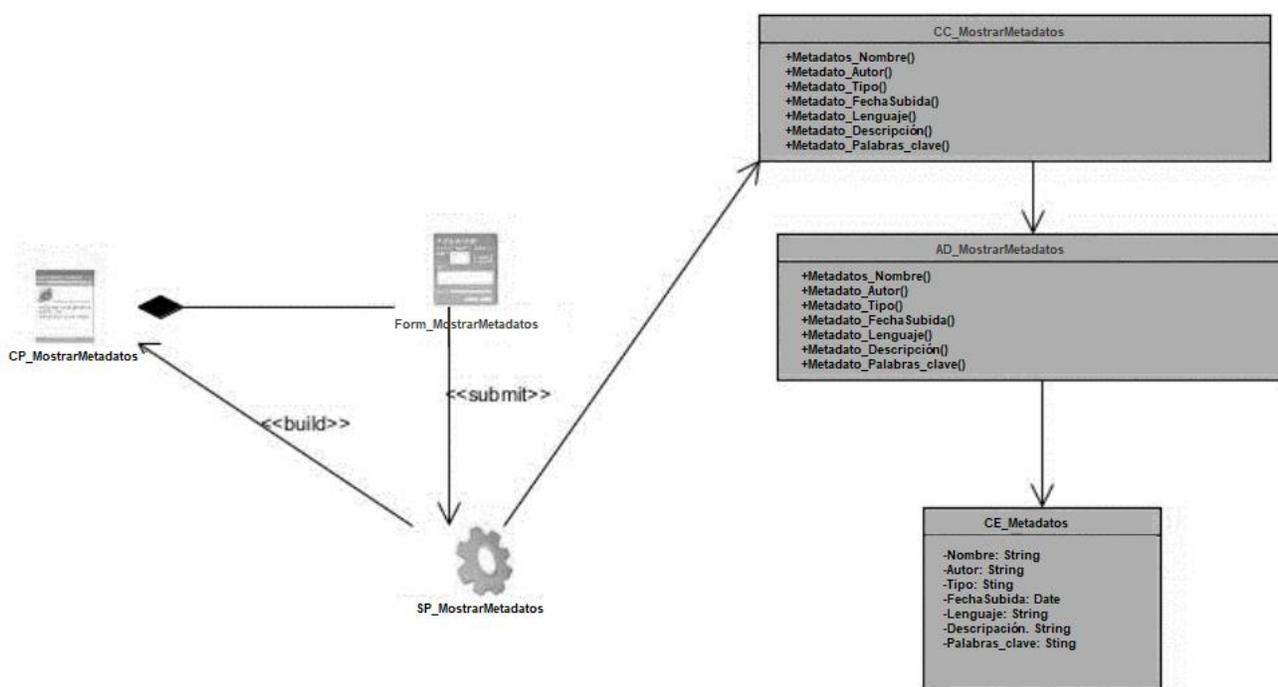


Figura 5. Diagrama de Clase del Diseño Mostrar Metadatos

II.4 Modelo de despliegue

Un diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado (UML) que muestra la arquitectura de ejecución de un sistema, utilizado para entender cómo se visualizará el hardware y el software físico de un sistema (Arango Fernández, 2017).

En la figura se muestra el diagrama de despliegue realizado a la presente solución donde quedan definidas las relaciones entre los diferentes nodos físicos que lo componen.

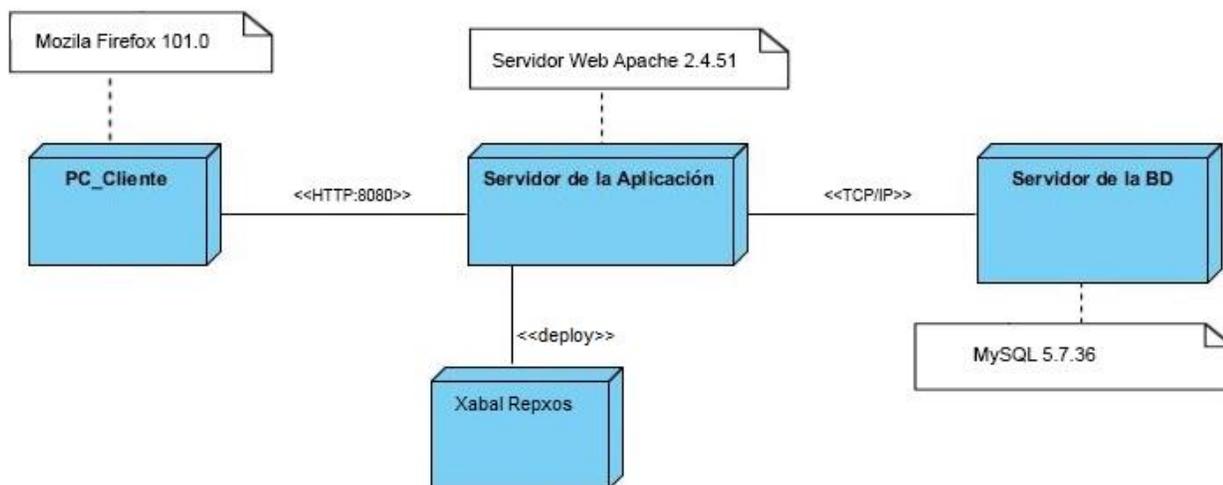


Figura 6. Modelo de Despliegue

Dispositivos

PC_Cliente:

La estación de trabajo contiene los navegadores web (Firefox, Chrome, etc.) para acceder a las páginas que interactúan con el sistema hospedado en el servidor de aplicaciones utilizando el protocolo de comunicación HTTP:8080.

Servidor de la Aplicación:

Hospeda el código fuente de la aplicación y que le brinda al usuario las interfaces para realizar los procesos del sistema. Esta estación se comunica con el servidor de base de datos donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo TCP/IP.

Servidor MySQL:

Este servidor es el encargado del almacenamiento de los datos del sistema.

Conexiones:

HTTP:8080:

Protocolo para establecer la conexión entre el dispositivo de acceso cliente y el servidor de aplicaciones.

TCP/IP:

Para establecer la conexión entre el servidor de aplicaciones y el servidor de base de datos.

Conclusiones del capítulo

Durante el análisis y diseño del módulo de integración se considera que:

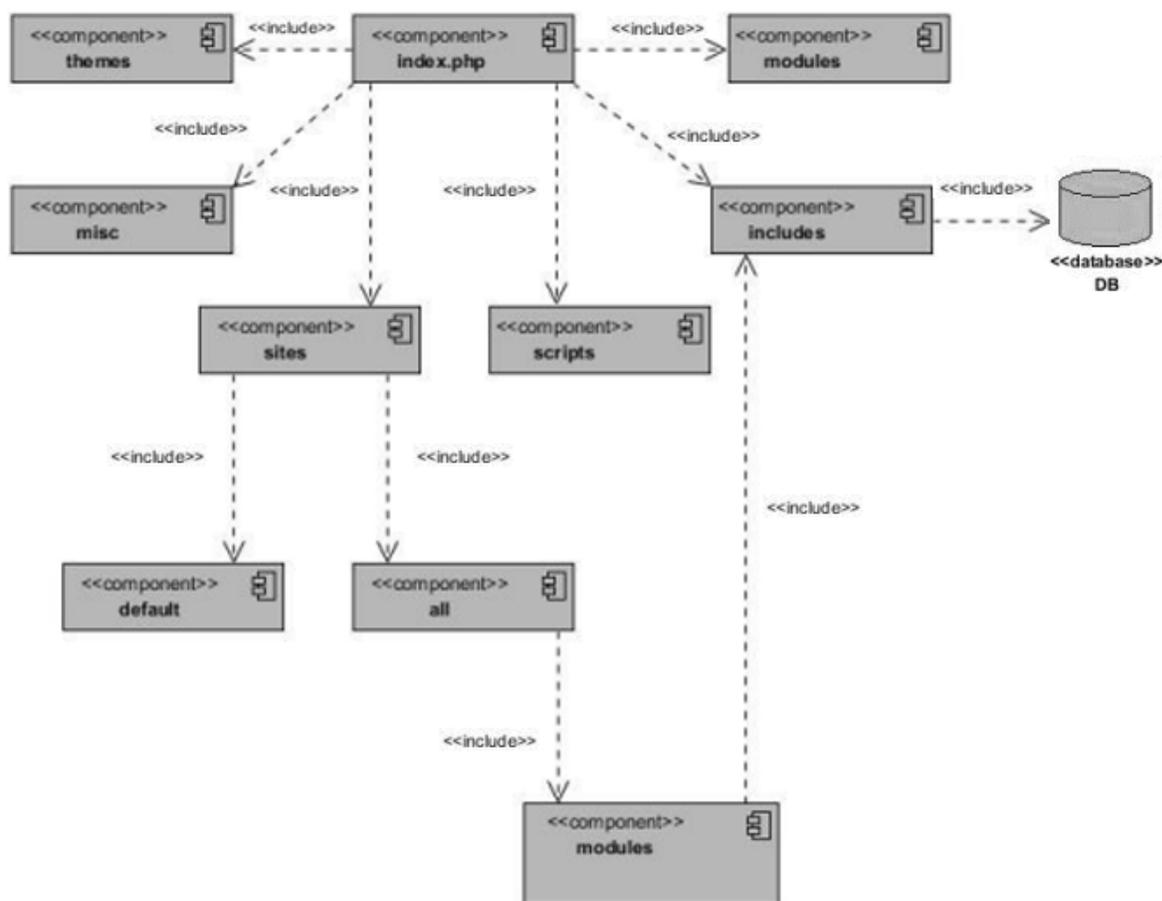
- La elaboración del modelo conceptual garantizó un mejor entendimiento del contexto del negocio a informatizar, actualmente no hay interoperabilidad entre la plataforma XABAL Repxos y Drupal, por lo que fue necesario su creación para conocer las necesidades del cliente y los usuarios finales.
- Los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos y descritos a través de las historias de usuarios, sirvieron de guía para desarrollar las distintas funcionalidades del módulo, permitiendo conocer las restricciones del mismo y sus características principales entre ellas la de mostrar metadatos de documentos científicos.
- Los artefactos generados según la metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron una guía fundamental para el desarrollo del módulo, pues se utilizaron soluciones semejantes a las de problemas comunes ya existentes.
- El modelo de despliegue propuesto permitió comprender las relaciones entre los componentes de hardware de la solución permitiendo entender el conjunto de la arquitectura del proyecto, mostrando la relación de XABAL Repxos con los elementos físicos del sistema y su despliegue.

CAPÍTULO III: VALIDACIÓN DE LA INTEGRACIÓN DE XABAL REPXOS CON DRUPAL.

En el presente capítulo se presentan los diagramas asociados, se describen los estándares de codificación, se plantea y se describe la estrategia de validación y se analizan los resultados de las pruebas realizadas que permiten evaluar la calidad de la propuesta de solución.

III.1 Diagrama de componentes

El diagrama de componentes permite ilustrar los componentes de un sistema de software conectados por las relaciones de dependencias lógicas entre cada uno de ellos, ya sean estos ficheros de código fuente, binarios o ejecutables. Brinda una vista arquitectónica de alto nivel del sistema, ayudando a los desarrolladores a visualizar el camino de la implementación. La realización del diagrama posibilita tomar decisiones respecto a las tareas de implementación y los requisitos (Leal Rondón, 2019).



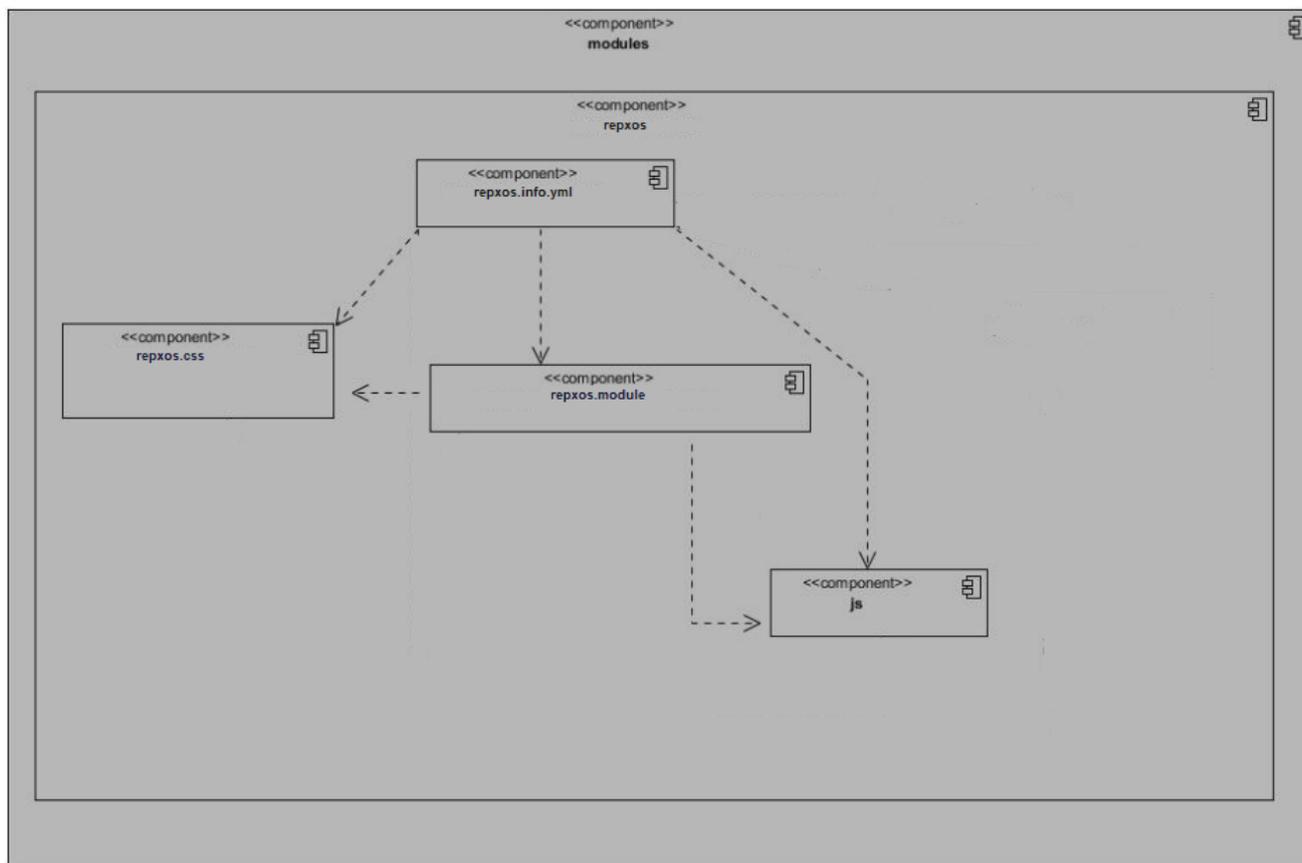


Figura 7. Diagrama de Componentes

Descripción de los componentes del diagrama

Index.php: página de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS Drupal.

Themes: conjunto de temas que representan la forma en que se visualiza el sistema, los mismos vienen por defecto en el CMS Drupal 9.

Misc: este directorio presenta un grupo de archivos necesarios para el sistema como: JavaScript y CSS e imágenes.

Includes: este directorio es importante debido a sus funciones y scripts que posibilitan el funcionamiento de los módulos y del CMS Drupal, incluye además las API de conexión a base datos y que permiten la abstracción de Drupal a la base de datos empleada.

Scripts: contiene utilidades adicionales que no utiliza Drupal directamente, pero que se pueden utilizar desde la línea de comandos de shell.

Modules: en este componente se encuentran los módulos básicos del núcleo que permiten las distintas funcionalidades de Drupal. En este directorio se encuentra la propuesta de solución.

Sites: contiene los extras y modificaciones que se añaden a la distribución original. En esta se encuentran los módulos adicionales creados, añadidos o descargados del repositorio de módulos de Drupal.

Default: contiene las clases de configuración settings.php.

Db: es la base datos de Drupal. Gestiona el acceso a la información almacenada referente al funcionamiento del sistema.

repxos.info.yml: contiene información básica sobre el módulo (nombre, descripción, versión de Drupal, requisitos mínimos, paquete al que pertenece el módulo y ficheros que incluye).

repxos.module: archivo que contiene las llamadas a las funciones hook de Drupal y algunas funciones básicas del sistema.

repxos.css: contiene el diseño visual del módulo, como se debe mostrar visualmente.

js: contiene contenido de actualización dinámica, controla multimedia y anima imágenes del módulo.

III.2 Estándares de codificación

Los estándares de codificación constituyen un principio fundamental en el desarrollo de software. Permiten que el código creado refleje un estilo armonioso, sea fácil de leer, entendible y modificado independientemente del desarrollador del producto. Contribuyen a orientar al equipo de desarrollo, permiten asegurar que el código presente calidad y no contenga errores (Álvarez Jarel, 2018).

Etiquetas de apertura y cierre PHP

Cuando PHP analiza un fichero, busca las etiquetas de apertura y cierre, que son `<?php` y `?>`, y que indican a PHP dónde empezar y finalizar la interpretación del código. En los archivos.module y .inc. se puede omitir la etiqueta de cierre siempre y cuando se trabaje solo con PHP, esto evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre `?>`), que serían identificados como salida HTML y podrían provocar un error muy típico, "Cannot modify header information-headers already sent by...". Por tanto, la etiqueta de cierre final del archivo (`?>`) es opcional en Drupal (García García, 2017).

Indentación

Consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para que el código sea más legible. En programación se utiliza la indentación para anidar elementos. En Drupal se debe indentar con 2 espacios, nunca con tabuladores. Además, no se debe dejar espacios en blanco al final de cada línea (Álvarez Jarel, 2018).

```

104  */
105  public function build() {
106  //
107  // $xml = 'modules/senit_servicio/repxos.xml';
108  // $xml = file_get_contents("xml->{$file}");
109  $file = simplexml_load_file('modules/senit_servicio/repxos.xml');
110

```

Figura 8. Indentación de la propuesta de solución

Operadores

Los operadores binarios, que se utilizan entre dos valores, deben tener un espacio de separación entre ellos. Por ejemplo, `$numero = 5`, en lugar de `$numero=5`. Esto se aplica a operadores como `+`, `-`, `*`, `/`, `=`, `==`, `!=`, `>`, `<`, `.` (Concatenación de cadenas), `.=`, `+=`, `-=`, etc. Los operadores unarios como `++`, `--` no deben tener separación. Por ejemplo, `$numero++` (García García, 2017).

```

info[] = [
  'name' => $item['name'],

```

Figura 9. Operadores de la propuesta de solución

Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres, las comillas dobles permiten interpretar el valor de una variable, entonces para `<h1>{$title}</h1>`, el resultado sería el valor de la variable `$title`. También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple (Leal Rondón, 2019).

```

'archivonombre' => $item['bitstreams'][0]['name'],
'archivoformat' => $item['bitstreams'][0]['format'],

```

Figura 10. Uso de comillas de la propuesta de solución

Uso de punto y coma (;) en código PHP

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo `<?php print $title ?>`. En Drupal es siempre obligatorio: `<?php print $title; ?>` (Leal Rondón, 2019).

```

$url = 'http://127.0.0.1:8080/rest/items/?expand=all';

```

Figura 11. Uso de punto y coma (;) en código PHP de la propuesta de solución

Estructuras de control

Con respecto a las estructuras de control, hay que tener en cuenta las siguientes normas (Álvarez Jarel, 2018):

- Debe haber un espacio entre el comando que define la estructura (if, while, for, etc.) y el paréntesis de apertura. Esto es así para no confundir las estructuras de control con la nomenclatura de las funciones.
- La llave de apertura "{" se situará en la misma línea que la definición de la estructura, separada por un espacio.
- Se recomienda emplear siempre las llaves {} aún en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control).
- Las estructuras else y elseif se escribirán en la línea siguiente al cierre de la sentencia anterior.

```
foreach ($current as $item) {
    $info[] = [
        'name' => $item['name'],
        'archivonombre' => $item['bitstreams'][0]['name'],
        'archivoformat' => $item['bitstreams'][0]['format'],
        'metadatos' => $item['metadata'],
        'autor' => $item['metadata'][0]['key'],
        'autorV' => $item['metadata'][0]['value']
    ];
}
```

Figura 12. Estructuras de control de la propuesta de solución

Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guion bajo. Siempre se debe escribir como prefijo el nombre del módulo, tema, entre otros para evitar así generar nombres duplicados. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.

En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros. Como excepción, es posible utilizar más de un espacio antes de una asignación (=) para mejorar la presentación, cuando se estén realizando varias asignaciones en bloque (Álvarez Jarel, 2018).

```
public function build() {
```

Figura 13. Funciones de la propuesta de solución

Arrays

Los valores dentro de un array (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados. Cuando se superes los 80 caracteres en la declaración del arreglo, se debe escribir cada uno de los elementos en una única línea, la indentación debe tener 2 espacios. En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector (Leal Rondón, 2019).

```
'archivonombre' => $item['bitstreams'][0]['name'],
```

Figura 14. Arrays de la propuesta de solución

Comentar el código

Para la realización de comentarios se emplea /* */ para comentarios en varias líneas y // para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, ésta deberá escribirse en mayúsculas (por ejemplo, TRUE o FALSE) (Leal Rondón, 2019).

```

/*
echo "<strong>Using direct method...</strong><br />";
$names = $file->xpath('/items/item/name');
foreach($names as $name) {
    echo "Found $name<br />";
}

if(isset($_GET['enviar'])){
    $busqueda = $_GET['busqueda'];
    $consulta = $names LIKE '%busqueda%';

    while($row = $consulta->fetch_array()){
        echo $row['$names'].'<br>';
    }
}
*/

// kint($info[0][0]['plato']);

```

Figura 15. Comentar el código de la propuesta de solución

III.3 Validación de la propuesta de solución

Los procesos de pruebas se definen como el proceso de establecer datos iniciales al producto de software e inspeccionar los resultados obtenidos, permitiendo determinar si el software funciona realmente como se espera, con la calidad y eficiencia requeridas. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa o aplicación. Las pruebas de software responden fundamentalmente a dos interrogantes, ¿se ha obtenido un buen producto?, ¿se ha desarrollado de forma correcta? Con el objetivo de comprobar el correcto funcionamiento del sistema se realizan las pruebas funcionales, de rendimiento y de integración (García Castell, 2018).

III.3.1 Funcionales

Se denominan pruebas funcionales, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. Estas pruebas permiten detectar los posibles defectos derivados de errores en la fase de programación. Para su realización es necesario diseñar un conjunto de casos de prueba que son utilizados para someter a las diferentes funcionalidades del módulo y de este modo verificar su correcta ejecución (Rodríguez Lorenzo, 2017).

A continuación, se muestra el diseño de casos de pruebas correspondiente a la funcionalidad mostrar metadatos. El resto de los casos de prueba se pueden encontrar en el Anexo 5.

Tabla 5: Variables empleadas en el caso de prueba basado en la HU Mostrar Metadatos

No	Nombre del campo	Clasificación	Descripción
V1	Metadatos	Metadatos(Puede ser tipo String, tipo Date)	Se muestran los metadatos vinculados a la tesis correspondiente.

Las celdas de la tabla contienen V (indica válido), I (indica inválido), N/S (No es necesario llenar)

Tabla 6: Caso de prueba correspondiente a la HU Mostrar Metadatos

Escenario	Descripción	V1	R/ del sistema	Flujo Central
EC 1.1 Se muestran los datos correctamente.	El sistema muestra los metadatos asociados a la tesis correspondiente.	V Test 1992	El sistema muestra los metadatos asociados a la tesis correspondiente.	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de Drupal. 2. El sistema muestra en la columna Metadatos una serie de metadatos.

<p>EC 1.2 Se muestran los datos incorrectamente</p>	<p>El sistema muestra los metadatos asociados a una tesis que no es la correspondiente.</p>	<p>I Tesis: Portal Web Facultad 1 Metadatos: Autor: Gian Michel</p>	<p>El sistema muestra los metadatos asociados a una tesis distinta.</p>
---	---	---	---

Resultado de las pruebas funcionales

Para validar que el sistema cumpla con las funciones específicas se realizaron tres iteraciones donde se encontraron 18 no conformidades, en la primera iteración se recoge un total de 13 no conformidades de las cuales todas fueron resueltas. La segunda iteración arrojó un total de 5 no conformidades siendo completamente solucionadas y para una tercera iteración y final no se detectaron no conformidades. A continuación, se muestra una gráfica donde se desglosan las no conformidades detectadas en las tres iteraciones realizadas al módulo de integración.

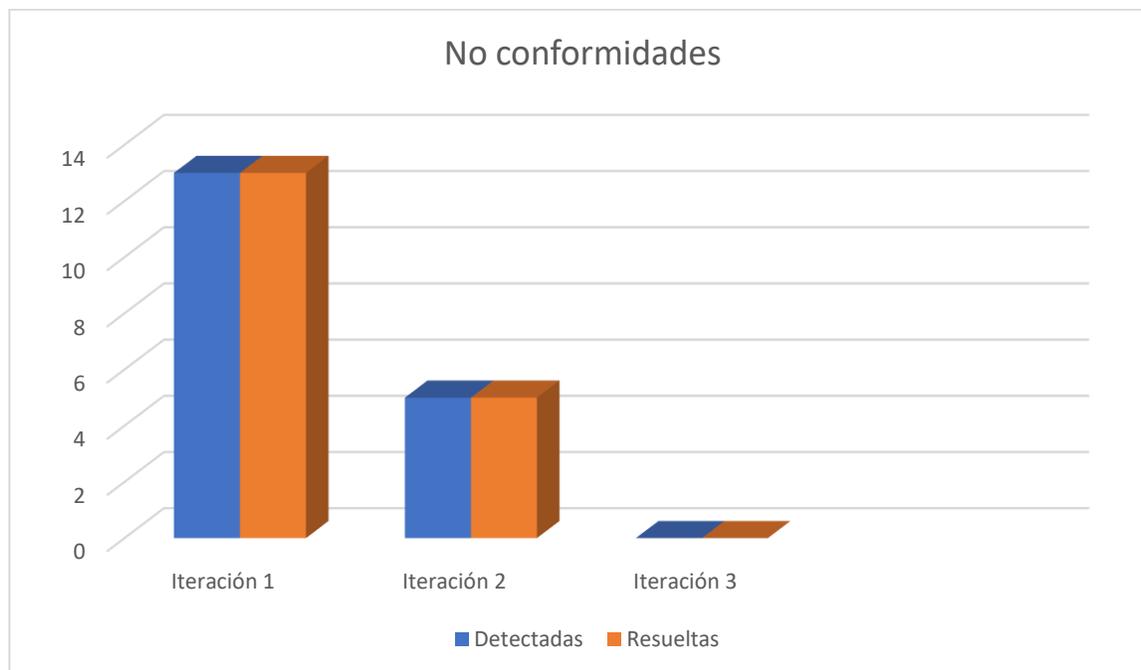


Figura 16. Comportamiento de las no conformidades por iteraciones

(Fuente: Elaboración propia)

Entre las no conformidades identificadas, los principales errores detectados se corresponden a:

- Presencia de errores ortográficos en los nombres de los campos y encabezados de las tablas. (8)
- El sistema presenta problemas de idioma. (5)
- El sistema muestra metadatos que no se corresponden con las tesis correspondientes. (5)

Las no conformidades de ortografía estaban relacionadas con problemas de acentuación, para resolverlo se corrigió en el código fuente los errores ortográficos. Para solucionar los errores correspondientes con problemas del idioma se añadió estos mensajes en el archivo de traducción del módulo. Los errores relacionados con los metadatos se creó una función que validara el acceso correcto a cada metadato correspondiente a su respectiva tesis.

III.3.2 Integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Las mismas validan que estos componentes realmente funcionan juntos, son llamados correctamente y, además, transfieren los datos correctos en el tiempo preciso y por las vías de comunicación establecidas (Suárez Larrudet, 2016).

Resultado de las pruebas de integración

Tabla 7: Resultado de las pruebas de integración

Sistema	Funcionalidades	Funcionalidad integrada	Resultado de la prueba
CMS Drupal	La integración con el CMS Drupal permite mostrar las funcionalidades del módulo.	Las funcionalidades que se integran con el CMS Drupal son: Mostrar número, Mostrar título, Mostrar Autor, Mostrar Fecha de Subida, Mostrar Metadatos. Todo esto de las tesis asociadas al repositorio institucional Xabal Repxos	El módulo se integra logrando mostrar los documentos científicos que se publican en el repositorio institucional Xabal Repxos, sin afectar las funcionalidades del sitio.

El desarrollo de las pruebas garantizó que el módulo no afecta las funcionalidades del CMS en que fue instalado, manteniendo el correcto funcionamiento del mismo y facilitando la visibilidad de los documentos científicos publicados en el repositorio institucional Xabal Repxos.

III.3.3 Rendimiento

Las pruebas de rendimiento de software se centran en determinar la velocidad con la que el sistema bajo pruebas realiza una tarea en las condiciones particulares del escenario de pruebas (Pérez Fuentes y Ramírez López, 2016).

Carga

Mediante la ejecución de las pruebas de Carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la misma (Pérez Fuentes y Ramírez López, 2016).

Estrés

Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de Usuarios y/o procesos, una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes

niveles para asegurar una mejor capacidad de concurrencia de usuarios y/o procesos que se verá reflejada en una óptima operación de negocio (Pérez Fuentes y Ramírez López, 2016).

Resultados de las pruebas de rendimiento

Para las pruebas de rendimiento se utiliza el software Apache JMeter 5.5. Para ello se definen las propiedades de la PC utilizada.

Hardware de prueba:

- Tipo de procesador: Intel(R) Core(TM) 2 i5-3380M CPU @ 2.90GHz 2.90GHz
- RAM: 8 GB DDR3.
- Tipo de Red: Ethernet 10/100Mbps.

Software instalado:

- Tipo de servidor web: Apache 2.4.51
- Plataforma: SO Windows 10 (PC cliente) (PC servidor).
- Servidor de BD: MySQL 5.7.36

Resultado de las pruebas de rendimiento

Etiqueta: El nombre de la muestra (conjunto de muestras).

Muestras: El número de muestras para cada URL (peticiones).

Mín: El mínimo tiempo (milisegundos) transcurrido para las muestras de la URL dada.

Máx: El máximo tiempo (milisegundos) transcurrido para las muestras de la URL dada.

% Error: Porcentaje de las peticiones con errores.

Rendimiento: Rendimiento medido en base a peticiones respondidas por segundo/minuto/hora.

Usuarios: Cantidad de usuarios concurrentes.

Las pruebas fueron realizadas con 60/20, 50,250, 645 y 646 muestras de datos, cuyas muestras son equivalentes a realizar 3, 50, 250, 645 y 646 peticiones HTTP respectivamente.

Se muestran a continuación las muestras de datos de 3, 645 y 646 peticiones HTTP respectivamente:

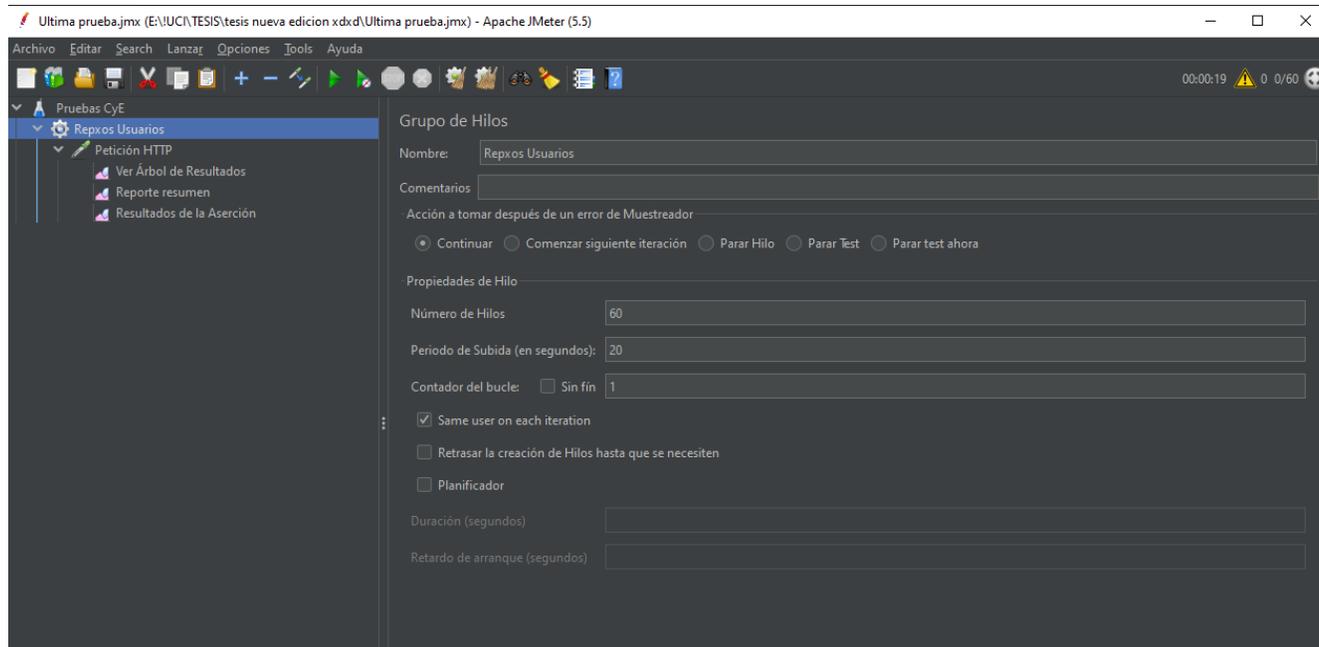


Figura 17. Representación de la configuración de las pruebas de rendimiento. Parte 1.

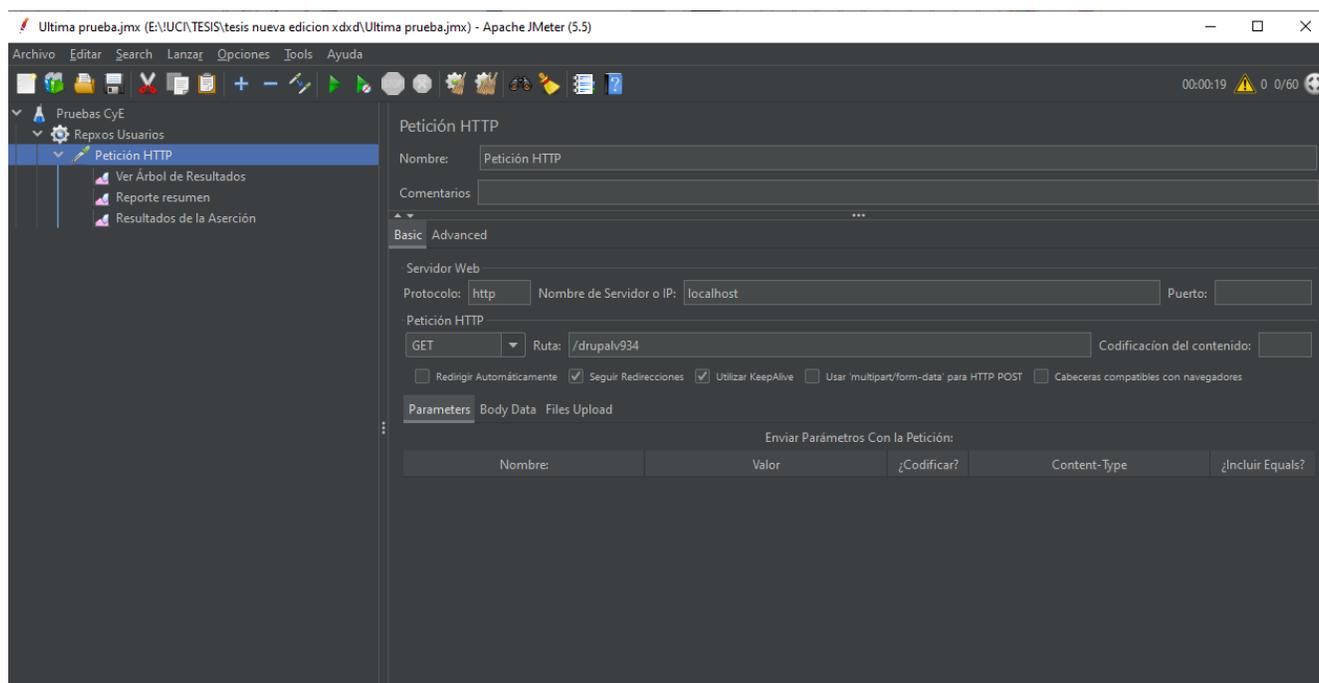


Figura 18. Representación de la configuración de las pruebas de rendimiento. Parte 2.

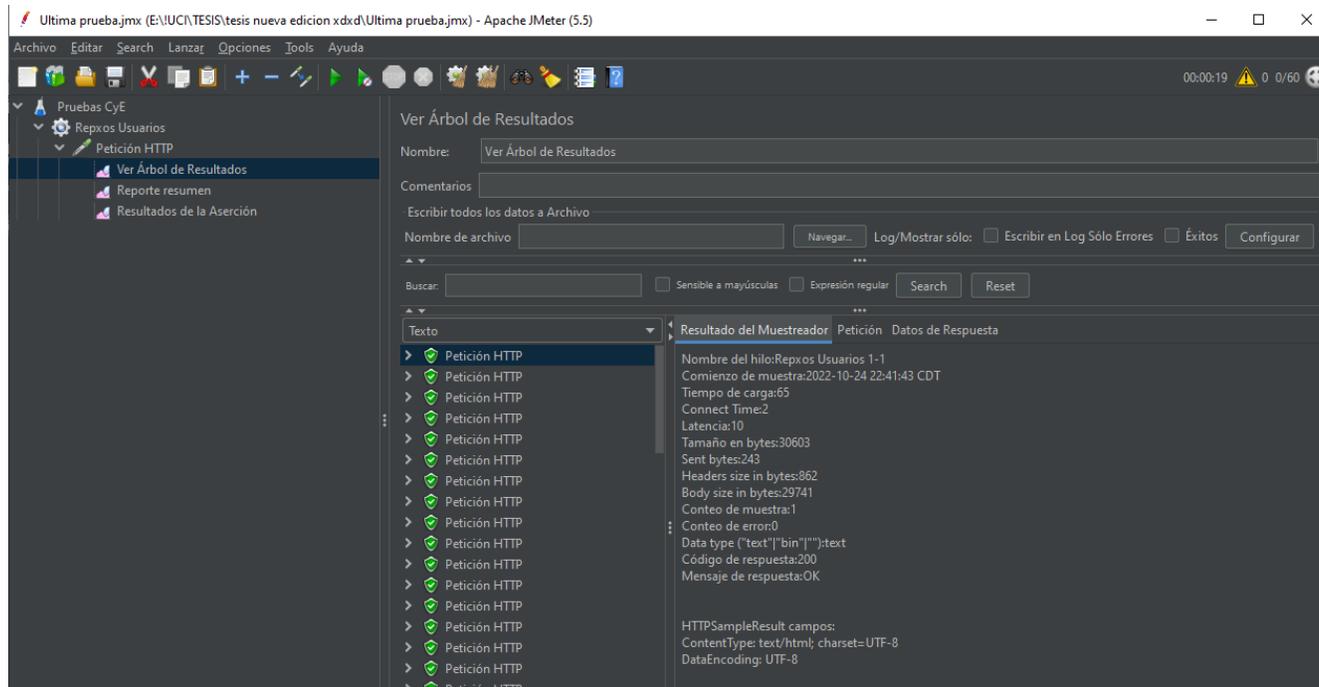


Figura 19. Representación de la configuración de las pruebas de rendimiento. Parte 3.

Al realizar la prueba con 3 muestras se ejecutó correctamente, como se muestra en la Figura 12, en un tiempo de 1 segundos como se evidencia en el siguiente cálculo:

$$\frac{\text{Muestras}}{\text{Rendimiento}} = \frac{3}{3} = 1$$

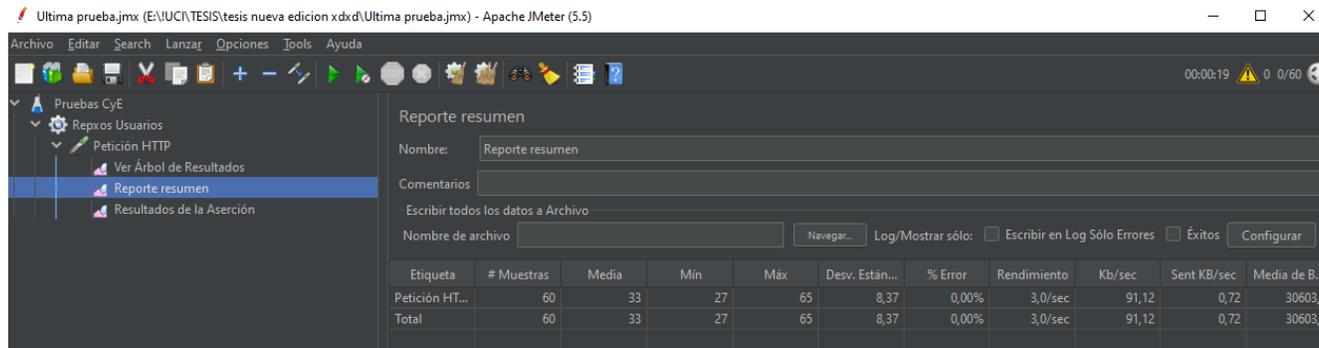


Figura 20. Representación del resultado de la prueba de rendimiento con 3 muestras.

Al realizar la prueba con 645 muestras se ejecutó correctamente, como se muestra en la Figura 13, en un tiempo de 9 segundos aproximadamente como se evidencia en el siguiente cálculo:

$$\frac{\text{Muestras}}{\text{Rendimiento}} = \frac{645}{54,2} = 8,58 (9)$$

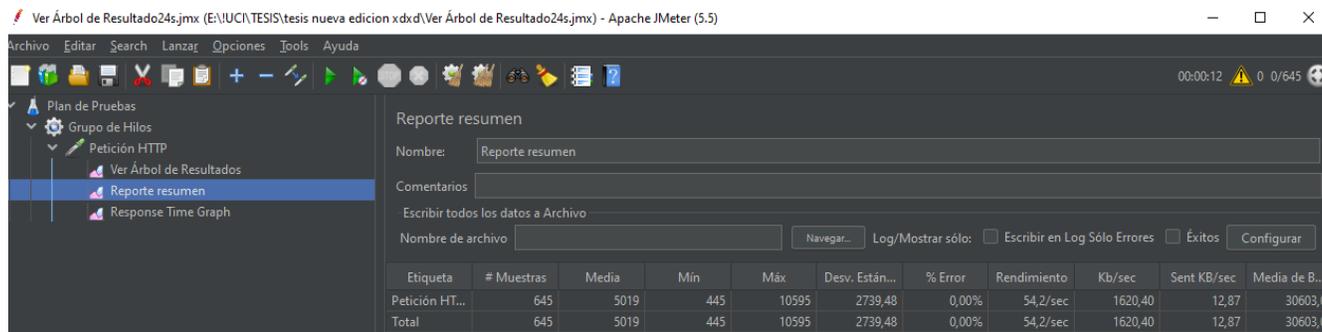


Figura 21. Representación del resultado de la prueba de rendimiento con 645 muestras.

Al realizar la prueba con 646 muestras que son representadas en la Figura 14 como número de hilos, no se pudo observar el comportamiento de la muestra, debido a que la máquina servidora dejó de responder ante una carga de 646 usuarios.

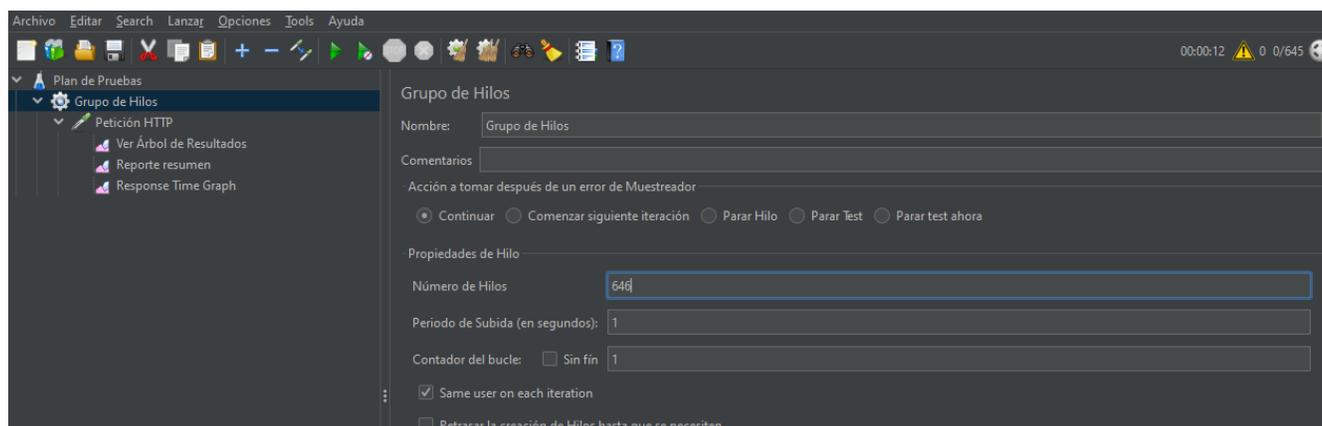


Figura 22. Representación del resultado de la prueba de rendimiento con 646 muestras.

Con los resultados obtenidos en cada prueba se demuestra que, en un ámbito de trabajo normal bajo este entorno de pruebas, puede permitir una carga de 645 usuarios concurrentemente.

III.3.4 Pruebas de aceptación

Es la prueba final antes del despliegue del sistema. Su objetivo es comprobar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue diseñado (Muñoz González, 2018). En la presente investigación se decidió realizar la prueba de aceptación ya que es necesario para la evaluación de la solución que el cliente esté de acuerdo con el funcionamiento del módulo de integración de XABAL Repxos con Drupal.

La evaluación realizada por el cliente consistió en la ejecución de pruebas funcionales. Las recomendaciones realizadas fueron solucionadas y el cliente quedó conforme.

III.3.4.1 Técnica de ladov

La técnica de ladov es utilizada para conocer el grado de satisfacción individual y grupal de los usuarios a partir de una encuesta elaborada. La aplicación de esta técnica es una vía para el estudio de satisfacción, los criterios que se utilizan se fundamentan en las relaciones que se establecen entre las tres preguntas cerradas, que se intercalan dentro de un cuestionario y cuya relación el encuestado desconoce (Leal Rondón, 2019). Para llevar a cabo dicha técnica se realizó una encuesta a especialistas del centro CIDI de la facultad 1 de la universidad de las ciencias informáticas, la cual cuenta de 3 preguntas cerradas y 2 abiertas.

Tabla 8. Cuadro lógico de ladov

3- Se siente satisfecho con el trabajo realizado y los resultados obtenidos en la propuesta de solución?	1- ¿Consideran necesario un módulo para integrar XABAL Repxos con Drupal?								
	Sí			No sé			No		
	2- ¿Considera usted necesario agregarle alguna otra funcionalidad al módulo de integración de XABAL Repxos con Drupal?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da la mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	3	4
No me gusta nada	6	6	6	6	4	4	3	3	4
No sé decir	2	3	6	3	3	3	6	6	4

Esta técnica también permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma: Máxima satisfacción (+ 1), Satisfecho (0.5), No definido (0), Insatisfecho (- 0.5) y Máxima insatisfacción (- 1). A partir de la cantidad de respuestas es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

$$ISG = \frac{A(+1) + B(0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

En esta fórmula A, B, C, D y E representan la cantidad de sujetos que interactúan con la aplicación y N representa el total de sujetos. La escala de valores del índice grupal que se toma al aplicar la técnica se encuentran entre:

- -1 y -0.5 Insatisfacción.
- -0.49 y 0.49 Contradicción.
- 0.5 y 1 Satisfacción

Los resultados obtenidos fueron los siguientes:

Tabla 9. Resultados de las escalas de satisfacción

Total de usuarios(N)	5	Escala
Máxima satisfacción	2	A
Satisfecho	2	B
No definido	1	C
Insatisfecho	0	D
Máxima insatisfacción	0	E

Como resultado del cálculo del ISG, se obtuvo:

$$ISG = \frac{2(+1) + 2(0.5) + 1(0) + 0(-0.5) + 0(-1)}{5} = 0.6$$

A partir de análisis realizado se puede concluir que el uso de la Técnica de ladov permitió mostrar el buen resultado de la propuesta de solución desarrollada, expresado cuantitativamente en el alto Índice de Satisfacción Grupal obtenido (ISG= 0.60), lo que refleja aceptación de la propuesta por parte de los encuestados. Las respuestas con respecto a las preguntas abiertas brindadas por los encuestados reafirman los beneficios que traerá el módulo de integración de XABAL Repxos con Drupal, por lo que, se puede afirmar que se cumplió el objetivo principal de la investigación desarrollada.

Conclusiones del capítulo

- La confección del diagrama de componentes permitió establecer una mayor comprensión de la arquitectura y funcionamiento de los elementos del módulo de integración de Repxos con Drupal permitiendo conocer en detalle cada componente de este.
- La realización de las pruebas de integración arrojó que el módulo funciona correctamente y que sus funcionalidades no se vieron afectadas con la integración. Los 18 errores encontrados durante las 2 primeras iteraciones realizadas en las pruebas funcionales fueron corregidos satisfactoriamente. Con la aplicación de las pruebas de carga y estrés se determinó que el módulo puede permitir una carga de 645 usuarios concurrentemente con un tiempo promedio de respuesta de 9 segundos. La técnica de loadov reafirmó los beneficios que traerá el módulo de integración de XABAL Repxos con Drupal entre ellos la agilización de los servidores de los sitios creados en Drupal.

CONCLUSIONES FINALES

En la presente investigación se realizó un proceso de desarrollo de software, con el objetivo de lograr un producto de calidad en el tiempo establecido. Una vez completada la investigación, se puede concluir que:

- El estudio de soluciones informáticas existentes, relacionadas con la presente investigación, permitió identificar funcionalidades bases para el desarrollo del módulo de integración de XABAL Repxos con Drupal entre las que se encuentran mostrar los metadatos de los documentos que se publican.
- Como resultado del proceso de diseño se logró definir el ambiente de desarrollo, así como la metodología a utilizar, lo que permitió la modelación de los artefactos que contribuyeron al diseño del módulo de integración de XABAL Repxos con Drupal.
- La obtención de los requisitos funcionales y no funcionales, garantizó que la solución respondiera a las necesidades del cliente y de los usuarios finales, sirviendo de guía para el desarrollo de las distintas funcionalidades del módulo.
- El módulo implementado facilita la obtención de documentos científicos publicados en la Universidad para una mayor recopilación de información posibilitando que el Centro de Innovación y Desarrollo para Internet (CIDI) utilice estos documentos para la creación y desarrollo de portales web que beneficien a la sociedad.

RECOMENDACIONES

1. Ampliar la investigación realizada, estudiando la influencia que ejercen las versiones del CMS Drupal y su aplicación en distintos sistemas web.
2. Continuar el decurso investigativo concentrando los esfuerzos en la creación de nuevos modelos para Drupal.

REFERENCIAS BIBLIOGRÁFICAS

- About the Apache HTTP Server Project - The Apache HTTP Server Project. [en línea], [sin fecha]. [Consulta: 4 julio 2022]. Disponible en: https://httpd.apache.org/ABOUT_APA-CHE.html.
- ACOSTA GUZMÁN, G., 2019. Sistema tutorial inteligente para el apoyo de la enseñanza de la asignatura de Teleinformática. En: Accepted: 2022-07-15T13:04:25ZCompany: Universidad de las Ciencias Informáticas. Facultad 2.Distributor: Universidad de las Ciencias Informáticas. Facultad 2.Institution: Universidad de las Ciencias Informáticas. Facultad 2.Label: Universidad de las Ciencias Informáticas. Facultad 2.publisher: Universidad de las Ciencias Informáticas. Facultad 2. [en línea], [Consulta: 9 agosto 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10284>.
- AGUDELO BENJUMEA, M.M., 2020. Los metadatos. En: Accepted: 2020-09-01T21:52:50Z [en línea], [Consulta: 4 junio 2022]. Disponible en: <http://148.202.167.116:8080/xmlui/handle/123456789/3631>.
- ÁLVAREZ JAREL, E., 2018. Módulo de revisión por pares para el Sistema de Gestión Multi-eventos. En: Accepted: 2022-02-15T15:59:20ZCompany: Universidad de las Ciencias Informáticas. Facultad 1Distributor: Universidad de las Ciencias Informáticas. Facultad 1Institution: Universidad de las Ciencias Informáticas. Facultad 1Label: Universidad de las Ciencias Informáticas. Facultad 1publisher: Universidad de las Ciencias Informáticas. Facultad 1 [en línea], [Consulta: 17 julio 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/9928>.
- Apache JMeter - Apache JMeter™. [en línea], [sin fecha]. [Consulta: 24 octubre 2022]. Disponible en: <https://jmeter.apache.org/>.
- ARANGO FERNÁNDEZ, D.L., 2017. Módulo para la publicación automática y manual de contenidos web en redes sociales para Drupal 7. En: Accepted: 2019-09-11T15:19:29Z [en línea], [Consulta: 7 agosto 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/7962>.
- CABO DÍAZ, L., OROZCO MOLINA, L., SARDIÑAS SUÁREZ, Y. y HIDALGO GUILLÉN, J., 2011. API para la publicación de las actividades colaborativas de Moodle sobre servicios web. En: Accepted: 2016-09-14T19:32:43Z [en línea], [Consulta: 9 junio 2022]. Disponible en: https://repositorio.uci.cu/jspui/handle/ident/TD_04619_11.
- CARDONA DUQUE, F.A. y CASTAÑO MARÍN, J.W., 2015. Técnicas utilizadas para la toma y elicitación de requerimientos en la ingeniería del software. [en línea], [Consulta: 9 agosto 2022]. Disponible en: <https://hdl.handle.net/11059/6110>.
- CASTAÑEDA BUENO, L., [sin fecha]. Biblioteca Digital - Universidad icesi: Laboratorio de programación - presentación - requerimientos de software. [en línea]. [Consulta: 9 agosto 2022]. Disponible en: https://repository.icesi.edu.co/biblioteca_digital/handle/10906/4084.
- DEL PINO ACOSTA, A., 2019. Módulo de planificación de las producciones para la Unión de Industrias Militares en función de la demanda. En: Accepted: 2022-07-13T12:06:31ZCompany: Universidad de las Ciencias Informáticas. Facultad 2.Distributor: Universidad de las Ciencias Informáticas. Facultad 2.Institution: Universidad de las Ciencias Informáticas. Facultad 2.Label: Universidad de las Ciencias Informáticas. Facultad 2.publisher: Universidad

de las Ciencias Informáticas. Facultad 2. [en línea], [Consulta: 9 agosto 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10276>.

Drupal - Open Source CMS. *Drupal.org* [en línea], 2018. [Consulta: 6 junio 2022]. Disponible en: <https://www.drupal.org/home>.

DUPERET CABRERA, E., PÉREZ MARTÍNEZ, D.G., CEDEÑO RODRÍGUEZ, M.Y., RAMÍREZ MUSTELIER, A. y MONTOYA ACOSTA, L.A., 2015. Importancia de los repositorios para preservar y recuperar la información. *MEDISAN*, vol. 19, no. 10, pp. 1283-1290. ISSN 1029-3019.

FEIJOO ORTIZ, Y. y GARCÍA PORTAL, L.M., 2019. Componente informático para la detección de carcinomas renales en imágenes de tomografía abdominal. En: Accepted: 2022-07-15T13:04:46ZCompany: Universidad de las Ciencias Informáticas. Facultad 2.Distributor: Universidad de las Ciencias Informáticas. Facultad 2.Institution: Universidad de las Ciencias Informáticas. Facultad 2.Label: Universidad de las Ciencias Informáticas. Facultad 2.publisher: Universidad de las Ciencias Informáticas. Facultad 2. [en línea], [Consulta: 9 agosto 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10286>.

GARCÍA CASTELL, R., 2018. *Subsistema de visualización y clasificación automática de noticias Sistema de Análisis y Clasificación Automática de Noticias (SACAN)* [en línea]. bachelorThesis. S.l.: Universidad de las Ciencias Informáticas. Facultad 1. [Consulta: 25 octubre 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/9952>.

GARCÍA GARCÍA, A., 2017. Módulo para la administración de contenidos en aplicaciones web desarrolladas con Drupal 7. En: Accepted: 2019-09-12T15:43:10Z [en línea], [Consulta: 11 septiembre 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/7970>.

GÓMEZ MONTIEL, Y., 2019. Módulo clientes del Sistema de reseña, calificación y reserva de destinos turísticos. En: Accepted: 2022-07-13T12:07:15ZCompany: Universidad de las Ciencias Informáticas. Facultad 2.Distributor: Universidad de las Ciencias Informáticas. Facultad 2.Institution: Universidad de las Ciencias Informáticas. Facultad 2.Label: Universidad de las Ciencias Informáticas. Facultad 2.publisher: Universidad de las Ciencias Informáticas. Facultad 2. [en línea], [Consulta: 9 agosto 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10279>.

HERNÁNDEZ, N.B., JIMÉNEZ, L.G.M. y GODOY, C.M., 2018. SISTEMAS GROUPWARE PARA EL DISEÑO DE DIAGRAMA DE CLASES UML EN AMBIENTES TÁCTILES. *Pistas Educativas* [en línea], vol. 39, no. 127. [Consulta: 9 agosto 2022]. ISSN 2448-847X. Disponible en: <http://itcelaya.edu.mx/ojs/index.php/pistas/article/view/1081>.

HERNÁNDEZ-REINOZA, H.J., VILLOTA-IBARRA, C., JIMÉNEZ-BUILES, J.A., HERNÁNDEZ-REINOZA, H.J., VILLOTA-IBARRA, C. y JIMÉNEZ-BUILES, J.A., 2021. Metodología lúdica para la enseñanza de la ingeniería de requisitos basada en esquemas preconceptuales. *Revista EIA*, vol. 18, no. 35, pp. 1-15. ISSN 1794-1237. DOI 10.24050/reia.v18i35.1394.

Historia | Universidad de las Ciencias Informáticas. [en línea], [sin fecha]. [Consulta: 17 noviembre 2022]. Disponible en: <https://www.uci.cu/universidad/historia>.

Ideal Modeling & Diagramming Tool for Agile Team Collaboration. [en línea], [sin fecha]. [Consulta: 7 junio 2022]. Disponible en: <https://www.visual-paradigm.com/>.

Introduction to HTML. [en línea], [sin fecha]. [Consulta: 4 julio 2022]. Disponible en: https://www.w3schools.com/html/html_intro.asp.

LABRADA DELGADO, E.J., 2019. Módulo Referencias Bibliográficas de XABAL REPXOS 4.0 en la tecnología Alfresco Community Edition 5.2. En: Accepted: 2022-04-

- 01T19:09:17ZCompany: Universidad de las Ciencias Informáticas. Facultad 2.Distributor: Universidad de las Ciencias Informáticas. Facultad 2.Institution: Universidad de las Ciencias Informáticas. Facultad 2.Label: Universidad de las Ciencias Informáticas. Facultad 2.publisher: Universidad de las Ciencias Informáticas. Facultad 2. [en línea], [Consulta: 8 junio 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10064>.
- LEAL RONDÓN, M.T., 2019. Portal web de la Facultad 1. En: Accepted: 2022-04-06T13:09:52Z [en línea], [Consulta: 11 septiembre 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10088>.
- LENZANO RAMÍREZ, J.E., 2019. Herramienta para la identificación de requisitos no funcionales de software. En: Accepted: 2022-07-15T13:04:56ZCompany: Universidad de las Ciencias Informáticas. Facultad 2.Distributor: Universidad de las Ciencias Informáticas. Facultad 2.Institution: Universidad de las Ciencias Informáticas. Facultad 2.Label: Universidad de las Ciencias Informáticas. Facultad 2.publisher: Universidad de las Ciencias Informáticas. Facultad 2. [en línea], [Consulta: 9 agosto 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10287>.
- MANOLA, N., 2022. Modernizing the Global Repository Network: European landscape. *COAR Annual Meeting 2022* [en línea]. Madrid, Spain. [Consulta: 4 junio 2022]. Disponible en: <https://zenodo.org/record/6573911>.
- MONTENEGRO, J.M., 2020. *IONIC Rest API* [en línea]. TypeScript. S.l.: s.n. [Consulta: 9 junio 2022]. Disponible en: <https://github.com/julenmelgar/ionic-rest-api>.
- MUÑOZ GONZÁLEZ, D. de la C., 2018. API REST asociada al repositorio de Nova. En: Accepted: 2022-02-15T15:27:58ZCompany: Universidad de las Ciencias Informáticas. Facultad 1Distributor: Universidad de las Ciencias Informáticas. Facultad 1Institution: Universidad de las Ciencias Informáticas. Facultad 1Label: Universidad de las Ciencias Informáticas. Facultad 1publisher: Universidad de las Ciencias Informáticas. Facultad 1 [en línea], [Consulta: 17 julio 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/9923>.
- PDF superior Módulo para la integración de la plataforma Moodle con el repositorio institucional DSpace - 1Library.Co. [en línea], [sin fecha]. [Consulta: 9 junio 2022]. Disponible en: <https://1library.co/title/modulo-integracion-plataforma-moodle-repositorio-institucional-dspace>.
- PÉREZ FUENTES, D. y RAMÍREZ LÓPEZ, A.R., 2016. *Módulo de distribución automática de tareas a nodos de transcodificación en la Plataforma XILEMA AGORAV* [en línea]. bachelorThesis. S.l.: Universidad de las Ciencias Informáticas. Facultad-6. [Consulta: 25 octubre 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/7893>.
- PHP: Documentation. [en línea], [sin fecha]. [Consulta: 7 junio 2022]. Disponible en: <https://www.php.net/docs.php>.
- PIÑEIRO, M.M., SERA, E.F.M. y PUENTE, Á. de J.P., 2015. Los repositorios de acceso abierto como alternativa para la visibilidad de la ciencia en las universidades: estudio de caso. *Revista Cubana de Información en Ciencias de la Salud* [en línea], vol. 26, no. 4. [Consulta: 3 junio 2022]. ISSN 2307-2113. Disponible en: <http://www.acimed.sld.cu/index.php/acimed/article/view/796>.
- Protocolo de evaluación de la aceptación de los repositorios institucionales por parte de los usuarios: en el marco de una colección de recursos sobre sustentabilidad energética. [en línea], [sin fecha]. [Consulta: 4 junio 2022]. Disponible en: <https://gredos.usal.es/handle/10366/139563>.

- ¿Qué es JavaScript? - Aprende sobre desarrollo web | MDN. [en línea], [sin fecha]. [Consulta: 17 julio 2022]. Disponible en: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript.
- REA-PEÑAFIEL, X.M., MANCERO-MENOSCAL, T.B., ROSERO-REA, D.C. y IMBA-QUINGO-ESPARZA, D.E., 2020. Web Services REST: una revolución en la forma de acceso a datos. *Revista Ibérica de Sistemas e Tecnologías de Informação*, no. E32, pp. 41-49. Repositorio Digital: API REST asociada al repositorio de Nova. [en línea], [sin fecha]. [Consulta: 6 junio 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/9923>.
- REPXOS 3.0 | Universidad de las Ciencias Informáticas. [en línea], [sin fecha]. [Consulta: 23 mayo 2022]. Disponible en: <https://www.uci.cu/investigacion-y-desarrollo/productos/xabal/repxos-30>.
- RODRÍGUEZ LORENZO, H.D., 2017. *Sistema para la gestión de medios de la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC* [en línea]. bachelorThesis. S.I.: Universidad de las Ciencias Informáticas. Facultad 1. [Consulta: 25 octubre 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/7952>.
- SANTANA, S., PERERO, L.R., DELDUCA, A.G. y DAPOZO, G.N., 2020. Evaluación de técnicas para la validación de requerimientos en entornos de trabajo para el desarrollo de software. *XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz)* [en línea]. S.I.: s.n., [Consulta: 7 agosto 2022]. ISBN 978-987-3714-82-5. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/103994>.
- SANTOS HERNÁNDEZ, W.D. y SERRANO PARREÑO, J.A., 2017. Desarrollo de una api rest con sus aplicaciones web y móvil para la venta de ropa online de la Empresa Roosman. En: Accepted: 2017-05-03T14:54:52Z [en línea], [Consulta: 6 junio 2022]. Disponible en: <http://www.dspace.uce.edu.ec/handle/25000/9668>.
- SILVA AVILA, A.E., LEDEZMA ZAMORA, E.G., CASTORENA PEÑA, J.A., DOMÍNGUEZ LUGO, A.J. y RIOJAS MARTÍNEZ, A., 2018. *Utilidad del Lenguaje Unificado de Modelado (UML) en el desarrollo de software profesional dentro del sector empresarial y educativo*. 2018. S.I.: Cienciacierta.
- SUÁREZ LARRUDET, D.R., 2016. *Módulo para la gestión del catálogo anual de superación profesional y servicios académicos de la Facultad 1* [en línea]. bachelorThesis. S.I.: Universidad de las Ciencias Informáticas. Facultad 1. [Consulta: 25 octubre 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/7591>.
- TRUJILLO PÉREZ, A., 2019. Módulo de búsqueda y navegación de documentos de XABAL Repxos en la tecnología de Alfresco Community 5.2. En: Accepted: 2022-04-01T19:07:47Z Company: Universidad de las Ciencias Informáticas. Facultad 2. Distributor: Universidad de las Ciencias Informáticas. Facultad 2. Institution: Universidad de las Ciencias Informáticas. Facultad 2. Label: Universidad de las Ciencias Informáticas. Facultad 2. publisher: Universidad de las Ciencias Informáticas. Facultad 2. [en línea], [Consulta: 3 junio 2022]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10063>.
- Visual Studio Code - Code Editing. Redefined. [en línea], [sin fecha]. [Consulta: 7 junio 2022]. Disponible en: <https://code.visualstudio.com/>.
- Wayback Machine. [en línea], 2017. [Consulta: 7 junio 2022]. Disponible en: <https://web.archive.org/web/20170414151225/https://www.gitbook.com/download/pdf/book/bagder/everything-curl>.

Zotero | About. [en línea], [sin fecha]. [Consulta: 7 junio 2022]. Disponible en:
<https://www.zotero.org/about/>.

ANEXOS

Anexo 1: Entrevista realizada a especialistas del proyecto Xabal Repxos y a los especialistas del centro CIDI

Objetivo: obtener información sobre el uso de Xabal Repxos y el CMS Drupal, así como requisitos funcionales y no funcionales de la propuesta de solución.

1. ¿Cree usted que es importante la integración de la plataforma Xabal Repxos con el CMS Drupal?
2. ¿Cuáles son los aspectos fundamentales a tener en cuenta cuando se consume un servicio web?
3. ¿Cree usted necesario la creación de un módulo que permita integrar estos servicios? ¿Por qué?
4. ¿Cuáles son las características que tendría este servicio?

Anexo 2: Guía de observación al proceso de uso y control de la plataforma Xabal Repxos

Observador: Gian Michel Rodríguez Pérez

Lugar: Laboratorio de la Facultad 2

Objetivo: conocer el funcionamiento del repositorio, su estructura y control.

1. ¿Qué información brinda actualmente el repositorio?
2. ¿Cómo agregar nuevo contenido al repositorio?
3. ¿Qué información contienen los archivos que se agregan al repositorio?

Anexo 3: Historias de Usuario

Historia de usuario del RF2

(Fuente: Elaboración propia)

Historia de usuario	
Número: 1	Nombre del requisito: Mostrar listado de elementos
Programador: Gian Michel Rodríguez Pérez	Iteración Asignada: 1 iteración
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: Medio	Tiempo Real: 1 semanas
Descripción: La aplicación muestra un listado de elementos al cargar los elementos que contiene.	
Observaciones: El módulo mostrará el listado del elementos cargados.	

Prototipo de Interfaz:

NO.	Nombre
1	Elemnto 1
2	Elemnto 2
3	Elemnto 3
4	Elemnto 4
5	Elemnto 5

*Historia de usuario del RF3**(Fuente: Elaboración propia)*

Historia de usuario	
Número: 2	Nombre del requisito: Mostrar Nombre elemento
Programador: Gian Michel Rodríguez Pérez	Iteración Asignada: 1 iteración
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: Medio	Tiempo Real: 1 semanas
Descripción: La aplicación muestra el nombre de cada elemento cargado.	
Observaciones: El módulo mostrará el nombre de cada elemento cargado.	
Prototipo de Interfaz:	
Título	
Sistema de Gestión de Licencias del Personal Aeronáutico del Instituto de la Aeronáutica Civil de Cuba versión 2.0	

*Historia de usuario del RF4**(Fuente: Elaboración propia)***Historia de usuario**

Número: 3	Nombre del requisito: Mostrar Autor
Programador: Gian Michel Rodríguez Pérez	Iteración Asignada: 1 iteración
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: Medio	Tiempo Real: 1 semanas
Descripción: La aplicación muestra el autor de cada elemento cargado.	
Observaciones: El módulo mostrará el autor de cada elemento cargado.	
Prototipo de Interfaz:	
<p>Autor(es)</p> <p><i>Cruz Amarán, Damaris; González Fuentes, Osay; Alayo Rondón, Alexei; Pérez Bejerano, Yunier</i></p>	

*Historia de usuario del RF5
(Fuente: Elaboración propia)*

Historia de usuario	
Número: 4	Nombre del requisito: Mostrar Fecha Subida
Programador: Gian Michel Rodríguez Pérez	Iteración Asignada: 1 iteración
Prioridad: Alta	Tiempo Estimado: 1 semanas
Riesgo en Desarrollo: Medio	Tiempo Real: 1 semanas
Descripción: La aplicación muestra la fecha en que fue subido cada elemento cargado.	

Observaciones: El módulo mostrará la fecha de subida de cada elemento cargado.

Prototipo de Interfaz:

Fecha de
publicación

jun-2019

*Historia de usuario del RF6
(Fuente: Elaboración propia)*

Historia de usuario	
Número: 5	Nombre del requisito: Mostrar metadatos
Programador: Gian Michel Rodríguez Pérez	Iteración Asignada: 1 iteración
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: Medio	Tiempo Real: 1 semanas
<p>Descripción: La aplicación muestra un elemento con sus metadatos ordenados, a partir de la interacción con el nombre del elemento. Inicialmente el usuario da click en el nombre del elemento y este muestra una nueva pestaña con los metadatos asociados al elemento ya sean: Nombre, tipo, Fecha de subida, Autor, lenguaje, descripción, entre otros.</p>	
<p>Observaciones: El módulo mostrará los metadatos del elemento correspondiente en una nueva pestaña de forma ordenada.</p>	
<p>Prototipo de Interfaz:</p>	

Metadatos

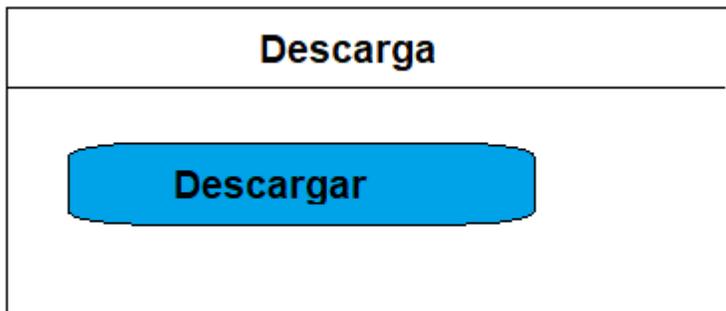
- 1-Metadato 1
- 2-Metadato 2
- 3-Metadato 3
- 4-Metadato 4

Historia de usuario del RF7
(Fuente: Elaboración propia)

Historia de usuario	
Número: 6	Nombre del requisito: Descargar elemento
Programador: Gian Michel Rodríguez Pérez	Iteración Asignada: 1 iteración
Prioridad: Media	Tiempo Estimado: 1 semanas
Riesgo en Desarrollo: Medio	Tiempo Real: 1 semanas
Descripción: La aplicación muestra un símbolo al final de cada elemento que permitirá descargar el ítem correspondiente.	

Observaciones: El módulo mostrará un símbolo que posibilitará la descarga del elemento correspondiente.

Prototipo de Interfaz:



Anexo 4: Diagramas de Clases del Diseño

Diagrama de Clase del Diseño Mostrar Listado de Elementos

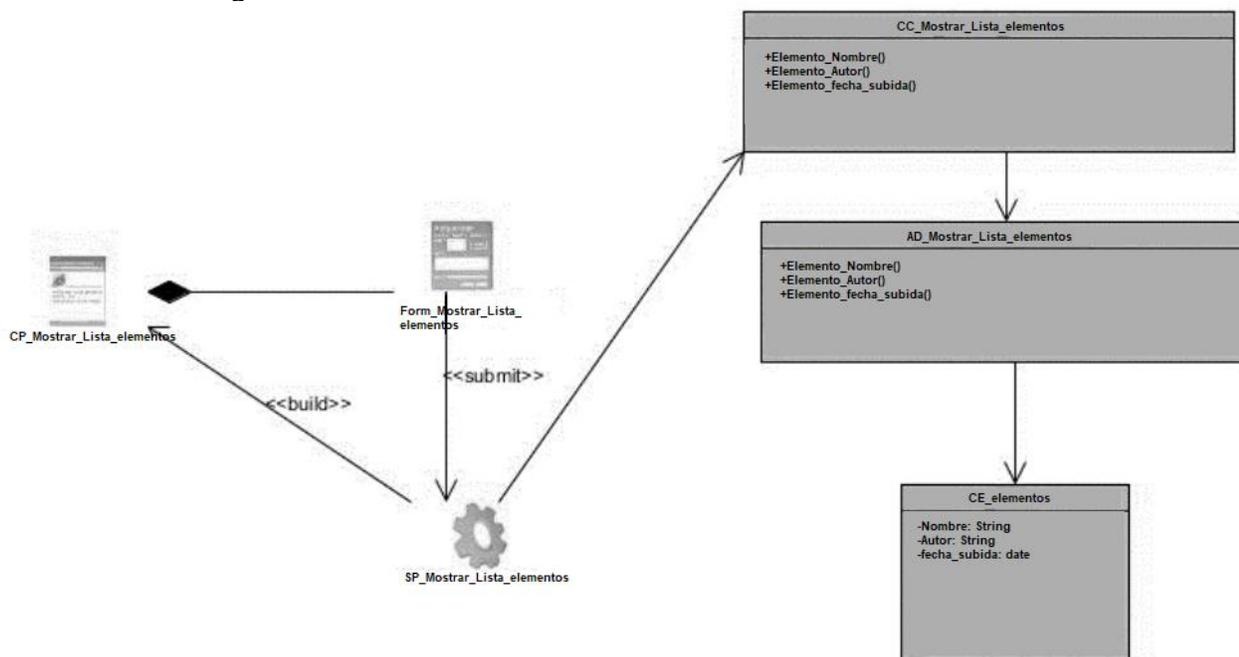
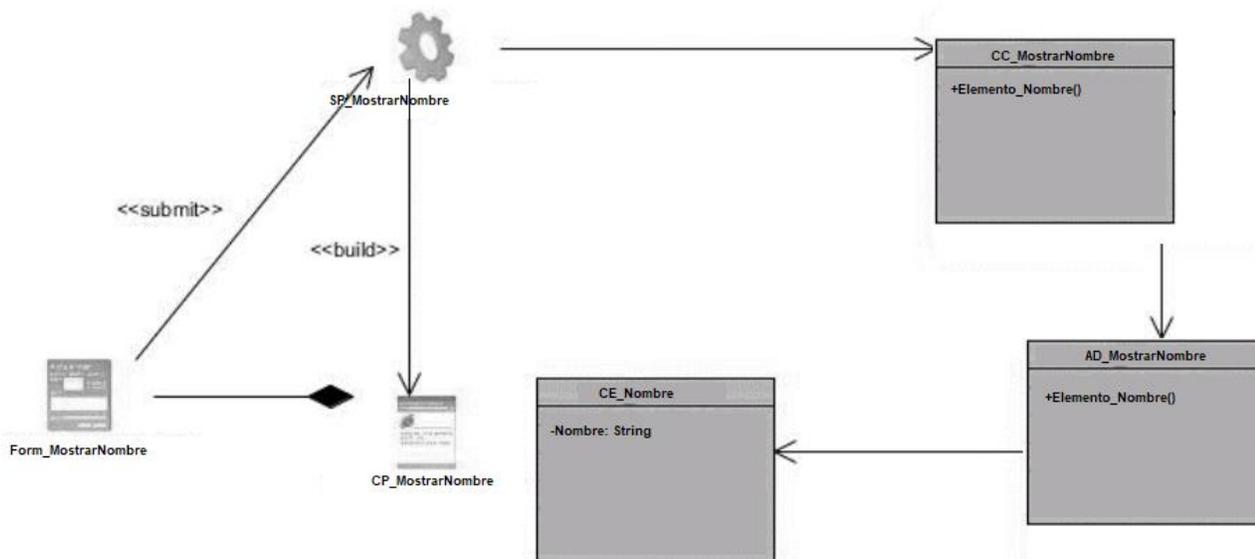


Diagrama de Clase del Diseño Mostrar Nombre



Anexo 5: Pruebas Funcionales

Variables empleadas en el caso de prueba basado en la HU Mostrar Listado de Elementos

No	Nombre del campo	Clasificación	Descripción
V1	Elementos	Elementos (Puede ser tipo String)	Se muestran los elementos publicados.

Las celdas de la tabla contienen V (indica válido), I (indica inválido), N/S (No es necesario llenar).

Caso de prueba correspondiente a la HU Mostrar Listado de Elementos

Escenario	Descripción	V1	R/ del sistema	Flujo Central
EC 1.1 Se muestran los datos correctamente.	El sistema muestra los elementos publicados.	V Test (Nombre del elemento)	El sistema muestra el listado de elementos publicados.	3. El usuario accede a la página principal de Drupal. 4. El sistema muestra en la columna Nombre los elementos publicados en una columna.

<p>EC 1.2 No se muestran elementos publicados</p>	<p>El sistema no muestra los elementos publicados.</p>	<p>(Vacío)</p>	<p>El sistema no muestra el listado de elementos publicados (Se queda en blanco).</p>
---	--	----------------	---

Anexo 6: Pruebas de Rendimiento

Al realizar la prueba con 50 muestras se ejecutó correctamente, en un tiempo de 1,08 segundos como se evidencia en el siguiente cálculo:

$$\frac{\text{Muestras}}{\text{Rendimiento}} = \frac{50}{46,3} = 1,08$$

Rendimiento 46,3

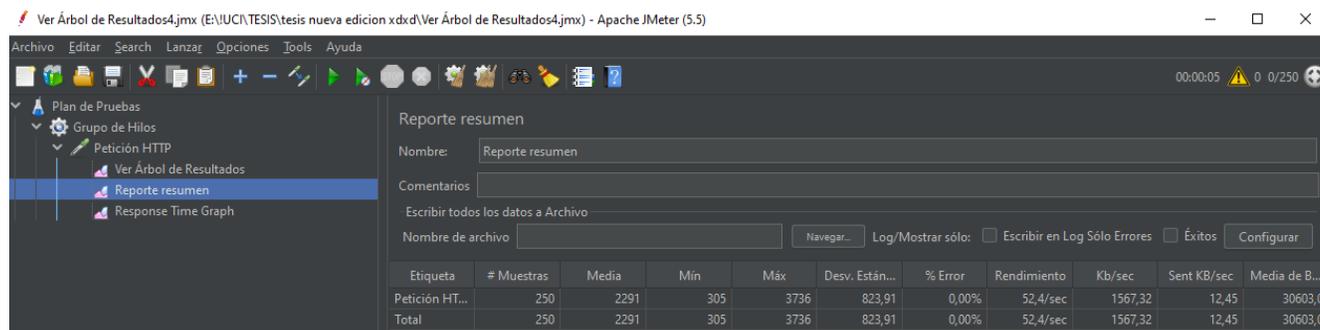


Representación del resultado de la prueba de rendimiento con 50 muestras.

Al realizar la prueba con 250 muestras se ejecutó correctamente, en un tiempo de 4,8 segundos como se evidencia en el siguiente cálculo:

$$\frac{\text{Muestras}}{= 250} = 4,8$$

Rendimiento 52,4



Representación del resultado de la prueba de rendimiento con 250 muestras.