



Facultad de Ciencias y Tecnologías Computacionales

“Revista Cubana de las Ciencias Informáticas 2.0”

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

Autor: Rosaida López Villavicencio

Tutor: Ing. Yosbel Falero Vento

Co-tutor: Ing. Bernardo Hernández González

La Habana, diciembre de 2022

“Año 64 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Rosaida López Villavicencio, con carné de identidad 99032403130 soy autora de este trabajo y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmo la presente declaración jurada de autoría a los ____ días del mes de diciembre del año 2022.

Rosaida López Villavicencio

Firma del Autor

Ing. Yosbel Falero Vento

Firma del Tutor

Ing. Bernardo Hernández González

Firma del Tutor

DATOS DE CONTACTO

Generales del tutor

Nombre y apellidos: Yosbel Falero Vento.

Centro de trabajo: Universidad de las Ciencias Informáticas.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Cargo: Director de Informatización de la UCI y Jefe del Proyecto que desarrolla la Plataforma Picta.

Categoría docente: Profesor Instructor.

Año de graduación: 2007

Teléfono Móvil: 52189354

Correo electrónico: yfalero@uci.cu

Generales del co-tutor

Nombre y apellidos: Bernardo Hernández González.

Centro de trabajo: Universidad de las Ciencias Informáticas.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Categoría docente: Profesor Asistente.

Año de graduación: 2015

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Teléfono Móvil: 55508118

Correo electrónico: bhernandez@uci.cu

AGRADECIMIENTOS

A mami por consentirme hasta de más, por ser la mujer más fuerte que conozco y heredarme su carácter y su enseñanza, a ti que sin tener nada me has dado todo. Eres lo más preciado que tengo en la vida, te debo todo lo que soy ahora. Te amo.

A papa por tanto amor y sacrificio, por tus consejos, gracias por engendrar en mí la bondad y la sensibilidad, por creer en mí cuando yo no lo hacía, por ser mi confidente, mi fuente de la sabiduría, mi amigo, el amor de mi vida.

A mi hermano Osbel por ser esa persona valiente y emprendedora que lucha por sus sueños y se sacrifica hasta lograrlos. Por tu alegría, por tus celos, por tus berrinches, por ser una mejor versión de mí misma. Te admiro mucho.

A Juanki por tanta paciencia, por ser familia, amigo, maestro, tutor y ejemplo, por estar de lleno para mí en este proceso tan difícil, gracias por enseñarme lo que es la verdadera pasión por la programación, por abrirme las puertas de tu casa y hacerme sentir parte de la familia tan hermosa que has creado. Las palabras no me alcanzarían para agradecer y lo sabes.

A Ekaterina Ramírez por los chistes, las pelis, por toda la ayuda, gracias por estar para mí.

A mi tía Aida, mi ídolo, mi faro, guerrera incansable, mi segunda madre. Aquí está mi título... lo prometido es deuda.

A Rafaela Felina y Segundo, mis abuelos de crianza, por la entrega desmedida, los amo con todo el corazón.

A mis tres tesoros María Lucía, Verónica y Alejandro, gracias a los tres por sacarme de muchos baches desde su inocencia y con un amor sincero.

A Adriano porque me conoces más que a mí misma y supiste entenderme más que nadie, por apoyarme en todo momento y ser parte de este proceso.

A mis tutores Yosbel y Bernardo por las enseñanzas y por darme todo el apoyo que estuvo en sus manos.

A Yanelis Benítez, tus alumnos te estaremos siempre agradecidos.

A Ray gracias por fomentar mi disciplina en el estudio, gracias por estar, antes, durante y después.

A Lisbeth por tus mensajitos de, "ya te queda poco", "dale que tú puedes", vales oro amiga.

A la familia que escogí, mis amigos de la Engels, gracias por estar a pesar de los años y la distancia.

A mis amigos de la universidad, ya veo nuestros futuros separados, pero están en mí por siempre. A Leydis Mariam, Naissa y su familia, Claudia, Amanda, Danay, Daylen, Ale, Alfredo, Brian, Javier García, Rixon, Álvaro, Jesús Alejandro, Silvilanio Patricio, Jainy. Por estar presente en mi crecimiento personal. Gracias por las risas, las fiestas, pero también por las lágrimas y desvelos, por las noches de estudio, por los cafés de madrugada y las canciones a viva voz. Gracias por las anécdotas y por tanto amor. Gracias a todos

DEDICATORIA

A mis padres Raquel y Oscar por tanto sacrificio, por enseñarme todo lo que se y hacerme crecer como persona.

A mi hermano Osbel por ser mi ejemplo para seguir.

A Juan Carlos por las noches de desvelo y dedicación, por no dejarme derrumbar ni un solo segundo.

A todas las personas que de una forma u otra han puesto su granito de arena para que este sueño fuera posible.

RESUMEN

Actualmente las Tecnologías de la Información y las Comunicaciones han logrado un avance considerable, por lo que la información es uno de los factores más importantes dentro de la sociedad. La comunicación científica ha estado en constantes transformaciones que afectan de manera intensa al propio modelo y a su principal vehículo que son las revistas. La Universidad de Ciencias Informáticas junto a otros organismos bajo el sello editorial de Ediciones Futuro crearon la Revista Cubana de Ciencias Informáticas, la cual se dedica a la publicación periódica de trabajos científicos especializados que reflejen los avances de las Ciencias Informáticas y de la Computación a nivel nacional e internacional. Dicha revista con el transcurso del tiempo ha ido perdiendo visibilidad en los buscadores académicos internacionales. Teniendo en cuenta lo anterior se planteó el desarrollo de la versión 2.0 de la revista que cuenta con un nuevo diseño visual actualizado, montada en la última versión del Sistemas de Gestión de Revistas Científicas Electrónicas utilizado, además, de contar con la mayoría de la información contenida en la versión anterior. Por este motivo se hace necesario desarrollar una herramienta de migración de datos que consiga migrar al menos la información más importante de la revista y de esta forma cumplir con el objetivo del proyecto. Se describen los estilos arquitectónicos y patrones de diseños utilizados en la implementación de la solución propuesta. Y se evalúa por separado el correcto funcionamiento de los códigos que componen la aplicación mediante la validación de este, obteniendo resultados satisfactorios.

PALABRAS CLAVE

Revista, información, migración de datos, implementación.

ABSTRACT

Nowadays, Information and Communication Technologies have made considerable progress, making information one of the most important factors in society. Scientific communication has been undergoing constant transformations that have an intense effect on the model itself and its main vehicle, which are the journals. The University of Computer Science, together with other organizations under the publishing house Ediciones Futuro, created the Cuban Journal of Computer Science, which is dedicated to the periodical publication of specialized scientific works that reflect the advances in Computer Science and Computing at a national and even international level. Over time, this journal has been losing visibility in international academic search engines. Taking into account the above, the development of version 2.0 of the journal was proposed, which has a new updated visual design, mounted on the latest version of the Electronic Scientific Journals Management System used, in addition to having most of the information contained in the previous version. For this reason, it is necessary to develop a data migration tool that manages to migrate at least the most important information of the journal and thus meet the objective of the project. The architectural styles and design patterns used in the implementation of the proposed solution are described. And the correct functioning of the codes that make up the application is evaluated separately by means of acceptance tests, obtaining satisfactory results.

KEY WORDS

Magazine, information, data migration, implementation.

TABLA DE CONTENIDOS

Contenido

INTRODUCCIÓN.....14

CAPÍTULO I: FUNDAMENTOS TEÓRICO-METODOLÓGICOS SOBRE LOS SISTEMAS PARA LA GESTION EDITORIAL DE LA RCCI.....21

 I.1 Conceptos asociados al problema..... 21

 I.1.1 Publicación electrónica:..... 21

 I.1.1.2 Las ventajas de las publicaciones electrónicas son: (Travieso Aguiar 2003)..... 22

 I.1.1.3 Revistas académicas..... 25

 I.1.2 Revistas científicas..... 26

 I.1.2.1 Indexación e indización de revistas científicas..... 27

 I.2.2 Algunos ejemplos de revistas científicas electrónicas cubanas son las siguientes:..... 29

 I.2.2.3 Revista Cubana de Ciencias Informáticas..... 29

 I.3 Sistemas para la gestión de revistas académicas..... 31

 I.3.1 Open Journal System (OJS)..... 33

 I.4 Herramientas y metodologías..... 35

 I.4.1 Proceso Unificado Abierto..... 40

CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA REVISTA CUBANA DE CIENCIAS INFORMATICAS 2.0.....42

 II.1 Propuesta de solución..... 42

 Modelo de dominio..... 42

 II.2 Definición de los casos de uso del sistema..... 46

 II.3 Patrones de Casos de Uso utilizados..... 51

 II.4 Arquitectura de la solución..... 52

 Conclusiones del capítulo..... 57

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA.....58

 III.1 Modelo de implementación..... 58

 III.1.1 Diagrama de Componentes..... 58

 III.1.2 Modelo de datos..... 59

 III.1.3 Diagrama de Despliegue..... 60

 III.2 Migración de datos..... 61

 Descripción del mapeo: Tanto en la tabla *publications* como en la tabla *articles* existen los mismos campos. Se hizo coincidir la relación entre las mismas ya que se refieren a los mismos datos..... 63

 III.2.1 Estándares de codificación..... 63

 III.3 Pruebas..... 64

 III.3.1 Pruebas unitarias, camino básico..... 65

 III 3.2 Pruebas de caja negra..... 68

 III.3.3 Pruebas de Aceptación..... 71

 III.3.4 Pruebas de rendimiento..... 72

 Conclusiones del capítulo..... 73

 CONCLUSIONES FINALES..... 74

RECOMENDACIONES.....75

ÍNDICE DE TABLAS

Tabla 1 "Definición de los actores del sistema"	42
Tabla 2 "Descripción textual del Caso de Uso del Sistema"	43
Tabla 3 "Descripción de los nodos correspondientes al Diagrama de Despliegue"	57
Tabla 4 "Ejemplo de relación entre tablas y atributos"	58
Tabla 5 "Secciones a probar en el Caso de Uso del Sistema Gestionar publicación"	64
Tabla 6 "Descripción de las Variables"	66

ÍNDICE DE FIGURAS

Figura 1:"Diagrama de modelado del dominio"	42
Figura 2 " Diagrama de Casos de Uso del Sistema"	45
Figura 3 "Patrón Arquitectónico Modelo vista Controlador"	51
Figura 4 "Diagrama de clases del diseño gestionar número"	52
Figura 5 "Patrón Experto"	53
Figura 6 "Patrón Creador"	53
Figura 7 "Patrón Creador"	54
Figura 8 "Patrón Decorador"	54
Figura 9 "Diagrama de Componentes"	57
Figura 10 "Diagrama entidad relación del producto de trabajo"	58
Figura 11 "Diagrama de despliegue"	59
Figura 12 "Relación de las bases de datos implícitas en la migración"	60
Figura 13 "Fragmento de código del método migratePublicationsAction ()"	64
Figura 14 "Grafo de flujo del método migratePublicationsAction ()"	65
Figura 15 "Representación de los resultados de prueba"	69
Figura 16 "Prueba de carga utilizando JMeter"	70

OPINIÓN DEL(OS) TUTOR(ES)

<Contenido de la opinión de los tutores>

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada>

Introducción

INTRODUCCIÓN

Las TIC que corresponden a las siglas de Tecnologías de la información y Comunicaciones, constituyen herramientas necesarias para transformar la información con el uso de dispositivos electrónicos y programas que elaboran, realizan cambios, guardan, protegen y recuperan dicha información, que hoy por hoy es usada y compartida por la sociedad a través de la gran red del Internet (Ulco Simbaña and Baldeón Egas 2020).

El desarrollo acelerado de las telecomunicaciones e informática en la segunda parte del siglo pasado y el surgimiento y proliferación de Internet en la década pasada, ha influenciado prácticamente en todos los campos del quehacer humano. La digitalización se esparció en todo el mundo; ahora no sólo es importante reducir la separación entre “los que tienen” y “los que no tienen”, es fundamental reducir también la separación que Internet acentúa entre los que saben y los que no saben. El impacto de la digitalización en la sociedad es evidente y cada vez más amplio (Díaz Lazo, Pérez Gutiérrez et al. 2011).

Los cambios y transformaciones de la sociedad exigen de las personas mayores niveles de desempeño, en tal sentido se requiere el dominio de nuevas cualidades, capacidades y competencia para enfrentar con solvencia los problemas de la vida cotidiana, de ahí que la implementación de las TIC en los distintos procesos de formación, constituye un aspecto de mucha importancia más aun cuando esta herramienta posibilitaría y facilitaría el desarrollo de nuevas habilidades. Asumiendo que las TIC ponen al alcance de las personas información suficiente con respecto a los distintos avances en materia de conocimiento, es preciso fortalecer el uso de estas herramientas (Ulco Simbaña and Baldeón Egas 2020).

La comunicación científica es el conjunto de procesos de presentación, transmisión e intercambio de información científica en la sociedad humana; constituye el mecanismo principal de existencia y desarrollo de la ciencia (Borges 2017). Durante las últimas décadas, la comunicación científica ha estado en constantes transformaciones que afectan de manera intensa al propio modelo y a su principal vehículo que son las revistas. La digitalización, la concentración de la edición en manos de unos cuantos grupos editoriales y el acceso abierto son los tres hitos más impactantes a nivel global y se hacen presentes en fenómenos como el crecimiento exponencial del número de revistas científicas. Además, la aparición de nuevos modelos del negocio y tarifas, al tiempo que también se aprecia una alta disponibilidad de información científica de manera gratuita (Vázquez 2018).

El número de revistas científicas latinoamericanas y españolas indexadas en bases de datos internacionales ha sido siempre limitado en comparación con el número total de revistas científicas publicadas en los países primermundistas principalmente porque las mismas no están disponibles en inglés.

En los últimos veinte años Perú, Ecuador y Colombia, son los países que más han aumentado sus

Introducción

publicaciones en América Latina. El crecimiento en su producción científica fue fluctuante hasta antes del año 2013. A partir del año 2014 inició un crecimiento constante que coincidió con la implementación de la nueva Ley Universitaria; a través de la cual se establecieron criterios para que las universidades consideraran a la investigación como una necesidad, siendo un factor importante para que las avalen o no en su funcionamiento (Carvajal-Tapia and Carvajal-Rodríguez 2018).

Particularmente, Cuba ha defendido siempre el concepto de que el uso masivo de las TIC no es un fin sino una herramienta poderosa para lograr el progreso. El proyecto de desarrollo económico y social llevado a cabo por el pueblo cubano, lo coloca en una situación ventajosa para emprender el reto de la informatización y el tránsito hacia una sociedad basada en el conocimiento.

A pesar de todas las dificultades económicas a las que se enfrenta, no se ha quedado atrás en cuanto a la implementación y desarrollo de las TIC dada la voluntad política del gobierno. Ejemplo de ello lo es el uso de las mismas en todos los niveles de enseñanza. La creación del Ministerio de las Comunicaciones (MINCOM) ha sido una estrategia vital para el desarrollo de las TIC en Cuba. La Universidad de las Ciencias Informáticas (UCI) tiene como objetivo fundamental la formación de recursos humanos en este campo.

En dicho escenario es donde entra la UCI junto a otros organismos (MES, Ministerio de Educación Superior), MINCOM, entre otros, se unen para crear un espacio publicitario para las Ciencias Informáticas y la Computación, surgiendo así la Revista Cubana de las Ciencias Informáticas (RCCI) que a través del sello editorial Ediciones Futuro, publica los contenidos de la misma.

Ediciones Futuro se proyecta como un sello editorial de referencia en el campo de las Ciencias de la Computación e Informática, que publica y divulga la producción científica de profesionales cubanos y extranjeros vinculados a las actividades de formación, investigación y desarrollo, propiciando un espacio para la visibilidad y la difusión de sus resultados científicos. Esta tiene como objetivo garantizar el funcionamiento del sistema editorial de la UCI.

Esta publicación no es el primer espacio surgido en Cuba para la discusión, el debate y la socialización de los resultados investigativos en el campo de las Ciencias Informáticas y de la Computación. Otras publicaciones nacionales han sido pioneras en ello, vale la pena mencionar entre las precursoras a la revista Investigación Operacional de la Universidad de La Habana, fundada en 1966. Otros intentos como la Revista CID. Electrónica y procesos de datos en Cuba, desaparecieron en medio de la difícil situación económica de los '90. No obstante, la RCCI es el primer esfuerzo editorial que se propone, de manera sistemática e intensa, la publicación periódica de trabajos científicos especializados que reflejen los avances de las Ciencias Informáticas y de la Computación a nivel nacional, e incluso internacional (Melchor 2006).

Introducción

La RCCI tuvo su primer número en diciembre del 2006, está indizada en importantes repositorios de información científico-técnica como Scielo, Redalyc, DOAJ y Latindex como resultado del riguroso trabajo desarrollado. La misma cuenta con un sistema de arbitraje por pares académicos y periodicidad trimestral.

Esta tiene como público, fundamentalmente, a la Comunidad Científica de las Ciencias Informáticas y Ciencia de la Computación, estudiantes, profesores e investigadores de estas ramas en Cuba y en el extranjero. Aunque se distribuye a todos los autores, árbitros o revisores, editores asociados, bibliotecas universitarias del MES y el MINED (Ministerio de Educación), Joven Club de Computación y Electrónica, organismos, instituciones y personalidades, esa cifra no permite toda la visibilidad que sus contenidos deben tener.

Existe una situación desfavorable en cuanto al alcance y visibilidad que los contenidos de la RCCI pueden tener, considerando la necesidad de hacer llegar conocimiento e información científica sobre las investigaciones en las ramas de las Ciencias Informáticas, Ciencia de la Computación y otras afines, a la comunidad científica cubana; y considerando además la importancia de proporcionarle a la revista presencia en Internet con una mayor diseminación de los contenidos de la edición impresa, e incrementando su prestigio y reconocimiento nacional e internacional.

Desde que se creó la RCCI no se ha actualizado, lo que ha generado que la revista pierda posicionamiento web a nivel internacional y no está dentro de las primeras revistas en los buscadores académicos. Con esto surge la necesidad de desarrollar una nueva versión del sistema para la gestión del flujo editorial de dicha revista. La propuesta debe desarrollarse en una de las últimas versiones de los sistemas para la gestión editorial de revistas científicas reconocidos, lo cual posibilitará indexar la revista en bases de datos internacionales que implican este requerimiento para completar el proceso.

La implementación de este software permite que las revistas puedan posicionarse e indizarse en las bases de datos de manera más transparente y estandarizada en sistemas como Latindex Catálogo 2.0, Erih Plus, DOAJ, Dialnet, MIAR, REDIB, Google académico y SCIELO.

Es precisamente por la situación problemática antes descrita que **el problema científico** a resolver es: ¿Cómo optimizar la visibilidad de la gestión editorial de la Revista Cubana de Ciencias Informáticas?

Por lo que se plantea que **el objeto de estudio** son los sistemas para la gestión editorial de revistas científicas.

De aquí se deriva que **el campo de acción** es la gestión editorial de la Revista Cubana de Ciencias

Introducción

Informáticas.

Se plantea entonces como **objetivo general** diseñar una nueva versión digital de la Revista Cubana de Ciencias Informáticas que permita mayor visibilidad de los resultados de la investigación científica en la rama de las Ciencias Informáticas y la Computación.

Para guiar el proceso investigativo se plantean las siguientes **preguntas científicas**:

1. ¿Cuáles son los principales fundamentos teóricos metodológicos relacionados con las revistas científicas?
2. ¿Qué técnicas, tecnologías y herramientas de desarrollo se podrían utilizar en la implementación de la nueva versión de la Revista Cubana de Ciencias Informáticas?
3. ¿Qué técnicas de pruebas se pueden utilizar en la validación de la propuesta de solución y cómo se deben aplicar?

Para dar cumplimiento al objetivo general se definen las siguientes **tareas de investigación**:

1. Elaboración del marco teórico referencial relacionado con las revistas científicas en línea.
2. Seleccionar las principales tecnologías y herramientas que se ajustan a la situación real del negocio para el desarrollo de la nueva versión de la Revista Cubana de Ciencias Informáticas.
3. Elaboración del diseño de la nueva versión de la Revista Cubana de Ciencias Informáticas.
4. Realización de la implementación de la nueva versión de la Revista Cubana de Ciencias Informáticas.
5. Implementación de una herramienta de migración de los datos de la Revista Cubana de Ciencias Informáticas.
6. Aplicación de pruebas de funcionalidad de la nueva versión de la Revista Cubana de Ciencias Informáticas, para detectar posibles no conformidades.

Los Métodos teóricos que se utilizaron durante la investigación fueron:

- **Histórico-lógico:** para realizar un estudio de la trayectoria histórica, evolución y desarrollo de la Revista Cubana de Ciencias Informáticas, además, de estrategias para la migración de bases de datos existentes en el ámbito nacional e internacional.

Introducción

- **Analítico-sintético:** para analizar las características de las diferentes estrategias y tecnologías usadas en la migración de datos, tomando los elementos más importantes que permitan lograr un correcto diseño de la nueva estrategia.

Métodos Empíricos

- **Observación:** permitió estudiar de cerca el objeto de la investigación, las acciones, causas y consecuencias logrando conocer la esencia del problema planteado, analizando desde varios puntos de vista la propuesta de solución y otras soluciones existentes e identificando qué está hecho y qué falta por hacer.

CAPÍTULO I: FUNDAMENTOS TEÓRICO-METODOLÓGICOS SOBRE LOS SISTEMAS PARA LA GESTION EDITORIAL DE LA RCCI.

En el presente capítulo se abordan temas relacionados con las publicaciones electrónicas. En el mismo se tratan distintos conceptos como son: publicación electrónica, revista científica, artículo científico, entre otros. Además, se mencionan algunos de los lenguajes de programación que están vinculados al tema, con una breve explicación de cada uno de ellos.

I.1 Conceptos asociados al problema

El presente epígrafe está centrado en un estudio de los principales conceptos asociados al problema para lograr una mejor comprensión del mismo.

I.1.1 Publicación electrónica:

Las publicaciones electrónicas, expresión de un fenómeno de continuidad y cambio, retoman elementos propios de las publicaciones impresas como presentación, estructura y organización de la información. Durante siglos los lectores se habituaron a ellas. Ahora, con la utilización de las nuevas tecnologías para el almacenamiento y tratamiento de la información, ellos disponen de un producto cualitativamente superior que cumple con sus funciones de manera más amplia y efectiva. Existen diferentes formas de publicaciones electrónicas como las revistas digitales, periódicos y boletines electrónicos, libros y colecciones de ellas, llamadas comúnmente bibliotecas electrónicas (Travieso Aguiar 2003).

Acerca de la definición de publicación electrónica puede citarse un concepto que señala que “la publicación electrónica cubre un amplio espectro de medios, formatos y métodos para la creación y distribución de trabajos. El elemento común es que la publicación existe en un fichero de computadora, el cual puede leerse y distribuirse sin necesidad de publicarse en el sentido convencional de copias en papel. Según Barker (Barker 1996), es una publicación virtual, donde las palabras y las imágenes no tienen una forma física reconocible hasta que llegan al lector, quien puede leerla y visualizarla en una variedad de formas de acuerdo con sus necesidades, incluyendo la impresión en papel” (Acosta Rodríguez 1998).

Existen en la literatura varios enfoques sobre las publicaciones electrónicas: (Kindelán 2019)

- Es la forma digitalizada de la publicación impresa.
- Es aquella que en su proceso de producción y edición incluye el uso de
- computadoras.

- Es toda publicación soportada en medios magnéticos
- Es la que se disemina a través de redes
- Es toda publicación que se caracteriza por su capacidad multimedia, por ser interactiva y por almacenarse y distribuirse en forma electrónica.

I.1.1.2 Las ventajas de las publicaciones electrónicas son: (Travieso Aguiar 2003)

- Realizar búsquedas en el texto completo.
- Acceder instantáneamente, sin necesidad de desplazamiento.
- Vincular referencias, citas y publicaciones.
- Enlazar recursos relacionados como bases de datos u otros materiales complementarios como películas y animaciones, que facilitan la expresión de ideas difíciles de plasmar en un formato impreso.
- Relacionar autores y lectores por correo electrónico, se favorece entonces la comunicación científica.
- Publicar inmediatamente, a partir de un régimen de edición continua.
- Realizar correcciones y comentarios, seguir las ideas y sugerencias hechas por los lectores.
- Oportunidad y rapidez en la difusión y distribución de la información.
- Inclusión de información tridimensional, sonido y video, así como la facilidad de su manipulación.
- Disminuir los costos y el consumo de papel, se hacen copias impresas sólo de los artículos que realmente son de interés.

Algunas de las desventajas son:

- Alta inversión inicial.
- Incomodidad de la visualización en pantalla del dispositivo electrónico.
- Es indispensable un equipo para su lectura.
- No existe una reglamentación, se carece de un código de ética establecido.
- Favorece la proliferación de la información, además de que esta puede ser modificada por los usuarios.
- Se dificulta la identificación entre la información primaria y la secundaria.

Por todas las ventajas anteriormente mencionadas no se puede decir con exactitud si las publicaciones electrónicas lograrán sustituir a las publicaciones impresas, pero es válido destacar el salto revolucionario que las mismas han dado respecto al formato impreso. En la actualidad, las publicaciones electrónicas están

caracterizadas por la rapidez en la difusión y distribución de la información, además, la inclusión de información tridimensional, de sonido y video y las facilidades para su manipulación, entre otras características. Dicho esto, sin duda alguna, las mismas tienen varios puntos a favor que las hace necesarias e indispensables en el mundo editorial.

I.1.1.3 Artículo científico

El artículo científico se define como un informe escrito y publicado que describe resultados originales de una investigación, se escribe para otros no para el autor. El propósito de la publicación es divulgar, compartir y contrastar estos resultados con la comunidad científica interesada e incorporarlos, si son validados, al enriquecimiento del saber humano. El artículo científico constituye el medio comunicativo por excelencia de la comunidad científica. Según la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO), su propósito es comunicar los resultados de investigaciones, ideas y debates de una manera clara, concisa y fidedigna (Díaz 2016).

Un artículo científico es un Informe original, escrito y publicado, que plantea y describe resultados experimentales, nuevos conocimientos o experiencias que se basan en hechos conocidos. Su finalidad es poder compartir y contrastar estos resultados con el resto de la comunidad científica, y una vez validados, se incorporen como recurso bibliográfico a disponibilidad de los interesados (González 2018).

El artículo científico se define como un informe escrito y publicado que describe resultados originales de una investigación, se escribe para otros no para el autor. El propósito de la publicación es divulgar, compartir y contrastar estos resultados con la comunidad científica interesada e incorporarlos, si son validados, al enriquecimiento del saber humano. El artículo científico constituye el medio comunicativo por excelencia de la comunidad científica (Yolanda 2019).

¿Cuáles son sus características? (González 2018)

- Se refieren a un problema científico.
- Los resultados deben ser válidos y fidedignos (no necesariamente deben ser experimentales, también pueden ser teóricos).
- Son originales: cada artículo comunica por primera vez los resultados de una investigación.
- Son presentados en revistas científicas, conferencias y otros modos de presentación.
- Suele ocupar una media de 4-5 páginas de una revista (manuscritos de 12 folios a doble espacio). Acompañados de gráficos, tablas y ocasionalmente de fotografías y dibujos.

- El número de autores o firmantes no suele ser superior a seis, considerándose al primero como autor principal del artículo.
- Es un documento formal, público, controlado y ordenado.
- Debe cumplir con criterios claves de redacción.
- Posee rigor científico y carácter lógico.
- Debe tener claridad y precisión. Es necesario el uso de un lenguaje y vocabulario científico.
- Debe ser breve y conciso.
- Tener un estilo adecuado.
- Tener compatibilidad con la ética.

De esta forma se puede asumir que un artículo científico debe tener un nivel de organización tal que, ofrezca toda la información necesaria para que el lector pueda conocer la manera en que se realizó el trabajo investigativo y pueda a su vez evaluar las observaciones realizadas. El mismo tiene como objetivo de su publicación, la defensa o la réplica de una teoría, así como el avance del conocimiento, de esta forma será publicado con éxito y obtendrá el impacto deseado.

I.1.1.3 Revistas académicas

La Editorial científica Elsevier, (una de las más reconocidas y prestigiosas en el mundo de las ciencias), según su clasificación, define como revista científica, aquella revista que difunde resultados primarios de la investigación, facilita la difusión rápida de resultados provisionales que puedan llevar a documentos más completos, comunica resúmenes e informes presentados en congresos, ya sean solicitados o no, analiza las tendencias actuales de la investigación, presenta bibliografía, artículos recibidos o encargados. Brinda información secundaria en forma de referencias de la literatura actual sobre un tema en específico, así como primaria, ambas generalmente de carácter práctica (Hidalgo 2010).

En la actualidad, la creciente producción de información exige un espacio cada vez mayor para su almacenamiento. Asimismo, se requiere de una difusión casi instantánea de la información como resultado del llamado proceso de globalización o internacionalización del conocimiento. De manera paralela pequeños editores, editores institucionales y universitarios estuvieron tratando de buscar soluciones para la digitalización, optando en algunos casos por desarrollar sitios web (continuando los flujos de trabajo al modo tradicional o en el mejor de los casos por correo electrónico, que se iba imponiendo); por desarrollar soluciones

propias para la gestión de las publicaciones, que fueron implementando paulatinamente algunas fases del proceso editorial; o por contratar soluciones comerciales, que empezaban a surgir, aunque aún de manera tímida. (Vázquez 2018)

Dentro del amplio contexto de revista académica se encuentran las revistas científicas, las cuales se enfocan en recoger el progreso de la ciencia, y constituyen el mecanismo principal de las investigaciones científicas intentando asegurar un máximo de estándares de calidad, así como validez de las mismas.

I.1.2 Revistas científicas

Una revista científica es reconocida como una publicación periódica que presenta especialmente artículos científicos, sobre investigación y desarrollo de cualquier área de la ciencia. Su componente básico es el artículo científico como contribución a la ciencia (Macias 2019).

El artículo científico (full paper) es un informe escrito donde se representan los resultados originales de una investigación y se convierte en publicación válida o publicación científica primaria, cuando se publica por primera vez y su contenido informa lo suficiente para que se puedan evaluar las observaciones, repetir los experimentos y evaluar los procesos intelectuales realizados por él o los autores (Duperet Cabrera, Pérez Martínez et al. 2015).

La publicación científica presenta una importancia extrema en la existencia del conocimiento científico y en los patrones de su difusión y uso. Su especificidad en cuanto a procesos, resultados y en los sujetos actuantes, ya sean institucionales o humanos constituye un objeto de estudio de innegable complejidad. En las revistas científicas la gestión entonces abarca al proceso más general de gestión de la información y del conocimiento, como a todos los procesos que tanto interna como externamente definen la calidad de la publicación y su posicionamiento internacional (Macias 2019).

En teoría la visibilidad de una revista científica viene dado por el trabajo que realiza el equipo científico que la compone, donde se refleja el prestigio de los mismos y se le atribuyen reconocimientos por el trabajo realizado, pero la realidad es que una verdadera valoración de una publicación científica es aquella que se realiza no solo por su factor de impacto o visibilidad sino por la contribución a la comunidad científica y a la sociedad.

I.1.2.1 Indexación e indización de revistas científicas.

La indexación es el proceso en que una revista es evaluada según criterios editoriales y tecnológicos por un índice bibliográfico para conocer su calidad científica, editorial y documental (Murillo and Saavedra 2022).

La indización es la técnica de análisis de documentos el cual tiene por objetivo representar y describir el contenido de los mismos, mediante conceptos principales contenidos en ellos (palabras clave) o vocabularios controlados (descriptorios, términos o encabezamientos de materia), con la finalidad de guiar al usuario en la recuperación de documentos a los que necesita acceder; al mismo tiempo se debe hacer diferencia la palabra indización de la indexación, que es otro termino muy frecuentemente utilizado, se entiende como el proceso por el cual el buscador va rastreando el sitio en cuestión y al mismo tiempo va incorporando a su base de datos el contenido de esas URLs, es decir, se registra ordenadamente información para elaborar un índice. Este procedimiento es de vital importancia para las páginas web ya que es el que permite que la página web aparezca en alguno de los buscadores más importantes, aquella página que no se haya indexado como debería no aparecerá entre los resultados de una búsqueda, en palabras sencillas indexar es agregar una página sin importar en que formato se encuentre a la lista de resultados de un buscador (Aguirre García 2012).

En términos de indización internacional, se pueden denotar diferentes entidades que se encargan de realizarla, entre las cuales se encuentran Redalyc, DOAJ y Google Scholar.

En cuanto a los requisitos de las revistas internacionales que las revistas deben cumplir se determinan dos condiciones mínimas: el acatamiento de una serie de estándares (conocidos como “estándares de calidad formal” o de “normalización”) y el valor de sus contenidos. La calidad del contenido es otra condición indispensable para lograr la inclusión en las bases de datos internacionales.

Para determinar si una revista ingresa a un índice se toman en cuenta ciertos indicadores de calidad, que pueden ser agrupados en:

- Calidad del contenido de la investigación.
- Características técnicas o formales.
- Uso por parte de la comunidad científica. Para medir el uso que la comunidad le ha dado a un artículo, se suele utilizar un indicador que se relaciona con la cantidad de citas que recibe el artículo por parte de otros usuarios, este se denomina factor de impacto y se usa comúnmente en los estudios métricos de la información para determinar la visibilidad de un autor, de una disciplina o de una revista.

Entre las bases de datos para indexar revistas se encuentran:

- Scielo
- ISI
- EBSCO
- Scopus
- CUIDENplus
- Psycodoc
- Clase
- DOAJ
- IRESIE
- IDEAS
- LILACS
- Mathematical Reviews
- PERIÓDICA
- SSCI

La indización es un complemento importante para las revistas, pues es un indicador de reconocimiento y de calidad, por esto, cuando se logra la indización, se podrá aumentar el reconocimiento, confiabilidad y la gestión editorial estará fortalecida; cabe aclarar que el hecho de que una revista esté indizada, no significa que no debe actualizar sus lineamientos, políticas y procesos, debido a que lograr una correcta publicación científica es un proceso de mejora continua (Hernández, Ramos et al. 2020).

I.2.2 Algunos ejemplos de revistas científicas electrónicas cubanas son las siguientes:

- **La Revista Cubana de Ciencias Forestales (CFORES)** tiene como objetivo facilitar la comunicación científica entre investigadores y profesionales vinculados al desarrollo de las Ciencias Forestales de Cuba y el extranjero, contribuir a la divulgación y promoción de los principales resultados de la ciencia cubana en el área forestal y a la diseminación de resultados relevantes de la actividad entre la comunidad científica a nivel nacional e internacional (Torres Cala, González Pérez et al. 2018).
- **Revista Cubana de Salud Pública**, esta tiene como misión publicar artículos originales, o que hayan sido debatidos en simposios y congresos nacionales o internacionales realizados en nuestro país, referentes al desarrollo de la salud pública. Divulga las experiencias y los avances teórico-prácticos de la medicina social, y la organización y dirección del sistema de salud cubano. Ofrece la oportunidad a profesionales del campo de la salud y afines de publicar, sujeto a un proceso de arbitraje por expertos, los resultados de sus investigaciones, las reflexiones de la salud pública sobre tendencias políticas, económicas, sociales, epidemiológicas u otras, y revisiones de temas de acuerdo con el perfil de la revista (Pública 2022).
- **Revista de Ingeniería Electrónica, Automática y Comunicaciones (RIELAC)**. Propiciar la divulgación de los trabajos de investigación y de proyectos, así como experiencias docentes, científicas y técnicas de la ingeniería electrónica, automática y telecomunicaciones en el país y en el mundo. De esta manera la revista contribuye a la actualización y retroalimentación de profesionales y estudiantes, así como a la discusión e intercambio científico (Revista de Ingeniería Electrónica 2022).

I.2.2.3 Revista Cubana de Ciencias Informáticas

Es una revista científica, con arbitraje por pares, con frecuencia trimestral. En sus inicios se concibió y surgió en formato impreso y tiene un Consejo Editorial que toma las decisiones respecto a la política editorial en la Universidad, en general, y en la RCCI, en particular. Cuenta, además, con un Comité de expertos asesores, que según las temáticas en las que son especialistas de reconocido prestigio nacional e internacional, revisan las contribuciones que recibe la revista y asesora al Consejo Editorial en sus funciones (Informáticas 2006).

Los artículos publicados en título completo de la revista son indizados o resumidos por:

- Scientific Electronic Library Online – SciELO
- Red de Revistas Científicas de América Latina y el Caribe, España y Portugal - Redalyc
- Directory of Open Access Journals – DOAJ
- Sistema Regional de Información en Línea para Revistas Científicas de América Latina, el Caribe,

España y Portugal - Latindex

La realiza, publica y distribuye la UCI, pero no solo es utilizada en esta universidad, sino que también es muy provechosa para todos los interesados en publicar sus resultados en el país, e incluso en el mundo, en temáticas como: Ingeniería y Gestión de Software, Bioinformática, Tecnologías de la Información y las Telecomunicaciones, Algoritmos de Programación, entre otros.

Principios fundamentales del proceso de evaluación en la RCCI:

- **Autoría:** se limita a aquellos que han contribuido de manera sustancial en la concepción, diseño, ejecución o interpretación de la investigación. Todos los que han contribuido de esta manera deben ser listados como co-autores.
- **Arbitraje doble a ciegas:** el árbitro no conoce el autor del artículo, ni el autor conoce el árbitro que evalúa su trabajo, los artículos se someten a la consideración de dos especialistas entendidos en la materia para buscar un consenso en cuanto a la aceptación o rechazo del material presentado. Cuando ambos especialistas difieren en sus criterios, se busca el juicio de un tercero, para elaborar un veredicto final. Los árbitros serán independientes entre sí.
- **Confidencialidad:** obliga a todos los participantes del proceso de arbitraje a guardar estricta reserva de todo cuanto llegue a su conocimiento en el curso del mismo. Cualquier documento sometido a la Editorial para su posible publicación deberá permanecer en confidencialidad por parte de todos aquellos involucrados en el proceso de dictamen.
- **Transparencia:** obliga a los integrantes del consejo editorial a guardar transparencia sobre el proceso de evaluación.
- **Celeridad:** obliga al cumplimiento de los plazos establecidos en los procedimientos de aceptación, arbitraje y publicación de las contribuciones enviadas. La celeridad no es incompatible con el tiempo necesario para alcanzar una decisión justa.
- **Reconocimiento:** obliga a reseñar las fuentes contribuyentes al resultado, personales e institucionales, públicas o privadas, proyectos, inversores u otras fuentes de financiamiento, o de otra índole siempre que hayan sido influyentes en los resultados presentados.

I..3 Sistemas para la gestión de revistas académicas.

En este epígrafe se hace un análisis de las herramientas y tecnologías que serán usadas de formas complementarias para la realización de la nueva versión de la revista. Además, se mencionan sus principales características y los rasgos más significativos, así como los elementos técnicos que permitieron su elección para desarrollar la propuesta de solución.

Content Management System (CMS, por sus siglas en inglés, sistemas de gestión de contenidos) es una

aplicación que soporta creación y modificación de contenidos digitales. Por lo general, admite múltiples usuarios que pueden administrar el contenido en un entorno colaborativo ambiente. Las características de CMS varían ampliamente. Las funcionalidades de CMS incluyen formato gestión, edición de historial, control de versiones, indexación, búsqueda y recuperación. Por su naturaleza, los sistemas de gestión de contenidos soportan la separación de contenido y presentación. Un CMS típico consta de dos componentes principales (Adithela, Christie et al. 2018) :

Un CMS proporciona diferentes ventajas como se muestra a continuación (Adithela, Christie et al. 2018) :

- Capacidad para crear un aspecto y una sensación unificados
- Componentes reutilizables
- Control de versiones
- Fácil despliegue
- Reducción de la necesidad de codificar desde cero
- Gestión segura de los permisos
- Soporte para varias bases de datos
- Gestión de medios

Como respuesta a una necesidad real de contar con plataformas de gestión integrales que contemplasen todo el proceso editorial, surgen los llamados Electronic Scientific Journalsmanagement Systems (ESJMS, por sus siglas en inglés, Sistemas de gestión de revistas científicas electrónicas), los cuales tienen las siguientes características capaces (Vázquez 2018) :

- Gestionar un sitio web de información sobre una o varias publicaciones
- Permitir el registro de usuarios con distintos roles y su gestión, incluyendo sistemas de comunicación
- Enviar y recibir originales para su publicación
- Gestionar el proceso de revisión de originales mediante los sistemas habituales en comunicación científica (ciega, doble ciega, abierta...)
- Llevar a cabo el proceso de corrección de galeras
- Publicar finalmente los artículos aceptados, en distintos formatos para documentos electrónicos y su consulta a través de la web
- Permitir la interoperabilidad con otros sistemas y estándares
- Gestionar los derechos de acceso y distribución de los contenidos, incluyendo los pagos por acceso o suscripción.

I.3.1 Open Journal System (OJS)

El sistema abierto de gestión y publicación de revistas OJS de sus siglas en inglés (Open Journal System) es una solución de código abierto para la gestión y publicación de revistas académicas en línea. Su objetivo es mejorar la calidad académica de la publicación de revistas a través de una serie de innovaciones, entre las que se incluyen la mejora de la experiencia del lector, la transparencia de las políticas de las revistas y la mejora de la indexación. OJS apoya el principio de ampliar el acceso, este sistema no sólo pretende ayudar a la publicación de revistas, la experiencia adquirida por otras revistas ha demostrado que el uso de un formato electrónico a través de un OJS permite incrementar el acceso y visibilidad de las revistas latinoamericanas inaccesibles o desconocidas en la comunidad científica (González-Marín and Cuello 2019).

OJS permite realizar el proceso documental de una revista de manera completa: adquisición, elaboración, difusión de la información en formato digital; y por supuesto que ello posibilita la comunicación entre todos sus usuarios. El sistema de la plataforma como tal, reduce y agiliza el proceso de edición de la publicación científica y genera políticas que aseguran su calidad.

En el caso de la versión 3.3.0.10 posee una serie de mejoras respecto a versiones anteriores, que se podrían agrupar en las siguientes áreas:

Flujo de trabajo: en esta nueva versión se compone de cuatro fases (propuesta, revisión, editorial y producción) y es posible saltar de uno a otro sin necesidad de completar todos los pasos. Del mismo modo, también hay flexibilidad respecto de la participación de personas y roles en las distintas fases.

Roles: también pasan a ser más flexibles, pudiendo renombrar los títulos de estos, eliminar aquellos que no se van a usar, añadir otros nuevos que se adapten mejor a las necesidades de la revista u otorgar permisos adicionales a los ya existentes.

Acceso flexible a funciones y tareas: sin necesidad de tener que cambiar de rol dentro de la interfaz de administración para atender tareas de diferentes roles. Así, en función de los roles y permisos, todas las opciones de trabajo aparecen disponibles y se puede cambiar de unas a otras de manera transparente.

Registro simplificado: de modo que cualquier usuario puede participar en la revista, como lector, autor, revisor, etc. haciendo un registro simple, con unos pocos datos, que habrá de completarse solo si así se desea o si es solicitado por los editores.

Discusiones editoriales: dentro de cada una de las fases del workflow se incrustan hilos de discusión en los que se mantiene comunicación con los implicados en cada una de ella y van quedando archivadas todas estas informaciones para su consulta.

Interfaz personalizable y temas: ahora es posible separar el modo de visualización de la interfaz de lectura y la de administración de modo que, mientras la primera es perfectamente personalizable a través de plantillas y temas, la segunda permanece invariable, de manera que cualquier usuario de OJS 3 sabrá cómo manejarse

independientemente de la revista en la que se encuentre. Además, gracias al uso de estos temas es posible que la interfaz se adapte al uso de distintos dispositivos y tamaños de pantalla.

Editor de contenido enriquecido y edición multilingüe: OJS incluye un editor de contenido visual que permite una edición de textos muy completa y enriquecida, permitiendo además la inserción de imágenes, enlaces, etc. Como complemento además incorpora la opción de poder trabajar directamente con HTML, de manera que se puede insertar código para, por ejemplo, incrustar contenido de otros sitios web, o bien transformar el aspecto con el que se muestra el contenido incluido en cada apartado (Vázquez 2018).

OJS es una herramienta completa para la gestión de todo el flujo de trabajo editorial y de envío y la publicación de sus artículos y números en línea. Ofrece las siguientes características: (Project 2022)

- Front-end de lector responsivo con una selección de temas o diseños gratuitos
- Flujo de trabajo editorial flexible y configurable
- Envío y gestión en línea de todos los contenidos
- Módulo de suscripción con opciones de acceso abierto diferido
- Integración con servicios de publicación académica como Crossref, ORCID y DOAJ
- Recomendado por Google Scholar por su facilidad de indexación y descubrimiento
- Instalado y controlado localmente
- Dirigido y apoyado por la comunidad
- Multilingüe y traducido a más de 30 idiomas
- Amplias guías de usuario y vídeos de formación

OJS utiliza la arquitectura cliente- servidor q consiste en dos partes: un servidor y múltiples clientes. El componente del servidor proporcionará servicios a múltiples componentes del cliente. Los clientes solicitan servicios del servidor y el servidor proporciona servicios relevantes a esos clientes. Además, el servidor sigue escuchando las solicitudes de los clientes. (Huaman 2018)

- **Cliente:** Envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito una vez que son servidas sus solicitudes, termina el trabajo.
- **Servidor:** Es un programa que ofrece un servicio que se puede obtener en una red. Acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante. Al ser posible implantarlo como aplicaciones de programas, puede ejecutarse en cualquier sistema donde exista TCP/IP y junto con otros programas de aplicación. El servidor comienza su ejecución antes de comenzar la interacción con el cliente.

OJS 3.3.0.10 tiene muchas características nuevas y mejoradas con respecto a OJS 2, incluyendo un flujo de trabajo aún más dinámico y personalizable. También tiene un diseño más racionalizado con mayor flexibilidad

para los roles, las tareas y la gestión de archivos, este ha mejorado notablemente la experiencia del usuario con una claridad aún mayor, a la vez que sigue proporcionando una base sólida de funcionalidad básica que permite a los editores satisfacer sus diversos requisitos y prácticas.

I.4 Herramientas y metodologías

XAMP 3.2.4 (eXtensible Analytical Model Platform), un software que integra varios modelos analíticos en una única plataforma extensible. Un modelo analítico puede integrarse en XAMP como un binario -no se necesita el código fuente, además de un archivo de configuración que especifica qué entradas y salidas toma o produce el modelo. Dado que los modelos en sí no necesitan ser modificados o recompilados para ser integrados en XAMP permite añadir nuevos modelos o actualizar los antiguos a lo largo del tiempo, lo que permite el desarrollo de modelos en colaboración. Las entradas y salidas de un modelo en XAMP reciben etiquetas globales únicas. Con las etiquetas globales, XAMP puede inferir si las entradas (o salidas) de un modelo son las mismas o son iguales o distintas a las entradas (salidas) de otro modelo. XAMP proporciona una interfaz uniforme y un motor de consulta de modelos. Los usuarios pueden realizar búsquedas basadas en las entradas para para saber qué resultados pueden predecir los modelos a partir de un conjunto de entradas disponibles. Los usuarios también pueden realizar búsquedas basadas en los resultados para encontrar las entradas necesarias para que los modelos produzcan los resultados deseados (Wang and Solihin 2013).

Un sistema de gestión de bases de datos (SGBD) es una plataforma de software de sistema que almacena y recupera una colección de datos relacionados, se utiliza generalmente en áreas de gestión de datos específicos de la aplicación, donde el rendimiento, la disponibilidad y la escalabilidad tienen la máxima prioridad (Venegas Bravo 2022). En la actualidad los SGBD, permiten resolver problemas, brindando a los administradores de la información comodidad y eficiencia en el tratamiento de los datos.

MySQL 10.4.11 es uno de los sistemas de gestión de bases de datos más populares para aplicaciones basadas en la web. Es gratuito y además está constantemente actualizado con nuevas características y medidas de seguridad. Este sistema de bases de datos permite seleccionar y manejar datos de una gran cantidad diferente de tipos de tablas, con una seguridad excelente y una gran fama de no desperdiciar los recursos del servidor (Lozano Banqueri 2018).

Sus principales ventajas son: (Lozano Banqueri 2018)

- Ofrece una gran cantidad de características incluso en su versión gratuita.
- Existen multitud de interfaces de usuario y una gran comunidad detrás.
- Es compatible con otras bases de datos como Oracle.
- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes

para poder optimizar el SGBD a cada caso concreto.

- Es altamente confiable en cuanto a estabilidad.

El servidor Apache 2.4.41 está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. Este es el más común y más utilizado en todo el mundo. Además, es gratuito, y de código abierto, así que podríamos decir que corre sobre cualquier plataforma. El servidor Apache es una muestra de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar (Mendez Avendaño and Zapata Tabarez 2016).

PHPMYAdmin 5.0.1 es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas webs, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas (Carbonell 2015).

JavaScript es un lenguaje de programación de computadoras dinámico utilizado habitualmente en navegadores web para controlar el comportamiento de páginas web e interactuar con los usuarios. Permite comunicación asincrónica y puede actualizar partes de una página web o incluso reemplazar completamente su contenido. Verás que JavaScript es utilizado para mostrar información de fecha y hora, ejecutar animaciones en un sitio web, validar formularios, sugerir resultados mientras el usuario escribe en un cuadro de búsqueda y más (Dimes 2015).

El lenguaje de programación **PHP 7.4.2** (Preprocessed Hypertext Pages por sus siglas en ingles), su principal objetivo está encaminado a la construcción de páginas Web del lado del servidor, capaces de generar contenido dinámico en la World Wide Web (www). Ha estado desarrollándose con el tiempo, contiene varias bibliotecas para funciones matemáticas y de bases de datos. Construir una página dinámica basada en bases de datos es simple con PHP, este da soporte a varias bases de datos: Oracle, Sybase, PostgreSQL, SQLite, MySQL entre otros. En la actualidad se encuentra en consolidación, se está centrando en madurar aspectos relacionados con la integración de sus partes y ha dejado un tanto atrás la fase expansiva (Arias 2013).

PhpStorm 11.0.4 es un IDE para PHP que realmente "entiende" lo que se programa. Es compatible con PHP 5.3/8.1, ofrece prevención de errores sobre la marcha, una mejor finalización y refactorización de código, depuración sin configuración y un editor extendido para HTML, CSS y JavaScript. El IDE ofrece finalización

inteligente de código, resaltado de sintaxis, configuración extendida de formateo de código, detección de errores sobre la marcha, plegado de código y admite mezclas de lenguajes.

Symfony 5.4.17 es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo (Potencier and Zaninotto 2012).

Doctrine es el ORM (siglas en inglés de Object-Relational Mapping, es decir, el mapeo relacional de objetos), se puede utilizar a través de Composer en cualquier proyecto, ya que es totalmente independiente. En el caso de Symfony es el ORM que se utiliza internamente. Esto significa que va a trasladar los datos de una base de datos relacional a un sistema de clases y de objetos, donde las clases serían las tablas y los registros pasarían a ser lo equivalente a objetos. Este concepto es muy potente, porque ofrece la posibilidad de, internamente con nuestro framework y con nuestra aplicación, trabajar con clases y objetos, que es lo habitual y lo mejor para poder hacer ese código escalable y mantenible (Blanco 2018).

Lenguaje de soporte a la metodología seleccionada

El lenguaje de modelado unificado **UML 2.0** es un lenguaje de modelado de desarrollo de propósito general en el campo de la ingeniería de software, diseñado para proporcionar un enfoque estándar para visualizar el diseño del sistema. La motivación original para crear UML fue el deseo de estandarizar diferentes sistemas de notación y métodos de diseño de software. En UML, un diagrama de clases es uno de los seis tipos de diagramas de estructura. Los diagramas de clase son la base del proceso de modelado de objetos y modelan la estructura estática del sistema (Prihandoyo 2018).

Visual Paradigm para UML 8.0: herramienta CASE

Las herramientas CASE (Computer Aided Software Engineering por sus siglas en inglés), se desarrollan para automatizar procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software. Se enfoca en un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo.

Se ha seleccionado Visual Paradigm como herramienta CASE, dado a que esta soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas, despliegue y contribuye a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Se caracteriza por el uso de un lenguaje estándar común al equipo de trabajo, que facilita la comunicación entre sus integrantes, es una herramienta fácil de instalar y actualizar, genera código para varios lenguajes de programación y exporta en varios formatos, es una tecnología libre y está disponible en varios idiomas (Pressman 2002).

I.4.1 Proceso Unificado Abierto

OpenUP es un proceso unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado; aplica una filosofía pragmática ágil que se centra en la naturaleza colaborativa del desarrollo de software. El esfuerzo personal en proyectos de OpenUP es organizado en microincrementos, los cuales representan cortas unidades de trabajo que aportan al progreso de un proyecto. OpenUP divide el proyecto en iteraciones planeadas, generalmente por semanas, enfocando cada iteración en una entrega con más valor para los interesados de una manera predecible (Ríos, Hinojosa et al. 2013).

Los principios básicos en los que se fundamenta OpenUP son:

- Equilibrar prioridades competitivas para maximizar el valor para los interesados: promover prácticas que permitan a los participantes del proyecto e interesados desarrollar una solución que maximiza los beneficios para los interesados y cumple con las restricciones impuestas en el proyecto.
- Colaborar para alinear intereses y compartir entendimientos: promover prácticas que fomenten un ambiente de equipo saludable, permite la colaboración y desarrollar un entendimiento compartido del proyecto.
- Enfocarse en la arquitectura prematura para minimizar riesgo y organizar el desarrollo: promover prácticas que permita al equipo enfocarse en la arquitectura para minimizar riesgo y organizar el desarrollo.
- Evolucionar para continuamente obtener retroalimentación y mejoramiento: promover prácticas que permitan al equipo obtener.(Caballero and Rojas 2018)

Todo proyecto en OpenUP consta de cuatro fases: inicio, elaboración, construcción y transición y cada una de estas fases se divide a su vez en iteraciones.

OpenUP se organiza en dos dimensiones: Contenido metodológico y contenido procedimental. El contenido metodológico es el que define elementos metodológicos tales como disciplinas, tareas, artefactos y procesos, independientemente de cómo se usen estos o se combinen. El contenido procedimental, por el contrario, es donde se aplican todos estos elementos metodológicos dentro de una dimensión temporal, pudiéndose crear multitud de ciclos de vida diferentes a partir del mismo subconjunto de elementos metodológicos.

El objetivo de OpenUP es ayudar al equipo de desarrollo, a lo largo de todo el ciclo de vida de las iteraciones, para que sea capaz de añadir valor de negocio a los clientes, de una forma predecible, con la entrega de un software operativo y funcional al final de cada iteración. El ciclo de vida del proyecto provee a los clientes de: una visión del proyecto, transparencia y los medios para que controlen la financiación, el riesgo, el ámbito, el valor de retorno esperado, etc. (Ríos, Hinojosa et al. 2013).

Conclusiones parciales

- En este capítulo se logró caracterizar todo lo referido al mundo editorial, donde se plasmaron conceptos tales como: publicación electrónica, revista científica, artículo científico, entre otros, lo cual posibilitó una comprensión profundizada del campo de acción en cuestión. Se realizó un estudio de los lenguajes y los CMS más utilizados a nivel mundial, lo que posibilitó seleccionar las herramientas y tecnologías que contribuirán con el desarrollo de la solución propuesta, las cuales facilitan y fortalecen el trabajo, aportando rapidez y eficacia a lo largo del ciclo de vida del desarrollo y garantiza una mayor usabilidad de la aplicación. Las tecnologías, herramientas y lenguajes a utilizar en la construcción de la nueva versión del sitio, por ser de software libres, se corresponden con las políticas de soberanía tecnológica que impulsa la Universidad de Ciencias Informáticas y nuestro país.

CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA REVISTA CUBANA DE CIENCIAS INFORMATICAS 2.0.

En este capítulo se profundiza en la situación de la RCCI y se analiza el proceso objeto a automatizar, para ello se realiza una descripción del proceso de desarrollo del sitio web a través del análisis y diseño del software. Se realiza un modelado del dominio del sistema dando lugar al levantamiento de requisitos funcionales y no funcionales que debe tener el sistema. Además, se realizará un diagrama de casos de uso del sistema, así como la propuesta de solución del mismo.

II.1 Propuesta de solución

Se plantea como posible solución, una nueva versión de la RCCI que cuenta con una interfaz visual moderna, centrada en la lectura, actualizada y amigable en una gama de colores de azul y blanco y con una nueva plantilla OJS3. Además, de que el sitio se desarrollará para adaptarse a los equipos móviles, como Smartphone y tablets, lo que se conoce como diseño web responsivo. En cuanto a la documentación de la revista se realizará un estudio que involucra dos instalaciones de OJS, donde se analizan las semejanzas y diferencias entre las bases de datos de la versión 2.4.6.0 (la vieja) y la versión 3.3.0.10 (la actual), lo cual posibilitará analizar de manera profunda cuanta diferencia y similitud existe entre las mismas, en cuanto a tablas, tuplas, tipo de datos y correspondencia, de tal forma que se pueda predecir una posible migración de datos. Durante este análisis se realizarán un conjunto de scripts a la base de datos actual en cada etapa de todo un flujo editorial para de esta manera facilitar el análisis de las diferencias antes mencionadas y proponer una herramienta de migración.

La herramienta propuesta sería una solución implementada en Symfony por la capacidad de este framework para manejar más de una base de datos y mapear con facilidad las entidades de las mismas utilizando el ORM (por sus siglas en inglés, Object Relational Mapper, mapeo objeto-relacional) Doctrine para la implementación de la migración.

Modelo de dominio

El modelo de dominio es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. Este modelo ayuda a comprender los conceptos que utilizan los usuarios, es decir, los conceptos con los que trabajan y con los que deberá trabajar la aplicación. Su objetivo es contribuir a la comprensión del contexto y de los aspectos básicos del dominio de la aplicación. Se describe mediante diagramas de UML, especialmente mediante diagramas de clases.

- Su utilidad radica en ser una forma de “inspiración” para el diseño de los objetos software.
- Es entrada para muchos de los artefactos que se construyen en un proceso software
- Un modelo de dominio muestra las clases conceptuales significativas en un dominio del problema.
- Se centra en las abstracciones relevantes, vocabulario del dominio e información del dominio.
- Es el artefacto clave del análisis orientado a objetos.

- En UML se utilizan los diagramas de clases para representar los modelos de dominio (Vázquez Ingelmo, García-Holgado et al. 2019).

Como primer paso se identificarán todos los conceptos que se utilizarán en el diagrama, mediante un glosario de términos sobre los nombres:

- **Portal:** Sitio Web que contiene las publicaciones de la Revista Cubana de Ciencias Informáticas.
- **Artículo:** Cualquier información que se publica asociada a la Revista.
- **Editor:** El Editor supervisa todo el proceso de revisión, edición y publicación dentro de la Revista
- **Enlaces:** Ofrece información sobre entidades u organizaciones relacionadas con la publicación o el patrocinio de la revista.
- **Usuario:** Cualquier persona que visite el sitio en busca de información asociada con la revista.
- **Eventos:** Promociona eventos relacionados con la temática de la revista del país.
- **Autor:** Persona que tenga publicados o publique artículos en la revista.
- **Revista:** Revista Cubana de Ciencias Informáticas que presenta una serie de artículos publicados relacionados con el tema investigativo y científico.
- **Servicios:** Servicios que ofrece la revista a toda persona que visita el portal.

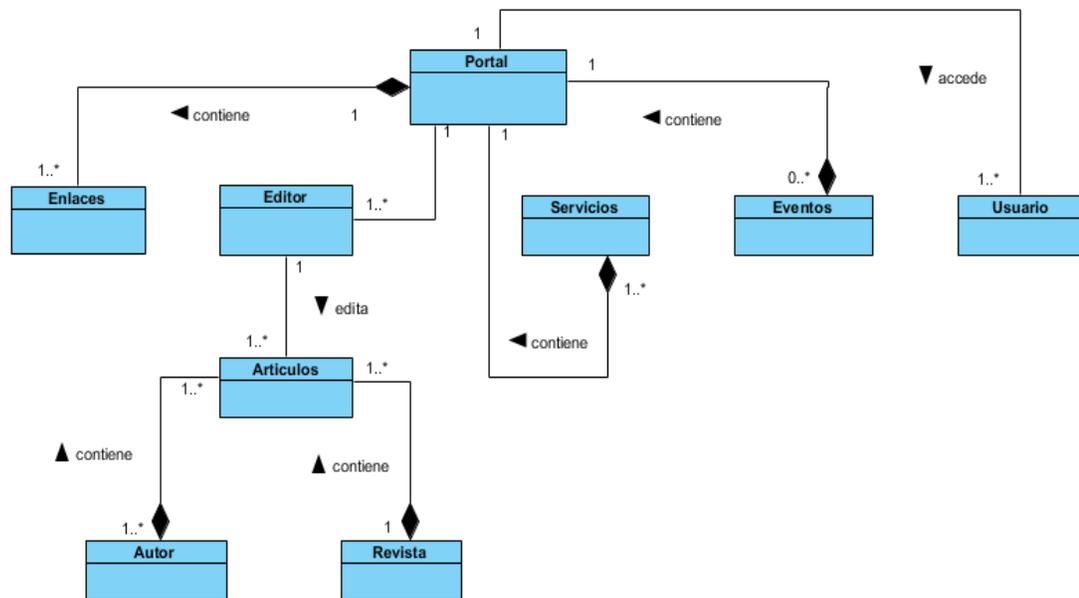


Figura 1: "Diagrama de modelado del dominio"

Los requisitos funcionales definen las funciones que el sistema será capaz de realizar. A continuación, se muestran los requerimientos funcionales que debe cumplir la aplicación a desarrollar:

1. Crear número de la revista.
2. Buscar número de la revista.
3. Editar número de la revista.
4. Registrar usuario.
5. Autenticar usuario.
6. Mostrar enlaces a otros sitios de interés.
7. Buscar publicación.
8. Editar publicación.
9. Modificar publicación.
10. Eliminar publicación.
11. Buscar rol de usuario.
12. Crear rol de usuario.
13. Modificar rol de usuario.
14. Eliminar rol de usuario.
15. Adjuntar archivo de la publicación.
16. Asignar rol a usuario.
17. Buscar contenido por palabras claves.
18. Actualizar estado de la publicación.
19. Enviar notificación de estado de la publicación.

Requerimientos no funcionales del Sistema:

Por Requisitos No Funcionales (en lo adelante RNF) se entiende el conjunto de propiedades o cualidades que el producto debe tener, características que lo hacen más agradable, rápido o confiable.

Usabilidad:

RNF1- Aunque el sistema está orientado a usuarios con un dominio en el manejo de computadoras el mismo se caracterizará por un diseño sencillo e intuitivo.

Soporte:

RNF2- Se requiere MySQL 10.4.24 como servidor de bases de datos. Por las características que este presenta, las que fueron expuestas en el capítulo anterior.

RNF3- Versión de PHP 8.1.6 o superior. Esta es la última versión del lenguaje de programación Web a utilizar en el desarrollo de esta investigación

Seguridad:

RNF4- Al sistema solo puedan acceder el personal autorizado a manejar la información que en el mismo se manipula otorgando los niveles de acceso correspondientes a cada uno de los roles.

Visibilidad:

RNF5- Necesidad de que el sistema se pueda visualizar en cualquier navegador y desde cualquier sistema operativo.

Posicionamiento Web:

RNF6- Permitir navegar a la página inicio desde cualquier página.

RNF7- Necesidad de que la aplicación aparezca en los primeros cinco resultados de búsqueda en cualquier buscador de Internet.

Interfaz

RNF8- El sistema contendrá un diseño gráfico que no debe ser complejo, con colores suaves en la gama de colores azul, blanco y rojo que no afecten la visión de los usuarios y totalmente libre de anuncios que puedan distraer a las personas que usan el sistema

II.2 Definición de los casos de uso del sistema

El diagrama de caso de uso es un tipo de diagrama UML de comportamiento y se usa frecuentemente para analizar varios sistemas. Permiten visualizar los diferentes tipos de roles en un sistema y cómo esos roles interactúan con el mismo.

Para modelar los casos de uso del sistema de la RCCI se logró agrupar los requisitos funcionales del sistema en los casos de uso que se mostrarán a continuación:

Tabla 1“ Definición de los actores del sistema”

Actores	Descripción
Autor	Es el encargado de la gestión de las publicaciones.
Usuario	Es un usuario autenticado que puede navegar por el sitio y desempeñar uno o varios roles en el sistema
Administrador	Encargado de la configuración de la revista como los envíos en el flujo de trabajo editorial además, de controlar los permisos y accesos a los usuarios.
Editor	El Editor supervisa todo el proceso de revisión, edición y publicación. Crea los números de la revista, programa los envíos para su publicación, organiza el Índice y publica el número como parte del Proceso de Publicación

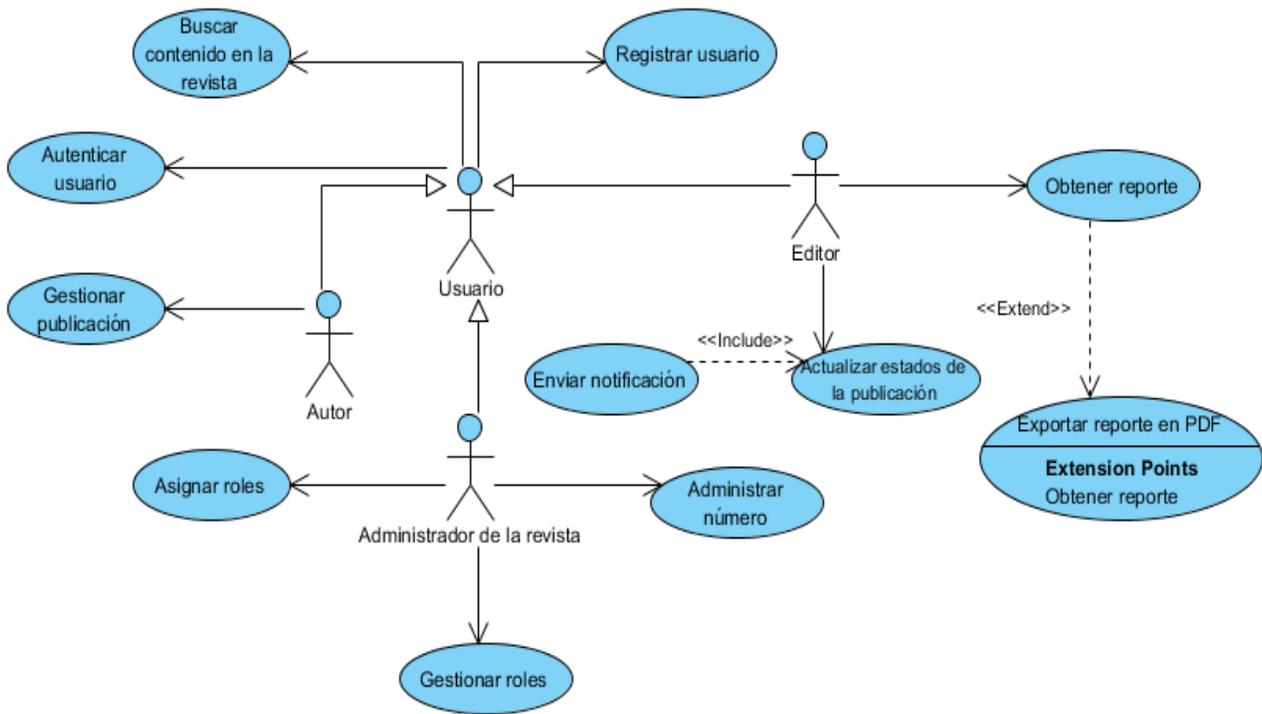
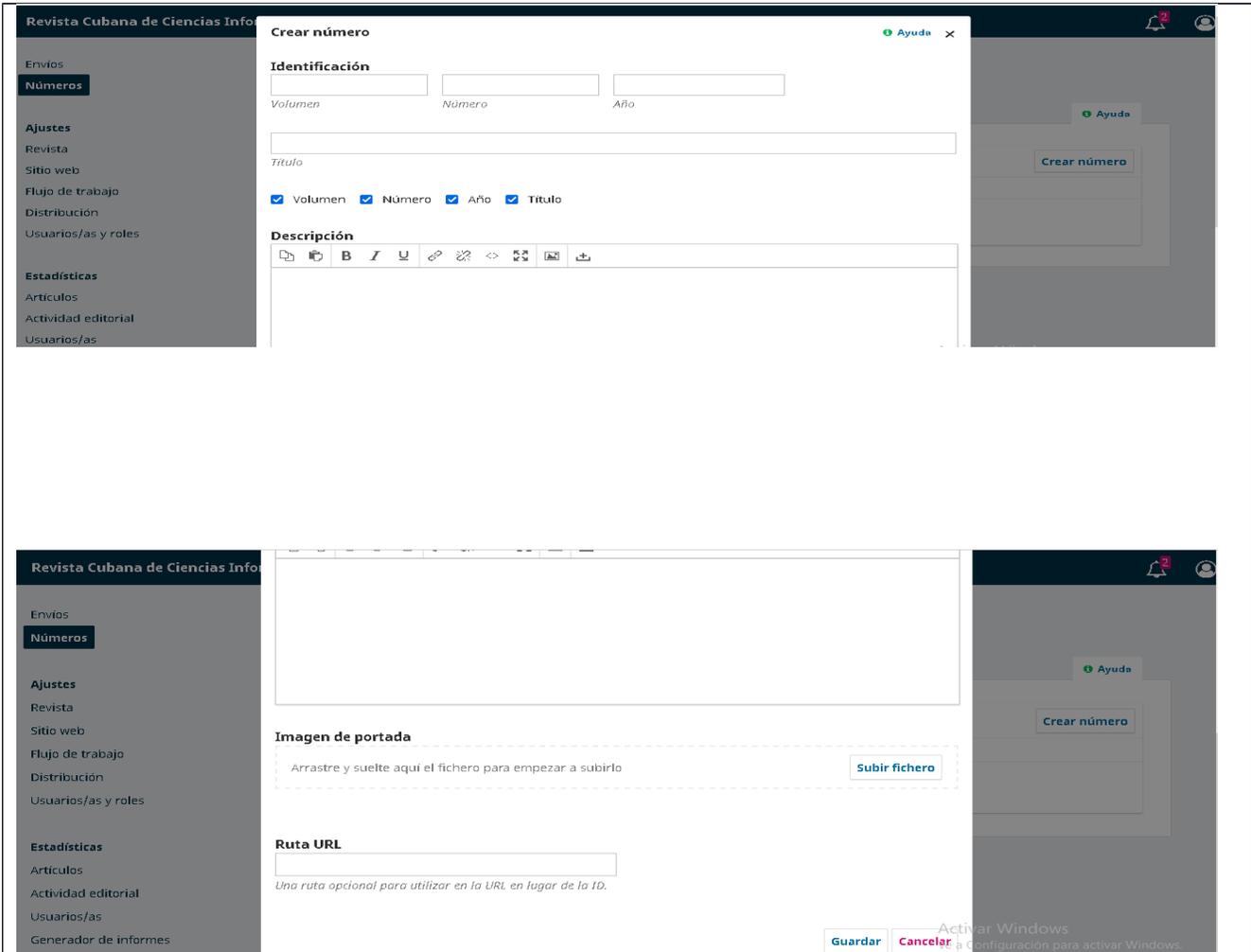


Figura 2 “ Diagrama de Casos de Uso del Sistema”

Tabla 2 “ Descripción textual del Caso de Uso del Sistema”

Caso de uso 1	Gestionar Número
Propósito	Brindar toda la información sobre los números de la revista.
Actores	Editor
Resumen	El caso de uso se inicia cuando el actor desea eliminar, modificar, crear o buscar un número. En cualquiera de los cuatro casos se muestra una página con un formulario correspondiente a la opción seleccionada para realizar la operación.
Referencias	R7, R8, R9, R10, R15, R18
Precondiciones	Usuario del sistema autenticado
Curso normal de los eventos	
Sección crear número	
Acción del actor	Respuesta del sistema
1. El actor selecciona la opción crear número	2. El sistema muestra un formulario para que el actor introduzca los datos del número.
3. El actor entra los datos necesarios para crear un número.	4. El sistema crea un nuevo número.
Prototipo de interfaz	



Sección modificar publicación	
Acción del actor	Acción del actor
1. El actor selecciona la opción editar.	2. El sistema muestra un listado con todos los números.
3. El actor escoge el número del cual desea modificar sus datos.	4. El sistema muestra el formulario del número escogido con los datos que se permite modificar
5. El actor modifica los datos y los envía.	6. El sistema actualiza los datos del número.
Prototipo de interfaz	



Sección buscar número

Acción del actor	Respuesta del sistema
1.El actor selecciona la opción ver o previzualizar número.	2. El sistema muestra el número seleccionado.
3. El actor selecciona el número deseado	

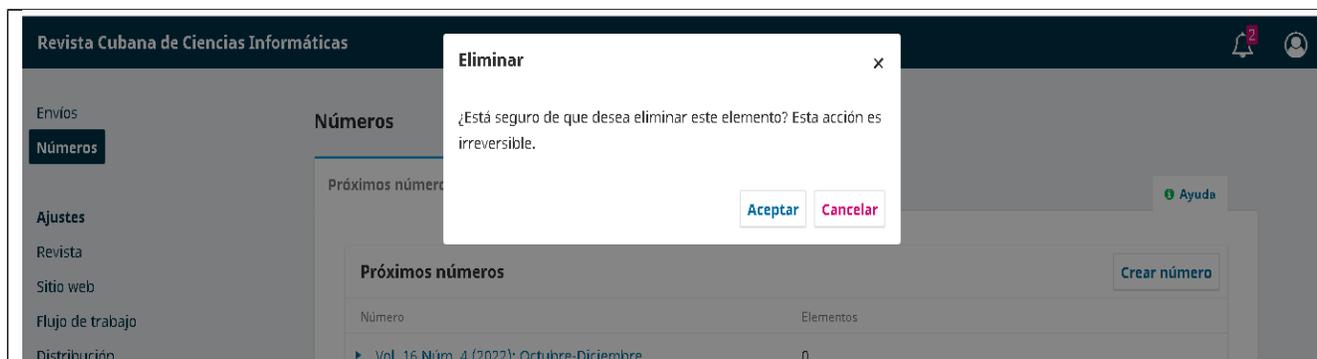
Prototipo de interfaz



Sección eliminar número

Acción del actor	Respuesta del sistema
1. El actor selecciona la opción Eliminar número.	2. El sistema muestra el listado de los números existentes.
3. El actor selecciona el número que desea eliminar.	4. El sistema muestra una alerta para verificar que la eliminación es posible según el estado del número.
5. El actor confirma la acción.	6. El sistema elimina el(los) usuario(s).

Prototipo de interfaz



Flujo alternativo	
Viene del paso 5 del curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El actor confirma la acción.	2. El sistema no realiza ninguna acción. Se retorna al paso 2 del curso normal de los eventos.
Flujo alternativo	
Acción del actor	Respuesta del sistema
1. El actor no confirma la acción	2. El sistema no realiza ninguna acción. Se retorna al paso 2 del curso normal de los eventos.

II.3 Patrones de Casos de Uso utilizados

Los patrones de Casos de Uso aplicados en la construcción del diagrama de Casos de Uso del Sistema se describen brevemente a continuación:

CRUD Completo

Este patrón consiste en un caso de uso para administrar información, nos permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, cambiar y eliminar o dar de baja. El mismo se manifiesta en los Casos de Uso: “Gestionar Publicación” y “Gestionar Usuarios”.

CRUD Parcial

Indica que en caso de que solo algunas de las cuatro operaciones sean simples mientras que otras son complejas, se puede agrupar las operaciones simples en un caso de uso y dejar las otras modeladas como un caso de uso separado. Este se manifiesta en “Administrar número”

Generalización/Especialización entre actores

Una relación de generalización de una clase hija de actor a otra clase padre de actor indica que el hijo hereda el rol que la clase padre puede jugar respecto a un Caso de Uso. Varios actores pueden jugar el mismo rol en un Caso de Uso particular, como se pone de manifiesto en el diagrama de Casos de Uso del Sistema con la generalización/especialización establecida entre los actores del sistema (González, Condori et al. 2021). Este patrón se evidencia en “Administrador de la revista”, “Editor” y “Autor” al asumir el rol del actor “Usuario”.

Extensión Concreta o Inclusión

Este patrón se divide en concreta extensión o concreta inclusión. Poniéndose en evidencia el empleo de la

concreta inclusión al establecerse una relación de inclusión por particionamiento entre el caso de uso del sistema “Actualizar estados de la publicación” y “Enviar notificación”. Una relación de inclusión es una relación desde un Caso de Uso base a un Caso de Uso de inclusión, que especifica cómo el comportamiento definido para el Caso de Uso de inclusión se inserta explícitamente dentro del comportamiento definido para el Caso de Uso base. Se utiliza para dividir partes de un flujo de trabajo de cuyos resultados, y no del método para obtenerlo, depende el Caso de Uso base. Se puede hacer esta partición si simplifica la comprensión del Caso de Uso base o si el comportamiento separado puede reutilizarse en otros Casos de Uso (Castro Aguilera 2017).

II.4 Arquitectura de la solución

Los patrones arquitectónicos, o patrones de arquitectura, dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones. Expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. No son soluciones completas en sí mismas, sino que ofrecen conocimientos parciales que, cuando se combinan con otros patrones, llevan al diseñador a una solución para un dominio de problema dado.

Modelo-Vista-Controlador (Alonso-Aranda 2019)

Es una arquitectura con una solución sencilla y potente para poner en práctica la separación entre el frontend y el backend, la cual en poco tiempo se convirtió en la arquitectura estándar para el resto de lenguajes de programación orientados a objetos, sobre todo en aquellos desarrollos donde primaba el uso de interfaces de usuario, ya que su funcionamiento se fundamentaba en la separación del código en tres capas distintas, autónomas, y cada una con una responsabilidad definida, a las que llamaron Modelos, Vistas y Controladores.

Modelo

Es la parte del sistema que maneja directamente los datos, es decir, el que realiza las operaciones para obtener los resultados, por tanto, contendrá aquellos submódulos/clases necesarios para acceder, mostrar o refrescar dicha información. En el diseño más común de una aplicación, los datos estarán alojados en una base de datos externa, de modo que en el modelo se definirán las conexiones necesarias del SGBD usualmente con una capa de acceso a datos, si se utiliza el concepto de capas. En cualquier caso, estos datos ‘externos’ no pertenecen al modelo.

Vista

Las vistas, contienen aquellos módulos que se van a encargar de materializar las interfaces de usuario de la aplicación, de modo que siempre mostrarán la información más actualizada, así el programador puede despreocuparse de ellas, ya que es el módulo quien se encarga automáticamente de dicha actualización.

De igual manera que el modelo hacía referencia al backend de nuestra aplicación, las vistas hacen referencia el

frontend; es decir, se encargan de definir la interfaz de usuario, en el caso de una aplicación de escritorio, o la apariencia de una página web, en el caso que aplique.

Controlador

Es el enlace imprescindible entre las vistas y los modelos, de modo que sirve para dar respuesta a la comunicación bidireccional entre ambos elementos. Sin embargo, ni se encarga de manipular directamente los datos, responsabilidad del Modelo; ni de definir la interfaz de usuario en el que estos van a ser mostrados, tarea de la vista. Únicamente hace de puente, establecimiento los mecanismos y reglas para responder a la dinámica de la comunicación.

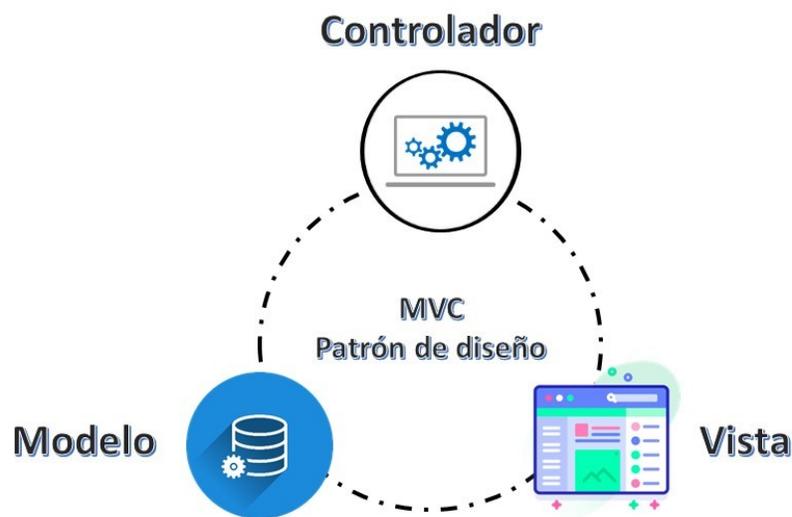


Figura 3 "Patrón Arquitectónico Modelo vista Controlador"

Modelado del diseño

El modelo de diseño es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es utilizado como entrada esencial en las actividades relacionadas con la implementación. El modelo de diseño puede contener: diagramas, clases, paquetes, subsistemas, interfaces, relaciones y los atributos (Pressman 2002).

Diagrama de clases del diseño

En la siguiente figura se muestra el diagrama de clases del diseño correspondiente al caso de uso Gestionar Número, el cual se distribuye en las siguientes clases: las clases "EntityManager" y "AbstractFacade" son proporcionadas por el framework de desarrollo; "NumeroController" que se localiza en el paquete "Controller".

Donde la client page Index hace una llamada de tipo link al server page NumeroController, el cual crea una clase client page ModificarNumero que contiene un formulario que permite realizar un submit a dicha server page. Este contiene los métodos mostrarNumero(), crearNumero(), updateNumero() y destroyAndViewsNumero(), y la clase Numero contiene el modelo de la base de datos donde se encuentran los atributos y métodos de acceso a los mismos.

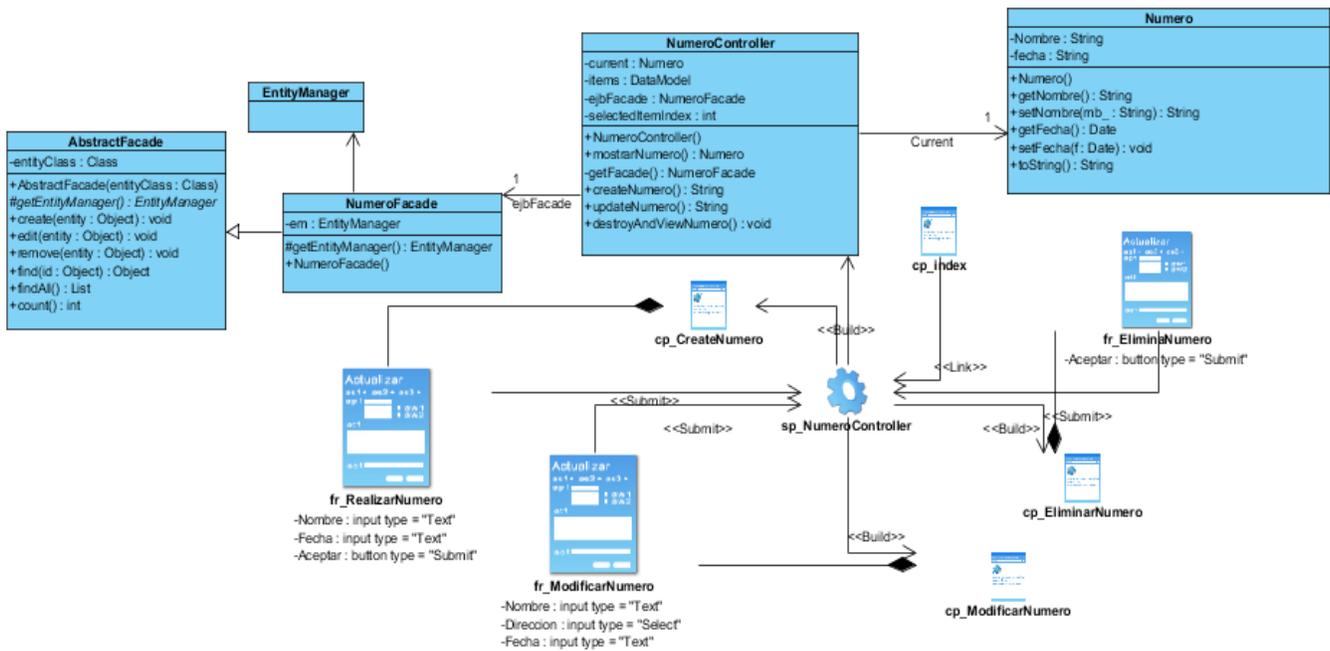


Figura 4 "Diagrama de clases del diseño gestionar número".

Patrones de Diseño

Los patrones GRASP General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades en sus siglas en inglés) describen los principios fundamentales para asignar responsabilidades a los objetos.

Principales patrones GRASP:

- Experto: es el patrón más utilizado en la asignación de responsabilidades, es un principio básico que suele utilizarse en diseño orientado a objetos. Con la utilización de este patrón se conserva el encapsulamiento, los objetos se valen de su propia información para hacer lo que se pide. (Grau Rodríguez 2019)

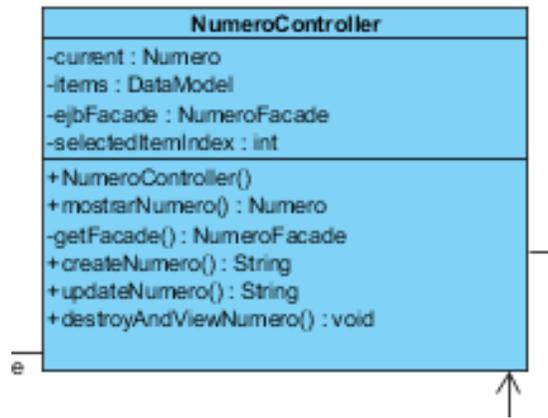


Figura 5 "Patrón Experto"

- Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. (Grau Rodríguez 2019)

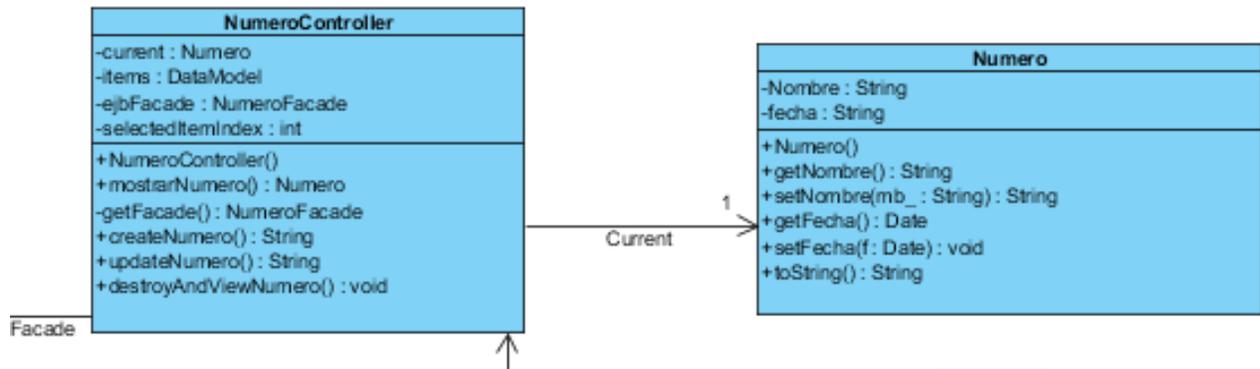


Figura 6 "Patrón Creador"

- Controlador: Garantiza que la empresa o los procesos de dominio sean manejados por la capa de los objetos del dominio y no por la interfaz. Al delegar a un controlador la responsabilidad de la operación de un sistema entre las clases del dominio favorece la reutilización de la lógica para manejar los procesos afines del negocio en aplicaciones futuras.

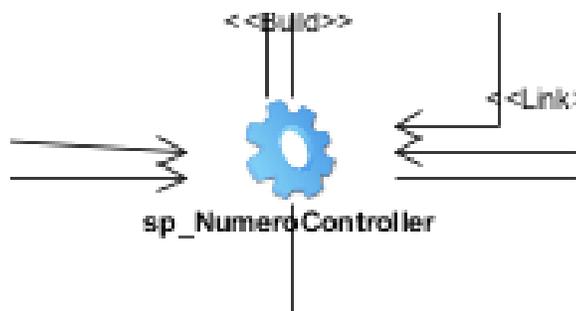


Figura 7 "Patrón Creador"

Patrones GOF

GOF es un acrónimo que significa Gang of Four o Banda de los Cuatro, refiriéndose a los autores del libro Design Patterns.

Decorador: Este patrón es de tipo estructural y permite añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender funcionalidad (Andrés López 2022).

El patrón decorador se pone de manifiesto con el uso de plantillas "twig", las cuales heredan de la plantilla "base.html.twig", redefiniéndose mediante elementos que constituyen otros objetos. Permite declarar regiones o bloques editables dentro de los cuales se realizan las modificaciones particulares de cada página, manteniendo una apariencia homogénea entre todas. Se aplica con la intención de proporcionar una forma flexible de introducir o eliminar funcionalidades a las páginas de la aplicación, solo estableciendo los bloques que pueden ser redefinidos sin modificar su apariencia externa o su función.

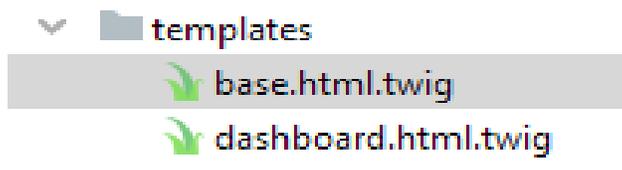


Figura 8 "Patrón Decorador"

Conclusiones del capítulo

En el presente capítulo se realizó un modelado del dominio donde se logró la obtención y la especificación de los requisitos donde se analizaron las principales funcionalidades que deben estar presentes en el diseño de la propuesta de solución del sistema que dieron lugar al modelado de casos de uso con su respectiva descripción, lo que permitió el entendimiento entre el equipo de desarrollo y el cliente de la aplicación en función de lo que el sistema debe realizar y las características que debe poseer. Además, se hace uso del patrón Modelo Vista Controlador que es el patrón arquitectónico empleado para lograr un mejor funcionamiento del sistema, así como los patrones de diseño más factibles para la construcción de este.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA

En este capítulo se realizarán la implementación y prueba de la solución propuesta, se analizarán los artefactos principales de la misma, así como el modelo de implementación que incluye el diagrama de componentes donde se ejemplifica la distribución física del sistema basado en la arquitectura utilizada. Además, se reflejan los resultados obtenidos en las distintas pruebas realizadas al sistema. validar la solución es un proceso que asegura la correcta migración de los datos a un sistema deseado.

III.1 Modelo de implementación.

El modelo de implementación describe cómo los elementos del diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y los lenguajes de programación utilizados, y cómo dependen los componentes uno de otros (Montoya Castillo, Rivero Guevara et al. 2011).

III.1.1 Diagrama de Componentes

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. Representa como cada subsistema es dividido en componentes y muestra las dependencias entre ellos (Mar Cornelio, Bron Fonseca et al. 2020).

En la siguiente figura se ilustra el diagrama de componentes, este constituye una representación de las principales dependencias entre los elementos fundamentales que conforman la estructuración del sistema ubicados en los paquetes “Views”, “Controller”, “Entity”, “Doctrine” y “Data Base”.

Las vistas agrupan las extensiones “html.twig”, las cuales heredan del componente “base.html.twig”. En el paquete “Controller” se encuentra el “DefaultController” encargado de manejar los elementos referentes al negocio. En el paquete “Entity” se manejan los subpaquetes “Main” y “ojsOld” que son las bases de datos con las que se interactúa en el sistema para la realización de la migración de datos.

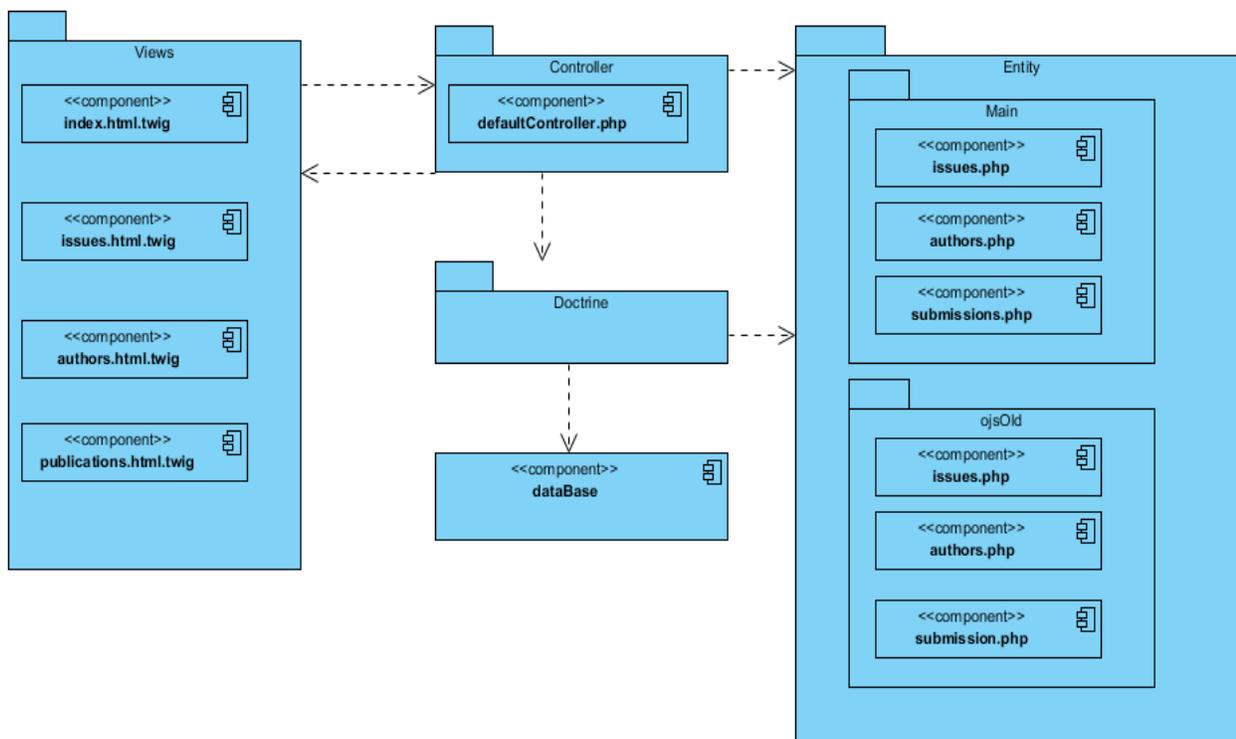


Figura 9 "Diagrama de Componentes"

III.1.2 Modelo de datos

Un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, permite describir el tipo de datos que incluye la base y la forma en que se relacionan, las restricciones de integridad y las operaciones de manipulación de los datos.

Terminada la primera etapa se realiza el mapeo de datos, definiendo qué datos de la fuente origen se guardarán en qué campos de la fuente destino, se verificó que los formatos de los mismos sean compatibles. Debido al alto número de tablas en la base de datos de OJS se realizó un análisis, donde se creó a modo de resumen un diagrama de entidad relación según las tablas de la base de datos con las que se interactuó y las que intervienen en la migración de datos.

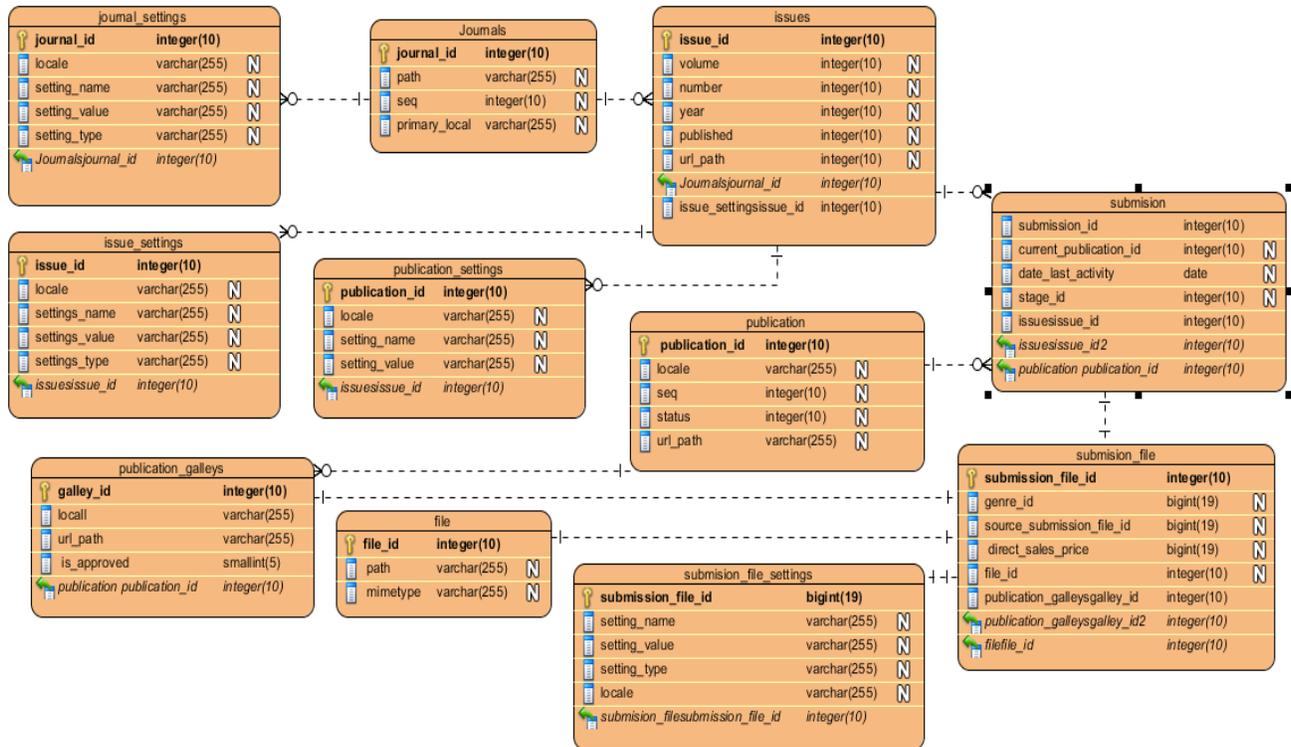


Figura 10 "Diagrama entidad relación del producto de trabajo"

III.1.3 Diagrama de Despliegue

El diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta.

El propósito del diagrama de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. Permite el mapeo de procesos dentro de los nodos, asegurando la distribución del comportamiento a través de aquellos que son representados.

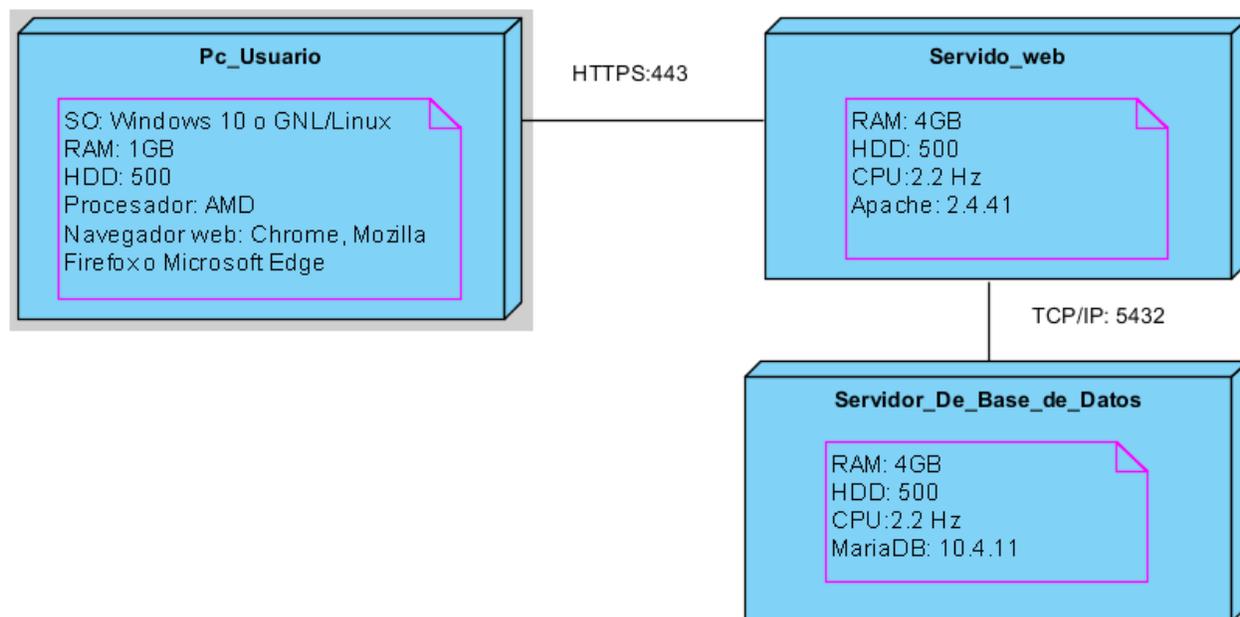


Figura 11 "Diagrama de despliegue"

Tabla 3" Descripción de los nodos correspondientes al Diagrama de Despliegue"

Nodos	Descripción
PC_Usuario	Nodo que representa aquellas PC que pueden ser utilizadas por los usuarios para acceder a la aplicación.
Servidor Web	Nodo que representa el servidor donde está alojada la revista en cuestión.
Servidor de Base de Datos	Nodo que representa el servidor donde se encuentra la Base de Datos que contiene los datos persistentes de la aplicación.

III.2 Migración de datos.

Se hace necesario realizar una migración de datos debido a que la nueva versión de la RCCI necesita una actualización de contenidos, los cuales están almacenados en la versión anterior de dicha revista

Proceso de migración de datos.

Análisis de la información: se realiza el levantamiento de la información y se verifica la calidad de los datos en la fuente de origen, con el propósito de definir los datos deseados y los tiempos reales en el resto de las etapas del proceso. Al término de esta etapa se identificó la información deseada y se determinó que la misma tenía la calidad necesaria.

Mapeo de datos: se diseña el mapeo de datos donde se hace coincidir las fuentes de origen y la fuente

destino.

Simulación de carga: el objetivo de esta etapa es migrar los datos deseados.

Cargas Reales: se realiza la migración de los datos desde la base de datos de la versión 2.4.6.0 hacia la versión actual.

Revisión de resultado: se realizan pruebas de aceptación, por parte del cliente, las cuales garantizan no solo la correcta migración de los datos, sino que también garantiza que los datos migrados no afecten el correcto funcionamiento de la aplicación.

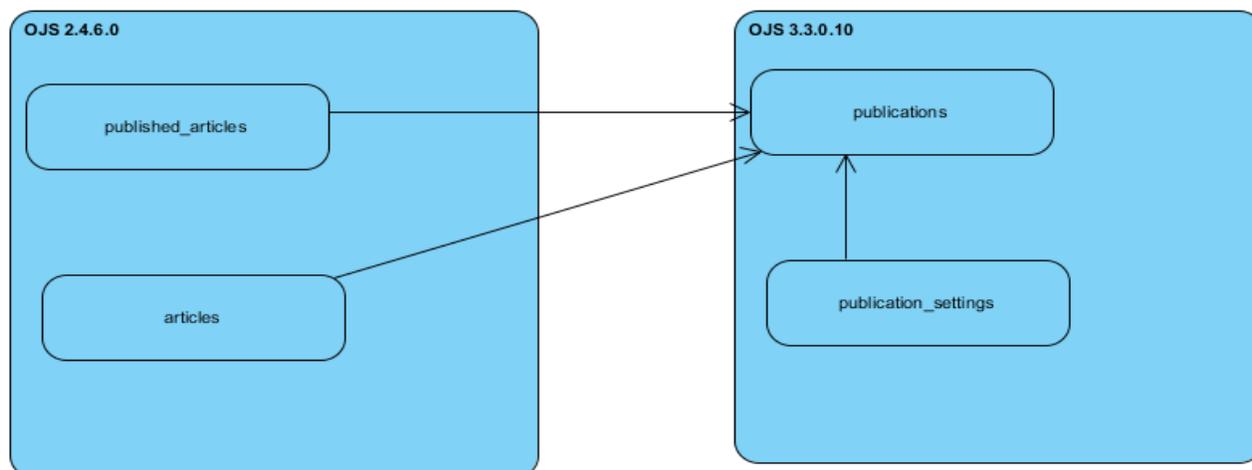


Figura 12 "Relación de las bases de datos implícitas en la migración"

Tabla 4 "Ejemplo de relación entre tablas y atributos"

Campos en la tabla OJS 2	Campos en la tabla OJS 3
Publications	Articles
article_id	publication_id
access_status	access_status
date_published	date_published
last_modified	last_modified
Locale	Locale
submission_id.primary_contact.primary_contact_id	primary_contact_id
section_id	section_id
Seq	Seq
submission_id	submission_id
Status	Status
	url_path
Versión	Versión

Descripción del mapeo: Tanto en la tabla *publications* como en la tabla *articles* existen los mismos campos. Se hizo coincidir la relación entre las mismas ya que se refieren a los mismos datos.

III.2.1 Estándares de codificación

El objetivo de los estándares de codificación de software es inculcar prácticas de programación probadas que conduzcan a un código seguro, confiable, comprobable y mantenible. Por lo general, esto significa evitar las prácticas de codificación inseguras conocidas o el código que puede causar un comportamiento impredecible.

Tamaño y organización de las líneas de código:

La longitud de línea es aproximadamente de 80 caracteres. En caso de que una expresión ocupe más de este rango, podrá romper o dividirse en una nueva línea y deberán estar todas alineadas con respecto a la anterior, para mantener la legibilidad del código.

Comentar el código:

Los comentarios de documentación suelen escribirse al principio de un archivo o de cada función y se utilizan para generar documentación de ayuda a través de aplicaciones que extraen la información a partir de las etiquetas empleadas.

Indentación y espacios en blanco:

Para facilitar la comprensión del código se utilizan tabuladores o espacios en blanco en determinadas líneas del mismo. Las líneas no deben tener espacios en blanco al final. Se utiliza una indentación de dos espacios.

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 son necesarias si se desean incluir variables dentro de las cadenas de texto o cuando el texto puede incluir alguna comilla simple.

Nomenclaturas utilizadas:

PascalCase establece que los nombres de los identificadores, las variables, métodos y clases están compuestos por una o más palabras juntas. Con ella cada palabra inicia con letra mayúscula y el resto en minúscula. Todas las clases están nombradas con la utilización del estándar PascalCase, nombrándolas de acuerdo al propósito y la función que corresponda. Ejemplo: las clases FilterGroup, FilterSettings, IssuesGalley.

CamelCase es similar a PascalCase con diferencia en la letra inicial del identificador no comienza con mayúscula. Esta notación se utilizó para el nombre de funciones, atributos y variables.

Ejemplo: articleGalleys, articlesFile, publishedArticles.

III.3 Pruebas

Las pruebas de software son las investigaciones empíricas y técnicas que comprenden una fase muy importante del proceso de desarrollo. Tienen como objetivo fundamental determinar y asegurar la calidad, fiabilidad y robustez de un sistema, dentro del contexto o escenario donde está previsto que sea utilizado.

Pruebas de caja negra:

Las pruebas de caja negra, también llamadas pruebas de comportamiento o pruebas funcionales, se centran en los requisitos funcionales del software. Es decir, las técnicas de pruebas de caja negra permiten derivar conjuntos de condiciones de entrada que ejercitarán plenamente todos los requisitos funcionales de un programa. Las pruebas de caja negra no son una alternativa a las técnicas de caja blanca de caja blanca. Más bien, es un enfoque complementario que probablemente descubrirá una clase de errores diferente a la de las técnicas de caja blanca diferente de los métodos de caja blanca (Pressman 2005).

Pruebas de caja blanca:

Esta prueba se basa en el conocimiento del funcionamiento interno del sistema, podemos probar que todos los módulos encajan unos con otros, es decir, desde una visión interna.

Una prueba unitaria o test unitario consiste en fragmentos de código que prueban otras unidades de código de una aplicación como son: el diseño estructurado de una función o un procedimiento, en diseño orientado a objetos, una clase. Cuando una aplicación supera todas sus pruebas unitarias, al menos puede confiar en que su función de bajo nivel es correcta.

III.3.1 Pruebas unitarias, camino básico.

La complejidad ciclomática es una métrica o medición de software que proporciona una evaluación cuantitativa de la complejidad lógica de un programa. La complejidad ciclomática aplicada a las pruebas de camino básico permite definir el número de caminos independientes del conjunto básico de un programa y nos permitirá saber el número de pruebas a realizar para asegurarnos de que se ejecuta al menos una vez todas las líneas del programa.

```

public function migratePublicationsAction(): Response
{
    $emV3 = $this->getDoctrine()->getManager();
    $emV2 = $this->getDoctrine()->getManager( name: 'ojs_old');
    $settingTotal = 0;
    //Cargar todas las publicaciones
    $publishedArticles = $this->getDoctrine()
        ->getRepository( persistentObject: \App\Entity\Ojs_old\PublishedArticles::class, persistentManagerName: 'ojs_old')
        ->findAll();
    $articles = $this->getDoctrine()
        ->getRepository( persistentObject: \App\Entity\Ojs_old\Articles::class, persistentManagerName: 'ojs_old')
        ->findAll();
    $articlesKeyId = [];
    foreach ($articles as $article) {...}
    $articleSettings = $this->getDoctrine()
        ->getRepository( persistentObject: \App\Entity\Ojs_old\ArticleSettings::class, persistentManagerName: 'ojs_old')
        ->findAll();
    $articleSettingsKeyId = [];
    foreach ($articleSettings as $articleSetting) {...}
    $submissionId = 1;
    $total = 0;
    //Sacar todas los Article_Galleys para relacionar los ficheros
    $articleGalleys = $this->getDoctrine()
        ->getRepository( persistentObject: \App\Entity\Ojs_old\ArticleGalleys::class, persistentManagerName: 'ojs_old')
        ->findAll();

        ->findAll();
    $articlesGalleysKeyId = [];
    /** @var ArticleGalleys $articleGalley */
    foreach ($articleGalleys as $articleGalley) {...}
    $articlesFiles = $this->getDoctrine()
        ->getRepository( persistentObject: \App\Entity\Ojs_old\ArticleFiles::class, persistentManagerName: 'ojs_old')
        ->findAll();
    $articlesFilesKeyId = [];
    foreach ($articlesFiles as $articlesFile) {...}

    foreach ($publishedArticles as $publishedArticle) {...}
    $emV3->flush();

    return new JsonResponse([
        'has_error' => false,
        'message' => '',
        'resultado' => $total.' publicaciones',
    ]);
}

```

Figura 13 "Fragmento de código del método migratePublicationsAction ()"

En la figura siguiente se muestra el grafo obtenido:

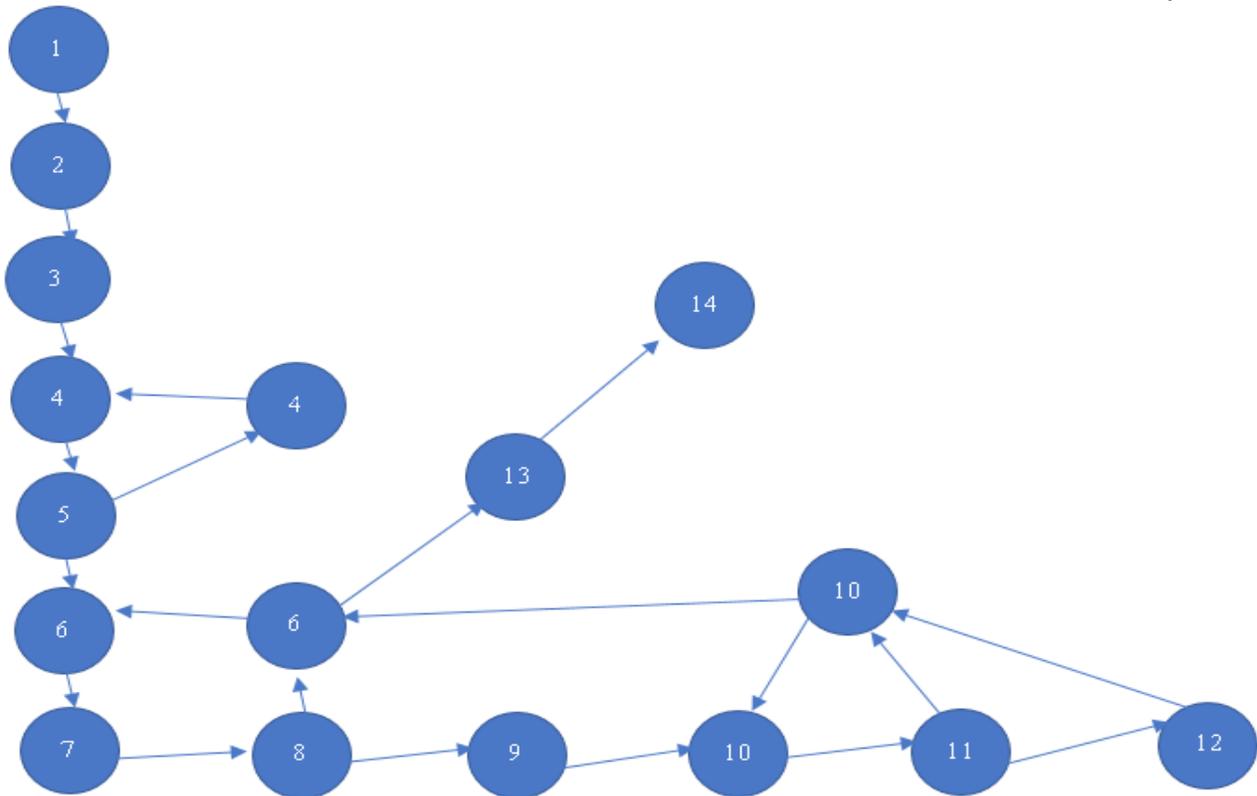


Figura 14 "Grafo de flujo del método migratePublicationsAction ()"

La complejidad ciclomática $V(G)$ para un grafo de flujo G se define como $V(G) = E - N + 2$ donde E es el número de aristas del grafo de flujo y N el número de nodos del grafo de flujo.

También se define como $V(G) = P + 1$ donde P es el número de nodos predicado contenidos en el grafo de flujo.

La complejidad ciclomática también se corresponde con el número de regiones del grafo de flujo (r).

$$V(G) = (E - N) + 2 = (21 - 17) + 2 = 4 + 2 = \underline{6}$$

$$V(G) = P + 1 = 5 + 1 = \underline{6}$$

$$r = \underline{6}$$

Después de haber calculado las tres variantes para obtener la complejidad ciclomática del grafo se puede deducir que existen 6 caminos posibles de flujo. A continuación, se describen las rutas independientes.

C#1 = 1, 2, 3, 4, 5, 6, 7, 8, 6', 13, 14.

C#2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 10', 6', 13, 14.

C#3 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 10', 6', 13, 14.

C#4 = 1, 2, 3, 4, 5, 4', 5, 6, 7, 8, 6', 13, 14.

C#5 = 1, 2, 3, 4, 5, 6, 7, 8, 6', 6, 7, 8, 6', 13, 14.

C#6 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 10', 10, 11, 12, 10', 6', 13, 14.

III 3.2 Pruebas de caja negra.

El particionamiento de equivalencia es un método de prueba de caja negra que divide el do- principal de entrada de un programa en clases de datos de las que se pueden derivar casos de prueba principal de un programa en clases de datos de las que se pueden derivar casos de prueba. Un caso de prueba ideal descubre por sí solo una clase de errores (por ejemplo, la introducción incorrecta de todos los datos de caracteres) que, de otro modo, requerirían la ejecución de muchos casos de prueba antes de que se observe el error general (Pressman 2005).

Diseño de Casos de Prueba del Caso de Uso: Gestionar Publicación.

Este Caso de Uso se inicia cuando el usuario selecciona la opción de crear una nueva publicación y finaliza con la realización de una de las siguientes operaciones sobre una solicitud: Crear, Actualizar, Mostrar o Eliminar.

Condiciones de ejecución: El usuario debe estar autenticado en el sistema y debe tener permisos para realizar las mencionadas operaciones. Debe seleccionarse la publicación sobre la cual se pretende realizar una de las siguientes acciones: Crear, Actualizar, Mostrar o Eliminar.

Tabla 5 "Secciones a probar en el Caso de Uso del Sistema Gestionar publicación"

Caso de Prueba		
Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: "Crear Publicación"	EC 1.1: El usuario inserta y selecciona datos correctos en los campos de entrada	El sistema almacena una nueva Solicitud. El sistema muestra un mensaje indicando que se ha creado la Publicación por tanto debe cargar sus documentos asociados.
	EC 1.2: El usuario deja campos vacíos.	El sistema muestra un mensaje indicando que se deben especificar todos los campos para crear una publicación.
	EC 1.3: El usuario introduce datos incorrectos.	El sistema muestra un mensaje indicando que existen datos incorrectos.
SC 2: "Actualizar Publicación".	EC 2.1: El usuario modifica correctamente los datos en los campos.	El sistema modifica los datos persistentes de la publicación y muestra un mensaje confirmando el éxito de la operación.
	EC 2.2: El usuario deja campos vacíos.	El sistema muestra un mensaje indicando que todos los campos son requeridos.
	EC 2.3: El usuario introduce datos incorrectos.	El sistema muestra un mensaje indicando que existen datos incorrectos.

SC 3: “Mostrar Publicación”.	EC 3.1: Mostrar detalles de una publicación.	El sistema muestra los datos de una publicación dada así como la posibilidad de descargar sus documentos asociados y ver sus recomendaciones, si estas existen.
SC 4: “Eliminar Publicación”.	EC 4.1: Eliminar una publicación.	El sistema elimina la publicación en cuestión, sus documentos asociados y recomendaciones. Muestra un mensaje de confirmación.

Tabla 6 “Descripción de las Variables”

No.	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Volumen	Numérico	No	Debe ser un número natural diferente de cero.
2	Número	Numérico	No	Debe ser un número natural diferente de cero.
3	Año	Numérico	No	Debe ser un número natural diferente de cero y mayor que 2000.
4	Fecha	Data_time	No	Puede ser una combinación de caracteres, letras, números, signos, símbolos.

Resultados

Con la ejecución de las pruebas de Caja Negra al caso de prueba anterior, se identificó en qué medida satisface la aplicación las funcionalidades implementadas. Una vez concluida la técnica de partición de equivalencia se obtuvo como resultado un 95% de resultados satisfactorios y un 5% de inconformidades.

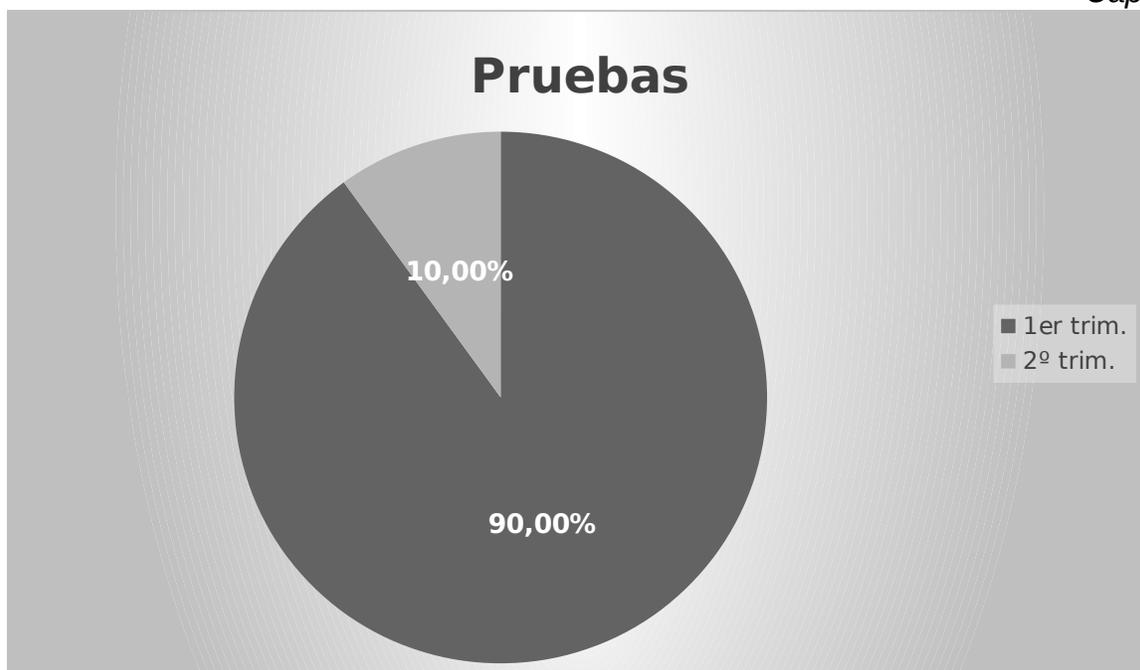


Figura 15 "Representación de los resultados de prueba"

Luego de corregidas las No Conformidades después de una segunda iteración se decide la realización de una tercera iteración de los diseños anteriormente aplicados obteniéndose resultados satisfactorios, por lo que se determina la no realización de una nueva iteración ya que se comprobó que el sistema cumple con los objetivos propuestos.

III.3.3 Pruebas de Aceptación

Las pruebas de aceptación representan aquella fase del ciclo de vida de desarrollo de software en el que el equipo de desarrollo y el área usuaria de un sistema de información tienen que garantizar que el sistema desarrollado se corresponde con los requerimientos definidos (GONZÁLEZ, DOMINGUEZ et al. 2018).

Las pruebas de aceptación Alfa se son una forma de prueba de aceptación del usuario. El sistema desarrollado fue sometido a pruebas de aceptación por el Departamento de Informatización de la UCI, que junto al equipo de desarrollo y los especialistas de dicho centro lograron verificar las funcionalidades del sistema y de informar las deficiencias y no conformidades encontradas.

III.3.4 Pruebas de rendimiento.

Las pruebas de rendimiento de software establecen una comprobación del funcionamiento del sistema ante distintos escenarios de prueba lo que quiere decir que se basan en garantizar la calidad del software en condiciones particulares de trabajo.

La imagen muestra los resultados obtenidos como parte de la aplicación de la prueba de carga, realizada con la herramienta JMeter 2.4 con una simulación de 20 usuarios concurrentes en el sistema. Se reflejan datos como el tiempo de respuesta en milisegundos (ms) y los errores mediante la columna Status para cada petición y la media del tiempo de respuesta, que no sobrepasa los 50 segundos.

Muestra #	Start Time	Thread Name	Label	Tiempo de Muestra (ms)	Status	Bytes
80	23:58:15.652	Grupo de Hilos 1-80	/text/form_field	40970		6005
81	23:58:15.745	Grupo de Hilos 1-94	/text/form_field	49939		6005
82	23:58:15.674	Grupo de Hilos 1-84	/text/form_field	52212		6005
83	23:58:15.669	Grupo de Hilos 1-83	/text/form_field	53273		6005
84	23:58:15.657	Grupo de Hilos 1-81	/text/form_field	58378		6005
85	23:58:15.641	Grupo de Hilos 1-78	/text/form_field	52402		6005
86	23:58:15.687	Grupo de Hilos 1-86	/text/form_field	54374		6005
87	23:58:15.647	Grupo de Hilos 1-79	/text/form_field	56496		6005
88	23:58:15.759	Grupo de Hilos 1-97	/text/form_field	50430		6005
89	23:58:15.718	Grupo de Hilos 1-91	/text/form_field	54504		6005
90	23:58:15.610	Grupo de Hilos 1-76	/text/form_field	54514		6005
91	23:58:15.684	Grupo de Hilos 1-85	/text/form_field	53693		6005
92	23:58:15.728	Grupo de Hilos 1-92	/text/form_field	54954		6005
93	23:58:15.755	Grupo de Hilos 1-96	/text/form_field	57937		6005
94	23:58:15.615	Grupo de Hilos 1-73	/text/form_field	50920		6005
95	23:58:15.693	Grupo de Hilos 1-87	/text/form_field	62062		6005
96	23:58:15.619	Grupo de Hilos 1-75	/text/form_field	55955		6005
97	23:58:15.618	Grupo de Hilos 1-72	/text/form_field	52952		6005
98	23:58:15.704	Grupo de Hilos 1-89	/text/form_field	60240		6005
99	23:58:15.663	Grupo de Hilos 1-82	/text/form_field	65295		6005
100	23:58:15.751	Grupo de Hilos 1-95	/text/form_field	63223		6005

No. de Muestras 20 Última Muestra 63223 Media 48218 Desviación 15241

Figura 16 "Prueba de carga utilizando JMeter"

Se realizan pruebas funcionales al proceso de migración, para determinar si los datos son traspasados con éxito a la nueva base de datos. Fue necesario realizar dos iteraciones de pruebas debido a la ocurrencia de errores en la primera iteración, donde se detectaron errores. Dichos errores se clasificaron en tres grupos, tuplas con datos nulos, tuplas con datos duplicados y clases sin llaves primarias. Antes de realizar la segunda iteración se modificó el diseño de algunas transformaciones del proceso para eliminar los errores del tipo tuplas con datos duplicados, se levantaron las restricciones de la base de datos destino para eliminar los errores del tipo tuplas con datos nulos. Además, se crearon llaves primarias compuestas ya que existían campos que por sí solos no cumplen con la condición para ser clave primaria. En la realización de la segunda iteración se obtuvo como resultado la migración del 100% de los datos deseados.

Conclusiones del capítulo Durante el presente capítulo se desarrolló todo el proceso de implementación y pruebas del sistema propuesto donde se plasmaron el diseño de componentes y el despliegue de la solución. Se realizaron distintos tipos de pruebas (Caja Blanca y Caja Negra) aplicando las técnicas de Camino Básico y Partición de Equivalencia, las cuales verificaron las funcionalidades del sistema demostrando que el mismo cumple con dichas funcionalidades

CONCLUSIONES FINALES

A partir de la investigación realizada y el desarrollo del presente trabajo se consiguió arribar a las siguientes conclusiones:

- Con el estudio del estado del arte y las soluciones informáticas existentes que se relacionan con la investigación en cuestión, se logró sentar las bases para una posterior solución del problema.
- Las tecnologías, herramientas y lenguajes utilizados en la construcción de la nueva versión del sitio, por ser de software libres, se corresponden con las políticas de soberanía tecnológica que impulsa la Universidad de Ciencias Informáticas y nuestro país.
- Se concluyó que era propicio utilizar el CMS Open Journal System por las características descritas en anteriores capítulos y porque la primera versión de la RCCI está montada en dicho CMS lo que facilitaría la migración de su contenido a la actual versión.
- El Sitio permitirá una mayor visibilidad de los contenidos que publica la RCCI, proporcionándole presencia en Internet y contribuyendo a incrementar su prestigio y reconocimiento nacional e internacional.
- La realización de distintos tipos de pruebas (Caja Blanca y Caja Negra, Rendimiento y Aceptación) permitió la verificación de las funcionalidades del sistema, demostrando que el mismo cumple con dichas funcionalidades.

RECOMENDACIONES

Utilizar la estrategia diseñada en esta investigación para las posteriores migraciones realizadas entre las versiones 2.4.6.0 y 3.3.0.10 de OJS.

Utilizar la estrategia diseñada en esta investigación para la migración de datos entre estructuras de almacenamiento con características iguales o similares a las definidas en la investigación.

Aprovechar la arquitectura de la propuesta de migración de datos para agregar otros metadatos que garanticen la visualización de los artículos durante toda la etapa de publicación u otro tipo de contenido que eventualmente sea agregado a la nueva versión.

REFERENCIAS BIBLIOGRÁFICAS

- Acosta Rodríguez, H. J. A. (1998). "Publicaciones electrónicas. Experiencias de una entidad del sector de la salud." **6**(2): 107-117.
- Adithela, S. P., et al. (2018). Django content management system evaluation and integration with apache airavata. Proceedings of the Practice and Experience on Advanced Research Computing: 1-4.
- Aguirre García, J. D. J. R. C. C. M. (2012). "Respecto a las Indizaciones e Indexaciones." **15**(1): 5-5.
- Alonso-Aranda, C. (2019). "MODELO-VISTA-CONTROLADOR. LENGUAJE UML."
- Andrés López, P. d. (2022). "Plataforma de gestión para “The great quiz of databases”."
- Arias, M. A. (2013). Introducción a PHP, IT Campus Academy.
- Barker, P. J. T. e. l. (1996). "Living books and dynamic electronic libraries." **14**(6): 491-501.
- Blanco, N. (2018). "Qué es Doctrine y cómo funciona."
- Borges, R. M. M. J. R. C. d. T. d. l. S. (2017). "Ciencia y comunicación científica." **7**(4).
- Caballero, C. E. A. and S. J. T. I. y. A. Rojas (2018). "Comparacion Prince2 y OpenUp para desarrollo de software." **6**(1): 77-83.
- Carbonell, M. J. I. A. L. C. D. B. D. D. (2015). "Phpmyadmin."
- Carvajal-Tapia, A. E. and E. J. R. d. l. F. d. M. Carvajal-Rodríguez (2018). "Status of scientific production in Medicine in South America. 1996-2016." **66**(4): 595-600.
- Castro Aguilera, T. E. (2017). Sistema de Gestión de Incidencias del Grupo Azucarero AZCUBA, Universidad de las Ciencias Informáticas. Facultad 6.
- Díaz Lazo, J., et al. (2011). "Impacto de las tecnologías de la información y las comunicaciones (TIC) para disminuir la brecha digital en la sociedad actual." **32**(1): 81-90.
- Díaz, R. M. L. J. R. C. d. H., Inmunología y Hemoterapia (2016). "La redacción de un artículo científico." **32**(1): 57-69.
- Dimes, T. J. E. B. I. (2015). "JavaScript una guía de aprendizaje para el lenguaje de programación JavaScript Dimes T, editor."
- Duperet Cabrera, E., et al. (2015). "Importancia de los repositorios para preservar y recuperar la información." **19**(10): 1283-1290.

Referencias bibliográficas

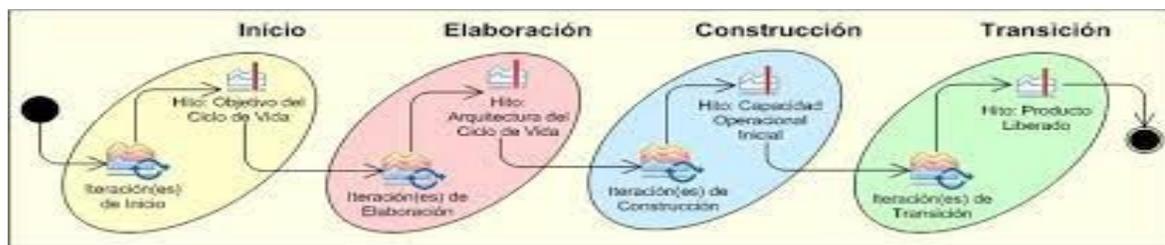
- González-Marín, N. R. J. A. d. O. and C. d. C. y. Cuello (2019). "El sistema abierto de gestión y publicación de revistas "OJS"." **47**(3): 151-151.
- González, A. J. R. d. h. w. h. c. s.-a.-i. c.-a. c.-e.-u.-a.-c.-o.-p.-p.-l.-u. Í. (2018). "¿ Cómo escribir un "artículo científico" para la universidad."
- González, J. E. R., et al. (2021). "INGENIERÍA DE REQUERIMIENTOS un enfoque práctico."
- GONZÁLEZ, J. P., et al. (2018). "Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental."
- Grau Rodríguez, D. (2019). Sistema inteligente para el agrupamiento de patrones de diseño de recursos educativos en lenguajes potenciales, Universidad de las Ciencias Informáticas. Facultad 4.
- Hernández, D. Y. H., et al. (2020). "Importancia de las revistas de acceso abierto: la indización como meta fundamental." (26): 81-98.
- Hidalgo, L. A. R. (2010). Facultad 10, Universidad de las Ciencias Informáticas.
- Huaman, W. J. R. d. h. m. c. m. l.-.-p.-c.-d.-a.-d.-s.-d. b. b. (2018). "Los 10 patrones comunes de arquitectura de software."
- Informáticas, R. C. d. C. (2006).
- .
- Kindelán, A. H. (2019). "LAS PUBLICACIONES ELECTRÓNICAS: UN CONCEPTO, UNA CLASIFICACIÓN Y UN ANÁLISIS DE SU IMPACTO EN LOS PROFESIONALES DE LA INFORMACIÓN."
- Lozano Banqueri, J. M. (2018). "Creación y gestión de una base de datos con MySQL y phpMyAdmin."
- Macias, M. E. R. J. H. M. (2019). "El enfoque basado en proceso en la gestión editorial de las revistas científicas." **19**(3): 637-659.
- Mar Cornelio, O., et al. (2020). "Sistema de Laboratorios Remoto para el estudio de la Microbiología y Parasitología Médica." **12**(2).
- Melchor, G. M. (2006). "Revista Cubana de Ciencias Informáticas, vol. 1, núm. 1."
- Mendez Avendaño, D. A. and C. C. Zapata Tabarez (2016). "OCS gestor de ponencias."
- Montoya Castillo, A., et al. (2011). Análisis, diseño e implementación del módulo transmisiones del Sistema de Investigación e Información Policial.

Referencias bibliográficas

- Murillo, D. and D. Saavedra (2022). "Importancia de la Indexación y medición de las revistas científicas en la UTP."
- Potencier, F. and F. J. P. o. S. h. f. p. o. a. w.-s. Zaninotto (2012). "symfony."
- Pressman, J. D. (2002). Last resort: psychosurgery and the limits of medicine, Cambridge University Press.
- Pressman, R. S. (2005). Software engineering: a practitioner's approach, Palgrave macmillan.
- Prihandoyo, M. T. J. J. I. J. P. I. (2018). "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web." **3**(1): 126-129.
- Project, P. K. (2022). "Open Journal Systems."
- Pública, R. C. d. S. (2022). "Revista Cubana de Salud Pública."
- Revista de Ingeniería Electrónica, A. y. C. (2022). "Revista de Ingeniería Electrónica, Automática y Comunicaciones."
- Ríos, S., et al. (2013). Aplicación de la metodología openup en el desarrollo del sistema de difusión de gestión del conocimiento de la espe, 10, Escuela Politécnica del Ejército, Ecuador. Recuperado el.
- Torres Cala, Y., et al. (2018). "El proceso de innovación en el sector forestal cubano y su contribución a la economía nacional." **6**(2): 146-161.
- Travieso Aguiar, M. J. a. (2003). "Las publicaciones electrónicas: una revolución en el siglo XXI." **11**(2): 1-2.
- Ulco Simbaña, L. E. and P. F. J. C. Baldeón Egas (2020). "Las tecnologías de la información y comunicación y su influencia en la lectoescritura." **16**(73): 426-433.
- Vázquez, Á. M. D. (2018). "GESTIÓN Y EDICIÓN DE REVISTAS ACADÉMICAS CON SOFTWARE LIBRE. EL USO DE OPEN JOURNAL SYSTEMS 3."
- Vázquez Ingelmo, A., et al. (2019). "Resultados preliminares tras tres años aplicando aprendizaje basado en proyectos en ingeniería del software."
- Venegas Bravo, J. A. (2022). "Análisis comparativo de rendimiento de gestores de base de datos NOS-QL documentales."
- Wang, Y. and Y. Solihin (2013). XAMP: An eXtensible Analytical Model Platform. 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), IEEE.
- Yolanda, C. S. (2019). "La redacción de un artículo científico."

ANEXOS

Anexo # 1: Metodología Open Up



Anexo # 2: Interfaz del Nuevo sitio de la RCCI.



Anexo # 3: Interfaz de la herramienta para la migración de datos OJS Migrator.

OJS Migrator Dashboard Instrucciones

Importante
Para comenzar el proceso de migración, primero se debe agregar manualmente las revistas existentes en la base de datos vieja. Esto es con el objetivo de que se realice todas las inicializaciones necesarias por parte del **OJS 3.x** y evitar errores a la hora de importar el resto de los datos.

PASOS	RESULTADOS	COMPLETADO
 Migrar Ediciones(Issues)	66 ediciones	
 Migrar Autores	4320 autores	
 Migrar Publicaciones	730 publicaciones	
 Clear All Data		

Al terminar de migrar los componentes debe limpiar la cache y las sesiones en el panel de administración del **OJS 3.x**

Anexo 4: Acta de aceptación de la propuesta de solución.

La Habana, 24 de noviembre de 2022
 "Año 64 de la Revolución"

ACTA DE ACEPTACIÓN

De una parte, la Dirección de Informatización, en lo sucesivo DIN, de la Universidad de las Ciencias Informáticas, representado en este acto por: Ing. Yosbel Falero Vento, y de otra parte la estudiante: Rosaida López Villavicencio.

Primero: Que en cumplimiento de los requisitos funcionales han sido efectuadas las implementaciones correspondientes.

CONSIDERANDO: Que los hitos realizados han sido desarrollados con la calidad requerida y bajo las condiciones pactadas y aprobadas por las partes.

CONSIDERANDO: Que los hitos que se han ejecutado cumplen con los requerimientos establecidos.

CONSIDERANDO: Que el producto brinda una herramienta de migración de datos que incide en la calidad de la gestión editorial de la Revista Cubana de Ciencias Informáticas.

POR TANTO: Las partes acuerdan formalizar mediante la presente Acta, la aceptación del producto:

"Revista Cubana de las Ciencias Informáticas 2.0"

Y para que así conste, se extiende la presente Acta en dos (2) ejemplares, rubricados por las partes.


 Estudiante Rosaida López Villavicencio


 Ing. Yosbel Falero Vento

Entrega



Recibe

Universidad de las Ciencias Informáticas
 Carretera San Antonio Km 2 ½ Torrens. Boyero,
 La Habana, Cuba.
 Teléfono: +53 7 837 7020 y +53 7 837 7021