



Facultad de Ciencias y Tecnologías Computacionales

Sistema Informático para el Observatorio Científico, Tecnológico y de Innovación en el MINCEX.

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

Autora:

Yarianna Montesino Martinez.

Tutor(es):

Msc. Yaneisis Pérez Heredia.

La Habana, mayo de 2022

Año 64 de la Revolución

DECLARACIÓN DE AUTORÍA

Yarianna Montesino Martinez, autora del trabajo de diploma con título “**Sistema Informático para el Observatorio Científico, Tecnológico y de Innovación para el MINCEX**” concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autores de su contenido. Para que así conste firma(n) la presente a los <día> días del mes de <mes> del año <año>.

Yarianna Montesino Martinez

Firma del Autor

Msc. Yaneisis Pérez Heredia.

Firma del Tutor

DATOS DE CONTACTO

Nombre y apellidos del autor: Yarianna Montesino Martinez

Correo electrónico: yariannamm@estudiantes.uci.cu

Institución a la que pertenece: Universidad de las Ciencias Informáticas

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de la Habana, Cuba.

Nombre y apellidos del tutor: Yaneisis Pérez Heredia

Correo electrónico: yheredia@uci.cu

Títulos académicos: Ms. Informática Aplicada e Ing. Ciencias Informáticas

Formación de Postgrado: Maestría en Informática Aplicada

Lugar de trabajo: Universidad de las Ciencias Informáticas

Responsabilidades Laborales: Profesora y Profesora Guía

Experiencia Profesional: Cuenta con 18 años de experiencia como profesora universitaria impartiendo asignaturas como Programación e Ingeniería de Software; diversos artículos presentados en eventos científicos; cumplió misión internacionalista por dos años en Angola.

AGRADECIMIENTOS

Mis más sinceros agradecimientos:

A la Universidad de las Ciencias Informáticas por darme la oportunidad de formarme como profesional.

A mis padres por siempre apoyarme en mis decisiones y por sus sacrificios para que mi futuro fuera mejor.

A mis hermanos Yosbel y Yaima por siempre estar pendientes a mi carrera y por sus consejos a pesar de la distancia.

A mi novio por apoyarme este último año y ayudarme a ver la vida desde otra perspectiva.

A toda mi familia por estar al pendiente de mi carrera en especial a mi abuela María.

A mi tutora por brindarme todos sus conocimientos y guiarme en este proceso.

A mis profesores por todo lo que me han enseñado en el transcurso de estos cinco años.

DEDICATORIA

A mis padres y hermanos.

A mis compañeros de aula que por otras razones no pudieron llegar hasta aquí.

RESUMEN

En el mundo existen disímiles sistemas informáticos dedicados almacenar y procesar información que generan los Observatorios Tecnológicos, satisfaciendo las necesidades de muchas instituciones. En este sentido el Ministerio de Comercio Exterior y la Inversión Extranjera (MINCEX), opta por la creación de un observatorio científico, tecnológico y de innovación para gestionar todo lo referente a comercio exterior, inversión extranjera y cooperación internacional. La presente investigación contiene el análisis, implementación y desarrollo de un sistema informático para el Observatorio Científico en el Mincex. Para el desarrollo de la propuesta de solución, se emplearon herramientas de código abierto como Drupal, MySQL, Apache, PHP, CSS, HTML y como metodología de desarrollo el Proceso Unificado Ágil, por sus siglas en inglés Agile Unified Process, en su variación UCI de acuerdo con las políticas de informatización de la Universidad de las Ciencias Informáticas. Las pruebas realizadas al sistema obtenido garantizaron la confiabilidad del mismo y por tanto la entrega de una solución útil, que con su empleo fortalece la gestión de información.

Palabras claves: sistema informático, observatorio científico y de innovación, herramientas.

ABSTRACT

In the world there are dissimilar computer systems dedicated to storing and processing information generated by the Technological Observatories, satisfying the needs of many institutions. In this sense, the Ministry of Foreign Trade and Foreign Investment (MINCEX), opts for the creation of a scientific, technological and innovation observatory to manage every-thing related to foreign trade, foreign investment and international cooperation. This research contains the analysis, implementation and development of a computer system for the Scientific Observatory at Mincex. For the development of the solution proposal, open source tools such as Drupal, MySQL, Apache, PHP, CSS, HTML were used and as a development methodology the Agile Unified Process, for its acronym in English Agile Unified Process, in its UCI variation. in accordance with the computerization policies of the University of Informatics Sciences. The tests carried out on the obtained system guaranteed its reliability and therefore the delivery of a useful solution, which with its use strengthens information management. Keywords: computer system, scientific and innovation observatory, tools.

TABLA DE CONTENIDOS

INTRODUCCIÓN	11
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO	16
1.1 Conceptos asociados a la investigación.	16
1.2 Estudios de sistemas homólogos.	17
A continuación, se realiza un estudio de sistemas homólogos encontrados de observatorios científicos, tecnológicos y de innovación.	17
1.3 Metodologías de desarrollo de software.	19
1.4 Herramientas y tecnologías.	21
1.4.1 Lenguaje de Programación.	21
Conclusiones del capítulo.	24
CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO	26
2.1 Propuesta del sistema	26
2.1.1 Modelo del Dominio	26
Descripción de los conceptos del Modelo del Dominio	27
2.2 Especificación de Requisitos	27
2.2.1 Historia de usuario	32
2.3 Diseño Arquitectónico	34
2.4 Diseño de despliegue	42
Conclusiones del capítulo	43
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	45
3.1 Diagrama de Componentes	45
3.2 Estándar de codificación	46
3.3 Verificación de la propuesta de solución	48
Conclusiones del capítulo	55
CONCLUSIONES FINALES	56
RECOMENDACIONES	57
REFERENCIAS BIBLIOGRÁFICAS	58
ANEXOS	60

ÍNDICE DE TABLAS

Tabla 1: Requisitos Funcionales	28
Tabla 2: Historia de Usuario Insertar Noticia	32
Tabla 3: Historia de Usuario Insertar Evento	33
Tabla 4: Estándares de codificación utilizados	47
Tabla 5: Caso de prueba de requisito Crear Noticia	49
Tabla 6: Caso de Prueba Crear Noticia	49
Tabla 7: Comportamiento de las no conformidades durante cada iteración.	50
Tabla 8: Resultados de prueba de seguridad	52
Tabla 9: Lista de chequeo Prueba de Usabilidad	53

ÍNDICE DE FIGURAS

Figura 1. Modelo de dominio	27
Figura 2.....	35
Figura 3: Diagrama de clase del diseño Gestionar Noticia	37
Figura 4: Diagrama de clase del diseño Gestionar evento	38
Figura 5: Diagrama de secuencia Insertar Noticia	41
Figura 6: Diagrama de secuencia Insertar Evento	41
Figura 7: Diagrama de despliegue	42
Figura 8: Diagrama de Componente	46

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

La introducción y el uso de las Tecnologías de la Información y las Comunicaciones [1] y de Internet ha significado a escala mundial, un salto vertiginoso en el desarrollo científico técnico. Desde su llegada a los escenarios nacionales se han convertido en un elemento indispensable para establecer las líneas de desarrollo de la sociedad cubana, buscando dar solución a los problemas del hombre, y han transformado la manera de trabajar, liberándolo de las cargas más pesadas, optimizando los recursos, logrando mayor productividad y eficiencia.

La Informatización de la Sociedad se define en Cuba como el proceso de utilización ordenada y masiva de las nuevas tecnologías de la Informática y las Comunicaciones para satisfacer las necesidades de información y conocimiento de la sociedad. La estrategia seguida por la máxima dirección del país, de informatizar a la sociedad cubana, tiene al ciudadano como centro de sus objetivos.

Cuba está consciente de que una sociedad para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas y procesos. En este sentido, Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las TICs y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a nuestra sociedad acercarse más hacia el objetivo de un desarrollo justo, equitativo, sostenible y alcanzable.

Los Ministerios de la República de Cuba son organismos encargados de dirigir, ejecutar y controlar la política del Estado y el Gobierno en diferentes esferas. El Ministerio del Comercio Exterior (MINCEX) surge oficialmente por la Ley N°934¹, el cual constituyó un eslabón esencial en la política económica del Estado y permitió al gobierno cubano asumir el manejo y control de las importaciones y exportaciones del país. Posteriormente mediante el Decreto Ley N° 264², se dispuso la fusión del nuevo Ministerio del Comercio Exterior y la Inversión Extranjera, como Organismo de la Administración Central del Estado, asignándole como objetivo y misión

¹ Ley N°934 se divulga el 23 de febrero de 1961.

² Ley N°264 se divulga el 2 de marzo de 2009.

la de preparar y proponer al Gobierno la política integral del Estado en cuanto a la actividad de comercio exterior, la inversión extranjera, la colaboración económica con otros países, organizaciones y asociaciones extranjeras.

Actualmente el MINCEX creó un Departamento de Ciencia, Tecnología e Innovación con el objetivo de crear una cultura basada en la generación, la apropiación y la divulgación del conocimiento, y la investigación científica, la innovación y el aprendizaje permanente. Entre los proyectos de este nuevo departamento se encuentra, crear un Observatorio Científico, Tecnológico y de Innovación.

Según (Sarmiento Reyes et al., 2019), un observatorio es una organización creada por un colectivo con el fin de seguir la evolución de un fenómeno, normalmente de carácter económico o social, desde una posición ventajosa . Sin embargo, en la actualidad un observatorio requiere de monitorear el comportamiento de la organización y su entorno, asociado a un sistema de información o de vigilancia, que le garantice estar actualizado en su sector de desempeño, donde el uso de las tecnologías de la información y los ambientes colaborativos, contribuirán a un mejor proceso de búsqueda, recuperación y análisis de la información, necesario para la ciencia de datos y la toma de decisiones [3].

Los observatorios científicos de manera general están orientados a cumplir funciones de los sistemas nacionales de innovación, y a monitorear la actividad de ciencia, tecnología e innovación (ACTI) de un conjunto de ejes temáticos importantes para un país en particular.

Según de la Vega (2007), los observatorios de ciencia pueden agruparse en 4 tipos: Los de consorcio o estructura mixta; Los tutelados por los Ministerios de Ciencia y Tecnología o vinculados a las presidencias; Estructura esencialmente académica; Redes o estructuras de cooperación multilateral. Se considera que independientemente de la tipología, la creación de observatorios de CTI debe estar relacionada con el carácter innovador y dinámico de la ciencia [4].

A partir de la creación del Departamento de Ciencia, Tecnología e Innovación surge la necesidad de organizar, recopilar, clasificar y actualizar toda la información referente a las líneas estratégicas del MINCEX de forma eficiente y segura, ya que la única vía de realizar este trabajo es vía correo electrónico; los especialistas no cuentan con el medio adecuado para brindar detalles de los proyectos de negocio. Por otra parte, toda la información se encuentra en papel impreso la cual es muy engorroso y lento a la hora de realizar búsquedas u organizaciones, ya que dicha información se almacena en archivos, debido a esto también la información no tiene una correcta seguridad y pueden ocurrir pérdidas. Para esto se propuso la creación de un observatorio científico con el fin de apoyar la toma de decisiones. El observatorio debe ser capaz de **recopilar, analizar e interpretar** determinada información, con el fin de **mejorar** el conocimiento y de constituirse en una **herramienta** valiosa para cualquier entidad.

Teniendo en cuenta lo antes planteado se identifica el siguiente **problema de la investigación**: ¿Cómo gestionar los productos informáticos y temas relevantes que permitan la obtención de información adecuada, correcta y segura aplicando la ciencia, la tecnología y la innovación en el MINCEX?

En correspondencia con el problema señalado se define como **objeto de estudio**: Proceso de gestión de información que se manejan en el Observatorio.

Enmarcándose en el **campo de acción**: Informatización de los procesos que se desarrolla en el Observatorio.

Definiéndose como **objetivo general de la investigación**: Desarrollar un Sistema informático para la gestión de información del Observatorio de la Ciencia, Tecnología e Innovación en el MINCEX. Desglosándose este en los siguientes **objetivos específicos**:

1. Elaboración del Marco Teórico de la Investigación a partir del estado existente relacionado con la aplicación de la ciencia y la técnica en el MINCEX.
2. Definir características acordes con el Observatorio Científico -Tecnológico que se necesite.
3. Análisis y diseño de la solución que mejore el uso y la aplicación de la ciencia y la técnica en el MINCEX.

4. Implementación de la solución.
5. Validación de la solución propuesta.

Para dar cumplimiento al objetivo planteado se define las siguientes **tareas de la investigación**:

1. Revisión y análisis de las bibliografías correspondientes al tema.
2. Selección de la metodología de desarrollo, lenguajes de programación y herramientas que faciliten la creación y garanticen la calidad del sistema.
3. Confección de los diagramas de clases del diseño con los diagramas de secuencia correspondientes.
4. Desarrollo de sistema informático para el observatorio científico, tecnológico y de innovación en el MINCEX.
5. Desarrollo de los casos de prueba que permitan validar la solución propuesta.

Para dar cumplimiento a las tareas de la investigación se utilizaron los siguientes **métodos científicos**:

Métodos Teóricos:

Analítico- Sintético: se utiliza para analizar, conceptualizar y extraer lo esencial de las características de los sistemas informáticos, lenguajes de programación y servidores de Base de Datos para almacenar información.

Modelación: se utiliza para crear el proceso de diseño y modelar los diagramas de negocio en el Visual Paradigm.

Histórico- Lógico: se utiliza para realizar un estudio del arte para investigar observatorios científicos que presentan características similares.

Métodos Empíricos:

Entrevista: se emplea en encuentros con los trabajadores del Mincex para conocer la necesidad del desarrollo de la propuesta de solución, a su vez definir sus funcionalidades y restricciones que se imponen.

Estructura Capítular.

La investigación está estructurada en 3 capítulos, a continuación, se muestra el resumen de cada uno de estos:

Capítulo1: “Fundamentos y referentes teórico - metodológicos sobre el objeto de estudio”: En este capítulo se exponen los enfoques teórico-metodológicos imprescindibles para la comprensión del término Observatorio y cada una de las posiciones y conceptos asociados a él. Se realiza un análisis de las herramientas a emplear para el diseño del Observatorio y se define el entorno de desarrollo a utilizar para darle solución al problema planteado en la investigación.

Capítulo2: “Diseño de la solución propuesta al problema científico”: En este capítulo se profundiza sobre las funcionalidades que debe tener el sistema. Se presentan los requisitos funcionales y no funcionales. Se muestran los casos de usos del sistema y su descripción, así como los actores que interactúan con el sistema. Además, describe el diseño de la solución propuesta, se brinda una descripción de los estilos y patrones arquitectónicos más utilizados y la aplicación de los mismos para la confección de los diagramas de clases y el modelo del diseño del sistema.

Capítulo3: “Validación de la solución propuesta”: En este capítulo se exponen los aspectos relacionados con el proceso de implementación del sistema. Se describe la solución a través del diagrama de componentes y los estándares de codificación, elementos que se tuvieron en cuenta para el producto final. Se documenta el proceso de obtención de la solución y una vez concluida la fase de implementación se realizan las pruebas funcionales, de seguridad y de rendimiento, que validan si el sistema cumple con los requerimientos establecidos.

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

En el presente capítulo se declara la base conceptual que sustenta la investigación, se realiza un estudio de los sistemas homólogos, las herramientas, tecnologías y metodologías con el objetivo de determinar las más factibles para el desarrollo de la posible solución.

1.1 Conceptos asociados a la investigación.

Para ayudar a entender el desarrollo de la investigación, se relacionan a continuación los conceptos principales que sirven de soporte.

Sistema de Gestión

Un sistema de gestión es el medio a través del cual una organización gestiona las partes interrelacionadas de su negocio para lograr sus objetivos. Estos objetivos pueden relacionarse con una serie de temas diferentes, incluyendo la calidad del producto o servicio, la eficiencia operativa, el desempeño ambiental, la salud y la seguridad en el lugar de trabajo y muchos más [5].

Gestión de Información

Probablemente la definición más citada sobre la Gestión de Información es la de (Linda Woodman, 1985), en la cual plantea que “La gestión de la información es todo lo relacionado con: la obtención de la información adecuada, en la forma correcta, para la persona u organización indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta” [6].

En términos perfectamente claros la Gestión de Información se entiende por tener la información adecuada, para la persona que lo necesita, en el momento que lo necesita, al mejor precio posible para toma la mejor de las decisiones [7].

Sistema de Gestión de Información (SGI)

Un SGI permite la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. El SGI aprovecha al máximo sus recursos de información en función de la mejora

continua y de la toma de decisiones de la organización a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa [8].

Sistema Informático [9]

El concepto más completo y preciso de Sistema Informático según la Organización Internacional de Normalización (ISO) es: Conjunto de elementos que hacen posible el tratamiento automático de la información.

Observatorio

El vocablo “Observatorio” se asocia a un lugar (edificio) o posición que sirve para hacer observaciones por medio de instrumentos apropiados y dedicados a observaciones comúnmente astronómicas o meteorológicas; pero hoy las empresas lo han asumido con gran popularidad como herramienta colectiva para potenciar la capacidad de detección de cambios tecnológicos, para así obtener y ofertar información relevante, basado en el empleo de técnicas y procedimientos de vigilancia tecnológica y dirigido a un colectivo bien definido de usuarios, producto de la gestión de información, orientado al apoyo de los procesos de innovación tecnológica de las organizaciones (empresariales, sociales y/o gubernamentales). Por tanto (Solórzano, 2007) a los observatorios los define como el producto de la unión de un grupo de personas con una amplia visión en materia de negocios, basada en la experiencia que han acumulado a lo largo del tiempo de trabajo en la empresa y grupos de expertos en la materia a tratar [10].

1.2 Estudios de sistemas homólogos.

A continuación, se realiza un estudio de sistemas homólogos encontrados de observatorios científicos, tecnológicos y de innovación.

1.2.1 Ámbito Internacional.

El primer observatorio de ciencia y tecnología en el mundo, se creó en 1990 y ese modelo de organización se ha desarrollado en varios países [11]. Algunos de estos observatorios pioneros han sido:

- **El Observatoire des Sciences et des Techniques [12] francés en el año 1990.**
El Observatorio de la Ciencia y la Tecnología fue fundado en 1990 y es un departamento de Hcéres desde 2015. Sus análisis informan el pensamiento estratégico de los actores de la educación superior y la investigación y contribuyen a la evaluación del impacto de las políticas públicas. Su trabajo incluye: la

producción periódica de indicadores y análisis sobre la investigación en ciencia y tecnología, los análisis de apoyo a las evaluaciones de Hcéres, la realización de proyectos específicos y la atención de pedidos externos [13].

- **El Observatorio Colombiano de Ciencia y Tecnología (OCyT) en 1999.**

El Observatorio de Ciencia y Tecnología fue constituido el 13 de agosto de 1999 e inscrito ante la Cámara de Comercio de Bogotá el 19 de octubre de 1999. Puede usar indistintamente el nombre abreviado de El Observatorio o la sigla OCyT. Es una asociación civil de participación mixta y de carácter privado, sin ánimo de lucro, con patrimonio propio organizada bajo las leyes colombianas dentro del marco de la Constitución Política y las normas de Ciencia y Tecnología y regida por ellas, en especial por las regulaciones previstas para las corporaciones en el Código Civil y por sus Estatutos [14].

- **El Observatorio Venezolano de Ciencia, Tecnología e Innovación (ONCTI) en 2006.**

El Observatorio Nacional de Ciencia, Tecnología e Innovación (ONCTI) es el organismo responsable de recopilar, sistematizar, categorizar, analizar e interpretar información a los fines de facilitar la formulación de las políticas públicas en ciencia, tecnología e innovación y sus aplicaciones. Fue creado el 7 de junio de 1990 según Decreto No. 34.386 y transformado el 23 de octubre de 2006 según Decreto No. 4.923 [15].

1.2.2 Ámbito Nacional.

También existen observatorios o centros de vigilancia tecnológica en Cuba, desarrollados por organismos, universidades y gobiernos locales, algunos de los más recientes son: el observatorio Demográfico de Cuba, el Observatorio de Gobierno Digital (OGD), el Observatorio Social y Laboral y el Observatorio Cubano de Ciencias Económicas. Todos impulsados por las nuevas políticas aprobadas en materia de ciencia y responden a la

necesidad creciente de monitorear constantemente el comportamiento del entorno con el propósito de apoyar los procesos de tomas de decisiones [16].

Observatorio Demográfico de Cuba se constituye en medio de la epidemia de la COVID-19, cuando salta a la vista la necesidad cada vez mayor de conocer las características sociodemográficas de la población [12].

Observatorio de Gobierno Digital (OGD), plataforma científica y colaborativa cuyo propósito es evaluar, socializar y acompañar el desarrollo de esta actividad en Cuba, realiza una encuesta publica de satisfacción ciudadana sobre servicios y tramites en línea [17].

Observatorio Social y Laboral, adscripto al Ministerio de Trabajo y Seguridad Social (MTSS), que constituye un instrumento para elevar la eficiencia del proceso de toma de decisiones a través de su participación en el diseño, gestión y evaluación de políticas sociales y laborales, así como para el cierre de brechas identificadas mediante la asesoría a la sociedad y sus organizaciones [3].

Observatorio Cubano de Ciencias Económicas, es la personalización y customización para la Asociación Nacional de Economistas y Contadores de Cuba (ANEC); tiene como objetivo socializar el conocimiento científico e innovaciones en el campo de las ciencias económicas, con impacto en las instancias gubernamentales cubanas, los actores económicos nacionales, los medios académicos, las instituciones científicas y otros públicos interesados en el país [3].

1.3 Metodologías de desarrollo de software.

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de informático. Suele estar documentada y promovida por algún tipo de organización ya sea esta pública o privada. Tienen como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la

calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto, exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se clasifican en dos clases: las metodologías tradicionales o robustas y las ágiles o ligeras [18].

Para el desarrollo de la solución se utilizará la metodología definida por el centro DroidLab para el desarrollo de proyectos. La misma consiste en una versión desarrollada por la UCI basada en la metodología de desarrollo de software AUP.

Metodología de desarrollo de software AUP versión UCI

La Universidad de las Ciencias Informáticas (UCI) desarrolló una versión de la metodología de desarrollo de software AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre [19].

La metodología de software AUP-UCI a partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos, como son: CUN (Casos de uso del negocio), DPN (Descripción de proceso de negocio) o MC (Modelo conceptual) y existen tres formas de encapsular los requisitos los cuales son: CUS (Casos de uso del sistema), HU (Historias de usuario), DRP (Descripción de requisitos por proceso), surgen cuatro escenarios para modelar el sistema en los proyectos, los cuales son [19]:

- 1 Escenario No 1:** Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- 2 Escenario No 2:** Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- 3 Escenario No 3:** Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

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

A partir del análisis e investigación realizada, se utilizó el escenario #4 con Historia de usuarios de la metodología AUP-UCI, debido a que dicha metodología es apropiada para proyectos pequeños permitiendo disminuir las probabilidades de fracaso, por ser un producto de fácil uso utilizando cualquier herramienta. Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. De igual manera, también es la metodología que más utilidad tiene en la UCI por ser una versión desarrollada en dicha institución.

1.4 Herramientas y tecnologías.

1.4.1 Lenguaje de Programación.

Un lenguaje de programación consiste en un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Estos suelen usarse para crear programas que controlen el comportamiento físico y lógico de una máquina y para expresar algoritmos con precisión. A continuación, se muestran los lenguajes de programación utilizados en el desarrollo de la propuesta de solución:

1.4.1.1 Lenguajes del lado del cliente

HyperText Markup Language 5 (HTML)

Lenguaje de publicación especificado como un estándar por el W3C (World Wide Web Consortium) que permite la creación de páginas web. Inicialmente fue presentado por Tim Berners-Lee que propuso un sistema basado en hipertexto para el intercambio de información en la Web. La aparición del lenguaje influyó notablemente en el crecimiento de Internet, dónde la información era distribuida mediante colecciones fragmentadas de textos, imágenes y sonidos. HTML es independiente de la plataforma utilizada y se basa fundamentalmente en el uso de etiquetas estructurales y semánticas, adecuadas para la creación de documentos relativamente simples que permiten simplificar su estructura [1].

Cascading Style Sheets 3 (CSS)

Con el crecimiento de Internet y la aparición del lenguaje HTML para la creación de páginas web, el W3C demostró la necesidad de un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos creados, de manera que pudieran visualizarse de igual forma en cualquier navegador web. A partir de entonces surgieron entre varias propuestas los lenguajes CHSS (Cascading HTML Style Sheets) y SSP (Stream-based Style Sheet Proposal), el primero realizado por Hakon Wium Lie y el segundo por Bert Bos, que a finales de 1994 y 1995 se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta denominado CSS (Cascading Style Sheets) [20].

CSS es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo utilizado para especificar el aspecto de una página web. Se basa en reglas que rigen el comportamiento del estilo de los elementos.

1.4.1.2 Lenguajes del lado del servidor

Hypertext Pre-Processor 7 (PHP)

Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web. PHP está enfocado principalmente a la programación de scripts del lado del servidor. Puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, macOS, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario de PHP FastCGI, como Lighttpd y Nginx. PHP funciona tanto como módulo como procesador de CGI [21].

Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos (p.ej., para MySQL) [21].

1.4.2 Servidor Web

Fue escogido Apache en su version 2.4 ya que es un servidor web de código abierto, multiplataforma y gratuito. Este web server es uno de los más utilizados en el mundo, actualmente el 43% de los sitios webs funcionan con él. Este servidor web desarrollado por Apache Software Foundation lleva en funcionamiento desde 1995.

La función esencial del servidor Apache es servir las webs alojadas en el servidor a los diversos navegadores como Chrome, Firefox, Safari. Apache consigue que la comunicación entre el servidor web y el cliente web (usuario que solicita la información) sea fluida y constante. Haciendo que cuando un usuario haga una petición HTTP a través de navegador para entrar a una web o URL específica, Apache devuelva la información solicitada a través del protocolo HTTP [22].

1.4.3 Sistema Gestor de Base de Datos MySQL

MySQL

Sistema gestor de base de datos relacional, disponible para múltiples plataformas. MySQL posee alta velocidad al realizar las operaciones, lo que le hace que posea un buen rendimiento. Es considerado como de mediana escala y es adecuado para aplicaciones web de tamaño medio. MySQL permite que varios usuarios a través de múltiples hilos puedan acceder a datos relacionales y soporta múltiples herramientas para la gestión [23].

1.4.4 Sistema Gestor de Contenidos

Drupal 8

Es un software de gestión de contenidos. Se usa para hacer muchos de los sitios web y aplicaciones que usa todos los días. Drupal tiene excelentes funciones estándar, como la creación de contenido fácil, un rendimiento confiable y una seguridad excelente. Pero lo que lo distingue es su flexibilidad; la modularidad es uno de sus principios fundamentales. Sus herramientas lo ayudan a crear el contenido versátil y estructurado que necesitan las experiencias web dinámicas [24].

Drupal 9 lleva adelante la filosofía de innovación continua que comenzó en Drupal 8. Es la actualización más fácil en más de una década de historia de Drupal y es más fácil de usar que nunca.

1.4.5 Motor de plantillas

Twig

Es un motor de plantillas desarrollado para el lenguaje de programación PHP y que nace con el objetivo de facilitar a los desarrolladores de aplicaciones web que utilizan la arquitectura MVC (Modelo-Vista-Controlador) el trabajo con la parte de las vistas, gracias a que se trata de un sistema que resulta muy sencillo de aprender y capaz de generar plantillas con un código preciso y fácil de leer [\[25\]](#).

Actualmente el código se distribuye bajo licencia BSD y es utilizado por el framework Symfony 2, aunque puede ser utilizado directamente con proyectos desarrollados en PHP en el que no interviene ese framework.

1.4.6 Lenguaje Unificado de Modelado

Visual Paradigm 8.0

Es una herramienta CASE multiplataforma que cuenta con versiones gratuitas y provee fácil integración con el resto de las herramientas de desarrollo. Ayuda a una más rápida construcción de aplicaciones de calidad y con menor costo. También permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación [\[26\]](#).

Se decidió utilizar esta herramienta en el desarrollo de la aplicación por las múltiples ventajas que posee al brindarle soporte a todo el ciclo de vida de implementación del software a desarrollar.

Conclusiones del capítulo

Durante este capítulo se trataron los elementos teóricos que sustentan a la propuesta de solución del problema planteado, en este sentido se concluye:

La sistematización de los conceptos fundamentales asociados al dominio de la presente investigación y sus relaciones, proporcionó una mayor comprensión de la propuesta de solución.

El análisis de los sistemas homólogos existentes, permitió llegar a la conclusión que no se ajustan completamente a las necesidades de la propuesta de solución, pero si aportó una visión más amplia del objeto de estudio de la investigación. Dada estas condiciones se evidenció la necesidad de incorporarle a la propuesta de solución ciertos parámetros que no se encontraron en los sistemas homólogos, como son: cambio de idioma (inglés, español); glosario con palabras y expresiones de mayor empleo en el proceso y la operatoria de comercio exterior, inversión extranjera y cooperación internacional mostrándose en tres idiomas (español, inglés, francés) y fórum topic con temas de interés para los visitantes.

El estudio de las tecnologías adecuadas para el desarrollo de observatorios científicos determinó que para el desarrollo de la investigación se debe emplear la metodología de desarrollo AUP en su variante UCI, los lenguajes de programación CSS, HTML, Twig y PHP, el gestor de bases de datos MySQL y como herramienta de modelado Visual Paradigm. El estudio de los sistemas de gestión de contenidos más utilizados en la actualidad arrojó que para el desarrollo del observatorio científico se debe utilizar Drupal.

CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

En este capítulo se realiza una breve reseña de las características del sistema, especificando además los requisitos funcionales y no funcionales con los que cuenta el observatorio científico. Haciendo uso del lenguaje de modelado UML y la herramienta de Visual Paradigm se elaboran diagramas que ayudan a la comprensión de la estructura del sistema.

2.1 Propuesta del sistema

Con la finalidad de darle solución al problema planteado en esta investigación, se propone la creación de un Observatorio Científico con la misión de monitorear, analizar, generar y difundir información útil sobre comercio exterior, inversión extranjera y cooperación internacional desde cualquier dispositivo electrónico de cara a internet. Dicho observatorio posee secciones que dan paso a la divulgación de artículos, eventos, noticias y documentos relacionados al ministerio. Se brindan servicios como las alertas, estudios donde se propone líneas de investigación y proyectos en desarrollo, y el usuario puede elegir entre los idiomas español e inglés cuando accede al mismo.

2.1.1 Modelo del Dominio

El modelo del dominio representa las clases y relaciones más significativas a tener en cuenta en el desarrollo del sistema, se definen las identidades, atributos y relaciones existentes entre los objetos, representados mediante clases. Este modelado facilita la comprensión del problema ya que describe y limita el alcance del dominio [27].

A continuación, se muestra el diagrama del dominio realizado por la autora utilizando la herramienta Visual Paradigm.

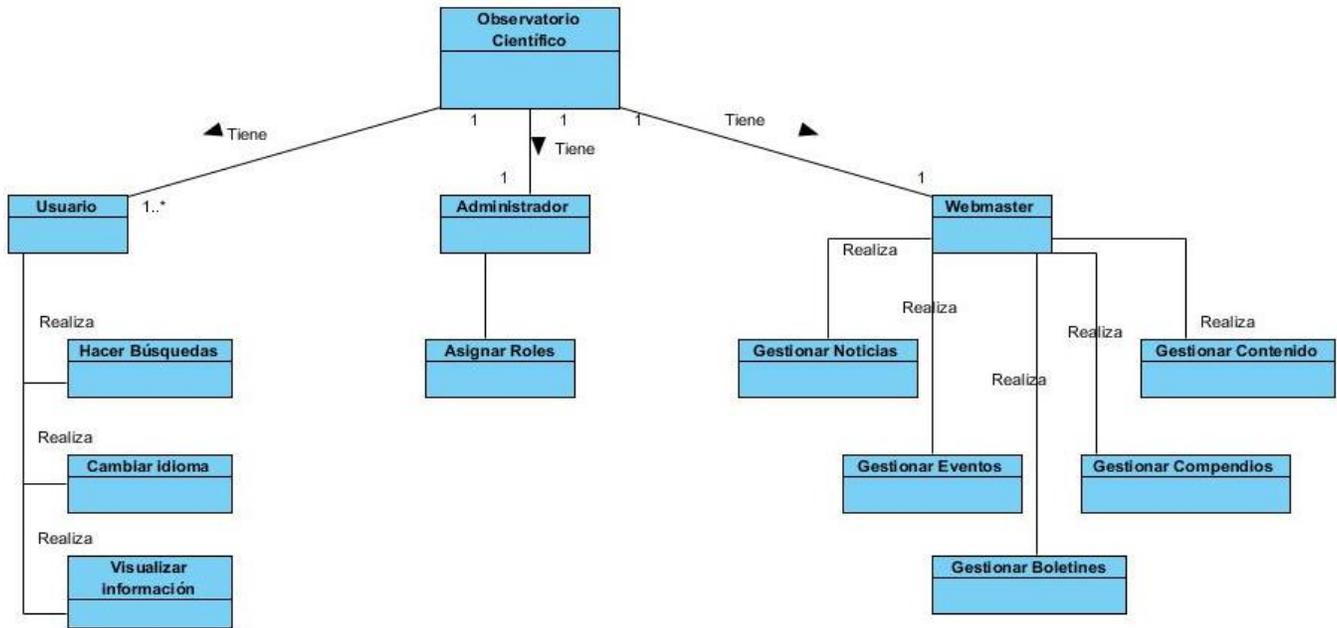


Figura 1. Modelo de dominio

Descripción de los conceptos del Modelo del Dominio

Observatorio Científico: Representa un conjunto de contenidos que se pueden consultar.

Usuario: Persona que interactúa con el contenido que puede visualizar noticias, eventos, etc.

Administrador: Encargado de gestionar los roles y crear, modificar, eliminar y visualizar los contenidos.

Webmaster: Encargado de seleccionar, recopilar, organizar y divulgar la información en materia de comercio exterior, inversión extranjera y cooperación internacional.

2.2 Especificación de Requisitos

Como paso fundamental en el proceso de desarrollo de la solución propuesta se realiza la especificación de requisitos. Los requisitos de software son las necesidades de los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar [28].

Requisitos Funcionales

Los requisitos funcionales son condiciones o capacidades con los que debe cumplir el sistema, son acuerdos entre el cliente y los desarrolladores sobre lo que debe o no debe hacer el sistema [29].

Tabla 1: Requisitos Funcionales

No	Nombre	Complejidad
RF1	Autenticar usuario	Baja
RF2	Crear usuario	Alta
RF3	Editar usuario	Media
RF4	Mostrar usuario	Baja
RF5	Mostrar listado de usuarios	Baja
RF6	Eliminar usuario	Media
RF7	Crear evento	Alta
RF8	Editar evento	Media
RF9	Eliminar evento	Media
RF10	Mostrar evento	Baja
RF11	Crear alerta	Alta
RF12	Editar alerta	Media
RF13	Eliminar alerta	Media
RF14	Mostrar alerta	Baja
RF15	Crear información sobre ¿Quiénes somos?	Alta
RF16	Editar información sobre ¿Quiénes somos?	Media
RF17	Mostrar información sobre ¿Quiénes somos?	Baja
RF18	Eliminar información sobre ¿Quiénes somos?	Media
RF19	Crear artículo	Alta
RF20	Editar artículo	Media
RF21	Mostrar artículo	Baja
RF22	Editar página básica	Media
RF23	Mostrar página básica	Media
RF24	Crear boletín	Alta
RF25	Eliminar boletín	Media

RF26	Editar boletín	Media
RF27	Mostrar boletines	Baja
RF28	Crear información sobre Investigación documental	Alta
RF29	Editar información sobre Investigación documental	Media
RF30	Eliminar información sobre Investigación documental	Media
RF31	Mostrar información sobre Investigación documental	Baja
RF32	Crear información sobre Informes temáticos	Media
RF33	Editar información sobre Informes temáticos	Media
RF34	Eliminar información sobre Informes temáticos	Media
RF35	Mostrar información sobre Informes temáticos	Baja
RF36	Crear traducción de contenido	Alta
RF37	Editar traducción de contenido	Media
RF38	Mostrar traducción de contenido	Baja
RF39	Eliminar traducción de contenido	Media
RF40	Crear enlace de Revistas	Media
RF41	Editar enlace de Revistas	Media
RF42	Crear información sobre Tesis	Media
RF43	Editar información sobre Tesis	Media
RF44	Eliminar información sobre Tesis	Media
RF45	Mostrar información sobre Tesis	Baja
RF46	Crear término	Alta
RF47	Editar término	Media
RF48	Eliminar término	Media
RF49	Mostrar término	Baja
RF50	Crear información de interés	Alta
RF51	Editar información de interés	Media
RF52	Eliminar información de interés	Media
RF53	Mostrar información de interés	Baja
RF54	Realizar búsqueda	Baja
RF55	Insertar servicio	Baja
RF56	Editar servicio	Baja
RF57	Eliminar servicio	Baja
RF58	Mostrar servicio	Baja
RF59	Cambiar de idioma	Alta
RF60	Crear compendio	Alta

RF61	Editar compendio	Media
RF62	Mostrar compendio	Baja
RF63	Visualizar compendio	Baja
RF64	Crear traducción de contenido	Alta
RF65	Editar traducción de contenido	Media
RF66	Mostrar traducción de contenido	Baja
RF67	Eliminar traducción de contenido	Media
RF68	Crear Fórum topic	Alta
RF69	Editar Fórum topic	Media
RF70	Eliminar Fórum topic	Media
RF71	Mostrar Fórum topic	Baja
RF72	Crear Nota Informativa	Alta
RF73	Editar Nota Informativa	Media
RF74	Eliminar Nota Informativa	Media
RF75	Mostrar Nota Informativa	Baja
RF76	Mostrar datos de contacto	Media
RF77	Crear noticia	Alta
RF78	Editar noticia	Media
RF79	Eliminar noticia	Media
RF80	Mostrar noticia	Baja

Requisitos No Funcionales

Los requisitos no funcionales, como su nombre lo sugieren, son aquellos que no se refieren a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento [30].

Los requerimientos no funcionales definen las características o cualidades generales que se esperan de un sistema y establecen restricciones sobre el producto, el proceso de desarrollo de software y establecen restricciones externas que el software debe lograr [30].

Usabilidad

RNF1: Diseño limpio y claro que facilite la navegación.

RNF2: Se debe lograr una estructura simple y lo más intuitiva posible de la arquitectura de información.

Seguridad

RNF3: Ante los errores que puedan ocasionarse en el sistema no se deben mostrar detalles de información que puedan comprometer su seguridad e integridad.

RNF4: La información manejada por el sistema estará protegida de accesos no autorizados.

Software

RNF5: Servidor web Apache en su version 2.4.7.

RNF6: Lenguaje de programación PHP 7.3 y como sistema de gestión de contenidos Drupal 8.

RNF7: Servidor de Base de dato MySQL en su version

Disponibilidad

RNF8: Se debe garantizar la disponibilidad de la información en el sistema durante todo el día.

Interfaz

RNF9: La aplicación debe ser visible en navegadores como Internet Explorer 9 o superior, Mozilla Firefox 103.0.2 o superior y Chrome v.45.

Hardware

RNF10: El servidor de base de datos debe poseer una capacidad mínima de 20 GB.

RNF11: Los servidores web y de base de datos deben poseer como mínimo 1 GB de memoria RAM.

Eficiencia

RNF12: El sistema debe permitir que 50 usuarios interactúen con él de manera concurrente.

RNF13: El tiempo de demora del sistema en cada transición debe ser menor de (5) segundos aproximadamente.

2.2.1 Historia de usuario

Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal. Describen, en una o dos frases, una funcionalidad de software desde el punto de vista del usuario, con el lenguaje que éste emplearía. El foco está puesto en qué necesidades o problemas soluciona lo que se va a construir [31].

La prioridad del negocio [31]:

- Alta: cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
- Media: cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
- Baja: cuando constituyen procesos que se deben tener en cuenta, pero su ausencia no perjudica el flujo principal del negocio.

Tabla 2: Historia de Usuario Insertar Noticia

Historia de Usuario	
Número: RF77	Nombre: Crear Noticia
Usuario: CSCEIE	
Programador: Yarianna Montesino Martinez	Iteración asignada: Primera iteración
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Bajo	Tiempo Real: 5 días

Descripción: El CSCEIE podrá crear noticias en el sistema. De una noticia se registra: el título, lenguaje, imagen y una descripción.

Observaciones:

Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente la noticia.

Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.

Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.

Prototipo de Interfaz:

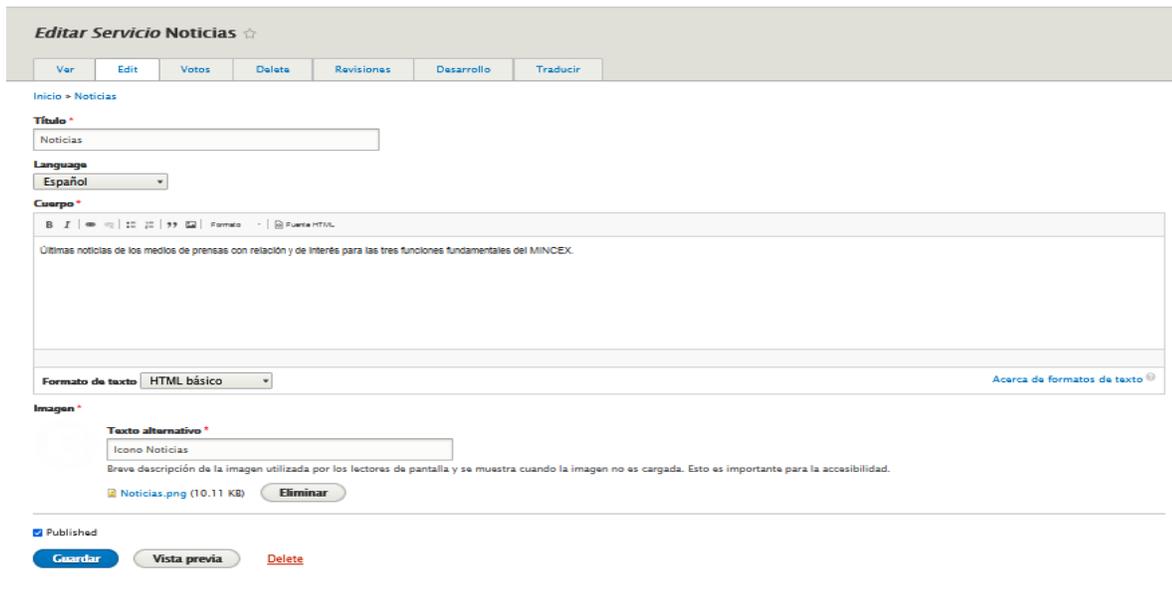


Tabla 3: Historia de Usuario Insertar Evento

Historia de Usuario	
Número: RF7	Nombre: Crear Evento
Usuario: CSCEIE	
Programador: Yarianna Montesino Martinez	Iteración asignada: Primera iteración
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Bajo	Tiempo Real: 5 días

Descripción: El CSCEIE podrá crear eventos en el sistema. De un evento se registra: el nombre del evento, una descripción del evento, el URL y el logo que lo identificará.

Observaciones:

Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el evento.

Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.

Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.

Prototipo de Interfaz:

The screenshot shows a web interface for creating an event. At the top, there is a header 'Crear Eventos' with a star icon. Below it, a breadcrumb trail reads 'Inicio > Añadir contenido'. The form is divided into several sections:

- Título ***: A text input field.
- Cuerpo (Editar resumen) ***: A rich text editor with a toolbar containing icons for bold, italic, underline, link, unlink, list, and image. Below the editor is a 'Formato de texto' dropdown menu set to 'HTML básico' and a link 'Acerca de formatos de texto'.
- Imagen ***: A file upload section with an 'Examinar...' button. Below the button, it says 'No se ha seleccionado ningún archivo.' and provides instructions: 'Máximo 1 fichero. Límite de 3 MB. Tipos permitidos: png webp jpeg jpeg.'
- ENLACE ***: A section with a 'URL' input field and a 'Texto del enlace' input field. Below the URL field, there is a small text block: 'Start typing the title of a piece of content to select it. You can also enter an internal path such as `/node/add/` or an external URL such as `http://example.com`. Enter `<front>` to link to the front page. Enter `<nofollow>` to display link text only. Enter `<button>` to display keyboard-accessible link text only.'

At the bottom of the form, there is a 'Published' checkbox which is checked, and two buttons: 'Guardar' and 'Vista previa'.

2.3 Diseño Arquitectónico

Para la creación del observatorio se utiliza el diseño de arquitectura de cinco capas debido que es la utilizada por el CMS Drupal. El diseño en capas permite una navegación fácil y diferencia los contenidos en dependencia de la autorización que posean los usuarios y sus necesidades. Seguidamente se muestra una imagen que representa esta arquitectura y además se explica cada una de estas capas [32].

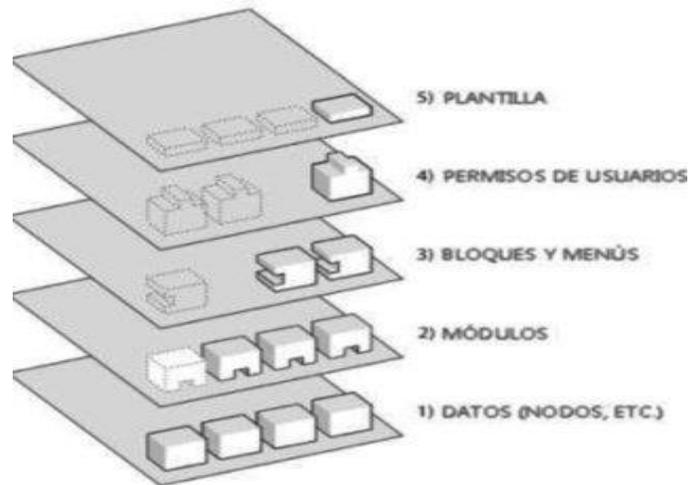


Figura 2

Datos: En esta capa se encuentran los datos que se generan al crear un contenido, por ejemplo: noticias, y demás elementos básicos de Drupal. También se encuentra la base de datos, que tienen como nombre pw_observatorio donde se almacena cada dato que el propio sistema genera con cada acción que necesite de datos.

Módulos: Los módulos son funcionalidades que están integradas al núcleo de Drupal o que son creados por terceros dentro de la comunidad programadores de este CMS. Cada módulo tiene una funcionalidad concreta para la que fue creada ya sea añadir más campos a un contenido, generar contenido aleatorio e incluso un módulo puede depender de otro para enriquecer las funcionalidades de este, como sucede con el módulo de comercio electrónico que posee muchos más módulos que se basan en él. Existen miles de módulos dentro del repositorio de Drupal que contribuyen a mejorar el propio sistema, ya sea creados por los desarrolladores del proyecto Drupal, su comunidad o grandes empresas.

Bloques y Menús: Los bloques a menudo son el resultado de un módulo en específico como las vistas o pueden crearse y luego adaptarlos al sistema. Pueden proporcionar un espacio de información concreta como de contacto o un listado corto y ordenado de algún tipo de contenido. Se utilizaron varios bloques que se situaron en la página inicial del sistema tales como el bloque de cambio de idioma, el bloque de la marca donde se muestra el logo del sistema, bloques de contenidos organizados en el sistema en dependencia de su contenido. Los menús son navegadores que enlazan cada página a los contenidos más importantes

dentro del sistema y a todas las demás páginas de Drupal mediante una URL predefinida.

Permisos de usuario: La mayoría de los sistemas web son multiusuario, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta funcionalidad Drupal dispone de un registro de usuarios y de roles que permiten especificar que tareas pueden realizar y a que contenidos puede acceder cada tipo de usuario (ejemplos: roles de administrador, organizador y cliente).

Plantillas: Esta es la capa superior y por tanto es la capa de personalización del sistema, también llamado tema dentro del sistema. Esta capa está integrada por plantillas TWIG compuestas además de HTML y el uso CSS. Con dichas plantillas se define la organización que tendrá cada componente del sistema desde lo más grande hasta un simple nodo de información, tanto en su distribución dentro de la estructura como su personalización en cuanto a tamaños y colores.

Diagrama de clases del diseño web

Un diagrama de clases del diseño con estereotipos web tiene el mismo objetivo o propósito que un diagrama de clases tradicional, con la particularidad de que se emplea para el modelado de aplicaciones web [33]. El mismo es evidenciado en la siguiente figura:

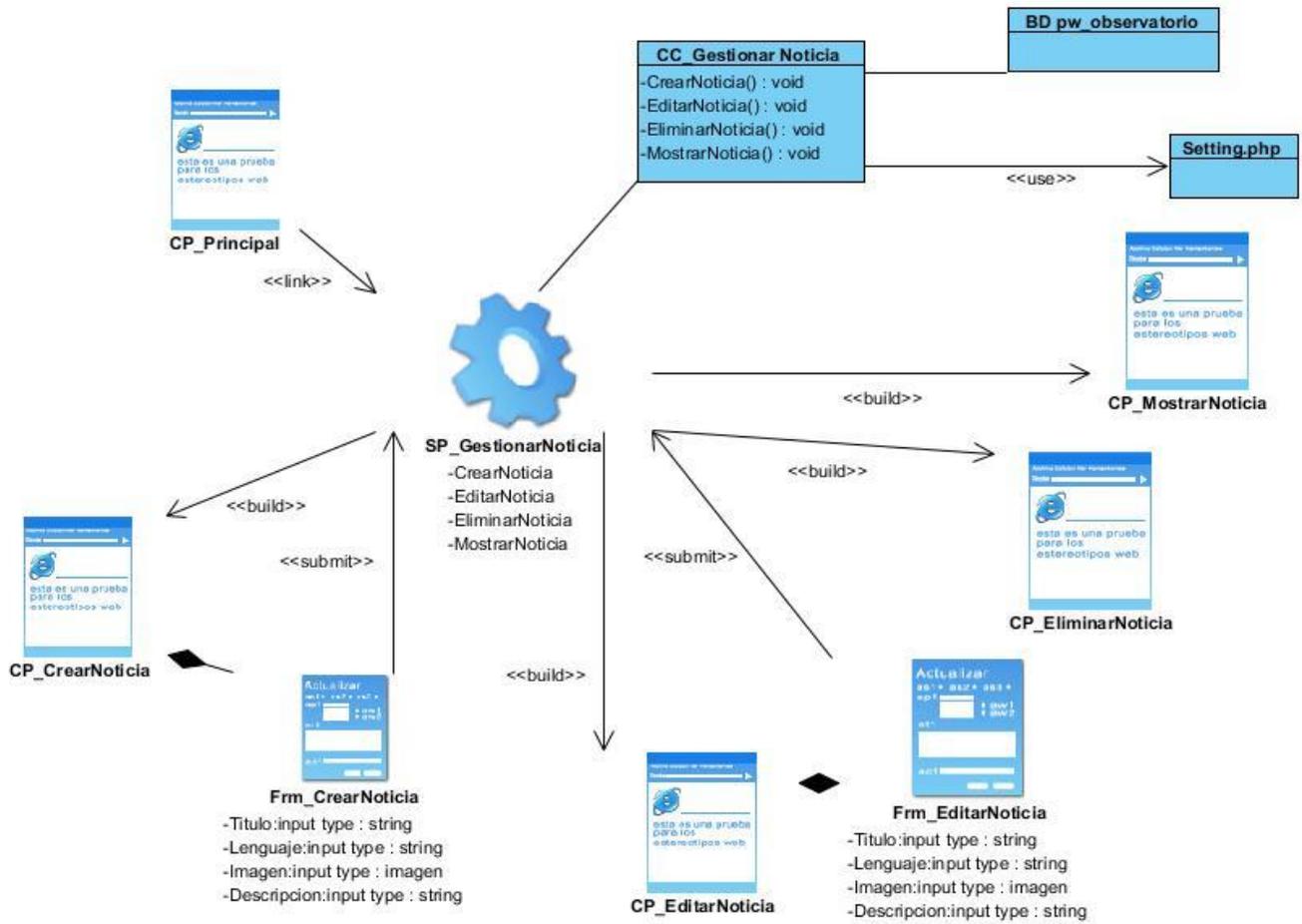


Figura 3: Diagrama de clase del diseño Gestionar Noticia

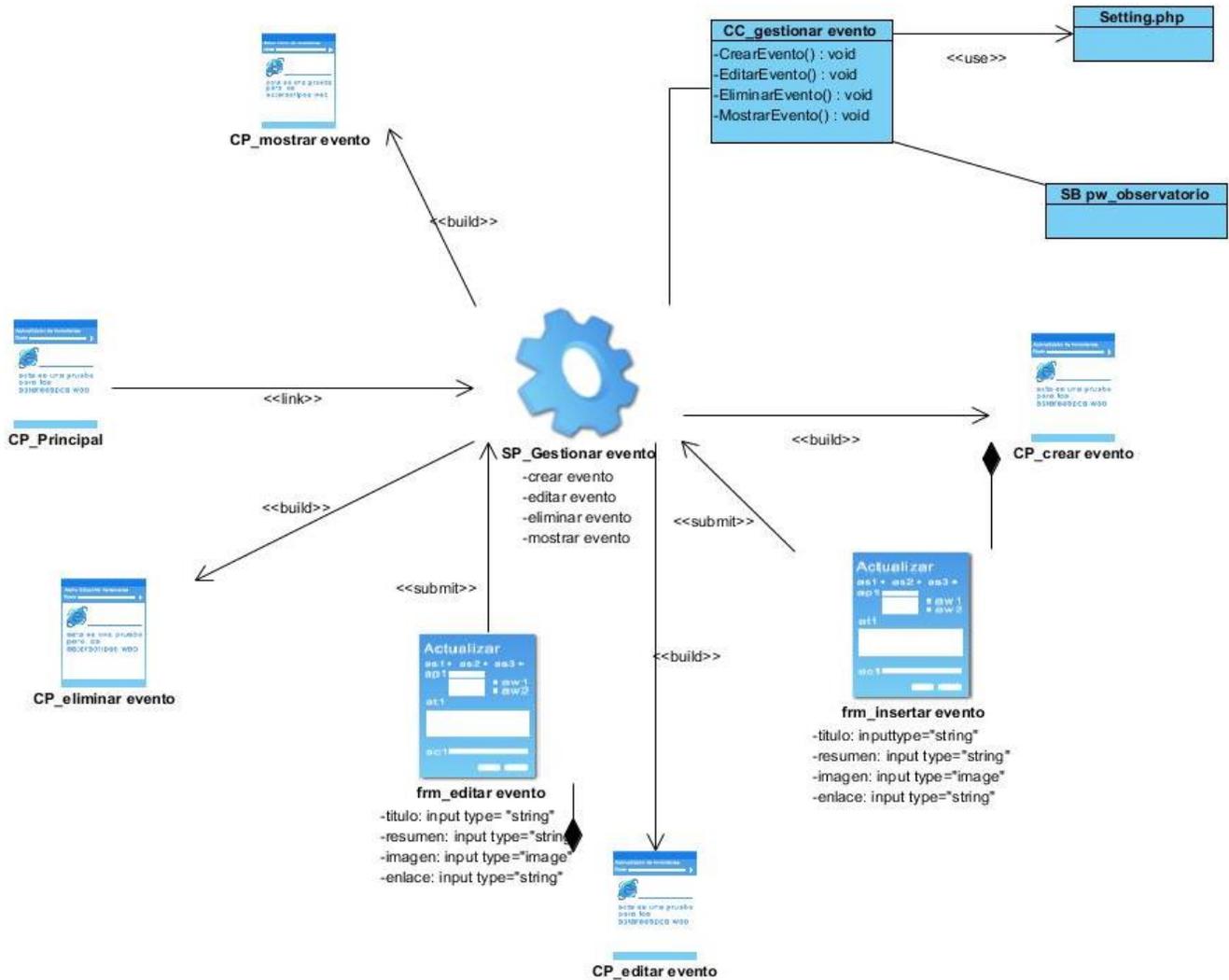


Figura 4: Diagrama de clase del diseño Gestionar evento

Patrones de diseño

Un patrón de diseño provee un esquema para refinar los componentes de un sistema de software, o las relaciones entre ellos. Estos brindan soluciones a los problemas que pueda haber en el diseño de un software [34].

Los patrones Singleton Pattern (Patrón de Instancia Única), Decorator Pattern (Patrón Decorador), Observer Pattern (Patrón de Observador), Bridge Pattern (Patrón de Puente), Chain of Responsibility Pattern (Patrón de Cadena de Responsabilidad) y Command Pattern (Patrón

de Comando) pueden verse empleados en Drupal en varias de sus características y funcionalidades.

- **Singleton Pattern:** Garantiza que exista una única instancia de una clase. Si los módulos y los temas en Drupal se piensan como objetos, este patrón está presente, debido a que en general estos objetos no encapsulan datos, lo que separa a un módulo de otro es el conjunto de funciones que contiene. Esto garantiza la presencia de una única instancia y el mecanismo de acceso global a ella [\[35\]](#).
- **Decorator Pattern:** Asigna responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Permite no tener que crear clases que hereden de la primera, incorporando nuevas funcionalidades, sino otras que la implementan y se asocian a ella [\[36\]](#).
En Drupal el uso de la API de nodo, permite a los módulos extender arbitrariamente el comportamiento de todos los nodos. Para los archivos cargados y adjuntados a un nodo, se podría diseñar un nuevo tipo de nodo con las demás características del nodo, que brinde la posibilidad de cargar archivos con la concesión de que cada nodo solicite la posibilidad de tener archivos adjuntos. Este comportamiento podría ser imitado por el uso del patrón decorador envolviéndolo alrededor de cada objeto nodo
- **Observer Pattern:** Este patrón de Observador es generalizado en todo Drupal, ya que muchos de los ganchos de Drupal esencialmente permiten que los módulos se registren como observadores de los objetos de Drupal. Por ejemplo, cuando se realiza una modificación a un vocabulario en el sistema de taxonomía de Drupal, un enlace de taxonomía como `hook_taxonomy_vocabulary_update ()` se llama en todos los módulos que lo implementan. Al implementar el gancho, los módulos se han registrado como observadores del objeto de vocabulario; cualquier cambio en él se puede actuar según corresponda [\[37\]](#).
- **Bridge Pattern:** La capa de abstracción de la base de datos Drupal se implementa de manera similar al patrón de diseño de Bridge. Los módulos deben escribirse de forma independiente del sistema de base de datos que se utiliza, y la capa de abstracción lo proporciona. Se pueden escribir nuevas capas de base de datos que se ajusten a la API definida por el puente, agregando soporte para sistemas de bases de datos

adicionales sin la necesidad de modificar el código del módulo. Esto permite separar los módulos bases existentes en el capa núcleo de Drupal de los módulos auxiliares que son añadidos de forma manual [37].

- **Chain of Responsibility Pattern:** El sistema de menús de Drupal sigue el patrón Cadena de responsabilidades. En cada petición de una página, el sistema de menús de Drupal determina si hay algún módulo para responder la petición, si el usuario tiene acceso al recurso solicitado y qué función se debe llamar para procesar la petición. En este proceso se trasmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos. De esta forma se continúa la cadena hasta que un módulo atiende la petición, un módulo deniegue el acceso o la cadena se agote [37].
- **Command Pattern:** Muchos de los ganchos de Drupal usan el patrón de Comando para reducir la cantidad de funciones que son necesarias para implementar, pasando la operación como un parámetro junto con los argumentos. De hecho, el sistema de gancho en sí usa este patrón, de modo que los módulos no tienen que definir cada gancho, sino solo los que les interesa implementar [37].

Diagrama de secuencia

El diagrama de secuencias es un esquema conceptual que permite representar el comportamiento de un sistema, para lo cual emplea la especificación de los objetos que se encuentran en un escenario y la secuencia de mensajes intercambiados entre ellos, con el fin de llevar a cabo una transacción del sistema. [38]

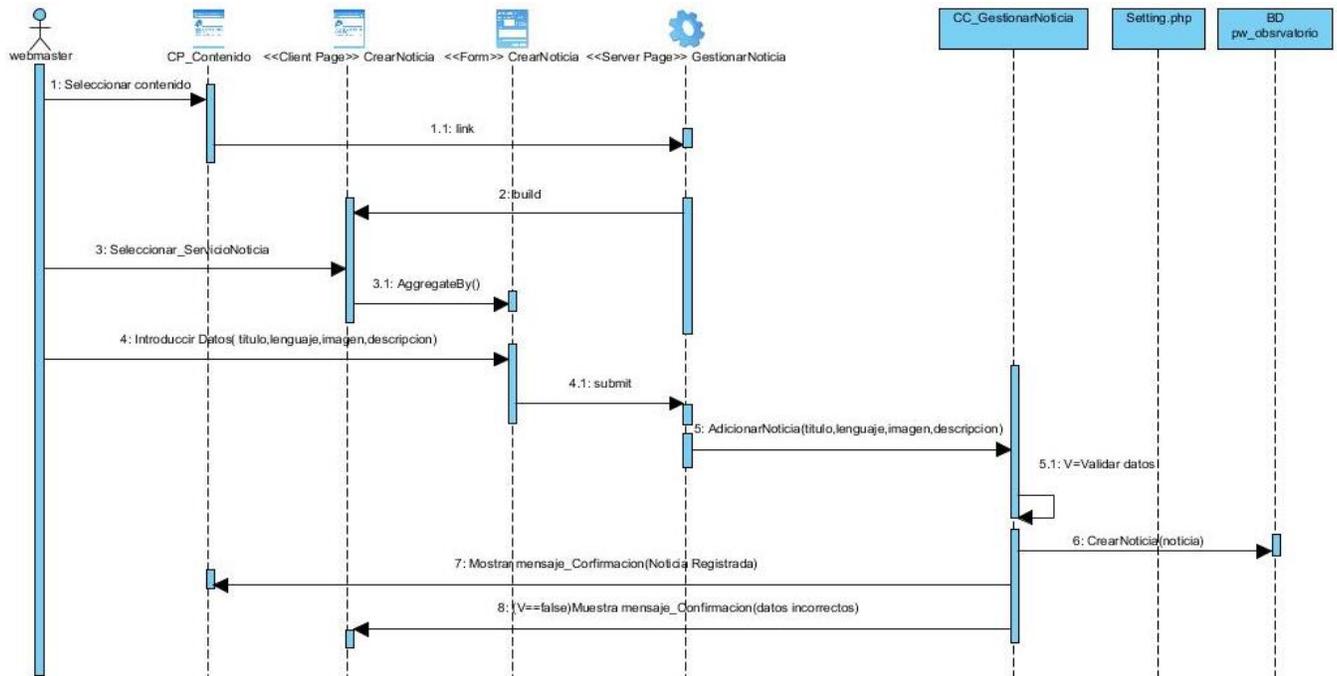


Figura 5: Diagrama de secuencia Insertar Noticia

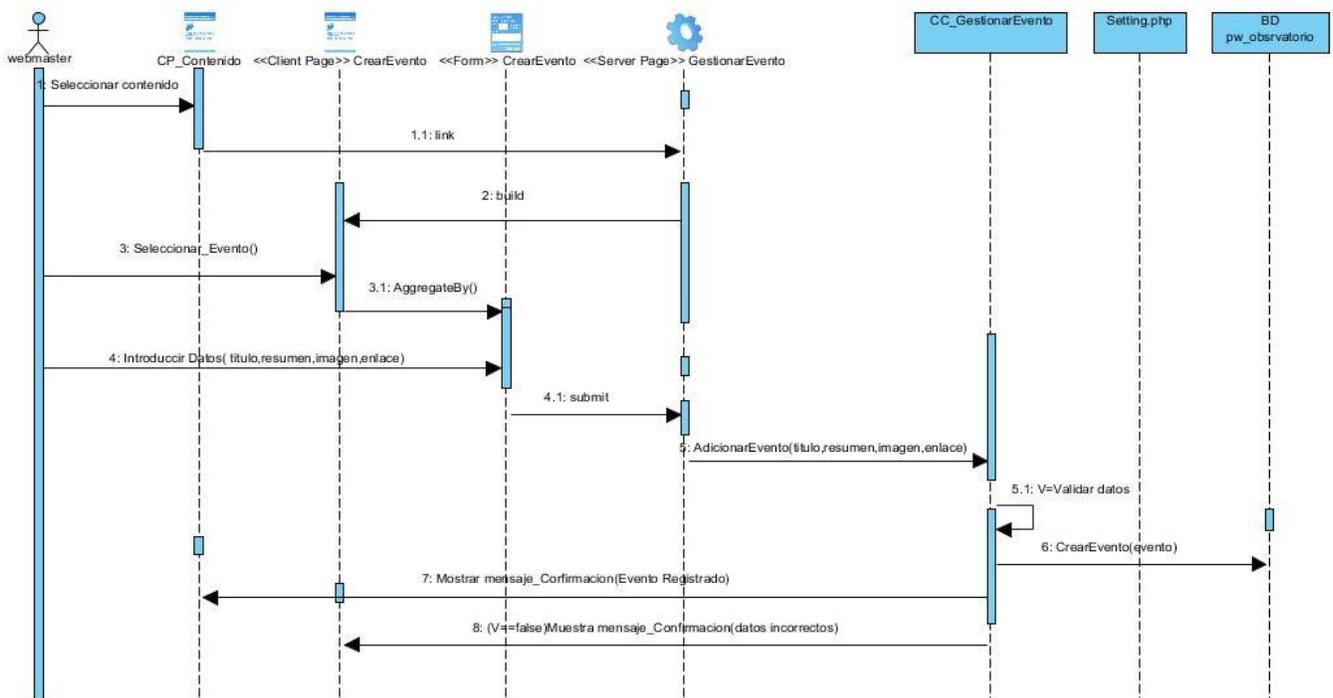


Figura 6: Diagrama de secuencia Insertar Evento

2.4 Diseño de despliegue

Representa de forma visual las relaciones físicas que existen entre los componentes de software y hardware en el sistema. Los nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño [39]. A continuación, se presenta al modelo de despliegue correspondiente a la propuesta de solución.

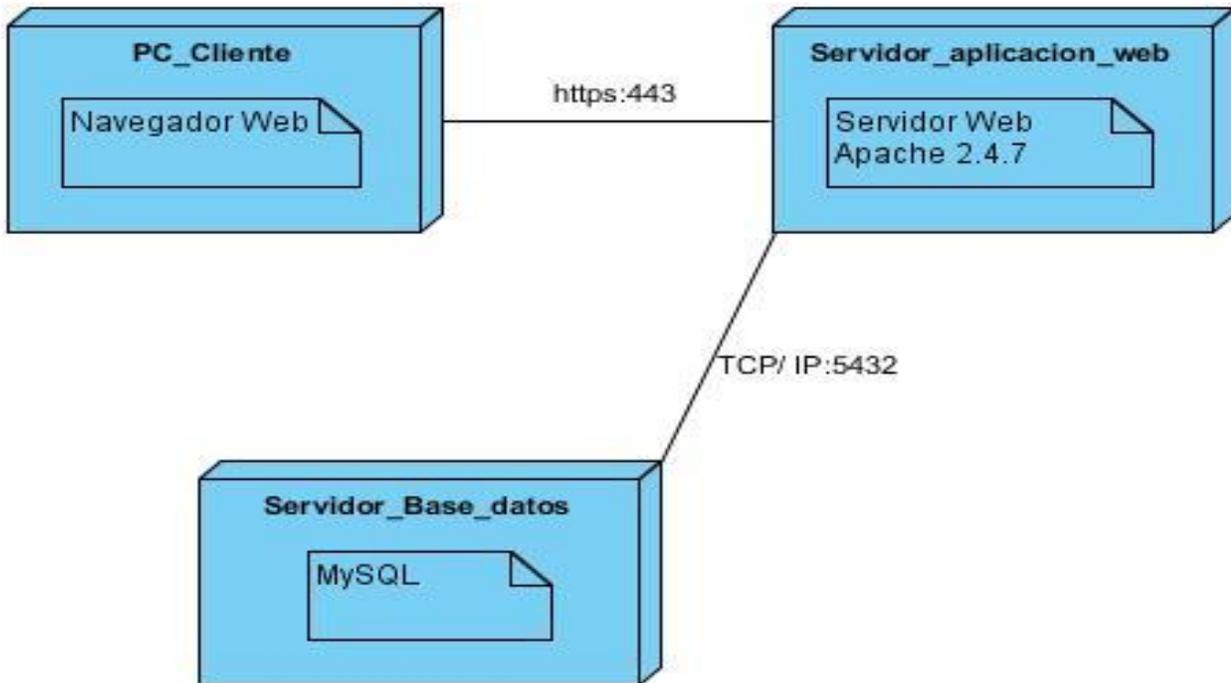


Figura 7: Diagrama de despliegue

Descripción de elementos e interfaces de comunicación:

PC_Cliente: La estación de trabajo necesita un navegador web para conectarse al sistema hospedado en el servidor de aplicaciones utilizando el protocolo de comunicación HTTP/HTTPS.

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

Servidor de BD: Este servidor es el encargado del almacenamiento de los datos del sistema. Se comunica con el servidor de aplicaciones del sistema, posibilitando el acceso mediante el usuario con privilegios para las operaciones determinadas a realizarse en el mismo.

HTTPS: Protocolo para establecer a través del puerto 443 la conexión segura entre el dispositivo de acceso cliente y el servidor de aplicaciones. La conexión es por cable vía modem, LAN o red inalámbrica con una velocidad de más de 64 Kbps.

TCP: Protocolo para establecer la conexión entre el servidor de aplicaciones y el servidor de base de datos. Para el servidor de base de datos de MySQL se define el puerto 3306. La conexión entre el servidor web y el servidor de base de datos permite dar órdenes y obtener información de esta.

Conclusiones del capítulo

Lo requisitos funcionales y no funcionales identificados a partir de proceso de especificación de requisitos permitieron desarrollar las distintas funcionalidades y las capacidades que debe cumplir el sistema para solucionar las necesidades detectadas.

La identificación de los patrones de diseño y el estudio arquitectónico garantizaron disminuir el impacto de los cambios futuros en el código fuente del sistema.

Los diagramas y modelos desarrollados permitieron un mayor entendimiento del sistema a implementar mostrando las relaciones existentes entre las distintas clases, las funcionalidades presentes en la misma y la organización temporal de los mensajes.

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

El presente capítulo presenta los componentes y estándares de codificación que sustentan la implementación del sistema y se describe el proceso de validación de la solución implementada, mediante la utilización de los casos de pruebas.

3.1 Diagrama de Componentes

El diagrama de componentes muestra los componentes de un sistema de software conectados por las relaciones de dependencias lógicas entre cada uno de ellos. Provee una vista arquitectónica de alto nivel del sistema, ayudando a los desarrolladores a visualizar el camino de la implementación. Cada componente representa una unidad de código (fuente, binario o ejecutable), que permita mostrar las dependencias en tiempo de compilación y ejecución. La realización del diagrama posibilita tomar decisiones respecto a las tareas de implementación y los requisitos[9].

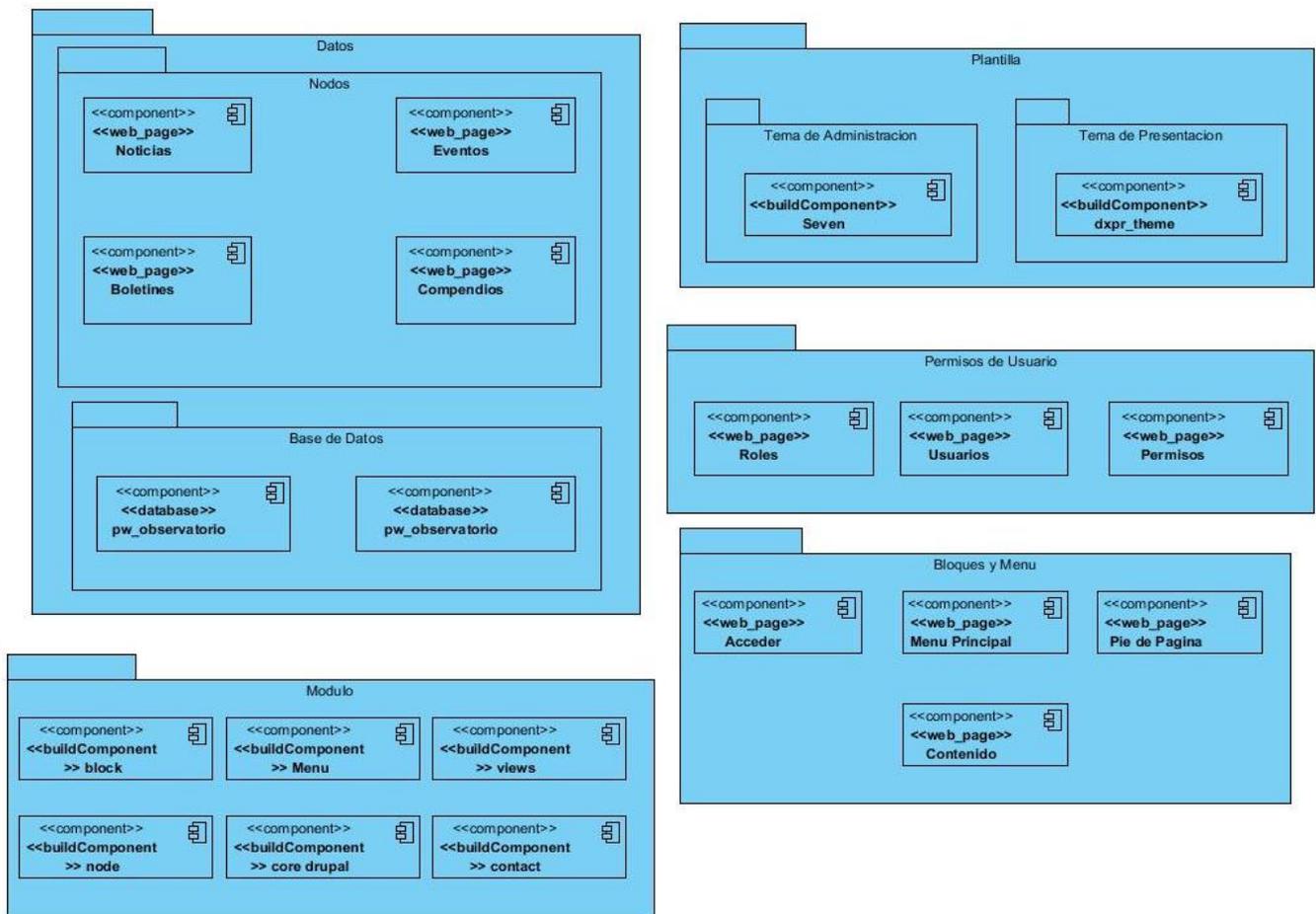


Figura 8: Diagrama de Componente

3.2 Estándar de codificación

Los estándares de codificación constituyen un principio esencial en el desarrollo de *software*. Garantizan que el código obtenido sea fácil de leer, entendido y modificado independientemente de quien haya sido el desarrollador del producto. Son una guía para el equipo de desarrollo, permiten asegurar que el código presente calidad y no contenga errores. Drupal proporciona a sus desarrolladores un conjunto de normas para fomentar el código de una forma uniforme para todos[40]. A continuación, se detallan los estándares de codificación utilizados en la implementación de la solución propuesta.

Tabla 4: Estándares de codificación utilizados

Tipo de Estándar	Descripción
Identación	Usar una sangría de 2 espacios, sin etiquetas. Las líneas no deben tener espacios en blanco al final. Los archivos deben tener el formato \n como final de línea (finales de línea de Unix), no \r\n (finales de línea de Windows). Todos los archivos de texto deben terminar en un solo salto de línea (\n).[40]
Etiquetas de apertura y cierre PHP	Al usar el lenguaje PHP se debe usar las etiquetas <code><?PHP y ?></code> para delimitar el código de dicho lenguaje y no las versión corta <code><? y ?></code> . De forma general se omite la etiqueta de cierre de PHP al final de los archivos. <code>module</code> y <code>.inc</code> . Esto evita que existan espacios no deseados al final del archivo que puedan ser interpretados como salida de HTML y puedan crear un error típico: <i>"Cannot modify header information - headers already sent by..."</i> [40]
Operadores	Todos los operadores binarios (operadores que están entre dos valores), como <code>+</code> , <code>-</code> , <code>+=</code> , <code>!=</code> , <code>==</code> , <code>></code> etc. Deben tener un espacio antes y después del operador, para facilitar la lectura. Por ejemplo, una asignación debe tener el formato <code>\$foo = \$bar;</code> en vez de <code>\$foo=\$bar;</code> . Los operadores unarios (operadores que operan sobre un solo valor), como <code>++</code> , no deben tener espacios entre el operador y la variable o número que se utiliza.[40]
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. Por ejemplo, <code>"<h2>\$header</h2>"</code> . También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.[40]
Uso de punto y coma	El lenguaje PHP requiere punto y coma al final de la mayoría de las líneas, pero permite que se omitan al final de los bloques de código. Los estándares de codificación de Drupal los requieren, incluso al final de los bloques de código.[40]

Estructura de control	<p>Con respecto a las estructuras de control, hay que tener en cuenta las siguientes normas:[40]</p> <ul style="list-style-type: none"> • Debe haber un espacio entre el comando que define la estructura (<i>if</i>, <i>while</i>, <i>for</i>, etc.) y el paréntesis de apertura, para distinguirlos de las llamadas a funciones. • La llave de apertura {se situará en la misma línea que la definición de la estructura, separada por un espacio. • Se recomienda usar siempre las llaves {}, incluso en situaciones donde son técnicamente opcionales. • Las estructuras <i>else</i> y <i>elseif</i> se escribirán en la línea siguiente al cierre de la sentencia anterior.
Arreglos	<p>Los valores dentro de un <i>array</i> (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados.</p> <p>Cuando la línea de declaración del <i>array</i> supera los 80 caracteres, cada elemento se debe escribir en una única línea, intentándolo una vez (2 espacios). Tomar en cuenta la coma en el final del último elemento del arreglo ¡Esto no es un error! Ésta ayuda a prevenir errores de análisis sintáctico, si otro elemento es colocado al final de la lista más tarde.[40]</p>
Nombres de archivos	<p>Todos los archivos de documentación deben tener en el nombre la extensión “.txt” para hacer más fácil su lectura además deben ser en mayúsculas. [40]</p>

3.3 Verificación de la propuesta de solución

La estrategia de pruebas es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistemática. Por esta razón, durante el proceso de desarrollo de software, debe definirse una plantilla para la prueba del este: un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos[41].

La prueba de software es un elemento de un tema más amplio que usualmente se conoce como verificación y validación. La verificación se refiere al conjunto de tareas que garantizan

que el software implementa correctamente una función específica. La validación es un conjunto diferente de tareas que aseguran que el software que se construye sigue los requerimientos del cliente[42].

Pruebas de Caja Negra

Las pruebas de diseño de caja negra, son las que utilizan el análisis de la especificación, tanto funcional como no funcional, sin tener en cuenta la estructura interna del programa para diseñar los casos de prueba y, a diferencia de las pruebas de caja blanca, estas pruebas se suelen realizar durante las últimas etapas de la prueba.

[43].

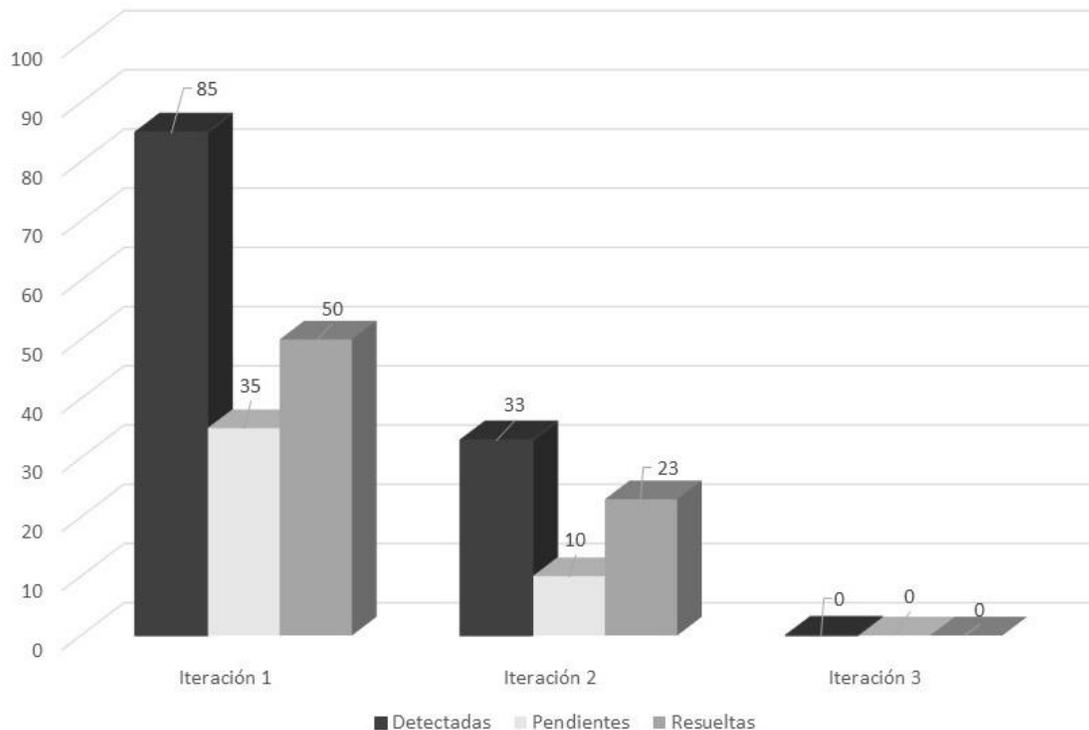
Tabla 5: Caso de prueba de requisito Crear Noticia

Escenario	Descripción	Título	Descripción	Imagen Principal	Respuesta del sistema	Flujo Central
EC81.1 Crear noticia de forma correcta.	El sistema crea noticias de forma correcta.	V	V	Prueba	El sistema almacena la información y muestra un mensaje de confirmación de la acción." Contenido o Título ha sido creado"	1-El usuario accede al bloque superior Administrar /Contenido/+Añadir Contenido o atajos/Añadir Contenido y el sistema muestra todos los tipos de contenidos que pueden ser añadidos. 2-
		Título de la noticia.	Contenido de la noticia.			
EC81.2 Crear noticias dejando campos vacíos.	El sistema no crea noticias dejando campos obligatorio vacíos.	I	I	I	El sistema no almacena información y muestra los siguientes de error "El campo Título es obligatorio." "El campo Descripción es obligatorio." "El campo Imagen es obligatorio."	El usuario selecciona el tipo de contenido Servicio Noticias. 3-El sistema muestra un formulario para que el usuario introduzca la información.4- El usuario introduce la información y presiona el botón "Guardar".

Tabla 6: Caso de Prueba Crear Noticia

Caso de prueba Crear Noticia.	
Código de caso de prueba: 1	Nombre de Historia de Usuario: Crear Noticia
Nombre de la persona que realiza la prueba: Yarianna Montesino Martinez	
Descripción de la prueba: Prueba a la funcionalidad crear noticia	
Entrada/ Pasos de la ejecución: La entrada consta de la introducción de los datos en los siguientes campos: <ul style="list-style-type: none"> • Título: Firman Vietnam y Cuba diez acuerdos en el ámbito empresarial. • Lenguaje: español. • Descripción: Descripción de la noticia Firman Vietnam y Cuba diez acuerdos en el ámbito empresarial. • Imagen: <div data-bbox="258 772 1166 1528" data-label="Image"> </div> 	
Resultado esperado: Se añade la noticia.	
Evaluación de la prueba: Satisfactoria.	

Tabla 7: Comportamiento de las no conformidades durante cada iteración.



Como se puede apreciar en la tabla 7, se realizaron 3 iteraciones de pruebas funcionales donde se encontraron un total de 118 errores. La gran mayoría de los errores fueron solucionados durante las dos primeras iteraciones. Los errores pendientes fueron resueltos en las siguientes iteraciones. Algunos de los errores detectados fueron:

1. Imágenes dañadas.
2. Opciones funcionando de forma errónea.
3. Errores de traducción.

En la última iteración no se encontraron nuevos errores, lo que evidencia la buena funcionalidad de la solución implementada.

Pruebas de Seguridad

Las pruebas de seguridad se definen como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones web, buscando disminuir el impacto de ataques a ellas y pérdida de información importante[44].

La seguridad en aplicaciones web busca asegurar la confidencialidad, disponibilidad e integridad de los datos y funciones que maneja el software, teniendo en cuenta el impacto que pueden tener fallas de seguridad según el contexto empresarial[44].

Para evaluar la seguridad de la solución implementada se empleó la herramienta Acunetix WVS de la cual se obtuvieron los resultados que se muestra en la siguiente tabla:

Tabla 8: Resultados de prueba de seguridad

Categorías de vulnerabilidades	Cantidad de errores
Formularios HTML sin protección CSRF	2
Credenciales de usuarios enviadas en texto plano	4
Vínculos rotos	2
Campos de contraseña con auto completamiento activado	4
Campos de usuario y contraseña mostrados	0
Total	12

Los formularios HTML sin protección CSRF resultaron ser una falsa alarma durante la prueba. Los errores de credenciales de usuarios enviadas en texto plano fueron corregidos mediante la utilización del módulo Secure Login, que permite la conexión con el sistema mediante el protocolo seguro HTTPS. Los vínculos rotos fueron reparados y los campos de contraseña con auto completamiento activado fue solucionado mediante la asignación del valor off a la opción de autocomplete en las etiquetas de formularios existentes en el código fuente del sistema. Luego de haber corregido los errores de seguridad existentes se puede decir que el sistema implementado es seguro.

Pruebas de Usabilidad

La usabilidad se define como el alcance de un producto o servicio usado por usuarios específicos, para lograr un objetivo con efectividad, eficiencia y satisfacción en un contexto específico. La efectividad se valora en términos de la precisión y completitud con la que el usuario alcanza sus objetivos, así como los errores cometidos, mientras que la eficiencia se refiere al tiempo y

esfuerzo para alcanzar los objetivos. La satisfacción se considera en términos de la ausencia de incomodidad y una actitud positiva hacia el uso de la aplicación[45].

Una manera de evaluar la usabilidad es mediante pruebas de usabilidad, las cuales involucran medir la ejecución de usuarios sobre la realización de ciertas tareas en un ambiente controlado[45].

Para mejorar la interacción con el usuario se realizó una prueba de usabilidad mediante el uso de una lista de chequeo aplicable a portales y aplicaciones web. Dicha lista tiene como objetivo analizar la usabilidad del portal web a través de indicadores, distribuidos en 9 categorías. En la siguiente tabla se muestran los resultados de dicha prueba:

Forma de uso

El mismo se evalúa de 1 en caso de mal (cuando la respuesta al indicador sea “No”) y 0 en caso que elemento revisado no presente errores (cuando la respuesta al indicador sea “Sí”).

NP (No Procede): Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso[45].

Tabla 9: Lista de chequeo Prueba de Usabilidad

Indicador a evaluar	Evaluación	NP
Visibilidad del sistema		
¿La página refleja la identidad de la empresa, logos, compañía...)?	0	
¿Cada pantalla empieza con un título que describe su contenido?	0	
¿Cuándo se selecciona un icono se diferencia de los no seleccionados?	0	
¿Los enlaces del menú se resaltan cuando se seleccionan?	0	
¿Los iconos que aparecen se identifican claramente con lo que representan?	0	
¿El menú de navegación aparece en un lugar destacado?		X
¿No utiliza más de siete opciones principales en el menú de navegación?		X
¿Si la respuesta a una acción se retrasa, aparece un mensaje o indicio como que el sistema está procesando la acción?	1	
¿El sitio le indica al usuario en que parte de la estructura del sitio web se encuentra, es decir si muestra 'migas de pan'?		X
¿El nombre de los enlaces es el mismo que el título de la página a la que dirige?	0	
¿El logo de la organización está ubicado en el mismo lugar en todas las páginas,	0	

y hacer clic en el logo retorna al usuario a la página más lógica (Ejemplo: la página de inicio)?		
¿Los títulos de las páginas, tablas e imágenes son descriptivos y distintivos?	0	
¿Las etiquetas de las categorías describen con precisión la información de las mismas?	0	
¿Cuándo una tarea involucra documentos fuente, la interfaz es compatible con las características del documento fuente?	0	
¿Las imágenes se muestren con buena resolución?	0	
¿No se muestran errores ortográficos?	0	
¿No hay ninguna imagen con información relevante?		X
Lenguaje común entre sistema y usuario		
¿El lenguaje es simple, con un tono adecuado?	0	
¿La información que se presenta en la aplicación es fácil de entender y memorizar?	0	
¿Utiliza los conceptos establecidos para las funciones estándar? ("buscar" para las búsquedas, etc.)	0	
¿Evita el lenguaje técnico: términos informáticos o propios de Internet?	0	
¿Se utiliza siempre la misma nomenclatura para las mismas funciones?	0	
¿Los acrónimos y abreviaturas son definidos al ser usados por primera vez?		x
¿No hace uso de términos extranjeros?	0	
¿Utiliza un texto específico y descriptivo en los vínculos?	0	
¿La información es de rápida lectura, y con una disposición asequible?	0	
¿Los vínculos basados en nombres de la gente, conducen a las biografías cortas o a sus propios blogs, no a un correo electrónico?		X
¿Si se desea incluir un enlace de correo electrónico, se muestra el correo y no el nombre de la persona?		X

En tablas anteriores se muestra un total de 28 indicadores de usabilidad, el sistema implementado posee 21 de ellos, mientras que el resto de estos no procede. El sistema cumple con 20 de los 21 indicadores de usabilidad que utiliza, lo que indica un porcentaje alto del 95.23 de usabilidad para la solución desarrollada.

Conclusiones del capítulo

El desarrollo de este capítulo permitió implementar un sistema con calidad en cuanto a funcionalidades y organización, mediante los estándares de codificación definidos y a las pruebas una vez concluido su desarrollo. La representación del diagrama de componentes permitió visualizar con más facilidad la estructura general del sistema. La ejecución de las pruebas permitió detectar las deficiencias presentes, subsanarlas en el menor tiempo posible y ofrecer un sistema con mayor seguridad y usabilidad. Como resultado de esta fase se obtiene un producto funcional para entregar al cliente, el que debe garantizar el cumplimiento de los requisitos estipulados con la calidad requerida.

CONCLUSIONES FINALES

El estudio realizado sobre los referentes teóricos permitió definir el alcance de la investigación y la naturaleza de la solución, planteándose como solución un sistema informático para gestionar la información de forma digital del MINCEX.

El análisis de las diferentes herramientas y tendencias para el desarrollo de sistemas informáticos, permitió seleccionar un ambiente de desarrollo idóneo para la implementación de la propuesta de solución, tomándose como base de desarrollo el CMS Drupal.

La combinación de conocimientos adquiridos de las distintas áreas del conocimiento como son la programación, la ingeniería y base de datos, entre otras, permitieron, a partir del análisis, diseño e implementación, obtener todas las características funcionales y no funcionales necesarias para realizar el sistema informático que da solución al problema planteado.

La definición y aplicación de una estrategia de pruebas compuesta por diferentes técnicas y niveles, permitió comprobar el correcto funcionamiento del sistema desde etapas tempranas para así garantizar su calidad, rendimiento, y seguridad.

La implementación de la solución y su posterior validación con el cliente, permitió comprobar que el sistema informático contribuye a gestionar la información del MINCEX.

RECOMENDACIONES

Se recomienda añadir un soporte de chat con inteligencia artificial que permita responder a las preguntas más frecuentes de los usuarios.

REFERENCIAS BIBLIOGRÁFICAS

1. Equipo, V., *Diseño básico de páginas web en HTML*. 2009: Editorial Vértice.
2. Reyes, Y.R.S., M.D. Fernández, and M.B.I. Abreu, *Observatorios: clasificación y concepción en el contexto iberoamericano*. Revista Cubana de Información en Ciencias de la Salud (ACIMED), 2019. **30**(2 %@ 2307-2113).
3. Rodríguez, Y.S., et al., *Observatorio Científico, Tecnológico y de Innovación de Cuba para la sostenibilidad de las ciencias*. Bibliotecas. Anales de investigación, 2021. **17**(3): p. 60-71 %@ 1683-8947.
4. De la Vega, I., *Tipología de Observatorios de Ciencia y Tecnología. Los casos de América Latina y Europa*. Revista española de documentación científica, 2007. **30**(4): p. 545-552 %@ 1988-4621.
5. Segura, F.O., *Sistema de gestión: Una guía práctica*. 2005: Ediciones Díaz de Santos.
6. Pérez, D., et al., *Determinación de las necesidades de formación e información utilizando el modelo AMIGA*. Rev Apic, 2013. **15**(1): p. 17-40.
7. Aja Quiroga, L., *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. Acimed, 2002. **10**(5): p. 7-8 %@ 1024-9435.
8. Fernández Marcial, V., *La gestión de la información y las habilidades informacionales*. 2008.
9. Prieto Olivares, L.H.J.A.y.D.d.S., *Diagrama De Componentes*. 2013.
10. Quintana, A.G., R.R. Villavicencio, and O.B. Betancourt, *Observatorio Científico Tecnológico: Propuesta de un modelo para el sector empresarial cubano*. Revista Digital Sociedad de la Información, 2009. **16**.
11. Quintero, J.J. and S.E. Romero, *ELEMENTOS QUE CONFORMAN LOS OBSERVATORIOS DE CIENCIA, TECNOLOGÍA E INNOVACIÓN. LA CIENCIA Y LA EDUCACIÓN ESCENARIO PARA LA CONSTRUCCIÓN DE PAZ*, 2017: p. 1823.
12. Giráldez Reyes, R., et al., *Observatorio de datos públicos en la gestión de la dinámica del envejecimiento poblacional en territorios*. Revista Universidad y Sociedad, 2021. **13**(6): p. 59-67 %@ 2218-3620.
13. Larivière, V., et al., *The place of serials in referencing practices: Comparing natural sciences and engineering with social sciences and humanities*. Journal of the American Society for Information Science and Technology, 2006. **57**(8): p. 997-1004 %@ 1532-2882.
14. Jaramillo Salazar, H., *El observatorio colombiano de ciencia y tecnología*. 1999.
15. Aguilar, M., *Indicadores de ciencia, tecnología e innovación en Venezuela y su impacto en el desarrollo de políticas públicas*. Telos: revista de Estudios Interdisciplinarios en Ciencias Sociales, 2017. **19**(1): p. 119-146 %@ 2343-5763.
16. López Vélez, P.A., et al., *Concepto, modelo y metodología del Observatorio de Innovación Social*. 2015: Corporación Universitaria Minuto de Dios.
17. Naser, A. and G. Concha, *Datos abiertos: Un nuevo desafío para los gobiernos de la región*. 2012.
18. Pressman, R.S., *A practitioner's approach*. Software Engineering, 2010. **2**: p. 41-42.
19. Alfonso Benítez, D., *Herramienta para generar productos de trabajos de la metodología variación AUP-UCI*. 2017, Universidad de las Ciencias Informáticas. Facultad 3.
20. Durango, A., *Diseño Web con CSS: 2ª Edición*. 2015: IT Campus Academy.
21. Preprocessor, H., *PHP*. Disponible online: <http://php.net/manual/es/intro-what-is.php>, 2002.

22. Kabir, M.J., *Apache Server Bible*. 1998: IDG Books Worldwide, Inc.
23. Combaudon, S., *MySQL 5.7: administración y optimización*. 2018: Ediciones Eni.
24. Patel, S.K., V.R. Rathod, and J.B. Prajapati, *Performance analysis of content management systems-joomla, drupal and wordpress*. International Journal of Computer Applications, 2011. **21**(4): p. 39-43.
25. Cristina, V.D.M., *ESTUDIO DEL FRAMEWORK SYMFONY2 PARA EL*.
26. Zorrilla Pila, L.E., *Portal web del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas*. 2016, Universidad de las Ciencias Informáticas. Facultad 1.
27. Vázquez-Ingelmo, A. and F.J. García-Peñalvo, *Modelo de Dominio*. 2019.
28. Agut, R.M., *Especificación de Requisitos Software según el estándar de IEEE 830*. Universidad Jaume I. Departamento de Informática. Paper, 2001.
29. Díaz, I., J. Sánchez, and O. Pastor. *Metamorfosis: Un Marco para el Análisis de Requisitos Funcionales*. 2005.
30. Buitrón, S.L., B.L. Flores-Rios, and F.J. Pino, *Elicitación de requisitos no funcionales basada en la gestión de conocimiento de los stakeholders*. Ingeniare. Revista chilena de ingeniería, 2018. **26**(1): p. 142-156 % @ 0718-3305.
31. Fuentes, J.R.L.X., *Desarrollo de software ágil: Extreme Programming y Scrum*. 2015: IT Campus Academy.
32. Verdecia Conde, R., *Módulo para la gestión de boletines en aplicaciones web desarrolladas con Drupal 7*. 2016, Universidad de las Ciencias Informáticas. Facultad 1.
33. M Zapata, C., B. Mary Estrada, and F. Arango, *Un método para el refinamiento interactivo del diagrama de clases de UML*. Dyna, 2007. **74**(153): p. 253-266 % @ 0012-7353.
34. Díaz, M.P., S. Montero, and I. Aedo, *Ingeniería de la web y patrones de diseño*. Pearson. Prentice Hall, 2005.
35. Stencel, K. and P. Węgrzynowicz. *Implementation variants of the singleton design pattern*. 2008. Springer.
36. Malcher, V., *Decorator pattern in web application*. International journal of advanced information technology, 2013. **3**(4): p. 13 % @ 2231-1548.
37. Summer, A., *Drupal*.
38. Zapata, C.M. and G.L. Garcés, *Generación del diagrama de secuencias de UML 2.1. 1 desde esquemas preconceptuales*. Revista EIA, 2008(10): p. 89-103 % @ 1794-1237.
39. Torres Crego, A., *Desarrollo de un mecanismo de autenticación centralizada para el entorno de aplicaciones de la UCI*. 2016, Universidad de las Ciencias Informáticas. Facultad-5.
40. Galens Molina, M. and I. Camejo Monroy, *Generador de portales web*. 2015, Universidad de las Ciencias Informáticas. Facultad 1.
41. Palacios, M., J. García-Fanjul, and J.J.X.J.e.I.d.S.y.B.d.D. Tuya, *Definición de una Estrategia de Pruebas basada en Acuerdos de Nivel de Servicio*. p. 519-524.
42. Sanchez Peño, J.M., *Pruebas de software. fundamentos y técnicas*. 2015.
43. Montoya, E.S., *Prueba funcional del software: un proceso de verificación constante*. 2013: Instituto Tecnológico Metropolitano.
44. Diaz Diaz, S.M., *Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software*. 2014.
45. Morales, G.S., C. Mezura-Godoy, and E.J.T.E.R.C. Benítez-Guerrero, *Proceso de Pruebas de Usabilidad de Software: Software Usability Testing Process*. 2018. **5**(1): p. 27-32.

ANEXOS

Historia de Usuario	
Número: RF24	Nombre: Crear Boletín
Usuario: CSCEIE	
Programador: Yarianna Montesino Martinez	Iteración asignada: Primera iteración
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en desarrollo: Bajo	Tiempo Real: 3 días
Descripción: El CSCEIE podrá crear boletines en el sistema. De un boletín se registra: el título, lenguaje, un resumen, portada, archivo PDF, tipo de boletín, numero de edición y fecha.	
Observaciones: Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el boletín. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.	
Prototipo de Interfaz: No aplica	

Historia de Usuario	
Número: RF76	Nombre: Crear Nota Informativa
Usuario: CSCEIE	
Programador: Yarianna Montesino Martinez	Iteración asignada: Primera iteración
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en desarrollo: Bajo	Tiempo Real: 3 días
Descripción: El CSCEIE podrá crear Nota Informativa en el sistema. De una Nota Informativa se registra: el título, imagen, archivo PDF, fecha, un resumen y autor.	
Observaciones: Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente la nota informativa. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.	
Prototipo de Interfaz: No aplica	