



Universidad de Ciencias Informáticas

FACULTAD 1

Módulo de reporte y gestión del desempeño de los editores del sitio web "Fidel, Soldado de las Ideas"

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

Autor: Tomás Amaury Abreu Quintana

Tutores:

MSc. Hubert Viltres Sala

MSc. Disnayle Jorge Chacón

Ing. Geidy Acosta Méndez

La Habana, diciembre 2021

Agradecimientos

A mis padres y mis abuelas por apoyarme y guiarme siempre, por ser mis ejemplos a seguir, por confiar en mí. Porque sin ellos no hubiera llegado hasta aquí y por ser los mejores del mundo.

A la Universidad de las Ciencias Informáticas por haberme dado la oportunidad de estudiar en este centro de altos estudios y por formarme como una profesional.

A toda mi familia, tías, tíos, por su cariño y apoyo a lo largo de mi vida.

A mi madrina por estar siempre presente.

A mi novia por ser mi apoyo y comprenderme siempre, aún en los momentos más difíciles.

A mis amigos, que me regaló la Universidad a lo largo de estos años tan hermosos.

A mis tutores por ayudar a que este proceso fuera un poco más sencillo, por estar siempre atentos y sus orientaciones.

A mi Facultad, por todo lo que ayudó en mi desarrollo, por todos los logros, amigos y profesores que tuve, los llevaré a todos siempre en mis pensamientos.

A ese maravilloso claustro de profesores que nos acompañó a lo largo de esta carrera.

*A mis amistades de toda la vida que han estado pendientes todo este tiempo y me han acompañado
Gracias a todas las personas que de una forma u otra me han apoyado, ayudado, aconsejado, animado
durante toda mi carrera.*

Dedicatoria

A mis padres y a mi abuela por su presencia y apoyo, los amo.

Declaración de autoría

Por este medio yo: Tomás Amaury Abreu Quintana, con carné de identidad: 97010608520 declaro que soy el autor principal del trabajo titulado “Módulo de reporte y gestión del desempeño de los editores del sitio web “Fidel, Soldado de las Ideas” ”, autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en beneficio de la gestión del conocimiento, así como a hacer uso de los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los 11 días del mes de diciembre de 2021



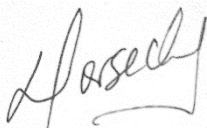
Firma del autor

Tomás Amaury Abreu Quintana



Firma del tutor

MSc. Hubert Viltres Sala



Firma de la tutora

MSc. Disnayle Jorge Chacón



Firma de la tutora

Ing. Geidy Acosta Méndez

Resumen:

El presente trabajo se sustenta en una necesidad práctica para el seguimiento y control del trabajo de editores del sitio web “Fidel: Soldado de las Ideas”. A partir de esta demanda se plantea como objetivo de la investigación: desarrollar un módulo para la generación de reportes y la gestión del desempeño de los editores del sitio web “Fidel, Soldado de las Ideas” versión 3. Para la implementación se utilizó Drupal en su versión 9.2.1, y como guía del proceso, la metodología de desarrollo de software programación extrema (XP). Se emplearon los lenguajes de programación PHP 7.4.19, HTML 5 y CSS 3. La solución cuenta con PHPMy Admin 5.1 y MySQL 5.7.31 como gestores de bases de datos. El modelado se realizó con la herramienta Visual Paradigm 16.2 y el lenguaje de modelado fue UML. Durante el estudio se conformó la primera visión de implementación del sistema, con una estrategia para lograr una construcción flexible y robusta a través de una arquitectura afín, patrones de diseño bien establecidos y estándares de codificación. Se recomienda la realización de todas las pruebas planteadas en el plan de pruebas para comprobar el correcto funcionamiento del módulo y con su ejecución, se espera contribuir a optimizar el tiempo de generación de reportes y a mejorar la calidad del sistema en general.

Palabras claves: *Drupal, módulo, reporte, evaluación.*

Índice:

Introducción	1
Capítulo 1: Generación de reportes y evaluación del desempeño de editores de páginas webs	7
1.1 Reporte.....	7
1.2 Estudio de herramientas existentes para generación de reportes automatizados y evaluación del desempeño:	8
1.3 Tecnologías y metodología asociadas al desarrollo de la solución	10
1.3.1 CMS	11
1.3.2 Herramienta CASE y lenguaje para el modelado	13
1.3.3 Lenguaje de programación	15
1.3.4 Framework de desarrollo	18
1.3.5 Editor de código fuente	19
1.3.6 Gestor de Base de Datos	20
1.3.7 Servidor Web.....	22
1.3.8 Herramientas de prueba	22
1.3.9 Metodología de Desarrollo de Software	23
1.4 Conclusiones parciales	28
Capítulo 2: Análisis y diseño del módulo para la generación de reportes y evaluación del desempeño.	29
2.1 Descripción de la propuesta de solución.....	29
2.2 Modelo de dominio	29
2.3 Requisitos de software	32
2.3.1 Requisitos funcionales	32
2.3.2 Requisitos no funcionales	33
2.4 Definición de casos de uso	34

2.4.1 Diagrama de casos de uso.....	35
2.5 Arquitectura de Drupal.....	35
2.6 Patrones de diseño en Drupal	37
2.7 Modelos de diseño.....	38
2.7.1 Diagramas de clases del diseño.....	38
2.7.2 Diagramas de secuencia del diseño (DS).....	39
2.8 Conclusiones parciales.....	41
Capítulo 3: Implementación y prueba del módulo para la generación de reportes y evaluación del desempeño.....	42
3.1 Diagrama de componentes.....	42
3.1.1 Modelo de despliegue	45
3.2 Estándares de codificación	46
3.3 Validación de la propuesta de solución.....	50
3.3.1 Funcionales.....	50
3.3.2 Integración	51
3.3.3 Seguridad	51
3.3.4 Casos de pruebas.....	51
3.3.5 Pruebas de rendimiento.....	53
3.4 Conclusiones parciales.....	54
Conclusiones	55
Recomendaciones	56
Bibliografía.....	57
Anexos.....	61

Introducción

El sitio web "Fidel, Soldado de las Ideas" fue creado por la Oficina de Información del Comité Central del Partido Comunista de Cuba en el año 2008, concebido como una enciclopedia del pensamiento y la obra del Comandante en Jefe. Se gestó por iniciativa de Cubadebate con el apoyo de la Universidad de Ciencias Informáticas, de instituciones y personas que atesoraban una parte del caudal de la vida y obra del líder histórico de la Revolución Cubana y otras que contribuyeron a su compilación.

En el portal se encuentra concentrada la mayor parte de los discursos, reflexiones, documentos, artículos, libros, cartas y mensajes escritos por él; además de libros, artículos, cartas, mensajes, entrevistas, valoraciones y reconocimientos dedicados a él. También está publicada su biografía oficial, la cronología de su vida y la reseña de los principales viajes realizados, así como una amplia galería de fotos, caricaturas, carteles, audios y videos. Cuenta además, con una sección de noticias y un espacio para la interacción con sus usuarios.

Al conservar un volumen considerable de la obra intelectual y de la vida personal de Fidel, el sitio permitirá el estudio y divulgación del pensamiento de una de las figuras políticas y revolucionarias más importantes del siglo XX y de inicios del XXI. Por la relevancia y trascendencia universal de su obra, el acceso y uso de esta información adquiere un extraordinario valor histórico, teórico y práctico para diferentes campos del conocimiento, así como para el análisis y búsqueda de soluciones ante disímiles situaciones en la contemporaneidad. Al mismo tiempo, el sitio se encuentra en constante evolución y en este sentido se generan nuevas demandas de su actualización tecnológica.

En el sitio web aparece publicado un amplio volumen de información recopilada mayoritariamente en español y en otros idiomas. Estos contenidos deben cumplir unos requisitos de calidad, entre los que se encuentran su exactitud, pertinencia, estar fundados en documentos y autores, expresados en un lenguaje correcto y un estilo apropiado para la audiencia prevista en función de los objetivos. Otros elementos de la calidad de los contenidos están relacionados con la ortografía, la traducción, la correcta citación de las fuentes, la estructura de interrogación, la ordenación de la información y los recursos hipertextuales. (García de León y Garrido Díaz, 2002)

Para el funcionamiento del sitio es necesario realizar un trabajo editorial minucioso de revisión de documentos provenientes de diferentes fuentes, medios de prensa, colaboradores personales, artistas e intelectuales. Estos nutren el proceso permanente de actualización de sus contenidos. La revisión y producción de contenidos se lleva a cabo por un pequeño equipo de edición. Esta labor es clave en el funcionamiento y calidad del sitio. En este sentido, se conciben dos aspectos fundamentales, la edición de la web, y el trabajo periodístico. Para la medición de la calidad de la edición deben cumplirse las normas del manual editorial, y un requisito mínimo de entradas o comentarios moderados. Se espera que por día se produzcan como mínimo 15 entradas, y se moderen 1500 comentarios. En el caso del trabajo periodístico se exige un mínimo de cuatro trabajos al mes por editor.

Por la variedad de información que se publica y por la fluctuación de los editores es necesario disponer de una herramienta para el seguimiento y control de su labor. Para esto, el equipo de editores debe realizar informes periódicos sobre los contenidos que suben al sitio. Estos deben ser clasificados en diferentes categorías que incluyen tanto la obra de Fidel, sus artículos, discursos e intervenciones, citas, libros, cartas y mensajes, como la obra producida sobre él: valoraciones, poemas, fotografías, pinturas, entre otras. En dichos reportes se recopilan las estadísticas sobre los diferentes tipos de contenidos publicados por cada editor, la fecha de publicación e idioma. De este modo es posible también evaluar periódicamente el rendimiento del trabajo individual y del equipo editorial, al medir la cantidad y calidad de los contenidos que se revisan y publican. Hasta hoy, la generación de estos reportes se lleva a cabo manualmente. A partir del análisis realizado se han identificado algunas deficiencias en el proceso de control del trabajo editorial y en la realización de los informes que requieren ser solucionadas. Entre ellas se encuentran:

- No se dispone de un mecanismo centralizado que gestione el desempeño de los editores.
- Existe dificultad para controlar el avance del trabajo de los editores en la revisión y publicación de los contenidos.
- No se dispone de un sistema de diseño automatizado de reportes flexibles que permita su generación periódica.

Estas carencias en los mecanismos de gestión/control del trabajo de editores y de generación de los reportes pueden solucionarse con el uso de herramientas informáticas disponibles. La utilización de los recursos

tecnológicos de la informática permitiría la automatización de este proceso y de este modo, construir un registro sistemático de la información con la reducción de errores humanos. Al mismo tiempo contribuiría a minimizar el tiempo de generación de reportes y a mejorar su calidad. Además, permitiría la medición de la gestión del desempeño individual y del equipo editorial.

Teniendo en cuenta estos argumentos, este trabajo se plantea como **problema de investigación**: ¿Cómo contribuir a la automatización de la generación de reporte y la gestión del desempeño de los editores del sitio web: "Fidel, Soldado de las Ideas" versión 3?

El objeto de estudio se enfoca en la automatización del reporte y gestión de desempeño de los editores del sitio web "Fidel, Soldado de las Ideas" versión 3.

Como objetivo general se propone:

- Desarrollar un módulo para la generación de reporte y la gestión del desempeño de los editores del sitio web "Fidel, Soldado de las Ideas" versión 3.

Para dar salida al objetivo general se plantean los siguientes **objetivos específicos**:

1. Sistematizar los referentes teóricos fundamentales que sustentan el desarrollo de un módulo informático para situaciones de generación de reporte y gestión del desempeño de editores.
2. Diagnosticar el estado del proceso de generación de reporte y la gestión del desempeño de los editores del sitio web "Fidel, Soldado de las Ideas" versión 3.
3. Diseñar las funcionalidades del módulo de reporte y gestión del desempeño de los editores del sitio web "Fidel, Soldado de las Ideas" versión 3.
4. Implementar las funcionalidades del módulo de reporte y gestión del desempeño de los editores del sitio web "Fidel, Soldado de las Ideas" versión 3.
5. Validar las funcionalidades del módulo de reporte y gestión del desempeño de los editores del sitio web "Fidel, Soldado de las Ideas" versión 3.

Se elige como **campo de acción** el sitio web "Fidel, Soldado de las Ideas" versión 3, desarrollado en Drupal 9.

Este estudio se orienta por la siguiente **hipótesis**: el diseño y funcionamiento de un módulo de reporte y gestión del desempeño de los editores basado en las herramientas de Drupal 9.2.1 contribuye a mejorar el control de dicho desempeño y la calidad de los reportes que se generan en el sitio web "Fidel, Soldado de las Ideas".

Tareas de Investigación.

1. Realización de un estudio sobre las tendencias en la generación de reportes y evaluación del desempeño.
2. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
3. Selección de la metodología de desarrollo.
4. Definición de los requisitos funcionales y no funcionales de la propuesta de solución.
5. Implementación de la propuesta de solución.
6. Documentación de las pruebas de carga y estrés, de aceptación, integración

Los métodos científicos que se utilizan en la presente investigación son:

Métodos teóricos:

Análisis-Síntesis: para el estudio de los conceptos sobre las herramientas para la generación de reportes y evaluación del desempeño de editores, en las fuentes bibliográficas disponibles.

Análisis Histórico-Lógico: utilizado para realizar un estudio de cómo se ha comportado la creación y funcionamiento de herramientas para la generación de reportes y evaluación del desempeño de editores.

Inductivo-Deductivo: para el estudio de las principales iniciativas de generación de reportes y evaluación del desempeño, así como los algoritmos utilizados para lograrlo, con el objetivo de determinar las alternativas viables para su empleo en la presente investigación.

Modelación: para la representación de la solución propuesta a través de la modelación de diagramas que permiten modelar el proceso a seguir para desarrollar la solución creando una abstracción con el objetivo de explicar la realidad.

Métodos empíricos:

Observación: utilizado para obtener información sobre los sistemas de generación de reportes y evaluación del desempeño existentes.

Análisis documental: se emplea para la consulta de la literatura especializada en las temáticas afines a la investigación y para obtener información sobre los contenidos de los reportes editoriales y la evaluación del desempeño existentes.

El presente reporte se estructura en tres capítulos, además de las secciones de introducción, conclusiones, recomendaciones, bibliografía y anexos.

Capítulo 1: Generación de reportes y evaluación del desempeño

Se realiza un análisis de los principales conceptos asociados a la generación de reportes y evaluación del desempeño en los portales web que son necesarios para la comprensión de la presente investigación. Se presenta un estudio sobre los principales algoritmos que se utilizan para la generación de reportes y evaluación del desempeño. Además, se realiza un análisis de las herramientas, tecnologías y metodologías para el desarrollo de software.

Capítulo 2: Análisis y diseño del módulo de generación de reporte y la gestión del desempeño de los editores del sitio web “Fidel, Soldado de las Ideas” versión 3.

Se describe la propuesta de solución. Se especifican los requisitos funcionales y no funcionales. Se realiza el análisis y diseño a través de la transformación de los requisitos en casos de uso del sistema. Se muestran los diferentes diagramas de clases del diseño, diagramas de secuencia y de casos de uso.

Capítulo 3: Implementación y prueba del módulo de generación de reporte y la gestión del desempeño de los editores del sitio web “Fidel, Soldado de las Ideas” versión 3.

Se realiza la implementación del módulo y se especifican las pruebas realizadas al mismo para comprobar que cumple con los requisitos planteados y asegurar la calidad del software, para ello se utilizan diagramas de diseño, de despliegue, de componentes, así como los diferentes casos de prueba y resultados de los mismos.

Capítulo 1: Generación de reportes y evaluación del desempeño de editores de páginas webs

La generación de reportes sobre la revisión y publicación de contenidos en los sitios webs permite monitorear sistemáticamente y evaluar el rendimiento y calidad del trabajo editorial. El uso de las herramientas informáticas para la creación de reportes editoriales contribuye a dinamizar el flujo de trabajo y resulta en un instrumento para la toma de decisiones. En este capítulo se definirán los conceptos de reportes, los criterios para la evaluación del desempeño, así como las herramientas de generación de reportes automatizados.

1.1 Reporte

Los reportes o de gestión de informes facilitan la distribución de la información a los distintos niveles de la estructura organizativa. Al mismo tiempo, proporcionan la información requerida sobre la rutina productiva de los editores. Para diseñar e implantar un sistema de *reporting* es necesario realizar previamente un estudio de las necesidades de información y del propósito que se persigue con el reporte.

El reporte del trabajo de editores contiene datos recabados. Dicha información aparece organizada de forma tal que muestra aspectos nuevos o distintos sobre la temática abordada. El reporte, como cualquier trabajo escrito, debe tener un orden lógico, ser claro y preciso en las ideas, tener una buena presentación; su contenido debe respaldar la información presentada a partir de argumentos, reflexiones y referencias a las fuentes utilizadas. Un reporte debe ser claro y preciso. El editor tiene que presentar la información de manera tal que permita identificar con claridad la descripción de su rutina productiva, tanto en términos de la cantidad de la producción como de la calidad, en determinados plazos de tiempo. De este modo también, el reporte constituye un medio para monitorear el desempeño del editor y al mismo tiempo permitiría evaluar las rutinas productivas individuales y del medio, de manera sistemática.

Los reportes son documentos que contienen los detalles de un evento específico. Contienen información obtenida sobre el evento elegido de forma organizada.

Características de los reportes (visita, documental, etc.)

- Claridad.

- Concisión.
- Debe ser personal.
- Debe tener un orden lógico.
- Puede ser subjetivo, pero debe respaldar la información presentada con argumentos, reflexiones y referencias a fuentes confiables que deben ser debidamente citadas.

1.2 Estudio de herramientas existentes para generación de reportes automatizados y evaluación del desempeño:

Existen varias herramientas informáticas que se utilizan para la generación de reportes. Entre ellas:

Triconex

El software generador de reporte *Triconex* hace que los reportes exhaustivos sean sencillos. Permite una generación de reportes rápida y precisa de uno o más controladores o proyectos que involucran a la familia de soluciones de seguridad *Triconex*. El programa ayuda a combinar la información de controladores múltiples en un solo reporte. Los reportes predeterminados están disponibles para el uso en bloque de la función puerto a puerto e IDs duplicadas, así como para referencias cruzadas de nombres de etiquetas. Y los reportes personalizados pueden ser generados para cualquier conjunto de datos disponibles en la base de datos del programa. El generador de reportes *Triconex* también es usado en conjunto con el software de validación de seguridad *Triconex* para producir casos de prueba automatizados. Crea automáticamente proyectos de prueba y archivos de configuración asociados.

GEReport

Es una herramienta destinada al diseño, generación y configuración de los reportes relacionados con los datos históricos almacenados en una fuente de datos. Además, luego de contar con toda la información del reporte es posible exportarlo como imagen y en los formatos HTML, PDF y Excel. Para la interacción con las aplicaciones externas, el sistema implementa un servicio que expone los metadatos de los reportes para poder utilizarlos sin restricciones de lenguajes y plataformas. El sistema se creó sobre un entorno web y se

desarrolló siguiendo lo establecido por el Proceso Unificado de Desarrollo (RUP), utilizando UML como lenguaje de modelado. Para la gestión de la base de datos se seleccionó PostgreSQL, en la implementación se utilizaron los lenguajes de programación PHP, con CodeIgniter 2.0 como marco de trabajo del lado del servidor y JavaScript con Dojo Toolkit 1.8 para el trabajo del lado del cliente. Se utiliza en la Universidad de Cienfuegos por analistas y programadores del Grupo de Estudios y Desarrollo de Ingeniería y Sistemas, perteneciente a la Facultad de Ingeniería.

Reportes

Reportes es una aplicación en *AuditBond*. Puede utilizar la aplicación Reportes para crear informes personalizados a partir de Vistas de *HighBond* preensambladas.

Características y funcionalidades

- Crear reportes personalizados: genera reportes usando cualquier combinación de campos proporcionados en una Vista de *HighBond*.
- Visualizar datos: agrega múltiples gráficos a un reporte para resaltar tendencias de datos, valores atípicos o patrones.
- Compartir reportes: exporta, envía por correo electrónico o difunde reportes para compartir datos con otras personas.
- Mostrar datos: muestre reportes en tableros de mando y presentaciones para presentar información clave a las partes interesadas relevantes

Piktochart

Esta herramienta cuenta con varias funcionalidades para realizar infografías, presentaciones, posters o reportes. Posee varias opciones de edición y controles para que puedas ajustar el diseño, fuentes y fondos de acuerdo a tus necesidades y gustos.

Además, con *Piktochart* se pueden incluir gráficos simples e importar datos de archivos CSV. La versión gratuita incluye una gran variedad de imágenes e iconos vectoriales. Si bien las plantillas gratuitas son simples son más variadas que en Infogr.am.

Web Reports:

Web Reports inaugura una nueva categoría de herramientas, que se sitúan entre las herramientas CRM (sobre todo dedicadas a la recopilación personalizada de datos), y las herramientas de Datawarehouse o almacén de datos (que se concentran esencialmente sobre las bases de datos de la empresa). El objetivo de *Web Reports* es permitir la consulta en línea o intranet de cuadros de indicadores evolutivos que hacen referencia a encuestas en curso, encuestas pasadas y datos de la empresa. La realización de un sitio de consulta dinámica en la red o en intranet con *Web Reports* no requiere de programación web o HTML. Este permite a los encargados de estudios o a los equipos poner a disposición sobre la red, datos brutos, cuestionarios, análisis, reportes y criterios de selección, haciéndolos accesibles a los usuarios elegidos en función de sus códigos de acceso.

Análisis del estudio:

A partir del estudio realizado a las herramientas anteriores, se determina no utilizar ninguna de ellas como solución al problema planteado, debido a que no cubren las necesidades específicas de la Oficina de Información del Comité Central del Partido Comunista de Cuba. Dado que se necesita un módulo específico creado para el sitio web "Fidel, Soldado de las Ideas", capaz de dar información concreta en forma de tabla para su posterior análisis. Por tanto, no ofrecen la solución requerida al problema planteado en esta investigación sobre cómo generar reportes y evaluar el desempeño de los editores. A partir de estos elementos se dispone a implementar un módulo que cubra las necesidades identificadas en dicha entidad, de modo que sea capaz de satisfacer los requerimientos de la demanda expuesta por el cliente.

1.3 Tecnologías y metodología asociadas al desarrollo de la solución

Las herramientas son programas o aplicaciones que ofrecen la posibilidad de realizar varias funciones con diferentes propósitos. Por otra parte, las tecnologías son el conjunto de conocimientos técnicos, que

permiten diseñar y crear bienes y servicios. A continuación, se muestran todas las herramientas y tecnologías utilizadas para el desarrollo del módulo.

1.3.1 CMS

Con esta denominación se conoce una herramienta de software que permite crear, organizar y publicar documentos y otros contenidos de forma colaborativa. Los sistemas de gestión de contenidos (*Content Management System*, en adelante CMS] están formados por un conjunto de aplicaciones web que, de un modo similar a un portal, operan tanto en Internet como en una intranet.

Su principal ventaja consiste en el hecho de que permiten organizar y mostrar contenidos sin que sea necesario poseer grandes conocimientos de programación web. Con su ayuda se pueden generar y publicar noticias, crear taxonomías (sistemas de clasificación) con las que el administrador clasifica contenidos. También se pueden insertar logotipos e imágenes personalizadas o corporativas del portal, añadir secciones, administrar las bases de datos de usuarios, entre otras.

Funciones básicas de un gestor de contenidos:

➤ Creación de contenidos:

Cualquier usuario que no tenga demasiados conocimientos sobre publicación web puede generarlos gracias a los editores integrados. Por lo general, se trata de editores del tipo WYSIWYG con los que se puede ver el aspecto final del documento a medida que se elabora.

Los CMS ofrecen la posibilidad de preparar diferentes tipos de documentos (noticias, foros de discusión, libros en formato digital, entre otros, según las necesidades del portal e incluso crear nuevos tipos a partir de los tipos básicos. Además, pueden incluir paquetes ofimáticos integrados, así como herramientas para la importación de documentos y la edición de archivos XML.

➤ Gestión de contenidos:

Cuando un usuario crea un documento, se almacena en una base de datos junto con otras informaciones de interés, como son el nombre del autor, las fechas de creación y publicación, el tipo de documento, los permisos de acceso, entre otras. Además, se consignan también los usuarios a los que se ha dado

de alta, la estructura del portal, los estilos y temas utilizados, así como toda la información generada a partir de las funcionalidades del CMS.

Las labores de gestión de contenidos se pueden delegar o compartir con los usuarios. Basta con asignarles un perfil determinado (editor, administrador, autor, etc.). El CMS proporciona la infraestructura necesaria para que se comuniquen y garantiza la consecución del ciclo de trabajo.

➤ **Publicación de contenidos:**

Cuando el usuario da por terminado un contenido, puede publicarlo en ese mismo momento o bien programarlo para que aparezca en una fecha determinada. Esta tarea se lleva a cabo según el patrón general de diseño establecido en el gestor. De esta manera, se simplifica el trabajo y se garantiza que el CMS mantenga un aspecto homogéneo.

El autor, si su perfil se lo permite, podrá publicar el documento dentro de la sección para la que han desarrollado esos contenidos.

➤ **Presentación del gestor de contenidos:**

Por lo general, los CMS siguen la normativa internacional de accesibilidad. Pueden utilizarse desde cualquier navegador y permiten establecer el idioma, el huso horario e incluso, la moneda deseada.

1.2. Características generales de los sistemas gestores de contenidos

- El *frontend*, la parte visible del CMS para los usuarios que acceden como invitados o usuarios registrados.
- El *backend*, el área de administración del CMS donde se llevan a cabo todas las tareas de actualización, configuración y gestión del sitio.
- Módulos para la configuración y personalización del sitio web.
- Gestor de usuarios, perfiles y derechos de acceso.

- Módulos para la gestión del contenido del sitio web [creación, edición, publicación de contenidos, etc.).
- Plantillas para personalizar el aspecto o el diseño del sitio web.
- Extensiones que aumentan la funcionalidad del gestor de contenidos.
- Posibilidad de múltiples idiomas y localización.

Drupal 9:

Drupal es un software de gestión de contenido. Se utiliza para crear muchos de los sitios web y aplicaciones que se usan con frecuencia. Drupal tiene excelentes características estándar, entre las que se reconocen la creación de contenido sencilla, rendimiento confiable y excelente seguridad. Se distingue, además, por su flexibilidad. La modularidad es uno de sus principios fundamentales. Sus herramientas le ayudan a crear el contenido estructurado y versátil que resulta necesario en las experiencias web dinámicas.

También es una excelente opción para crear marcos digitales integrados. Puede ampliarse con uno o múltiples complementos. La funcionalidad de Drupal puede ampliarse a través de los módulos. Los temas le permiten personalizar la presentación de su contenido. Las distribuciones son paquetes de Drupal empaquetados que puede usar como kits de inicio. Es posible mezclar y combinar estos componentes para mejorar las habilidades centrales de esta herramienta. Igualmente, se puede integrar Drupal con servicios externos y otras aplicaciones en su infraestructura. Se considera que ningún otro software de gestión de contenido es tan potente y escalable.

El proyecto Drupal es un software de código abierto. Cualquiera puede descargarlo, usarlo, trabajar en él y compartirlo con otros. Se basa en principios como la colaboración, la globalización y la innovación. Se distribuye bajo los términos de la Licencia Pública General GNU (GPL). No se requiere abonar una tarifa de licencia para su uso, por lo que resulta accesible. (Drupal Association, 2020).

1.3.2 Herramienta CASE y lenguaje para el modelado

Una herramienta CASE es conjunto de programas especializados y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo

de un *software*. La inclusión de las herramientas CASE están tomando cada vez más relevancia en la planeación y ejecución de proyectos que involucran sistemas informáticos, debido a que permiten aumentar la productividad en el desarrollo de *software*. Estas herramientas, suelen inducir a sus usuarios a la correcta utilización de metodologías que implican una reducción tanto en el costo del proyecto, como en el tiempo de desarrollo del producto de software final (Battaglia, Neil, Martínez, y Vincenz, 2017).

Visual Paradigm 16.2

Es una potente herramienta profesional CASE (por sus siglas en inglés *Computer Aided Software Engineering*) que utiliza el modelado UML para el desarrollo de aplicaciones. Soporta el ciclo completo del desarrollo del *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, permite dibujar todos los tipos de diagramas de clases y generar documentación. Por lo que es ideal para ingenieros de software, analistas y arquitectos de sistemas que están interesados en construir sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Se considera muy completa y fácil de usar, con soporte multiplataforma y excelentes facilidades de interoperabilidad con otras aplicaciones (Visual Paradigm, 2021).

En esta investigación se decide utilizar esta herramienta en su versión 16.2, por las ventajas que ofrece. Ya que genera código y permite exportarlo en HTML, tiene licencia gratuita y comercial, utiliza un lenguaje estándar común y su diseño es centrado en casos de uso y enfocado en el negocio lo que permite generar un software de mayor calidad.

UML

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es un lenguaje de modelado visual común para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. Este lenguaje describe, a través de diagramas, los límites, la estructura y el comportamiento de un sistema. Es el más conocido y utilizado en la actualidad por ser el mejor para visualizar, especificar, documentar y construir los métodos o procesos de un sistema (Larman, 2005).

UML consiste en diferentes tipos de diagramas que describen al sistema y los objetos que contiene, este representa buenas prácticas para la construcción y documentación de diferentes aspectos del modelado de sistemas de software y de negocios. Está compuesto por diversos elementos gráficos que se combinan, y existen reglas para combinar tales elementos. Lo que hace posible representar diversas perspectivas de un mismo sistema. Por lo tanto, este será el lenguaje de modelado utilizado durante la investigación.

1.3.3 Lenguaje de programación

Un lenguaje de programación es un elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis. Para controlar el comportamiento de una máquina, particularmente una computadora (Parra, 2018).

JavaScript

JavaScript es uno de los recursos que han surgido para incorporar dinamismo y capacidades al lenguaje HTML. Actualmente es la tecnología más extendida en el enriquecimiento de páginas web del lado del cliente. Es un lenguaje de programación basado en prototipos, multiparadigma, dinámico y con soporte para programación orientado a objetos. Se utiliza principalmente para permitir mejoras en la interfaz de usuario y en las páginas web dinámicas. Entre las acciones típicas que se pueden realizar en JavaScript se tienen dos vertientes. Por un lado, los efectos especiales sobre páginas web y por el otro, JavaScript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas o tablas de cálculo (Parra, 2018).

El lenguaje JavaScript se inserta en documentos HTML, de forma que su código queda reflejado en la propia página y no es llamado o cargado de ninguna fuente externa. Se trata de un lenguaje interpretado puro y sensible a mayúsculas, aunque algunas implementaciones ignoran en parte este último extremo. Se escoge JavaScript para el desarrollo de la presente solución informática porque proporciona los medios para controlar las ventanas del navegador y el contenido que muestran, programar páginas dinámicas simples y optimizar los tiempos de carga y el tráfico del servidor. Además, permite capturar los eventos generados por el usuario y responder a ellos, comprobar los datos que el usuario introduce en un formulario antes de enviarlos y comunicarse con dicho usuario mediante diversos métodos.

CSS 3

Hojas de Estilos en Cascada o por sus siglas en inglés CSS, *Cascading Style Sheets*, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje marcado. Es muy utilizado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML. Es una tecnología muy utilizada por muchos sitios web para crear páginas visualmente atractivas, pues permite definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, así como la posición de cada elemento dentro de la página (Aubry, 2019).

Este lenguaje será utilizado para el desarrollo de la solución propuesta pues presenta muchas ventajas, entre las que se destacan la flexibilidad, compatibilidad, continuidad, la separación del contenido y presentación y la unificación del diseño de las páginas del sitio. Además, facilita la publicación de contenidos en múltiples formatos de presentación y brinda una gran consistencia al sitio.

PHP 7.4.19

Hypertext Pre-Processor, por su acrónimo en inglés, es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP es muy simple de utilizar para los principiantes, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. Además, ha sido ampliamente portado y puede ser desplegado en la mayoría de los servidores web en casi todos los sistemas operativos y plataformas, de forma gratuita.

Es similar a otros lenguajes informáticos como C o C++, rápido, de sintaxis cómoda y su sencillez contribuye a su rápido aprendizaje. Es un lenguaje interpretado, multiplataforma, se le pueden agregar extensiones fácilmente y dispone de una gran cantidad de librerías (F. Fernández, 2021). La meta del lenguaje es permitir rápidamente a los desarrolladores la generación de páginas dinámicas. Con PHP en su versión 7.4.19 se puede realizar el procesamiento de información en formularios, foros de discusión, entre otras.

Este lenguaje será utilizado en el desarrollo del sistema por las múltiples ventajas que posee, entre las que se encuentran:

- Posee una comunidad muy grande de desarrolladores y existe abundante documentación, tutoriales, ejemplos de código, foros etc.
- Es fácil de aprender comparado con otros lenguajes de programación.
- Capacidad de conexión con la mayoría de los manejadores de base de datos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Es un lenguaje que incluye soporte para la programación orientada a objeto.

HTML 5

Hyper Text Markup Language o HTML es un lenguaje informático de composición de documentos que cuenta con instrucciones especiales que son interpretadas por el navegador y como resultado se muestran textos, imágenes, vídeos y enlaces. Utilizando HTML se puede definir la apariencia de las páginas creadas de diferentes formas. Contiene una gran variedad de etiquetas que son las encargadas de mantener la estructura e indicar la semántica de un documento, convirtiéndose en un lenguaje informático para presentación con soporte de aspectos visuales (Aubry, 2019).

Es un estándar reconocido en todo el mundo, multiplataforma, soportado por muchos navegadores. Con él, se puede obtener un producto atractivo y rápido sin mucho esfuerzo por parte del desarrollador. Permite establecer enlaces entre diferentes documentos y la introducción de referencias a otras páginas por medio de enlaces de hipertexto. HTML5 es la actualización de HTML y, también es un término de mercadotecnia para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de Javascript.

Se decide utilizar HTML5 ya que presenta mejoras con respecto a las anteriores versiones de HTML. Además, incluye un nuevo conjunto de etiquetas para dar estructura a un documento HTML, abandonando el uso de tablas y etiquetas div para la maquetación de las páginas. En esta versión, los formularios son mejorados con una redefinición de nuevos tipos de elementos para la etiqueta input, y han sido incorporados al lenguaje nuevos atributos para los elementos de formulario.

jQuery

jQuery es una biblioteca multiplataforma de JavaScript que ofrece una sólida infraestructura que brinda un grupo importante de ventajas y facilidades para la creación de aplicaciones de mediana y alta complejidad del lado del cliente. Está orientada a facilitar la labor en cuanto a la creación de interfaces de usuario, efectos dinámicos y otros aspectos. Es un producto estable, bien documentado y con un gran equipo de desarrolladores a cargo de su mejora y actualización. Posee una dilatada comunidad de creadores de “plugins” o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, y otros (García, 2016).

Será utilizada en esta investigación pues es la biblioteca de JavaScript más utilizada, por un amplio margen y tiene importantes características que la hacen destacar, como la selección de elementos DOM y la manipulación de la hoja de estilos CSS. Además de permitir la creación de poderosas páginas web dinámicas, con efectos y animaciones personalizadas. Presenta gran compatibilidad con varios navegadores como Mozilla Firefox, Internet Explorer, Safari, Opera y Google Chrome, soporta extensiones JSON y es un software libre y de código abierto.

1.3.4 Framework de desarrollo

Un marco de trabajo, o framework, es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje informático interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Utilizar un marco permite agilizar los procesos de desarrollo, asegura unas buenas prácticas y la constancia del código. Ayuda a desarrollar aplicaciones con mayor rapidez, pues posee una estructura definida y una organización para el desarrollo y mantenimiento del software desarrollado.

Bootstrap

Es un framework o biblioteca multiplataforma que posee un conjunto de herramientas de código abierto para el diseño de sitios web. Contiene plantillas de diseño basadas en HTML y CSS, así como extensiones de

JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo *front-end* o interfaz de usuario. Es compatible con la mayoría de los navegadores web (Fernández, 2021). Además, contiene plantillas con tipografías, botones, cuadros, menús de navegación y otros elementos.

Bootstrap es modular, consistente y proporciona un conjunto de hojas de estilo que proveen definiciones básicas de estilo para todos los elementos HTML. Esto otorga una uniformidad al navegador y al sistema de anchura, da una apariencia moderna para el formateo de los elementos de texto, tablas y formularios.

1.3.5 Editor de código fuente

Un editor de código fuente es un editor de texto diseñado específicamente para editar el código fuente de programas informáticos, los cuales ofrecen múltiples herramientas para facilitar el trabajo y aumentar la productividad. Estos tienen características diseñadas exclusivamente para simplificar y acelerar la escritura de código fuente (Microsoft, 2021). Existen diferentes editores de código y su uso se basa en dependencia del sistema operativo que se utilice o a la gran variedad de lenguajes de programación que existen. Estos editores también proveen un modo conveniente de ejecutar un compilador, un intérprete, un depurador o cualquier otro programa que sea relevante en el proceso de desarrollo de software.

Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración y control integrado. Es gratuito y de código abierto, además, es compatible con varios lenguajes de programación como C++, HTML, Java, JavaScript y PHP. Tiene la capacidad de crear extensiones que analizan código y herramientas para análisis estático (Microsoft, 2021).

Visual Studio Code también es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos del teclado y las preferencias. Además, permite modificar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea y el lenguaje de programación del documento activo. Por lo que será utilizado en el desarrollo del sistema propuesto.

1.3.6 Gestor de Base de Datos

Un sistema Gestor de Base de Datos (SGBD) o *DataBase Management System* (DBMS) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible. Su objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta (Marín, 2019).

Las características de un Sistema Gestor de Base de Datos SGBD son:

- **Abstracción de la información:** los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Así, se definen varios niveles de abstracción.
- **Independencia:** la independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima:** un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante.
- **Consistencia:** será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad:** los SGBD deben garantizar que la información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado.
- **Integridad:** se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

- Respaldo y recuperación: los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- Control de la concurrencia: un SGBD debe controlar el acceso concurrente a la información, que podría derivar en inconsistencias (Hechavarría, 2002).

Por lo anteriormente planteado se escoge el siguiente Sistema Gestor de Base de Datos (SGBD):

MySQL

Es un Sistema Gestor de Base de Datos multihilo y multiusuario utilizado en la gran parte de las páginas web actuales. Tiene muchas ventajas pues presenta un gran rendimiento y facilidad de uso, soporte multiplataforma, facilidad para instalar y configurar, y soporte SSL. No se necesita disponer de un *hardware* o *software* de alto rendimiento para su ejecución, soporta gran cantidad de datos y ofrece un sistema de contraseñas y privilegios muy seguro.

MySQL está considerada como la base de datos de código abierto más popular del mundo, por lo que es utilizado por muchos sitios web como Wikipedia, Google, Twitter, Flickr y YouTube. Se puede utilizar con diversos lenguajes de programación incluyendo C, C++, C#, Java, PHP, Python, entre otros. Además, es muy popular en aplicaciones web como Wordpress, Joomla y Drupal, por ser una base de datos muy rápida en la lectura de información. Este SGBD ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.

PhpMyAdmin 5.1.0

Es una herramienta gratuita escrita en PHP con la intención de manejar la administración de MySQL a través de una interfaz gráfica de usuario, utilizando un navegador web. Es un desarrollo de código abierto y permite crear, modificar y eliminar registros y tablas de la base de datos (Marín, 2019). Está disponible en una gran cantidad de idiomas y es el administrador por defecto de muchos paneles de control como Lammpp y Xampp. Es muy fácil de utilizar y permite realizar búsquedas en la base de datos.

1.3.7 Servidor Web

Es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones con el cliente y cediendo una respuesta en cualquier lenguaje o cualquier aplicación del lado del cliente. Proporciona recursos solicitados a través del protocolo HTTP o de manera segura a través del protocolo HTTPS (Pérez, 2020). Actualmente los servidores web más utilizados son Apache y Nginx, por lo que se muestra a continuación una comparación entre ellos:

Apache

El servidor HTTP Apache es un servidor web de código abierto, altamente configurable y multi-plataforma. Es extensible y muy popular, por lo que es fácil de conseguir soporte. Es posible su ejecución en una multitud de sistemas operativos, lo que lo hace prácticamente universal. Apache es una tecnología gratuita y actualmente existen muchos módulos para Apache que son adaptables a éste, y están ahí para que sean instalados cuando sea necesario (Pérez, 2020). Su implementación se ha realizado y se sigue realizando de forma colaborativa, con prestaciones, características y funcionalidades equivalentes a las de cualquier servidor comercial.

Entre sus características destacan: Trabaja con PHP y otros lenguajes de script, teniendo todo el soporte que se necesita para tener páginas dinámicas. Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor y permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

1.3.8 Herramientas de prueba

Son herramientas informáticas que permiten reproducir la funcionalidad de una aplicación informática. Utilizadas para incrementar la calidad, fiabilidad y corregir los errores presentes en el sistema creado. Estas herramientas pueden tener varias clasificaciones: de gestión de pruebas, para pruebas funcionales y para pruebas de carga y rendimiento. Además, llevan consigo aplicativos que permiten realizar pruebas autónomas y masivas.

JMeter

Es un proyecto de Apache utilizado como una herramienta de prueba de carga para analizar y medir el rendimiento de una variedad de servicios, con énfasis en aplicaciones web. Se utiliza para realizar pruebas de carga, estrés y capacidad a las aplicaciones web del protocolo HTTP. JMeter soporta aserciones para asegurar que los datos recibidos son correctos, por lo que es una herramienta de realización de pruebas automáticas. Se considera como una herramienta ideal, pues detecta más de 6500 vulnerabilidades de aplicaciones web. Escanea código abierto y aplicaciones personalizadas y, además, detecta vulnerabilidades críticas con un 100% de precisión (Pérez, 2020).

Acunetix

Es una herramienta de seguridad de aplicaciones web automatizada, es un líder mundial en seguridad web. Es capaz de escanear cualquier sitio web que es accesible a través de HTTP o HTTPS. Actualmente Acunetix es capaz de comprobar más de 500 tipos diferentes de vulnerabilidades y anomalías en un sistema. Es centrado en el alto rendimiento, ya que escanea las aplicaciones y brinda los resultados en el menor tiempo posible. Además, ayuda a identificar la ubicación exacta de la vulnerabilidad en el código fuente o el código de bytes, lo que acelera en gran medida la corrección de las vulnerabilidades. Acunetix, a diferencia de la mayoría de los productos, puede descubrir vulnerabilidades peligrosas fuera de banda u ocultas y detecta más de 50 mil vulnerabilidades de red y configuraciones erróneas. (Acunetix, 2019).

1.3.9 Metodología 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 informático. Estas permiten reducir el nivel de dificultad, organizar tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar. Se pueden dividir en dos grandes grupos: las ligeras/ágiles y las pesadas/tradicionales.

Las metodologías ágiles se enfocan en la toma de decisiones y en la evolución de las soluciones según la necesidad del proyecto. Continuamente se adaptan y se ajustan a los requisitos y al entorno de desarrollo. Por otra parte, las metodologías tradicionales o pesadas se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar y requiere una extensa documentación, ya que pretende prever todo de antemano.

Programación extrema (XP)

Conocida por sus siglas XP (*eXtreme Programming*), es una metodología basada en un conjunto de reglas y buenas prácticas para el desarrollo de software en ambientes muy cambiantes con requisitos imprecisos, por ende, está enfocada en la retroalimentación continua entre el equipo de desarrollo y el cliente.

Es por ello que iniciando el proyecto se deben definir todos los requisitos, para luego invertir el esfuerzo en manejar los cambios que se presenten y así minimizar las posibilidades de error. XP tiene como base la simplicidad y como objetivo la satisfacción del cliente.

Características de XP

- Metodología basada en prueba y error para obtener un software que funcione realmente.
- Fundamentada en principios.
- Está orientada hacia quien produce y usa software (el cliente participa muy activamente).
- Reduce el coste del cambio en todas las etapas del ciclo de vida del sistema.
- Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo.
- Cliente bien definido.
- Los requisitos pueden cambiar.
- Grupo pequeño y muy integrado (2-12 personas).
- Equipo con formación elevada y capacidad de aprender

La programación extrema optimiza los tiempos y se adapta al desarrollo de sistemas grandes y pequeños sin mayor documentación, el código es claro y simple, así mismo complementa los conocimientos entre los miembros del equipo, gracias a la programación en parejas. Sin embargo, una desventaja de esta

metodología ágil, es que luego de cada entrega el sistema puede ir creciendo según sean las peticiones del cliente.

Fases de la Programación Extrema

Planeación

La Metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores. El proyecto comienza recopilando las historias de usuarios, las que constituyen a los tradicionales casos de uso. Una vez obtenidas estas historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

Conceptos básicos en la planificación

Las Historias de Usuarios, las cuales son descritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar.

El Plan de Entregas (*Release Plan*), establece que las historias de usuarios serán agrupadas para conformar una entrega y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto.

Plan de Iteraciones (*Iteration Plan*), las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.

Reuniones Diarias de Seguimiento (*Stand – Up Meeting*), el objetivo es mantener la comunicación entre el equipo y compartir problemas y soluciones.

Diseño

La Metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

Simplicidad, Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione.

Soluciones “Spike”, Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados “Spike”), para explorar diferentes soluciones.

Recodificación (“Refactoring”), Consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de crearlo más simple, conciso y entendible. Las metodologías de XP sugieren re codificar cada vez que sea necesario.

Metáforas, XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, así como guiar la estructura del mismo. Una buena metáfora debe ser fácil de comprender para el cliente y a su vez debe tener suficiente contenido como para que sirva de guía a la arquitectura del proyecto.

Codificación

Disponibilidad del Cliente, Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. El involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP. Al comienzo del proyecto, el cliente debe proporcionar las historias de usuarios. Pero, dado que estas historias son expresamente cortas y de “alto nivel”, no contienen los detalles necesarios para realizar el desarrollo del código. Estos detalles deben ser proporcionados por el cliente, y discutidos con los desarrolladores, durante la etapa de desarrollo.

Uso de Estándares, XP promueve la programación basada en estándares, de manera que sea fácilmente entendible por todo el equipo, y que facilite la re codificación.

Programación Dirigida por las Pruebas (“Test-Driven Programming”), En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de los test, es usualmente realizada sobre el final del proyecto, o

el final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, primero se escriben los test que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas. Las pruebas a las que se refiere esta práctica son las unitarias, realizadas por los desarrolladores. La definición de estos test al comienzo, condiciona o “dirige” el desarrollo.

Programación en Pares, XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien parece que esta práctica duplica el tiempo asignado al proyecto (y por ende, los costos en recursos humanos), al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

Integraciones Permanentes: todos los desarrolladores necesitan trabajar siempre con la “última versión”. Realizar cambios o mejoras sobre versiones antiguas causan graves problemas, y retrasan al proyecto. Es por eso que XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. Idealmente, todos los días deben existir nuevas versiones publicadas. Para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.

Propiedad Colectiva del Código, En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, una pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o re codificar.

Ritmo Sostenido, La Metodología XP indica que debe llevarse un ritmo sostenido de trabajo. El concepto que se desea establecer con esta práctica es planificar el trabajo de forma a mantener un ritmo constante y razonable, sin sobrecargar al equipo.

Pruebas

Pruebas Unitarias: todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“*Test-Driven Programming*”). El hecho de que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código.

Detección y Corrección de Errores: cuando se encuentra un error ("*Bug*"), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.

Pruebas de Aceptación: son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación.

1.4 Conclusiones parciales

Como resultado de la investigación realizada en este capítulo se puede concluir que:

- El estudio de los referentes teóricos permitió identificar que ninguna de las herramientas estudiadas brinda una solución al problema planteado.
- El estudio de las herramientas permitió detectar las funcionalidades básicas del módulo.
- Se identificaron las principales herramientas y tecnologías necesarias para el desarrollo del módulo

Capítulo 2: Análisis y diseño del módulo para la generación de reportes y evaluación del desempeño.

En este capítulo se presenta una propuesta general del módulo para la generación de reportes y evaluación del desempeño. Los requisitos funcionales (RF) y los requisitos no funcionales del módulo (RNF) identificados a partir del modelo conceptual construido, que generan los casos de uso del mismo (CU).

2.1 Descripción de la propuesta de solución

El módulo para la generación de reportes y evaluación del desempeño en portales web debe ser capaz de realizar generación de reportes en formato texto en los portales web desarrollados en Drupal; generando reportes de determinados rangos de fecha, basado en diferentes criterios tales como: tipo de publicación, idioma y editor que realiza cada publicación. Permitiendo la personalización de reportes a través de URL, publicación, idioma y editor. Para brindar como resultado un reporte con los nodos que posean duplicidad parcial o total.

2.2 Modelo de dominio

El modelo de dominio es una representación de clases conceptuales significativas en un problema, no un diagrama que describe clases de software.

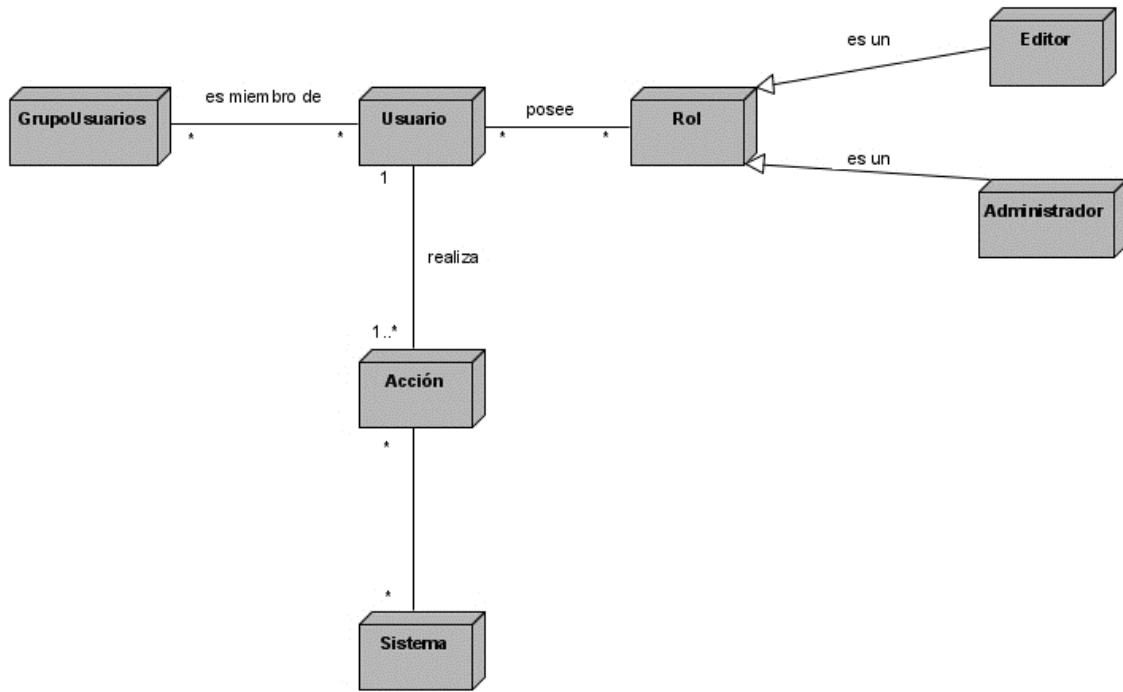


Figura1. Modelo de dominio

Tabla 1. Definición de conceptos del dominio

Concepto	Descripción
Acción	Acción que realiza un usuario sobre el sistema.
Sistema	Conjunto de partes que funcionan relacionándose entre sí con un objetivo preciso.
Grupo de usuarios	Grupo conformado por usuarios que poseen los mismos permisos.
Rol	Función que juega un usuario dentro del sistema.
Administrador	Permite realizar todas las acciones del sistema.
Editor	Rol con nivel de acceso limitado.
Usuario	Usuario que interactúa con el sistema.

Fuente: Elaboración propia

El módulo presenta un sistema jerárquico que permite el acceso a sus funcionalidades solamente a los usuarios autorizados:

- Editor: puede consultar la generación de reportes y evaluar el desempeño que devuelve el módulo en correspondencia con los parámetros establecidos por el administrador.

- Administrador: puede acceder a todas las funcionalidades del módulo, incluidas la selección de parámetros para los reportes y la asignación de permisos.

2.3 Requisitos de software

Los requisitos de software son las cualidades que debe tener el producto de software. Ellos se dividen en dos grupos: los requisitos funcionales y los no funcionales. Los primeros constituyen especificaciones detalladas de las funciones con que debe cumplir el producto y, los no funcionales, que son características del producto, pero no forman parte de sus funcionalidades obligatorias.

2.3.1 Requisitos funcionales

Tabla 2. Requisitos funcionales

No.	Requisito	Descripción	Complejidad	Prioridad
RF1	Configurar parámetros del reporte.	El módulo debe ser capaz de configurarse para adaptar la generación de reportes según características temporales o permanentes dispuestas por el administrador	Media	Media
RF2	Realizar reporte	El módulo debe ser capaz de generar un reporte general de todo el contenido publicado en el sitio	Alta	Alta
RF3	Realizar reporte por rango de fechas.	El módulo debe ser capaz de generar un reporte a partir de una fecha inicial y una final	Alta	Alta
RF4	Realizar reporte por idioma.	El módulo debe ser capaz de generar un reporte dado uno o varios idiomas	Alta	Alta
RF5	Realizar reporte por editor.	El módulo debe ser capaz de generar un reporte dado uno o varios editores	Alta	Alta

RF6	Realizar reporte por tipo de publicación.	El módulo debe ser capaz de generar un reporte dado uno o varios tipos de publicaciones	Alta	Alta
RF7	Asignar permisos.	El módulo debe ser capaz asignarle permisos específicos a cada usuario	Media	Media

Fuente: Elaboración propia

2.3.2 Requisitos no funcionales

➤ Soporte

- El módulo debe correr sobre los gestores de bases de datos *PHPMy Admin 5.1* y *MySQL 5.7.31*.
- Disponer de un manual de ayuda que describe como se realiza el proceso de instalación.
- El módulo debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.

➤ Usabilidad

- Debe poseer una interfaz intuitiva y agradable al usuario usando colores en la gama de verde y una letra Arial 24.
- El módulo podrá ser utilizado por personas con pocos conocimientos de informática.
- Internacionalización.

➤ Seguridad

- Se define el acceso al módulo y sus funcionalidades mediante la asignación de permisos por roles de usuarios.
- La carpeta donde se encuentre el portal solo tendrá permiso de lectura.

➤ **Fiabilidad**

- Los datos obtenidos de los reportes deben estar en correspondencia con los parámetros introducidos a la hora de las solicitudes.

➤ **Apariencia o interfaz externa**

- El portal debe ser compatible con los navegadores Chrome, Firefox, Safari, Opera y Microsoft Edge.
- El sistema debe poseer un diseño web adaptable.

➤ **Rendimiento**

- El sistema debe tener un tiempo de respuesta de 1000ms

2.4 Definición de casos de uso

A continuación, se definen los casos de uso que se corresponden con los requisitos identificados, los cuales serán una de las bases durante las fases de análisis, diseño e implementación del módulo.

Tabla 3. Casos de uso del sistema

CU	Referencia
CU1: Asignar permisos	RF7.
CU2: Configurar parámetros de reporte	RF1
CU3: Realizar reporte	RF2

CU4: Realizar reporte específico	RF3, RF4, RF5, RF6
---	--------------------

Fuente: Elaboración propia

2.4.1 Diagrama de casos de uso

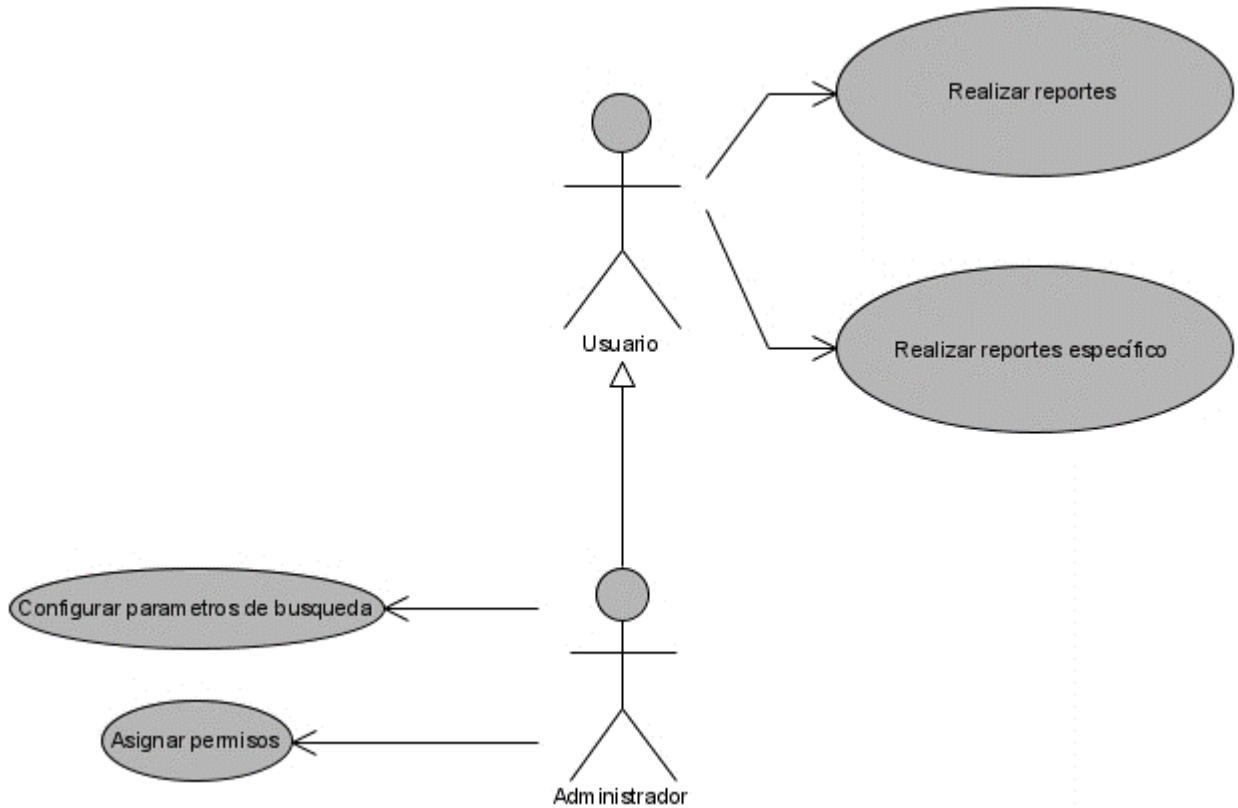


Figura 2. Diagrama de casos de uso

2.5 Arquitectura de Drupal

Drupal posee una arquitectura estructurada en diferentes capas, por lo que para el desarrollo del módulo se propone el trabajo con la arquitectura n-capas, específicamente 5 capas que son descritas a continuación (Vandyk, 2011):

Plantillas (templates): Esta capa establece la apariencia gráfica que se le muestra al usuario. Esta separación entre la información y los estilos permite cambiar la apariencia del portal web sin necesidad de modificar los contenidos.

Vistas (views): Es la capa encargada de mostrar en los temas los cambios realizados a través de los módulos.

Entidades (entities): Representa las entidades, que engloban los nodos, los usuarios, las taxonomías y los comentarios. Esta nueva estructura permite que sea posible añadirle campos a todo aquello que sea una entidad.

Módulos (*modules*): Engloba los elementos que operan sobre los nodos otorgando funcionalidades a Drupal. Permiten incrementar sus capacidades o adaptarlas a las necesidades de cada portal web.

Base de Datos (*database*): Esta capa es la encargada de gestionar el acceso a la información almacenada referente al funcionamiento del sistema y a los contenidos que serán mostrados a través del tema activo.

El módulo para la realización de reporte se encuentra ubicado dentro de la capa Módulos de la arquitectura de Drupal 9, pero se interrelaciona con las restantes capas. El módulo hace uso de la capa Base de Datos (BD), específicamente de su API para realizar consultas que sean independientes al gestor utilizado. Además de su interacción con la capa Entidades cuando crea sus permisos de acceso.

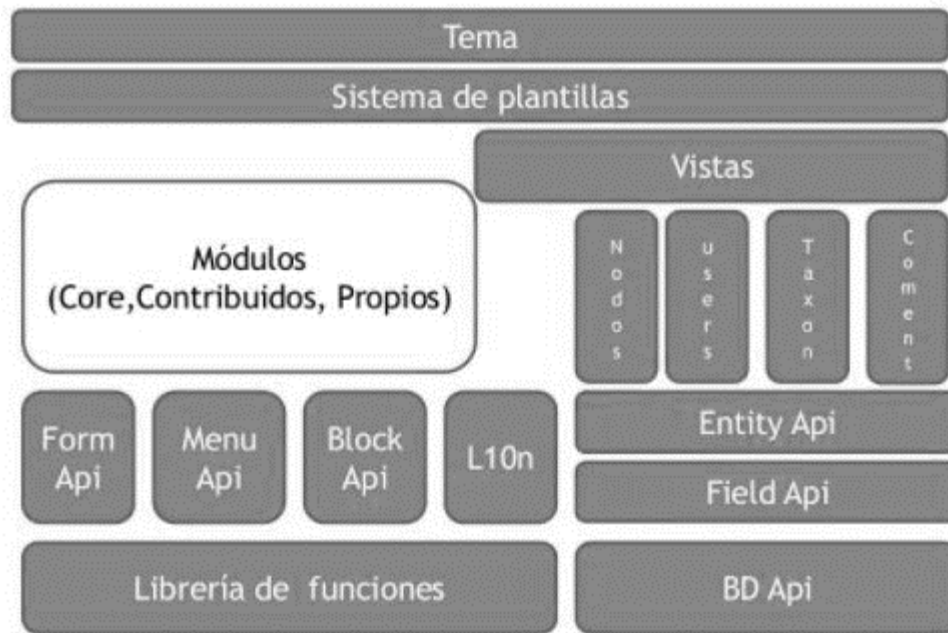


Figura 3. Arquitectura de Drupal 9. Estructura en 5 capas (Vandyk, 2011)

2.6 Patrones de diseño en Drupal

Drupal hace uso de patrones de diseño como: *singleton* (sencillo o instancia única), *decorator* (decorador), *observer* (observador), *bridge* (puente), *chain of responsibility* (cadena de responsabilidad) y *command* (comando). Estos permiten diseñar sistemas seguros y que a su vez cumplan con los estándares de diseño establecidos por normas internacionales para el desarrollo de aplicaciones web. Los patrones usados para el desarrollo del módulo se evidencian de la forma siguiente:

Observer (Observador)

Los módulos que implementan un *hook* de inserción o actualización de una entidad, son declarados como observadores de las entidades con las que interactúan. En el caso del módulo desarrollado, el patrón se evidencia en la inserción de los permisos *Administrator Access to report* y *Common user access to report*.

Bridge (Puente):

La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón puente. El módulo implementado es programado de manera que sea independiente del motor de base de datos que utiliza el sistema. Esto se logra por la capa de abstracción de base de datos, sobre la que se pueden desarrollar consultas siguiendo la API definida.

Este patrón se evidencia en el módulo para la generación de reporte y evaluación del desempeño de los editores que fue programado logrando independencia del motor de BD que se utilice. Esto permite conectarse a la BD y realizar consultas abstrayéndose del SGBD que se emplee para manejar los datos almacenados.

Chain of responsibility (Cadena de responsabilidades):

El sistema de menú de Drupal es la evidencia del patrón cadena de responsabilidades. En cada petición de una página, el sistema de menú de Drupal determina si hay algún módulo para responder la petición, si el usuario tiene acceso al recurso solicitado y que función se debe llamar para procesar la petición. En este proceso se transmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos.

2.7 Modelos de diseño

2.7.1 Diagramas de clases del diseño

Un diagrama de clases de diseño (DCD) representa las especificaciones de las clases e interfaces *software* en una aplicación. Las clases de diseño de los DCD muestran las definiciones de las clases *software* en lugar de los conceptos del mundo real (Larman, 2003).

A continuación, se observa el diagrama de clases del diseño del CU # 3: Realizar reportes.

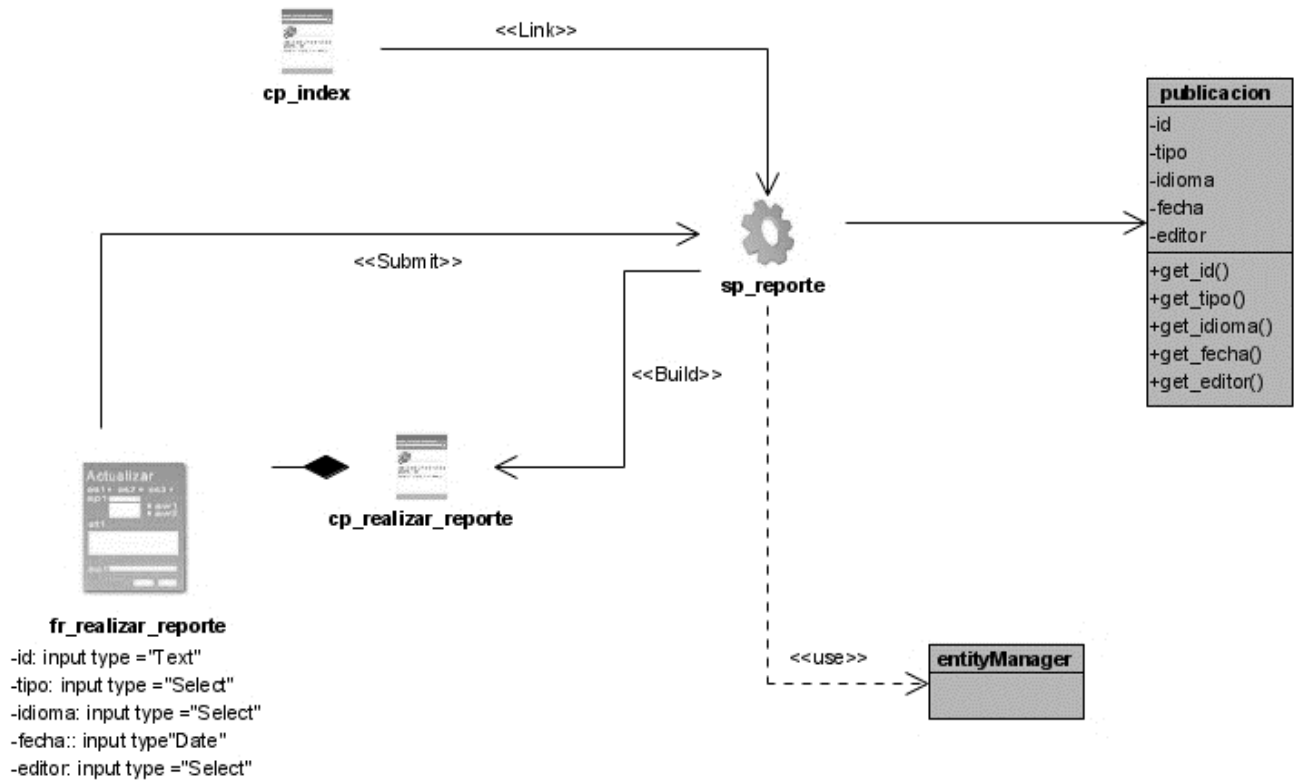


Figura 4. Diagrama de clases del diseño: Realizar reporte

2.7.2 Diagramas de secuencia del diseño (DS)

Ilustran las interacciones en un tipo de formato con aspecto de una valla, en el que cada objeto nuevo se añade a la derecha, mostrando claramente la secuencia y ordenación en el tiempo de los mensajes con una notación simple (Larman, 2003). A continuación, se observa el diagrama de secuencia del caso de uso: Realizar reporte

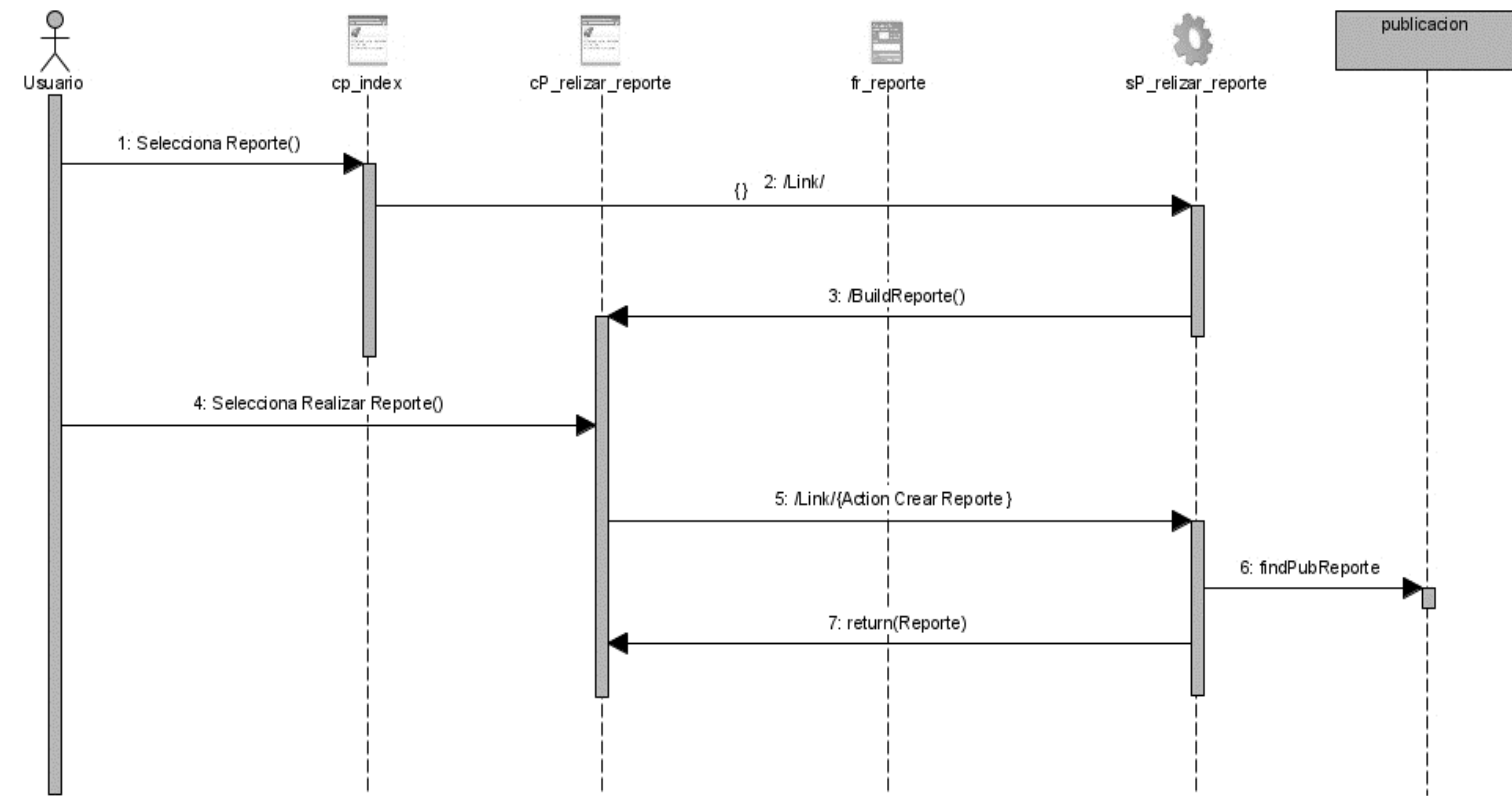


Figura 5. Diagrama de secuencia del caso de uso: Realizar reporte

2.8 Conclusiones parciales

El desarrollo de este capítulo permitió arribar a las siguientes conclusiones:

- Los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, sirvieron de guía para desarrollar las distintas funcionalidades y de este modo satisfacer las necesidades detectadas.

- Los artefactos generados según la metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.

Capítulo 3: Implementación y prueba del módulo para la generación de reportes y evaluación del desempeño.

En el presente capítulo se presentan los componentes y los estándares de codificación que sustentan la implementación del módulo para la generación de reportes y evaluación del desempeño en portales web. Se describe y fundamenta el proceso de validación de la solución implementada, mediante la utilización de los casos de pruebas y la herramienta de seguridad.

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 del código (fuente, binario o ejecutable), que permite 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 (Rivera, 2008).

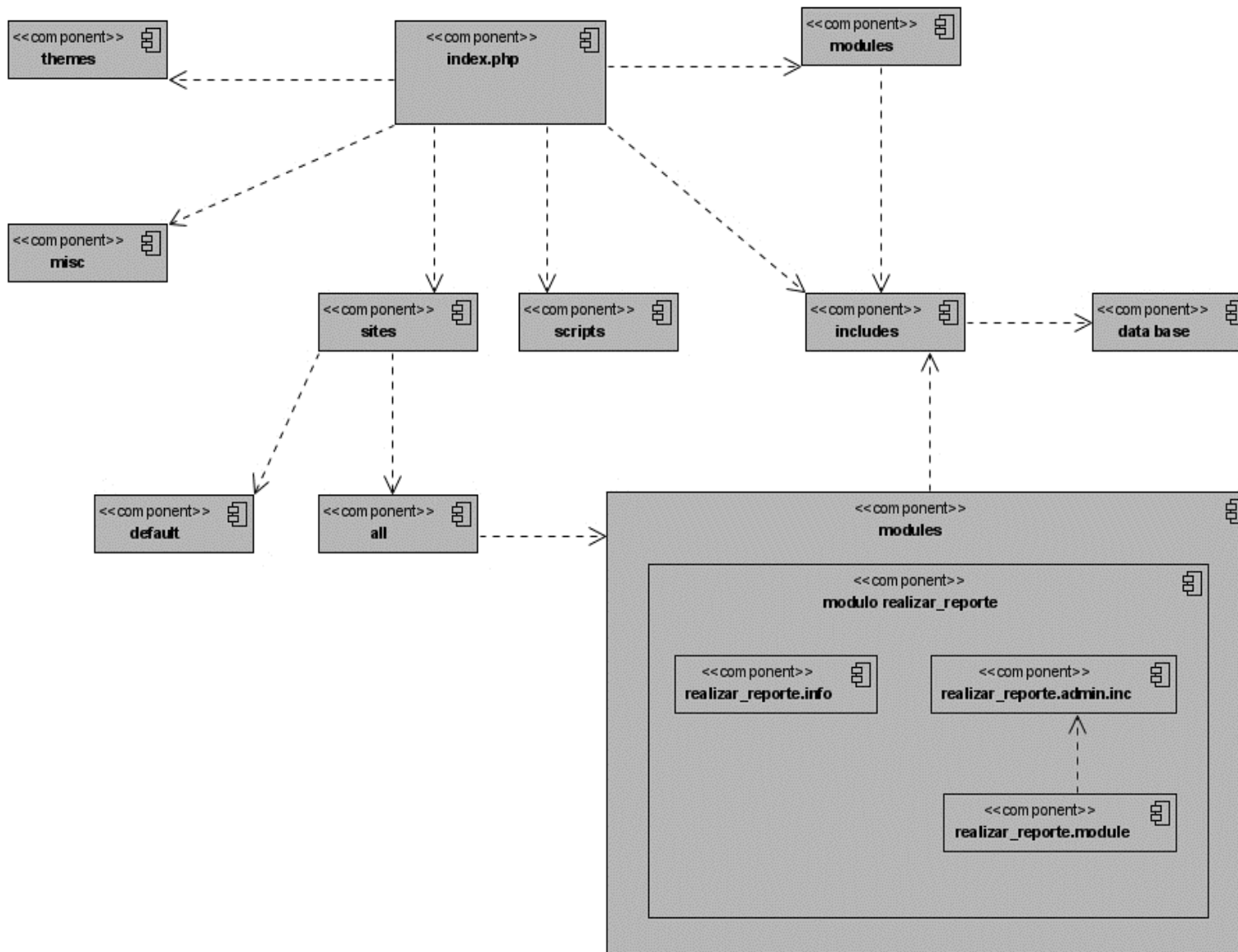


Figura 6. Diagrama de componentes del módulo para la detección de contenido duplicado en portales web

La tabla que se muestra a continuación describe cada uno de los componentes representados en el diagrama.

Tabla 3. Descripción de componentes

Componentes	Descripción
<i>index.php</i>	Este componente es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS Drupal.
<i>themes</i>	En este componente se incluyen los temas que vienen con la distribución de Drupal.
<i>misc</i>	Este componente incluye archivos JavaScript e imágenes requeridas por el sistema.
<i>includes</i>	Este componente tiene un conjunto de ficheros indispensables para el funcionamiento del CMS Drupal.
<i>scripts</i>	Contiene utilidades adicionales que no utiliza Drupal directamente, pero que se pueden utilizar desde la línea de comandos de shell. Por ejemplo, el script password-hash.sh permite obtener una contraseña codificada a partir de la contraseña original.
<i>modules</i>	En este componente se encuentran los módulos básicos del núcleo de Drupal.
<i>sites</i>	La carpeta sites contiene los extras y modificaciones que se añaden a la distribución original. En esta se encuentran los módulos adicionales creados, añadidos o descargados del repositorio de módulos de Drupal, colocados en <i>sites/all/modules</i> . La carpeta incluye tras la instalación el archivo de configuración del portal (<i>settings.php</i>).

3.1.1 Modelo 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 (Jacobson, 2000). En la siguiente figura se muestra el diagrama de despliegue correspondiente a la solución propuesta.

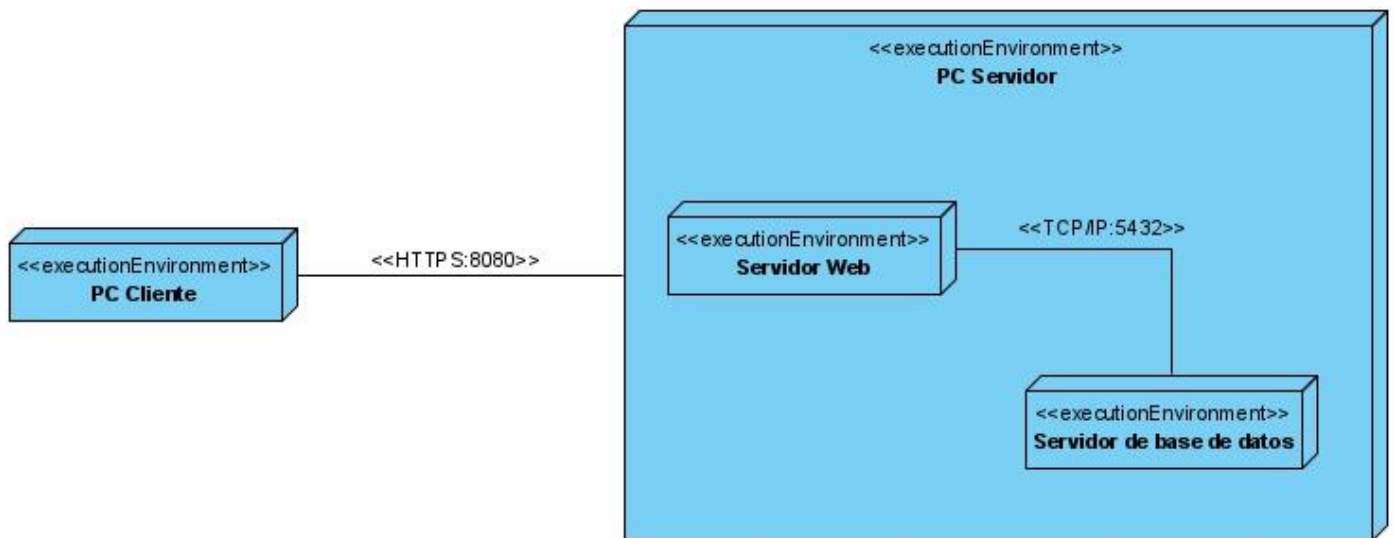


Figura 7. Diagrama de despliegue.

PC: Estación de trabajo que presenta un navegador web para conectarse a los portales web hospedados en el servidor web utilizando el protocolo de comunicación HTTPS, que además tienen instalado el módulo para la detección de contenido web duplicado.

Servidor web: Estación de trabajo que hospeda el código fuente del módulo y que brinda a los usuarios las interfaces del mismo para realizar los procesos definidos para cada uno de los roles. Esta estación se comunica con el servidor de BD donde se almacenan los datos de los portales realizando la comunicación mediante el protocolo TCP/IP.

Servidor de BD: Servidor encargado del almacenamiento de los datos de los portales. Se comunica con el servidor web.

3.2 Estándares de codificación

Los estándares de codificación permiten que el código generado durante la implementación de un proyecto se pueda leer con una mayor facilidad. Además, permiten la comprensión y modificación del código independientemente de su autor permitiendo además que el mismo presente una alta calidad y no presente errores. Drupal brinda a sus desarrolladores las siguientes normas de codificación.

Indentación

La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos.

En Drupal se debe indentar con 2 espacios, nunca con tabuladores. Además, no se debe dejar espacios en blanco al final de cada línea.

Etiquetas de apertura y cierre de PHP

Cuando se escribe en PHP, siempre se deben utilizar las etiquetas `<?php` y `?>`, y en ningún caso la versión corta `<?` y `?>`. En general se omite la etiqueta de cierre de PHP (`?>`) al final de los archivos. `module` y `.inc`. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre `?>`), que serían identificados como salida HTML y podrían provocar un error muy típico, "Cannot modify header information - headers already sent by...".

Por tanto, la etiqueta de cierre final del archivo (`?>`) es opcional en Drupal.

No hay que confundir esto con el uso normal del lenguaje PHP en archivos que también contienen HTML (como por ejemplo los archivos de plantilla `.tpl.php`), donde cada fragmento de PHP debe llevar sus correspondientes etiquetas de apertura y cierre, para diferenciarlo del código HTML.

Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, `$numero = 3`, en lugar de `$numero=3`. Esto se aplica a operadores como `+`, `-`, `*`, `/`, `=`, `==`, `!=`, `>`, `<`, `.` (Concatenación de cadenas), `.=`, `+=`, `-=`, etc.

Los operadores unarios como ++, -- no deben tener separación. Por ejemplo, \$numero++.

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, "<h1>\$title</h1>". También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.

Uso de punto y coma (;) en código PHP

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

- Correcto: <?php print \$title; ?>

- Incorrecto: <?php print \$title ?>

Estructuras de control

Con respecto a las estructuras de control, hay que tener en cuenta las siguientes normas:

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

Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo o tema, para evitar así duplicidad de funciones.

En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.

Ejemplo:

```
function forum_help ($path, $arg) {
```

En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros, como se muestra en el siguiente ejemplo:

```
$field = field_info_instance('node', 'taxonomy_forums', $node->type);
```

Como excepción, es posible usar más de un espacio antes de una asignación (=) para mejorar la presentación, cuando se estén realizando varias asignaciones en bloque:

```
$numero1 = foo($a, $type);
```

```
$primer_valor = foo2($b);
```

```
$i = foo3();
```

Arrays

Los valores dentro de un array (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados.

Cuando la línea de declaración del array supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

```
$vector1 = array(1, 2, 'clave' => 'valor');

$vector2 = array(

'forum' => 'foro1',

'template' => 'forums',

'arguments' => array('tid' => NULL, 'topics' => NULL),

'size' => 128,

);
```

Nombres de módulos

El nombre de un módulo nunca debería incluir guiones bajos, aunque se componga de varias palabras. De esta forma es más fácil identificar el módulo al que pertenece una función, ya que el prefijo o nombre del módulo es todo aquello que esté antes del primer guión bajo. Por ejemplo, es aconsejable utilizar mimodulo en lugar de mi_modulo. Esta regla no es obligatoria.

Nombres de archivos

Los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación, que tendrán extensión .txt y el nombre en mayúsculas. Por ejemplo, README.txt, INSTALL.txt, etc.

Comentar el código

En este apartado se debe diferenciar entre los comentarios para aclarar determinados fragmentos de código, que se insertan en cualquier punto del mismo, y los comentarios de documentación.

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.

En el primero de los casos se pueden utilizar las etiquetas /* */ para comentarios en varias líneas y // para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, ésta deberá escribirse en mayúsculas (por ejemplo, TRUE o FALSE).

3.3 Validación de la propuesta de solución

Las pruebas de software son procedimientos realizados para verificar la calidad de un software y pueden ser aplicadas periódicamente. Estas tienen como objetivo identificar posibles errores en la aplicación. Existen varias estrategias de pruebas que suelen ser utilizadas, dentro de las que se pueden mencionar:

Pruebas de Caja Blanca

Dirigidas a las funciones internas del sistema. Consisten en una verificación técnica del software que los desarrolladores pueden usar para examinar si su código trabaja como es esperado. Se realizan probando la lógica de la aplicación y comprobando el estado del software en varios puntos, verificando que el resultado de dicho estado coincida con lo esperado.

Pruebas de Caja Negra

Se desarrollan sobre la interfaz visual del software y se centran en los requisitos funcionales de la aplicación, sin adentrarse en el funcionamiento interno de la aplicación. A través de su realización se pueden encontrar errores de interfaz, funciones incorrectas, errores de salida y problemas con el acceso a datos.

Pruebas de Sistema

Las pruebas de sistema tienen como objetivo verificar el sistema de software para comprobar si el mismo cumple con sus requisitos. Cuenta con distintos tipos de pruebas, las cuales algunas son funcionales, de usabilidad, de rendimiento, de seguridad, entre otras.

3.3.1 Funcionales

Las pruebas de funcionalidad se realizan con el objetivo de garantizar el funcionamiento adecuado de los requisitos funcionales. Para su realización es necesario diseñar un conjunto de casos de prueba que son

utilizados para someter a las diferentes funcionalidades del módulo y de este modo verificar su correcta ejecución.

3.3.2 Integración

Las pruebas de integración se realizan para confirmar que el módulo se integra de forma correcta con el sistema. Para la realización de ellas se probó el módulo en el sitio web “Fidel, Soldado de las ideas” desarrollado utilizando como gestor de bases de datos MySQL. El desarrollo de las pruebas garantizó que el módulo presentaba un correcto funcionamiento en ambos portales y que las funcionalidades de los portales no eran afectadas.

3.3.3 Seguridad

Las pruebas de seguridad certifican que los datos o el sistema solamente son accedidos por los actores definidos y según los niveles de acceso. Tienen como objetivo evaluar la confidencialidad, integridad y disponibilidad de los datos. Estas aumentan la credibilidad e imagen corporativa, así como, el control de la información sensible. Para la realización de las pruebas de seguridad se utilizará la herramienta Acunetix, que divide la criticidad de los errores en 4 niveles (Alto, Medio, Bajo e Informativo).

3.3.4 Casos de pruebas

Los casos de pruebas definen un conjunto específico de entradas de pruebas, ejecución de condiciones y resultados esperados. Un buen diseño de los casos de prueba es fundamental para el éxito de las pruebas. Un caso de prueba especifica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. El caso de prueba simula las interacciones del actor con el sistema para verificar que el sistema hace lo que se espera de él. Así, el principal artefacto para obtener pruebas del sistema son los requisitos funcionales.

Caso de prueba: Realizar reporte

Descripción general:

El sistema muestra un bloque con las opciones realizar reporte y realizar reporte específico. Luego de seleccionarla se muestra un listado con todos los documentos que se utilizan. Donde se puede seleccionar cualquiera de ellos.

Condiciones de Ejecución:

El actor debe estar autenticado en el sistema y, además debe tener permisos para realizar las mencionadas operaciones. Para esto debe seleccionarse la opción realizar reporte.

Tabla 5. Caso de prueba. Realizar reporte

Escenarios	Descripción de la funcionalidad	Respuesta del sistema	Flujo central
SC 1: Realizar reporte	EC 1.1: El usuario selecciona la opción realizar reporte	El sistema muestra un reporte con los datos de las publicaciones del último mes	-Se busca la sección deseada para el reporte -El usuario exporta el reporte

Fuente: Elaboración propia

Caso de prueba: Asignar permisos

Descripción general:

El sistema muestra un bloque con las opciones configurar parametros del reporte y asignar permisos. Luego de seleccionarla se muestra un listado con todos los usuarios que se encuentran creados en el sitio. Donde se puede seleccionar cualquiera de ellos.

Condiciones de Ejecución:

El actor debe estar autenticado en el sistema y, además debe ser administrador para realizar las mencionadas operaciones. Para esto debe seleccionarse la opción Asignar permisos.

Tabla 6. Caso de prueba Asignar permisos

Escenarios	Descripción de la funcionalidad	Respuesta del sistema	del Flujo central
SC 1: Asignar permisos	EC 1.1: El usuario selecciona la opción Asignar permisos	El sistema muestra un listado con los usuarios creados en el sitio y los permisos que presentan.	1-El usuario accede al menú asignar permisos. 2-El sistema muestra un listado con todos los usuarios 3-El usuario selecciona el permiso a asignar al usuario marcado. 4-El sistema guarda los cambios y muestra un mensaje de notificación

Fuente: Elaboración propia

3.3.5 Pruebas de rendimiento

Como parte de la ejecución de las pruebas de rendimiento, con el objetivo de identificar en qué medida satisface el módulo las funcionalidades implementadas se planea realizar tres iteraciones de pruebas. Para esto se utilizará la herramienta JMeter en la interfaz donde aparece el módulo, para generar diferentes cargas (usuarios/tráfico) recurrentes para crear un entorno de prueba de la vida real y reconocer cuellos de botella. Estas se realizan generalmente para observar el comportamiento de una aplicación bajo una

cantidad esperada de peticiones y muestra los tiempos de respuesta de todas las transacciones importantes de la aplicación.

3.4 Conclusiones parciales

En el capítulo se realizó el análisis de los tipos de pruebas a realizar al módulo. Se confeccionaron casos de prueba con el fin de comprobar el funcionamiento del mismo. Obteniendo como resultado las siguientes conclusiones:

- La elaboración de los diagramas y la descripción de los componentes que lo conforman, permitió una mejor comprensión de la estructura del módulo a implementar.
- Aplicar los estándares de codificación permitió obtener en el módulo para la generación de reportes y evaluación del desempeño en el sitio: **“Fidel, Soldado de las ideas”** un código legible, estándar y fácil de comprender.

Conclusiones

Con la investigación realizada se logró el desarrollo del módulo para la generación de reportes y evaluación del desempeño en el sitio web: “**Fidel, Soldado de las ideas**”. El mismo perseguía resolver las dificultades presentadas por la Oficina de Información del Comité Central del Partido Comunista de Cuba a la hora de realizar reportes y evaluar el desempeño de los editores. Los objetivos propuestos fueron cumplidos con satisfacción, generando cada uno de ellos los siguientes resultados:

- La elaboración del marco teórico permitió identificar la necesidad de crear el módulo para la generación de reportes y evaluación del desempeño del sitio: “**Fidel, Soldado de las ideas**”, determinando la selección de las herramientas, metodología y tecnologías factibles a utilizar en el desarrollo del módulo.
- Los artefactos generados durante el flujo de trabajo de análisis y diseño sirvieron para conformar la primera visión de la implementación del sistema, defendiéndose durante la misma una estrategia para lograr una construcción flexible y robusta a través de la utilización de una arquitectura afín, patrones de diseño bien establecidos y estándares de codificación.

Recomendaciones

A modo de recomendación se propone:

- Continuar desarrollando funcionalidades para mejorar el trabajo de los editores del sitio web: "Fidel, Soldado de las Ideas" versión 3.
- Realizar las pruebas planteadas en el plan de pruebas para comprobar el correcto funcionamiento del módulo.

Bibliografía

Arquitectura y Diseño de Software. Diseño de aplicaciones web con extensiones de UML. (s.f.).

Arredondo Guzmán, M. (2020). Taller: Implementación de sitios web con Drupal. Obtenido de https://www.visibilidadweb.unam.mx/sites/default/files/docs/Taller_Drupal.pdf

Atenea Tech SLNE. (23 de junio de 2020). Recuperado el 2021, de <https://ateneatech.com/blog/crear-un-modulo-para-drupal-8>

Aubry, C. (2019). HTML 5 y CSS 3: revolucione el diseño de sus sitios web (4ta ed). España: Eni Ediciones. Obtenido de <http://desarrolloweb.com>

Battaglia, N., Roxana, M., Neil, C., & De Vincenz, M. (2017). Una propuesta de evaluación de herramientas CASE para la enseñanza. La Plata.

Bron Fonseca, B., Mestre, K., & Simón Grass, W. (2019). *Sistema informático para la gestión de reportes de incidencias de mantenimiento en la Facultad de Ciencias y Tecnologías Computacionales* . Universidad de las Ciencias Informáticas. La Habana: Ediciones Futuro .

CALISOFT. (2017). Requisitos para la calidad de los sistemas informáticos y productos de software: SubComité 7 Ingeniería de Software y Sistemas .

Coaquira Tuco, C. (2018). Apuntes Universitarios. Revista de Investigación. 8(3). Recuperado el 2021, de <https://www.redalyc.org/articulo.oa?id=467657107004>

Documentation. (2021). (Drupal's online documentation is © 2000-2021) Obtenido de <https://www.drupal.org/docs/creating-custom-modules/adding-stylesheets-css-and-javascript-js-to-a-drupal-module>

Drupal Association. (2020). *Drupal*. Recuperado el 21 de junio de 2021, de <http://www.drupal.org/>

eSinergia. (2019). Recuperado el 6 de Agosto de 2021, de <https://esinergia.co/>

Fernández, F. (2021). Aplicaciones web con php. España: Rama Editorial.

- Gaitán Quintanilla, R. (2015). *Diseño e implementación de un software*. Santa Tecla. El Salvador: Escuela Especializada en Ingeniería (ITCA-FEPADE).
- García Recio, J. (2021). HTML5, CSS3 y jQuery. España: RA-MA S.A Editorial Y Publicaciones.
- García, A. M. (2015). *Sistema de información basado en reportes para la gestión de proyectos*. Trabajo de Diploma. Título de Máster, Universidad de las Ciencias Infomáticas (UCI). Obtenido de <https://www.researchgate.net/publication/277649761>
- González Santillán, Á., & Hernández Zágada, M. (2014). Diseño y desarrollo reportes en formato PDF a partir de base de datos remotas en tiempo real con php y Mysql. *Revista Iberoamericana de Producción Académica y Gestión Educativa*(1). Obtenido de <https://www.pag.org.mx/index.php/PAG/article/view/86>
- Gutiérrez, D. (2011). Arquitectura de Software. *Estilos Arquitectónicos*. Venezuela.
- Hechavarría. (2002). Aplicación web para la gestión de eventos científicos en la Universidad de las Ciencias Informáticas.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). *Metodología de la Investigación Quinta Edición*. México: Mc Graw-Hill, Companies, Inc. Recuperado el 2021
- Jiménez-Hidalgo, S., Giménez-Toledo, E., & Salvador-Bruna, J. (2008). Los sistemas de gestión editorial como medio de mejora de la calidad y la visibilidad de las revistas científicas. *El profesional de la información*, vol 7(n 3). Recuperado el 6 de Agosto de 2021, de <https://digital.csic.es/bitstream/10261/36347/1/sege.pdf>
- Larman, C. (2003). *UML y patrones: introducción al análisis y diseño orientado a objetos*. Recuperado el 18 de noviembre de 2021, de <<http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDiseno.pdf>>.
- Larman, C. (2005). *Applying UML and patterns: An introduction to object-oriented analysis and iterative development*. España: Icaria Editorial.

- Loaysa-Yupanqui, M. (2014). *La gestión de la información y su relación con la toma de decisiones en la dirección de la infraestructura de la Universidad Nacional José María Aregudas Andahuaylas*. Tesis Doctoral, Perú. Recuperado el 15 de agosto de 2021 de <https://repositorio.unajma.edu.pe/handle/123456789/197>
- Marín, R. (2019). Los gestores de bases de datos más usados en la actualidad. *Revista Digital ENESEM*. Obtenido de <http://www.revistadigital.enesem.es>
- Microsoft. (2021). Visual Studio. Obtenido de <http://www.code.visual.studio.com>
- Modelando Diagrama de Despliegue desde Visual Paradigm for UML v8.0. (s.f.).
- Molina Hernández, Y., Granda Dihigo,, A., & Velázquez Cintra, A. (2019). Los requisitos no funcionales de software. Una estrategia para su desarrollo en el Centro de Informática Médica. *Revista Cubana de Ciencias Informáticas,, vol. 13(2)*. Recuperado el 2021 de julio de 7, de : <http://www.redalyc.org/articulo.oa?id=378362738006>
- Paradigm, V. (2021). Visual Paradigm Online. Obtenido de <http://www.visual-paradigm.com>
- Parra, A. (2018). *JavaScript*. España: Anaya. Obtenido de <http://www.devolver.mozilla.org/>
- Pérez, N. P. (2020). Solución informática para la solución del servidor web durante la migración a código abierto. *Revista Cubana de Ciencias Informáticas*.
- Prats, J., Serrano-Muñoz, J., Prieto, T., & Rico, O. (Mayo de 2009). Bibliotecia. La Biblioteca digital de la UPC. Drupal como gestión de contenidos 2.0. *ResearchGate*. Obtenido de <https://www.researchgate.net/publication/33423052>
- Pressman, R., y Maxim, B. (2015). *Software engineering: a practitioner's approach, eighth edition*. New York: McGraw-Hill Education.
- Ramírez Saldaña, D. (2006). La Importancia de la evaluación del desempeño en la Gerencia de la Ciencia. 7(4). Recuperado el 2021, de <http://www.redalyc.org/revista.oa?id=1815>

- Ramos, D. (9 de Agosto de 2018). *Seed Em*. Recuperado el 28 de junio de 2021, de <https://www.seedem.co/es>
- Rodríguez-Salas, K. (2002). Gestión de la información en las organizaciones. *Bibliotecas*, 10(2), 19-34. Recuperado el 2021
- Rouse, M. (10 de Octubre de 2021). *TechTarget*. Obtenido de <https://www.computerweekly.com/es/definicion/MySQL>
- Salgado-Castillo , J., & Calderón-Pinzón, L. T. (2014). Sistemas de control de gestión y desempeño organizacional: una revisión conceptual. México.
- Sánchez Henríquez, J., & Calderón Calderón , V. (2012). Diseño del proceso de evaluación del desempeño del personal y las principales tendencias que afectan su auditoría. (32), 54-82. Obtenido de www.redalyc.org/revista.oa?id=646
- Sommerville. (2011). *Ingeniería de Software*. México: Pearson Educación.
- Sorroche Salas, A. (2011). *Sistema Automatizado de Gestión de Identidades*. Trabajo de Diploma, Universidad Central "Marta Abreu", Departamento de Automática y Sistemas Computacionales , Las Villas. Recuperado el 17 de agosto de 2021

Anexos

Anexo 1

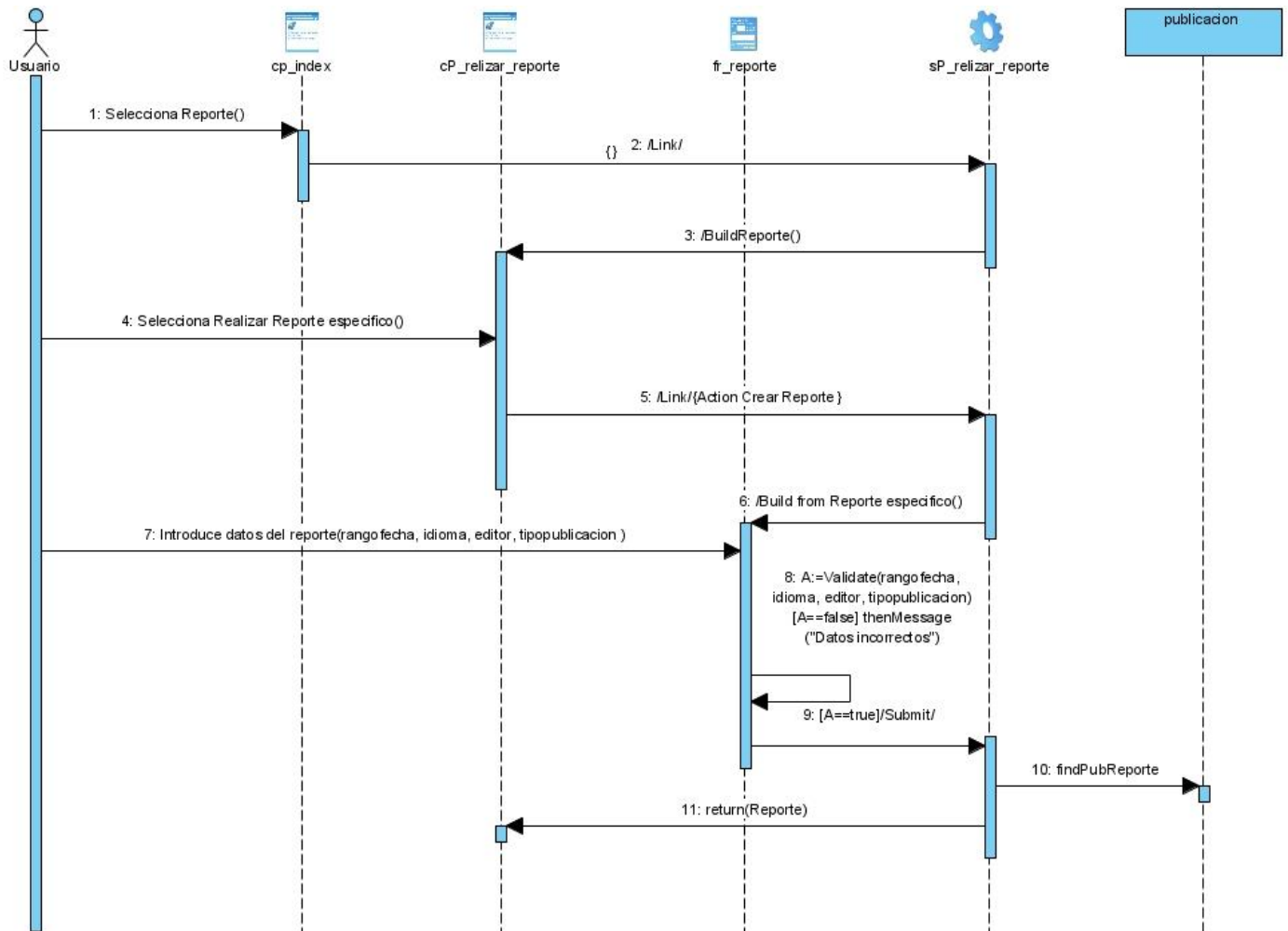


Figura 8. Diagrama de despliegue del caso de uso Realizar reporte específico.