



Universidad de las Ciencias
Informáticas

FACULTAD 1

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS

Portal Web Casa Editora Abril 2.0

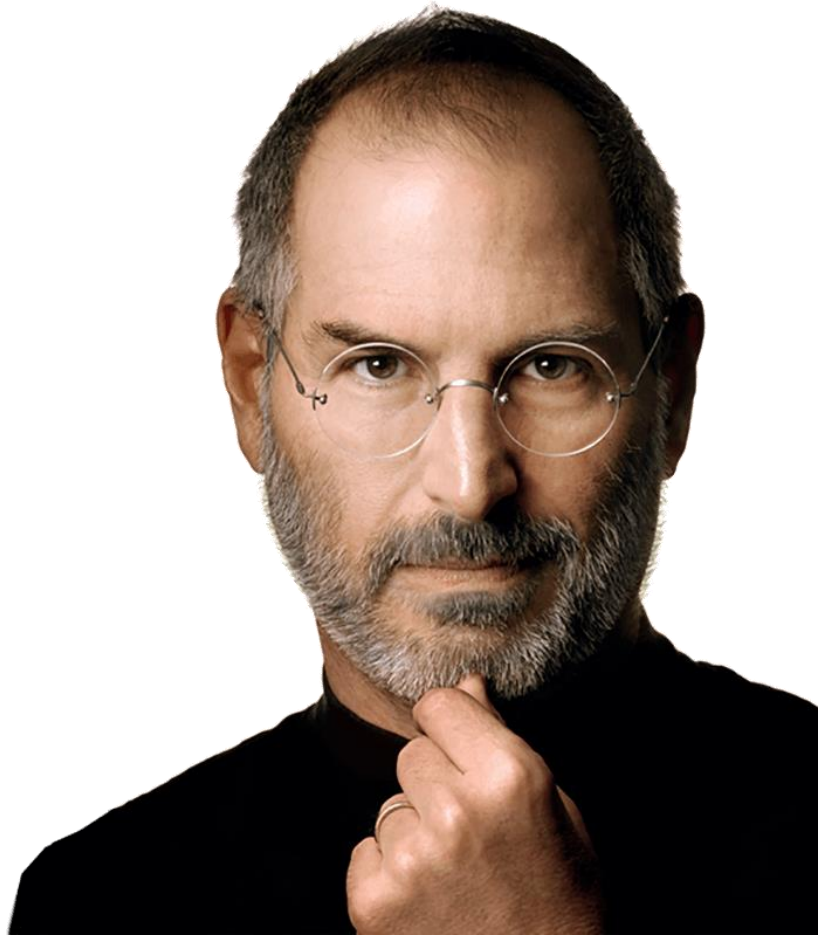
AUTOR: Frank Verdecia Blanco

TUTORES:

Prof. Asistente. MSc. Geidis Sánchez Michel

Especialista. Ing. Nolberto Rojas Pérez

La Habana, diciembre 2021



«Tu trabajo va a llenar gran parte de tu vida, la única manera de estar realmente satisfecho es hacer lo que creas que es un gran trabajo y la única manera de hacerlo es amar lo que haces.»

Steve Jobs

PENSAMIENTO

DECLARACIÓN DE AUTORÍA

Se declara que somos los únicos autores de la presente tesis y se le concede a la Universidad de las Ciencias informáticas los derechos patrimoniales, con carácter exclusivo.

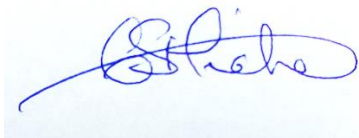
Para que así conste se firma a los 12 días del mes de diciembre del año 2021.

Frank Verdecia Blanco



Firma del Autor

Prof. Asistente. MSc. Geidis Sánchez Michel



Firma del Tutor

Especialista. Ing. Nolberto Rojas Pérez



Firma del Tutor

DECLARACIÓN DE AUTORÍA

DATOS DE CONTACTO

Datos del Autor:

Frank Verdecia Blanco

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: fverdecia@estudiantes.uci.cu

Datos de los Tutores:

Geidis Sánchez Michel

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: gsanchez@uci.cu

Nolberto Rojas Pérez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: nrojas@uci.cu

AGRADECIMIENTOS

DEDICATORIA

RESUMEN

La Casa Editorial Abril en sus páginas abre una amplia gama de temas dirigidos al enriquecimiento de la cultura política e ideológica, resaltando los valores del socialismo, el antiimperialismo, el amor a la obra de la Revolución y a los sentimientos de identidad nacional y de solidaridad. El cuerpo editorial de la Casa Editora Abril ha solicitado a la Universidad de las Ciencias Informáticas específicamente al Centro de Innovación y Desarrollo de Internet la realización de una nueva versión del Portal web para la Casa Editora Abril, debido a que la versión actual fue desarrollada hace varios años con una tecnología obsoleta. La presente investigación expone los resultados y tiene como objetivo desarrollar una nueva versión de dicho Portal web que permita corregir las deficiencias de la versión actual para lograr la correcta divulgación del quehacer diario de la Institución de cara a Internet. Se describen las herramientas utilizadas, el CMS Drupal sobre el que se implementó el sistema y los elementos necesarios para su construcción. Se muestra, además, el resultado de la estrategia de validación que fue aplicada a la propuesta de solución, que permite entregar al cliente un producto eficiente, funcional, seguro, usable y que cumple con sus especificaciones.

Palabras clave: portal web, cultura política, divulgación, tecnología obsoleta.

ÍNDICE

Introducción	6
Capítulo 1. Fundamentación teórica de la investigación.....	10
1.1 Conceptos básicos relacionados con el dominio de la investigación	10
1.1.1 Portales web	10
1.1.2 Evolución de los portales web.....	11
1.1.3 Clasificación de los portales web.....	11
1.2 Análisis de portales web homólogos	12
1.2.1 Tendencias en el diseño web:	12
1.2.2 Portales Internacionales.....	14
1.2.3 Portales Nacionales	14
1.2.4 Resultados de análisis de estudio de portales homólogos.....	15
1.3 Tecnologías y herramientas	17
1.3.1 Metodología de desarrollo de software.....	17
1.3.2 Sistemas de Gestión de Contenido.....	19
1.3.3 Lenguaje para el modelado	20
1.3.4 Lenguajes de programación	21
1.3.5 Servidores de aplicaciones	22
1.3.6 Gestor de base de datos.....	23
1.3.7 Herramientas	24
1.4 Conclusiones parciales	25
Capítulo 2. Análisis y propuesta del Portal Web Casa Editora Abril 2.0	26
2.1 Propuesta de solución.....	26
2.1.1 Vistas de la propuesta solución	27
2.2 Especificación de los requisitos.....	29
2.2.1 Requisitos funcionales.....	29
2.2.2 Requisitos no funcionales.....	33

2.3	Descripción de requisitos de software (Historias de Usuarios)	35
2.3.1	Modelación de las HU.....	35
2.4	Arquitectura.....	39
2.5	Patrones de diseño	41
2.5.1	Características de los Patrones de Diseño.....	41
2.5.2	Tipos de Patrones de Diseño.	42
2.6	Diagrama de clases del diseño con estereotipos web (DCD)	45
2.7	Diagrama de secuencia.....	45
2.8	Conclusiones parciales	46
Capítulo 3.	Implementación y pruebas del Portal Web Casa Editora Abril 2.0	47
3.1	Estándares de codificación de Drupal	47
3.2	Modelo de despliegue	50
3.3	Diagrama de componentes	51
3.4	Estrategia de prueba	52
3.4.1	Pruebas funcionales	53
3.4.2	Pruebas seguridad	56
3.4.3	Pruebas rendimiento (carga y estrés)	58
3.5	Conclusiones parciales	60
CONCLUSIONES		61
RECOMENDACIONES		62
REFERENCIAS BIBLIOGRÁFICAS		63
ANEXOS		67
Anexo 1: Diagrama de clases con estereotipos web para la HU Gestionar Libro		67
Anexo 2: Diagrama de secuencia para la HU Añadir Libro		68
Anexo 3: Diagrama de clases con estereotipos web para la HU Gestionar Revista.		69
Anexo 4: Diagrama de secuencia para la HU Añadir Revista		70
Anexo 5: Vista sección Revistas.....		71
Anexo 6: Vista sección Libros		72

ÍNDICE DE ILUSTRACIONES

Figura 1. Sistema de gestión de contenido (Béjar Cáceres, 2017)	20
Figura 2. Vista sección banner superior y carrusel de imágenes. (Elaboración propia).....	27
Figura 3. Vista sección Últimos Artículos. (Elaboración propia).....	28
Figura 4. Vista sección Artículo destacado. (Elaboración propia)	28
Figura 5. Arquitectura de la aplicación. (Drupal.org, 2016)	40
Figura 6. Diagrama de clases de la Historia de Usuario Gestionar Artículo. (Elaboración propia)	45
Figura 7. Diagrama de secuencia de la Historia de Usuario Añadir Artículo. (Elaboración propia)	46
Figura 8. Estándares de codificación (Drupal.org, 2021).....	48
Figura 9. Estándares de codificación (Drupal.org, 2021).....	49
Figura 10. Modelo de despliegue (Elaboración propia)	50
Figura 11. Diagrama de componentes. (Elaboración propia).....	52
Figura 12. Resultado de iteraciones de las pruebas funcionales. (Elaboración propia).....	56

ÍNDICE DE TABLAS

Tabla 1: Criterio de comparación para el estudio de portales web homólogos. (Elaboración propia)	16
Tabla 2: Arquitectura de información. (Elaboración propia)	26
Tabla 3: Requisitos funcionales. (Elaboración propia)	29
Tabla 4: Historia de Usuario Añadir Artículo. (Elaboración propia)	35
Tabla 5: Historia de Usuario Modificar Artículo. (Elaboración propia)	37
Tabla 6: Historia de Usuario Eliminar Artículo. (Elaboración propia).....	38
Tabla 7: Caso de prueba Añadir Artículo. (Elaboración propia)	53
Tabla 8: Caso de prueba Añadir Artículo. (Elaboración propia)	54
Tabla 9: Resultado pruebas de seguridad. (Elaboración propia).....	57
Tabla 10: Resultado prueba de rendimiento. (Elaboración propia).....	59

INTRODUCCIÓN

Las tecnologías de la información y las comunicaciones (TICs) se han constituido en elementos sustantivos inherentes al desarrollo de todas las esferas de la vida. Es por ello que el crecimiento de las organizaciones es directamente proporcional al buen uso que hagan estas de las TICs para automatizar procesos claves y costosos dentro de su funcionamiento. La incorporación de las TICs permite a una empresa competir eficientemente y tener la información disponible en el lugar y momento en el que se necesita por lo que éstas se han convertido en arma esencial para ofrecer productos y servicios con mayor calidad.

El portal web de la Casa Editora Abril en sus páginas abre un amplio abanico de temas dirigidos al enriquecimiento de la cultura política e ideológica, ética, histórica, científica y ecologista, resaltando los valores del socialismo, el antiimperialismo, el amor a la obra de la Revolución y a los sentimientos de identidad nacional y de solidaridad. Rescatando el cartel político, la intencionalidad gráfica y sobre todo un mensaje que aborda, con el mayor compromiso, los temas de la sociedad cubana y el mundo de hoy; atendiendo a las necesidades formativas de los niños, adolescentes y jóvenes cubanos.

El cuerpo editorial de la Casa Editora Abril ha solicitado al Centro de Innovación y Desarrollo de Internet de la Universidad de las Ciencias Informáticas, la realización de una nueva versión del Portal web para la Casa Editora Abril, debido a que la versión actual fue desarrollada hace varios años con una tecnología obsoleta, presentando un diseño poco atractivo para la actualidad, debido a la desactualización respecto a las nuevas tendencias en diseño y desarrollo web, deficiencias que atentan contra su popularidad y visibilidad en internet.

Por otra parte, el sistema fue desarrollado sobre la versión 7 de Drupal, la cual tiene como principal limitación que se le dejará de brindar soporte. Esto significa que, la comunidad de desarrollo de Drupal, no corregirá errores de funcionalidad ni programará parches y, sobre todo, no solucionará fallas de seguridad.

Esto trae como consecuencia que, el Portal web para la Casa Editora Abril, como el resto de los sistemas desarrollados sobre Drupal 7, quedará vulnerables a nuevos ataques y a violaciones de su seguridad, que pudieran ser letales para el sistema y a los cuales no se les encontrarán solución dentro de la comunidad de Drupal.

De igual forma, los errores de funcionamiento del CMS que puedan aparecer luego del EOL (*End Of Life*) de Drupal 7, no serán atendidos por la comunidad, por lo cual encontrarle solución requerirá tal vez de un tiempo considerable, que pudiera ser vital para la permanencia del sitio en Internet.

Por lo planteado con anterioridad se define, como **problema de investigación**: ¿Cómo migrar el portal web de la Casa Editorial Abril a una versión superior que permita corregir las deficiencias de la versión anterior para lograr la correcta divulgación del quehacer diario de la Institución de cara a Internet? Considerando como **objeto de estudio**: la migración de portales web. Se plantea como **objetivo general** de la investigación: desarrollar el portal web de la Casa Editorial Abril a una versión superior que permita corregir las deficiencias de la versión anterior para lograr la correcta divulgación del quehacer diario de la Institución de cara a Internet. Por lo tanto, se define como **campo de acción**: La migración del portal web de la Casa Editora Abril.

Para guiar la investigación se definen las siguientes **preguntas científicas**:

1. ¿Cuáles elementos deben tenerse en cuenta para la migración del portal web de la Casa Editora Abril?
2. ¿Cuáles son los referentes teóricos relacionados con el desarrollo de portales web de corte social, que sustentan la investigación?
3. ¿Qué elementos deben tenerse en cuenta para llevar a cabo el análisis y diseño para la migración del portal web de la Casa Editora Abril en su versión 2.0 que facilite la divulgación del quehacer diario de la Institución de cara a Internet?
4. ¿Cómo materializar, en términos de componentes y código fuente, los elementos especificados para la migración del portal web de la Casa Editora Abril a su versión 2.0 que facilite la divulgación del quehacer diario de la Institución de cara a Internet?
5. ¿Qué resultados se obtendrán al validar, a través de una estrategia de pruebas de software, la migración del portal web de la Casa Editora Abril a su versión 2.0 que facilite la divulgación del quehacer diario de la Institución de cara a Internet?

Para dar respuesta a las preguntas científicas formuladas, es necesario dar cumplimiento a las siguientes **tareas de investigación**:

1. Caracterizar los referentes teóricos relacionados con el desarrollo de portales web de corte social.
2. Caracterizar el portal web de la Casa Editora Abril.

3. Definir el ambiente de desarrollo para la migración del portal web de la Casa Editora Abril a su versión 2.0 que facilite la divulgación del quehacer diario de la Institución de cara a Internet.
4. Diseñar la versión 2.0 del Portal web para la Casa Editora Abril que contribuya a la divulgación del quehacer diario de la Institución de cara a Internet.
5. Implementar la versión 2.0 del Portal web para la Casa Editora Abril que contribuya a la divulgación del quehacer diario de la Institución de cara a Internet.
6. Probar la versión 2.0 del Portal web para la Casa Editora Abril que contribuya a la divulgación del quehacer diario de la Institución de cara a Internet.

Los **métodos de investigación** utilizados para dar solución a la presente investigación son:

Teóricos:

- **Analítico-Sintético:** Se utiliza para analizar y comparar elementos fundamentales relacionados con el proceso de desarrollo de portales web políticos y definir las herramientas.
- **Histórico-Lógico:** Se emplea para realizar un estudio de los conceptos de forma cronológica, el uso actual de los portales web de política y características de los mismos.

Empíricos:

- **La Observación:** Se utiliza para conocer el funcionamiento de los sistemas existentes en la actualidad y detectar carencias que pueden ser corregidas en el portal web para poder satisfacer al usuario final.
- **La Encuesta:** Recolecta datos mediante la realización de un cuestionario para conocer las necesidades del cliente, definir las funcionalidades del portal web e identificar los requisitos de cada usuario.

El trabajo está estructurado en 3 capítulos, los cuales se describen a continuación:

Capítulo 1: Fundamentación teórica de la investigación. En este capítulo se estudiarán los conceptos fundamentales que se tratarán a lo largo de la investigación, los diferentes sistemas similares existentes y su funcionamiento, además se definirán las herramientas, tecnologías y la metodología a utilizar.

Capítulo 2: Análisis y propuesta del Portal Web Casa Editora Abril 2.0. En este capítulo se propondrá una solución a partir de una descripción detallada de las funcionalidades que se implementarán. Se describirán las historias de usuario que se desarrollarán en cada iteración y se definirá un plan de entrega del producto. Se asentarán todos los elementos correspondientes al diseño.

Capítulo 3: Implementación y pruebas del Portal Web Casa Editora Abril 2.0. Se diseñará el modelo físico de la base de datos. Además, se materializa la propuesta del sistema que da cumplimiento a los objetivos trazados con la investigación y la validación del mismo a través de las pruebas realizadas.

En el siguiente capítulo se tiene como objetivo abordar los aspectos teóricos asociados al problema de la investigación. Se explica mediante un estudio realizado de diferentes herramientas que tienen un funcionamiento similar a la aplicación que se quiere desarrollar y lo que se puede utilizar de ellas. También se seleccionará la metodología de desarrollo a utilizar, las herramientas y tecnologías, y se describirá las características de estas.

1.1 Conceptos básicos relacionados con el dominio de la investigación

Se hace necesario dominar los conceptos básicos asociados al dominio de investigación para una correcta comprensión de la situación problemática y abordar en la propuesta solución todas las características que garanticen un sistema al corriente de las tendencias actuales de desarrollo web.

1.1.1 Portales web

El origen de la Web está vinculado al surgimiento de Internet. Las últimas décadas del pasado siglo y lo que va del presente se han caracterizado por un gran desarrollo de la informática, principalmente en la transmisión de datos mediante redes y el desarrollo de la red de redes conocida como Internet. En ella se produce un enorme cúmulo de información que es importante saber dónde está, cómo buscarla, cómo utilizarla y hacer de ella una fuente de conocimiento útil y contextualizado (Ibarra, 2011).

En este sentido la Web puede resultar apropiada para proporcionar una vía que permita acceder a la información antes mencionada, mediante la tecnología, cliente servidor, o sea que haya un servidor Web al que acceden los clientes por medio de navegadores, que son aplicaciones para visualizar documentos Web y permiten leer, las páginas Web, en ellos (Ibarra, 2011).

Un portal web o portal público se refiere a un sitio web o servicio que ofrece una amplia gama de recursos y servicios, como correo electrónico, foros, motores de búsqueda y centros comerciales en línea. Los primeros portales web eran servicios en línea, como América Online (AOL), que proporcionaban acceso a la Web. Los motores de búsqueda se han transformado en portales web para atraer y mantener un público más amplio (Singh & Rao, 2016).

En la actualidad el desarrollo de portales web se encuentra en constante cambio. Las empresas, ya sean grandes o pequeñas, necesitan un sitio web capaz de competir en un mercado donde el público demanda estar al corriente de las últimas tendencias en diseño, seguridad, fiabilidad y usabilidad.

1.1.2 Evolución de los portales web

La web está en constante renovación y evolución. Desde su creación el año 1966, con esa primera red *Arpanet*, hasta el posterior nacimiento del Internet que conocemos, no ha dejado de cambiar y perfeccionarse. Hemos pasado de una web 1.0 a la 2.0, 3.0 y ahora llega la web 4.0.

- La **web 1.0**, fue la primera (apareció hacia 1990) y en ella solo se podía consumir contenido. Se trataba de información a la que se podía acceder, pero sin posibilidad de interactuar; era unidireccional (Latorre, 2018).
- La **web 2.0**, (apareció en 2004) y contiene los foros, los blogs, los comentarios y después las redes sociales. La web 2.0 permite compartir información. Y aquí estamos, de momento la mayor parte de los consumidores (Latorre, 2018).
- La **web 3.0** (fue operativa en el 2010) y se asocia a la web semántica, un concepto que se refiere al uso de un lenguaje en la red. Por ejemplo, la búsqueda de contenidos utilizando palabras clave (Latorre, 2018).

La **web 4.0**. empezó en el 2016 y se centra en ofrecer un comportamiento más inteligente y más predictivo, de modo que podamos, con sólo realizar una afirmación o una llamada, poner en marcha un conjunto de acciones que tendrán como resultando aquello que pedimos, deseamos o decimos.

1.1.3 Clasificación de los portales web

Podemos distinguir fundamentalmente dos tipos de portales

Portales generales, horizontales o mega portales:

Son aquellos cuyo contenido abarca casi todos los temas posibles de Internet. Están encaminados a un usuario estándar de Internet, su contenido recoge información de interés general como noticias de actualidad, información sobre el clima, servicios de valor añadido, etc. Es habitual, además, que estos portales estén dirigidos a una comunidad específica de usuarios (Rios et al., 2017).

Portales especializados o temáticos:

Los portales especializados son fruto también de la verticalización de la información en Internet. Son modelos de portales los cuales tratan de proyectar los servicios genéricos de un portal horizontal en un ámbito más específico, con un criterio temático. Los portales especializados pretenden pues satisfacer las necesidades de información de una comunidad de usuarios concreta (Rios et al., 2017).

Según esta última afirmación podemos destacar dos tipos de portales especializados:

- **Portales empresariales:** Es un sitio Web que proporciona información de la empresa o institución a la que pertenece fundamentalmente a los empleados de la misma. Normalmente este tipo de portales corporativos son una prolongación natural de las intranets corporativas en las que se ha cuidado la organización de la información a través de la www que suministran (Rios et al., 2017).
- **Portales verticales:** Un portal es un sitio Web que proporciona información servicios a una industria en particular. Es el equivalente industrial específico de los portales generales de la internet, pero, además de ofrecer servicios típicos centran su cobertura de contenidos en un tema o sector concreto (Rios et al., 2017).

La propuesta solución se enmarca entre los portales especializados o temáticos, siendo concretamente un portal empresarial que brinda información, hacia un sector en específico, de los quehaceres y servicios de la institución.

1.2 Análisis de portales web homólogos

Se realizó un estudio de portales nacionales e internacionales con el objetivo de identificar características y funcionalidades que pudieran incluirse en la solución final. Los parámetros fueron seleccionados de acuerdo a las tendencias actuales de desarrollo web. Se listan a continuación dichos parámetros:

- Enlaces a redes sociales
- Tendencias en el diseño web
- Noticias
- Buscador

1.2.1 Tendencias en el diseño web:

En el diseño web, como en toda disciplina del diseño, existen estilos y tendencias influenciadas por una serie de factores sociales y tecnológicos. El factor tecnológico ha condicionado el diseño web desde el inicio

de la *World Wide Web*, no solo en aspectos técnicos, también en su concepción, como es el caso del diseño orientado a dispositivos de menor tamaño, que se ha extendido a lo largo de la Web, aplicándose sus normas en todos los contextos.

La web se presenta con mayor claridad y simplicidad, dando prioridad al contenido y jugando con el espacio en blanco. El texto quiere llamar la atención del usuario, adquiere mayor tamaño y densidad con el uso habitual de la negrita. Se mantienen las **tipografías** sin serifa, aunque aumenta el uso de tipografías más elaboradas (Barba Soler, 2015).

Las **imágenes** y los **videos** pasan a ser de gran tamaño, ocupando el ancho de la pantalla y funcionando en ocasiones como *background* para el mensaje principal. La interfaz de usuario tiende a botones de gran tamaño, como mejora destinada a la experiencia de usuario en las pantallas de tamaño reducido (Barba Soler, 2015).

El **menú principal** se mantiene fijo, se queda visible o escondido en la parte superior o lateral de la pantalla para facilitar el acceso a las opciones de navegación. El *Flat Design* o **diseño plano** se convierte en tendencia en la Web, tanto para el diseño de contenidos como para la interfaz y sus elementos. Estos diseños se componen de colores planos y formas simplificadas, sin relieves ni sombreados difuminados. La influencia de la ilustración es evidente en esta tendencia. Se utilizan paletas de colores habituales en la ilustración vectorial, con colores frescos y tonos pastel, con cierta tendencia a la calidez (Barba Soler, 2015).

Las ilustraciones animadas se vuelven cada vez más comunes. Algunos diseñadores centran la interacción de usuario en **animaciones** interactivas, desarrolladas con CSS3 e implementadas con JavaScript. Las animaciones en CSS3 permiten mayor fluidez que las desarrolladas con JavaScript, ya que el código es representado por el navegador antes de empezar la animación. Cada vez más diseñadores optan por una propuesta **adaptable (responsive)** ante una versión móvil y otra de escritorio (Barba Soler, 2015).

Las webs que continúan teniendo una versión móvil suelen desarrollar apps para los dispositivos, ofreciendo su descarga en el sitio web. Los diseños flexibles han dado lugar a estructuras de una sola página, construyendo páginas web verticales, con efectos impresionantes mediante *Parallax* o webs de galerías y contenidos con *scroll* infinito. Por otra parte, aparecen webs con apariencia de *layouts* o diapositivas, donde la estructura puede ser de una sola página, pero los apartados ocupan una o varias diapositivas (Barba Soler, 2015).

1.2.2 Portales Internacionales

Portal Web de TeleSur (<https://www.telesurtv.net>)

En la página principal se ubica en la parte superior un banner con el logotipo de la organización, un menú con siete enlaces, con los que se puede acceder a todo el contenido del portal, un buscador interno, enlaces a redes sociales, suscripción al boletín y opción de idioma en los que está disponible el portal. El cuerpo se compone de una única columna con un diseño moderno en forma de tarjetas donde se encuentran noticias destacadas, videos, imágenes y enlaces a blogs externos. En el pie de página se ubican enlaces a páginas legales del sitio, a servicios y a sitios externos. También se puede apreciar la suscripción al boletín y enlaces a redes sociales. Como colores destacados del sitio se aprecian el azul oscuro y rojo.

Portal Web de RT en Español (<https://actualidad.rt.com>)

En la página principal, en la parte superior se encuentra un banner con el logotipo de la organización, un menú principal con seis enlaces con los que se puede acceder a todo el contenido del portal, un buscador interno y enlaces a redes sociales. El cuerpo presenta un diseño de dos columnas con formato de tarjetas donde se ubican las noticias, videos e imágenes. En el pie de página se ubican los enlaces a sitios de interés, datos de contacto, enlaces a redes sociales y suscripción a un boletín. Como colores destacados del sitio se aprecian el blanco y el negro.

1.2.3 Portales Nacionales

Portal Web de Casa Editora Abril (<http://www.casaeditoraabril.cu>)

La página principal presenta una estructura de dos banners superiores con dos menús situados diagonalmente entre ellos, una sección de un carrusel con imágenes y un buscador interno. El cuerpo contiene un diseño de tres columnas con cuatro secciones donde se encuentra entre ellas revistas y libros. En el pie de página se ubican datos de contacto, suscripción a boletín y enlaces a sitios de interés. Presenta una gama de colores con verde en diferentes tonalidades, además de blanco, negro y carmelita.

Entre las principales deficiencias encontradas en el portal web se encuentran, la presencia de enlaces rotos en menú, secciones y páginas internas. Ausencias de enlaces a redes sociales de la institución. Deficiencias en la visualización del contenido por diversidad de colores apagados, gran número de elementos en pantalla y no presentar un diseño *responsive* para poder ser visualizado en cualquier medio actual. Debido a la

versión desactualizada del sistema gestor de contenido Drupal en la versión 7, los módulos se encuentran sin soporte actual atentando contra la seguridad del portal web.

Portal Web de Cubadebate (<http://www.cubadebate.cu>)

La página principal presenta un banner superior con enlaces a secciones de interés del sitio, un buscador interno, enlaces a redes sociales, el logotipo y dos noticias destacadas. Le siguen dos banners con el fondo de los colores destacados del sitio, que presentan dieciocho enlaces con los que se pueden acceder a todo el sitio. El cuerpo presenta un diseño de dos columnas con la sección de noticias en la primera y opiniones en la segunda. En el pie de página se localizan las opciones para los diferentes idiomas en los que está disponible el portal web, así como enlaces a otros sitios de interés. Como colores destacados del sitio se aprecian el rojo y el negro.

Portal Web de Casa Editora Verde Olivo (<https://www.verdeolivo.cu>)

En la página principal se encuentra en la parte superior un banner con el logotipo de la institución, enlaces a redes sociales y un buscador interno, un menú principal con seis enlaces, que permiten acceder a todo el contenido del portal, y una imagen destacada del portal. El cuerpo se divide en cuatro secciones principales donde se encuentran noticias, revistas, libros y enlaces a secciones del portal. En el pie de página se encuentra la información de contacto y enlaces a sitios externos. Como colores destacados del sitio se aprecian el verde olivo y blanco.

1.2.4 Resultados de análisis de estudio de portales homólogos

De los portales web analizados se pudo constatar que ofrecen servicios similares como:

Informativos:

Noticias (Presente en casi todos los portales). Constituyen una de las fuentes principales de visitas.

Enlaces (Presente en todos los portales). Los enlaces conducen a portales web relacionados con la temática del portal analizado, estos se encontraron fundamentalmente en el pie de página.

Socialización

Acceso a redes sociales (Presente en casi todos los portales). Estos enlaces permiten difundir los diversos contenidos de los portales en las principales redes sociales como *YouTube*, *Facebook* y *Twitter*.

Elementos Complementarios:

Buscador (Presente en todos los portales). Facilita la búsqueda de contenido dentro del portal web

Tendencias en el diseño web (Presente en todos los portales web). Diseño minimalista, directo y agradable a la vista de cualquier usuario.

Tabla 1: Criterio de comparación para el estudio de portales web homólogos. (Elaboración propia)

Portal Web	Enlaces a redes sociales	Tendencias en el diseño web	Noticias	Buscador
TeleSur	Facebook Twitter Instagram Pinterest Youtube Paper	Adaptable (Responsive) Diseño plano Minimalista Animaciones	Título, Fecha, Imagen destacada, Etiquetas, comentarios, compartir en redes sociales	Si
RT en Español	Facebook Twitter Instagram Youtube Telegram	Adaptable (Responsive) Diseño plano Minimalista Gran número de animaciones	Título, Fecha, Imagen destacada, Etiquetas, compartir en redes sociales	Si
Casa Editora Abril	No	Gran número de elementos en pantalla	Título, Fecha, Imagen destacada, autor	Si
Cubadebate	Facebook Twitter Instagram Youtube Flickr Telegram	Adaptable (Responsive) Diseño plano Gran número de elementos en pantalla	Título, Fecha, Imagen destacada Categoría, Comentarios, compartir en redes sociales	Si
Casa Editorial Verde Olivo	Facebook Twitter Instagram Youtube Flickr Picta	Adaptable (Responsive) Diseño plano Minimalista	Título, Fecha, Imagen destacada, Autor, Comentarios	Si

Teniendo en cuenta los resultados del análisis de los portales web se puede concluir que cuentan con la mayoría de los servicios planteados. Por esto se plantea incluir estos elementos. Sin embargo, un portal web es un servicio personalizado para una empresa organización o institución, ningún portal homólogo analizado constituye una solución para el cliente.

1.3 Tecnologías y herramientas

En este epígrafe se expondrán todas las herramientas y tecnologías necesarias para la planificación y desarrollo del software, resultado del análisis de las necesidades requeridas.

1.3.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software son procesos que se siguen a la hora de diseñar una solución o un programa específico. Tiene que ver, por tanto, con la comunicación, la manipulación de modelos y el intercambio de información y datos entre las partes involucradas, son enfoques de carácter estructurado y estratégico que permiten el desarrollo de programas con base a modelos de sistemas, reglas, sugerencias de diseño y guías (Anna Pérez, 2016). Estas se pueden clasificar en ágiles o tradicionales. Las metodologías tradicionales están enfocadas al estricto control del proceso, estas han demostrado ser eficientes para proyectos de gran tamaño. Las metodologías ágiles se enfocan más en crear un *software* que funcione, que en mantener una documentación extensa y detallada.

El desarrollo ágil combina una filosofía y un conjunto de directrices de desarrollo. La filosofía busca la satisfacción del cliente y la entrega temprana de *software* incremental; equipos de proyectos pequeños y con alta motivación; métodos informales y una simplicidad en el desarrollo (Ruedas, 2016).

Dentro de las metodologías ágiles se escoge AUP porque es el que más se ajusta a las necesidades del desarrollo; ya que el mismo es un proyecto de mediana complejidad y poco tiempo de desarrollo (escaso margen para corregir errores) y requisitos bien definidos.

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (Rodríguez Sánchez, 2015).

AUP-UCI es una variación de la metodología AUP en unión con el modelo CMMI-DEV v 1.3 desarrollada en la UCI. Es flexible y no requiere de una gran cantidad de desarrolladores, concisa en el aspecto de la documentación, permitiendo generar solo la necesaria y no la especificada para cada flujo de trabajo como lo hace RUP. Está diseñada para trabajar en proyectos pequeños donde la atención se centra en las actividades que realmente son importantes. Permite el uso de herramientas de cualquier tipo, incluyendo aquí las de código abierto, es una metodología que se ajusta y aprovecha las ventajas que brindan las metodologías ágiles (Rodríguez Sánchez, 2015).

De las fases comprendidas en la metodología AUP se simplificaron en (Rodríguez Sánchez, 2015):

- **Inicio:** En la fase inicial se realiza un estudio de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo, costo y decidir si se ejecuta o no el proyecto.
- **Ejecución:** En esta fase se recogen las actividades que desarrolla AUP de elaboración, construcción y transición. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño. Se realizan las actividades planificadas para el desarrollo del software teniendo en cuenta los requisitos y arquitectura.
- **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades de cierre del proyecto.

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo. El Modelado del Negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el *software* desarrollado va a cumplir su propósito. AUP-UCI propone tres variantes para modelar el negocio: Casos de Uso del Negocio (CUS), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC). A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos, existen tres formas de encapsular los requisitos Casos de Uso del Sistema (CUS), Historias de Usuario (HU) y Descripción de Requisitos por Proceso (DRP), surgen cuatro escenarios para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC.

Se evaluó el negocio a informatizar y se obtuvo un negocio muy bien definido. El cliente dispone del tiempo para acompañar al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos, por lo que el escenario cuarto con las Historias de Usuario es el adecuado para encapsular los requisitos. El tratamiento de las historias de usuario es muy dinámico y flexible, lo que permite que estas puedan reemplazarse por otras más específicas o generales, añadir-se nuevas o ser modificadas de una manera más fácil, sin invertir tanto tiempo y esfuerzo.

Dadas las características anteriores se comprende que estas se ajustan perfectamente al trabajo a realizar, pues es idónea para la entrega de productos en cortos períodos de tiempo y satisface las necesidades del proceso. Se basa en las relaciones interpersonales como clave para el desarrollo de software propiciando un buen ambiente de trabajo. Su funcionamiento depende de la retroalimentación continua entre el cliente y el equipo de desarrollo, además de la simplicidad en las soluciones implementadas, por lo que se escoge AUP-UCI como metodología de desarrollo para guiar la investigación.

1.3.2 Sistemas de Gestión de Contenido

Un sistema de gestión de contenido (CMS, del inglés *Content Management System*) es una herramienta de software que permite crear, organizar y publicar documentos y otros contenidos de forma colaborativa. Está formado por un conjunto de aplicaciones web que, de un modo similar a un portal, operan tanto en internet como una intranet.

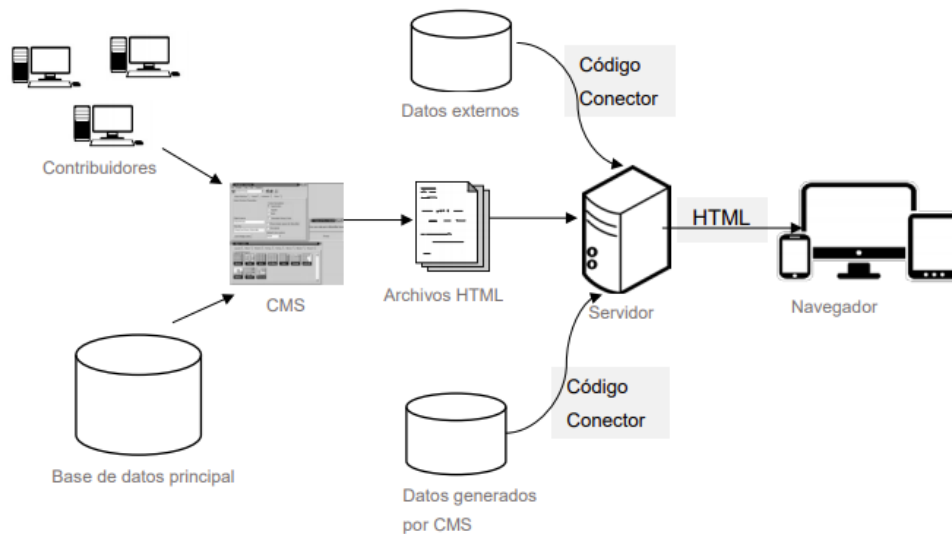


Figura 1. Sistema de gestión de contenido (Béjar Cáceres, 2017)

Su principal ventaja consiste en el hecho de que permite organizar y mostrar contenidos. Con su ayuda se pueden generar y publicar noticias, crear taxonomías (sistema de clasificación) con los que el administrador clasifica contenidos, insertar logotipos e imágenes personalizadas o corporativas del portal, añadir secciones, administrar las bases de datos de usuarios, entre otros (Mcgraw-hill.es, 2018).

Drupal es un software de gestión de contenido. Se utiliza para crear muchos de los sitios web y aplicaciones que usa todos los días. Pero lo que lo distingue es su flexibilidad; la modularidad es uno de sus principios fundamentales. Sus herramientas le ayudan a crear el contenido estructurado y versátil que necesitan las experiencias web dinámicas (Drupal.org, 2015).

Se selecciona Drupal en su versión 8, porque cuenta con la estabilidad y adaptabilidad necesaria, tiene excelentes características estándar, como creación de contenido sencilla, rendimiento confiable y excelente seguridad.

1.3.3 Lenguaje para el modelado

El lenguaje de modelado unificado (UML) es un estándar para la representación visual de objetos, estados y procesos dentro de un sistema. Por un lado, el lenguaje de modelado puede servir de modelo para un proyecto y garantizar así una arquitectura de información estructurada; por el otro, ayuda a los

desarrolladores a presentar la descripción del sistema de una manera que sea comprensible para quienes están fuera del campo. UML se utiliza principalmente en el desarrollo de software orientado a objetos. Al ampliar el estándar en la versión 2.0, también es adecuado para visualizar procesos empresariales (IONOS.es, 2018).

Los diagramas UML se utilizan para representar los siguientes componentes del sistema:

- Objetos individuales (elementos básicos)
- Clases (combina elementos con las mismas propiedades)
- Relaciones entre objetos (jerarquía y comportamiento/comunicación entre objetos)
- Actividad (combinación compleja de acciones/módulos de comportamiento)
- Interacciones entre objetos e interfaces

1.3.4 Lenguajes de programación

En la interacción entre los medios de los usuarios y los servidores en internet, se debe tomar en cuenta las distintas tecnologías para ambos extremos, el lado del cliente se le conoce como “*frontend*” y el lado del servidor se le conoce como “*backend*”(Platzi, 2017).

Luego de un estudio de conjunto con el cliente, se seleccionaron las herramientas y tecnologías necesarias acordes al proyecto en cuestión. Se tiene en cuenta para este las necesidades del cliente como: el tiempo de desarrollo, el cual es en un periodo corto, por lo que el ambiente de desarrollo debe cumplir con los estándares actuales de desarrollo, además de contribuir a un proceso ágil del mismo.

Lenguajes lado del cliente (*fronted*)

HTML 5 (*HyperText Markup Language*, versión 5): Es un lenguaje de marcas, diseñado para estructurar textos y presentarlos en forma de hipertexto y multimedia. Es un lenguaje que se escribe mediante etiquetas, que aparecen especificadas por corchetes angulares (< >) (Gauchat, 2012).

CSS3 (*Cascade Style Sheets*, versión 3): Las siglas CSS significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, etc. La idea de CSS es la de utilizar el concepto de separación de presentación y contenido, intentando que los documentos HTML incluyan sólo información y datos, relativos al significado de la información a transmitir (el contenido),

y todos los aspectos relacionados con el estilo (diseño, colores, formas, etc.) se encuentren en un documento CSS independiente (la presentación) (Aubry, 2014).

JavaScript (JS): Lenguaje de programación dinámico orientado a objetos funcional que permite crear efectos especiales en las páginas web y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones *Javascript* y ejecutarlas. Es un lenguaje bastante sencillo, rápido y fácil de aprender por personas de poca experiencia. Permite que elementos de una página posean movimiento, cambien de color u otro efecto similar (Gauchat, 2012).

Bootstrap 4: Permite crear interfaces web con CSS y *Javascript* que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador, un móvil o de una Tablet. Es el kit de herramientas de código abierto *frontend* (lado cliente) más popular del mundo, presenta amplios componentes prediseñados y potentes complementos de JavaScript (Bootstrap, 2021).

Lenguajes lado del servidor (*backend*)

PHP 7.2.9 (acrónimo recursivo de PHP: *Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web. Se usa fundamentalmente para la ejecución de scripts del lado servidor, para esto son necesarias tres cosas: el analizador de PHP (servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada (PHP, 2021).

1.3.5 Servidores de aplicaciones

El Proyecto del servidor HTTP Apache es un esfuerzo de desarrollo de software colaborativo destinado a crear una implementación de código fuente sólida, de calidad comercial, con funciones y disponible de forma gratuita de un servidor HTTP (Web). El proyecto es administrado conjuntamente por un grupo de voluntarios ubicados en todo el mundo, que utilizan Internet y la Web para comunicarse, planificar y desarrollar el servidor y su documentación relacionada. Este proyecto es parte de *Apache Software Foundation*. Además, cientos de usuarios han contribuido con ideas, código y documentación al proyecto. Este archivo está destinado a describir brevemente la historia del servidor HTTP Apache y reconocer a los muchos contribuyentes (Apache, 2021).

Entre sus principales características se encuentran:

- Es gratuito y de fuente abierta.
- Instalación y configuración sencilla
- Altamente extensible y adaptable mediante módulos
- Funciones incorporadas para autenticación y validación de usuarios.
- Soporte para lenguajes como Perl, PHP y Python.

Apache presenta gran cantidad de ventajas lo que hacen una opción segura para cualquier desarrollo de software:

- **Soporte:** es uno de los servidores más usados desde hace más de dos décadas por lo que detrás existe una comunidad muy grande acompañado una extensa documentación.
- **Multiplataforma:** funciona tanto en Microsoft Windows, Unix/Linux y Mac por lo cual es un excelente servicio tanto para servidor como para un entorno de desarrollo en cualquier escritorio.
- **Funcionalidades:** contiene una gran cantidad de módulos tanto oficiales como de terceros que permiten ampliar las funcionalidades del servidor.
- **Sencillez:** es uno de los servidores más sencillos de instalar y configurar, sobre todo en distribuciones Linux donde forma parte de los repositorios de cada distribución y en el caso de Windows se lo puede encontrar empaquetado junto con el servidor MySQL y PHP en un solo instalador que facilita enormemente la instalación.
- **Seguridad:** Apache integra funcionalidades para brindar seguridad al software gracias a los módulos de Autorización y Autenticación y funciones de Control de Acceso, además de un soporte para cifrado por certificados SSL/TLS.

Apache se utiliza en su versión 2.4.34.

1.3.6 Gestor de base de datos

Un Sistema Gestor de Base de Datos (SGBD) es un software que sigue un modelo de sistema de base de datos y, por lo tanto, resulta decisivo a la hora de configurarla, administrarla y utilizarla. Solo cuando el sistema gestor de base de datos está instalado y configurado, los usuarios pueden introducir y consultar los datos. Los permisos de lectura y escritura, así como las funciones de administración generales, se

establecen mediante las interfaces específicas de la aplicación y el lenguaje de definición de datos correspondiente (IONOS.es, 2020c).

MySQL 5.0.12 es un SGBD relacional, multihilo, multiplataforma y multiusuario. es el sistema de gestión de bases de datos *Open Source* más popular del mundo y es conocido por su rendimiento y fiabilidad. Se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia (MySQL, 2021).

phpMyAdmin 4.8.3 es una herramienta de software libre escrito en PHP, esta facilita la administración del servidor MySQL a través de la web, dispone de una interfaz gráfica y es de libre distribución. Permite realizar todo tipo de operaciones sobre bases de datos (phpMyAdmin, 2021).

1.3.7 Herramientas

Visual Paradigm 8.0

En su versión 8.0 es la herramienta CASE (del inglés *Computer Aided Software Engineering*) a la que se le dará uso, la cual es una herramienta de diseño y gestión multiplataforma que puede ser usado para sistemas de TI (Tecnologías de la Información). Visual Paradigm proporciona a los desarrolladores de software la plataforma de desarrollo vanguardia para crear aplicaciones de calidad rápido y barato. Facilita una excelente interoperabilidad con otras herramientas CASE y la mayoría de los IDE líderes. Además, brinda ciertas ventajas o facilidades como (Visual Paradigm, 2021):

- **Modelado UML:** permite dibujar todo tipo de diagramas como los de clase, los de paquete, etc.
- **Desarrollo ágil:** cuenta con un conjunto de herramientas ágiles aplicables que lo ayudan a construir un sistema de software de manera ágil.
- **Diseño de bases de datos:** permite dibujar diagramas para ayudar en el modelado de bases de datos.
- **Modelado de proceso del negocio:** puede dibujar diagramas como el de flujo de datos para ayudar en el modelado de procesos de negocio.
- **Mapeo relacional de objetos:** permite acceder a la base de datos en un enfoque relacional de objetos al codificar.
- **Colaboración en equipos:** permite realizar modelos de forma colaborativa y simultánea.

Acunetix 11: Se emplea para escanear el portal web para así identificar vulnerabilidades del mismo. *Acunetix Web Vulnerability Scanners* una herramienta de seguridad de aplicaciones Web automatizada.

Acunetix WVS es capaz de escanear cualquier sitio Web o aplicación Web que es accesible a través del protocolo HTTP / HTTPS. Sin embargo, no todas las pruebas se pueden realizar de forma automática, y por lo tanto *Acunetix WVS* proporciona herramientas de penetración manuales para pruebas particulares (Acunetix, 2021).

JMeter 5.4.1: Apache JMeter es un software de código abierto, una aplicación Java 100% pura diseñada para evaluar el comportamiento funcional de pruebas de carga y medir el rendimiento (Apache JMeter, 2021).

Su funcionamiento se basa en simular visitantes a la aplicación o servicio permitiendo a los usuarios crear y enviar solicitudes HTTP (Protocolo de transferencia de hipertexto) al servidor. A continuación, se recopilan los datos de respuesta del servidor y los datos estadísticos se muestran visualmente para los usuarios en forma de gráficos e informes. A partir de los datos recopilados y creados, los usuarios pueden observar el comportamiento de la aplicación y los cuellos de botella de rendimiento, ofreciendo información sobre dónde deben realizarse las mejoras de las aplicaciones.

1.4 Conclusiones parciales

En el capítulo se reflejan los elementos teóricos necesarios para la propuesta de solución del portal web, por lo tanto:

- La definición del marco teórico permitió conocer todos los elementos teóricos para sustentar la solución.
- El estudio de los portales homólogos identificó características necesarias en la elaboración e implementación del portal web.
- El análisis de la metodología AUP-UCI trazó las pautas adecuadas a seguir en el desarrollo.
- El estudio de las herramientas y tecnologías permitió definir el entorno de desarrollo adecuado para la propuesta de solución.

CAPÍTULO 2. ANÁLISIS Y PROPUESTA DEL PORTAL WEB CASA EDITORA ABRIL 2.0

En el presente capítulo se expondrá la propuesta solución, con las funciones, características del sitio y la relación entre ellas, los estilos arquitectónicos y patrones de diseño del sistema gestor de contenidos Drupal. Se definen los requisitos funcionales y no funcionales y los artefactos necesarios previo a la implementación del portal web.

2.1 Propuesta de solución

El portal web Casa Editora Abril está conformado por un conjunto de páginas dinámicas que se irán implementando según su prioridad en el portal. El banner superior está conformado por el logo principal de la institución, enlaces a redes sociales y un buscador interno. El menú principal con el que se podrá acceder a todo el contenido del portal. Después se encontrará la imagen destacada del momento, que pudiera ser un anuncio de un libro, revista o de un evento. Los últimos artículos (tres) según la fecha se encontrarán en la sección que lleva dicho nombre, un artículo destacado de interés le seguirá. A continuación, dos secciones principales correspondientes a las revistas y libros y una colección de imágenes destacadas. En el pie de página se ubican el registro de visitas, contáctenos: con los datos como el correo electrónico, dirección física, teléfono entre otros, enlaces a sitios de interés y la opción de suscripción al boletín mediante correo electrónico.

Tabla 2: Arquitectura de información. (Elaboración propia)

Logo		Enlaces redes sociales	Buscar
Menú Principal			
Carrusel de Imágenes			
Últimos Artículos			

Artículo Destacado			
Revistas			
Libros			
Colección de Imágenes			
Registro de visitas	Contáctenos	Sitios de interés	Suscripción Boletín

2.1.1 Vistas de la propuesta solución

A continuación, se muestran vistas de la propuesta solución correspondiente a la estructura de la página principal.

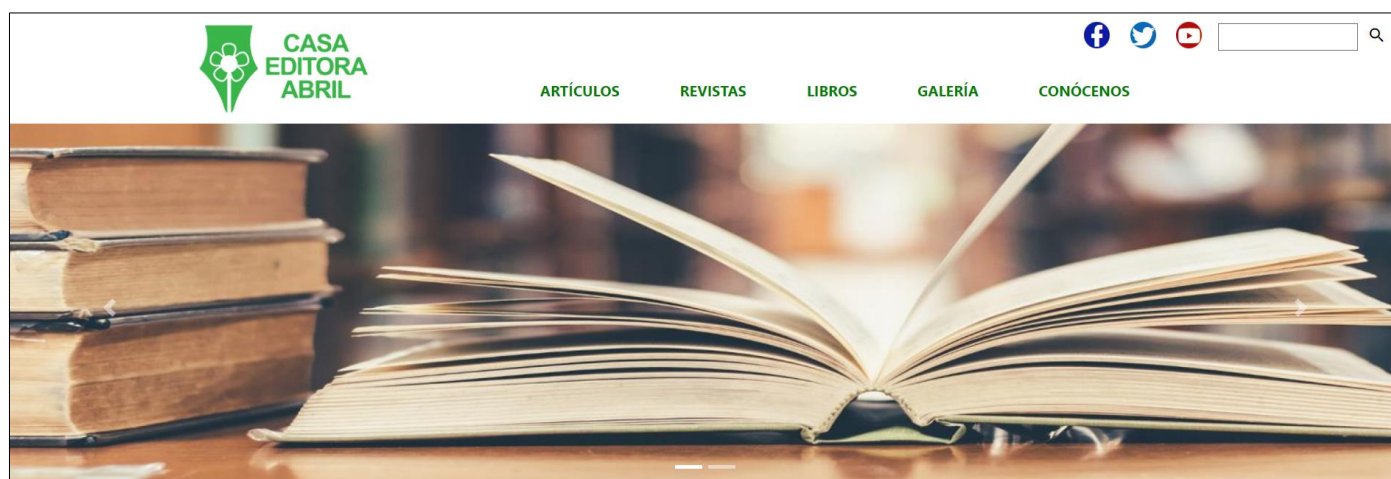



Figura 2. Vista sección banner superior y carrusel de imágenes. (Elaboración propia)

Últimos Artículos




Lorem ipsum dolor sit amet

Jueves, November 25, 2021 - 01:31
Frank Verdecia Blanco

Lorem ipsum dolor sit amet consectetur adipiscing elit suscipit, posuere inceptos facilisi auctor molestie pellentesque eleifend, porttitor laoreet velit enim mus justo massa.

[Leer más](#)




Lorem ipsum dolor sit amet

Jueves, November 25, 2021 - 01:05
Frank Verdecia Blanco

Lorem ipsum dolor sit amet consectetur adipiscing elit suscipit, posuere inceptos facilisi auctor molestie pellentesque eleifend, porttitor laoreet velit enim mus justo massa.

[Leer más](#)



Lorem ipsum dolor sit amet

Jueves, November 25, 2021 - 00:59
Frank Verdecia Blanco

Lorem ipsum dolor sit amet consectetur adipiscing elit suscipit, posuere inceptos facilisi auctor molestie pellentesque eleifend, porttitor laoreet velit enim mus justo massa.

[Leer más](#)

Figura 3. Vista sección Últimos Artículos. (Elaboración propia)

Artículo destacado



Lorem ipsum dolor sit amet

Jue, 25/11/2021 - 01:48
Frank Verdecia Blanco

Lorem ipsum dolor sit amet consectetur adipiscing elit suscipit, posuere inceptos facilisi auctor molestie pellentesque eleifend, porttitor laoreet velit enim mus justo massa. Eros placerat morbi aliquet sed ligula phasellus et, etiam fusce cursus donec imperdiet taciti orci tellus, est lobortis parturient nam erat torquent. Curabitur convallis taciti posuere euismod primis diam rhoncus fringilla et pretium semper, maecenas vehicula congue himenaeos lobortis natoque mi eu cubilia turpis, duis laoreet id sapien litora nunc mattis in erat felis.

Figura 4. Vista sección Artículo destacado. (Elaboración propia)

2.2 Especificación de los requisitos

2.2.1 Requisitos funcionales

En coordinación con el cliente se identificaron un total de 48 requisitos funcionales, le fue dada una prioridad de acuerdo a la importancia fijada por el cliente y su complejidad para el desarrollo. Para la correcta implementación del sistema se prevé que la propuesta solución cuente con los siguientes requisitos:

Tabla 3: Requisitos funcionales. (Elaboración propia)

Requerimientos funcionales			
No	Requerimiento	Descripción	Prioridad
RF1.	Autenticar usuario	El usuario introduce sus datos, el sistema verifica que se encuentra en la base de datos y el usuario queda autenticado con el nivel de privilegios asignado, si los datos no son válidos el sistema muestra un mensaje de error.	Media
RF2.	Asignar rol a usuario	El sistema permite que los usuarios con rol administrador asignen roles a los usuarios.	Media
RF3.	Crear usuario	El sistema permite a los usuarios con rol administrador crear los usuarios.	Alta
RF4.	Modificar usuario	El sistema permite que los usuarios con rol administrador editen los usuarios.	Media
RF5.	Mostrar usuario	El sistema permite que los usuarios con rol administrador muestren información de los usuarios	Baja
RF6.	Mostrar listado de usuarios	El sistema permite que los usuarios con rol administrador listen todos los usuarios	Baja

RF7.	Eliminar usuario	El sistema permite que los usuarios con rol administrador eliminen usuarios	Baja
RF8.	Añadir artículo	El sistema permite que los usuarios con rol de editor creen los artículos.	Alta
RF9.	Modificar artículo	El sistema permite que los usuarios con rol de editor o administrador modifiquen los artículos.	Media
RF10.	Eliminar artículo	El sistema permite que los usuarios con rol de editor eliminen los artículos.	Media
RF11.	Mostrar artículo	El sistema permite que todos los usuarios observen los artículos.	Alta
RF12.	Añadir libro	El sistema permite que los usuarios con rol de administrador añadan libros.	Alta
RF13.	Modificar libro	El sistema permite que los usuarios con rol de administrador modifiquen los libros.	Media
RF14.	Mostrar libro	El sistema permite a los usuarios ver los libros.	Alta
RF15.	Descargar libro	El sistema permite que los usuarios descarguen el libro publicado en un documento pdf.	Media
RF16.	Eliminar libro	El sistema permite a los usuarios con rol de administrador eliminar los libros.	Baja
RF17.	Mostrar libro	El sistema muestra a los usuarios los libros.	Media
RF18.	Añadir comentario	El sistema permite que los usuarios añadan comentarios.	Alta

RF19.	Moderar comentario	El sistema permite que los usuarios con rol de administrador den permisos para mostrar el comentario.	Alta
RF20.	Responder comentario	El sistema permite que los usuarios respondan comentarios	Baja
RF21.	Eliminar comentario	El sistema permite que los usuarios con rol de administrador puedan eliminar los comentarios.	Media
RF22.	Mostrar comentario	El sistema permite que los usuarios puedan ver los comentarios.	Alta
RF23.	Añadir revista	El sistema permite que los usuarios con rol de administrador añadan revistas.	Media
RF24.	Mostrar revista	El sistema permite que se muestren las revistas en el portal web.	Media
RF25.	Modificar revista	El sistema permite que los usuarios con rol de administrador modifiquen las revistas.	Media
RF26.	Exportar revista	El sistema permite que los usuarios descarguen la revista publicada en un documento pdf.	Media
RF27.	Eliminar revista	El sistema permite que los usuarios con rol de administrador eliminen las revistas.	Media
RF28.	Mostrar enlaces a las redes sociales	El sistema permite que los usuarios accedan a los perfiles oficiales de la institución en las redes sociales.	Alta
RF29.	Añadir enlaces de interés	El sistema permite que los usuarios con rol de administrador añadan otros sitios de interés.	Alta

RF30.	Eliminar enlaces de interés	El sistema permite que los usuarios con rol de administrador eliminen estos sitios de interés.	Baja
RF31.	Mostrar enlaces de interés	El sistema permite que los usuarios puedan ver y acceder a sitios de interés.	Baja
RF32.	Realizar búsqueda avanzada	El sistema permite la búsqueda avanzada de la información que contiene.	Baja
RF33.	Realizar búsqueda simple	El sistema permite la búsqueda simple de la información que contiene.	Baja
RF34.	Añadir colección de imágenes	El sistema permite que los usuarios con rol de administrador añadan colecciones de imágenes.	Alta
RF35.	Editar colección de imágenes	El sistema permite que los usuarios con rol de administrador editen colecciones de imágenes.	Media
RF36.	Eliminar colección de imágenes	El sistema permite que los usuarios con rol de administrador eliminen colecciones de imágenes.	Media
RF37.	Mostrar colecciones de imágenes	El sistema permite que los usuarios vean las colecciones de imágenes.	Alta
RF38.	Mostrar estadísticas de visitas	El sistema permite que los usuarios vean la cantidad de visitas por día y total en el sitio.	Media
RF39.	Listar contenido	El sistema permite tener un listado de todo el contenido presente en el portal.	Media
RF40.	Editar página básica	El sistema permite que los usuarios con rol de administrador editen las páginas básicas.	Alta

RF41.	Mostrar página básica	El sistema permite mostrar al cliente las páginas básicas.	Alta
RF42.	Crear boletín	El sistema permite que los usuarios administradores creen el boletín	Media
RF43.	Editar boletín	El sistema permite que los usuarios administradores puedan editar el boletín	Baja
RF44.	Mostrar boletín	El sistema permite que los usuarios con rol administrador vean los boletines	Baja
RF45.	Eliminar boletín	El sistema permite que los usuarios con rol de administrador eliminen boletines	Baja
RF46.	Enviar boletín	El sistema permite que los usuarios con rol de administrador envíen un boletín a los usuarios suscritos.	Media
RF47.	Suscribirse a boletín	El sistema permite que los usuarios se suscriban para recibir boletines.	Media
RF48.	Cancelar Suscripción a boletín	El sistema permite que los usuarios puedan cancelen su suscripción al boletín.	Baja

2.2.2 Requisitos no funcionales

Los requisitos no funcionales son aquellos requisitos que no describen información a guardar, ni funciones a realizar, sino que son propiedades que hacen al producto usable, rápido o confiable. Además, se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software (Sommerville, 2005).

RnF 1: Usabilidad

RnF 1.1: El sistema debe presentar una interfaz agradable e intuitiva.

RnF 2: Eficiencia

RnF 2.1: El sistema debe permitir a los usuarios trabajar de manera concurrente con un mínimo de 100 usuarios.

RnF 2.2: El sistema debe demorar en cada transición un tiempo menor a cinco segundos aproximadamente.

RnF 3: Seguridad

RnF 3.1: Cada usuario solo tendrá acceso a las funcionalidades que le corresponde de acuerdo a su rol asignado.

RnF 3.2: En caso de alguna falla en el sistema, los errores no mostrarán información que comprometa la seguridad e integridad del mismo.

RnF 3.3: Se accederá a las páginas del sistema a través del protocolo HTTPS.

RnF 3.4: Se debe contar con un sistema de salvadas externas tanto del código como de la base de datos para garantizar la fiabilidad del portal web.

RnF 4: Software

RnF 4.1: Servidor web Apache en su versión 2.4.

RnF 4.2: Servidor de base de datos MySQL en su versión 5.7.24 o superior.

RnF 4.3: Lenguaje de programación PHP 7 y como sistema gestor de contenidos Drupal 8.7.

RnF 4.4: El sistema debe tener visibilidad en los principales navegadores web como Chrome v.45, Firefox v.38, Safari v.9, Opera v.30.

RnF 4.5: Uso de licencia GNU/GPL para el CMS Drupal.

RnF 4.6: Uso de licencia BSD de MySQL.

RnF 4.7: Uso de licencia PHP License.

RnF 5: Hardware

RnF 5.1: El servidor de aplicaciones web debe poseer una capacidad mínima de 80 GB.

RnF 5.2: El servidor de base de datos debe poseer una capacidad mínima de 40 GB.

RnF 5.3: Los servidores de aplicaciones web y de base de datos deben poseer como mínimo 1 GB de memoria RAM.

2.3 Descripción de requisitos de software (Historias de Usuarios)

Una vez que los requisitos son establecidos y estimados es necesario que sean descritos minuciosamente para facilitar su desarrollo. Las Historias de Usuarios (HU) “se escriben desde la perspectiva del cliente, aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido de estas debe ser concreto y sencillo.

El tratamiento de las HU es muy dinámico y flexible, en cualquier momento pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas pocas semanas.

2.3.1 Modelación de las HU

Las HU serán representadas mediante tablas divididas por las siguientes secciones: Número, Nombre del Requisito, Programadores Responsables, Iteración Asignada, Prioridad en Negocio (Se define según la prioridad que tengan en el desarrollo y se mide en Baja, Media y Alta), Riesgo en Desarrollo (establecido por el riesgo que representa cometer un error en el desarrollo de la misma), Tiempo Estimado (tiempo estimado que se demorará el desarrollo de la HU) Tiempo Real (tiempo que se demoró en realidad el desarrollo de la HU), Descripción, Observaciones, Prototipo de Interfaz.

Tabla 4: Historia de Usuario Añadir Artículo. (Elaboración propia)

Número: 8	Nombre del requisito: Añadir artículo
Programador: Frank Verdecia	Iteración Asignada: Primera Iteración
Prioridad: Alta	Tiempo Estimado: 1.05h

Riesgo en Desarrollo: Alto	Tiempo Real: 1.10h
<p>Descripción: Los usuarios autenticados con privilegios de editor o administrador podrán crear artículo en el sistema, para ello deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> • Título: Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 255 caracteres. • Categoría: Obligatorio. Lista desplegable. Valores (Lista de categorías). Valor por defecto (-Seleccione un valor-). • Imagen: Obligatorio. campo de tipo archivo. Tamaño máximo 512KB. Permite solo los archivos con las extensiones png, gif, jpg, jpeg. • Título de la imagen: Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 1024. • Texto alternativo: Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 512. • Cuerpo: Obligatorio. Área de texto. Permite todos los caracteres. • Fecha: Obligatorio. Calendario desplegable. Formato dd/mm/aaaa. Permite seleccionar de un calendario desplegable. Valor por defecto (fecha actual). • Autor: Opcional. Campo de autocompletamiento. Permite todos los caracteres. El autor tiene que haber estado creado en el sistema. • Palabras clave: Opcional. Campo de autocompletamiento. Permite solo los caracteres a-z, A-Z, coma y espacio. Las cadenas compuestas por los caracteres a-z, A-Z, deben estar separadas por coma (,). • Publicado: Opcional. Campo de selección. Valor por defecto (marcado). 	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el artículo. 2. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. 	

3. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.
Prototipo de interfaz: No aplica

Tabla 5: Historia de Usuario Modificar Artículo. (Elaboración propia)

Número: 9	Nombre del requisito: Modificar artículo
Programador: Frank Verdecia	Iteración Asignada: Primera Iteración
Prioridad: Media	Tiempo Estimado: 0.45h
Riesgo en Desarrollo: Medio	Tiempo Real: 0.50h
<p>Descripción: Los usuarios autenticados con privilegios de editor o administrador podrán modificar artículo en el sistema, para ello pueden modificar los siguientes campos:</p> <ul style="list-style-type: none"> • Título: Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 255 caracteres. • Categoría: Obligatorio. Lista desplegable. Valores (Lista de categorías). Valor por defecto (-Seleccione un valor-). • Imagen: Obligatorio. campo de tipo archivo. Tamaño máximo 512KB. Permite solo los archivos con las extensiones png, gif, jpg, jpeg. • Título de la imagen: Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 1024. • Texto alternativo: Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 512. • Cuerpo: Obligatorio. Área de texto. Permite todos los caracteres. 	

<ul style="list-style-type: none"> • Fecha: Obligatorio. Calendario desplegable. Formato dd/mm/aaaa. Permite seleccionar de un calendario desplegable. Valor por defecto (fecha actual). • Autor: Opcional. Campo de autocompletamiento. Permite todos los caracteres. El autor tiene que haber estado creado en el sistema. • Palabras clave: Opcional. Campo de autocompletamiento. Permite solo los caracteres a-z, A-Z, coma y espacio. Las cadenas compuestas por los caracteres a-z, A-Z, deben estar separadas por coma (,). • Publicado: Opcional. Campo de selección. Valor por defecto (marcado).
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha modificado satisfactoriamente el artículo. 2. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. 3. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.
<p>Prototipo de interfaz: No aplica</p>

Tabla 6: Historia de Usuario Eliminar Artículo. (Elaboración propia)

Número: 10	Nombre del requisito: Eliminar artículo
Programador: Frank Verdecia	Iteración Asignada: Primera Iteración
Prioridad: Media	Tiempo Estimado: 1.10h
Riesgo en Desarrollo: Medio	Tiempo Real: 1.15h

Descripción: Los usuarios autenticados con privilegios de editor o administrador podrán eliminar artículo en el sistema.
Observaciones: <ul style="list-style-type: none"> 1. El usuario debe estar autenticado 2. El usuario debe poseer los permisos necesarios para realizar la gestión del artículo.
Prototipo de interfaz: No aplica

2.4 Arquitectura

Al utilizar el CMS Drupal para el desarrollo del portal web, la arquitectura y los patrones de software a utilizar son heredados por el mismo. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una serie de elementos básicos. Estos son los nodos (*nodes*), módulos (*modules*), bloques y menús (*blocks & menus*), permisos de usuario y plantillas (*templates*). (Drupal.org, 2016)

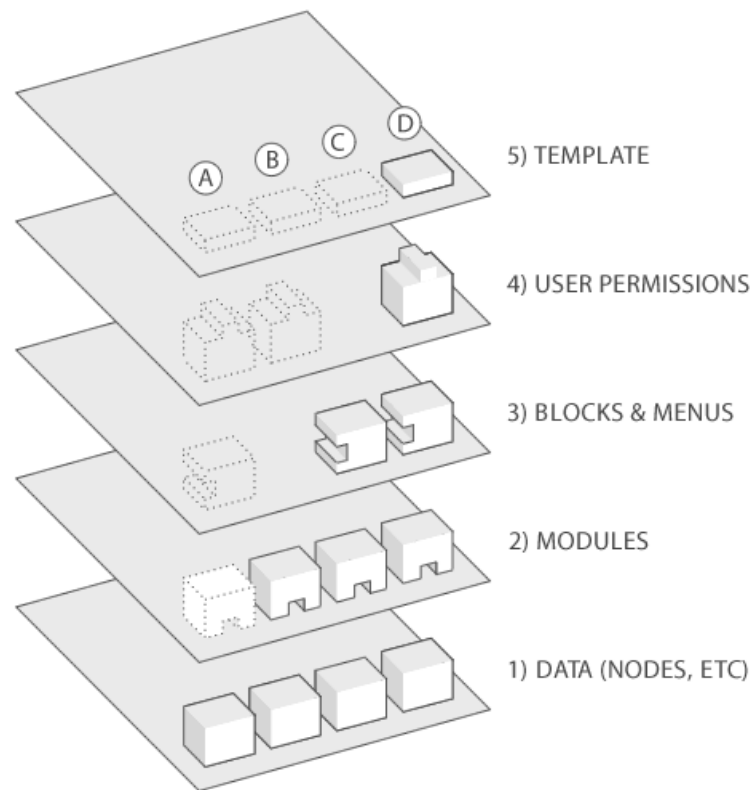


Figura 5. Arquitectura de la aplicación. (Drupal.org, 2016)

1. **Datos:** son los elementos básicos en que *Drupal* almacena la información y los contenidos. Así, a medida que el portal web crece, lo va haciendo el número de nodos los cuales van formando un “depósito de nodos” cada vez mayor, (tipos de contenido a publicar ejemplo noticias y eventos).
2. **Módulos:** Son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada portal. Esta es conocida como la segunda capa de la arquitectura del CMS (ejemplos los módulos *block* y *views*).
3. **Bloques y Menú:** Estos permiten estructurar y organizar los contenidos en el portal web. (ejemplo: Revistas y Artículos).
4. **Permisos de usuario:** La seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta funcionalidad *Drupal* dispone de un registro de

usuarios y de roles que permiten especificar que tareas pueden realizar y a que contenidos puede acceder cada tipo de usuario (ejemplos: roles de administrador, editor y usuario).

5. **Plantillas:** Es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del portal web sin necesidad de modificar los contenidos. Se compone fundamentalmente de plantillas HTMLX, CSS y PHP que mejoran la apariencia, diseño, fuente, color y estilos que por defecto utiliza el gestor. En la propuesta de solución esta capa contiene el tema *tema1*, así como su codificación en los lenguajes *JQuery* y PHP.

2.5 Patrones de diseño

El patrón de diseño es una descripción del problema y la esencia de su solución, de modo que la solución puede reutilizarse en diferentes configuraciones; no es una especificación detallada (Sommerville, 2005).

Los patrones representan una forma de describir las mejores prácticas de diseño para poderlos reutilizar, así mismo detallan aspectos relacionados con el diseño de los subsistemas, centrándose en aspectos más específicos de la aplicación; los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y para el diseño de interfaces.

2.5.1 Características de los Patrones de Diseño.

Los patrones de diseño ayudan a un diseñador a lograr un buen diseño más rápidamente, haciendo que sean más fáciles reutilizar diseños y arquitecturas, utilizando técnicas que ya han sido probadas anteriormente. Cada patrón de diseño se centra en un problema concreto, describiendo cuando y como utilizarlo. Las principales características de los patrones de diseño son (Tenazaca, 2016):

- Ser efectivo para resolver problemas similares.
- Reutilizable.
- Aplicable a diferentes problemas de diseño en distintas circunstancias
- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.

- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Los patrones de diseño son soluciones bien pensadas a problemas conocidos de la programación, ya que abstraen el comportamiento de un determinado problema en una configuración de componentes y están clasificados de acuerdo con el nivel de abstracción que tenga cada patrón.

Los patrones se clasifican en:

Patrones Creacionales: se encargan de la inicialización, configuración de objetos y creación de instancias. Resuelven problemas relacionados con la creación de instancias de objetos (Tedeschi, 2014).

Patrones estructurales: Se encargan de aislar la interfaz de la implementación, es decir, definen como las clases y objetos se asocian para formar estructuras más complejas (Tedeschi, 2014).

Patrones de comportamiento: Se encargan de la descripción de las clases y objetos y la comunicación e interacción entre ellos en tiempo de ejecución (Tedeschi, 2014).

2.5.2 Tipos de Patrones de Diseño.

Patrones GRASP:

Los patrones GRASP, más que patrones propiamente dichos, son una serie de “buenas prácticas” de aplicación recomendable en el diseño de software. Entre ellos, los patrones Experto, Creador, Bajo acoplamiento y Alta cohesión, estos guardan directa relación con la creación y asignación de responsabilidades a los objetos (Tabares, 2010).

Todo patrón tiene un nombre, plantea un problema y aporta una solución. Así, el patrón Experto plantea el problema “¿Cuál es el principio fundamental mediante el cual se asignan responsabilidades a los objetos?” y aporta la solución “Asignar la responsabilidad a la clase que tiene la información necesaria para cumplirla” (Larman et al., 1999).

El patrón **Experto** posibilita una adecuada asignación de responsabilidades facilitando la comprensión del sistema, su mantenimiento y adaptación a los cambios con reutilización de componentes. El patrón **Creador** aporta un principio general para la creación de objetos, una de las actividades más frecuentes en programación. El patrón **Bajo Acoplamiento** es una medida de la fuerza con que una clase se relaciona con otras, porque las conoce y recurre a ellas; una clase con bajo acoplamiento no depende de muchas

otras, mientras que otra con alto acoplamiento presenta varios inconvenientes: es difícil entender cuando está aislada, es ardua de reutilizar porque requiere la presencia de otras clases con las que esté conectada y es cambiante a nivel local cuando se modifican las clases afines. El patrón **Alta Cohesión** es una medida que determina cuán relacionadas y adecuadas están las responsabilidades de una clase, de manera que no realice un trabajo colosal; una clase con baja cohesión realiza un trabajo excesivo, haciéndola difícil de comprender, reutilizar y conservar (Tabares, 2010).

Patrones GOF

El catálogo de patrones más famoso es el contenido en el libro “*Design Patterns: Elements of Reusable Object-Oriented Software*”, también conocido como: El libro GOF (*Gang-Of-Four Book*). Según este documento (Gamma, 1997), estos patrones se clasifican por su propósito en creacionales, estructurales y de composición, mientras que respecto a su ámbito se clasifican en clases y objetos.

- **Instancia única (*Singleton*):** Este patrón está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un objeto único. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Dentro del core de Drupal se utiliza este patrón de diseño en diversas tareas como la gestión de manejo de conexiones con la base de datos y pensando en los módulos y temas de Drupal como objetos para llevar a cabo la gestión de dichos elementos (Buytaert, 2016)

Un ejemplo que evidenciará este patrón en el portal web Casa Editora Abril es el proceso de creación de tipos de contenidos como artículo, revista o libro, al cual se le asigna un identificador propio que evita la existencia en la base de datos de elementos repetidos constituyendo así una instancia única.

- **Decorador (*Decorator*):** Asigna responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Permite no tener que crear clases que hereden de la primera, incorporando nuevas funcionalidades, sino otras que la implementan y se asocian a ella (Drupal.org, 2016).

Drupal hace un amplio uso del patrón decorador. El polimorfismo de los objetos del nodo es muy amplio, pero esto es solo una pequeña parte del poder del sistema de nodos. Más interesante es el uso de varios ganchos de nodo (*hook_node_load ()*, *hook_node_view ()*, etc.), que permiten a los módulos arbitrarios extender el comportamiento de todos los nodos (Drupal.org, 2016).

- **Observador (*Observer*):** Define una dependencia entre objetos, de forma tal que cuando uno de los objetos cambie su estado el observador notifica este cambio y se actualizan todos los objetos que dependen de él. El objetivo principal es desacoplar la clase de los objetos clientes del objeto aumentando la modularidad del lenguaje (Drupal.org, 2016).

La interacción anterior también es similar al uso de observadores en sistemas orientados a objetos. Este patrón de Observador es generalizado en todo Drupal, ya que muchos de los ganchos de Drupal esencialmente permiten que los módulos se registren como observadores de los objetos de Drupal. Por ejemplo, cuando se realiza una modificación a un vocabulario en el sistema de taxonomía de Drupal, se llama a un gancho de taxonomía como *hook_taxonomy_vocabulary_update ()* en todos los módulos que lo implementan. Al implementar el gancho, los módulos se han registrado como observadores del objeto de vocabulario; cualquier cambio en él se puede actuar según corresponda (Drupal.org, 2016).

- **Orden (*Command*):** Encapsula una petición en un objeto, permitiendo así parametrizar a los clientes con diferentes peticiones, hacer cola o llevar un registro de las peticiones y poder deshacer las operaciones (Drupal.org, 2016).

Muchos de los ganchos de Drupal usan el patrón de Comando para reducir la cantidad de funciones que son necesarias para implementar, pasando la operación como un parámetro junto con los argumentos. De hecho, el sistema de gancho en sí usa este patrón, de modo que los módulos no tienen que definir cada gancho, sino solo los que les interesa implementar (Drupal.org, 2016).

- **Cadena de responsabilidades (*Chain of responsibility*):** Evita acoplar el emisor de una petición a su receptor, dando a más de un objeto la posibilidad de responder a la petición. Permite establecer una cadena de los objetos receptores a través de los cuales pasan una petición formulada por algún objeto emisor (Drupal.org, 2016).

El sistema de menús de Drupal sigue el patrón de la Cadena de responsabilidad. En cada solicitud de página, el sistema de menús determina si hay un módulo para manejar la solicitud, si el usuario tiene acceso al recurso solicitado y qué función se llamará para hacer el trabajo. Para hacer esto, se pasa un mensaje al elemento del menú correspondiente a la ruta de la solicitud. Si el elemento del menú no puede manejar la solicitud, se pasa por la cadena. Esto continúa hasta que un módulo maneja la solicitud, un módulo niega el acceso al usuario o la cadena se agota (Drupal.org, 2016).

2.6 Diagrama de clases del diseño con estereotipos web (DCD)

Un diagrama de clases del diseño con estereotipos *web* tiene el mismo objetivo o propósito que un diagrama de clases tradicional, con la particularidad de que se emplea para el modelado de aplicaciones *web* (Pressman, 2010). A continuación, se modela el DCD para la HU gestionar artículo.

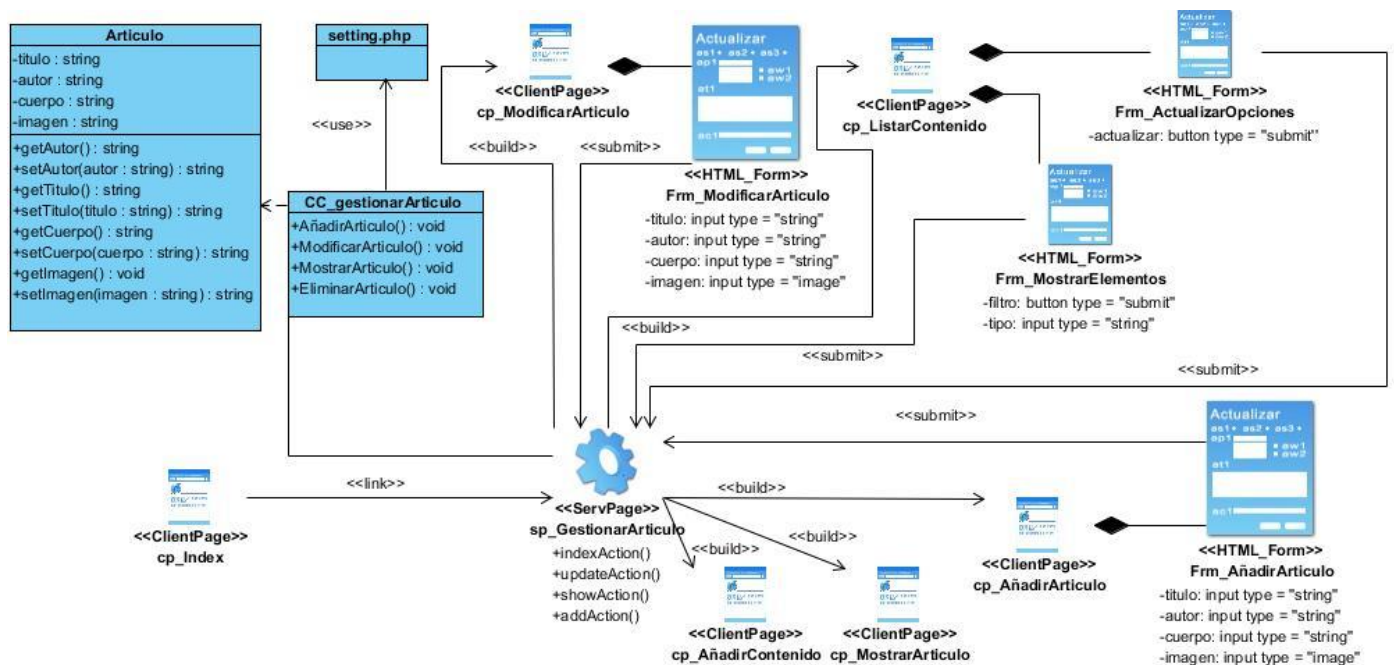


Figura 6. Diagrama de clases de la Historia de Usuario Gestionar Artículo. (Elaboración propia)

2.7 Diagrama de secuencia

El diagrama de secuencia es un tipo de diagrama del lenguaje unificado de modelado (UML) que, a su vez, se trata de un lenguaje orientado a objetos y está compuesto por elementos gráficos. Este representa los eventos en orden cronológico, razón por la que a veces se le llama diagrama de eventos o escenario de eventos. El diagrama de secuencia describe básicamente cómo los objetos (e instancias) intercambian mensajes en un orden determinado. Los objetos son los bloques de construcción básicos de los diagramas UML y representan ciertas características de un elemento del sistema, que varían dependiendo del

diagrama. En las interacciones, los objetos son las conocidas como líneas de vida (IONOS.es, 2019). A continuación, el diagrama de secuencia de la HU Añadir Artículo:

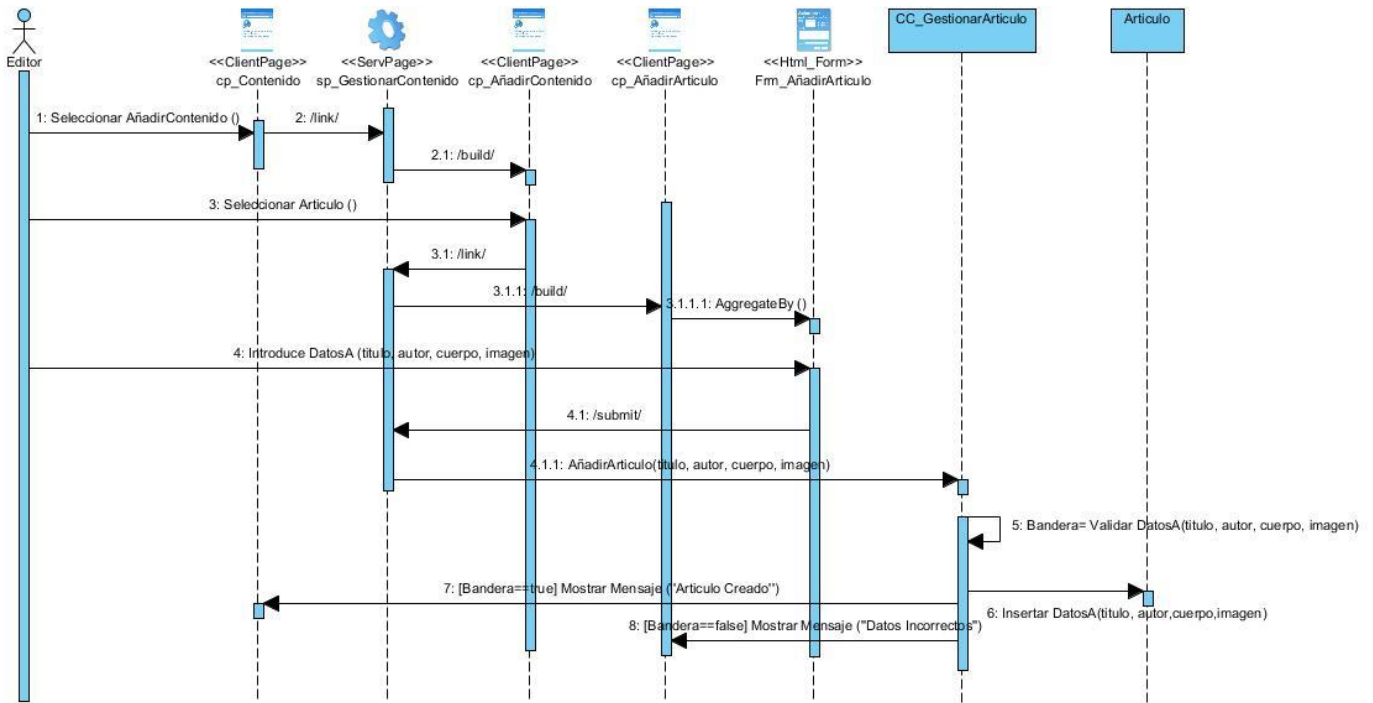


Figura 7. Diagrama de secuencia de la Historia de Usuario Añadir Artículo. (Elaboración propia)

2.8 Conclusiones parciales

- El modelo de diseño es esencial previo a implementar, permitiendo identificar las principales clases del sistema, sus métodos asociados y las relaciones existentes entre ellas, lo que posibilita un aumento de la reutilización y reducción del acoplamiento.
- Los requisitos funcionales y no funcionales identificados a partir del proceso de extracción de requisitos permiten desarrollar las distintas funcionalidades que debe cumplir el sistema para solucionar las necesidades requeridas.
- Los modelos y diagramas desarrollados facilitan un mejor entendimiento del sistema a implementar mostrando las relaciones existentes entre las diferentes clases, las funcionalidades presentes en estas y la ordenación temporal de los mensajes.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL PORTAL WEB CASA

EDITORIA ABRIL 2.0

En el presente capítulo se exponen los componentes usados en la implementación del sistema. Se revisan los requisitos antes definidos con el resultado de la implementación. Se realizan pruebas de software para verificar el correcto funcionamiento del sistema. El proceso de pruebas está dirigido a componentes del software, con el objetivo de medir el grado en que se cumplen los requisitos exigidos por el cliente y detectar la mayor cantidad de errores en el sistema de gestión para lograr su corrección.

3.1 Estándares de codificación de Drupal

Por código fuente se entiende todo texto legible por un ser humano y redactado en un lenguaje de programación determinado (IONOS.es, 2020).

Los estándares de codificación son un elemento fundamental en la implementación de proyectos, permitiendo que el código fuente generado sea fácil de leer y modificar independientemente de quien ha sido su autor. Son una guía para el equipo de desarrollo, permiten asegurar que el código presente alta calidad y no contenga errores.

Los estándares de codificación de Drupal son independientes de la versión y "siempre actualizados". Todo el código nuevo debe seguir los estándares actuales, independientemente de la versión (principal). El código existente en versiones anteriores puede actualizarse, pero no necesariamente tiene que ser así. Especialmente para bases de código más grandes (como el núcleo de Drupal), actualizar el código de una versión anterior para los estándares actuales puede ser una tarea demasiado grande. Sin embargo, el código en las versiones actuales debe seguir los estándares actuales (Drupal.org, 2021).

A continuación, ejemplos de algunos estándares de codificación de Drupal:

Etiquetas de código PHP

Utilice siempre `<? php?>` Para delimitar el código PHP, no la abreviatura `<? ?>`. Esto es necesario para el cumplimiento de Drupal y también es la forma más portátil de incluir código PHP en diferentes sistemas operativos y configuraciones (Drupal.org, 2021).

Indentación y espacios en blanco

- Utilice una sangría de 2 espacios, sin tabulaciones.
- Las líneas no deben tener espacios en blanco al final.
- Los archivos deben formatearse con `\n` como final de línea (finales de línea Unix), no `\r\n` (finales de línea de Windows).
- Todos los archivos de texto deben terminar en una sola línea nueva (`\n`). Esto evita la advertencia detallada del parche "`\ No hay una nueva línea al final del archivo`" y hace que los parches sean más fáciles de leer, ya que es más claro lo que se cambia cuando se agregan líneas al final de un archivo.
- Todos los bloques al principio de un archivo PHP deben estar separados por una línea en blanco. Esto incluye el bloque `/** @file */`, la declaración del espacio de nombres y las instrucciones de uso (si están presentes), así como el código subsiguiente en el archivo. Entonces, por ejemplo, el encabezado de un archivo podría tener el siguiente aspecto (Drupal.org, 2021):

```
<?php

namespace This\Is\The\Namespace;

use Drupal\foo\Bar;

/**
 * Provides examples.
 */
class ExampleClassName {
```

Figura 8. Estándares de codificación (Drupal.org, 2021).

O, para un archivo que no es de clase (por ejemplo, `.module`):

```

<?php

/**
 * @file
 * Provides example functionality.
 */

use Drupal\foo\Bar;

/**
 * Implements hook_help().
 */
function example_help($route_name) {

```

Figura 9. Estándares de codificación (Drupal.org, 2021).

Operadores

- Todos los operadores binarios (operadores que se encuentran entre dos valores), como +, -, =, !=, ==, >, etc. deben tener un espacio antes y después del operador, para facilitar la lectura. Por ejemplo, una asignación debe tener el formato \$ foo = \$ bar; en lugar de \$ foo = \$ bar ;. Los operadores unarios (operadores que operan en un solo valor), como ++, no deben tener un espacio entre el operador y la variable o número en el que están operando.
- Las comprobaciones de desigualdad de tipo débil DEBEN utilizar el operador !=. El operador <> NO DEBE usarse en código PHP (Drupal.org, 2021).

Nombres de archivo

- Todos los archivos de documentación deben tener la extensión de nombre de archivo ".txt" para facilitar su visualización en sistemas Windows. Además, los nombres de archivo para dichos archivos deben estar en mayúsculas (por ejemplo, README.txt en lugar de readme.txt), mientras que la extensión en sí está en minúsculas (es decir, txt en lugar de TXT) (Drupal.org, 2021).

3.2 Modelo de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema. El modelo de despliegue define la arquitectura física del sistema. Se usa para modelar de manera detallada los nodos físicos y las asociaciones de comunicación que existen entre ellos. Del mismo modo queda especificado qué hardware, sistemas operativos, software de interfaces y soporte. El modelo de despliegue 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.

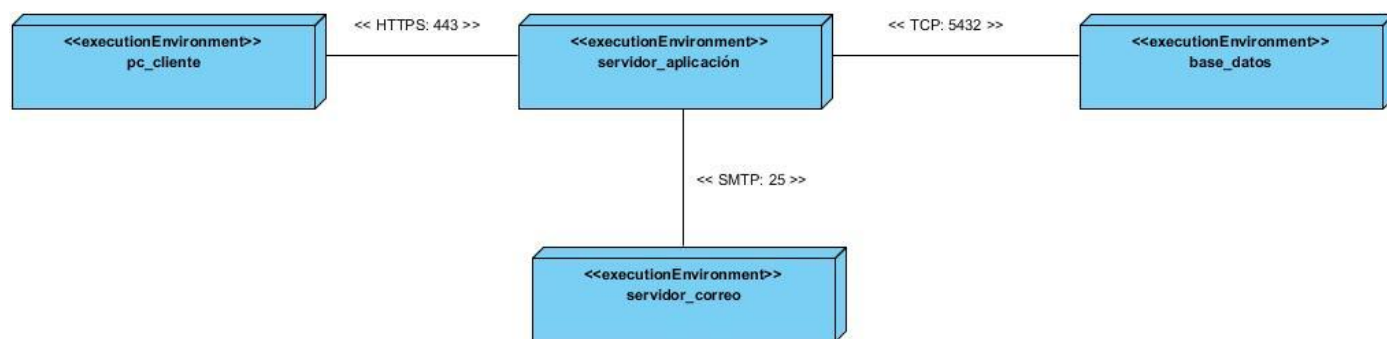


Figura 10. Modelo de despliegue (Elaboración propia)

pc_cliente: Es la estación de trabajo cliente que presenta un navegador *web* para conectarse a la aplicación hospedada en el servidor de aplicaciones utilizando el protocolo de comunicación *HTTPS*.

servidor_aplicacion: Es la estación de trabajo que hospeda el código fuente de la aplicación, les brinda a los usuarios las interfaces de la misma para realizar los procesos definidos por cada uno de los roles del sistema. Esta estación se comunica con el servidor de base de datos donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo *TCP*.

base_datos: Este servidor es el encargado del almacenamiento de los datos del sistema, y se comunica con el servidor de aplicaciones de dicho sistema.

servidor_correo: Este servidor es el encargado de la gestión de notificaciones a través del correo electrónico.

3.3 Diagrama de componentes

Los diagramas de componentes UML representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico.

En el contexto del UML, los componentes son partes modulares de un sistema independientes entre sí, que pueden reemplazarse con componentes equivalentes. Son autocontenidos y encapsulan estructuras de cualquier grado de complejidad. Los elementos encapsulados solo se comunican con los otros a través de interfaces. Los componentes no solo pueden proporcionar sus propias interfaces, sino que también pueden utilizar las interfaces de otros componentes, por ejemplo, para acceder a sus funciones y servicios. A su vez, las interfaces de un diagrama de componentes documentan las relaciones y dependencias en una arquitectura de software (IONOS.es, 2020).

Al encapsular los elementos, se impide el acceso directo a la estructura interna de los datos. Esto sirve, por ejemplo, para proteger los datos de accesos no autorizados. Se puede regular el acceso mediante interfaces diseñadas expresamente para ello, que solo muestran al usuario los métodos y elementos de datos de un objeto relevantes para él.

En conjunto, varios componentes forman una arquitectura de sistema más amplia. Los componentes pueden también contener otros componentes y basarse unos en otros, de modo que un componente puede presuponer la existencia de otros componentes (relación de dependencia). Asimismo, los módulos de software pueden ser aplicables a diferentes fases de realización: algunos componentes se utilizan principalmente para la planificación y la ingeniería de proyectos durante la fase de diseño, mientras que otros solo existen durante el tiempo de ejecución de un programa informático. En este contexto, también puede hablarse de los componentes de diseño y de tiempo de ejecución (IONOS.es, 2020) .

A continuación, el diagrama de componentes del portal web Casa Editora Abril:

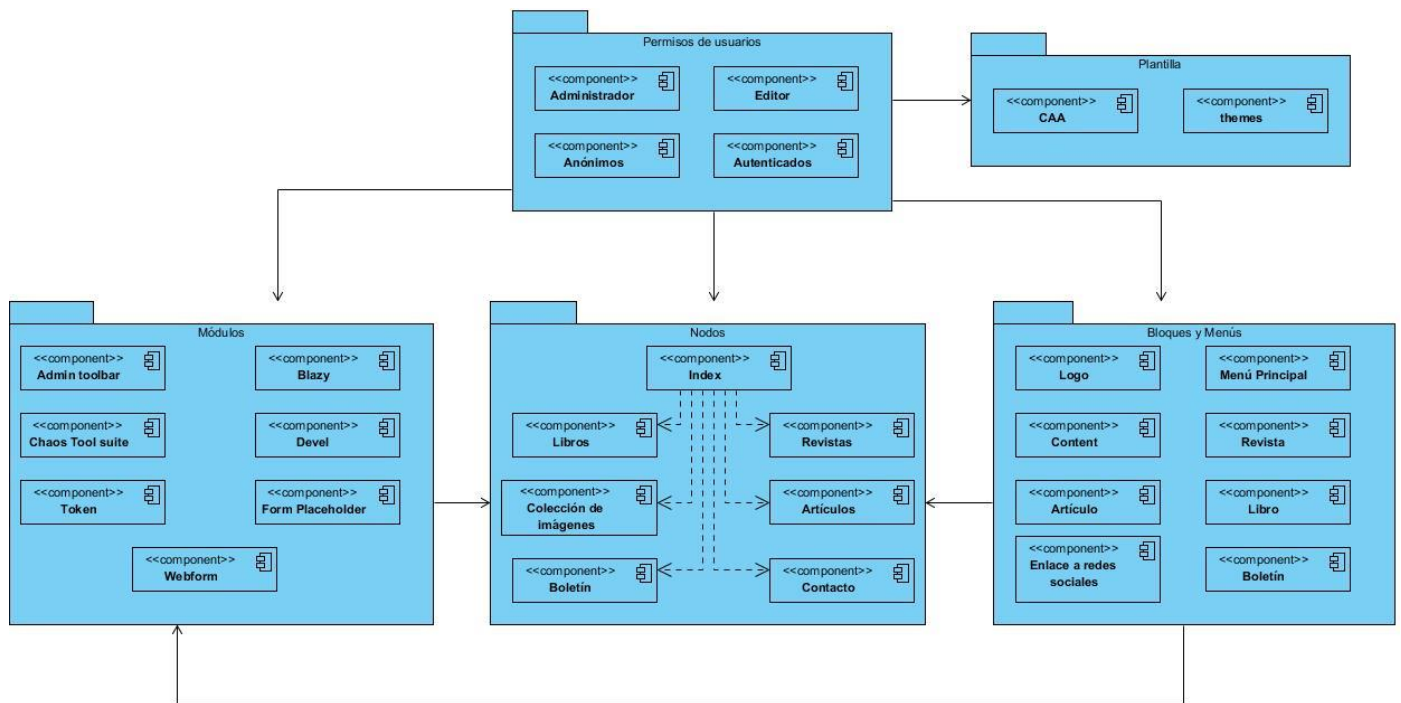


Figura 11. Diagrama de componentes. (Elaboración propia).

3.4 Estrategia de prueba

La calidad de software puede parecer un concepto alejado de la vida diaria de la mayoría de las personas; pero nada más lejos de la realidad. Cuando en nuestro ordenador aparece un mensaje de error, estamos ante un problema de calidad del software. Para las empresas, la calidad del software con que ellas operan es muy valorada, ya que éstos manejan información que debe protegerse de cualquier peligro que afecte la integridad de los datos. Con calidad nos referimos al cumplimiento de las expectativas de un producto.

Las normas ISO definen la calidad como "Grado en el que un conjunto de características inherentes cumple con los requisitos" y además como un "Conjunto de propiedades o características de un producto o servicio que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas" (ISO, 2015)

Cuando se dice que se tiene un software de calidad se puede entender que se puede modificar, que es confiable, eficaz y se deja usar fácilmente. Una de las tareas para asegurar estos resultados son las Pruebas. Las pruebas de software son las investigaciones empíricas y técnicas cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto (Myers et al., 2011).

Existen muchos tipos de pruebas al software que verifican que el software funcione como esperamos. Debe ser verificado el cumplimiento de las especificaciones planteadas al inicio del proyecto por el analista o el propio cliente, y/o eliminar los posibles errores que se puedan haber cometido en cualquier fase del desarrollo. Hoy es considerada, en muchos aspectos, una de las etapas críticas dentro del ciclo de vida del software.

3.4.1 Pruebas funcionales

Las pruebas funcionales están dirigidas a verificar la corrección y la completitud de una función. Dichas pruebas utilizan para su diseño las técnicas basadas en la especificación o técnicas de caja negra.

La **prueba de caja negra** es una técnica de pruebas de software que verifica la funcionalidad sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software. Su objetivo es demostrar que las funciones del software son operativas, las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene. Esta prueba solo se enfoca en las entradas y salidas del sistema, basándose en los requerimientos del software y las especificaciones funcionales. Permite obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (ISTQB, 2018).

Estas pruebas permiten encontrar:

- Funciones incorrectas.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

A continuación, se muestran dos casos de prueba a las Historias de Usuarios Adicionar Artículo y Modificar Artículo:

Tabla 7: Caso de prueba Añadir Artículo. (Elaboración propia)

Caso de prueba Añadir Artículo.	
Código de caso de prueba: 1	Nombre de Historia de Usuario:

	Añadir artículo
Nombre de la persona que realiza la prueba: Frank Verdecia Blanco	
Descripción de la prueba: Prueba a la funcionalidad añadir artículo	
Entrada / Pasos de la ejecución: La entrada consta de la introducción de los datos en los siguientes campos: <ul style="list-style-type: none"> • Título • Cuerpo • Autor • Imagen 	
Resultado esperado: Se añade la noticia	
Evaluación de la prueba: Satisfactorio	

Tabla 8: Caso de prueba Añadir Artículo. (Elaboración propia)

Caso de prueba Modificar Artículo.	
Código de caso de prueba: 2	Nombre de Historia de Usuario: Modificar artículo
Nombre de la persona que realiza la prueba: Frank Verdecia Blanco	
Descripción de la prueba: Prueba a la funcionalidad modificar artículo	

Entrada / Pasos de la ejecución: Se modifican los siguientes datos para editar un contenido de tipo artículo.

Datos actuales

- **Título:** “La casa editora abril como fuente de conocimiento”
- **Cuerpo:** El portal web de la Casa Editora Abril en sus páginas abre un amplio abanico de temas dirigidos al enriquecimiento de la cultura política e ideológica, ética, histórica, científica y ecologista.
- **Autor:** Jimmy Suárez Pupo
- **Imagen:** <http://www.editorabril.cu/content/uploads/2021/logo.png>

Datos después de modificados:

- **Título:** “La casa editora abril como fuente de conocimiento”
- **Cuerpo:** El portal web de la Casa Editora Abril en sus páginas abre un amplio abanico de temas dirigidos al enriquecimiento de la cultura política e ideológica, ética, histórica, científica y ecologista, resaltando los valores del socialismo, el antiimperialismo, el amor a la obra de la Revolución y a los sentimientos de identidad nacional y de solidaridad.
- **Autor:** Jimmy Suárez Pupo
- **Imagen:** <http://www.editorabril.cu/content/uploads/2021/logo.png>

Resultado esperado: Se modifica la noticia

Evaluación de la prueba: Satisfactorio

Análisis de los resultados de las pruebas funcionales

Para validar que el sistema cumpla con las funciones específicas para las cuales ha sido creado se realizan las pruebas funcionales que arrojan como resultados en una primera iteración 18 no conformidades de las cuales 10 fueron resueltas. En la segunda iteración quedaron pendientes 8 de las cuales se resolvieron 6 y

en la tercera iteración, de las 2 restantes, se resolvieron las 2. Se realiza una cuarta iteración la que no arroja no conformidades.

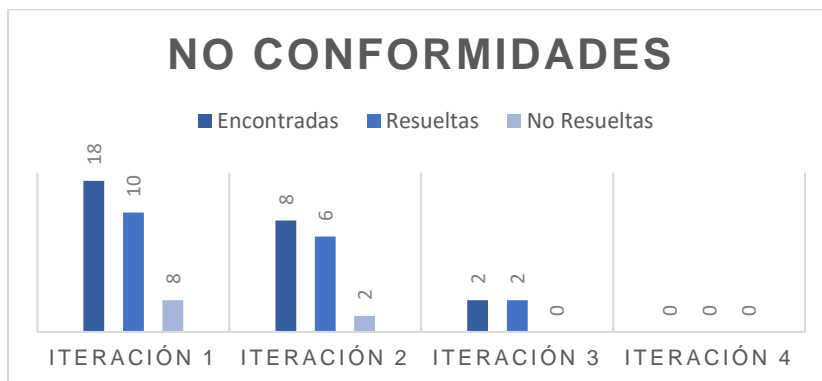


Figura 12. Resultado de iteraciones de las pruebas funcionales. (Elaboración propia).

Entre las no conformidades detectadas en el proceso de pruebas funcionales se encuentran:

- Errores de funcionamientos en módulos: se instalan módulos con versiones anteriores o se sustituyen por otros con funciones parecidas adaptadas al portal.
- Errores de estructuración en los contenidos mostrados en las vistas: se soluciona con una reestructuración de los contenidos.
- Errores de funcionamiento en algunos bloques: se arreglan instalando módulos y correcciones a estos.
- Errores ortográficos.

3.4.2 Pruebas seguridad

Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones web, buscando disminuir el impacto de ataques a ellas y pérdida de información importante (Diaz Diaz, 2014).

La seguridad en aplicaciones web busca asegurar la confidencialidad, disponibilidad e integridad de los datos y funciones que maneja el software, teniendo en cuenta el impacto que pueden tener fallas de seguridad según el contexto empresarial (Islam & Dong, 2008)

Para la realización de las pruebas de seguridad se planea utilizar la herramienta Acunetix WVS, que divide la criticidad de los errores en 4 niveles (Alto, Medio, Bajo e Informativo).

Acunetix es una herramienta automatizada para seguridad de aplicaciones web capaz de escanear sitios web y detectar vulnerabilidades como Inyección SQL, Cross Site Scripting y ataques XSS. El escáner de vulnerabilidades web de Acunetix utiliza sus rastreadores (*crawlers*) para analizar los directorios de tu sitio web. De esta manera es capaz de auditar tus sistemas de forma automática para detectar anomalías. Esta también permite el acceso a estadísticas y reportes bien estructurados de diferentes tipos, que permitirán analizar la evolución de la seguridad de aplicaciones web, priorizar amenazas y recibir recomendaciones acerca de cómo arreglar las fallas de seguridad (Acunetix, 2021).

Análisis de los resultados de las pruebas de seguridad

Con el objetivo de evaluar la seguridad de la solución propuesta la herramienta Acunetix WVS arrojó los siguientes resultados:

Tabla 9: Resultado pruebas de seguridad. (Elaboración propia)

Categorías de vulnerabilidades	Cantidad de errores
Vínculos rotos	2
Ataque de adivinación de contraseñas en la página de inicio de sesión	1
Las credenciales de usuario se envían en texto claro	5
Secuencias de comandos entre sitios	1
Total	8

La prueba realizada mediante la herramienta Acunetix WVS, arrojó que en la primera iteración se detectan 9 no conformidades, de ellas 1 de riesgo bajo, 6 alertas de riesgo medio y 2 alertas de riesgo alto. Estas deficiencias fueron corregidas en la primera iteración. Para proporcionar una mayor seguridad se configuró el servidor de forma tal que solo pudieran acceder al sistema los usuarios con permiso, así como la instalación de módulos para mayor seguridad, tales como *secure_login*. En una segunda iteración no se identificaron nuevas no conformidades, obteniéndose así una herramienta que cumple con los requisitos de seguridad.

3.4.3 Pruebas rendimiento (carga y estrés)

Las pruebas de rendimiento se diseñan para asegurar que el sistema pueda procesar su carga esperada. Éstas se ocupan tanto de demostrar que el sistema satisface sus requerimientos, como de descubrir problemas y defectos en el sistema (Sommerville, 2011).

Las pruebas de carga consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sistema esté funcionando, con el fin de detectar si el software instalado (programas y aplicaciones) cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware (servidor y el equipamiento computacional de redes y enlace que lo conecta a Internet) es capaz de soportar la cantidad de visitas esperadas (Pressman, 2005). Las pruebas de estrés evalúan la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan defectos potencialmente peligrosos (Pressman, 2005).

Para las pruebas de rendimiento se utilizará el software Apache Jmeter v5.4.1. Las propiedades de las PC a utilizar serían la siguientes:

Hardware de prueba (PC servidor)

- Sistema Operativo: Linux Mint v.20.2
- Microprocesador: Intel(R) Core (TM) i3-5005U CPU @2.00GHz 2.00GHz
- Memoria RAM: 4.00 GB
- Disco Duro: 1024 GB

Hardware de prueba (PC cliente):

- Sistema Operativo: Windows v.10
- Microprocesador: Intel(R) Core (TM) i5-7300HQ CPU @3.00GHz 2.00GHz
- Memoria RAM: 8.00 GB
- Disco Duro: 1024 GB
- Tipo de Sistema: Sistema operativo de 64 bits

Software instalado en ambas PC:

- Tipo de servidor web: Apache 2.4 (PC cliente).

- Plataforma: SO Windows (PC cliente) y SO Linux (PC servidor)
- Servidor de BD: MySQL 5.7.24 (PC cliente).

Luego de definido el hardware se configuran los parámetros del Apache JMeter logrando tres ambientes de simulación con un total de 25, 50, y 100 usuarios conectados concurrentemente, se realiza peticiones a diferentes páginas del portal web Casa Editora Abril.

Análisis de los resultados de las pruebas de rendimiento

Se utiliza la siguiente tabla para los resultados:

Tabla 10: Resultado prueba de rendimiento. (Elaboración propia)

Usuarios	Muestras	Media	Mediana	Min	Max	Línea 90%	%Error	Rend	Kb/s
25	100	3039	1236	195	5783	1735	0.00%	3.8/sec	14.4
50	125	2637	1185	104	4795	2567	0.00%	2.6/sec	5.9
100	250	2555	1077	85	4587	2075	0.00%	2.0/sec	4.1

Para un mejor entendimiento de las pruebas de rendimiento, se explica cada parámetro a continuación:

- **Usuarios:** total de usuarios.
- **Muestras:** el número de peticiones.
- **Media:** El tiempo medio transcurrido en milisegundos para un conjunto de resultados.
- **Mín:** El mínimo tiempo transcurrido en milisegundos para las muestras de la URL dada.
- **Máx:** El máximo tiempo transcurrido en un milisegundo para las muestras de la URL dada.
- **% Error:** Porcentaje de las peticiones con errores.
- **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.
- **Kb/s Recibidos:** Rendimiento medido en *Kbytes* por segundos.

De los resultados obtenidos en las pruebas se determina que la aplicación cumple con los requisitos que se presentan. El informe de Apache JMeter arroja que el portal web es capaz de responder 100 peticiones de 25 usuarios conectados simultáneamente en un tiempo promedio de 3039 milisegundos con un error de 0.0% esto evidencia que el portal puede responder correctamente las peticiones realizadas.

Se realizan 125 peticiones de 50 usuarios conectados simultáneamente con un tiempo promedio de 2637 milisegundos con un error de 0.0% demostrándose que el portal es capaz de responder correctamente las peticiones realizadas.

Por último, se realiza una prueba de estrés con un mayor de 100 usuarios conectados simultáneamente respondiendo una muestra de 250 peticiones en un tiempo promedio de 2555 milisegundos con un porcentaje de error de 0.0% demostrándose que el portal es eficaz a la hora de responder las peticiones de los usuarios.

3.5 Conclusiones parciales

- Al aplicar los estándares de codificación se facilita la comprensión y legibilidad del código fuente, pudiendo permitir desarrollar el portal en un menor tiempo y garantizando la facilidad y calidad de un futuro mantenimiento.
- La elaboración del modelo de despliegue permite describir la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos
- La realización de las pruebas de funcionalidad, seguridad y rendimiento (carga y estrés) facilita la calidad de la aplicación y correcta ejecución de las funcionalidades del sistema.

CONCLUSIONES

Con la culminación del presente trabajo de diploma se cumplieron los objetivos trazados, distinguiéndose de manera general los siguientes aspectos:

1. El estudio de los elementos teóricos vinculados a la problemática a resolver, en conjunto con el estudio de los sistemas homólogos permitieron sentar las bases teóricas, definiendo funcionalidades y características.
2. El análisis de las tecnologías, herramientas y metodologías utilizadas en la actualidad, así como la selección de los requisitos funcionales y no funcionales que se implementan constituyen elementos necesarios asociados al desarrollo del portal web de la Casa Editorial Abril.
3. Las pruebas que se realizan al portal web permiten verificar el funcionamiento, rendimiento y garantizar la seguridad del mismo.

RECOMENDACIONES

Se proponen las siguientes recomendaciones:

1. Trabajar en ampliar las funcionalidades a medida que surjan nuevas necesidades.

REFERENCIAS BIBLIOGRÁFICAS

Acunetix. (2021). *Product*. Acunetix. <https://www.acunetix.com/product/>

Anna Pérez. (2016). *¿Qué son las metodologías de desarrollo de software?* OBS Business School.
<https://www.obsbusiness.school/blog/que-son-las-metodologias-de-desarrollo-de-software>

Apache. (2021). *About the Apache HTTP Server Project—The Apache HTTP Server Project*.
https://httpd.apache.org/ABOUT_APACHE.html

Apache JMeter. (2021). *Apache JMeter—Apache JMeter™*. <https://jmeter.apache.org/>

Aubry, C. (2014). *HTML5 y CSS3: Para sitios con diseño web responsive*. Ediciones ENI.

Barba Soler, J. P. (2015). *Diseño y desarrollo web. Análisis de casos* [PhD Thesis]. Universitat Politècnica de València.

Béjar Cáceres, D. G. (2017). *Creación de un sitio web con interfaz adaptiva a móviles mediante el cms Drupal* [B.S. thesis].

Bootstrap, M. O., Jacob Thornton, and Bootstrap. (2021). *Bootstrap*. <https://getbootstrap.com/>

Buytaert, D. (2016). *Drupal.org*. Drupal.Org. https://www.drupal.org/docs/user_guide/en/index.html

Díaz Díaz, S. M. (2014). *Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software*.

Drupal.org. (2015, octubre 28). *About*. Drupal.Org. <https://www.drupal.org/about>

Drupal.org. (2016, noviembre 14). *Drupal User Guide*. Drupal.Org.
https://www.drupal.org/docs/user_guide/en/index.html

Drupal.org. (2021). *Coding standards*. Drupal.Org. <https://www.drupal.org/docs/develop/standards/coding-standards>

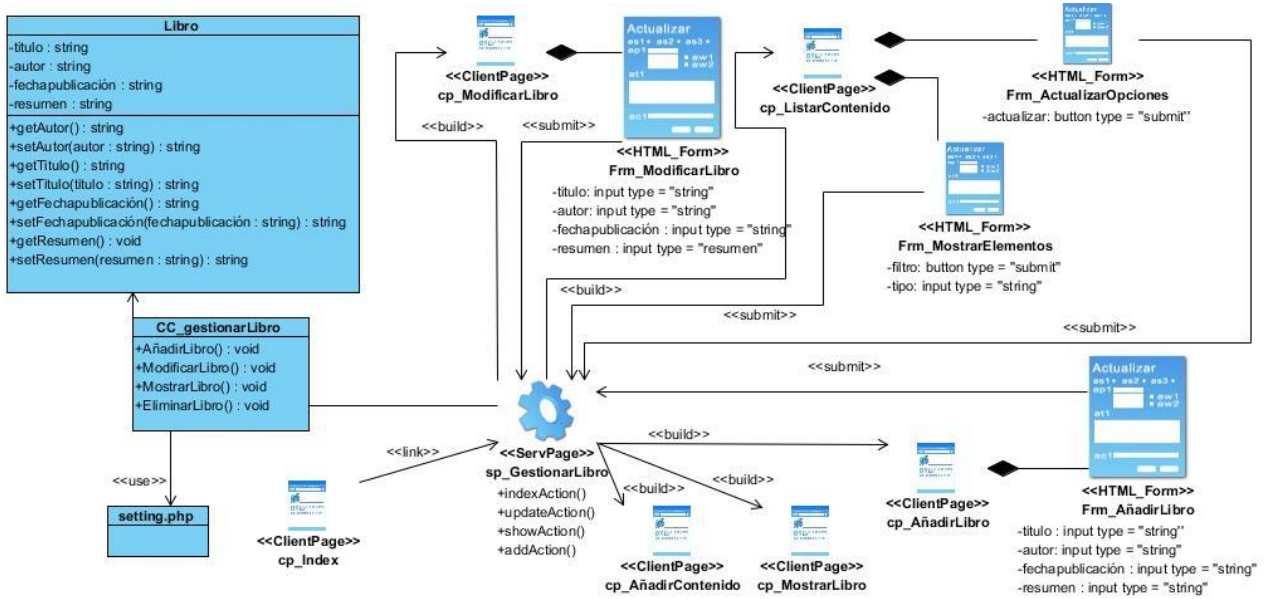
- Gamma, E. (1997). Going beyond objects with design patterns. *European Conference on Object-Oriented Programming*, 530-530.
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Marcombo.
- Ibarra, O. (2011). *La web. Antecedentes, problemática actual, clasificación y funciones*. 1-11.
<https://doi.org/10.6084/m9.figshare.14907963.v1>
- IONOS.es. (2018). *UML, lenguaje de modelado gráfico*. IONOS Digitalguide.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>
- IONOS.es. (2019). *Diagramas de secuencia: Mostrar interacciones con UML*. IONOS Digitalguide.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagramas-de-secuencia/>
- IONOS.es. (2020). *Diagrama de componentes UML: explicación, creación y ejemplo—IONOS*.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes/>
- IONOS.es. (2020). *El código fuente: ¿qué es y cómo se escribe?* IONOS Digitalguide.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/codigo-fuente-definicion-con-ejemplos/>
- IONOS.es. (2020). *Introducción al sistema gestor de base de datos (SGBD)*. IONOS Digitalguide.
<https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/sistema-gestor-de-base-de-datos-sgbd/>
- Islam, S., & Dong, W. (2008). Security requirements addressing security risks for improving software quality. *TUM*, 31.
- ISO. (2015). *ISO 9001:2015(es), Sistemas de gestión de la calidad—Requisitos*.
<https://www.iso.org/obp/ui/#iso:std:iso:9001:ed-5:v1:es>
- ISTQB. (2018). *Programa de estudios Nivel Básicos*.

- Larman, C., Rodríguez, L. M. H., & Anaya, H. C. (1999). *UML y Patrones: Introducción al análisis y diseño orientado a objetos* (Vol. 2). Prentice Hall.
- Latorre, M. (2018). Historia de las web, 1.0, 2.0, 3.0 y 4.0. *Universidad Marcelino Champagnat*, 1-8.
- Mcgraw-hill.es. (2018). *Unidad 2 Sistemas gestores de contenidos*. Mcgraw-hill.es. <https://www.mheducation.es/bcv/guide/capitulo/8448183924.pdf>
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.
- MySQL. (2021). *MySQL :: Why MySQL?* <https://www.mysql.com/why-mysql/>
- PHP. (2021). *PHP: ¿Qué puede hacer PHP? - Manual*. <https://www.php.net/manual/es/intro-whatcando.php>
- phpMyAdmin, phpMyAdmin. (2021). *PhpMyAdmin*. PhpMyAdmin. <https://www.phpmyadmin.net/>
- Platzi. (2017). *Qué es Frontend y Backend, diferencias y características—Platzi*. <https://platzi.com/blog/que-es-frontend-y-backend/>
- Pressman, R. S. (2005). *Software engineering: A practitioner's approach*. Palgrave macmillan.
- Rios, R. P., Maridueña, J., Betancourt, R., & Mosquera, F. (2017). Sistema basado en tecnología web para promocionar a productores agrícolas en la zona rural de la ciudad de Babahoyo, provincia de los ríos. *Revista Caribeña de Ciencias Sociales*, 2017_02.
- Rodríguez Sánchez, T. (2015). Metodología de desarrollo para la Actividad productiva en la UCI. *Universidad de las Ciencias Informáticas: Programa de mejora*.
- Ruedas, J. G. (2016). *Dirección y gestión de proyectos de tecnologías de la información en la empresa*. FC Editorial.
- Singh, T., & Rao, J. (2016). A Study of Web portal features As a Knowledge Management System in School Education. *Int Online Multidiscip J*, 1-3.
- Sommerville, I. (2005). *Ingeniería del Software*. Pearson Educación S.A.
- Sommerville, I. (2011). *Software engineering 9th*.

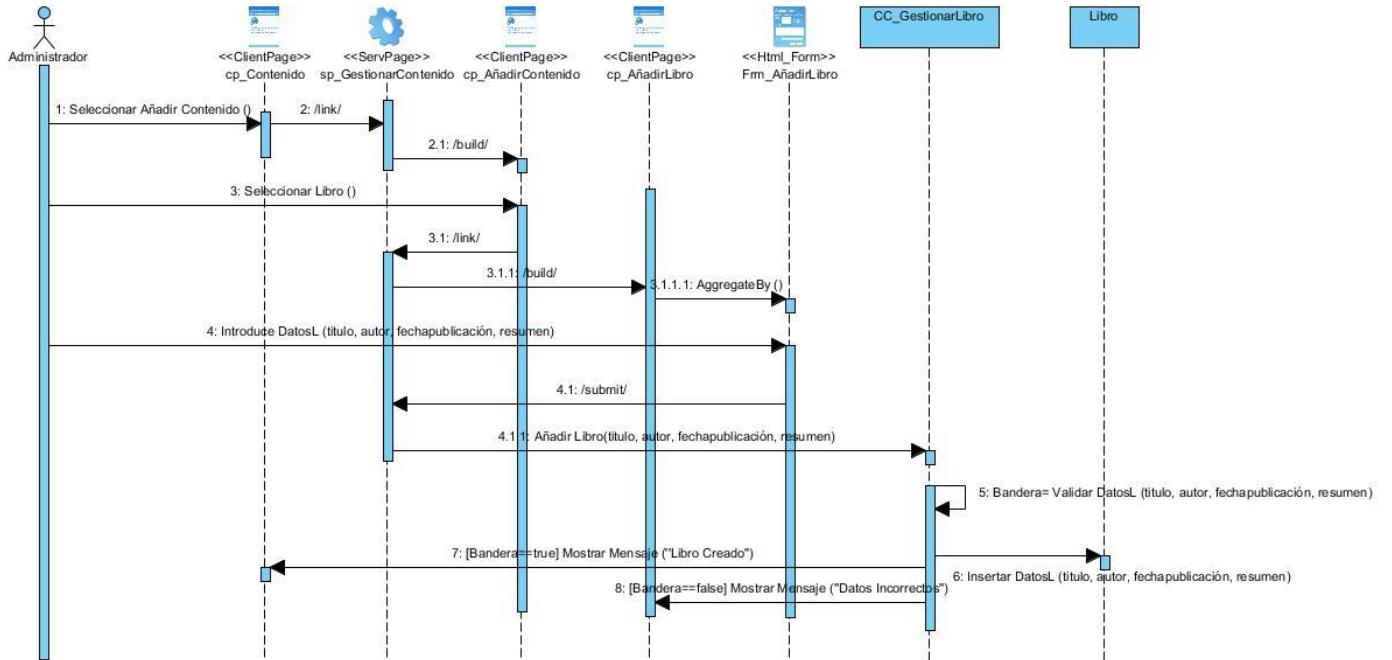
- Tabares, R. B. (2010). Patrones grasp y anti-patrones: Un enfoque orientado a objetos desde logica de programacion. *Entre Ciencia e Ingeniería*, 4(8), 161-173.
- Tedeschi, N. (2014). Microsoft Developer Network. *línea*. Available: <http://msdn.microsoft.com/eses/library/bb972240.aspx>. [Último acceso: 26 Mayo 2014].
- Tenazaca, M. A. (2016). *Consideraciones de Arquitectura de Software a nivel de Diseño Arquitectónico y Desarrollo de Software para minimizar vulnerabilidades en aplicaciones web basados en OWASP Top Ten 2013, caso de estudio arquitectura: SOA*. 134.
- Visual Paradigm. (2021). 1. Introduction of Visual Paradigm. *Visual Paradigm Community Circle*. <https://circle.visual-paradigm.com/docs/introduction-and-system-requirements/introduction-of-visual-paradigm/>

ANEXOS

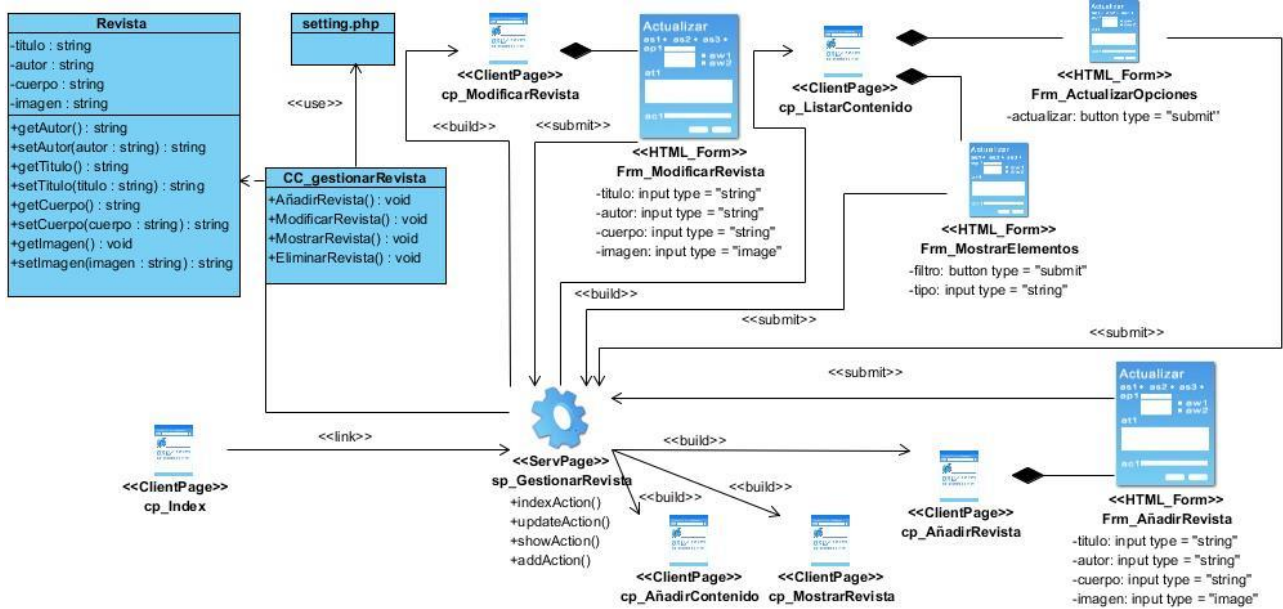
Anexo 1: Diagrama de clases con estereotipos web para la HU Gestionar Libro



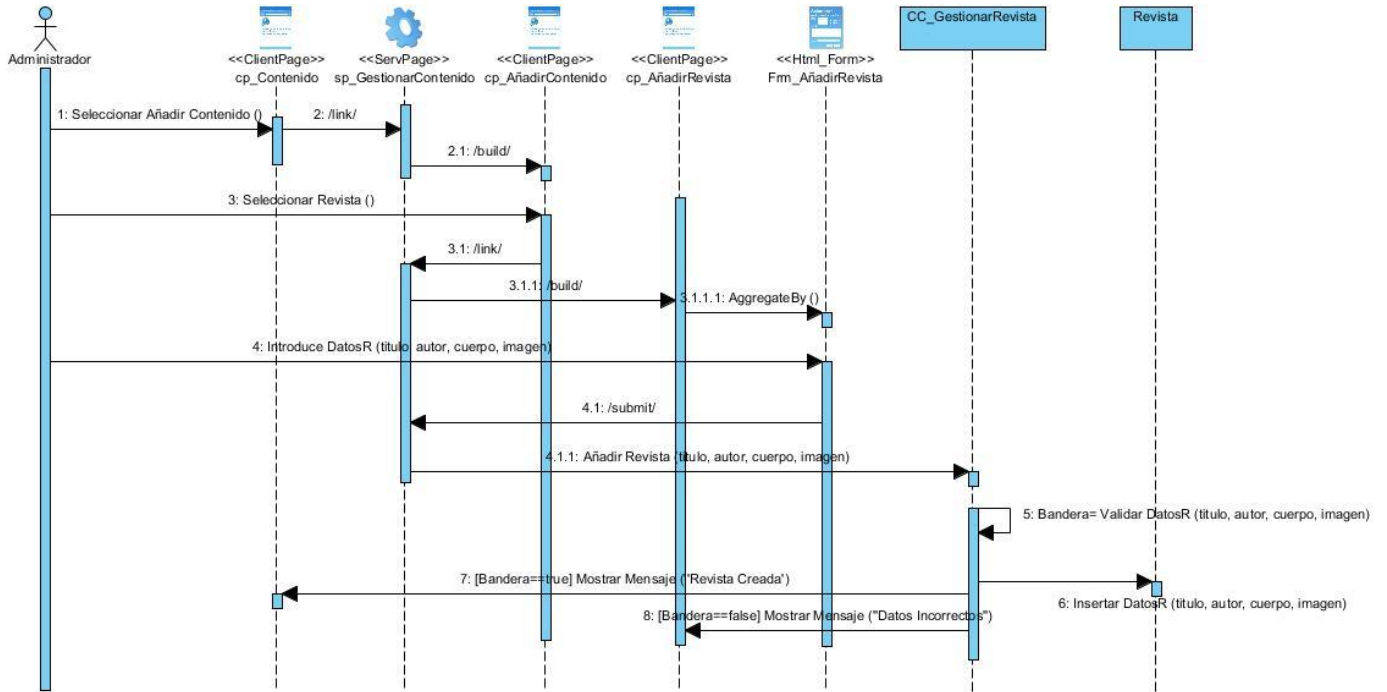
Anexo 2: Diagrama de secuencia para la HU Añadir Libro



Anexo 3: Diagrama de clases con estereotipos web para la HU Gestionar Revista



Anexo 4: Diagrama de secuencia para la HU Añadir Revista



Anexo 5: Vista sección Revistas

Revistas



Juventud Técnica

Miércoles, Julio 21, 1965 - 20:33

Revista Científico-Técnica Popular

Revista que divulga los últimos avances de la ciencia y la tecnología del quehacer cubano e internacional.



Caimán Barbudo

Martes, Octubre 25, 1966 - 21:31

En sus páginas se abordan temáticas relacionadas con todas las manifestaciones artísticas dentro y fuera de Cuba. Prestigiosas plumas han publicado en esta revista.



Zunzún

Viernes, Octubre 10, 1980 - 20:29

Revista Infantil dedicada a las niñas y niños de la enseñanza primaria. Su nombre Zunzún, un ave pequeña, intranquila y alegre, como son los niños.



Alma Mater

Miércoles, November 15, 1922 - 21:55

Revista de los universitarios cubanos, fundada en noviembre 1922, por Julio Antonio Mella, que la convirtió en la voz de la FEU.



Pionero

Sábado, Julio 21, 1990 - 20:22

Revista que ha dedicado más 40 años a la educación de las niñas, niños y adolescentes cubanos en los valores de nuestra sociedad.



Somos Jóvenes

Jueves, Marzo 6, 1997 - 21:14

Fundada en 1977, con su versión impresa y digital, tiene el propósito de dar a conocer cómo viven, estudian, trabajan, piensan, aman y sueñan los jóvenes cubanos.

Anexo 6: Vista sección Libros

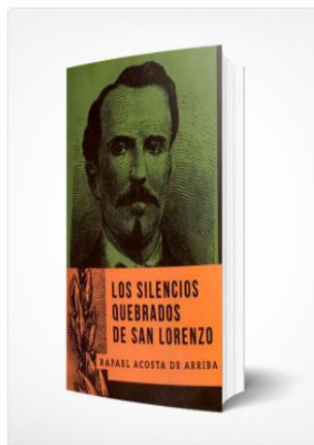
Libros



La Fuente Viva

Miguel Barnet

Nueva edición ampliada y corregida por el autor, su concepto de novela-testimonio e indagación en nuestras raíces culturales.



Los silencios quebrados de San Lorenzo

Rafael Acosta

Los silencios quebrados de San Lorenzo, texto de ensayos y estudios críticos sobre el Padre de la Patria Carlos Manuel de Céspedes