



Facultad de Ciencias y Tecnologías Computacionales

# **Desarrollo de un Portal Web de la Unión de Jóvenes Comunistas para el municipio de la Lisa**

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

**Autor:**

Alejandro Nuñez Rivas

**Tutores:**

Msc. Bienvenido Hanley Roque Orfe

Ing. Darvis Dorvigny Dorvigny

La Habana, noviembre de 2022

Año 64 de la Revolución

**DECLARACIÓN DE AUTORÍA**

Declaro por este medio que Alejandro Nuñez Rivas, soy el autor del Trabajo de Diploma Sistema para la gestión de préstamos y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración de autoría en La Habana a los \_\_\_\_\_ días del mes de diciembre del año 2022.

\_\_\_\_\_  
Firma del Autor

**Alejandro Nuñez Rivas**

\_\_\_\_\_  
Firma del Tutor

**Bienvenido Hanley Roque  
Orfe**

\_\_\_\_\_  
Firma del Tutor

**Darvis Dorvigny Dorvigny**

## FRASE

"Solo cultivo una pequeña parte de los alimentos que como, pero no planté ni perfeccioné esas semillas. Visto ropa que otras personas han hecho. Hablo un idioma que no inventé ni perfeccioné. Uso matemáticas que no descubrí. Estoy protegido por libertades y leyes que no concebí ni ayudé a legislar, y yo tampoco me aseguro de su cumplimiento ni de su adjudicación. Me emociona la música que yo no com-puse. Cuando necesité atención médica, no pude ayudarme a mí mismo a sobrevivir. Yo no inventé el transistor, el microprocesador, la programación orientada a objetos, ni la mayoría de las tecnologías con las que trabajo. Amo y admiro al ser humano, tanto a los vivos como a los muertos, y soy completamente dependiente de ellos para mi vida y mi bienestar".

**Steve Jobs**

## AGRADECIMIENTOS

A veces pensamos que merecemos grandes cosas, solo por hecho de ser. Es en este punto que la vida me ofrece la oportunidad de hacer pública mis palabras. Deseo expresar lo agradecido que estoy con mis padres por guiarme en mi formación como persona y acompañarme ahora como profesional.

Mi madre la reina de mis lágrimas y la dueña de mis sonrisas, gracias por verme como soy, por desvelarte incluso hoy cuando mañana de seguro no estoy.

Mi padre, a semejante héroe quiero agradecer, demostrándome cada día lo que valgo, lo que puedo hacer, por siempre estar presente, por inculcarme buenos valores y por impulsarme a ser mejor cada día, por todo su amor y preocupación.

A mi hermano que hace algunos años estuvo parado donde yo lo estoy ahora, se que no puede escuchar esto pero desde lo lejos siempre supiste cuidarme, protegerme y amarme como solo los verdaderos amores saben hacer. Por ser mi amigo, mi estabilidad emocional y esta de mas decir que este día también es para el.

A mis tutores Bienvenido y Darvis por ayudarme en este trabajo de formación como ingeniero y demostrarme que se puede con todo, a pesar de no contar con el tiempo suficiente siempre fui tratado con la mejor dedicación.

A todos lo educadores que han aportado en mi preparación como ingeniero. Muchas Gracias.

**DEDICATORIA**

A mi madre por su apoyo incondicional, ser un ejemplo de lucha que siempre me motiva a seguir adelante. Por ser la mejor madre del mundo.

A mi padre por ser mi mayor referente, por estar siempre presente y no dejar que tire la toalla cuando piense que todo está perdido.

A mi hermano por ser mi amigo, mi cómplice, el oyente de mis historias de vida a pesar de la distancia.

Mis amigos por ser mi familia incondicional.

## RESUMEN

La Universidad de las Ciencias Informáticas (UCI) tiene entre sus principales objetivos la producción de software encaminado a automatizar los procesos que se realizan en cada organización política. La Unión de Jóvenes Comunistas (UJC) organización en la que militan casi todos los jóvenes de la universidad y el municipio también requiere de los beneficios que brinda la Informatización. En esta investigación se dio solución a la deficiencia que posee la organización con respecto a la informatización de sus procesos, puesto que la información de la organización solo es conocida a través de reuniones o por correo electrónico, lo que dificulta la disponibilidad e integridad de la misma. Con el objetivo de solucionar esta situación el presente trabajo propone desarrollar un portal web para la UJC de la Lisa. Se realiza un profundo estudio del negocio, estructura y funcionamiento de la UJC, se investiga acerca de las principales tendencias en el modelado de software, las herramientas y metodologías actuales, se justifica la propuesta de herramientas para la implementación del sistema además de desarrollarse el análisis y diseño de la aplicación con sus respectivas pruebas que proveen de calidad al sistema.

**PALABRAS CLAVE:** sistema de gestión, organización, acta, militante.

**ÍNDICE**

INTRODUCCIÓN..... 6

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS..... 10

    Conceptos Básicos:..... 10

    1.1- Estado del arte..... 10

    1.2- Análisis de sistemas informáticos vinculados al campo de acción..... 11

    1.3- Metodologías de desarrollo de software..... 13

    1.4- Pruebas a realizar al sistema..... 16

    1.5- Arquitectura de software..... 18

    1.5.1- Estilos arquitectónicos..... 18

    1.5.2- Framework de desarrollo..... 20

    1.6- Lenguajes de programación..... 22

    1.6.1- PHP 8.0..... 22

    1.6.2- jQuery 2.1.4..... 23

    1.7- Herramientas de desarrollo..... 24

    1.7.1- Herramienta CASE..... 24

    1.7.2- Entorno de desarrollo integrado (IDE)..... 25

    1.7.3- Sistema Gestor de Bases de Datos (SGBD)..... 26

    Conclusiones del capítulo..... 28

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA..... 29

    2.1- Propuesta de solución..... 29

        Modelo de Dominio..... 29

    2.2- Especificación de requisitos..... 31

    2.2.1- Descripción de los requisitos de software (Historia de Usuarios)..... 37

    2.3- Análisis y Diseño..... 39

    2.4- Modelado de Diseño..... 43

    2.4.1- Diagramas de clases del diseño con estereotipo web (DCD)..... 43

    2.4.2- Diagramas de Secuencia..... 44

---

Conclusiones del capítulo.....	45
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	47
3.1- Modelo de despliegue.....	47
3.2- Diagrama de componentes.....	48
3.3- Estándares de codificación de WordPress.....	50
3.3.1 PHP.....	50
3.3.2 HTML.....	52
3.3.3 CSS.....	52
3.3.4 JavaScript.....	53
3.4- Estrategia de Validación del portal web UJC de La Lisa.....	54
3.4.1- Pruebas de Rendimiento.....	54
3.4.2- Pruebas Unitarias.....	57
3.4.3- Pruebas de Seguridad.....	61
3.4.4- Pruebas de Funcionales.....	62
3.4.5- Pruebas de Usabilidad.....	67
3.4.6- Pruebas de Aceptación.....	68
3.5 Criterios de expertos.....	70
Conclusiones del capítulo.....	72
CONCLUSIONES GENERALES.....	73
RECOMENDACIONES.....	74
REFERENCIA BIBLIOGRAFÍA.....	75
ANEXOS.....	78

**ÍNDICE DE TABLA**

<b>Tabla 1.</b> Comparación entre la metodología ágil y tradicional.....	16
<b>Tabla 2.</b> Requisitos Funcionales.....	33
<b>Tabla 3.</b> Crear Acta.....	39
<b>Tabla 4.</b> Crear Noticia.....	41
<b>Tabla 5.</b> Resultado de las pruebas de Carga y Estrés.....	57
<b>Tabla 6.</b> Caso de Prueba de la ruta independiente 1.....	61
<b>Tabla 7.</b> Caso de prueba basado en el RF “Crear Acta” .....	65
<b>Tabla 8.</b> Resultado de las pruebas de usabilidad.....	67
<b>Tabla 9.</b> Cuadro Lógico de ladov para usuarios potenciales.....	68
<b>Tabla 10.</b> Satisfacción de usuarios potenciales.....	69
<b>Tabla 11.</b> Validación de las variables de investigación.....	71

---

**ÍNDICE DE FIGURA**

<b>Figura 1</b> . Arquitectura HEADLESS.....	42
<b>Figura 2</b> . Diagrama de Clase del Diseño para HU (Crear Noticia).....	45
<b>Figura 3</b> . Diagrama de Secuencia para HU (Crear Noticia).....	46
<b>Figura 4</b> . Modelo de Despliegue.....	49
<b>Figura 5</b> . Diagrama de Componentes.....	50
<b>Figura 6</b> . Código PHP.....	51
<b>Figura 7</b> . Uso de espacio en PHP.....	51
<b>Figura 8</b> . Condiciones Yoda.....	52
<b>Figura 9</b> . Selectores de código CSS.....	53
<b>Figura 10</b> . Orden de Prioridades de código CSS.....	54
<b>Figura 11</b> . Método utilizado como ejemplo para la técnica de ruta básica.....	59
<b>Figura 12</b> . Grafo de flujo de método get(self, request, *args, **kwargs).....	60
<b>Figura 13</b> . Resultados de las Pruebas Funcionales.....	66
<b>Figura 14</b> . Satisfacción de usuarios potenciales con el sitio web.....	70

### INTRODUCCIÓN

La gestión de la información y el conocimiento, las tecnologías, así como las personas que en ello intervienen, existen desde los albores de la humanidad, cuando el hombre en su afán de comunicación, utilizaba diferentes técnicas para dejar constancia, intercambiar, transmitir y compartir sucesos y apreciaciones de su quehacer y modo de vida, a través de pinturas rupestres, papiros y otros soportes. Esto le permitió de alguna manera gestionar información y por ende generar el conocimiento en sus coterráneos y su descendencia. En la medida en que transcurre el tiempo, aparecen nuevas formas y tecnologías de intercambio producto del desarrollo humano (Vidal LMJ, 2012).

En el desarrollo de la informática, uno de los campos en los que más han influido las tecnologías es en la gestión de la información. La gestión de documentos electrónicos juega un papel muy importante, ya que desarrolla tecnologías para tareas como el manejo de documentos en formato digital, la conversión a dicho formato de los documentos de papel. El proceso de gestión de la información en las organizaciones políticas del país es amplio y realizado por personas que se encargan de recopilar la información procedente de sus homólogos en la base. Esta información debe ser revisada, auditada, analizada y comparada con los datos archivados para, por último, almacenarla para próximas consultas que se quieran realizar, con la eliminación de los riesgos que trae consigo el deterioro de los documentos por su uso (Alexeis García, 2011).

Con la creación de Internet, las empresas e instituciones vieron su surgimiento como una gran oportunidad para el intercambio de información y conocimiento. Además, se eliminaba la limitación de sólo interactuar en un área geográfica en específico, ya que se puede navegar desde cualquier parte del mundo, hacia lugares muy distantes. Con este fin se crean los portales web, cuya función principal es: servir de puerta de entrada única para ofrecer al usuario, de forma fácil e integrada, el acceso a los recursos, contenidos y servicios que brinda (PISCITELLI, 2005).

La Unión de Jóvenes Comunistas (UJC) constituida el 4 de abril de 1962, es la organización política de la juventud cubana, creada con el fin de asegurar la unidad de los jóvenes cubanos, movilizarlos en torno a la Revolución Socialista y contribuir a su educación política. El ingreso es de carácter voluntario y selectivo. En ella militan jóvenes que estén entre los 14 y 30 años de edad. Su trabajo se proyecta más allá de sus miembros, está dirigida a toda la población joven del país. Su principal objetivo es la formación integral de cada joven.

El municipio de la Lisa agrupa a la juventud de vanguardia de la sociedad cubana y cuenta con un porcentaje elevado de miembros entre estudiantes, profesores y trabajadores. Desde el propio surgimiento de la institución, las estructuras de la UJC han necesitado utilizar las potencialidades que las tecnologías de la información y las comunicaciones ofrecen, los documentos de actas ordinarias son las que proceden de las reuniones mensuales de cada comité de base (CB), al igual en los comités primarios (CP) donde se generan de manera manual los reportes que constituyen el listado de acuerdos tomados en cada una de las reuniones realizadas, como son, reporte sobre sanciones, rendiciones de cuenta, listado de ausentes, de presentes, por ciento de asistencia, la participación de estructuras superiores. Independientemente de la realización de las actas, la calidad de las mismas se ve afectada debido a que los secretarios en muchas ocasiones no cuentan con los modelos adecuados para el tipo de reunión que están realizando, toda esa documentación debe perdurar en el tiempo debido que en caso crítico es necesario revisar más de 200 archivos mensualmente. Esta descentralización de la información provoca que exista más de un responsable de la información y que se archive en diferentes lugares, impidiendo el acceso a la misma y retrasando así los análisis estadísticos, que de manera muy periódica se ven afectados debido a la gran cantidad de documentos, el número de actividades y el intercambio entre los miembros de la organización que conviven en el municipio.

La Universidad de las Ciencias Informáticas (UCI) entre sus objetivos de trabajo tiene la tarea de convertir el municipio de La Lisa como ciudad inteligente, encaminados a automatizar los procesos que se realizan en las organizaciones políticas y gubernamentales y darle visibilidad en la red para que divulguen su quehacer diario. Es de gran importancia que la UJC en el municipio de la Lisa no cuenta con ninguna aplicación para la gestión de la información que en ella se maneja y para dar visibilidad a la misma.

A partir de lo antes expuesto se plantea como **problema de la investigación**: ¿Cómo gestionar la información asociada a los procesos de la UJC de la Lisa, de manera que aumente la calidad de la información y contribuya a la toma de decisiones?

Por tanto, la investigación presenta como **objeto de estudio** el proceso de gestión de información de la UJC, enmarcado como **campo de acción** informatización de la gestión de los procesos de la UJC en la Lisa.

Se define como **objetivo general**, desarrollar un portal web para la UJC del municipio la Lisa que permita la gestión de información, divulgación y promoción del trabajo de la organización.

Para lograr el objetivo general se plantean los siguientes **objetivos específicos**:

- Realizar un estudio del arte de los principales referentes teóricos de los sistemas de gestión de información para elaborar nuestro marco teórico.
- Realizar la identificación de requisitos, análisis y diseño del sistema como aproximación a la investigación.
- Diseñar un portal web para la UJC del municipio la Lisa teniendo en cuenta las funcionalidades y servicios con los que debe contar.
- Implementar un sistema informático que responda a las características del diseño planteado.
- Validar la solución desarrollada mediante la realización de pruebas de software.

Como posible resultado de la implementación y dándole cumplimiento al objetivo general de la investigación se obtendrá un portal web para la UJC del municipio de la Lisa, que permitirá la divulgación y promoción del trabajo de la misma. Para dar cumplimiento a lo antes abordado se proponen una serie de tareas a desarrollar en el transcurso de la investigación:

- 1- Caracterización de las tendencias actuales de los portales web para la difusión del trabajo de organizaciones principalmente las políticas.
- 2- Identificación de requisitos funcionales y no funcionales para el desarrollo del portal web.
- 3- Validación de los requisitos aplicando las técnicas definidas.
- 4- Selección de las tecnologías, herramientas, estándares, patrones y metodologías que permitan desarrollar una solución.
- 5- Implementación de las funcionalidades del portal web a desarrollar.
- 6- Aplicación de pruebas funcionales, rendimiento, seguridad, usabilidad, unitarias y de aceptación a las funcionalidades de la solución desarrollada.

### **Métodos teóricos:**

**Histórico-lógico:** se empleará durante la investigación en los antecedentes y las tendencias actuales de los sistemas de gestión de información, además la profundización de los conceptos y términos.

**Analítico-sintético:** se utilizará en el estudio de las bibliográficas para realizar la fundamentación teórica de la investigación y sintetizar los conceptos fundamentales que sean necesarios para la solución del problema.

### **Método empírico:**

**Entrevista:** se realizará entrevistas a dirigentes de la UJC para que faciliten la información necesaria para realizar el diseño e implementación del software

**Observación:** este método se utilizará con la finalidad de caracterizar y analizar detalladamente cómo se realiza actualmente el proceso de gestión de información, identificando los mecanismos para desarrollar un sistema acorde con las necesidades.

Este documento está estructurado en introducción, tres capítulos, conclusiones, recomendaciones, referencia bibliográfica, bibliografía, anexos y glosario de términos.

- **Capítulo 1** “Fundamentos Teóricos” se abordan los aspectos relacionados con los procesos de gestión de la información, además de la vinculación de estos con los portales web y la presentación de dicha información a través de este tipo de herramientas. Se presentan los lenguajes de modelado y de programación, se describen además tecnologías y metodologías que se emplean para llegar a la solución. Se realiza un estudio de los sistemas homólogos a la investigación y sus principales características.
- **Capítulo 2** “Características del Sistema” en el presente capítulo se realiza la descripción y diseño del modelado de las funcionalidades del portal web de la Unión de Jóvenes Comunista de la Lisa. Muestra las relaciones de los diferentes artefactos generados según la metodología definida.
- **Capítulo 3** “Validación de la solución Propuesta” en este capítulo se detalla la propuesta de solución al problema planteado. Se describe la organización del módulo en un diagrama de componentes y se especifican los estándares de codificación a utilizar. Además, se describen las pruebas realizadas para la validación del portal web y se exponen las interfaces como parte del resultado fin.

## **CAPÍTULO 1: FUNDAMENTOS TEÓRICOS**

En este capítulo se realizará una representación conceptual sobre el tema donde se proporcionará la información básica relacionada con el campo de acción, centrándose en el estudio de los sistemas de gestión de información de las organizaciones políticas y de masa, así como del estado del arte. Además, el estudio de sistemas similares, lenguajes de programación, sistemas gestores de base de datos, la metodología y las herramientas a emplear en solución al problema.

### **Conceptos Básicos:**

- Militante: miembro activo de una organización.
- Acta: no es más que un testimonio escrito de los hechos ocurridos en cualquier circunstancia, una comisión directiva de cualquier entidad
- Servidor web: son usados para servir Páginas Web solicitadas por ordenadores clientes

### **1.1- Estado del arte**

El mundo de las tecnologías evoluciona constantemente. El objetivo fundamental de la mayoría de las empresas a nivel mundial para lograr mayor desarrollo es la informatización. Existen muchas empresas que se dedican a proveer software para la gestión de estructuras organizativas en instituciones u organizaciones, así como la gestión de información de personas y el control de altas y bajas de las mismas.

### **Sistema de gestión de la información**

La gestión de la información es la denominación convencional de un conjunto de procesos por los cuales se controla su ciclo de vida. Un sistema de información se puede definir técnicamente como un conjunto de componentes relacionados que recolectan o recuperan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones, en el contexto de las organizaciones la cual se puede identificar como la disciplina que se encargaría de todo lo relacionado con la obtención de la información adecuada existen tres actividades necesarias para controlar operaciones, analizar problemas y crear nuevos servicios (Jesús Alioska, 2018). Estas actividades son:

- Entrada: captura o recolecta datos en bruto tanto del interior de la organización como de su entorno.

- **Procesamiento:** convierte esa entrada de datos en una forma más significativa.
- **Salida:** transfiere la información procesada a la persona que la usará o a las actividades para las que se utilizará.

Los sistemas de información también requieren retroalimentación, que es la salida que se devuelve al personal adecuado de la organización para ayudarle a evaluar o corregir la etapa de entrada.

### **Aplicaciones web**

El objetivo de desarrollar el sistema como una aplicación web es facilitar su administración, evitando ocupar espacio en discos duros y problemas de compatibilidad ya que estos solo necesitarían un navegador para acceder a la aplicación donde se van actualizando los cambios realizados, posibilitando un alto grado de comunicación.

Las aplicaciones web son programas informáticos que se ejecutan en un servidor web, al que los usuarios acceden a través de Internet, utilizando un navegador. Todos los datos con los que trabaja están almacenados en la web y son procesados en la misma, por lo que, a diferencia de las aplicaciones de escritorio, no necesitan instalarse en el ordenador, pero sí precisan de conexión a la red (Luján-Mora, 2012).

Las aplicaciones web están basadas en la **arquitectura cliente-servidor**. Combinan scripts del lado del servidor (PHP y ASP), para la gestión del almacenamiento y recuperación de la información, con scripts del lado del cliente (JavaScript y HTML), para ofrecer la información a los usuarios. Esto hace que en ellas se pueda interactuar a través de formularios online, sistemas de gestión de contenido, carritos de la compra, etc. (Aplicaciones Web).

### **1.2- Análisis de sistemas informáticos vinculados al campo de acción**

No existen antecedentes de sistemas a nivel internacional relacionados con la UJC; esta es una organización que solo radica en Cuba y en el ámbito nacional los sistemas que existen que pudieran aportar a esta investigación son en las universidades y Joven Club que se dedicaban a informar y tener noticias actualizadas del funcionamiento de la organización, sin embargo, se estudiaron algunos sistemas con propósitos similares de los cuales se realiza un análisis a continuación:

#### **Internacional:**

- Sistema de Gestión Escolar Guaicaipuro: **SIGE** es un sistema informático que fue diseñado para facilitar el control y registro, de todos los datos de los estudiantes y talento humano que labora en cada plantel educativo. Este programa permite al Ministerio del Poder Popular para la Educación de Venezuela concentrar toda la información a nivel nacional de los estudiantes que hacen vida en cada una de las instituciones, registro de notas de cada asignatura, número de estudiantes por grado, entre otras informaciones relacionadas con el plantel (SIGE, 2022).
- Sitio **PSOE** (Partido Socialista Obrero Español): Sitio principalmente informativo, aunque realiza gestión de algunos procesos, como la afiliación y el manejo de los afiliados en su estructura. Además, ofrece información sobre la mayoría de las actividades que se llevan a cabo dentro de la organización como los resultados electorales, biografías de sus dirigentes, documentos, noticias (PSEO, 2022).

### Nacional

En la Universidad de Ciencias Informáticas:

- Análisis y diseño de una aplicación informática para la gestión de información de la UJC en la Lisa: Es un trabajo de diploma que recoge el análisis y diseño de una aplicación informática para la gestión de información de la UJC en la Lisa, con el propósito de que posteriormente sirviera como base para la implementación del mismo. Este trabajo se adopta por parte del proyecto y del cliente con la idea de realizar un nuevo diseño e incorporar algunas ideas y procesos fundamentales que se debían poner en práctica.
- Sistema integral de gestión de la **UJC nacional**: módulo militante: Es un trabajo de diploma que presenta similitudes con el presente trabajo, sin embargo, presenta características diferentes en cuanto a la gestión de las estructuras debido a la reciente actualización del reglamento y los estatutos de la organización. Además, la aplicación que se propone como resultado será una aplicación de escritorio, la cual no cumple con las expectativas del cliente.
- Sistema informático para la gestión de la información de la UJC de la Facultad 2: Es un trabajo de diploma que presenta similitudes con el presente trabajo, sin embargo, presenta diferencias en cuanto a las funcionalidades que se quieren gestionar y aunque fue presentado como proyecto de tesis en el 2009, hasta nuestros días no se tiene conocimiento sobre su implantación o puesta en práctica en la UCI, ni que se haya continuado su mantenimiento o soporte.

Estos sistemas no cumplen con los requerimientos necesarios para la informatización de los procesos de la UJC, lo cual va en contra del paradigma de independencia tecnológica por la cual apuesta el país y son muy costosos; se gestionan de manera diferente procesos fundamentales de la organización ya que luego del cambio del reglamento y de los estatutos, estos cambiaron también, al igual que algunos modelos de documentos que variaron su estructura y/o contenido y por último, algunos de estos sistemas no fueron funcionales y no cumplieron los objetivos para los que fueron previstos teniendo en cuenta que no se conoce prácticamente de su existencia. Tales motivos avalan la necesidad de crear un sistema que cumpla con los requerimientos necesarios para una buena gestión de los efectivos de la organización y el manejo de los datos referentes de cada uno de ellos y que se ajuste a las necesidades del cliente.

### **1.3- Metodologías de desarrollo de software**

“Se define metodología de desarrollo de software como un conjunto de procedimientos, técnicas, herramientas y un soporte documental para el desarrollo de productos de software” (Pressman - 2005). La Metodología de Desarrollo de Software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. En un proyecto de desarrollo de software la metodología ayuda a definir: quién, cuándo y cómo debe hacerlo. La metodología para el desarrollo de software un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende actividades a seguir para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado (Chimbote U.C, 2017).

### **Metodologías ágiles**

Las metodologías ágiles presentan como principal particularidad la flexibilidad, los proyectos en desarrollo son subdivididos en proyectos más pequeños, incluye una comunicación constante con el usuario, son altamente colaborativos y es mucho más adaptable a los cambios. De hecho, el cambio de requerimientos por parte del cliente es una característica especial, así como también las entregas, revisión y retroalimentación constante (Cadavid, Fernández Martínez, & Morales Vélez, 2013).

Entre las principales metodologías ágiles están XP (eXtreme Programming), Scrum, Iconix, Crystal Clear, AUP (Agile Unified Process), entre otras. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan.

## Metodologías tradicionales

Según (Pressman, 2013), las metodologías de desarrollo tradicionales o clásicas son también llamados modelos de proceso prescriptivo, y fueron planteadas originalmente para poner orden en el caos del desarrollo de software que existía cuando se empezó a generar masivamente. La historia revela que estos modelos tradicionales que fueron presentados en la década de los 60, dieron cierta estructura útil al trabajo de la Ingeniería de software y constituyen un mapa razonablemente eficaz para los equipos de desarrollo.

**Tabla 1.** Comparación entre la metodología ágil y tradicional.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un control prefijado
El Cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo
Grupos pequeños (<10 integrantes) y trabajadores en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura de software	La arquitectura de software es esencial y se expresa mediante modelos

**Fuente:** Elaboración propia.

Después de haber estudiado las anteriores metodologías, el equipo de desarrollo decide utilizar la metodología AUP debido a que guía el ciclo de vida del proyecto para asegurar un producto de calidad. Se selecciona en consecuencia la metodología AUP-UCI teniendo en cuenta que es la metodolo-

gía adaptada al ciclo de vida de los proyectos productivos de la Universidad, es ampliamente usada en el área y es extremadamente flexible al proceso de desarrollo de software (Pérez Gómez, 2017).

### **AUP**

El proceso unificado ágil es un enfoque de modelado híbrido creado por Scott Ambler al combinar el Proceso unificado racional (RUP) con los métodos ágiles (AM). Al combinar ambos, Ambler creó un marco de proceso sólido que se puede aplicar a todo tipo de proyectos de software, grandes o pequeños. Los métodos ágiles proporcionaron valores, principios y prácticas a AUP (Edeki, 2013). Una apropiada implementación de la solución propuesta requiere la selección de una metodología que oriente el proceso de desarrollo del proyecto para garantizar un producto de calidad. De esta forma se decidió utilizar una variación de la metodología AUP para los proyectos productivos de la UCI.

### **Variación AUP-UCI**

Esta es una metodología adaptada al ciclo de vida definido para la actividad productiva de la UCI. Para obtener un software de calidad, la misma se apoya en el Modelo CMMI-DEV v1.3, el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. AUP-UCI define para el ciclo de vida de los proyectos 7 disciplinas tal como las define AUP, pero a un nivel más atómico. La metodología define 4 escenarios para desarrollar los requisitos. De estos, se selecciona el escenario 4, el cual modela el sistema mediante historias de usuario. De las 4 fases que propone AUP mantiene la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes 3 fases de AUP en una sola, llamada Ejecución y se agrega la fase de Cierre (Díaz Llanes, 2016).

A continuación, se presentan las 3 fases:

**Inicio:** esta etapa se mantiene de AUP, pero se modifica el objetivo de la misma; realización de la planeación del proyecto en cuanto a un estudio de la organización del cliente, el alcance del proyecto, estimaciones de tiempo esfuerzo y costo que serán necesarios para decidir si se desarrolla el producto o no.

**Ejecución:** en esta fase se recogen las actividades que desarrolla AUP de elaboración, construcción y transición. Se ejecutan las actividades requeridas para obtener el software y ajustar los planes del proyecto, realizando las tareas de modelado del negocio, obtención y análisis de los requisitos, elaboración de la arquitectura, diseño, implementación y liberación de producto.

**Cierre:** en esta fase se analizan los resultados del proyecto, su ejecución y se llevan a cabo actividades formales de cierre del proyecto.

#### **1.4- Pruebas a realizar al sistema**

Las pruebas son un elemento crítico para la calidad del software. La importancia de los costos asociados a los errores, promueve la definición y aplicación de un proceso de pruebas minuciosas y bien planificadas. Las pruebas permiten validar y verificar el software, entendiendo como validación del software, el proceso, externo al equipo de desarrollo, que determina si el software satisface los requisitos y verificación como el proceso interno que determina si los productos de una fase satisfacen las condiciones de dicha fase (Pressman, 2005).

El sistema, puede probarse de dos formas:

- ✓ conociendo la función específica para la que fue diseñado (pruebas de caja negra)
- ✓ conociendo el funcionamiento del producto (pruebas de caja blanca)

Según Pressman, las pruebas de caja negra se llevan a cabo sobre la interfaz del software. Se trata de demostrar que las funciones del software son operativas, que las entradas se manejan de forma adecuada y que se produce el resultado esperado. Las pruebas de caja negra buscan encontrar errores en cinco categorías (Pressman, 2005):

- ✓ Funciones incorrectas o ausentes;
- ✓ Errores de interfaz;
- ✓ Errores en estructuras de datos o en accesos a bases de datos externas;
- ✓ Errores de rendimiento;
- ✓ Errores de inicialización y terminación.

Las pruebas de caja blanca se centran en la estructura lógica interna del software. Se basan en un examen detallado de los procedimientos y caminos lógicos del sistema. Las principales técnicas de diseño de pruebas de caja blanca son:

- ✓ Pruebas de flujo de control
- ✓ Pruebas de flujo de datos

- ✓ Pruebas de bifurcación (branch testing)
- ✓ Pruebas de caminos básicos

**Findstack** es un sitio de revisión dedicado a encontrar las mejores herramientas y servicios de software para iniciar y hacer crecer un negocio exitoso. Se enfoca específicamente en brindar comparaciones fáciles de leer que ayudarán a elegir la solución adecuada para sus necesidades. Findstack ahora está compuesto por desarrolladores, especialistas en marketing digital, de negocio y tecnología, investigadores, periodistas. Son un grupo bastante heterogéneo algunos de ellos han tenido años de experiencia en la creación de sitios web, mientras que algunos son nuevos en ello. Es esta combinación de experiencia y habilidades lo que hace Findstack único y hacen que el contenido sea tan completo y relevante (Findstack, 2022).

El sitio Findstack está enfocado en la demanda siempre creciente de software B2B. Ayudan a una audiencia de aspirantes a fundadores, emprendedores independientes y a comparar plataformas de software, esperando encontrar las mejores herramientas tecnológicas para iniciar y hacer crecer su negocio. En este sitio se puede encontrar información sobre aplicaciones informáticas para contabilidad, útil para profesionales del sector. Con el tiempo, el objetivo es ser un lugar de referencia para los emprendedores novatos que buscan información empresarial, revisiones de software B2B y mejores prácticas centradas en SaaS, tendencias tecnológicas y crecimiento (Findstack, 2022).

**WordPress** es un sistema de gestión de contenidos enfocado a la creación de cualquier tipo de página web. Originalmente alcanzó una gran popularidad en la creación de blogs, para luego convertirse en una de las principales herramientas para la creación de páginas web comerciales. Fue lanzado por la empresa homónima en el 2003 (WordPress, 2015).

El sistema WordPress está desarrollado en el lenguaje PHP para entornos que ejecuten MySQL y Apache, bajo licencia GPL y es software libre. Sus fundadores son Matt Mullenweg y Mike Little. WordPress fue creado a partir del desaparecido *b2/cafeblog* y se ha convertido en el CMS más popular de la blogosfera y en el más popular con respecto a cualquier otro CMS de uso general (W3Techs, 2022).

Aunque la mayor parte de la documentación existente se basa en la utilización del motor de bases de datos MySQL sobre el servidor web Apache, este sistema de gestión de contenidos puede también instalarse en entornos que ejecuten PostgreSQL, MariaDB y Nginx. Otro punto a considerar sobre su

éxito y extensión es la enorme comunidad de desarrolladores y diseñadores, encargados de programarlo en su núcleo o creando complementos (llamados *plugins*) y plantillas (llamadas *temas*) para la comunidad. En marzo de 2019 era usado por el 33,4 % de todos los sitios en Internet y un 60,3 % de todos los sitios basados en gestores de contenido (CMS) (W3Techs, 2022).

## **1.5- Arquitectura de software**

La Arquitectura de Software es la forma en la que se organizan y estructuran los componentes de un sistema, interactuando y relacionándose entre sí y con el contexto, aplicando normas y principios de diseño y calidad, que fortalezcan y fomenten la usabilidad a la vez que dejan preparado el sistema, para su propia evolución. De este modo, los ingenieros, analistas, diseñadores y programadores, pueden trabajar bajo una línea común que les posibilite la compatibilidad necesaria para lograr el objetivo propuesto (Pablo Huet, 2022).

### **1.5.1- Estilos arquitectónicos**

Cada estilo arquitectónico describe una categoría del sistema que contiene: un conjunto de componentes, que realiza una función requerida por el sistema, un conjunto de conectores que posibilitan la comunicación, la coordinación y la cooperación entre los componentes; restricciones que definen como se pueden integrar los componentes que forman el sistema y modelos semánticos que permiten al diseñador entender las propiedades globales de un sistema para analizar las propiedades conocidas de sus partes constituyentes (Gabriel M. Barrera, 2018). Los estilos se clasifican en:

- Estilo de flujo de datos
  - Tuberías y filtros
- Estilos centrados en datos
  - Arquitecturas de pizarra o repositorio
- Estilos de llamada y retorno
  - Modelo-Vista-Controlador (MVC)
  - Arquitecturas en capas
  - Arquitecturas orientadas a objetos
  - Arquitecturas basadas en componentes

- Estilos de código móvil
  - Arquitectura de máquinas virtuales
- Estilos heterogéneos
  - Sistemas de control de procesos
  - Arquitecturas basadas en atributos
- Estilos Peer-to-Peer
  - Arquitecturas basadas en eventos
  - Arquitecturas orientadas a servicios (SOA)
  - Arquitecturas basadas en recursos

Por parte del equipo de arquitectura se tuvo en cuenta el primer patrón del estilo Servidor - Cliente: HEADLESS. Es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario, tratando al modelo, las vistas y los controladores como entidades separadas; lo que hace posible que cualquier cambio producido en el modelo se refleje automáticamente en cada una de las vistas. Entre sus ventajas se encuentran:

- Soporte de vistas múltiples: dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación web pueden utilizar el mismo modelo de objetos, mostrado de maneras diferentes.
- Adaptación al cambio: los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares o PDAS. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Entre las desventajas, se han señalado:

- Complejidad: el patrón introduce nuevos niveles de indirección y por lo tanto aumenta ligeramente la complejidad de la solución. También se profundiza la orientación a eventos del código de la interfaz de usuario, que puede llegar a ser difícil de depurar. En rigor, la configura-

ción basada en eventos de dicha interfaz corresponde a un estilo particular (arquitectura basada en eventos) que aquí se examina por separado.

- Costo de actualizaciones frecuentes: desacoplar el modelo de la vista no significa que los desarrolladores del modelo puedan ignorar la naturaleza de las vistas.

Se seleccionó la arquitectura HEADLESS teniendo en cuenta las facilidades de adaptación al cambio que ofrece, además de estar diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. También al separar la vista de la lógica de negocio y el controlador, permite la reutilización de los mismos y si en algún momento uno de sus componentes funciona mal, puede reemplazarse sin que los demás se vean afectados.

### 1.5.2- Framework de desarrollo

Según el diccionario de Oxford, la palabra framework significa: Una estructura básica que denominamos marco de trabajo, concepto o texto. Desde el punto de vista del desarrollo de software, un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Un framework facilita el desarrollo de software y evita los detalles de bajo nivel, permitiendo concentrar más esfuerzo y tiempo en identificar los requerimientos de software (A. Santa Fe, 2014).

### Symfony5

Este framework separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Es la versión más reciente de Symfony y ha sido ideado para explotar todas las nuevas características de PHP 5.3 y por eso es uno de los framework PHP con mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en un proyecto. Symfony2 también es el framework que más ideas incorpora del resto de los framework, incluso de aquellos que no están programados con PHP (J. Eguiluz, 2011).

Entre sus características más importantes se tiene:

- Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.

- La presentación usa *templates* y *layouts* que pueden ser construidos por diseñadores de HTML que no posean conocimientos del framework.
- Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las base de datos.
- El manejo de cache reduce el uso de banda ancha y la carga del servidor.
- El soporte de autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.
- El enrutamiento y las URLs inteligentes hacen amigable las direcciones de las páginas de la aplicación.
- Permite la paginación, clasificación y filtraje automáticos.
- Los plugins proveen un alto nivel de extensibilidad.
- Dispone de interacción con AJAX.

### **Bootstrap 4**

Bootstrap es conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales (Suraj, 2019).

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. Desde la versión 5.0 también soporta diseños sensibles. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles). Bootstrap es de código abierto y está disponible en GitHub.

#### Ventajas

- Utiliza componentes y servicios creados por la comunidad web, tales como: HTML5 shim, Normalize.css, OOCSS, jQuery UI, LESS (lenguaje de hojas de estilo) y GitHub.
- Es un conjunto de buenas prácticas que perduran en el tiempo.

- Dentro de sus características más destacadas, podríamos mencionar:
  - La implementación de HTML5 + CSS3.
  - Son archivos css, js, imágenes, textos y otros, necesarios para que la interfaz del usuario funcione correctamente.
  - El famoso Grid system, que por defecto incluye 12 columnas fijas o fluidas, dependiendo de si el diseño será responsivo o no.
  - El uso de LESS, que es una ampliación a las famosas hojas de estilo CSS, pero a diferencia de estas, funciona como un lenguaje de programación, permitiendo el uso de variables, funciones, operaciones aritméticas, entre otras, para acelerar y enriquecer los estilos en un sitio web.
  - Herramienta sencilla y ágil para construir sitios web e interfaces que ahorra realmente mucho tiempo de trabajo.
  - Además, tiene un tema por defecto bastante optimizado y que se puede modificar o crear uno propio.

#### Desventajas:

- Es necesario adaptarse a su forma de trabajo, si bien su curva de aprendizaje suave, se debe comprender y familiarizar con su estructura y nomenclatura.
- Se debe adaptar el diseño a una tabla de 12 columnas, que se modifican según el dispositivo.
- Bootstrap por defecto trae anchos márgenes y altos de línea y realizar cambios específicos es un poco tedioso.

### 1.6- Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión o como modo de comunicación humana; es un conjunto de símbolos y reglas sintácticas y semánticas

que definen su estructura y el significado de sus elementos y expresiones y que permite la comunicación entre el usuario y la computadora.

### 1.6.1- PHP 8.0

PHP es un popular lenguaje de programación de propósito general del lado del servidor que es especialmente adecuado para el desarrollo web por ser rápido, flexible y pragmático (Group, 2012). Originalmente fue diseñado para el desarrollo web de contenido dinámico. Está actualmente entre los proyectos de código abierto más populares (gracias en parte a la similitud de su sintaxis con el lenguaje). El código es interpretado por un servidor web con un módulo procesador de PHP que genera la página Web resultante. Su funcionamiento es muy básico, cuando el cliente realiza una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP, este procesa el script solicitado que generará el contenido de manera dinámica y el resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente (código HTML). También mediante extensiones es posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos (Arias, 2017).

Ventajas de este lenguaje:

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, se destaca su conectividad con MySQL y PostgreSQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Tiene manejo de excepciones (desde PHP5)

### 1.6.2- jQuery 2.1.4

jQuery es una biblioteca de JavaScript, que contiene procesos o rutinas ya listos para ser usados, lo que permite simplificar la forma de interactuar con los documentos HTML. Facilita la búsqueda de elementos en un documento y su posterior manipulación ya sea por adición de contenido, edición de atributos HTML y propiedades CCS, definición de controladores de eventos o desarrollo de animaciones. Es software libre y de código abierto debido a que se encuentra registrado bajo la licencia MIT y la licencia pública general de GNU v2, lo que permite su uso en proyectos libres y privativos.

Principales características:

- Selección de elementos DOM (Document Object Model).
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 3 y un plugin básico de Xpath (XML Path Language).
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones como *trim* () (elimina los espacios en blanco del principio y final de una cadena de caracteres), etc.

jQuery es un producto muy beneficioso que cuenta con una buena aceptación por parte de los programadores y un grado de penetración en el mercado muy amplio, convirtiéndolo en una de las mejores opciones. Por otra parte, el archivo que contiene la librería solo ocupa unos 56 KB, lo que es bastante razonable y no retrasará mucho la carga de la página web (M. A. Álvarez, 2009).

### 1.7- Herramientas de desarrollo

A continuación, se exponen las principales características, ventajas y desventajas de las herramientas seleccionadas para el desarrollo de la propuesta de solución.

### 1.7.1- Herramienta CASE

Las herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

#### Visual Paradigm for UML 8.0 (Lenguaje Unificado de Modelado)

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (VP Internacional, 2013).

Se caracteriza por:

- Disponibilidad en múltiples plataformas (Windows, GNU/Linux).
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar para todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería inversa.
- Licencia gratuita y comercial.
- Soportar aplicaciones Web.
- Generar código para Java y exportación como HTML.
- Una Fácil instalación y actualización.
- Tener un soporte de UML versión 2.1.
- Crear diagramas de flujo de datos.

- Soporte ORM - Generación de objetos Java desde bases de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (SGBD) existentes a diagramas de Entidad-Relación.
- Generador de informes.

Se decidió utilizar la herramienta Visual Paradigm for UML en su versión 8.0 debido a las características antes mencionadas, así como por las ventajas que brinda la herramienta y por dos razones fundamentales: la primera razón es que la aplicación a desarrollar será escrita en PHP y esta herramienta se integra fácilmente con este lenguaje; la segunda razón es que el equipo de desarrollo cuenta con una mayor experiencia en el empleo de esta herramienta respecto a las otras, posibilitando un considerable ahorro de tiempo en el proceso de desarrollo del software.

### **1.7.2- Entorno de desarrollo integrado (IDE)**

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto (Fergarcia, 2013).

### **VisualStudioCode 2022**

Es un editor de código fuente desarrollado por Microsoft para Windows, GNU/Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software propietario e incluye características personalizadas por Microsoft.

Visual Studio Code es un editor de código fuente construido sobre el framework Electron. Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un lenguaje dado, como se muestra en la siguiente tabla. Muchas de las características de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos. json (por ejemplo, preferencias del usuario). La paleta de comandos es una interfaz de línea de comandos. Sin embargo, desaparece si el usuario hace clic fuera de él o presiona una combinación de teclas en el teclado para interactuar con algo que está fuera de él. Esto también se aplica a los comandos que requieren mucho tiempo. Cuando esto sucede, el comando en progreso se cancela.

### 1.7.3- Sistema Gestor de Bases de Datos (SGBD)

Un sistema gestor de bases de datos es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos (Base Datos Ofimáticas, 2015).

Un sistema gestor de base de datos está compuesto de:

- El gestor de la base de datos: se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo.
- Diccionario de datos: es una base de datos donde se guardan todas las propiedades de la base de datos, descripción de la estructura, relaciones entre los datos, etc.
- El administrador de la base de datos: es una persona o grupo de personas responsables del control del sistema gestor de base de datos.
- Los lenguajes: un sistema gestor de base de datos debe proporcionar una serie de lenguajes para la definición y manipulación de la base de datos (M. A. Álvarez, 2009).

### PostgreSQL V.13

Un sistema gestor de bases de datos es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. En estos sistemas se propor-

ciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos (Base Datos Ofimáticas, 2015).

Un sistema gestor de base de datos está compuesto de:

- El gestor de la base de datos: se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo.
- Diccionario de datos: es una base de datos donde se guardan todas las propiedades de la base de datos, descripción de la estructura, relaciones entre los datos, etc.
- El administrador de la base de datos: es una persona o grupo de personas responsables del control del sistema gestor de base de datos.
- Los lenguajes: un sistema gestor de base de datos debe proporcionar una serie de lenguajes para la definición y manipulación de la base de datos (M. A. Álvarez, 2009).

PostgreSQL es el sistema de gestión de bases de datos objeto-relacional de código abierto más potente del mercado, distribuido bajo licencia BSD y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en lugar de multihilos para garantizarla estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

### **Conclusiones del capítulo**

1. El estudio de los conceptos y características asociadas a los portales web para la divulgación de la información permitió comprender el objeto de estudio de la investigación.
2. El estudio de los sistemas homólogos ratificó la necesidad de crear una solución que resuelva las exigencias del cliente.
3. El carácter de la investigación define AUP-UCI en su escenario 4 como metodología que guía el proceso de desarrollo que acompaña a la investigación.
4. Se definen las herramientas y lenguajes de desarrollo a emplear durante el diseño e implementación del portal web de La Lisa.

## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

La propuesta de solución que se caracteriza en este capítulo consta de un sistema de gestión que se apoya en una herramienta web. Este sistema trata la divulgación de información de la UJC de La Lisa. Utilizando la información recopilada en el capítulo anterior se planteó el desarrollo de un portal web utilizando el CMS WordPress con la intención de mejorar y perfeccionar el posicionamiento web del portal de tal forma que los usuarios puedan acceder a este a través de los motores de búsqueda.

En este capítulo se abordan las características que el sistema a desarrollar debe tener, definiendo las clases del dominio y la relación entre ellas. Se identifican los requisitos funcionales y no funcionales que debe cumplir el portal web. Se explica además la arquitectura y el diseño del sistema.

### **2.1- Propuesta de solución.**

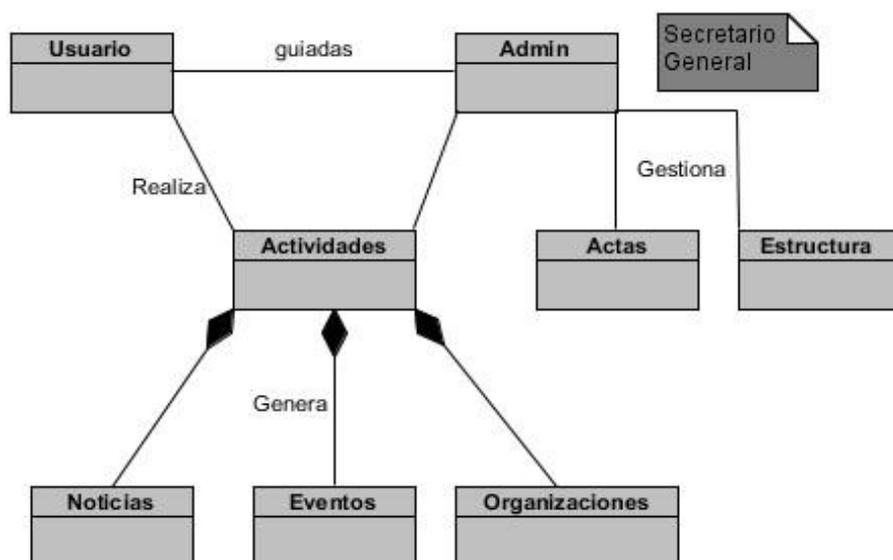
Para dar solución al problema existente se decide desarrollar un portal web utilizando el CMS WordPress, que podrá ser accedido por el usuario mediante una interfaz sencilla y amigable. Este portal contará con un espacio para la divulgación de las diferentes actividades propuestas por la organización tanto dentro como fuera del municipio, así como eventos e intercambios. El mismo mostrará las noticias más importantes referentes a la actividad política-cultural del país, no sólo del acontecer de la juventud sino también de noticias nacionales e internacionales. El portal permitirá a los usuarios conocer a los militantes más destacados de la organización que lo componen. La solución dispondrá información sobre los diferentes reportes y actas de las diversas reuniones que se realizan en la organización para la comunidad joven, así como mostrará entrevistas dirigidas a miembros de la Juventud Comunista. El usuario tendrá acceso a contenidos audiovisuales como imágenes y vídeos de los diferentes festivales, eventos y actividades organizadas por la UJC municipal como las del resto de la provincia. Dispondrá de un buscador de contenidos que permite buscar mediante un filtrado de letras, palabras o frases.

El sistema propuesto tiene como objetivo brindar una plataforma de publicación de contenido, para obtener como resultado un portal donde situar de forma organizada, información sobre dicha organización.

### **Modelo de Dominio**

El Modelo de Dominio representa los conceptos más importantes y significativos de las clases conceptuales en el desarrollo de un sistema. Su objetivo fundamental es definir las interrelaciones de los

objetos más importantes representados mediante clases. Es utilizado para la comprensión del entorno y en la planificación futura de la posible aplicación (Francisco G. & Andrea, 2018).



**Entre las opciones que brinda el sistema se encuentran:**

- Adicionar (dar alta), modificar y eliminar (dar baja) un militante de un CB introduciendo los datos del mismo en el caso que lo necesite.
- Adicionar, modificar o eliminar un CB del CP.
- La creación del acta especificando la fecha de realización, lugar de reunión, cantidad de presentes en el momento, por ciento de asistencia, lista de militantes ausentes seguido de sus causas de ausencia, miembro del organismo superior que visita la reunión y cargo que ocupa el mismo, las opiniones por puntos y los acuerdos tomados
- Generar y actualizar la plantilla del control de la militancia, que archiva por cada CB el listado de los militantes que posee, así como datos fundamentales de la misma: zona, esfera y sector de la economía, también se archivan datos como la fecha de ingreso a las filas de cada uno de los militantes, la fecha de arribo a la plantilla y la causa y la fecha de baja, así como su causa.
- Permite a los militantes ver su expediente de militante que contiene además de los datos personales, las sanciones que ha recibido, los méritos que le han otorgado, así como distinciones u otros reconocimientos durante toda su trayectoria en la organización.

- Aplicar una sanción a un militante especificando, la fecha de aplicada la sanción, la causa de la sanción y el tipo de sanción.
- Otorgar un reconocimiento a un militante especificando la fecha de otorgamiento, el motivo y el reconocimiento.
- Generar reportes.

## 2.2- Especificación de requisitos

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. El conjunto de todas las necesidades es el fundamento para el consiguiente desarrollo del sistema o componente (Pressman, 2013). Se definieron 45 requisitos funcionales que se enumeran a continuación.

### Requisitos Funcionales

Los requisitos funcionales para un sistema describen lo que debe hacer el sistema. Estos requisitos dependen del tipo de software que se está desarrollando, los usuarios esperados del software y el enfoque general adoptado por la organización al redactar los requisitos. Cuando se expresan como requisitos del usuario, los requisitos funcionales deben estar escritos en lenguaje natural para que los usuarios y administradores del sistema puedan entenderlos. Los requisitos funcionales del sistema amplían los requisitos del usuario y están escritos para desarrolladores de sistemas. Deben describir las funciones del sistema, sus entradas y salidas, y las excepciones en detalles (SOMMERVILLE, 2011).

**Tabla 2.** *Requisitos Funcionales.*

<b>Requisitos Funcionales (RF)</b>			
<b>No.</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Complejidad</b>
1	Autenticar usuario	El usuario introduce sus datos, el sistema lo comprueba y si son válidos, el usuario queda autenticado con el nivel de privilegios asignados, si los datos no son válidos el sistema muestra un mensaje de	Alta

		error.	
2	Crear usuario	El sistema permite que los usuarios con rol administrador creen los usuarios.	Media
3	Editar usuario	El sistema permite que los usuarios con rol administrador modifiquen los privilegios asignados a los usuarios.	Media
4	Mostrar usuario	El sistema permite que los usuarios con rol administrador visualicen los datos de los usuarios existentes.	Media
5	Eliminar usuario	El sistema permite que los usuarios con rol administrador eliminen cuentas de usuarios existentes.	Media
6	Crear estructura organizativa	El sistema permite a los usuarios con rol de administrador crear la estructura organizativa.	Alta
7	Modificar estructura	El sistema permite a los usuarios con rol de administrador modificar la estructura organizativa.	Media
8	Buscar la estructura	El sistema permite a los usuarios con rol de administrador buscar la estructura organizativa.	Media
9	Eliminar estructura	El sistema permite a los usuarios con rol de administrador eliminar la estructura organizativa.	Media
10	Crear militante (dar de alta)	El sistema permite a todos los secretarios de CB dar de alta a un militante.	Alta
11	Modificar militante	El sistema permite a todos los secretarios de CB modificar a un militante.	Media
12	Buscar militante	El sistema permite a todos los secretarios de CB buscar a un militante.	Media
13	Eliminar militante (dar de baja)	El sistema permite a todos los secretarios de CB dar de baja un militante.	Media
14	Insertar sanción	El sistema permite al secretario general insertar una sanción a un militante.	Alta
15	Modificar sanción	El sistema permite al secretario general modificar	Media

		una sanción a un militante.	
16	Mostrar sanción	El sistema permite al secretario general mostrar una sanción a un militante.	Media
17	Eliminar sanción	El sistema permite al secretario general eliminar una sanción a un militante.	Media
18	Agregar méritos	El sistema permite al secretario general agregar un mérito a un militante.	Media
19	Crear acta	El sistema permite a los usuarios con rol de administrador crear un acta.	Alta
20	Modificar acta	El sistema permite a los usuarios con rol de administrador modificar un acta.	Media
21	Buscar Acta	El sistema permite a los usuarios con rol de administrador buscar un acta.	Media
22	Listado de acuerdos	El sistema permite a los usuarios con rol de administrador obtener el listado de acuerdos acta.	Media
23	Militantes ausentes	El sistema permite a los usuarios con rol de administrador obtener el listado de los militantes ausentes.	Media
24	Listado de militantes	El sistema permite a los usuarios con rol de administrador tener un listado de todos los militantes.	Media
25	Crear plantilla	El sistema permite a los usuarios con rol de administrador crear una plantilla.	Alta
26	Modificar plantilla	El sistema permite a los usuarios con rol de administrador modificar una plantilla.	Media
27	Buscar plantilla	El sistema permite a los usuarios con rol de administrador buscar una plantilla.	Media
28	Eliminar plantilla	El sistema permite a los usuarios con rol de administrador eliminar una plantilla.	Media
29	Crear noticia	El sistema permite que los usuarios con rol de editor creen las noticias.	Alta
30	Editar noticia	El sistema permite que los usuarios con rol de edi-	Media

		tor editen las noticias.	
31	Mostrar noticia	El sistema permite que todos los usuarios observen las noticias.	Media
32	Eliminar noticia	El sistema permite que todos los usuarios eliminen las noticias.	Media
33	Crear galería	El sistema permite a los usuarios con rol de editor crear nuevos archivos a la galería.	Media
34	Editar galería	El sistema permite a los usuarios con rol de editor editar archivos de la galería.	Media
35	Mostrar galería	El sistema permite a todos los usuarios observar la galería.	Media
36	Eliminar galería	El sistema permite a los usuarios con rol de editor eliminar archivos de la galería.	Media
37	Generar reporte	El sistema permite a los usuarios con rol de administrador generar un reporte.	Alta
38	Mostrar contenidos más populares	El sistema permite a los usuarios con rol de editor mostrar el contenido más popular.	Media
39	Crear evento	El sistema permite a los usuarios con rol de editor crear nuevos eventos.	Alta
40	Editar evento	El sistema permite a los usuarios con rol de editor editar eventos.	Media
41	Mostrar evento	El sistema permite a todos los usuarios observar los eventos.	Media
42	Eliminar evento	El sistema permite a los usuarios con rol de editor eliminar evento.	Media
43	Búsqueda simple	El sistema permite al usuario realizar una búsqueda de los elementos del portal	Media
44	Búsqueda avanzada	El sistema permite al usuario con rol de administrador realizar una búsqueda avanzada sobre los militantes, actas, eventos y los comités	Media

45	Buscar	El sistema permite hacer búsquedas con cualquier con cualquier criterio.	Media
----	--------	--	-------

**Fuente:** elaboración propia.

### Requisitos No Funcionales

Los requisitos no funcionales, como su nombre indica, son requisitos que no están directamente relacionados con los servicios específicos que el sistema brinda a sus usuarios. Estos requisitos no funcionales generalmente especifican o restringen las características del sistema en su conjunto. Pueden relacionarse con propiedades emergentes del sistema, como la confiabilidad, el tiempo de respuesta y el uso de la memoria. Alternativamente, pueden definir restricciones en la implementación del sistema, como las capacidades de los dispositivos de E / S o las representaciones de datos utilizadas en las interfaces con otros sistemas (SOMMERVILLE, 2011).

Distribuidos, según el estándar de calidad internacional ISO/IEC 9126-1, en especificaciones de usabilidad, confiabilidad, portabilidad, eficiencia, funcionalidad y diseño, se obtuvo un total de catorce (13) requisitos no funcionales y quedan relacionados a continuación:

- RnF 1. Usabilidad
  - Diseño limpio y claro que facilite la navegación.
  - Se debe lograr una estructura simple y lo más intuitiva posible de la arquitectura de la información.
- RnF 2. Seguridad
  - Se asignarán los permisos de acceso, escritura, lectura en dependencia del rol que desempeñe cada usuario del sistema.
  - El sistema debe permitir usuarios con diferentes niveles de acceso.
  - La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
  - Llevar un registro de sucesos donde se archiven los eventos del sistema incluyendo los eventos de error, inicio de sesión, cierre de sesión y modificación de la información.

- La propuesta de desarrollo debe ser capaz de ejecutarse en los navegadores empleados en la UCI (por ejemplo: Opera, Firefox, Chrome), así como adaptar su interfaz a cualquier dispositivo
- RnF 3. Eficiencia
  - El sistema debe permitir que los usuarios interactúen con él de manera simultánea.
  - El tiempo de demora del sistema en cada transición debe ser menor de cinco (5) segundos aproximadamente.
- RnF 4. Funcionalidad
  - La aplicación debe gestionar y requerir información de usuarios para su uso.
  - La información manejada por el sistema estará protegida de accesos no autorizados.
  - Ante los errores que puedan ocasionarse en el sistema no se deben mostrar detalles de información que puedan comprometer su seguridad e integridad.
  - El sistema deberá ser utilizado por los usuarios con dominio uci.cu
  - El sistema deberá ser capaz, por cuestiones de seguridad, expirar la sesión del usuario una vez pasado diez (10) minutos de inactividad.
- RnF 5. Diseño
  - Las páginas generadas deben cumplir con los estándares para HTML 5 y CSS 3.
  - El software, bibliotecas o componentes empleados deben estar regidos en la medida de lo posible por licencias y patentes de software libre.
  - Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.
- RnF 5. Accesibilidad
  - Interfaces fáciles de usar por personas que tienen poca experiencia en el manejo de computadoras.
  - Se tiene un acceso limitado para los documentos respectivos a las actas de las reuniones y actividades realizadas.

### 2.2.1- Descripción de los requisitos de software (Historia de Usuarios)

Para planificar adecuadamente el desarrollo de la aplicación, es necesario estimar de forma ideal el tiempo que les tomará a los programadores la codificación de cada una de las historias de usuario. Las condiciones ideales se dan cuando se escribe el código sin distracciones y con una dedicación de tiempo completo. La estimación debe dar un máximo de dos semanas, si es necesario más tiempo debe considerarse una división de la historia de usuario en partes más pequeñas. A partir de la prioridad de las historias se decide cuáles de ellas se implementarán en las primeras iteraciones, las funcionalidades críticas del sistema deben ser codificadas en iteraciones tempranas del ciclo.

**Tabla 3.** Crear Acta.

<b>No. 14</b>	<b>Nombre:</b> Crear Acta	
<b>Programador:</b> Alejandro Nuñez Rivas		<b>Interacción Asignada:</b> 1era
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 2 h
<b>Riesgo de Desarrollo:</b> Medio		<b>Tiempo Real:</b> 2 h
<p><b>Descripción:</b> Los usuarios autenticados con privilegios de administración podrán crear artículo en el sistema, para ello deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> <li>• <b>Tipo de Acta:</b> Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 255 caracteres.</li> <li>• <b>Fecha:</b> Obligatorio. Calendario desplegable. Formato dd/mm/aaaa. Permite seleccionar de un calendario desplegable. Valor por defecto (fecha actual).</li> <li>• <b>Hora-Lugar:</b> Obligatorio. Lista desplegable. Valores (Lista de categorías). Valor por defecto (- Seleccione un valor-).</li> <li>• <b>Comité de Base:</b> Obligatorio. campo de número entero.</li> <li>• <b>Orden del Día:</b> Obligatorio. Área de texto. Permite todos los caracteres.</li> <li>• <b>Desarrollo:</b> Obligatorio. Área de texto. Permite todos los caracteres. Cajas de textos (Tabla de Acuerdos, número de acuerdos, fecha de los acuerdos)</li> <li>• <b>Activista de Acta:</b> Opcional. Campo de auto completamiento. Permite todos los caracteres. El autor tiene que haber estado creado en el sistema.</li> <li>• <b>Ocupación del secretario:</b> Obligatorio. Campo de auto completamiento. 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 (,).</li> </ul>		
<b>Observaciones:</b>		

--

<b>No. 24</b>	<b>Nombre:</b> Crear Noticia	
<b>Programador:</b> Alejandro Nuñez Rivas		<b>Interacción Asignada:</b> 1era
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 2 h
<b>Riesgo de Desarrollo:</b> Alto		<b>Tiempo Real:</b> 1.3 h
<p><b>Descripción:</b> El sistema permite a los usuarios con rol web máster y administrador crear contenidos de tipo noticia en el sistema, llenando los siguientes campos:</p> <ul style="list-style-type: none"> <li>● Título: Campo de texto, obligatorio. Admite todos los caracteres. Longitud máxima 60 caracteres.</li> <li>● Imagen: Campo de archivo de tipo imagen, obligatorio y cantidad limitada a un archivo. Admite archivos locales que deben ser de formato png o jpg.</li> <li>● Descripción: Área de texto, obligatoria. Admite todos los caracteres.</li> <li>● Fecha de publicación: Calendario desplegable, obligatorio. Formato DD/MM/AAAA.</li> <li>● Manifestación: Lista de selección, opcional y cantidad limitada a uno. Contiene un listado con los términos existentes de la taxonomía Manifestación (Artes Plásticas, Música, Danza, Teatro, Audiovisuales, Literatura).</li> <li>● Texto alternativo: Campo de texto, obligatorio. Admite todos los caracteres. Longitud máxima 024 caracteres. Este campo se muestra cuando al seleccionar una imagen desde la pestaña Subir al servidor, se presiona el botón Siguiente.</li> </ul>		
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. Si el usuario introduce los datos de forma correcta, el sistema emitirá un mensaje confirmando la culminación exitosa de la acción.</li> <li>2. Si el usuario introduce los datos de forma incorrecta, el sistema emitirá un mensaje notificando el error.</li> </ol>		

3. Si el usuario deja campos obligatorios vacíos, el sistema emitirá un mensaje notificando el error.
Prototipo elemental de interfaz gráfica de usuario:
<b>NO APLICA</b>

**Tabla 4.** *Crear Noticia.*

**Fuente:** elaboración propia.

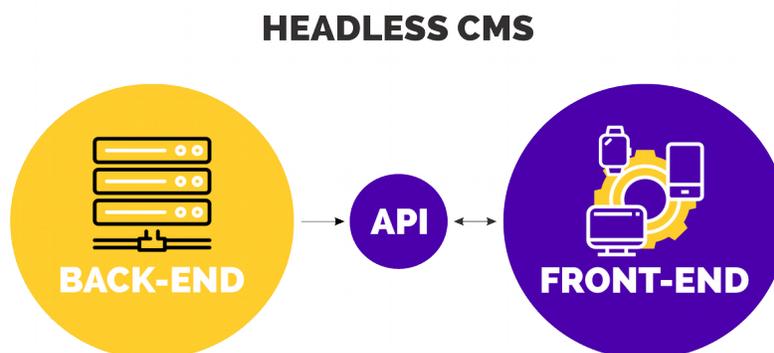
## 2.3- Análisis y Diseño

### Patrón Arquitectónico

Durante el ciclo de vida de desarrollo de un software utilizando la metodología AUP-UCI la fase de diseño se realiza justo después que se termina la fase de planificación. Durante esta fase se comienza a definir el diseño arquitectónico del software partiendo de las historias de usuario de la primera iteración y de la metáfora, un elemento que según Kent Beck el creador de la metodología, guía al equipo de proyecto durante todo el desarrollo del programa y que además sustituye en gran parte de lo que sería la arquitectura del sistema en modelos más pesados o tradicionales.

La arquitectura de la herramienta o de cualquier proyecto debe ser sencilla y responder al principio de “cuál es la solución más simple capaz de funcionar”. Debe ser una estructura que organiza la lógica del sistema de manera que al realizar cambios en partes no provoca cambios en otros componentes, a menos que sea absolutamente necesario. Debe enfocarse en cómo resolver los problemas identificados en el momento en que se diseña, sin tener en cuenta posibles variaciones o requerimientos futuros, los cuales serán enfrentados y agregados al diseño en el momento en que sean detectados (RJ. Advance, 2019).

Para definir correctamente la arquitectura de un software es necesario que se identifiquen los elementos arquitectónicamente significativos (clases, módulos, objeto, interfaces) y las relaciones que existen entre ellos. Para garantizar que los cambios o nuevas funcionalidades añadidas puedan ser asimilados por la arquitectura del sistema, esta estará formada por capas (presentación, lógica del negocio y acceso a datos) organizadas por responsabilidades (WordPress, 2020).



*Figura 1. Arquitectura HEADLESS.*

**Fuente:** WordPress, 2022.

### Patrones de Diseño

En la ingeniería del software, un patrón constituye el apoyo para la solución a los problemas más comunes que se presentan durante las diferentes etapas del ciclo de vida del software. Ahora bien, los patrones de diseño según The Gang of Four -GOF- “describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos”. En términos generales es posible decir que un patrón de diseño es una solución a un problema recurrente en el diseño de software (Guerrero, Suárez y Gutiérrez, 2013).

Ejemplo de estos patrones son: Singleton Pattern (Patrón de Instancia Única), Decorator Pattern (Patrón Decorador), Observer Pattern (Patrón de Observador), Bridge Pattern (Patrón de Puente), Chain of Responsibility Pattern (Patrón de Cadena de Responsabilidad) y Command Pattern (Patrón de Comando). Esto se basa en el hecho de que WordPress se desarrolló para crear aplicaciones web integrales. Por lo tanto, es necesario usar patrones de diseño para que las aplicaciones web se puedan extender como el viento

### Patrones de Diseño GOF (Gang of Four)

- **Singleton:** Si se piensa en los módulos y temas como objetos, entonces siguen el patrón singleton. En general, estos objetos no encapsulan datos; lo que separa un módulo de otro es el conjunto de funciones que contiene, por lo que debe considerarse como una clase con una instancia singleton.
- **Decorator:** WordPress hace un amplio uso del patrón decorador. Esto es solo una pequeña parte del poder del sistema de nodos. Más interesante es el uso de varios ganchos (hook) de nodo

(`hook_node_load()`, `hook_node_view()`, etc.), que permiten a los módulos arbitrarios extender el comportamiento de todos los nodos. Esta característica permite agregar una amplia variedad de comportamientos a los nodos sin la necesidad de subclasificar. Por ejemplo, un nodo de historia básica tiene solo unos pocos datos asociados: título, autor, cuerpo, avance y un puñado de metadatos. Una necesidad común es que los archivos se carguen y se adjunten a un nodo, por lo que se podría diseñar un nuevo tipo de nodo que tuviera las características del nodo de la historia más la capacidad de adjuntar archivos. El módulo de carga de WordPress satisface esta necesidad de una manera mucho más modular mediante el uso de la API de nodo para otorgar a cada nodo que le solicite la capacidad de tener archivos adjuntos. Este comportamiento podría ser imitado por el uso de decoradores, envolviéndolos alrededor de cada objeto de nodo. Más simplemente, los lenguajes que admiten categorías, como Objective-C, podrían aumentar la clase base común de todos los objetos de nodo para agregar el nuevo comportamiento. La implementación de WordPress es una simple ramificación del sistema de enlace y la presencia de `node_invoke()`.

- **Observer:** La interacción anterior también es similar al uso de observadores en sistemas orientados a objetos. Este patrón de Observador es generalizado en todo WordPress, ya que muchos de los ganchos de WordPress esencialmente permiten que los módulos se registren como observadores de los objetos. Por ejemplo, cuando se realiza una modificación a un vocabulario en el sistema de taxonomía de WordPress, un enlace de taxonomía como `hook_taxonomy_vocabulary_update()` se llama en todos los módulos que lo implementan. Al implementar el gancho, los módulos se han registrado como observadores del objeto de vocabulario; cualquier cambio en él se puede actuar según corresponda.
- **Bridge:** La capa de abstracción de la base de datos WordPress se implementa de manera similar al patrón de diseño de Bridge. Los módulos deben escribirse de forma independiente del sistema de base de datos que se utiliza, y la capa de abstracción lo proporciona. Se pueden escribir nuevas capas de base de datos que se ajusten a la API definida por el puente, agregando soporte para sistemas de bases de datos adicionales sin la necesidad de modificar el código del módulo.
- **Chain and Responsibility:** El sistema de menús de WordPress 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.

- **Command:** Muchos de los ganchos de WordPress 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.

## 2.4- Modelado de Diseño

### 2.4.1- Diagramas de clases del diseño con estereotipo web (DCD)

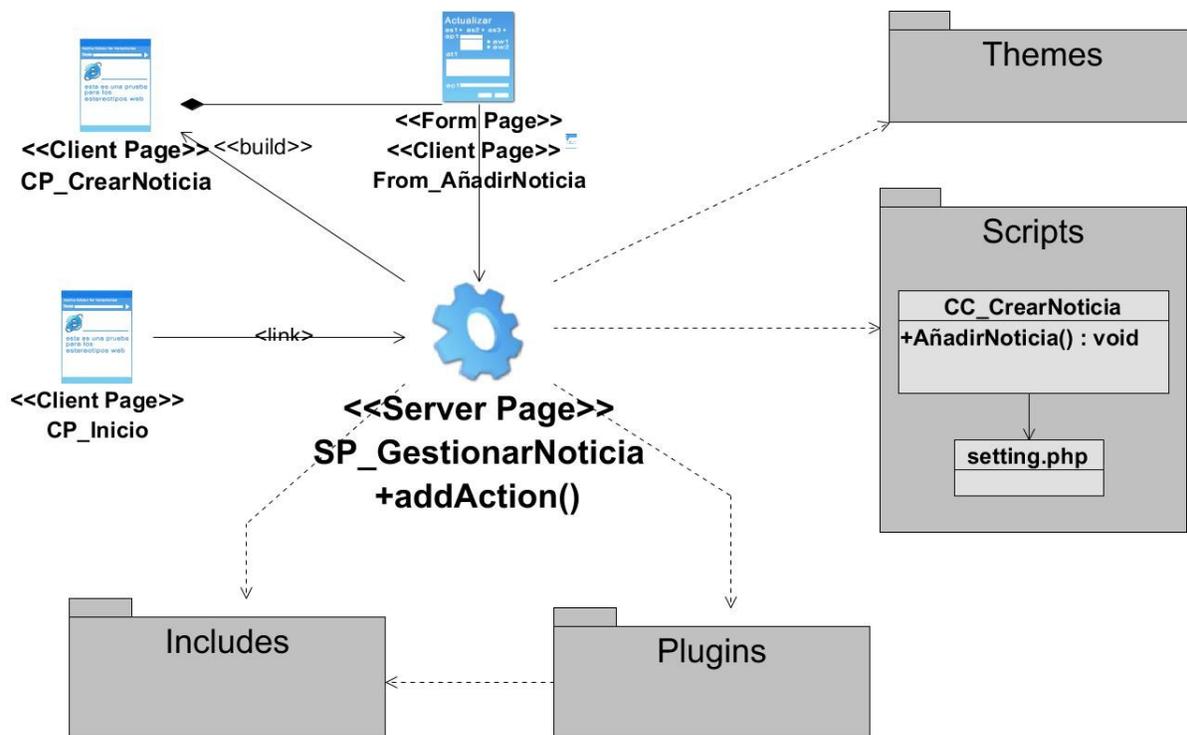
A continuación, se relaciona el diagrama de clases del diseño correspondiente a las historias de usuario: crear noticia, crear actividad y modificar actividad. En estos se muestran las relaciones entre las clases participantes, teniendo como característica particular sobre los diagramas de clases tradicionales, el uso de estereotipos UML especiales que son específicos para los diagramas de clases web, los cuales permiten modelar de forma certera la arquitectura y funcionamiento de aplicaciones de este tipo.

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

**Plugins:** se encuentran todos los plugins que forman parte del funcionamiento del CMS.

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

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

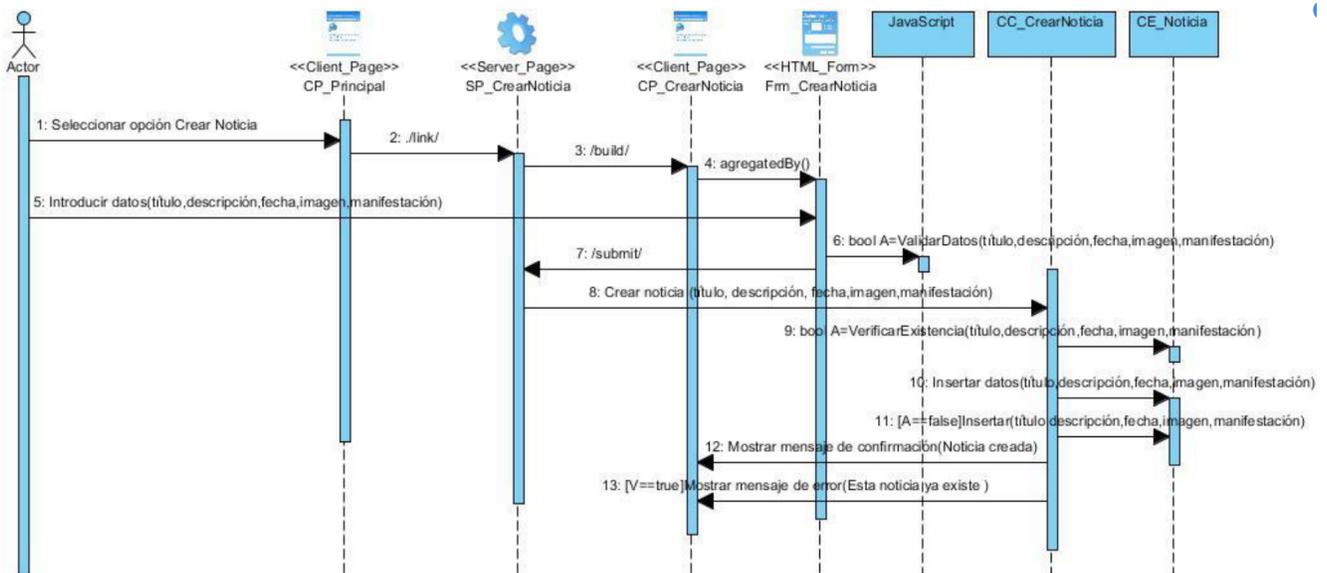


**Figura 2.** Diagrama de Clase del Diseño para HU (Crear Noticia).

**Fuente:** elaboración propia.

#### 2.4.2- Diagramas de Secuencia

Los diagramas de secuencia en el UML se utilizan principalmente para modelar las interacciones entre los actores y los objetos en un sistema y las interacciones entre los objetos mismos. El UML tiene una sintaxis rica para diagramas de secuencia, que permite modelar muchos tipos diferentes de interacción. Como su nombre lo indica, un diagrama de secuencia muestra la secuencia de interacciones que tienen lugar durante un caso de uso particular o una instancia (SOMMERVILLE, 2011). Las figuras 6, 7 y 8 presentadas a continuación son los diagramas de secuencia correspondientes a las historias de usuario mencionadas anteriormente (crear noticia, crear actividad y modificar actividad) respectivamente. Estos diagramas modelan las interacciones involucradas en las historias de usuario, donde un usuario puede crear una noticia, crear una actividad y modificar esta última.



**Figura 3.** Diagrama de Secuencia para HU (Crear Noticia).

**Fuente:** elaboración propia.

### Conclusiones del capítulo

Luego de realizar el análisis de la propuesta de solución y haber generado los artefactos que dispone la metodología AUP UCI en el escenario 4, se puede concluir lo siguiente:

1. Con el análisis de las características del sistema permitió identificar los principales requisitos funcionales y no funcionales del Portal Web de la UJC de la Lisa que permitieron desarrollar las distintas funcionalidades que debe presentar el sistema para solucionar las necesidades del cliente.
2. La identificación de los patrones de diseño y el estilo arquitectónico de la solución propuesta, permite disminuir el impacto de los cambios futuros en el código fuente de la misma.
3. El diseño de los diagramas de clases, facilitó el enfoque en cuanto a composición lógica y física de la propuesta de solución, así como permitió una mejor comprensión sobre cómo debe comportarse el portal web.
4. La generación de todos los artefactos requeridos por el modelo de desarrollo, documentan la solución propuesta, lo cual facilita su posterior mantenimiento (actualización o adición de funcionalidades).



## CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Antes de implementar el software es preciso haber llegado a un acuerdo entre el cliente y el grupo de desarrolladores para así resolver el problema planteado. En el momento de pasar al proceso de implementación es necesario realizar algunas actividades que garanticen la calidad del producto, para esto se realizan unas pruebas de validación que garantizan la conformidad del usuario respecto a la aplicación. En este capítulo se describe la organización del módulo en un diagrama de componentes y se especifican los estándares de codificación a utilizar. Además, se analizan las pruebas realizadas con el objetivo de evaluar la calidad del producto desarrollado y garantizar que este cumpla con las funcionalidades requeridas.

### 3.1- Modelo de despliegue

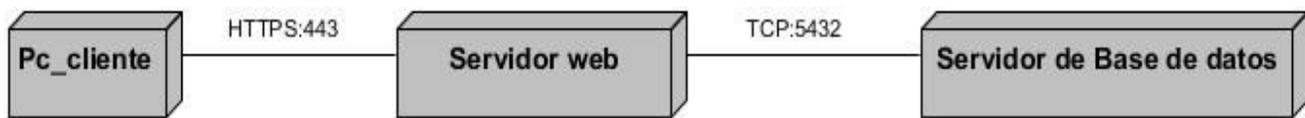
El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (Jimenez-González5, 2018).

A continuación, se describen los elementos que componen el diagrama de despliegue para el portal web UJC de La Lisa que se muestra en la Figura 4.

**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, y que 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/IP*. Además, se comunica con el servidor *LDAP* que le brinda la posibilidad de comprobar las credenciales de los usuarios que se autentican en el sistema.

**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.

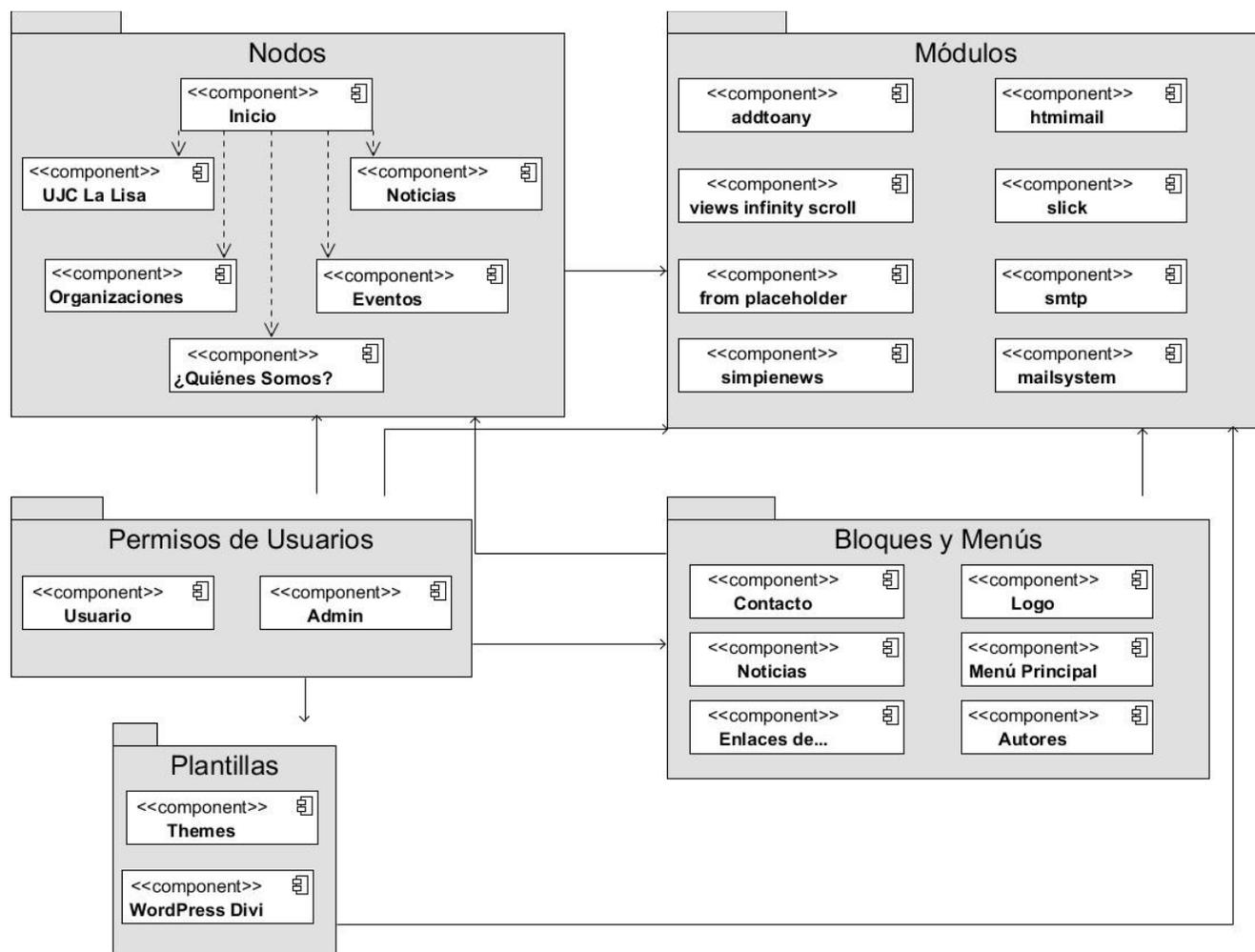


**Figura 4.** Modelo de Despliegue.

**Fuente:** elaboración propia.

### 3.2- Diagrama de componentes

El Diagrama de Componente describe cómo se organizan los componentes de acuerdo con el lenguaje de programación utilizado y al entorno de implementación, y cómo dependen los componentes entre sí. Un componente es el empaquetamiento físico de un elemento del diseño, como lo son las clases en el modelo de diseño. Según el lenguaje de programación utilizado, los componentes que se generan tienen sus características específicas, pues como se ha expresado anteriormente ellos no son más que el empaquetamiento físico de un elemento. En la Figura 10 se muestra el diagrama de Componentes de GESCAP (Jimenez-González5, 2018).



*Figura 5. Diagrama de Componentes.*

**Fuente:** elaboración propia.

A continuación, se describen los paquetes que componen el diagrama de componentes para el portal web UJC de La Lisa que se muestra en la Figura 5:

**Nodos:** elementos de procesamiento con al menos un procesador, memoria, y otros dispositivos.

**Módulos:** nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción.

**Bloques y menús:** muestra los elementos que componen el menú principal y el pie de página.

**Plantillas:** es el área de trabajo que brinda el CMS para personalización del sistema, conformados por las plantillas utilizadas y el editor visual Divi.

### 3.3- Estándares de codificación de WordPress

Las mejores prácticas en WordPress son difíciles de hacer cumplir, simplemente porque los desarrolladores que emplean un mal código pueden funcionar igual de bien (en la superficie), que el código que se ha realizado con buenas prácticas. Se puede aprender un poco sobre el desarrollo de WordPress, crear productos más compatibles con WordPress. Con esto desarrollarás código de calidad, lo cual será beneficioso para la comunidad de WordPress. En este momento WordPress tiene cuatro guías, una para cada tecnología web que se utiliza: PHP, HTML, JavaScript y CSS. Forman parte de un conjunto más amplio de conocimientos, el Manual Core. A continuación, se destaca lo más importante:

#### 3.3.1 PHP

##### Estilos En Bloques

Llaves de partida inicio deben colocarse al final de las líneas. Si existe una nueva condicional, esta debe añadirse justo después del cierre de la anterior condicional, justo de la siguiente manera:

```
<php>

[php]if (condition) {
// Do Something
} elseif (condition) {
// Do Something
} else {
// Do Something
} [[/php]
```

*Figura 6. Código PHP.*

**Fuente:** elaboración propia.

##### Uso De Espacio

Reducir el tamaño de los archivos de código es algo que beneficia cuestiones de velocidad, aunque esto es útil en ocasiones, hay otros escenarios en los que el uso de espacio es bastante útil para la lectura del código: (WordPress, 2015).

```
[php]
function my_function( $complete_array = null, $key_1 = 4, $key_2 = 'bar' ) {
if ( null == $complete_array ) {
$final_array = $complete_array;
} else {
$key_1 = (integer) $key_1;
$final_array[0] = 'this';
$final_array[ $key_1 ] = 'is';
$final_array[ $key_2 ] = 'an';
$final_array['last'] = 'example';
}
return $final_array;
}/php]
```

**Figura 7.** Uso de espacio en PHP.

**Fuente:** elaboración propia.

### Convenciones de nomenclatura

Éste puede ser difícil acostumbrarse, especialmente si vienes de otros lenguajes de programación. En pocas palabras: (WordPress, 2015).

- Los nombres de variables deben estar en minúsculas, palabras separadas por guiones bajos
- Los nombres de clase deben usar palabras en mayúsculas separadas por guiones bajos. Los acrónimos deben estar en mayúsculas
- Constantes deben estar en mayúsculas, separadas por guiones
- Los nombres de archivo deben estar en minúsculas, separados con guiones

### Condiciones Yoda

Escribir condiciones al revés te podría ayudar a prevenir errores de análisis. Puede ser un poco raro, pero es mejor para el código (WordPress, 2015).

```
[php]if ( 'alex' === $name ) {
echo 'Podría escribir un artículo';
}/php]
```

**Figura 8.** Condiciones Yoda.

**Fuente:** elaboración propia.

### 3.3.2 HTML

HTML no tiene que bastantes reglas asociadas a él, llega a mucho para hacer las cosas más modulares. Sólo hay cinco reglas que usted necesita saber la hora de escribir HTML: (WordPress, 2015).

- El código debe validar contra el validador del W3C.
- Las etiquetas HTML de cierre automático deben tener exactamente un espacio antes de la barra diagonal (es una especificación del W3C, no un motivo favorito de WordPress)
- Los atributos y etiquetas deben estar en minúsculas. La única excepción es cuando los valores de atributos están destinados para el consumo humano, en cuyo caso deben ser escritos de forma natural.
- Todos los atributos deben tener un valor y deben ser citado (`<input disabled>` No es la escritura correcta)
- La sangría debe lograrse mediante pestañas y debe seguir la estructura lógica.

### 3.3.3 CSS

CSS es otro lenguaje de programación bastante flexible para escribir, por lo que hay un montón de trabajo por hacer aquí. Aun así, las buenas prácticas son bastante fácil al momento de codificar (WordPress, 2015).

#### Selectores

Los selectores deben ser tan cualificados como sea necesario, ser humanamente legible, estar en minúsculas con palabras separadas con guiones y selectores de atributos deben utilizar comillas dobles. He aquí un ejemplo justo de lo que debes hacer: (WordPress, 2015).

```
[css]input[type="text"],
input[type="password"],
.name-field {
background: #f1f1f1;
} [[/css]
```

**Figura 9.** Selectores de código CSS.

**Fuente:** elaboración propia.

**Orden de propiedad**

Las reglas reconocen la necesidad de un poco de espacio personal aquí, ya que no prescriben un orden específico de reglas CSS. Lo que no dicen es que debes seguir una estructura semántica que tenga sentido. Las propiedades del grupo por sus relaciones o agruparlos alfabéticamente, simplemente no las escriben de forma aleatoria (WordPress, 2015).

El agregar las propiedades se basa en la lógica en la que se desee visualizar el contenido:

- visualización
- Posicionamiento
- Modelo de caja
- Los colores y la tipografía
- Otros

```
[css].profile-modal {  
display: block;  
position: absolute;  
left: 100px;  
top: 90px;  
background: #ff9900;  
color: #fff;  
} [ /css]
```

**Figura 10.** Orden de Prioridades de código CSS.

**Fuente:** elaboración propia.

### 3.3.4 JavaScript

*JavaScript* se ha convertido en un lenguaje bastante usado en la web, en WordPress no ha sido la excepción para ser usado. Para usar JavaScript en WordPress, debes basarte en la guía de jQuery, en la cual se refieren a la longitud de una línea de programación, la cual no debe ser extensa (WordPress, 2015).

#### **Punto y coma**

Es algo que sucede y es bastante común, tal vez sea la regla más sencilla es la que se pasa por alto con mayor frecuencia. Nunca, jamás, debes omitir un punto y coma, por la sencilla razón que tu código no podría funcionar (WordPress, 2015).

Asegúrate de que tu código tenga un punto y coma al final de cada sentencia válida, recuerda que los corchetes no llevan el punto y coma al final.

Sangría: Las tabulaciones siempre se deben utilizar sangría. También debes añadir espacio en el contenido de un cierre, incluso si el contenido de un archivo está contenido en uno (WordPress, 2015).

Líneas De Ruptura: Limitar hasta dónde llega determinada instrucción, nos sirve para conocer en qué punto termina y comienza un nuevo bloque de código. Con esto también nos ahorramos el dolor de cabeza que nos podría dar el punto y coma (WordPress, 2015).

Ciclos Con jQuery: De acuerdo con las normas de iteración con jQuery (`jQuery.each ()`) sólo debe usarse en objetos jQuery. Puedes usar los típicos ciclos de `for/in` para iterar otros tipos de datos.

### **3.4- Estrategia de Validación del portal web UJC de La Lisa.**

Al construir un software habitualmente se cometen errores. En la Industria, la técnica para solucionar los problemas derivados de dichos errores, serán las pruebas de software, que consistirán en una serie de pasos realizados antes y después de la construcción de este software.

Las pruebas son parte fundamental de cualquier proyecto, ya que ayudará a tener mejores resultados, a ofrecer una calidad mayor de nuestro producto y en consecuencia los clientes estarán más satisfechos.

Según Ilene Burnstein, el proceso de prueba de software tiene tres procesos principales: el desarrollo de los casos de prueba, la ejecución de estos casos de prueba y el análisis de los resultados de la ejecución (Peño, 2015).

### **3.4.1- Pruebas de Rendimiento**

La prueba de rendimiento se diseña para poner a prueba el rendimiento del software en tiempo de corrida, dentro del contexto de un sistema integrado. La prueba del rendimiento ocurre a lo largo de todos los pasos del proceso de prueba. Incluso en el nivel de unidad, puede accederse al rendimiento de un módulo individual conforme se realizan las pruebas. Sin embargo, no es sino hasta que todos los elementos del sistema están plenamente integrados cuando puede determinarse el verdadero rendimiento de un sistema (BIBIÁN, 2017).

Al Portal web se le realizó la prueba de rendimiento de tipo carga y estrés. Se utilizó la herramienta Apache JMeter en su versión 3.0, diseñado para pruebas de cargas de comportamientos funcionales y la medición del rendimiento. El ambiente de prueba estuvo conformado por:

#### **Hardware de prueba (PC servidor):**

- Sistema Operativo: Windows 11
- Microprocesador: Intel(R) Core (TM) i3-7100U CPU @ 2.40GHz 2.40 GHz
- Memoria RAM: 8.00 GB
- Disco Duro: 1000 GB
- Disco Sólido: 120 GB

#### **Hardware de prueba (PC cliente):**

- Sistema Operativo: Windows v.10
- Microprocesador: Intel(R) Core (TM) i5-5200U CPU @2.20GHz 2.20GHz
- Memoria RAM: 4.00 GB
- Disco Duro: 500 GB

**Software instalado en ambas PC:**

- Tipo de servidor web: Apache 3.0 (PC servidor).
- Plataforma: SO, Windows (PC servidor) y SO Windows (PC cliente).
- Servidor de BD: PostgreSQL 11.1 (PC cliente).

Luego de haber definido el *hardware* se configuran los parámetros del Apache JMeter logrando un total de 30 usuarios conectados concurrentemente. En la tabla 5 se puede observar los resultados obtenidos por el sistema.

Parámetros que componen la tabla para una mejor interpretación:

- **Usuarios:** Total de usuarios.
- **Muestras:** El número de peticiones.
- **Min:** El mínimo tiempo transcurrido en mili-segundos para las muestras de la URL dada.
- **Max:** El máximo tiempo transcurrido en un mili-segundos 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 *Kbyte* por segundos.

**Tabla 5.** Resultado de las pruebas de Carga y Estrés.

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

---

**Fuente:** elaboración propia.

### **Análisis de los resultados de las pruebas de rendimiento**

De los resultados obtenidos en las pruebas se determina que la aplicación cumple con los requisitos que se presentan. El informe de Apache JMeter arrojó 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.

También se le realizaron 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 le realizó 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.

### **3.4.2- Pruebas Unitarias**

Las pruebas unitarias se realizan sobre las funcionalidades internas de un módulo y se encargan de comprobar los caminos lógicos, ciclos (bucles) y condiciones que debe cumplir el programa (Pressman, 2010).

Para aplicar las pruebas unitarias, los autores de la presente investigación definen utilizar el método de Caja blanca. Con el empleo de este método es posible desarrollar casos de prueba que garanticen la ejecución, al menos una vez, de los caminos independientes (Pressman, 2010). Para aplicar este método se define el método de caja blanca.

**Método de caja blanca:** se aplica a las funcionalidades externas del sistema, con el objetivo de evaluar la complejidad lógica de un diseño procedimental y usar esta medida como guía para la definición de un conjunto básico de caminos de ejecución. Esta prueba permite garantizar que en los casos de prueba obtenidos a través del camino básico se ejecute cada sentencia del programa por lo menos una vez. La generación del grafo de flujo tiene que descomponer las condiciones compuestas en condiciones sencillas (Pressman, 2010). Para aplicar este método se define la técnica de Ruta básica que a continuación se describe.

### Táctica de ruta básica

La técnica de ruta básica es empleada en el método de Caja blanca, la misma tiene como objetivo comprobar que cada camino se ejecute independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Esta técnica de prueba debe ser utilizada para evaluar la efectividad de los métodos asociados a una clase, con el objetivo de asegurar que cada camino independiente sea ejecutado por lo menos una vez en el sistema (Pressman, 2010).

Pressman propone como estrategia para aplicar la ruta básica, realizar un análisis de la complejidad ciclomática de cada procedimiento que componen las clases del sistema; una vez concluido este paso se selecciona el método con valor de contener errores, además de que ofrece una medida del número de pruebas que deben diseñarse para validar la correcta implementación de una determinada función.

Para obtener los casos de prueba a partir de la técnica ruta básica, se debe construir el grafo de flujo para cada uno de los métodos desarrollados, a continuación se muestra un ejemplo de la aplicación de la técnica correspondiente al código del método `get(self, request, *args, **kwargs)` en la clase que valida los permisos de los usuarios al ingresar alguna vista de los módulos. Se toma como ejemplo ya que es uno de los métodos utilizados a nivel de seguridad. En la siguiente figura 11 se muestra dicho método y la identificación de cada uno de los nodos:

```

1  def post(self, request, *args, **kwargs):
2      try:
3          action = request.POST[ 'action' ]
4          if action = 'add':
5              with translation.atomic():
6                  products = json.loads(request.POST[ 'products' ])
7                  acta = Acta()
8                  acta.start_date = request.POST[ 'start_date' ]
9                  acta.end_date = request.POST[ 'end_date' ]
10                 acta.user_id = int(request.POST[ 'user' ])
11                 acta.description = request.POST[ 'description' ]
12                 acta.manifestation_id = int(request.POST[ 'manifestation' ])
13                 acta.save()
14                 for i in products:
15                     detail = ActaDate()
16                     detail.acta_id = acta.id
17                     detail.product_id = int(i[ 'id' ])
18                     detail.cant = int(i[ 'cant' ])
19                     detail.save()
20                     detail.acta.stock -= detail.cant
21                     detail.acta.save()
22             else:
23                 data[ 'error' ] = 'No ha ingresado a ninguna opcion'
24         except Exception as e:
25             dataclasses[ 'error' ] = str(e)
26         return JsonResponse(DatagramProtocol, safe=False)

```

Figura 11. Método utilizado como ejemplo para la técnica de ruta básica.

Fuente: elaboración propia.

A continuación, se muestra, en la figura 12, el grafo de flujo obtenido a partir del método anteriormente descrito.

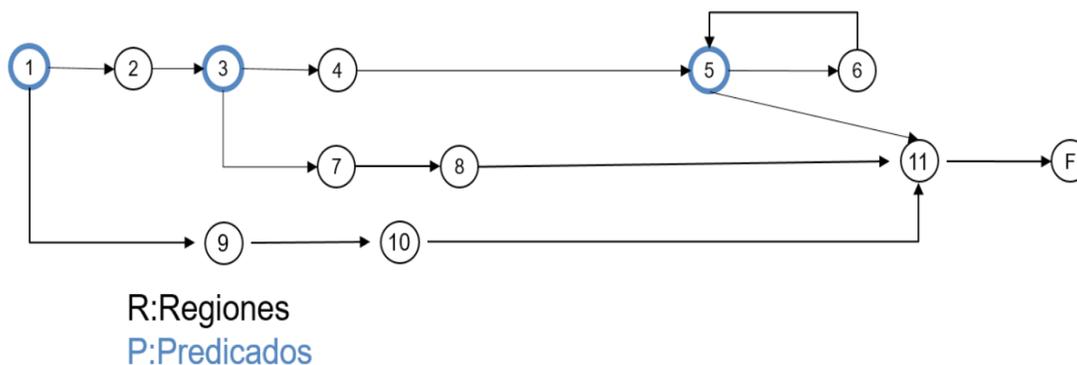


Figura 12. Grafo de flujo de método `get(self, request, *args, **kwargs)`.

**Fuente:** elaboración propia.

Luego se determina la complejidad ciclomática  $V(G)$  del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición. La complejidad ciclomática se obtiene a través de las siguientes variantes:

1.  $V(G) = R$
2.  $V(G) = E - N + 2$
3.  $V(G) = P + 1$

Conociendo que:

- **G:** Grafo de flujo (grafo)
- **R:** El número de regiones contribuye a estimar el valor de la complejidad ciclomática.
- **E:** Número de aristas
- **V(G):** Complejidad ciclomática
- **N:** Número de nodos del grafo
- **P:** Número de nodos predicados (nodos de donde parten al menos dos aristas).

Realizando los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado:

$$V(G) = R \text{ ----- } V(G) = 4$$

$$V(G) = E - N + 2 \text{ ----- } V(G) = 15 - 13 + 2 \text{ ----- } V(G) = 4$$

$$V(G) = P + 1 \text{ ----- } V(G) = 3 + 1 \text{ ----- } V(G) = 4$$

Una vez calculada la complejidad ciclomática, el valor obtenido representa el límite superior de pruebas que debe aplicarse (Pressman, 2010). Por lo que los conjuntos de caminos básicos son:

Ruta independiente 1: 1-2-3-4-5-6-5-11-F

Ruta independiente 2: 1-2-3-4-5-11-F

Ruta independiente 3: 1-2-3-7-8-11-F

Ruta independiente 4: 1-9-10-11-F

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos, en cada una de las funciones de la solución. A continuación, solamente se presenta el caso de prueba definido para la ruta independiente 2.

**Tabla 6.** Caso de Prueba de la ruta independiente 1.

<b>Descripción</b>	Se toma los datos que son llegados a través del método POST y son guardados en Acta y seguidamente se le asocia al acta, los eventos y sus respectivas cantidades al modelo ActaDate, concluyendo así el método.
<b>Condición de ejecución</b>	Se deben enviar datos a través del método POST
<b>Entradas</b>	<pre> action=='add', acta.start_date = request.POST['start_date'], acta.end_date = request.POST['end_date'], acta.user_id = int(request.POST['user']), acta.description = request.POST['description'], acta.manifestation_id = int(request.POST['manifestation']), detail.acta_id = int(i['id']), detail.descrip = int(i['cant']) </pre>
<b>Resultados esperados</b>	Se debe mostrar un mensaje especificando que se ha creado un pedido.

**Fuente:** elaboración propia.

### Resultados al aplicar la técnica de ruta básica

Esta técnica se aplicó a los métodos de las clases controladoras; estas clases fueron seleccionadas debido a que engloban las funcionalidades del sistema. WordPress posee un sistema de pruebas au-

tomatizadas con el marco de ejecución de pruebas y otras utilidades; puede simular solicitudes, insertar datos de prueba, inspeccionar la salida de su aplicación y en general verificar que el código esté correcto. Para la validación y completamiento de las pruebas fueron programadas seis clases de pruebas que realizan comprobaciones a los modelos y las vistas de la plataforma que garantizan que se probó el código a los principales módulos del sistema. Se ejecutaron cada uno de los casos de pruebas de las seis clases *tests.wp* en busca de errores de codificación, como condición de parada se tuvo en cuenta lo definido como resultado esperado en los casos de prueba. En ninguno de los casos se detectaron No conformidades, ya que en todos los casos el resultado fue el esperado.

### 3.4.3- Pruebas de Seguridad

Cualquier sistema basado en computadora que gestione información sensible o cause acciones que puedan dañar (o beneficiar) de manera inadecuada a individuos es un blanco de penetración inadecuada o ilegal. La penetración abarca un amplio rango de actividades: *hackers* que intentan penetrar en los sistemas por deporte, empleados resentidos que intentan penetrar por venganza, individuos deshonestos que intentan penetrar para obtener ganancia personal ilícita. La prueba de seguridad intenta verificar que los mecanismos de protección que se construyen en un sistema en realidad lo protegerán de cualquier penetración impropia (BIBIÁN, 2017).

A continuación, se muestran los plugins de seguridad instalados en la plataforma de WordPress:

1. **Limit Logging Attempts** (Versión: 2.25.8): detiene los ataques de fuerza bruta y optimiza el rendimiento de tu sitio limitando el número de intentos de inicio de sesión que son posibles a través del inicio de sesión normal, así como XMLRPC, Woocommerce y páginas de inicio de sesión personalizadas. Este complemento, con más de 2 millones de descargas en todo el mundo. Este plugin bloqueará una dirección de Internet (IP) y/o un nombre de usuario para que no realice más intentos después de que se haya alcanzado un límite específico de reintentos, lo que dificulta o imposibilita un ataque de fuerza bruta. Por defecto, WordPress permite intentos de acceso ilimitados. Esto puede hacer que las contraseñas se descifren fácilmente mediante la fuerza bruta (WordPress, 2020).
2. **MalCare WordPress Security** (Versión: 4.82): es el complemento de detección y eliminación de malware más rápido amado por miles de desarrolladores y agencias. Con una eliminación automática de malware con un solo clic, la primera en la industria, su sitio web de WordPress está limpio antes de que Google lo incluya en la lista negra o su proveedor de alojamiento web lo elimine. MalCare se ha desarrollado desde cero después de analizar más de 240 000 sitios web du-

rante más de 2 años y medio. Su metodología de escaneo inteligente nunca ralentizará su sitio de WordPress e identifica con precisión el malware más complejo que normalmente pasa desapercibido en otros complementos de seguridad populares de WP (WordPress, 2020).

### 3.4.4- Pruebas de Funcionales

Este tipo de pruebas se basa en las funcionalidades de un sistema que se describen en la especificación de requisitos, es decir, lo que hace el sistema (Peño, 2015).

#### Objetivo

Verificar la función del sistema al fijar la tensión en la validación de las funciones, métodos, servicios y casos de usos.

Tiene como meta validar que la aplicación:

- ✓ Cumpla con los requisitos funcionales especificados en el diseño de la solución.
- ✓ Cumpla con los requisitos No funcionales especificados en el diseño de la solución.
- ✓ Cumpla con las restricciones de entrada y salida de la información especificada en el diccionario de Datos, de cada caso de uso.
- ✓ Cumpla íntegramente con la estructura especificada en el Mapa de Navegación.

A continuación, se muestran ejemplos de diseño de casos de prueba de aceptación utilizado para detectar errores en la aplicación y mostrar si cumplía con los requisitos. Se describe el caso de prueba

#### **RF.25 - Crear plantilla:**

CAPÍTULO 3

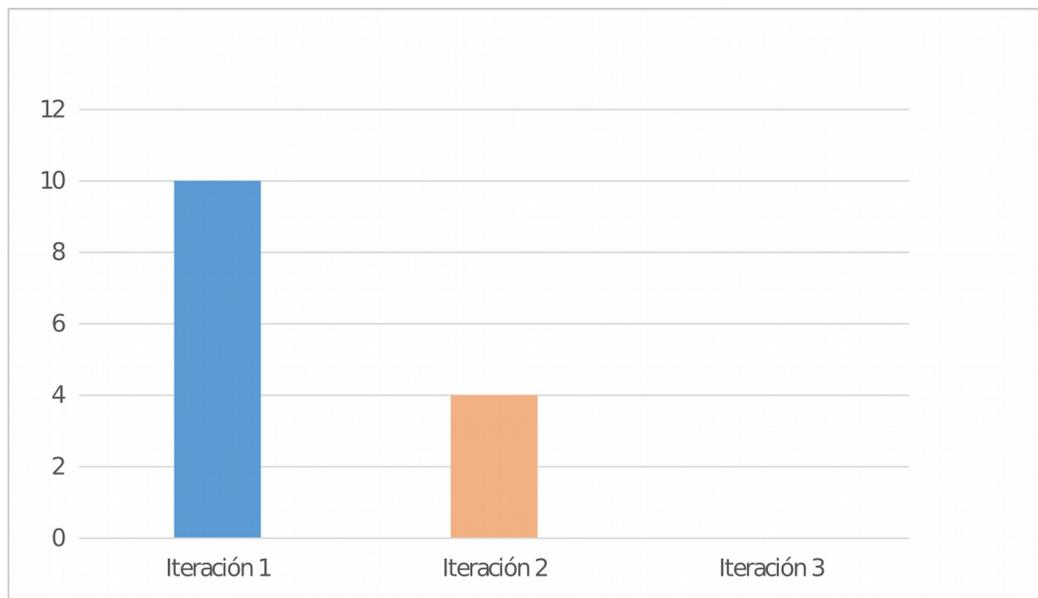
Escenario	Descripción	Título	Cuerpo	Imagen	Palabras Claves	Fecha	Autor	Respuesta del sistema
EC 1 Crear el acta de forma correcta.	El sistema adiciona el artículo de forma correcta.	X	Texto de prueba	Subir una imagen que cumpla con las restricciones	Secretarios, orden del día, desarrollo, acuerdos.	20/10/2022	Alejandro Nuñez Rivas	El sistema almacena la información y muestra un mensaje de confirmación de la acción: "Contenido Artículo se ha creado."
EC 2 Crear el acta de forma incorrecta.	El sistema adiciona el artículo de forma incorrecta.	X	Texto de prueba	Subir un archivo que no cumpla con las restricciones de tamaño, formato falso	Datos incorrectos con respecto a los requisitos de un acta	20/10/2022	Alejandro Nuñez Rivas	El sistema no almacena la información y muestra los siguientes mensajes de error:  "El archivo no se pudo guardar porque sobrepasa los 2 MB, el tamaño máximo para las subidas."  "El archivo de imagen no es válido o el tipo de imagen no está permitido. Tipos permitidos: png, jpg, gif"
EC 3 Crear el acta dejando	El sistema no adiciona artículo	Vacío	Vacío	No	No	22/10/2022	Alejandro Nuñez Ri-	El sistema no almacena la información y

campos vacíos.	vacíos dejando campos obligatorios vacíos.							vas	muestra los siguientes mensajes de error:  " El campo Título es obligatorio."  "El campo Cuerpo es obligatorio."
Flujo Central									
<ol style="list-style-type: none"> <li>1. El usuario accede al bloque superior Contenido/+Agregar Contenido o Agregar Contenido y el sistema muestra todos los tipos de contenidos que pueden ser añadidos.</li> <li>2. El usuario selecciona el tipo de contenido Artículo.</li> <li>3. El sistema muestra un formulario para que el usuario introduzca la información.</li> <li>4. El usuario introduce la información y presiona el botón: "Guardar".</li> </ol>									

**Tabla 7.** Caso de prueba basado en el RF "Crear Acta".

**Fuente:** elaboración propia.

A continuación, se muestran los resultados obtenidos en tres de las iteraciones de pruebas realizadas al portal web de la UJC de la Lisa, la Figura 13 brinda información del total de no conformidades encontradas y las que fueron resueltas por cada iteración. Con un total de 45 requisitos funcionales se detectaron 10 no conformidades en la primera iteración y 4 inconformidades en la segunda iteración, resolviéndose satisfactoriamente, en la tercera iteración no se encontró ninguna no conformidad.



**Figura 13.** Resultados de las Pruebas Funcionales.

**Fuente:** elaboración propia.

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

- Errores de estructuración en los contenidos mostrados en las vistas: se solucionó haciendo una reestructuración de los contenidos.
- Errores ortográficos.
- Errores de funcionamiento en algunos bloques: se arreglaron instalando módulos, corrigiendo el código de programación, entre otros.
- Errores de funcionamientos en módulos: se instalaron módulos con versiones anteriores o fueron sustituidos por otros con funciones parecidas adaptadas al portal.

### 3.4.5- Pruebas de Usabilidad

Según diversos estándares de la Ingeniería de Software, se puede definir la usabilidad como el grado en el que un producto puede ser utilizado por usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción, en un determinado contexto de uso. Como se puede apreciar, la usabilidad de un sistema está ligada a usuarios, necesidades y condiciones específicas (Carcasés, 2016).

Para la realización de las pruebas de usabilidad, se hace uso de la “Lista de Chequeo de Usabilidad para sitios web”, desarrollada por los especialistas del grupo de Seguridad del Departamento de Evaluación de Productos de Software (DEPSW), perteneciente al Centro Nacional de Calidad de Software (CALISOFT). A continuación, se muestran los resultados de dichas pruebas.

**Tabla 8.** Resultado de las pruebas de usabilidad.

Categoría de los indicadores	Indicadores	Proceden	Correctos	Incorrectos
Visibilidad del sistema	17	18	17	1
Lenguaje común entre sistema y usuario	11	7	7	0
Libertad y control por parte del usuario	29	22	21	1
Consistencia y estándares	33	23	23	0
Estética y diseño minimalista	18	6	6	0
Prevención de errores	8	3	2	1
Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	11	6	6	0
Ayuda y documentación	11	7	7	0
Flexibilidad y eficiencia de uso	6	6	6	0
<b>Total</b>	<b>144</b>	<b>98</b>	<b>95</b>	<b>3</b>

**Fuente:** elaboración propia.

Como se observa en la tabla, de los 144 parámetros originales de la lista de chequeo, solo proceden 98, se evaluaron como correctos 95 parámetros, identificando 3 no conformidades las cuales fueron resueltas, para un 100 % de usabilidad.

### 3.4.6- Pruebas de Aceptación

Las pruebas de aceptación comparan el comportamiento del sistema con los requisitos del cliente, sea cual sea la forma en que estos se hayan expresado. El cliente realiza, o específica, tareas típicas para comprobar que se satisfacen los requisitos o que la organización los ha identificado para el mercado al que se destina el software. Esta actividad puede incluir o no a los programadores (BIBIÁN, 2017).

Al concluir la implementación del portal web se pone a disposición de un grupo de usuarios y especialistas los cuales van a emitir su criterio una vez que hayan interactuado con este. El criterio de usuarios se analiza mediante la técnica de ladov y el de los especialistas mediante el método de preferencia.

#### Satisfacción de usuarios con el portal

La técnica de ladov constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov" de cada sujeto en la escala de satisfacción (Rodríguez, 2018).

En la investigación se utilizó la técnica de ladov la cual permitió medir el grado de satisfacción de los usuarios respecto a la solución desarrollada, aquellos que accedan a sus contenidos e interactúen con el portal. En el cuadro siguiente se muestran los resultados de dicha encuesta.

**Tabla 9.** Cuadro Lógico de ladov para usuarios potenciales.

	P#1 ¿Considera usted que el portal web Pensando Américas deba brindar servicios haciendo uso de la nueva propuesta que mejora la interfaz de usuario y la gestión de información?		
	<b>Sí</b>	<b>No sé</b>	<b>No</b>
P#2 ¿Utilizaría este portal web para acceder o recopilar información que mejoren la experiencia del usuario en cuanto a su uso?	P#3 ¿Le satisface el portal web desarrollado para mejorar la gestión de la información, debido a su nuevo diseño de interfaz y arquitectura de información?		

Sí	No sé	No	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me satisface mucho											4	5	2	1	2	4	4	4	4	1
No me satisface tanto											3	5	2	2	5	3	4	2	4	
Me da lo mismo											6	2	4	3	1	6	4	5	1	
Me insatisfecho más de lo que me satisface											2	3	1	4	2	3	5	2	1	
No me satisface nada											1	2	4	2	3	5	1	2	3	
No sé qué decir											2	3	4	1	3	4	2	4	1	

**Fuente:** elaboración propia.

La escala de satisfacción se muestra a continuación:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

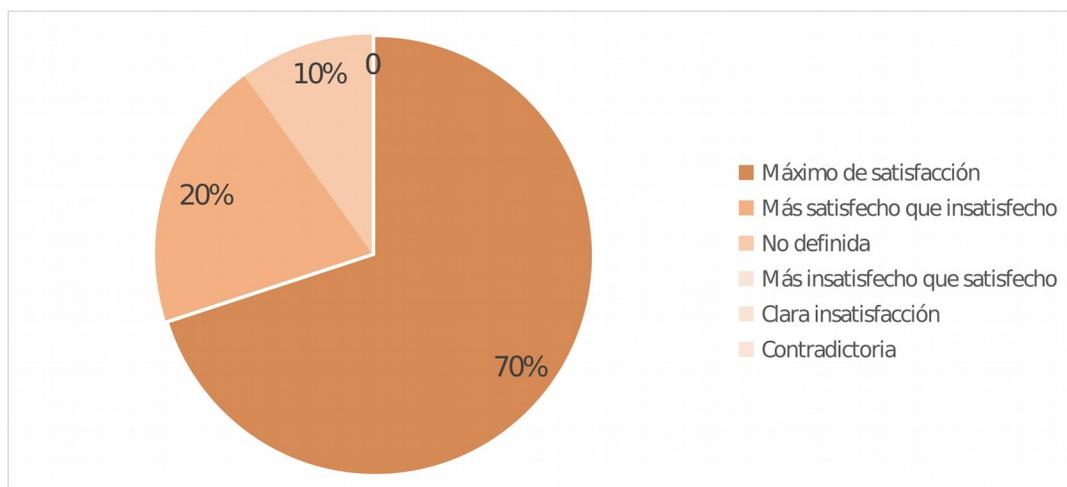
Para medir el grado de satisfacción del portal se escogió una muestra de 10 usuarios que tuvieran experiencia de la navegación de portales webs. La siguiente tabla muestra el grado de satisfacción de dichos usuarios.

**Tabla 10.** Satisfacción de usuarios potenciales.

Resultado	Cantidad	%
Máximo de satisfacción	7	70,00%
Más satisfecho que insatisfecho	2	20,00%
No definida	1	10,00%
Más insatisfecho que satisfecho	0	0%
Clara insatisfacción	0	0%
Contradictoria	0	0%

**Fuente:** elaboración propia.

A continuación, se representa los porcentajes de satisfacción de los usuarios en el portal.



**Figura 14.** Satisfacción de usuarios potenciales con el sitio web.

**Fuente:** elaboración propia.

En la figura 12 se muestra un esquema de pastel en el cual se refleja el porcentaje de satisfacción por parte de los usuarios en el portal mostrando un 70% de máximo de satisfacción, un 20% más satisfecho que insatisfecho, un 10% no definido y el resto un 0% de clara insatisfacción y contradictoria.

### 3.5 Criterios de expertos

Teniendo en cuenta la idea a defender que se plantea en la presente investigación, para analizar la relación causa efecto entre la variable independiente: “sistema para la gestión de información” y las variables dependientes: “celeridad y control”, se definen un conjunto de criterios de medida que permiten validar cómo a través de la propuesta de solución se logra la relación entre ambas variables. La obtención de estos criterios se realiza a partir de las principales deficiencias identificadas en la situación problemática y de los resultados de la encuesta realizada a los especialistas funcionales judiciales de los TPC.

Criterios de medida definidos:

- Tiempo de recopilar los datos al registrar un acta.
- Tiempo de búsqueda de un acta.
- Tiempo de notificación del retraso de la entrega de un acta.
- Control de la información.

A continuación, en la Tabla 11, se evalúan cada uno de los criterios de medida definidos. La evaluación se realiza a través de un antes de poseer el sistema y un después de poderlo usar, con el propósito de verificar cómo mediante la solución propuesta se contribuye a la celeridad del proceso de la gestión de la UJC de la Lisa. Los datos tenidos en cuenta en la comparación fueron tomados a partir de la realización de las pruebas de rendimiento, comparando el tiempo de respuestas del sistema con el tiempo promedio que se demoran los especialistas para cada una de las actividades que responden a los criterios de medida. Estos datos fueron tomados de los resultados de la encuesta realizada a los estudiantes y trabajadores de la UJC del municipio.

**Tabla 11.** Validación de las variables de investigación.

Atributo	Antes	Después
Tiempo para recopilar los datos al registrar un acta.	Según los resultados de la encuesta, se determinó que al registrar los escritos en el subproceso de forma manual se consume un tiempo elevado ya que, los especialistas plantean que se demoran entre 10 y 15 minutos para realizar esta actividad	Se disminuye considerablemente el tiempo, solo demora alrededor de 4 minutos en realizar un registro de un préstamo en el sistema.
Tiempo de búsqueda de un acta.	A raíz de la encuesta los especialistas plantean que se tardan hasta 2 minutos para realizar esta actividad.	Esta actividad utilizando el GEPRE se comprobó que se disminuyó el tiempo a 15 segundos aproximadamente.
Tiempo de notificación del retraso de la entrega de un acta.	A raíz de la encuesta los especialistas plantean que se tardan entre 1 a 4 días para realizar esta actividad.	Esta actividad utilizando el GEPRE se comprobó que automáticamente que se vence el plazo de entrega, se le notifica al secretario mediante correo electrónico y mediante el sistema sobre el vencimiento del plazo de entrega del acta.
Control de la información.	Los mismos datos con otras denominaciones. Llamadas telefó-	Tipología definida por tipo de acta. Notificaciones automáti-

---

---

	nicas y correos.	cas.
--	------------------	------

**Fuente:** elaboración propia.

La comparación realizada en la tabla anterior, a través de los criterios de medida antes definidos, demuestra cómo, a través de la personalización obtenida como resultado de la presente investigación, se contribuye a la celeridad del proceso de préstamos en la Dirección de Extensión Universitaria de la UCI.

### Conclusiones del capítulo

- La confección del modelo de despliegue nos permitió describir la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.
- La confección del diagrama de componentes permitió la integración de los componentes de software
- Al aplicar los estándares de codificación facilitó la comprensión y legibilidad del código fuente, permitiendo desarrollar el portal en un menor tiempo y garantizando la facilidad y calidad de un futuro mantenimiento.
- La realización de las pruebas de seguridad, funcionalidad, carga y estrés, guiaron la calidad de la aplicación y correcta ejecución de las funcionalidades del sistema.
- Las consultas a expertos garantizaron identificar y corregir las distintas no conformidades detectadas y así obtener un producto de mayor calidad.

### CONCLUSIONES GENERALES

El presente Trabajo de Diploma concluye con el desarrollo de un portal web para la UJC de La Lisa, el cual sirve de apoyo a la gestión y divulgación del quehacer de la organización.

Se destaca, además que:

- A partir de la sistematización de los principales referentes teóricos que sustenta la investigación se evidencia que el diseño de la propuesta de solución permitió generar los artefactos más significativos de acuerdo con la metodología de desarrollo de software AUP-UCI.
- La implementación del portal a través de las herramientas y lenguajes seleccionados permitió obtener un portal web capaz de gestionar la información referente a las actividades y el acontecer de eventos del Municipio de La Lisa; logrando el cumplimiento del objetivo planteado.
- La definición de una estrategia de prueba, permitió comprobar el correcto funcionamiento del portal web para la UJC de la Lisa, a partir del cumplimiento de los requisitos definidos por el cliente.
- Las técnicas de validación aplicadas a la propuesta de solución permitieron la detección y corrección de las no conformidades detectadas y evidenciaron que el portal constituye una solución funcional.

## RECOMENDACIONES

Con el cumplimiento del marcado objetivo en la presente investigación y luego de alcanzado el resultado esperado, se recomienda:

- Implementar una funcionalidad que integre los datos el portal web con el sistema SICOM.
- Continuar agregándole funcionalidades al portal web según las necesidades que se requieran para mejorar el proceso de gestión de la información, teniendo en cuenta los avances en las TIC.
- Brindar las posibilidades de navegar en el portal en distintos idiomas para lograr la internacionalización de las publicaciones.

## REFERENCIA BIBLIOGRAFÍA

**A. Santa Fe. (2014).** ALEGSA. DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA. <http://www.alegsa.com.ar/Dic/framework.php>.

**Alexeis García. (2011).** SCIELO.

[http://scielo.sld.cu/scielo.php?pid=S1024-94352001000300003&script=sci\\_arttext&tIng=en](http://scielo.sld.cu/scielo.php?pid=S1024-94352001000300003&script=sci_arttext&tIng=en).

**Arias. (2017).** Programación Web con PHP.

**Base Datos Ofimaticas. (2015).**

<https://basedatosofimaticas.wikispaces.com/3+Características+de+los+Sistemas+Gestores+de+Bases+de+Datos>.

**BIBIÁN. (2017).** Pruebas de calidad aplicadas al sitio web ALLISON. COLIMA-MEXICO: secretaria de Educación Pública.

**Cadavid, Fernández Martínez, & Morales Vélez. (2013).** Redalyc Prospectiva, <https://www.redalyc.org/pdf/4962/496250736004.pdf>.

**Carcasés. (2016).** Guía para los procesos de Usabilidad que se desarrollan en el Laboratorio de Pruebas de Software (LPS) de la Dirección de Calidad-UCI.

**C. Tejada. (2011).** Metodología de desarrollo de aplicaciones basada en PMBOK y metodologías.

**Chimbote U.C. (2017).** Metodología de Software, <http://repositorio.uns.edu.pe/handle/UNS/2843>.

**Díaz Llanes. (2016).** Módulos para la gestión de la información de los procesos de Cotización e Ingreso a las filas de la UJC en la UCI. <https://repositorio.uci.cu/handle/123456789/7797>.

**Fe, A. S. (2014).** ALEGSA. DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA. <http://www.alegsa.com.ar/Dic/framework.php>.

**Fergarciac. (2013).** <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.

**Findstack. (2022).** Obtenido de Findstack About: <https://findstack.com/es/about/>.

**Francisco G. & Andrea. (2018).** Modelo de Dominio - Ingeniería de Software I.

- Gabriel M. Barrera. (2018).** Estilo arquitectónico para aplicaciones IoT. Serie Documentos de Trabajo, Universidad del CEMA, <https://www.econstor.eu/handle/10419/203805>.
- Group. (2012).** ¿Qué es PHP? <http://php.net/manual/es/intro-what-is.php>.
- Guerrero, Suárez y Gutiérrez. (2013).** Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. [https://www.scielo.cl/scielo.php?pid=S071807642013000300012&script=sci\\_arttext&lng=en](https://www.scielo.cl/scielo.php?pid=S071807642013000300012&script=sci_arttext&lng=en).
- J. Eguiluz. (2011).** Desarrollo web ágil con Symfony2. <http://standars.optimeconsulting.net/wp-content/uploads/2014/12/desarrollo-agil-symfony-2.4.pdf>.
- Jesús Alioska. (2018).** Sitio para la gestión de información del Centro Universitario Municipal. EduVarona, <http://revistas.ucpejv.edu.cu/index.php/rVid/article/view/600>.
- Jimenez-González5 (2018).** web para La gestión de información en la escuela de capacitación de a Construcción de Pinar del Río, Cuba.
- Luján-Mora. (2012).** Repositorio Internacionales de la Universidad de Alicante, <http://rua.ua.es/dspace/handle/10045/16995>.
- M. A. Álvarez. (2009).** Desarrollo Web. <http://www.desarrolloweb.com/articulos/499.php>.
- Oya Suryana. (2008).** Membangun Blog, [https://books.google.es/books?hl=es&lr=&id=e7xaAfx\\_7wEC&oi=fnd&pg=PA1&dq=wordpress+blog&ots=fXhxdLZt1Q&sig=WTEuBdZOPPnEmyEbvAlZJgZucl#v=onepage&q=wordpress%20blog&f=false](https://books.google.es/books?hl=es&lr=&id=e7xaAfx_7wEC&oi=fnd&pg=PA1&dq=wordpress+blog&ots=fXhxdLZt1Q&sig=WTEuBdZOPPnEmyEbvAlZJgZucl#v=onepage&q=wordpress%20blog&f=false).
- Pablo Huet. (2022).** OpenWebinars. <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
- Peño. (2015).** Pruebas de Software. Fundamentos y técnicas, Madrid.
- PSEO. (2022).** <https://www.psoe.es/>.
- Pérez Gómez. (2017).** Las TIC en la resolución de problemas y pensamiento crítico, <https://repositorio.libertadores.edu.co/handle/11371/1470>.
- Pieterse y Marshall. (2010).** What is a Design Pattern? <https://citeseerx.ist.psu.edu/viewdoc>.
- PISCITELLI. (2005).** Editorial Gedisa, Barcelona, <https://santamaria.edu.uy/JPM/wp-content/uploads/2011/03/6PISCITELLI-Alejandro-Ecologia-de-la-Red.pdf>.

- PostgreSQL. (2011).** [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
- Pressman. (2005).** La Ingeniería del Software, un enfoque práctico.
- Pressman. (2010).** Los Requisitos del Software.
- Pressman. (2013).** Ingeniería de Software un enfoque práctico. Journal of Chemical Information and Modeling. <https://doi.org/10.1017/CBO9781107415324.004>.
- RJ. Advance. (2019).** Patrones de Arquitectura-Estilos. Patrones de Arquitectura Estilos. <http://rjco-deadvance.com/Patrones de Arquitectura-Estilos>.
- Rodríguez. (2018).** Metodología de desarrollo para la actividad productiva de la UCI.
- SIGE. (2022).** <https://consultasvenezuela.com/sistema-de-gestión-escolar>.
- SOMMERVILLE. (2011).** Software engineering 9th Edition. ISBN-10, vol. 137035152.
- Suraj. (2019).** A review paper on bootstrap framework.  
<https://www.irejournals.com/formatedpaper/1701173.pdf>.
- Vidal LMJ. (2012).** *Revista Cubana de Educación Médica Superior*, <https://www.medigraphic.com/cgi-bin/new/resumen.cgi?IDARTICULO=35891>.
- VP Internacional. (2013).** Visual Paradigm for UML product overview.
- W3Techs. (2022).** W3Techs - World Wide Web Technology Surveys: <https://w3techs.com/>
- WordPress. (2020).** [wordpress.com](https://wordpress.com).

## ANEXOS

## Anexo 1: Historia de Usuario “Crear Usuario”

<b>No. 2</b>	<b>Nombre:</b> Crear Usuario	
<b>Programador:</b> Alejandro Nuñez Rivas		<b>Interacción Asignada:</b> 1era
<b>Prioridad:</b> Media		<b>Tiempo Estimado:</b> 1 h
<b>Riesgo de Desarrollo:</b> Medio		<b>Tiempo Real:</b> 30 minutos
<p><b>Descripción:</b> Los usuarios con rol Superadmin pueden crear nuevos usuarios en el sistema, para ello deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> <li>● Nombre de usuario (Obligatorio. Campo de texto. Long min. 5, máx. 60 caracteres. Permiten los caracteres: (a-z, A-Z, 0-9))</li> <li>● Dirección de correo electrónico (Obligatorio. Campo de texto. Long máx. 254 caracteres. Permite direcciones de correo estructuralmente válidas (Estructura: usuario@-subdominios.dominio)).</li> <li>● Contraseña (Obligatorio. Campo de texto. Long min. 6, máx. 128 caracteres. Permite los caracteres a-z, A-Z, 0-9 y signos de puntuación).</li> <li>● Confirmar contraseña (Obligatorio. Campo de texto. Long min. 6, máx. 128 caracteres. Permite los caracteres a-z, A-Z, 0-9 y signos de puntuación).</li> <li>● Estado (Opcional. Campo de selección. Valores (Bloqueado, Activo). Valor por defecto (activo)).</li> <li>● Roles (Opcional. Campo de selección múltiple. Valores (Superadmin, administrador)).</li> <li>● Notificar al usuario acerca de su nueva cuenta (Opcional. Campo de selección. Valor por defecto (Desmarcado)).</li> </ul>		
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1- Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el usuario.</li> <li>2- Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.</li> </ol>		

- 3- Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse.
- 4- Si el usuario ya existe, el sistema emite un mensaje notificando que el usuario ya está en el sistema.

---

**Anexo 2: Caso de prueba basado en el requisito funcional "Crear Autor"**


---

Escenario	Descripción	Nombre	Contraseña	Imagen	Fecha	Rol	Respuesta del sistema
EC 1 Crear usuario de forma correcta.	El sistema adiciona usuario de forma correcta.	alejandrnr	*****	Imagen del usuario	27/10/2022	admin	El sistema almacena la información y muestra un mensaje de confirmación: "Usuario Agregado"
EC 2 Crear el usuario de forma incorrecta.	El sistema adiciona el usuario de forma incorrecta.	alejandrnr	*****	Subir un archivo que no cumpla con las restricciones de tamaño, formato falso	27/10/2022	admin	El sistema no almacena la información y muestra los siguientes mensajes de error:  "El archivo no se pudo guardar porque sobrepasa los 2 MB, el tamaño máximo para las subidas."  "El archivo de imagen no es válido o el tipo de imagen no está permitido. Tipos permitidos: png, jpeg, jpg, jpe, gif"
EC 3 Crear el usuario dejando campos vacíos.	El sistema no adiciona el usuario dejando campos obligatorios vacíos.	Vacío	Vacío	No	28/10/2022	admin	El sistema no almacena la información y muestra los siguientes mensajes de error:  " El campo usuario es obligatorio."  "El campo contraseña es obligatorio."

---

**Anexo 3:** Encuesta realizada al personal relacionado con el proceso de la UJC.

Esta encuesta es considerada anónima y personal, dirigida a los trabajadores y estudiantes que pertenecen a la UJC del municipio. El objetivo de la encuesta es determinar el tiempo promedio que demoran habitualmente al realizar un acta de la actividad correspondiente, la celeridad es muy importante para el perfeccionamiento de nuestro trabajo.

Por favor complete la encuesta cuidadosamente al leerla primero, y luego señale sus respuestas con una "X".

1. ¿Qué tiempo demora recopilar los datos al registrar un préstamo?
  - a)  Menos de 1 minuto.
  - b)  De 5 a 10 minutos.
  - c)  De 10 a 15 minutos.
  - d)  Otro ¿Cuál? \_\_\_\_\_
  
2. Si el técnico desea controlar la entrega de un préstamo realizado en una fecha determinada. ¿Qué tiempo promedio demora en encontrar el préstamo en los registros donde se encuentran almacenados dicho préstamo?
  - a)  Menos de 1 minuto.
  - b)  De 1 a 3 minutos.
  - c)  Más de 3 minutos.
  
3. Una vez que vence el tiempo de entrega de un préstamo, se le debe notificar a las personas el incumplimiento de no entregar los recursos prestados. Para esta actividad ¿qué tiempo promedio demora la notificación a los usuarios de este incumplimiento?
  - a)  Menos de 1 día.
  - b)  Entre 1 y 4 días.
  - c)  Más de 4 días.