

Universidad de las Ciencias Informáticas

Facultad 3

Dirección de Informatización



**Mantenimiento perfectivo de los Directorios del portal
web Intranet**

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

Autora:

Beatriz Pérez Delgado

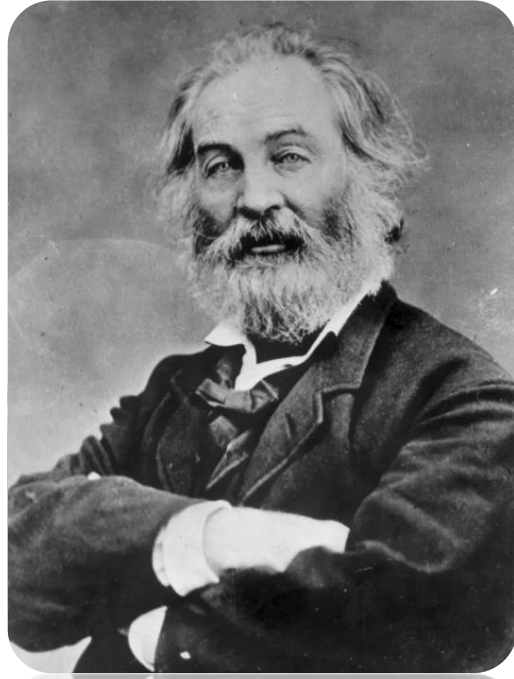
Tutores:

Drc. Yoan MartinezMarquez

Ing. Jany Coto García

La Habana, junio de 2020

“Año 62 de la Revolución”



"No dejes que termine el día sin haber crecido un poco, sin haber sido feliz, sin haber aumentado tus sueños, no te dejes vencer por el desaliento"

Walt Whitman

DECLARACIÓN DE AUTORÍA

Declaro ser la única autora del trabajo de diploma con título “Mantenimiento perfectivo de los Directorios del portal web Intranet” y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación con carácter exclusivo.

Firma de la Autora

Beatriz Pérez Delgado

Firma del Tutor

Drc. Yoan Martinez Marquez

Firma del Tutor

Ing. Jany Coto García

Agradecimientos

Hoy después de un largo recorrido este sueño se convierte en realidad. Quisiera comenzar agradeciendo a las personas más especiales de mi vida, a mi madre, por todos los sacrificios realizados desde que me dio la vida, por apoyarme en cada decisión, por ser madre, pero ante todo por ser amiga.

A mi padre por ser desde que tengo uso de razón ese ejemplo a seguir, esa figura masculina que en mi vida lo puede todo, por encaminarme en la vida y desde pequeña inculcarme la idea de ser independiente.

La vida no me alcanzará para pagarles todo lo que han hecho por mí y en ocasiones, aunque no se los demuestre los amo más que a mí misma.

A mi hermana por brindarme ese apoyo incondicional y estar presente en todos los momentos de mi vida.

A mis abuelos, aunque no están todos, sé que estarían orgullosos de verme echa una ingeniera.

A mis tías, primos y toda la familia en general por siempre brindarme ese apoyo y comprensión.

A mis amigas y amigos Rachel, Arletys, Sindia, Yi, Leo, Albertico, Kkchi y los que han estado a lo largo de estos años brindándome lo mejor de ellos y me han demostrado que puedo contar con ellos cuando más lo he necesitado.

A mis tutores por su paciencia y dedicación.

A todos los profesores que a lo largo de mi vida han contribuido en mi formación personal y profesional.

A Fidel, a la Revolución y a la Universidad de las Ciencias Informáticas.

Dedicatoria

Dedico este trabajo a mis padres, por ser mi fuente de fuerza e inspiración.

A mis hermanos y sobrinos, que los quiero con toda el alma y es el mejor regalo que mis padres me han podido dar.

A mis abuelos, tías y primos, aunque hoy están lejos, siempre los tengo muy cerca y sin su ayuda me hubiera sido difícil llegar tan lejos.

A la lista de amigos interminable que me han acompañado en estos años y a todas aquellas personas que de una forma u otra me apoyaron para que este día se volviera realidad.

Resumen

En la Universidad de las Ciencias Informáticas surge la estrategia de informatizar los procesos que se llevan a cabo en ella, para ello brinda diferentes servicios de búsqueda de información. El portal web Intranet es uno de los servicios que brinda la universidad. Dicho portal tiene integrado los directorios de personas, aplicaciones, teléfonos y cumpleaños de la universidad.

La presente investigación estuvo orientada a fundamentar y documentar el proceso de mantenimiento perfectivo de los directorios del portal web Intranet. Se realizó un análisis de fuentes bibliográficas relacionadas con el mantenimiento perfectivo y de sistemas existentes con objetivos similares que contribuyen a brindar información en diferentes instituciones nacionales e internacionales. Análisis que permitió definir todos los requisitos funcionales y no funcionales necesarios para el desarrollo del mantenimiento perfectivo de los directorios del portal web Intranet.

El proceso fue guiado por la metodología ágil de desarrollo de *software* Scrum y se utilizó el marco de trabajo Drupal en su versión 8. El sistema gestor de base de datos empleado fue PostgreSQL 10 y PhpStorm como entorno de desarrollo. Para lograr una aplicación con la menor cantidad de errores, se le realizaron pruebas de unidad, validación y sistema.

Como resultado del trabajo de diploma se obtuvo el mantenimiento perfectivo de los directorios del portal web Intranet en una nueva versión de Drupal, integrándoles nuevas funcionalidades al directorio de personas.

Palabras claves:

mantenimiento perfectivo, directorios, búsquedas, información

Índice.

Introducción.....	1
Capítulo 1: Fundamentación teórica de la Investigación.....	6
1.1 Conceptos y aspectos asociados al dominio del problema.....	6
1.1.1 Sistemas de búsqueda de información.....	8
1.1.2 Directorios.....	8
1.2 Estudio de sistemas homólogos.....	9
1.2.1 Directorio de la Universidad de Burgos.....	9
1.2.1 Directorio de la Universidad de Granada.....	9
1.2.2 ROAD.....	9
1.2.3 Directorio telefónico de ETECSA.....	10
1.2.4 Directorios de la Universidad de Ciencias Informáticas.....	11
1.3 Metodología, herramientas y tecnologías.....	13
1.3.1 Metodologías de desarrollo del <i>software</i>	13
1.3.2 Herramientas, tecnologías y lenguajes.....	15
1.4 Conclusiones parciales.....	23
Capítulo 2: Propuesta de solución de la Investigación.....	24
2.1 Descripción del sistema propuesto.....	24
2.2 Roles del proyecto.....	24
2.3 Especificación de los requisitos del sistema.....	25
2.3.1 Requisitos funcionales del sistema.....	25
2.3.2 Requisitos no funcionales del sistema.....	30
2.4 Patrón Arquitectónico.....	31
2.5 Patrones de diseño.....	32
2.5.1 Patrones GoF.....	32
2.6 Diagrama de despliegue.....	34
2.7 Conclusiones parciales.....	36

Capítulo 3: Implementación y Pruebas de la Investigación..... 37

- 3.1 Estándares de codificación 37
- 3.2 Validación de requisitos 42
 - 3.2.1 Criterios para validar los requisitos 42
 - 3.2.2 Técnicas de validación de requisitos 43
- 3.3 Pruebas 44
 - 3.3.1 Pruebas de unidad 44
 - 3.3.2 Prueba de integración 47
 - 3.3.3 Pruebas de sistema 49
 - 3.3.4 Pruebas de aceptación 51
- 3.4 Conclusiones parciales 52

Conclusiones..... 53

Recomendaciones..... 54

Referencias Bibliográficas. 55

Anexos 60

Índice de tablas.

TABLA 1: COMPARACIÓN DE LOS SISTEMAS SIMILARES EN CUANTO A LOS INDICADORES.....	12
TABLA 2: ROLES DEL PROYECTO	24
TABLA 3: REQUISITOS FUNCIONALES DEL SISTEMA	25
TABLA 4: HISTORIA DE USUARIO DEL REQUISITO FUNCIONAL REALIZAR BÚSQUEDA AVANZADA DE PERSONAS.	29
TABLA 5: ESTRATEGIA DE PRUEBAS.....	44

Índice de figuras.

FIGURA 1: PATRÓN ARQUITECTÓNICO MVC (ELABORACIÓN PROPIA).	32
FIGURA 2: EJEMPLO DEL PATRÓN SINGLETON.(ELABORACIÓN PROPIA).....	33
FIGURA 3: EJEMPLO DE UTILIZAR EL PATRÓN OBSERVADOR. (ELABORACIÓN PROPIA).....	33
FIGURA 4: EJEMPLO DE USO DEL PATRÓN ADAPTER. (ELABORACIÓN PROPIA)	34
FIGURA 5: DIAGRAMA DE DESPLIEGUE (ELABORACIÓN PROPIA)	34
FIGURA 6: EJEMPLO DEL USO DE LAS COMILLAS EN EL PORTAL WEB INTRANET.(ELABORACIÓN PROPIA).....	38
FIGURA 7: EJEMPLO DEL USO DEL PUNTO Y COMA EN EL PORTAL WEB INTRANET. (ELABORACIÓN PROPIA). 39	
FIGURA 8 EJEMPLO DEL USO DE OPERADORES EN EL PORTAL WEB INTRANET. (ELABORACIÓN PROPIA)	39
FIGURA 9: EJEMPLO DEL USO DE ESTRUCTURAS DE CONTROL EN EL PORTAL WEB INTRANET. (ELABORACIÓN PROPIA).....	39
FIGURA 10: EJEMPLO DEL USO DE LLAMADOS DE FUNCIÓN EN EL PORTAL WEB INTRANET. (ELABORACIÓN PROPIA).....	40
FIGURA 11: DECLARACIÓN DE ARREGLOS 1. (ELABORACIÓN PROPIA)	41
FIGURA 12: DECLARACIÓN DE ARREGLOS 2. (ELABORACIÓN PROPIA)	41

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC), han impactado en distintas áreas de la sociedad, facilitando la informatización de diversos sectores. Están presentes en todas las actividades y procesos que se realizan en las organizaciones, como herramienta necesaria en la gestión y búsqueda de información y como garantía para alcanzar la eficiencia, eficacia y efectividad. Por tanto, el uso de estas tecnologías ha conllevado al desarrollo de sistemas automatizados con búsquedas especializadas que permiten encontrar datos específicos de un tema determinado a gran velocidad. Búsquedas que se pueden iniciar por medio de palabras clave que identifican de una forma u otra el tema de interés de las personas. A medida que la información es más creciente e ilimitada, resulta necesario para las instituciones tener mecanismos que permitan localizarla, manipularla y tratarla.

En la actualidad, con los desarrollos tecnológicos y los avances en la ciencia computacional, hay sin duda un factor clave y decisivo en todos los sectores que es: la información; se puede afirmar que quien tiene la información, clara, concreta, segura y confiable domina a los demás en cualquiera que sea su área. Es por ello, que los sistemas de información son para las empresas como el sistema nervioso es para el humano; si no existieran los sistemas de información, se retrocede a la etapa primitiva, debido a que todo que rodea al ser humano son sistemas de información que agrupándolos ayudan a la toma de decisiones.

Un sistema de información es un conjunto de datos que interactúan entre sí con un fin común. En informática, estos ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización. La importancia de un sistema de información radica en la eficiencia y en la correlación de una gran cantidad de datos ingresados a través de procesos diseñados para cada área, con el objetivo de producir información válida para la posterior toma de decisiones (Chen, 2019).

Iván García expresa que el sistema de información en la empresa constituye el conjunto de recursos de la compañía que sirven como soporte para el proceso de captación, transformación y comunicación de la información. Un sistema de información debe resultar eficaz y eficiente. Resultará eficaz cuando facilita la información precisa, y será eficiente si lo hace con la menor cantidad posibles de recursos (García, 2018).

Los sistemas de información están cambiando, en la actualidad, la forma en que operan las organizaciones. Mediante su uso se obtienen grandes mejoras, ya que automatizan los procesos

operativos que se pueden llevar a cabo en toda empresa, proporcionan información de apoyo al proceso de tomas de decisiones y facilitan el logro de ventajas competitivas a través de su implantación dentro de las organizaciones (Hamidian y Ospino, 2015).

Por otro lado, constituye una tendencia para las instituciones implementar el uso de los sistemas de búsqueda de información como una estrategia eficiente en el intercambio de información y localización del personal que las conforman, pues su función es brindar información a los usuarios para que estos puedan consultar y nutrirse de datos específicos a partir de sus necesidades.

En Cuba se evidencian diversos tipos de sistemas de búsqueda de información, uno de ellos es la Guía Telefónica de ETECSA (Empresa de Telecomunicaciones de Cuba S. A.) la cual llega de forma anual y en formato impreso a los usuarios que tienen contrato de telefonía fija con dicha entidad. También cuenta con un sistema electrónico, llamado Páginas Amarillas que muestra información comercial de entidades del país. Otro ejemplo se evidencia en el Directorio Turístico que brinda información de los lugares más visitados dentro del país. Existen otras organizaciones como las universidades y centros de estudios que utilizan sistemas informáticos similares para consultar datos específicos de toda su comunidad.

La Universidad de las Ciencias Informáticas (UCI) es una de las instituciones que no ha quedado exenta a los avances tecnológicos de hoy en día. La universidad persigue la estrategia de informatizar los procesos que se llevan a cabo en ella, para ello brinda diferentes servicios de búsqueda de información, dentro de los diferentes servicios se pueden destacar el sistema de gestión universitaria, la biblioteca digital, el entorno virtual de aprendizaje, el portal Intranet, el periódico Mella y el Directorio UCI.

El portal web Intranet tiene integrado los servicios de los directorios universitarios (personas, teléfonos, cumpleaños y aplicaciones), aunque la solución propuesta se concibió por etapas, solo se venció la primera de ellas, que tenía como solución el directorio de personas. La tecnología empleada para el desarrollo del portal Intranet en su versión 3.0 fue el marco de trabajo Drupal 7. Drupal es un sistema de gestión de contenidos o *Content Management System* (por sus siglas en inglés CMS) que solo cuenta con soporte hasta noviembre del 2020, lo que trae consigo que una vez que culmine el tiempo de soporte la comunidad no se responsabiliza de garantizar soluciones a los disímiles problemas que se puedan presentar, que pueden ser: de rendimiento, mantenibilidad o escalabilidad. Por lo antes expuesto es necesario desarrollar el portal en una nueva tecnología.

La situación descrita anteriormente se puede resumir en las siguientes limitantes:

- ✓ El tiempo de respuesta del sistema pudiera ser menor con el avance tecnológico de hoy en día.
- ✓ Limitaciones en la codificación del sistema lo que trae consigo que al modificar algunas de sus partes afecta al *software* completo.
- ✓ Limitaciones en el avance del desarrollo de los directorios de manera que permitan el empleo de todos aun cuando estos no presenten todas sus funcionalidades.

A partir de la problemática anteriormente descrita se define como **problema de investigación** el siguiente: ¿Cómo contribuir a la escalabilidad, modularidad y disminución del tiempo de respuesta de los Directorios Universitarios del portal Intranet?

Para dar solución al problema antes expuesto se enmarca el **objeto de estudio** en el mantenimiento de *software* en los portales web, enfocando **el campo de acción** en el mantenimiento de *software* perfectivo de los Directorios del portal Intranet.

Como **objetivo general** se plantea perfeccionar la implementación de los directorios del portal Intranet de manera que contribuya a la escalabilidad, modularidad y disminución de su tiempo de respuesta. Se definen los siguientes **objetivos específicos**:

- Realizar la revisión bibliográfica sobre el mantenimiento de *software* de gestión.
- Identificar el estado actual del mantenimiento de *software* perfectivo de los Directorios del portal Intranet.
- Implementar los directorios de la Intranet de manera que contribuya a la escalabilidad, modularidad, y disminución de su tiempo de respuesta.
- Valorar la viabilidad de los directorios de la Intranet de manera que contribuya a la escalabilidad, modularidad, y disminución de su tiempo de respuesta.

Para dar cumplimiento a los objetivos específicos se proponen como **tareas de investigación**:

- Sistematización de los referentes teóricos sobre el mantenimiento de *software* de gestión.
- Aplicación de encuestas a los usuarios y caracterización de los sistemas homólogos.
- Codificación de los directorios de la Intranet de manera que contribuya a la escalabilidad, modularidad, y disminución de su tiempo de respuesta.
- Ejecución de las pruebas para valorar la viabilidad de los directorios de la Intranet de manera que contribuya a la escalabilidad, modularidad, y disminución de su tiempo de respuesta.

Durante el desarrollo de la investigación se emplean los siguientes métodos científicos:

Métodos teóricos

- **Histórico-Lógico:** se utiliza este método para realizar el estudio de sistemas homólogos investigando acerca de la evolución de otras aplicaciones o herramientas similares y metodología de desarrollo de *software* existente de manera tal que se pueda comprender cómo funcionarán en la presente investigación.
- **Analítico-Sintético:** permite realizar un análisis de los elementos relacionados con los sistemas de búsqueda y directorios existentes para posteriormente sintetizar dicho estudio y definir los diferentes conceptos que se manejan en la investigación. Así como un estudio de las herramientas, tecnologías y lenguajes a utilizar durante todo el desarrollo.

Métodos empíricos

- **Encuesta:** se realiza con el objetivo de recolectar información relacionada sobre el uso de la versión actual del sistema y así obtener la opinión de los usuarios referente a la necesidad de contar con un sistema que responda a sus necesidades como cliente. Se les realiza a los trabajadores de la Dirección de Informatización (DIN), para entender el proceso del negocio, el funcionamiento y la estructura de los procesos actuales, además de las dificultades existentes, obteniendo la mayor cantidad de información posible, para determinar las funcionalidades a implementar.
- **Entrevista:** se utiliza este método para facilitar mediante encuentros con los administradores del sistema, entender todo el proceso del negocio, el funcionamiento y la estructura de los procesos actuales, además de las dificultades existentes, obteniendo la mayor cantidad de información posible, para determinar las nuevas funcionalidades a implementar (Anexo1).
- **Revisión documental:** se realizó consultas de libros y artículos científicos digitales para el estudio de los referentes teóricos. Se citó fundamentalmente bibliografía del período 2011-2019 para asegurar la actualidad de los aspectos abordados.

Luego de terminada la investigación y comenzar a utilizar los resultados de la misma, se obtendrá como beneficio la mejora del proceso de búsqueda de información de personas, cumpleaños, teléfonos y aplicaciones de la UCI en el portal Intranet de manera que contribuya a la escalabilidad, modularidad, y rendimiento del *software*.

El presente trabajo investigativo se estructura en tres capítulos:

Capítulo 1. Fundamentación teórica: en este capítulo se abordan los conceptos fundamentales que tienen relación con el problema planteado. Se describen algunas soluciones existentes en el ámbito nacional e internacional que gestionan el proceso de abastecimiento de productos permitiendo obtener una valoración de las mismas. Finalmente se justifican las herramientas y tecnologías a utilizar para el desarrollo del sistema.

Capítulo 2. Propuesta de solución: en este capítulo se especifican los requisitos funcionales y no funcionales de la propuesta de solución, se describen todos los artefactos que se generan durante el diseño del sistema y se definen los patrones arquitectónicos que se van a utilizar. Además, se abordan las características fundamentales de los Directorios del portal web Intranet de la Universidad de las Ciencias Informáticas.

Capítulo 3. Implementación y pruebas: se formaliza la implementación de la propuesta de solución. A su vez se confeccionan los casos de prueba a realizar al *software*, se ejecutan y se presentan los resultados que permiten validar la solución propuesta.

Capítulo 1: Fundamentación teórica de la Investigación.

En el presente capítulo se realiza un estudio donde se muestran los principales conceptos y aspectos tratados en la investigación. Se valoran los sistemas existentes relacionados al dominio del problema y se realiza una breve descripción de las tecnologías a utilizar definidas por la Dirección de Informatización (DIN).

1.1 Conceptos y aspectos asociados al dominio del problema.

El objetivo de la ingeniería del *software* es producir un sistema, aplicación o producto de alta calidad. Para lograr este objetivo, se deben emplear métodos efectivos junto con herramientas modernas dentro del contexto de un proceso maduro de desarrollo del *software*. Al mismo tiempo se debe medir si la alta calidad se va a llevar a cabo a lo largo del desarrollo del *software* (Pressman y Maxim, 2015). A continuación, se verán tres variables que pueden emplearse en la valoración de la calidad de *software*. Además, se exponen dos aspectos importantes que están estrechamente relacionados al dominio del problema.

Escalabilidad

Según el autor Moreno(2019),se entiende por escalabilidad a la capacidad de crecimiento de un sistema, una red o un proceso. Es la propiedad deseable de toda aplicación que indica su habilidad para reaccionar, adaptarse y ampliarse para dar soluciones a nuevas funcionalidades, estar preparado para hacerse más grande, manejar el crecimiento continuo de trabajo de manera fluida y sin perder calidad en los servicios ofrecidos.

Se entiende por escalabilidad a la capacidad de adaptación y respuesta de un sistema con respecto a su rendimiento a medida que aumenta de forma significativa el número de usuarios del mismo. La escalabilidad supone un factor crítico en el crecimiento de un sistema (Ramos et al., 2017).

La autora define la escalabilidad como la propiedad de aumentar la capacidad de trabajo o tamaño de un sistema sin comprometer el funcionamiento y calidad del mismo.

Modularidad

La modularidad proporciona un mecanismo para realizar la filosofía. Cualquier sistema complejo puede dividirse en módulos (componentes), pero una buena práctica de ingeniería de *software* exige más. La modularidad debe ser efectiva. Es decir, cada módulo debe enfocarse exclusivamente en un aspecto bien restringido del sistema: debe ser coherente en su función y / o restringido en el

contenido que representa. Además, los módulos deben interconectarse de una manera relativamente simple: cada módulo debe exhibir un bajo acoplamiento con otros módulos, con fuentes de datos y con otros aspectos ambientales (Pressman y Maxim, 2015).

Según Sommerville (2011), la modularidad es el único atributo del *software* que permite gestionar un programa intelectualmente.

En la presente investigación la autora define la modularidad como la capacidad que tiene un sistema de ser estudiado, visto o entendido como la unión de varias partes que interactúan entre sí y que trabajan para alcanzar un objetivo común, realizando cada una de ellas una tarea necesaria para la consecución de dicho objetivo.

Rendimiento

El rendimiento se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos, el conjunto y la eficiencia. El mismo se prueba bajo condiciones operativas, configuraciones y cargas diferentes a fin de asegurar que el sistema responde a la interacción con el usuario y que maneja la carga extrema sin degradación operativa inaceptable (Pressman y Maxim, 2015).

La autora define el rendimiento como la rapidez con la que una función de *software* aparece para realizar su tarea. Se aplica principalmente a los aspectos de aceptación de usuario.

Mantenimiento del *software*

El autor Sommerville (2011), define el mantenimiento del *software* como el proceso general de cambiar un sistema después de que éste ha sido entregado. El término se aplica normalmente a *software* a medida en donde grupos de desarrollo distintos están implicados antes y después de la entrega. Los cambios realizados al *software* pueden ser cambios sencillos para corregir errores de código, cambios más extensos para corregir errores de diseño o mejoras significativas para corregir errores de especificación o acomodar nuevos requerimientos. Los cambios se implementan modificando los componentes del sistema existente y añadiendo nuevos componentes al sistema donde sea necesario.

Existen 4 tipos de mantenimiento que son:

- Mantenimiento adaptativo
- Mantenimiento preventivo
- Mantenimiento correctivo

- Mantenimiento perfectivo.

En la presente investigación se utilizará el mantenimiento perfectivo, debido a que el mantenimiento perfectivo puede significar perfeccionar el *software* implementando nuevos requerimientos; además, significa mantener la funcionalidad del sistema, pero mejorando su estructura y su rendimiento (Sommerville, 2011).

1.1.1 Sistemas de búsqueda de información.

Los sistemas de búsqueda de información son sistemas automáticos de recuperación de información que buscan todo tipo de información (imágenes, vídeos, documentos, etc...) en la *World Wide Web*, almacenándola en una enorme base de datos para arrojar la información solicitada. Además, son un conjunto de elementos interrelacionados destinados a la captura, depuración, almacenamiento, recuperación, actualización y tratamiento de dato; por lo que constituyen un eslabón importante en la devolución de resultados asociados a la información que desee el usuario.

Aunque buscar información en la red no sea fácil, existen herramientas de búsqueda que ayudan a encontrar lo que se desea, haciendo coincidir las palabras clave que el interesado indica en la búsqueda con la información más relevante que existe en la red. Existe una gama de recursos y herramientas que posee Internet para realizar búsquedas de información, entre las cuales se encuentran los **Motores de búsqueda o metabuscadores** que son sistemas de búsqueda por palabras claves, incorporan automáticamente páginas web mediante *robots*¹ de búsqueda por la red. Como operan en forma automática, los motores de búsqueda contienen generalmente más información que los directorios (Rodríguez, 2012). Estos *robots* o arañas, recorren las páginas recopilando información sobre los contenidos de los sitios de Internet en bases de datos.

Otros son los **índices temáticos o directorios** que son sistemas de búsqueda por temas o categorías jerarquizados, aunque también incluyen sistemas de búsqueda por palabras clave. Se trata de bases de datos de direcciones web elaboradas "manualmente", es decir, hay personas que se encargan de asignar cada página web a una categoría o tema determinado (Rodríguez, 2012).

1.1.2 Directorios

Los directorios se puede definir como una lista de personas o de organizaciones ordenadas alfabética o sistemáticamente, proporcionando direcciones, cargos, funciones y datos similares, de tal manera que permite al investigador localizar, identificar y obtener informaciones acerca de las personas u

¹ Programas que buscan continuamente por todos los servidores de Internet construyendo un índice de lo hallado. También conocidos como «arañas» por su continuo desplazamiento sobre la red.

organismos existentes en una industria, mercado, grupo, sector de actividad o disciplina determinada (TORRES, 2013). Se puede afirmar que son herramientas de apoyo que permiten realizar búsquedas de información en dependencia de los criterios especificados por el usuario, que consulta y organiza ficheros o datos a partir de una búsqueda determinada, estos ficheros o datos pueden estar organizados jerárquicamente. Además, puede mostrar la información de distintas formas y maneras posibles en dependencia de lo descrito por el usuario.

1.2 Estudio de sistemas homólogos.

En Cuba y en el mundo se han desarrollado diferentes aplicaciones que gestionan la información de forma automatizada. En la investigación se realiza un análisis de estas soluciones con el fin de obtener una aproximación de las nuevas funcionalidades que debe tener la solución. A continuación, se detallan sus principales características y se hace una valoración acerca de los resultados obtenidos.

1.2.1 Directorio de la Universidad de Burgos.

La Universidad de Burgos cuenta con varios directorios como son el de estudios; becas, empleo y prácticas; conserjerías; centros e institutos; departamentos; órganos unipersonales del gobierno; secretarías de centro; servicios universitarios. Cada directorio muestra la dirección, teléfono, correo, y horario en caso de necesitarlo (Universidad de Burgos, 2019).

1.2.1 Directorio de la Universidad de Granada.

El Directorio de la Universidad de Granada es un módulo integrado al sistema de dicha universidad, lo que implica que sea totalmente dependiente de este, el mismo permite buscar personas y entidades. La búsqueda puede ser realizada mediante entidad, nombre, cargo o puesto, teléfono, y el email (Directorio UGR, 2019).

1.2.2 ROAD.

ROAD (*Directory of Open Access Scholarly Resources*, por sus siglas en inglés), proporciona un acceso gratuito a un conjunto de registros bibliográficos del ISSN (*International Standard Serial Number*, por sus siglas en inglés) de acceso abierto: revistas, series monográficas, actas de conferencias, repositorios académicos. Estos registros, creados por la Red ISSN se han enriquecido con metadatos procedentes de bases de indexación, directorios e indicadores de revistas.

ROAD está vinculado con las acciones realizadas por la UNESCO (*United Nations Educational, Scientific and Cultural Organization*, por sus siglas en inglés), para promover el acceso público a los

recursos científicos. ROAD es un complemento de GOAP (*Global Open Access Portal*, por sus siglas en inglés), desarrollado por la UNESCO. Este portal presenta el estado de la información científica en acceso abierto (libre y gratuito) en todo el mundo (International Standard Serial Number, 2019).

Principales funciones:

- Búsqueda por faceta.
- Búsqueda por zona geográfica.
- Búsqueda avanzada por país, por tema, por índice, por indicador de desempeño y desde luego por número de ISSN.
- Presentación de las bases de datos fuente, cuya información se utiliza para enriquecer los registros bibliográficos.

1.2.3 Directorio telefónico de ETECSA.

El Directorio telefónico de ETECSA es el Directorio Telefónico Oficial de Cuba, este además de constituir un soporte informativo de referencia para la sociedad y una útil herramienta en la gestión comercial, también se ha propuesto promover valores patrimoniales, históricos, culturales y medioambientales del país. Su estructura interna se presenta en varias secciones:

Páginas Informativas: se ubican al inicio de cada Directorio Telefónico y cuentan con un índice de contenidos en el que se puede encontrar información actualizada sobre los principales servicios que brinda ETECSA, códigos internacionales, teléfonos útiles de uso más frecuente y de urgencias, así como orientaciones generales para el uso del propio directorio. Es en esta sección donde se publican temas generales de impacto social, cultural, histórico, del ámbito de la ciencia, la meteorología, la ecología y otros.

Páginas Blancas: reportan en orden alfabético la razón social, nombres y apellidos, dirección y números telefónicos de los abonados residenciales y entidades de cada localidad.

Páginas Azules: esta sección reagrupa por orden alfabético la razón social, dirección y números telefónicos correspondientes a los Organismos de la Administración Central del Estado y otras Entidades Nacionales, así como su representación en las provincias contenidas en los volúmenes de cada zona.

Páginas Verdes: páginas destinadas a las informaciones de entidades del estado.

Páginas Amarillas: aparecen clasificadas por categorías todas aquellas entidades que ubican su información comercial bajo un rubro o categoría preferencial. Además, se encuentra organizada por categorías de actividades relacionadas con la producción y los servicios, subcategorías, provincias (en orden geográfico) y alfabéticamente de acuerdo a la voz de ubicación seleccionada por la propia entidad (el Nombre o Marca Comercial), además se publica la dirección y dos teléfonos (Empresa de Telecomunicaciones de Cuba S.A., 2016).

1.2.4 Directorios de la Universidad de Ciencias Informáticas.

La Universidad de las Ciencias Informáticas, como parte de la comunicación institucional cuenta con un portal de cara a internet que permite mantener informado del acontecer en la Universidad y de las principales actividades que allí se desarrollan dentro del área de formación, producción, investigación y extensionista. Por otra parte, está el Portal de la Intranet Universitaria, donde se muestran los diferentes acontecimientos e informaciones de interés para la comunidad. También se encuentra el Directorio UCI, es el sistema que posibilita la búsqueda de todas las personas que estudian o laboran en la institución.

El directorio del Portal de la Universidad solo muestra la información de los principales directivos y algunas de las personas involucradas en el proceso de investigación. Este agrupa las búsquedas por tres criterios: personas, universidades y teléfonos; a continuación, se describen brevemente sus funcionalidades.

El Directorio de Personas: muestra los datos generales de los principales directivos de la Universidad y las personas que participan activamente en el proceso de investigación. Permite buscar por categoría docente (Instructor recién graduado, Instructor, Asistente, Auxiliar y Titular) y por categoría científica (Máster y Doctor) (Universidad de las Ciencias Informáticas, 2018).

El Directorio de universidades cubanas: muestra un listado de universidades y centros universitarios que existen en el país. Este permite buscar por provincias (Universidad de las Ciencias Informáticas, 2018).

El Directorio Telefónico: muestra los teléfonos de las áreas de la Universidad y de sus principales directivos. Permite buscar por el nombre del área (Universidad de las Ciencias Informáticas, 2018).

El portal de la Intranet Universitaria contiene un módulo de los directorios universitarios (directorío de personas, teléfonos, aplicaciones y cumpleaños). Hasta el momento solo se encuentra disponible el directorío personas: el cual permite buscar especificando varios aspectos como: *Nombre, Apellidos,*

Usuario, Provincia, entre otros. Además, tiene la funcionalidad de añadir una o varias personas a la lista de favoritos (Universidad de las Ciencias Informáticas, 2019).

Por su parte, el Directorio UCI es el sistema que posibilita la búsqueda de todas las personas que estudian o laboran en la institución. El mismo permite realizar búsquedas por diferentes criterios dentro de lo que se incluyen:

Directorio de personas: permite buscar especificando varios aspectos como: *Nombre, Apellidos, Solapín, No. Expediente, Usuario*, entre otros. Además de apoyarse en diversos filtros de búsquedas como: *Categoría, Cargo, Área, Provincia, Municipio*, entre otros.

El Directorio Telefónico: permite buscar por los criterios: *Teléfono, Área, Ubicación y Nombre de pizarra*. Además de contar con las secciones de *Páginas amarillas y Urgencias*.

El Directorio de Aplicaciones UCI: muestra un listado con todas las aplicaciones por área de procesos de la universidad, con el Nombre, URL² y la Visibilidad del sitio ya sea institucional, nacional o internacional.

El Directorio de los Cumpleaños: muestra un abecedario de la A-Z y el calendario del año actual, para apoyar la búsqueda del usuario(Universidad de las Ciencias Informáticas, 2019).

Bajo(B), Medio(M), Alto(A)

Tabla 1: Comparación de los sistemas similares en cuanto a los indicadores.

Sistemas homólogos	ROAD	Universidad de Granada	Universidad de Burgos	Directorio de ETECSA	Directorios de la UCI
Indicadores					
Eficiencia de ejecución	Si	No	No	No	Si
Velocidad de procesamiento	No	No	Si	No	No
Adaptación a nuevas funcionalidades	Si	Si	No	No	Si
Adaptación a un nuevo entorno	No	Si	Si	No	Si
Independencia del sistema	No	No	No	No	Si
Modularidad efectiva	No	No	No	No	Si

²Localizador Uniforme de Recursos es una dirección que se asigna a cada uno de los recursos disponibles en la red para que estos puedan ser localizados o identificados.

Una vez analizado dicho resultado se puede concluir que, aunque el principal objetivo de las instituciones estudiadas anteriormente es proporcionar una herramienta capaz de brindar información de los integrantes de la institución y en el caso de los directorios de ETECSA brindar información de las personas que utilizan el servicio de telefonía fija, no se ajustan a la situación problemática planteada en la presente investigación. En ninguno de los casos se analiza la factibilidad de adquirir el *software* porque el objetivo de la investigación es incluir al sistema nuevas funcionalidades. Se concluye de esta caracterización que el sistema puede tener funcionalidades similares a los directorios de la UCI, pero siempre respetando el negocio.

1.3 Metodología, herramientas y tecnologías.

Para el desarrollo del mantenimiento perfectivo de los directorios del portal web Intranet se utilizará una metodología de desarrollo ágil, así como herramientas y tecnologías libres de licencia que favorezcan posteriormente la creación de nuevas funcionalidades al sistema que fueron seleccionadas a partir de la arquitectura definida por el proyecto. A continuación, se describen cada una de ellas.

1.3.1 Metodologías de desarrollo del *software*.

La selección de una metodología de desarrollo de *software* adecuada, es un factor determinante en el éxito de un proyecto. Aunque no existe una metodología absoluta, algunas se ajustan mejor que otras, a las características y necesidades específicas de los proyectos de desarrollo. Dentro de las metodologías existen dos grandes grupos, las conocidas metodologías tradicionales o robustas y las metodologías ágiles o ligeras. Las primeras enfatizan en el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto y es recomendada para los proyectos con grandes equipos de desarrollo. Mientras que las ágiles dan mayor importancia a la capacidad de respuesta a los cambios, se enfatiza en la satisfacción del cliente y promueve el trabajo en equipo (Pressman y Maxim, 2015).

Para el desarrollo del mantenimiento perfectivo de los Directorios del portal Intranet se utilizará la metodología Scrum, debido a que fue definida a partir de la arquitectura del proyecto.

Scrum: es considerada un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Muestra la eficacia relativa de las prácticas de gestión de producto y de desarrollo. Los equipos son auto organizados y multifuncionales. Los auto organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. Entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener

retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto (Schwaber y Sutherland, 2017).

Al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán *sprints* (Trigas, 2015)

Cómo funciona Scrum:

Para entender todo el proceso de desarrollo de Scrum, se describirá de forma general las fases, los roles y los elementos que forman parte de dicha metodología.

- Primera fase: Planificación del *Backlog*: se realizará un documento en el que se reflejarán los requisitos del sistema por prioridades. Se definirá también la planificación del Sprint 0, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esa iteración. Se obtendrá además un *SprintBacklog*, que es la lista de tareas y el objetivo más importante del Sprint.
- Segunda fase: Seguimiento del *sprint*: en esta fase se hacen reuniones diarias en las que las tres preguntas principales para evaluar el avance de las tareas son:
 - ¿Qué trabajo se realizó desde la reunión anterior?
 - ¿Qué trabajo se realizará hasta una nueva reunión?
 - Inconvenientes que han surgido y que hay que resolver para poder continuar.
- Tercera fase: Revisión del *sprint*.

Cuando se finaliza el *Sprint* se realiza una revisión del incremento que se ha generado y se presentan los resultados finales.

Roles de Scrum

- *ProductOwner*: es la persona que toma las decisiones, la que realmente conoce el negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el *ProductBacklog*.
- *ScrumMaster*: es el encargado de comprobar que el modelo y la metodología funcionan. Elimina todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores.

- Equipo de desarrollo: suele ser un equipo pequeño de unas cinco a nueve personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Está involucrado en la estimación del esfuerzo de las tareas del *Backlog*.

Elementos de Scrum

- *ProductBacklog* (PB): Es una lista que abarca todo lo necesario para que el producto satisfaga las necesidades de clientes potenciales. Es el documento central de un proyecto basado en Scrum y la principal referencia a la hora de realizar cambios o plantear soluciones.
- *Sprint Backlog* (SB) o Lista de objetivos pendientes del *sprint*: Es un subconjunto de elementos del PB seleccionados a abordar en el *Sprint*, más un plan para entregarlos como Incremento del producto. Cuando ciertos objetivos no se cumplen y las soluciones no pueden implementarse en el momento, es necesario trasladarlas al siguiente ciclo de trabajo. La lista de objetivos pendientes ayudará a tenerlos presentes.
- Incremento: es la forma en que se mide el progreso que ha tenido el proceso en cada etapa. Para esta metodología, es esencial que cada iteración tenga un incremento; si no es así, esto revelará que algo ha fallado.

1.3.2 Herramientas, tecnologías y lenguajes.

Las herramientas y tecnologías constituyen un elemento importante a la hora de desarrollar una solución, pues son capaces de proveer una arquitectura sólida y más extensible para la implementación de los diferentes componentes en un sistema informático. En este epígrafe se describe cada una de ellas y se describe la función que realiza dentro de la solución informática.

Drupal 8

El marco de trabajo para el desarrollo de la investigación fue Drupal en su versión 8.

Drupal es un sistema de gestión de contenidos o CMS libre, modular, multipropósito y muy configurable que permite publicar artículos, imágenes, archivos y que también ofrece la posibilidad de otros servicios añadidos como foros, encuestas, votaciones, blogs, administración de usuarios y permisos. Es un sistema dinámico, en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno web.

Es un programa libre, con licencia GNU/GPL, escrito en PHP, combinable con MySQL, PostgreSQL, SQLite y otros gestores de bases de datos. CMS desarrollado y mantenido por una activa comunidad de usuarios. Además, se destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet o intranet, también se destaca por su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar diferentes tipos de sitios web (Peñaranda, 2015).

Drupal en general sigue un diseño modular muy robusto y Drupal 8 en particular ha vuelto a mejorar su escalabilidad al incorporar Symfony a su core, un *framework* conocido por seguir la mayoría de mejores prácticas y patrones de diseño para la web que facilitan el extender las funcionalidades de las aplicaciones sin perder eficiencia y velocidad. Además, Drupal 8 ha mejorado sus tiempos de respuesta y el manejo de grandes cantidades de tráfico gracias a la optimización del CSS y el JavaScript (Moreno, 2019)

A partir de lo antes expuesto se puede decir que Drupal además de un gestor de contenidos es un *framework* que otorga todas las herramientas necesarias para desarrollar cualquier tipo de aplicaciones web personalizadas con funcionalidades muy específicas y adaptadas a todas las necesidades de tu negocio (Moreno, 2019).

Drupal 8 tiene características novedosas con respecto a las versiones anteriores como son:

- Garantiza que el sistema sea modular y escalable.
- Rendimiento mejorado en el *front-end*.
- Accesibilidad mejorada.
- Basado totalmente en Symfony 2 y sus componentes.
- Compatibilidad total con PHP7, PostgreSQL y bases de datos SQLite.
- Integra librerías populares como composer, symfony 3, Guzzle, y Twig en su núcleo (core).
- CSS3 y HTML5 soportado de forma nativa.
- Adaptación a cualquier dispositivo usando salidas HTML5, diseño adaptativo y con filosofía Mobile-first.

Para el desarrollo de la propuesta de solución se utilizó este marco de trabajo debido a que el usado la solución anterior fue desarrollada en una versión precedente.

HTML5

El Lenguaje de Marcado para Hipertextos, es usado para describir y definir el contenido de una página web en un formato bien estructurado. En sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto e imágenes. Permite escribir texto de forma estructurada y agradable. Sus archivos son pequeños, permite un despliegue rápido, es fácil de aprender y lo admiten la mayoría de los navegadores. Su nivel de complejidad es bajo por lo que no necesita de grandes conocimientos cuando se cuenta con un editor de páginas web (De Luca, 2011). Se utilizó para la estructura de las vistas de la solución propuesta.

CSS3

Hojas de Estilo en Cascada (CSS, por sus siglas en inglés), hace referencia a un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML. Su objetivo es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. Las Hojas de Estilo en Cascada en su versión 3 están divididas en varios documentos separados, llamados "módulos". Donde cada módulo añade nuevas funcionalidades a las definidas en CSS 2, de manera que se preservan las anteriores para mantener la compatibilidad. Son un mecanismo simple para agregar estilo (por ejemplo, fuentes, colores, espacios) a los documentos web (Bert Bos, 2019). Se utilizó en el desarrollo de la solución para los estilos y el formato del sistema.

PHP 7.2

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que es posible realizar varias acciones como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies, entre otras.

El mismo puede emplearse en todos los sistemas operativos y admite la mayoría de servidores web de la actualidad, por ejemplo, Apache. Esto incluye cualquier servidor web que pueda utilizar el binario de PHP como nginx. Funciona tanto como módulo que como procesador de CGI (*Common Gateway Interface*, por sus siglas en inglés). Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

Una de las características más potentes y destacables es su soporte para un amplio abanico de bases de datos. PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos (PHP, 2019). Para el desarrollo del mantenimiento perfectivo de los Directorios del portal Intranet se utilizó php como lenguaje de programación por sus ventajas y facilidades anteriormente descritas.

Java Script 3.4.1

JavaScript, anteriormente LiveScript, es un lenguaje de programación creado por Netscape Communications, creadores del explorador web Navigator. Se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación (Flanagan, 2006). Se utilizó en el desarrollo de la solución para las validaciones de los eventos existentes en ella.

UML 2.4

UML es un lenguaje de representación visual que permite combinar diversos elementos gráficos y crear diagramas. Se usa para modelar sistemas y usa tecnología orientada a objetos. El lenguaje unificado de modelado describe lo que hará un sistema, pero no dice cómo implementarlo. Su objetivo es visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo. Involucra todo el ciclo de vida del proyecto y está pensado para varios lenguajes y plataformas. Es aplicable en el desarrollo de *software* entregando gran variedad de formas para dar soporte a una metodología de desarrollo de *software*, pero no especifica en sí mismo qué metodología o proceso utilizar (Object Management Group, 2007).

Visual Paradigm 8.0

Visual Paradigm, versión 8.0, es una herramienta multiplataforma de diseño y gestión fácil de usar para sistemas de tecnología informática. Es una herramienta CASE (*Computer Aided Software Engineering*, por sus siglas en inglés) aplicable en todo el ciclo de vida del desarrollo de *software*. Soporta UML (*Unified Modeling Language*, por sus siglas en inglés), BPMN (*Business Process*

Modeling Notation, por sus siglas en inglés), entre otras tecnologías. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. La herramienta proporciona abundantes tutoriales UML, demostraciones interactivas de UML y proyectos UML. Es fácil de instalar y actualizar y compatible entre ediciones (Visual Paradigm, 2019). Para el desarrollo de la solución se utilizó con el objetivo de modelar el diagrama de despliegue del sistema.

PostgreSQL 10.3

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que usa y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas. Ofrece muchas características modernas como son consultas complejas, llaves extranjeras, disparadores, vistas actualizables, integridad transaccional, control de concurrencia multiversión. Además, puede ser extendido por el usuario de muchas maneras, por ejemplo, agregando nuevos tipos de datos, funciones, operadores, funciones agregadas, métodos de índice, lenguajes de procedimiento. Debido a la licencia liberal, PostgreSQL puede ser utilizado, modificado y distribuido por cualquier persona de forma gratuita para cualquier propósito, ya sea privado, comercial o académico.

PostgreSQL se ha ganado una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto de características robustas, extensibilidad y la dedicación de la comunidad de código abierto detrás del *software* para ofrecer soluciones innovadoras y de alto rendimiento (PostgreSQL, 2019). Se utilizó con el objetivo de almacenar la información con la que trabaja el mantenimiento perfectivo de la solución.

PgAdmin III

Es una herramienta de código abierto para la administración de bases de datos PostgreSQL. Fue diseñado para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris. PgAdmin es desarrollado por una comunidad de expertos PostgreSQL en todo el mundo y está disponible en más de una docena de idiomas (PgAdmin, 2019). Para el desarrollo de la solución se utilizó con el objetivo de trabajar con los objetos que están almacenados en la base de datos, examinar sus propiedades y realizar tareas administrativas.

Elasticsearch 2

Es un motor de analítica y análisis distribuido y *open source* para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y desestructurados. Está desarrollado en Apache Lucene y fue presentado por primera vez en 2010 por Elasticsearch N.V. (ahora conocido como Elastic). Conocido por sus API REST simples, naturaleza distribuida, velocidad y escalabilidad.

La velocidad y escalabilidad de Elasticsearch y su capacidad de indexar muchos tipos de contenido significan que puede usarse para una variedad de casos de uso:

- Búsquedas de aplicaciones, sitios web y otros elementos
- Monitoreo de rendimiento de aplicaciones
- Analítica de Seguridad

Además, de su velocidad, la escalabilidad y la resistencia, Elasticsearch tiene una cantidad de características integradas poderosas que contribuyen a que el almacenamiento y la búsqueda de datos sean incluso más eficientes, como *data rollup* y gestión de ciclo de vida del índice. Algunas de las características son (Elastic, 2019):

- Es rápido ya que es una plataforma de búsqueda en casi tiempo real, lo que implica que la latencia entre el momento en que se indexa un documento hasta el momento en que se puede buscar en él es muy breve: típicamente, un segundo.
- Distribuido por naturaleza por lo que le permite escalar horizontalmente a cientos (o incluso miles) de servidores y gestionar petabytes de datos.
- Soporta una variedad de lenguajes como son: Java, JavaScript (Node.js), .NET (C#), PHP, Python, entre otros.

Se utilizó en el desarrollo de la solución propuesta para el acceso en tiempo real y de forma instantánea a los datos de las personas, teléfonos, cumpleaños y aplicaciones.

Apache2

El servidor HTTP (*Hypertext Transfer Protocol*, por sus siglas en inglés) Apache, es un servidor web HTTP de código abierto y multiplataforma que implementa el protocolo HTTP y la noción del sitio virtual según la normativa RFC 2616. Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido. Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Además, es un servidor web multi-

plataforma, y extensible. Cuenta con el mayor número de opciones de configuración de entre todos los servidores y es capaz de adaptarse a la mayoría de las situaciones y es extensible por medio de módulos. Es un servidor web modular, existen módulos de todo tipo, que permiten para extender el soporte a múltiples funcionalidades y tecnologías, además de lenguajes, se encuentra la posibilidad de usar SSL, la reescritura de URL, módulos para mejorar la seguridad (Apache, 2019). Utilizado como servidor web para visualizar el proceso de desarrollo del mantenimiento perfectivo a los Directorios del portal Intranet.

PhpStorm 2019.2.1

PhpStorm es ideal para trabajar con Drupal, Symfony, Laravel, WordPress, Joomla, y otros marcos de trabajo. Incluye tecnologías de front-end, tales como HTML 5, CSS, JavaScript entre otras, con disponibilidad de refactorizaciones, depuración y pruebas de unidad. Además, tiene herramientas de desarrollo integradas tales como: el sistema de control de versiones, bases de datos (SQL, PostgreSQL y otras), herramientas de línea de comando, Composer, REST Client y muchas otras herramientas. También tiene una compatibilidad completa con PHP. PhpStorm es reconocido por su Depurador visual libre de configuración, lo que le permite obtener un conocimiento extraordinario sobre lo que sucede dentro de su aplicación a cada paso (PhpStorm, 2019). Se utilizó para la realización de todas las tareas asociadas a la programación durante el desarrollo del mantenimiento perfectivo de los Directorios del portal Intranet.

jQuery 3.4.1

jQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Hace que cosas como el desplazamiento y la manipulación de documentos HTML, el manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript (js.foundation, 2019). Para el desarrollo de la solución permitió hacer uso de funciones JavaScript de una manera sencilla.

Bootstrap 4

Bootstrap 4 es la última versión de Bootstrap, el *framework* de CSS, HTML y JavaScript más popular, que nos permite desarrollar webs que se ajustan a cualquier resolución y dispositivo. En esta última versión, el *framework* se ha reescrito por completo, con nuevos componentes, estilos más

optimizados y aún más adaptabilidad (Thornton, Otto y Bootstrap, 2018). Fue utilizada para los estilos de los elementos de las vistas.

Twig

Es un motor y lenguaje de plantillas para PHP que es rápido, eficiente y seguro. La sintaxis de *Twig* se ha diseñado para que las plantillas sean concisas y muy fáciles de leer y de escribir, por lo que Symfony recomienda utilizar *Twig* para la creación de las mismas. El objetivo de *Twig* es conseguir que los diseñadores sin conocimientos de programación sean capaces de crear todas las plantillas de la aplicación de forma autónoma, sin ayuda de los programadores. De esta forma se acelera el desarrollo de las aplicaciones y se mejora la productividad. Por eso *Twig* ha sido ideado para que sea realmente fácil de aprender, leer y escribir por parte de profesionales sin un perfil técnico avanzado (Symfony y Twig, 2020). Fue utilizado con el objetivo de modelar las plantillas que serán usadas del lado del cliente en el desarrollo del mantenimiento perfectivo de los directorios del portal web Intranet.

GitLab 2.8

GitLab es una plataforma de desarrollo de *software* totalmente integrada que permite a su equipo ser transparente, rápido, efectivo y coherente desde el debate sobre una nueva idea hasta la producción, todo en la misma plataforma. GitLab ayuda a los equipos a mejorar el tiempo de ciclo de semanas a minutos, reducir los costos de desarrollo y el tiempo de comercialización al tiempo que aumenta la productividad del desarrollador (GitLab, 2020). Para el desarrollo de la solución se utilizó GitLab para el control de versiones por sus facilidades anteriormente descritas.

Apache JMeter 3.0

La aplicación Apache JMeter en su versión 3.0 es un *software* de código abierto, una aplicación Java 100%, diseñada para cargar el comportamiento funcional de las pruebas y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba. Esta herramienta se puede utilizar para probar el rendimiento tanto en recursos estáticos como dinámicos, aplicaciones web dinámicas. Además, se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su resistencia o analizar el rendimiento general bajo diferentes tipos de carga (Foundation, 2020). Fue utilizada como herramienta automatizada para realizar las pruebas de rendimiento de la solución.

1.4 Conclusiones parciales.

Luego del análisis realizado en el capítulo se concluye que:

- Se definieron los conceptos relacionados con el dominio del problema contribuyendo a una mayor comprensión del entorno del negocio.
- Después de realizar una valoración acerca de algunos sistemas homólogos en el ámbito nacional e internacional, permitió sentar las bases para el desarrollo del mantenimiento perfectivo de los directorios del portal web Intranet.
- Se determinaron las herramientas, tecnologías y metodología que se utilizarán para el desarrollo del mantenimiento perfectivo de los directorios del portal web Intranet, formando las bases propicias para la creación de una propuesta de solución, que cumpla con el objetivo de la investigación.

Capítulo 2: Propuesta de solución de la Investigación.

El presente capítulo tiene como objetivo describir la propuesta de solución del sistema. A partir del flujo descrito se extraen las características con las que contará el módulo a desarrollar y se modela su proceso de negocio, se listan los requerimientos funcionales y no funcionales. También se muestra el diagrama de despliegue, se describe el estilo arquitectónico y los patrones de diseño utilizados en los Directorios del portal web intranet de la universidad.

2.1 Descripción del sistema propuesto

La descripción de cualquier producto o solución informática es una idea, modelo o visión del *software* que se quiera implementar y esto permite que el cliente y el equipo de trabajo puedan imaginar cómo quedará el producto y la utilidad del mismo.

El mantenimiento perfectivo de los directorios del portal intranet tiene como objetivo realizar búsquedas rápidas y completas atendiendo a los diferentes criterios de búsquedas especificados, que permitan facilitar la comunicación e interacción de los usuarios. Además, del desarrollo de funcionalidades para efectuar búsquedas simples y avanzadas sobre la información de personas, teléfonos, aplicaciones y cumpleaños en la universidad.

Desde cada directorio se podrán gestionar cada uno de sus datos permitiendo crear sus listas personales añadiéndolos a favoritos. Por su parte, el Directorio de personas permitirá notificar vía correo electrónico y a través, de la página mediante notificaciones flotantes de algún cambio en los datos de los usuarios agregados a la lista de favoritos.

Esta solución estará implementada sobre una nueva versión (Drupal 8) del marco de trabajo del sistema anterior, debido a que esta nueva versión garantiza que el sistema sea modular y escalable, mejoras en el rendimiento del front-end, está basado totalmente en symfony 3y sus componentes, es adaptativo a cualquier dispositivo usando salidas HTML5 y la filosofía de Mobile-first.

Para una mejor comprensión de la propuesta de solución a continuación se ofrecen los artefactos que ayudan a entender de una forma técnica el desarrollo de la solución.

2.2 Roles del proyecto.

La metodología Scrum define roles a cada persona que interactúa en el proceso de desarrollo del producto. En la tabla 2 se muestran los roles del proyecto.

Tabla 2: Roles del proyecto

Roles del proyecto	
Propietario del producto	Universidad de las Ciencias Informáticas

Equipo de desarrollo	Beatriz Pérez Delgado
Scrum Master	Ing. Jany Coto García
Interesados	Dirección de Informatización Personas con acceso a la red de la universidad

2.3 Especificación de los requisitos del sistema

Es de vital importancia documentar lo que debe hacer el sistema una vez creado. Para definir esto se especifican los requisitos los cuales describen qué es lo que el sistema debe hacer (sus funciones) y sus propiedades esenciales y deseables (Sommerville, 2011). Los requisitos del *software* se clasifican en dos grupos: requisitos funcionales y requisitos no funcionales.

2.3.1 Requisitos funcionales del sistema

Son capacidades o condiciones que el sistema debe cumplir, de cómo debería reaccionar el mismo a entradas particulares y de cómo debería comportarse en situaciones específicas. Además, surgen de la razón fundamental de la existencia de un producto (Sommerville, 2011). En la siguiente tabla se especifican los requisitos funcionales del sistema.

Requisitos funcionales de los Directorios del portal Intranet

Tabla 3: *Requisitos funcionales del sistema*

No.	Nombre del requisito	Complejidad	Prioridad
Directorio de personas			
RF1	Realizar búsqueda avanzada de persona (s) <ul style="list-style-type: none"> ✓ Nombre ✓ Apellido ✓ Usuario ✓ Solapín ✓ Apartamento ✓ Categoría persona ✓ Categoría docente ✓ Rol universitario ✓ Cargo (<i>si no es estudiante</i>) 	Alta	Alta

	<ul style="list-style-type: none"> ✓ Área ✓ País ✓ Provincia ✓ Municipio ✓ Género ✓ Facultad ✓ Grupo (<i>si es estudiante</i>) ✓ Grado académico ✓ Unidad organizativa ✓ Edificio ✓ Teléfono ✓ Carrera ✓ Fundador UCI 		
RF2	Mostrar datos de la (s) persona (s) buscada (s)	Baja	Alta
RF3	Permitir enviar correo electrónico a las personas que poseen cuenta de correo electrónico	Media	Media
Directorio de personas con usuario autenticado			
RF4	Adicionar persona a la lista de favoritos del usuario	Alta	Alta
RF5	Eliminar persona (s) de la lista de favoritos del usuario	Media	Alta
RF6	Mostrar personas de la lista de favoritos del usuario.	Baja	Media
RF7	Notificar actualización de los datos de los usuarios agregados en la lista de favoritos del usuario.	Alta	Media
Directorio de teléfonos			
RF8	Realizar búsqueda avanzada de teléfono (s) <ul style="list-style-type: none"> ✓ Teléfono ✓ Área 	Alta	Alta
RF9	Mostrar datos de la búsqueda de teléfonos	Baja	Alta
RF10	Mostrar teléfonos de las páginas amarillas. <ul style="list-style-type: none"> ✓ Centros de reportes 	Media	Alta

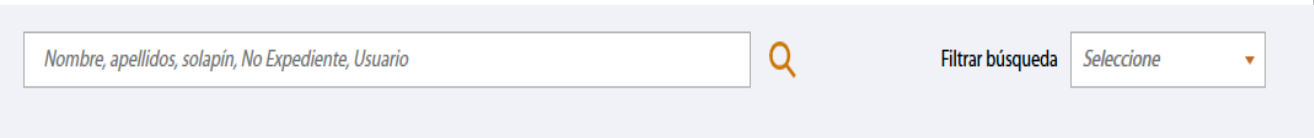
	<ul style="list-style-type: none"> ✓ Organizaciones ✓ Comercio y gastronomía ✓ Comunicaciones ✓ Docencia ✓ Servicios generales ✓ Información ✓ Salud ✓ Tecnología ✓ Transporte 		
RF11	Mostrar teléfonos de urgencias <ul style="list-style-type: none"> ✓ Puestos de Mandos ✓ Transporte ✓ Cuerpo de guardia del hospital ✓ Información 	Media	Alta
Directorio de teléfonos con usuario autenticado			
RF12	Adicionar teléfono a la lista de teléfonos favoritos.	Media	Media
RF13	Eliminar teléfono(s) de la lista de teléfonos favoritos.	Baja	Media
RF14	Mostrar teléfonos de la lista de teléfonos favoritos.	Media	Media
Directorio de cumpleaños			
RF15	Realizar búsquedas avanzadas de cumpleaños <ul style="list-style-type: none"> ✓ Nombre ✓ Apellidos ✓ Usuario ✓ Solapín 	Alta	Alta
RF16	Realizar búsqueda estática por inicial del nombre	Media	Alta
RF17	Mostrar datos de la búsqueda de cumpleaños	Media	Alta
RF18	Mostrar cumpleaños del día	Baja	Alta

RF19	Mostrar calendario para realizar búsqueda por fechas	Media	Alta
Directorio de cumpleaños con usuario autenticado			
RF20	Mostrar opción de recordar cumpleaños	Media	Media
RF21	Adicionar cumpleaños recordados	Baja	Media
RF22	Eliminar cumpleaños recordados	Baja	Media
RF23	Mostrar cumpleaños recordados	Media	Media
RF24	Mostrar configuración de cumpleaños <ul style="list-style-type: none"> ✓ Cantidad de días con antelación de la notificación (vía de notificación (correo electrónico, propio directorio, ambas vías)) 	Media	Baja
Directorio de aplicaciones			
RF25	Mostrar aplicaciones	Baja	Alta
RF26	Realizar búsquedas simples haciendo uso de los criterios: nombre o tipo de visibilidad.	Alta	Alta
RF27	Agrupar por esferas <ul style="list-style-type: none"> ✓ Facultades ✓ Docencia ✓ Comunidades ✓ Investigación ✓ Postgrado ✓ Producción ✓ Comunicación e información ✓ Gestión administrativa ✓ Servicios al ciudadano ✓ Servicios telemáticos ✓ Soporte técnico ✓ Organizaciones 	Media	Alta
Directorio de aplicaciones con usuario autenticado			

RF28	Adicionar aplicaciones	Media	Media
RF29	Eliminar aplicaciones	Baja	Media
RF30	Mostar aplicaciones	Baja	Media

La siguiente tabla muestra la descripción del requisito funcional Realizar búsqueda avanzada de personas.

Tabla 4: Historia de Usuario del requisito funcional Realizar búsqueda avanzada de personas.

Número:	1	Nombre del requisito: Realizar búsqueda avanzada de personas
Programador: Beatriz Pérez Delgado	Iteración Asignada: 1	
Prioridad: Alta	Tiempo Estimado: 1	
Riesgo en Desarrollo: N/A	Tiempo Real: 1	
Descripción:		
<p>Se muestra el campo <i>Buscar</i> por los criterios <i>Nombre, Apellidos, Solapín, No. Expediente, Usuario</i>. Las <i>Opciones de búsquedas</i> que contiene con el campo de filtrar búsqueda son: <i>Nombre, Apellido, Usuario, Solapín, Apartamento, Categoría persona, Categoría docente, Rol universitario, Cargo, Área, País, Provincia, Municipio, Género, Facultad, Grupo, Grado académico, Unidad organizativa, Edificio, Teléfono, Carrera, Fundador UCI</i>.</p>		
Observaciones:		
Prototipo elemental de interfaz gráfica de usuario:		
		
Página 1 1		

2.3.2 Requisitos no funcionales del sistema

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares (Sommerville, 2011). Son propiedades o cualidades que el producto debe tener. A continuación, se describen los requisitos no funcionales del sistema.

Rendimiento

- ✓ El tiempo de respuesta debe ser menor de 7 milisegundos.
- ✓ El sistema deberá soportar una conexión simultánea de al menos 1000 usuarios.

Portabilidad

- ✓ El sistema debe ser compatible con los sistemas operativos Linux, Ubuntu.

Despliegue:

- ✓ Servidor Web: Apache2
- ✓ Servidor Correo: Zimbra 8.7 y superiores
- ✓ Base de Datos: PostgreSQL 10
- ✓ Sistema gestor de contenido: Drupal 8.8.6 o superior.
- ✓ Lenguaje de programación: PHP 7.2 o superior.

Usabilidad

- ✓ El sistema podrá ser usado por personas con acceso a la red de la universidad.
- ✓ Se utilizará colores y letras apropiadas para las personas con discapacidades visuales.
- ✓ En el sistema podrán trabajar, usuarios con el mínimo de conocimiento en informática.

Apariencia o interfaz externa:

- ✓ La interfaz dispondrá de menús desplegados para agilizar y acelerar su utilización.

Hardware

- ✓ Para el cliente:
 - Se puede usar cualquier dispositivo que soporte un navegador y esté conectado a la red de la universidad.
- ✓ Para el servidor:
 - Memoria RAM: mínimo 4GB

- Disco duro: mínimo 500GB
- Se requiere una tarjeta de red

Seguridad

- ✓ Restringirá el acceso a sus funcionalidades de acuerdo a los roles asignados a los actores del sistema.
- ✓ La integridad de la información será garantizada mediante un mecanismo de control de acceso que permitirá en correspondencia con el usuario autenticado realizar las operaciones sobre el sistema permisibles al rol que ocupa dicho usuario.

2.4 Patrón Arquitectónico

Un patrón arquitectónico se puede considerar como una descripción abstracta estilizada de buena práctica, que se ensayó y puso a prueba en diferentes sistemas y entornos. Debe describir una organización de sistema que ha tenido éxito en sistemas previos. Debe incluir información sobre cuándo es y cuándo no es adecuado usar dicho patrón, así como sobre las fortalezas y debilidades del patrón (Sommerville, 2011). El patrón arquitectónico utilizado para el desarrollo del mantenimiento perfectivo a los Directorios del portal Intranet es el Modelo-Vista-Controlador (MVC).

El MVC es un patrón de arquitectura de *software* que separa presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente Modelo maneja los datos del sistema y las operaciones asociadas a esos datos. El componente Vista define y gestiona cómo se presentan los datos al usuario. El Controlador dirige la interacción del usuario (por ejemplo, teclas oprimidas, clics del mouse, etcétera) y pasa estas interacciones a Vista y Modelo(Sommerville, 2011).

El modelo: Debe representar la parte de la aplicación que implementa la lógica de negocio y el acceso a datos. Esto significa que debe ser responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos.

La vista: Debe hacer una presentación de los datos del modelo estando separada de los objetos del modelo. Es responsable del uso de la información de la cual dispone para producir cualquier interfaz de presentación de cualquier petición que se presente. En el desarrollo del mantenimiento perfectivo de los Directorios del portal Intranet para la presentación de los datos se utiliza *Twig* como motor de

plantillas, la biblioteca jQuery, en conjunto con los archivos CSS y JavaScript, que se encargarán de estructurar y aplicar estilos a las vistas creadas.

El controlador: Gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista. Los controladores pueden ser vistos como administradores, espera peticiones de los clientes, comprueba su validez de acuerdo a las normas de autenticación o autorización, delega la búsqueda de datos al modelo y selecciona el tipo de respuesta más adecuado según las preferencias del cliente. Finalmente delega este proceso de presentación a la capa de la Vista.

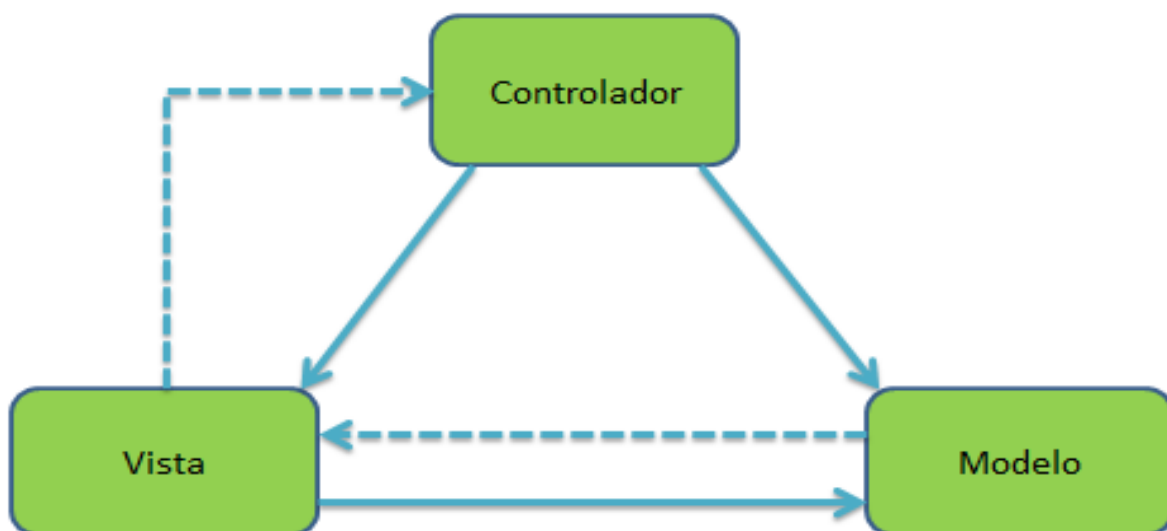


Figura 1: Patrón Arquitectónico MVC (Elaboración propia).

2.5 Patrones de diseño

Un patrón de diseño se caracteriza como “una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución”. Para el diseño de *software*, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución (Pressman y Maxim, 2015).

2.5.1 Patrones GoF

Los patrones "Banda de los Cuatro" (*Gang-of-Four*) describen las formas comunes en que diferentes tipos de objetos, pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases

y la formación de estructuras de mayor complejidad. Además, permiten crear grupos de objetos para ayudar a realizar tareas complejas (Gamma, 2002).

Instancia única (Singleton): es un patrón diseñado para restringir la creación de objetos pertenecientes a una clase u objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

```
11 abstract class ConfigFormBase extends FormBase {
12     use ConfigFormBaseTrait;
13     public function __construct(ConfigFactoryInterface $config_factory) {
14         $this->setConfigFactory($config_factory);
15     }
16 }
```

Figura 2: Ejemplo del patrón Singleton. (Elaboración propia).

Observador (Observer): es un patrón de comportamiento que permite que uno o varios objetos observadores, sean notificados cuando el objeto que están observando cambie de estado. La dependencia es de 1 a muchos, de forma que puede haber varios observadores sobre el objeto observado, conocido como Sujeto (Subject).

```
28 public function buildForm(array $form, FormStateInterface $form_state) {
29     $config = $this->config( name: 'generales.elasticsearchurl');
30     $form['es_persona'] = [
31         '#type' => 'textfield',
32         '#title' => $this->t( string: 'Persona'),
33         '#description' => $this->t( string: 'Url de los datos de "Personas" (Server de Elasticserach)'),
34         '#default_value' => $config->get('es_persona'),
35         '#required' => TRUE,
36         '#size' => 150
37     ];
38 }
39 }
```

Figura 3: Ejemplo de utilizar el patrón Observador. (Elaboración propia).

Adaptador (Adapter): es un patrón de diseño estructural que nos permite reutilizar una o varias clases, integrándolas o embebiéndolas dentro de otra clase que actúa como contenedor. En la clase contenedora podremos definir nuevos métodos y propiedades, que utilicen internamente solo los métodos y propiedades que necesite de las clases originales. Es una forma de no tener que modificar las clases originales, que pueden ser librerías o componentes externos, para integrarlas en un sistema que requiera de unos métodos específicos.

```

1  <?php
2  namespace Drupal\generales\Form;
3
4  use Drupal\Core\Form\ConfigFormBase;
5  use Drupal\Core\Form\FormStateInterface;
6
7  class ElasticsearchUrlForm extends ConfigFormBase{
8
9      /**
10     * {@inheritdoc}
11     */
12     protected function getEditableConfigNames() {
13         return [
14             'generales.elasticsearchurl',
15         ];
16     }

```

Figura 4: Ejemplo de uso del patrón Adapter. (Elaboración propia).

2.6 Diagrama de despliegue

El diagrama de despliegue se utiliza para mostrar la estructura física del sistema, incluyendo las relaciones entre el *hardware* y el *software* que se despliega, estas relaciones son representadas por los protocolos de comunicación que se utilizan para acceder a cada uno. En la siguiente figura puede visualizarse el diagrama de despliegue definido para la solución propuesta:

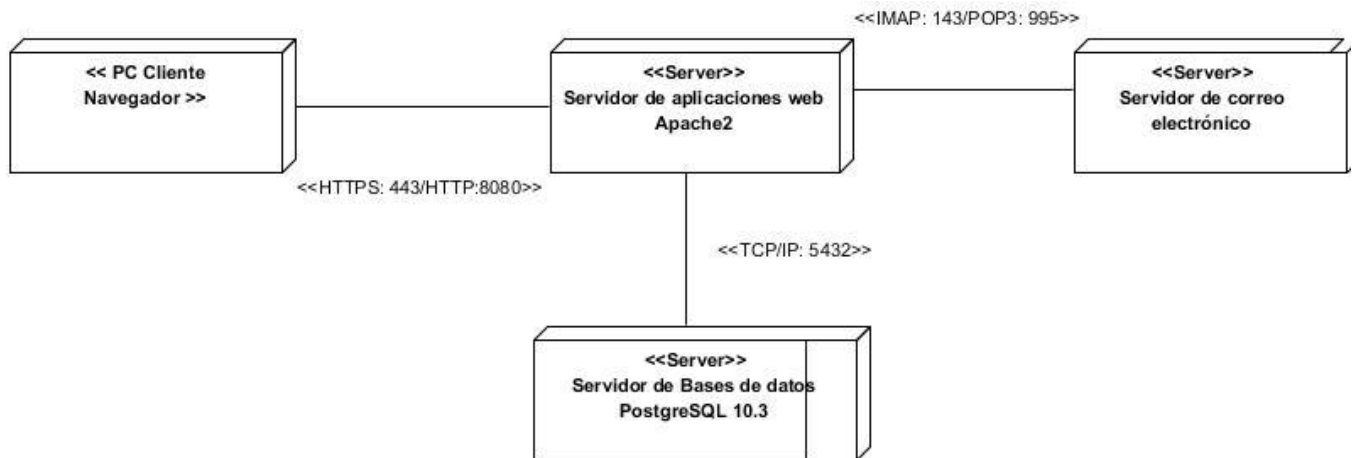


Figura 5: Diagrama de despliegue (Elaboración propia).

El diagrama de despliegue representado muestra la siguiente distribución:

Cliente: dispositivo cliente capaz de conectarse al servidor de aplicaciones mediante el protocolo de comunicaciones HTTP o HTTPS.

Servidor de aplicaciones web: computadora en que se encuentra el servidor web Apache2, este será el lugar en que se gestione todo el contenido de la aplicación. El mismo establecerá comunicación con los ordenadores clientes mediante protocolo HTTPS o HTTP, con el servidor de base de datos por medio del protocolo TCP/IP y con el servidor de correo electrónico mediante los protocolos IMAP o POP3.

Servidor de base de datos: ordenador en que se encuentra el gestor de base de datos PostgreSQL capaz de mantener persistente la información generada y a utilizar.

Servidor de correo electrónico: permite hacer uso del correo electrónico.

HTTP: protocolo de transferencia de hipertexto o HTTP (*HyperText Transfer Protocol*) es el protocolo de red que permite la transferencia de documentos de hipermedia en la red, generalmente entre un navegador y un servidor, para que los humanos puedan leerlos. HTTP se considera textual porque toda la comunicación se realiza en texto plano. También es un protocolo sin estado, ya que una comunicación no es consciente de las comunicaciones que se hayan podido producir anteriormente (Krishnamurthy y Rexford, 2001).

HTTPS: protocolo de transferencia de hipertexto seguro, por sus siglas en inglés, *Hypertext Transfer Secure Protocol* (HTTPS), es un protocolo de red basado en HTTP por lo que está orientado a transacciones sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores y sigue el esquema petición-respuesta entre un cliente y un servidor (Goyal et al., 2020).

TCP/IP: base de Internet y sirve para enlazar computadoras que utilizan diferentes sistemas operativos. Familia de protocolos utilizada para la conexión entre el servidor web y el servidor donde se encuentra ubicada la base de datos (Forouzan, 2017).

IMAP: El protocolo de acceso a mensajes de Internet (en inglés *Internet Message Access Protocol* o IMAP), es un protocolo de aplicación que permite el acceso a mensajes almacenados en un servidor de Internet. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet (Heinlein y Hartleben, 2008).

POP3: Post Office Protocol (POP3, Protocolo de Oficina de Correo o "Protocolo de Oficina Postal") protocolo utilizado en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto, denominado Servidor POP. Es un protocolo de nivel de aplicación en el Modelo OSI (Schmied y Chellis, 2006).

2.7 Conclusiones parciales

Una vez finalizado este capítulo, se arribaron a las siguientes conclusiones:

- ✓ Se determinaron los requisitos funcionales y no funcionales constituyendo una guía fundamental para el desarrollo del mantenimiento perfectivo de los Directorios del portal Intranet de la universidad.
- ✓ Con la utilización de la arquitectura Modelo-Vista-Controlador y el uso de los patrones de diseño GoF instancia única, observador y adaptador se garantizará una mayor organización y código más legible.
- ✓ Con la realización del modelo de despliegue se ilustra la comunicación de los distintos componentes en los cuales se divide el sistema.

Capítulo 3: Implementación y Pruebas de la Investigación.

En el presente capítulo se propone explicar a partir de los resultados del diseño, la implementación del Mantenimiento perfectivo de los directorios del portal web Intranet. Se definen los estándares de codificación y las pruebas de *software* como elemento esencial que garantiza la calidad del mantenimiento y revisión final del cumplimiento de los requisitos, especificaciones de diseño y codificación.

3.1 Estándares de codificación

Los estándares de codificación son reglas de codificación que permiten tener una programación homogénea, comprendiendo todos los aspectos de la generalización del código, pues la aplicación debe de estar implementada como si un único programador escribiera el código de una sola vez. La usabilidad de estos permite conservar el código fuente entendible y fácil de mantener, además de mejorar la forma en la que se programa (Ramos et al. 2017).

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación, es de gran importancia para la calidad del *software*. La aplicación de estándares de codificación además posibilita que el *software* que se obtiene sea fácil de comprender y de mantener en el tiempo. El uso de los mismos tiene ventajas tales como:

- Asegurar la legibilidad del código entre distintos programadores, facilitando la depuración del mismo.
- Proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado.
- Facilitar la portabilidad entre plataformas y aplicaciones.

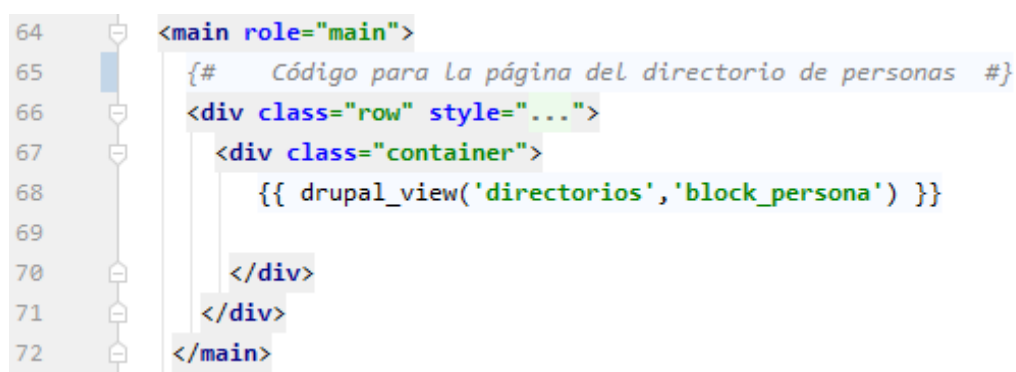
Las convenciones de código o estándares de codificación son importantes para los programadores por las siguientes razones:

- El 80% del coste del código de un programa va a su mantenimiento.
- Casi ningún *software* es mantenido toda su vida por el autor original.
- Las convenciones de código mejoran la lectura del *software* lo que permite entender código nuevo de manera más óptima y rápida.
- Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.

A continuación, se presentan algunos de los estándares de codificación definidos para el mantenimiento perfectivo de los directorios del portal web Intranet(Drupal, 2020):

Uso de comillas

Drupal no tiene un estándar fuerte para el uso de comillas simples vs. comillas dobles. Mientras sea posible, se debe mantener la consistencia en cada módulo, y respetar el estilo personal de otros desarrolladores. Con este aviso en mente: las cadenas de comillas simples son conocidas por ser más rápidas, debido a que el analizador o intérprete no tiene que buscar en las variables “en línea”.



```
64 <main role="main">
65   {# Código para la página del directorio de personas #}
66   <div class="row" style="...">
67     <div class="container">
68       {{ drupal_view('directorios', 'block_persona') }}
69     </div>
70   </div>
71 </main>
```

Figura 6: Ejemplo del uso de las comillas en el portal web Intranet. (Elaboración propia).

Su uso se recomienda, a excepción de dos casos.

1. Uso en variables “en línea”, Ejemplo:<h2>\$title</h2>
2. Cadenas traducidas donde se puede evitar el escape de comillas simples encerrando la cadena en comillas dobles. Una de estas cadenas podría ser: "He's a good person." Ésta sería " He\’s a good person." con comillas simples. Tal escape no puede ser manejado adecuadamente por los generadores de archivos.pot para la traducción de texto, y además es un poco difícil de leer.

Utilización del punto y coma (;)

El lenguaje PHP requiere puntos y comas al final de la mayoría de las líneas, pero permite ser omitidos al final de bloques de código. Los estándares de programación de Drupal los requieren, incluso al final de bloques de código. En particular para una línea de bloques PHP:

```

82 function obtenerPersonaDadoID($id_persona){
83 // $url_data = variable_get('persona_url_config');
84 $url_data = variable_get( name: 'directorio_url_config');
85 $sterminos = '';
86 $sterminos .= '{ "term" : {"baja" : false}} ';
87 $sterminos .= ', {"term":{"persona_id.keyword":"" . $id_persona . ""}}';
88 //se arma la variable dinamica con la cadena a filtrar para la consulta
89 $variables = '';
90 if ($variables=='')
91 | $variables = '*';
92 $result = general_query($url_data,$size=1,$from=0,$sterminos,$variables);
93 //decodificar la respuesta json a un array
94 $datos_personales = json_decode($result);
95 return $datos_personales;
96 }

```

Figura 7: Ejemplo del uso del punto y coma en el portal web Intranet. (Elaboración propia)

Operadores

Todos los operadores binarios (operadores que están entre dos valores), como +, -, +=, =, ==,> deben tener un espacio antes y después del operador, para facilitar la lectura. Por ejemplo, una asignación debe tener el formato (foo = \$bar;) en vez de (\$foo=\$bar;). Los operadores unarios (operadores que operan sobre un solo valor), como ++, no deben tener espacios entre el operador y la variable o número que se utiliza.

```

46 {% set view_d_persona = drupal_view_result('directorios','block_persona') %}

```

Figura 8: Ejemplo del uso de operadores en el portal web Intranet. (Elaboración propia).

Estructuras de control

Las estructuras de control incluyen: **if, for, while,switch**. Aquí se presenta un ejemplo de la sentencia **if**:

```

74 {% if page.footer %}
75 <footer role="contentinfo">
76   {{ page.footer }}
77 </footer>
78 {% endif %}

```

Figura 9: Ejemplo del uso de estructuras de control en el portal web Intranet (Elaboración propia).

Las sentencias de control deben tener un espacio entre la palabra clave (**keyword**) y el paréntesis de apertura, para distinguirlas de las llamadas de función. Se le recomienda siempre utilizar paréntesis de

llave, incluso en situaciones en que son técnicamente opcionales. Tenerlos incrementa la legibilidad y disminuye la probabilidad de introducir errores lógicos cuando nuevas líneas son agregadas.

Llamados de función

Las funciones deben ser llamadas sin espacios entre el nombre de la función, el paréntesis de apertura, y el primer parámetro; con espacios entre las comas y cada parámetro, y sin espacios entre el último parámetro, el paréntesis de cierre, y el punto y coma. Por ejemplo:

```
68 | | | {{ drupal_view('directorios', 'block_persona') }}
```

Figura 10: Ejemplo del uso de llamados de función en el portal web Intranet (Elaboración propia).

Como se muestra en la figura 8, debe haber un espacio en cada lado del signo igual usado para asignar el valor de retorno de una función a una variable. En el caso de un bloque de asignaciones relacionadas, se pueden insertar más espacios para mejorar la legibilidad.

Etiquetas de código PHP

Siempre se debe utilizar `<?php?>` para delimitar el código PHP, no de la forma reducida `?>`. Esto es requerido para la conformidad (aceptación) de Drupal y es también la manera más portable de incluir código PHP en diferentes sistemas operativos y configuraciones.

Funciones y variables

Las funciones y variables deben ser nombradas usando minúsculas, y las palabras deben ser separadas utilizando un guion bajo. Las funciones deben tener además el nombre del grupo/módulo como prefijo, para evitar el conflicto de nombres entre los módulos.

Variables persistentes

Las variables persistentes (variables/configuraciones definidas usando las funciones de Drupal `variable_get()`/ `variable_set()`) deben ser nombradas usando minúsculas y las palabras deben ser separadas utilizando un guion bajo (`_`). Deben usar el nombre del grupo/módulo como prefijo, para evitar el conflicto de nombres entre los módulos.

Constantes

Las constantes deben estar siempre en mayúsculas, con guiones bajos para separar las palabras. Esto incluye a las constantes PHP predefinidas como `TRUE`, `FALSE` y `NULL`. Los nombres de las constantes definidas en los módulos deben también tener una mayúscula como prefijo del módulo por el cual son definidas.

Nombre de archivos

Todos los archivos de documentación deben tener en el nombre la extensión “.txt” para hacer más fácil su lectura en sistemas Windows. Además, los nombres para este tipo de archivos deben ser todo en mayúsculas. Ejemplos: README.txt, INSTALL.txt, TODO.txt.

Incluyendo código

Donde sea que esté incondicionalmente incluyendo un archivo de clase, se debe utilizar: `require_once()`, al igual que incluyendo un archivo de clase (por ejemplo, métodos de fábrica “*factorymethod*”), utilice `include_once()`. Cualquiera de éstas, asegurará que los archivos de clase sean incluidos una sola vez. Ellas comparten la misma lista de archivos, por lo cual no necesita preocuparse por mezclarlos: Un archivo incluido con `require_once()` no será incluido de nuevo con `include_once()`.

Arreglos

Los arreglos deben tener en su formato un espacio separando cada elemento (después de la coma), y espacios alrededor del operador “=>” (asociación de llave):

```
10 return array(  
11     '#markup' => t( string: 'Hola mundo'),  
12 );
```

Figura 11: Declaración de arreglos 1 (Elaboración propia).

Si la línea que declara un arreglo se extiende por más de 80 caracteres (a menudo en el caso de la declaración de formularios y menús), cada elemento debe dividirse en su propia línea, y tener una sangría de un nivel:

```
28 public function buildForm(array $form, FormStateInterface $form_state) {  
29     $config = $this->config( name: 'generales.elasticsearchurl');  
30     $form['es_persona'] = [  
31         '#type' => 'textfield',  
32         '#title' => $this->t( string: 'Persona'),  
33         '#description' => $this->t( string: 'Url de los datos de "Personas" (Server de Elasticserach)'),  
34         '#default_value' => $config->get('es_persona'),  
35         '#required' => TRUE,  
36         '#size' => 150  
37     ];
```

Figura 12: Declaración de arreglos 2 (Elaboración propia).

Se debe tener en cuenta el uso de la coma en el final del último elemento del arreglo. El signo !no constituye un error, sino que ayuda a prevenir errores de análisis sintáctico, si otro elemento es colocado al final de la lista más tarde.

3.2 Validación de requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proyecto y el producto (Ramos et al. 2017).

Para validar los requisitos del sistema según Pressman, se pueden chequear mediante un cuestionario guiado por un conjunto de interrogantes, con el objetivo de descubrir la mayor cantidad de errores posibles (Pressman, 2009).

3.2.1 Criterios para validar los requisitos

A continuación, se muestran las preguntas utilizadas, algunas fueron agregadas por la autora de la presente investigación a las planteadas por Pressman por ser consideradas necesarias para la validación.

Interrogantes para la validación de requisitos:

¿Está el requisito claramente definido?

¿Puede interpretarse mal?

¿Está identificado el origen del requisito (por ejemplo: persona, apellido, solapín)?

¿El planteamiento final del requisito ha sido contrastado con la fuente original?

¿El requisito está delimitado en términos cuantitativos?

¿Qué otros requisitos hacen referencia al requisito estudiado?

¿El requisito incumple alguna restricción definida?

¿El requisito es verificable? Si es así, ¿se pueden efectuar pruebas para verificar el requisito?

¿Se puede seguir el requisito en el modelo del sistema que se ha desarrollado?

¿Está el requisito asociado con los rendimientos del sistema o con su comportamiento?

- ¿El requisito está implícitamente definido?
- ¿El requisito es modificable?
- ¿El requisito está completo?
- ¿El requisito puede ser implementado?
- ¿El requisito puede ser probado?
- ¿El resultado de la evaluación de impacto es positivo?

Resultado de aplicar los criterios de validación

Luego de aplicar el conjunto de interrogantes para validar los requisitos definidos para el desarrollo del

sistema, se obtuvo el 100 % de aprobación por parte del cliente.

3.2.2 Técnicas de validación de requisitos

Con el objetivo de obtener una mayor calidad y demostrar que los requisitos definidos realmente describen el sistema que el cliente necesita; se utilizó la técnica para la validación de requisitos siguiente:

Generación de casos de prueba: se realizaron los diseños de casos de pruebas para cada uno de los requisitos obtenidos, permitiendo verificar que todos se pudieran probar y determinar en la medida de la complejidad del diseño de caso de prueba, identificando requisitos que deberían ser reconsiderados y cuáles pueden ser los más difíciles de implementar.

Resultado de aplicar la técnica de validación

Como resultado de este proceso se identificaron inconsistencias en las especificaciones tales como la falta de concordancia entre la complejidad de la especificación y la registrada en el documento de Evaluación de requisitos. Descripciones de requisitos poco detalladas o ambiguas y numerosos errores ortográficos.

3.3 Pruebas

Para evaluar la calidad del sistema que se está desarrollando y verificar el cumplimiento de los objetivos trazados, se aplicaron un conjunto de pruebas definidas por Pressman en su libro de Ingeniería del *software* “Un enfoque práctico”, en su séptima edición.

Tabla 5: Estrategia de pruebas

Pruebas	Métodos	Técnicas
Prueba de unidad	Caja blanca	Prueba de Camino básico
Pruebas de integración	Caja negra	Incremental
Pruebas de sistema: pruebas de rendimiento	Caja negra	Automatizada(Apache JMeter)
Pruebas de aceptación	Caja negra	Alfa

3.3.1 Pruebas de unidad


Las pruebas de unidad tienen como objetivo verificar la unidad más pequeña del diseño del *software*. Estas pruebas se concentran en la lógica del procesamiento interno y en las estructuras de datos tales como: código fuente, archivos binarios, archivos de datos, entre otros. Este tipo de prueba se puede aplicar en paralelo a varios componentes (Pressman y Maxim, 2015).

Las pruebas de unidad se realizan mediante el método de caja blanca o estructural, denominada a veces prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. La técnica de prueba de caja blanca utilizada en la investigación es el camino básico, que permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución(Sommerville, 2011).

Complejidad ciclomática

La Complejidad Ciclomática se realiza a todos los métodos o algoritmos de un sistema informático. A continuación

Tabla 6: Prueba de unidad (Elaboración propia).

Fórmula 1	Fórmula 2	Fórmula 3	Grafo resultante
$V(G) = (A - N) + 2$ $V(G) = 6 - 6 + 2$ $V(G) = 2$	$V(G) = P + 1$ $V(G) = 1 + 1 = 2$	$V(G) = R$ $V(G) = 2$	
<p>A: es la cantidad de aristas.</p> <p>N: la cantidad de nodos.</p> <p>P: es el número de nodos predicado contenidos en el grafo de flujo G</p> <p>R: representa la cantidad de regiones en el grafo</p>			
<p>Complejidad ciclomática</p>			
<p>Como se puede observar, después de aplicadas las fórmulas 1, 2 y 3 a la funcionalidad obtenerPersonaDadoUsuario, posee una complejidad ciclomática igual a 2, lo cual demuestra que el grafo está bien diseñado y que existen 2 caminos lógicos e independientes.</p>			
<p>Prueba de camino básico</p>			
<p>Una vez calculada la complejidad ciclomática se define como límite superior 2, lo que indica que hay que realizarle al código 2 pruebas, para garantizar que este se ejecute completamente al menos una vez. La funcionalidad obtenerPersonaDadoUsuario posee poco riesgo debido a que el resultado arrojado por la métrica pertenece al intervalo entre 1 -10.</p> <p>El total de caminos independientes establecidos fue de 2 y a continuación se muestran:</p> <ul style="list-style-type: none"> ✓ Camino 1: 1, 2, 3, 4, 5, 6. <p>El primer camino comienza con la inicialización de las variables, seguidamente verifica si la variable es válida, si se cumple esta condición se crea una nueva variable para guardar la</p>			

información, luego se guarda la información con un arreglo y se crea una nueva variable para obtener los resultados y se retorna esta variable.

✓ Camino 2: 1, 2, 4, 5, 6.

El segundo camino comienza con la inicialización de las variables, seguidamente verifica si la variable es válida, si no se cumple esta condición se crea una nueva variable para guardar la información, luego se guarda la información con un arreglo y se crea una nueva variable para obtener los resultados y se retorna esta variable en este caso retorna vacío porque no obtuvo la persona dado el usuario.

```

157 function obtenerPersonaDadoUsuario($usuario){
158     $url_data = variable_get('directorio_url_config');
159     $terminos = '';
160     $terminos .= '{ "term" : {"baja" : false}} ';
161     $terminos .= ', {"term":{"nombre_usuario.keyword":"' . $usuario . '"}}';
162     $variables = '';
163     if ($variables=='')
164         $variables = '*';
165     $result = general_query($url_data,$size=1,$from=0,$terminos,$variables);
166     $datos_personales = json_decode($result);
167     return $datos_personales;
168 }
    
```

Se llevaron a cabo tres iteraciones para detectar no conformidades, las cuales arrojaron los siguientes resultados como se muestra en la tabla:

Tabla 7: Tabla de iteraciones (Elaboración propia).

Iteración	Cantidad de no conformidades	Asociadas a
Primera	8	Errores de validación, entradas y salidas incorrectas.
Segunda	4	Errores de validación.
Tercera	0	-

3.3.2 Prueba de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del *software* y realizar pruebas para descubrir errores asociados con la interconexión. Los programadores escriben pruebas de integración de código para asegurarse de que las pequeñas unidades de código trabajen juntas según lo previsto. Esta práctica ha sido adoptada por muchos equipos, porque es una forma inteligente de analizar el diseño de *software* y prevenir defectos.

Existen dos tipos de integración, la incremental y la no incremental. La integración no incremental intenta construir el programa usando un enfoque donde todos los componentes se combinan de antemano. La corrección es difícil porque el aislamiento de las causas se complica por la vasta extensión de todo el programa. Una vez que se corrigen errores, aparecen otros nuevos y el proceso continúa en un bucle aparentemente interminable. Sin embargo, la integración incremental construye y prueba el programa en pequeños incrementos, donde los errores son más fáciles de aislar y corregir. Es más probable que las interfaces se prueben completamente; y se puede aplicar un enfoque de prueba sistemático (Pressman y Maxim 2015).

Para realizar las pruebas de integración de la solución propuesta se decidió emplear la integración incremental por las facilidades que brinda al desarrollador realizando las pruebas de manera menos compleja y obteniendo mejores resultados. A continuación, se presenta las diferentes estrategias empleadas.

Estrategias de integración

Top-down (descendente): La prueba de integración descendente es un enfoque incremental para la construcción de la arquitectura del *software*. Los módulos se integran moviéndose hacia abajo a través de la jerarquía de control, comenzando con el módulo de control principal (programa principal). Los módulos subordinados al módulo de control principal se incorporan a la estructura ya sea en profundidad o primero en amplitud (Pressman y Maxim, 2015)

El proceso de integración se realiza en una serie de cinco pasos:

1. El módulo de control principal se utiliza como controlador de prueba y los apéndices se sustituyen por todos los componentes directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración seleccionado (es decir, primero en profundidad o primero en amplitud), los apéndices subordinados se reemplazan uno a la vez con componentes reales.

3. Las pruebas se realizan a medida que se integra cada componente.
4. Al completar cada conjunto de pruebas, se reemplaza otro apéndice con el componente real.
5. Se pueden realizar pruebas de regresión para garantizar que no se hayan introducido nuevos errores.
6. El proceso continúa desde el paso 2 hasta que se construye toda la estructura del programa.

Bottom-up (ascendente): Las pruebas de integración ascendente, como su nombre lo indica, comienzan la construcción y las pruebas con módulos atómicos (es decir, componentes en los niveles más bajos en la estructura del programa). Debido a que los componentes se integran de abajo hacia arriba, la funcionalidad proporcionada por los componentes subordinados a un nivel dado siempre está disponible y se elimina la necesidad de apéndices (Pressman y Maxim, 2015)

Una estrategia de integración de abajo hacia arriba se puede implementar con los siguientes pasos:

1. Los componentes de bajo nivel se combinan en grupos (a veces llamados construcciones) que realizan una sub-función de *software* específica.
2. Un controlador (un programa de control para pruebas) está escrito para coordinar el ingreso y salida de prueba.
3. Se prueba el conjunto o construcción.
4. Los controladores se eliminan y los grupos se combinan moviéndose hacia arriba en la estructura del programa.

Se empleó un acercamiento práctico de manera descendente para los niveles superiores de la estructura del programa y ascendente para los niveles inferiores combinando lo mejor de ambas técnicas.

Los métodos de caja negra, también denominados pruebas de comportamiento, se centran en los requisitos funcionales del software. O sea, permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Este método intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a base de datos externas, errores de rendimiento y errores de inicialización y de terminación (Pressman y Maxim, 2015).

Resultados

Se aplicó la prueba a la integración al módulo Directorios, obteniéndose los resultados que a continuación se muestran:

Tabla 8: Prueba de integración (Elaboración propia).

Caso de prueba de integración del componente: Mostrar datos de la persona	
Sistema/componente al que se integra	Directorios
Condiciones de ejecución	Debe existir la persona.
Descripción de la prueba	Comprobar que el componente Mostrar datos de la persona muestre correctamente los datos.
Pasos de ejecución	El probador interactúa con el componente. Se inserta algún parámetro de búsqueda con valores válidos y no válidos. Se emite o no la información según corresponda en cada instante.
Resultados esperados	El componente “Mostrar datos de la persona” brinda toda la información solicitada. Posee las validaciones pertinentes a los campos y muestra satisfactoriamente los datos.
Evaluación	Prueba satisfactoria.

3.3.3 Pruebas de sistema

El *software* es solo un elemento de un sistema más grande, a este lo acompañan otros elementos como, por ejemplo, el hardware que lo soporta, la información que procesa y las personas que lo utilizan. Las pruebas de sistemas verifican que todos los elementos se acoplan correctamente y que se logra la función o rendimiento general del sistema. Los pasos tomados durante el diseño y las pruebas del *software* pueden mejorar en gran medida la probabilidad de una integración exitosa del *software* en el sistema más grande. Se debe anticipar posibles problemas de interfaz y diseñar rutas de manejo de errores que prueben toda la información proveniente de otros elementos (Pressman y Maxim, 2015).

3.3.3.1 Pruebas de rendimiento

Para sistemas en tiempo real e integrados, el *software* que proporciona la función requerida pero no cumple con los requisitos de rendimiento es inaceptable. Las pruebas de rendimiento están diseñadas para probar el rendimiento en tiempo de ejecución del *software* en el contexto de un sistema integrado. Estas pruebas se realizan en todos los pasos del proceso de prueba. Incluso a nivel de unidad, el rendimiento de un módulo individual puede evaluarse a medida que se realizan las pruebas. Sin embargo, no es hasta que todos los elementos del sistema están completamente integrados que se puede determinar el verdadero rendimiento de un sistema. Las pruebas de rendimiento a menudo se combinan con las pruebas de estrés y generalmente requieren instrumentación tanto de hardware como de *software*. Es decir, a menudo es necesario medir la utilización de los recursos (por ejemplo, los ciclos del procesador) de una manera precisa. La instrumentación externa puede monitorear los intervalos de ejecución, registrar eventos (por ejemplo, interrupciones) a medida que ocurren, y muestrea los estados de las máquinas regularmente. Al instrumentar un sistema, el probador puede descubrir situaciones que conducen a la degradación y la posible falla del sistema (Pressman y Maxim, 2015).

Resultados de las pruebas de rendimiento.

Una prueba de rendimiento se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. El sistema se prueba con un número de 5000 visitas diarias y 1000 usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la prueba. Esta prueba puede mostrar los tiempos de respuesta de la aplicación. Para llevar a cabo las pruebas de rendimiento se empleó la herramienta Apache JMeter en su versión 3.0.

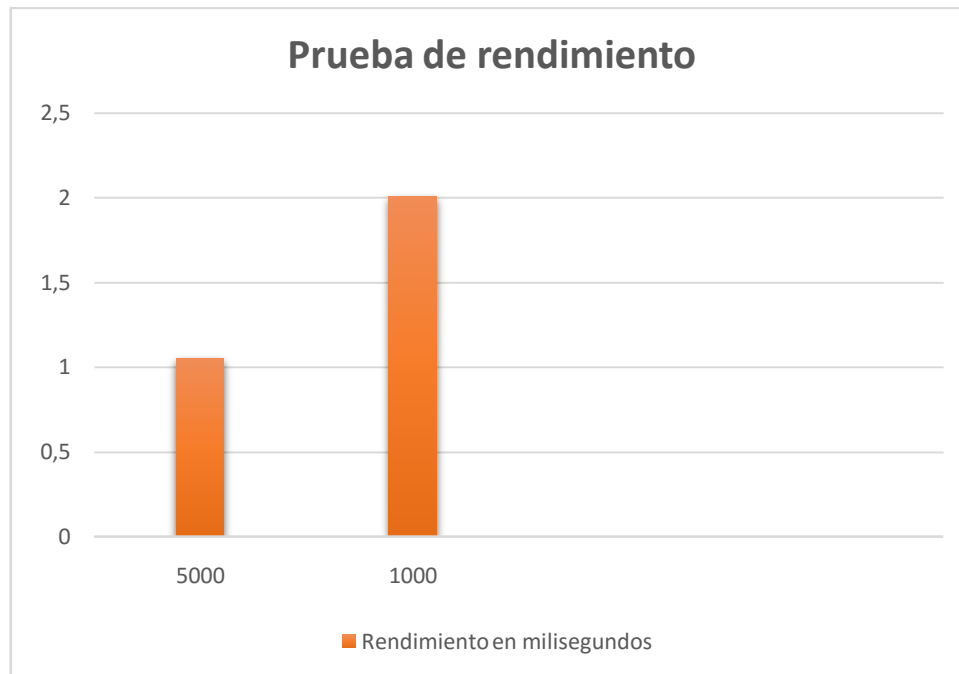


Figura 13: Resultado de la prueba de rendimiento (Elaboración propia).

Se puede observar que las peticiones son ejecutadas en tiempos inferiores a 7milisegundos. Para todas las visitas, la ocurrencia de errores se mantiene en 0.0%, evidenciándose que todas las peticiones realizadas se ejecutan satisfactoriamente. También se demuestra que la propuesta de solución es estable, debido a que se mantuvo brindando servicios sin incurrir en errores.

3.3.4 Pruebas de aceptación

Cuando se crea un *software* personalizado para un cliente, se realizan una serie de pruebas de aceptación para que el cliente pueda validar todos los requisitos. Conducido por el usuario final en lugar de por los ingenieros de *software*, una prueba de aceptación puede ir desde una "prueba de manejo" informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, las pruebas de aceptación pueden llevarse a cabo durante un período de semanas o meses, descubriendo así errores acumulativos que podrían degradar el sistema con el tiempo. Para descubrir errores que solo el usuario final parece poder encontrar se utiliza un proceso llamado prueba alfa y beta.

La prueba alfa se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales. El *software* se utiliza en un entorno natural con el desarrollador "mirando por encima del hombro" de los usuarios y registrando errores y problemas de uso. Las pruebas alfa se realizan en un

ambiente controlado. La prueba beta se realiza en uno o más sitios de usuarios finales. A diferencia de las pruebas alfa, el desarrollador generalmente no está presente. Por lo tanto, la prueba beta es una aplicación "en vivo" del *software* en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que se encuentran durante las pruebas beta y los informa al desarrollador a intervalos regulares (Pressman y Maxim, 2015).

Se aplicó las pruebas alfa, tomando como referencia las historias de usuarios comprobando que el sistema responde a las necesidades del cliente.

3.4 Conclusiones parciales

- Se detallaron los estándares de codificación empleados en el mantenimiento perfectivo de los directorios del portal web Intranet permitiendo una mejor organización y comprensión del código.
- Con los resultados de la validación de los requisitos identificados se logró obtener un listado de requisitos correctamente redactados y descritos.
- La descripción de las pruebas utilizadas para asegurar la calidad del *software*, permitió obtener resultados satisfactorios, asegurando que el sistema implementado no contiene errores y tiene la aceptación requerida.

Conclusiones

Al terminar la presente investigación se dio solución a la problemática planteada, contribuyendo a mejorar la escalabilidad, modularidad y disminución del tiempo de respuesta, se obtuvieron resultados que permiten arribar a las siguientes conclusiones:

- Con el estudio de soluciones informáticas homólogas se identificaron las principales tendencias en el desarrollo de los directorios en diferentes instituciones, tomando algunas características de los directorios implementados en la universidad.
- El análisis de los principios teóricos de la investigación, enfocados a los sistemas de búsqueda de información, permitió un mejor entendimiento de los procesos de este tipo en la universidad.
- El empleo de un proceso de desarrollo con enfoque ágil, herramientas y tecnologías libres favoreció el trabajo del equipo y como resultado se obtuvo un sistema que cumple con los requerimientos definidos.
- Se generaron los artefactos correspondientes al modelado del negocio donde se describieron y se diseñaron las funcionalidades del proceso de mantenimiento perfectivo de los directorios del portal web intranet.
- Con la adopción de un estándar de código se logró la organización, legibilidad y comprensión del código generado en la implementación del sistema propuesto.
- Una vez aplicadas las pruebas de software, se garantiza el cumplimiento de las necesidades y requisitos establecidos por el cliente.

Recomendaciones

Una vez concluida la investigación y el desarrollo del mantenimiento perfecto de los directorios del portal web intranet se recomienda:

1. Diseñar un mecanismo para realizar recomendaciones de búsquedas a los usuarios basándose en los criterios seleccionados y el historial de las búsquedas realizadas.

Referencias Bibliográficas.

- APACHE, 2019. Welcome to The Apache Software Foundation! [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: <https://www.apache.org/>.
- BERT BOS, 2019. Cascading Style Sheets. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.w3.org/Style/CSS/Overview.en.html>.
- CHEN, C., 2019. Significado de Sistema de información. *Significados* [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.significados.com/sistema-de-informacion/>.
- DE LUCA, D., 2011. *HTML5: entienda el cambio, aproveche su potencial* [en línea]. S.I.: Fox Andina. ISBN 978-987-1773-79-4. Disponible en: <https://books.google.com.cu/books?id=DGwjKIXWPm8C>.
- DIRECTORIO UGR, 2019. Directorio de la Universidad de Granada > Presentación. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://directorio.ugr.es/>.
- DRUPAL, 2020. Coding standards. [en línea]. Disponible en: <https://www.drupal.org/docs/develop/standards>.
- ELASTIC, 2019. Open Source Search: The Creators of Elasticsearch, ELK Stack & Kibana | Elastic. [en línea]. [Consulta: 3 diciembre 2019]. Disponible en: <https://www.elastic.co/>.
- EMPRESA DE TELECOMUNICACIONES DE CUBA S.A., 2016. Directorio telefónico. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: http://www.etcса.cu/directorio_telefonico/.
- FLANAGAN, D., 2006. *JavaScript: The Definitive Guide*. S.I.: O'Reilly Media, Inc. ISBN 978-0-596-10199-2.
- FOROUZAN, M.-H., 2017. *Tcp/Ip Protocol Suite, 4E* [en línea]. S.I.: McGraw-Hill Education (India) Pvt Limited. ISBN 978-0-07-070652-1. Disponible en: <https://books.google.com.cu/books?id=Vx7M6pSmyRMC>.
- FOUNDATION, A.S., 2020. Apache JMeter. [en línea]. Disponible en: www.jmeter.apache.org.

- GAMMA, E., 2002. *Patrones de diseño: elementos de software orientado a objetos reutilizable* [en línea]. S.I.: Pearson Educación. ISBN 978-84-7829-059-8. Disponible en: https://books.google.com.cu/books?id=gap_AAAACAAJ.
- GARCÍA, I., 2018. ¿Qué es un sistema de información?| Definición de sistema de información. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.emprendepyme.net/que-es-un-sistema-de-informacion.html>.
- GITLAB, 2020. The first single application for the entire DevOps lifecycle - GitLab | GitLab. [en línea]. [Consulta: 15 febrero 2020]. Disponible en: <https://about.gitlab.com/>.
- GOYAL, D., BALAMURUGAN, S., PENG, S.L. y VERMA, O.P., 2020. *Design and Analysis of Security Protocol for Communication* [en línea]. S.I.: Wiley. ISBN 978-1-119-55564-3. Disponible en: <https://books.google.com.cu/books?id=O3LQDwAAQBAJ>.
- HAMIDIAN, B.F.F. y OSPINO, G.R.S., 2015. ¿Por qué los sistemas de información son esenciales? [en línea], vol. 38. ISSN 1316-5852. Disponible en: <http://servicio.bc.uc.edu.ve/derecho/revista/idc38/art07.pdf>.
- HEINLEIN, P. y HARTLEBEN, P., 2008. *The Book of IMAP: Building a Mail Server with Courier and Cyrus* [en línea]. S.I.: Open Source Press. ISBN 978-1-59327-177-0. Disponible en: <https://books.google.com.cu/books?id=YUUSdUtay1QC>.
- INTERNATIONAL STANDARD SERIAL NUMBER, 2019. ROAD, Directorio de Recursos Científicos y Universitarios en acceso abierto | ISSN. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.issn.org/es/servicios-y-prestaciones/servicios-en-linea/road-directorio-de-recursos-cientificos-y-universitarios-en-acceso-abierto/>.
- JS.FOUNDATION, J.F.-, 2019. jQuery. [en línea]. [Consulta: 10 diciembre 2019]. Disponible en: <https://jquery.com/>.
- KRISHNAMURTHY, B. y REXFORD, J., 2001. *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement* [en línea]. S.I.: Addison-Wesley. ISBN 978-0-201-71088-5. Disponible en: https://books.google.com.cu/books?id=g_1SAAAAMAAJ.

- MORENO, B., 2019. Drupal 8, el CMS más escalable. *Blog de Hiberus Tecnología* [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.hiberus.com/crecemos-contigo/drupal-8-cms-mas-escalable/>.
- OBJECT MANAGEMENT GROUP, 2007. About the Unified Modeling Language Specification Version 2.1.2. [en línea]. [Consulta: 10 marzo 2020]. Disponible en: <https://www.omg.org/spec/UML/2.1.2#specification-metadata>.
- PEÑARANDA, C.O.P., 2015. *Diseño e implementación de un sitio web con Drupal* [en línea]. S.I.: Universidad Politécnica de Valencia. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/54031/POZO%20-%20Dise%C3%B1o%20e%20implementaci%C3%B3n%20de%20un%20sitio%20web%20con%20Drupal..pdf?sequence=1>.
- PGADMIN, 2019. pgAdmin - PostgreSQL Tools. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.pgadmin.org/>.
- PHP, 2019. PHP: Hypertext Preprocessor. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.php.net/>.
- PHPSTORM, 2019. PhpStorm: el IDE rápido e inteligente para programación en PHP de JetBrains. *JetBrains* [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: <https://www.jetbrains.com/phpstorm/>.
- POSTGRESQL, 2019. PostgreSQL: About. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.postgresql.org/about/>.
- PRESSMAN, R.S. y MAXIM, B.R., 2015. *Software Engineering: A Practitioner's Approach* [en línea]. S.I.: McGraw-Hill Education. ISBN 978-1-259-25315-7. Disponible en: <https://books.google.com.cu/books?id=XQztoAEACAAJ>.
- RAMOS, D., NORIEGA, R., LAÍNEZ, J.R. y DURANGO, A., 2017. *Curso de Ingeniería de Software: 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5441-3253-2.
- RODRÍGUEZ, J.M., 2012. *Sistema de búsqueda en Internet* [en línea]. S.I.: Universidad Autónoma del Estado de Hidalgo. Disponible en: https://www.uaeh.edu.mx/docencia/VI_Lectura/maestria/documentos/LECT43.pdf.

- SCHMIED, W. y CHELLIS, J., 2006. *MCSA / MCSE: Exchange Server 2003 Implementation and Management Study Guide: Exam 70-284* [en línea]. S.l.: Wiley. ISBN 978-0-7821-5139-8. Disponible en: <https://books.google.com.cu/books?id=EJ0EfhFhiJoC>.
- SCHWABER, K. y SUTHERLAND, J., 2017. *La Guía Definitiva de Scrum: Las Reglas del Juego* [en línea]. noviembre 2017. S.l.: s.n. [Consulta: 26 febrero 2020]. Disponible en: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>.
- SOMMERVILLE, I., 2011. *Software engineering 9th Edition*. S.l.: s.n. ISBN 10137035152.
- SYMFONY y TWIG, 2020. Twig The flexible, fast, and secure template engine for PHP. [en línea]. Disponible en: www.twig.symfony.com.
- THORNTON, J., OTTO, M. y BOOTSTRAP, 2018. Bootstrap. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://getbootstrap.com/>.
- TORRES, E.V., 2013. INTRODUCCIÓN. *prezi.com* [en línea]. [Consulta: 9 diciembre 2019]. Disponible en: <https://prezi.com/vb-b3rziyx1o/introduccion/>.
- TRIGAS, M., 2015. *Metodología Scrum*. 2015. S.l.: s.n.
- UNIVERSIDAD DE BURGOS, 2019. Directorio | Universidad de Burgos. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.ubu.es/la-universidad/organizacion/directorio>.
- UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2018a. Directorios P | Universidad de las Ciencias Informáticas. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.uci.cu/directorio/personas>.
- UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2018b. Directorios T | Universidad de las Ciencias Informáticas. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.uci.cu/directorio/telefonos>.
- UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2018c. Directorios U | Universidad de las Ciencias Informáticas. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://www.uci.cu/directorio/universidades>.

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2019a. Directorio de personas | Directorio UCI. [en línea]. [Consulta: 30 noviembre 2019]. Disponible en: <https://directorio.uci.cu/>.

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2019b. Directorio de personas | Intranet 3.0. [en línea]. [Consulta: 2 diciembre 2019]. Disponible en: <https://intranet.uci.cu/directorio/personas>.

VISUAL PARADIGM, 2019. Ideal Modeling & Diagramming Tool for Agile Team Collaboration. [en línea]. [Consulta: 26 febrero 2020]. Disponible en: <https://www.visual-paradigm.com/>.

Anexos

Encuesta

Dirigida A: Especialistas

Objetivo: Identificar las características de los directorios del portal Intranet relacionados con: **escalabilidad**, **modularidad**, y **rendimiento**, que a su juicio ofrezca oportunidades de mejoras.

Gracias por su colaboración

Nombre y apellidos:

Rol:

Año de graduación:

Años de experiencia:

Preguntas

1. Actualmente el portal Intranet se encuentra desarrollado sobre el CMS Drupal en su versión 7 el cual tiene soporte solo hasta el 2020, lo cual representa un elemento desfavorable para la comunidad. Una vez que se entre en este periodo ¿cuál sería la respuesta del equipo de desarrollo ante esta dificultad?
2. El portal Intranet tiene integrado los directorios de la universidad, actualmente funcionando el de personas, ¿Considera que el sistema puede adaptarse a una nueva tecnología?
3. En el caso de que una de las alternativas para dar solución al soporte sea migrar a una tecnología ¿Cuál sería la más indicada y por qué?
4. Atendiendo a la respuesta anterior ¿Considera que el sistema puede adaptarse a un nuevo entorno y nuevas funcionalidades sin perder la calidad en los servicios ofrecidos?
5. Teniendo en cuenta que el rendimiento se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos y la eficiencia, ¿Qué tiempo demora en mostrar la respuesta el sistema?
6. ¿Cuántas personas al mismo tiempo soporta el sistema?
7. Atendiendo a que la nueva versión del CMS mejora la efectividad de la modularidad ¿Cree usted que

Anexo 1: Encuesta realizada a los trabajadores de la DIN.

Anexo 2: Caso de prueba correspondiente a la funcionalidad Mostrar datos de la persona.

Escenario	Descripción	Número de páginas	Cantidad por páginas	Respuesta del sistema	Flujo central
<p>EC 2.1 Mostrar datos de la persona correctamente.</p>	<p>Este escenario muestra toda la información relacionado con la persona que coincida con los parámetros de búsqueda establecidos.</p>	<p>N/A</p>	<p>N/A</p>	<p>El sistema muestra un listado con los datos de las personas que coincidan con los parámetros de búsqueda establecidos: Nombre, Apellido, Usuario, Solapín, Apartamento, Categoría persona, Categoría docente, Rol universitario, Cargo (si no es estudiante), Área, País, Provincia, Municipio, Facultad, Grupo (si es estudiante), Grado académico, Edificio, Teléfono, Carrera</p>	<p>1. Al acceder el usuario al portal web Intranet, se muestran por defecto todos los iconos flotantes que están contenidos dentro. 2. El usuario selecciona el icono del directorio de personas. 3. Introduce algún parámetro de búsqueda o se selecciona del botón filtrar búsqueda.</p>
<p>EC 2.2 Seleccionar correctamente la cantidad de personas por</p>	<p>Mediante este escenario el usuario escoge la cantidad de personas a mostrar por</p>	<p>N/A</p>	<p>El usuario escoge la opción que desea</p>	<p>El sistema muestra la cantidad de elementos según la opción escogida (Se puede escoger</p>	<p>1. Al acceder el usuario al portal web Intranet, se muestran por defecto todos</p>

página.	página (los valores a escoger son 5, 10, 20, 40).		que sea mostrada por el sistema (Se puede escoger 5, 10, 20, 40).	5, 10, 20, 40).	los iconos flotantes que están contenidos dentro. 2. El usuario selecciona el icono del directorio de personas. 3. Introduce algún parámetro de búsqueda o se selecciona del botón filtrar búsqueda y luego selecciona cuantos elementos desea sean mostrados.
EC 2.3 Seleccionar la página siguiente.	Mediante este escenario el usuario siguiente para mostrar las personas de la siguiente página.	El usuario selecciona la opción Siguiete debajo del listado de elementos.	N/A	El sistema muestra la página solicitada por el usuario.	1. Al acceder el usuario al portal web Intranet, se muestran por defecto todos los iconos flotantes que están

						<p>contenidos dentro.</p> <p>2. El usuario selecciona el icono del directorio de personas.</p> <p>3. Introduce algún parámetro de búsqueda o se selecciona del botón filtrar búsqueda y luego selecciona debajo del listado de elementos la opción Siguiente.</p>
EC 2.4	No existe la persona.	Mediante este escenario en caso de que no exista ninguna persona se muestra un listado vacío.	NA	NA	El sistema muestra el listado vacío.	<p>1. Al acceder el usuario al portal web Intranet, se muestran por defecto todos los iconos flotantes que están contenidos dentro.</p>

					<p>2. El usuario selecciona el icono del directorio de personas.</p> <p>3. Introduce algún parámetro de búsqueda o se selecciona del botón filtrar búsqueda y se muestra un listado vacío indicado que no existe la persona.</p>
--	--	--	--	--	--

Anexo 3:Historia de usuario del requisito funcional Realizar Mostrar datos de la persona.

Número:	2	Nombre del requisito: Mostrar datos de la persona
Programador: Beatriz Pérez Delgado	Iteración Asignada: 1	
Prioridad: Alta	Tiempo Estimado: 1	
Riesgo en Desarrollo: N/A	Tiempo Real: 1	
Descripción:		
<p>Se muestra el listado de personas que aparecen según el parámetro de búsqueda que haya sido introducido por el usuario. Se muestra con los datos <i>Nombre, Apellido, Usuario, Solapín, Apartamento, Categoría persona, Categoría docente, Rol universitario, Cargo, Área, País, Provincia, Municipio, Género, Facultad, Grupo, Grado académico, Unidad organizativa, Edificio, Teléfono, Carrera.</i></p>		
Observaciones:		

Prototipo elemental de interfaz gráfica de usuario:N/A