

Universidad de las Ciencias Informáticas

Facultad 1



## **Estrategia de pruebas funcionales para el desarrollo de Portales web con DRUPAL en CIDI.**

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

**Autor:**

Daylin Hernández Pérez.

**Tutores:**

Prof. As. Ing. Yordanka Fuentes Castillo.

Ing. Ively Viera Fajardo.

Prof. As. Ing. Serguey González Garay.

La Habana, enero de 2021  
“Año 63 de la Revolución”

## Declaración de autoría

Declaro por este medio que yo **Daylin Hernández Pérez**, con carné de identidad **98070607216** soy el autor principal del trabajo titulado “**Estrategia de pruebas funcionales para el desarrollo de Portales web con DRUPAL en CIDI**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los 7 días del mes de diciembre de 2021.



Daylin Hernández Pérez

Nombre del autor

Autor



Prof. As. Ing. Yordanka Fuentes Castillo

Nombre del tutor



Ing. Ively Fajardo

Nombre del tutor



Prof. AS. Ing. Serguey Gonzales Garay

Nombre del tutor

## **Resumen**

La realización de pruebas de software es una parte esencial y fundamental para cualquier sistema, ya que garantiza añadir cierto grado de calidad antes de ser liberado para los usuarios finales. Además de garantizar calidad, es reducir costo y esfuerzo. Hoy en día en la automatización de las pruebas consiste en el uso de software especial para controlar la ejecución de las pruebas y la comparación entre los resultados obtenidos y los resultados esperados. Actualmente el centro CIDI emplea un esfuerzo mayor durante el proceso de pruebas funcionales manuales teniendo en cuenta la cantidad de proyectos en los que se desenvuelven esto puede traer consecuencias graves para el centro. La presente investigación expone el resultado del desarrollo de una nueva estrategia de pruebas.

Palabras clave: Pruebas, software, calidad, automatización.

---

## Índice

<b>Introducción</b> .....	1
<b>Capítulo 1: “Fundamentación teórica del uso de las pruebas de software.”</b> .....	8
1.1    Introducción. ....	8
1.2    Conceptos asociados al dominio del problema de la investigación. ....	8
1.2.1    Pruebas de software. ....	8
1.2.2    Estrategia de prueba de software. ....	8
1.2.2.1    Descripción de la estrategia de prueba de software.....	9
1.2.3    Técnicas de diseño de pruebas.....	9
1.2.3.1    Enfoques principales para el diseño de casos de prueba. ....	10
1.2.3.2    Diseño de las pruebas del sistema.....	10
1.3    Pruebas funcionales automatizadas en aplicaciones desarrolladas con Drupal 8.....	11
1.3.1    Configuración para ejecutar pruebas PHPUnit .....	12
1.3.2    Ejecutando pruebas.....	15
1.3.3    Automatizando test funcionales con Behat.....	17
1.3.4    Pasos para hacer un test Behat: .....	18
1.4    Tipos de pruebas. ....	20
1.5    Diagnóstico inicial de la situación actual de la ejecución de las pruebas funcionales.....	26
1.6    Tecnologías para el modelado y el desarrollo que serán utilizadas. ....	29
1.6.1    Metodología de desarrollo. ....	29
1.6.2    Herramientas para el desarrollo. ....	30
1.6.3    Herramientas para pruebas.....	34
1.6.4    Lenguaje y herramienta para el modelado. ....	35
1.7    Conclusiones. ....	35
<b>Capítulo 2</b> .....	37
2.1    Introducción.....	37
2.2    Propuesta de solución.....	37

---

2.3 Estructura del portal:.....	37
2.4 Especificación de los requisitos.....	41
2.4.1 Requisitos funcionales.....	41
2.4.2 Requisitos no funcionales.....	44
2.5 Descripción de requisitos de software (Historias de Usuario).....	45
2.6 Modelado de diseño.....	48
2.6.1 Diagramas de clases del diseño con estereotipos web (DCD).....	48
2.6.2 Diagramas de Secuencia.....	49
2.7 Estrategia para aplicar las pruebas funcionales al portal.....	50
2.7.1 Estrategia para las pruebas funcionales manualmente.....	50
2.7.2 Estrategia para pruebas funcionales automatizadas en Drupal.....	54
2.8 Conclusiones.....	58
<b>Capítulo 3: “Validación de la propuesta para las pruebas de software.”.....</b>	<b>60</b>
3.1 Introducción.....	60
3.2 Modelo de despliegue.....	60
Diagrama de componentes.....	61
3.3 Estándares de codificación de Drupal.....	62
3.4 Aplicación de la estrategia de validación del portal web IPROYAZ.....	65
3.4.1 Resultados de las pruebas no funcionales:.....	65
3.4.2 Resultados de las pruebas funcionales:.....	68
3.4.3 Pruebas automatizadas:.....	69
3.5 Propuesta de estrategia final:.....	70
3.5 Interfaces del portal web Iproyaz:.....	72
3.6 Conclusiones parciales del capítulo:.....	73
<b>Conclusiones Generales.....</b>	<b>74</b>
<b>Recomendaciones.....</b>	<b>75</b>
<b>Referencias Bibliográficas.....</b>	<b>76</b>
<b>Tablas.....</b>	<b>78</b>
Historias de usuario.....	80

---

<b>Figuras</b> .....	87
Diagrama de clases del diseño para historia de usuario. ....	87
Diagramas de secuencia para historias de usuario .....	89
<b>Casos de Prueba</b> .....	93
Encuesta .....	96

---

**Índice de tabla**

Tabla 1NC de aplicaciones .....	26
Tabla 2NC de documentación. ....	27
Tabla 3 NC en la aplicación por iteración en cada proyecto. ....	28
Tabla 4 NC en la documentación por cada proyecto. ....	28
Tabla 5CP Mostrar servicio.....	53
Tabla 6Descripción de las variables.....	54
Tabla 7 Resultados de la prueba de carga y estrés.....	66
Tabla 8 Chequeo de usabilidad.....	67
Tabla 9 Tabla de NC de una de las NC encontradas. ....	69
Tabla 10 Ventajas y desventajas de las pruebas manuales y las pruebas automatizadas.....	71
Tabla 11 Descripción de las clases válidas y clases inválidas.....	78
Tabla 12 Descripción del Caso de Prueba. ....	78
Tabla 13 Descripción de la tabla de No Conformidades para pruebas funcionales. ....	79
Tabla 14 Descripción de la tabla de No Conformidades para Pruebas no Funcionales.....	79
Tabla 15Crear usuario .....	80
Tabla 16 Editar usuario.....	81
Tabla 17 Mostrar listado de usuarios.....	83
Tabla 18Cancelar cuenta de usuario.....	83
Tabla 19 Crear noticia.....	84
Tabla 20 Mostrar noticia. ....	87

**Índice de figuras**

Figura: 1 Arquitectura de Información para la página de 1er nivel. Ejemplo: Home sección 1 .....39

Figura: 2 Arquitectura de Información para la página de 1er nivel. Ejemplo: Home sección 2.....41

Figura: 3 CP Crear servicio .....51

Figura: 4 CP Editar servicio .....52

Figura: 5Diagrama de despliegue .....61

Figura: 6Diagrama de Componente.....62

Figura: 7 Resultado de las pruebas de funcionales por casos de prueba.....69

Figura: 8 Test Behat creado.....70

Figura: 9 Banner del portal web lproyaz. ....72

Figura: 10 Bloque de servicios del portal web lproyaz. ....73

Figura: 11 Gestionar usuario .....88

Figura: 12 Gestionar Noticia. ....89

Figura: 13 Editar servicio .....90

Figura: 14 Mostrar servicio .....91

Figura: 15 Crear noticia .....92

Figura: 16 Crear usuario .....93

Figura: 17Editar usuario .....94

Figura: 18 Mostrar listado de usuario .....95

### **Introducción**

En los últimos años, las Tecnología de la Información y la Comunicación (TIC) han tomado un papel importantísimo en la sociedad y se utilizan en múltiples o variadas actividades. Las llamadas TIC son los recursos y herramientas que se utilizan para el proceso, administración y distribución de la información a través de elementos tecnológicos, como: ordenadores, teléfonos, televisores, etc.(«Universidad Latina de Costa Rica», s. f.). Hoy en día ya forman parte de la mayoría de sectores: educación, robótica, administración pública, empleo y, empresas, salud, entre otros.

Las TIC permiten tener acceso a la información de manera fácil y rápida en cualquier formato, esto es posible a través de la digitalización de la información que es almacenada en grandes cantidades; con lo cual puede conseguir información y comunicarse inmediatamente a pesar de encontrarse a kilómetros de la fuente original. Las nuevas TIC se caracterizan por permitir la comunicación bidireccional entre personas o grupos sin importar donde se encuentren.(Biblioteca Medica Nacional, s. f.) En este sentido actualmente los portales web son una clara demostración de la comunicación bidireccional, pues cada vez son más los usuarios que acceden a Internet buscando información, producto y/o servicio que brindan los mismos.

Hoy en día la mayoría de las empresas cuentan con un portal web que le permite ofrecer, informar y publicitar contenidos o vender, productos y servicios al resto del mundo; por eso no contar con una web supone una gran desventaja competitiva con respecto a otras empresas que si la tienen. Los portales web representan un cambio de paradigma para las empresas online, que pasan a construir sitios web pensando en el cliente, en lugar de centrarse en los productos. Idealmente, un portal web permite diseñar sitios con una mejor navegación basados en las necesidades del usuario.

Un aspecto fundamental que hace que los portales web cuenten con una mayor cantidad de visitas es la calidad del mismo. Un sitio de calidad tiene las siguientes características: utilidad, usabilidad y deseabilidad. Teniendo en cuenta este aspecto, se puede decir que la gestión de la calidad del software es uno de los procesos más importante dentro del ciclo de desarrollo. Es un pilar fundamental a lo largo del ciclo de vida de un producto, teniendo como propósito entender las expectativas del cliente en términos de calidad, y poner en práctica un plan proactivo para satisfacer esas expectativas. El proceso de pruebas, y específicamente, la aplicación de pruebas de sistema son las encargadas de comprobar el buen

funcionamiento y la calidad del software. (Toll Palma & Ril Gil)

No cabe duda que las TIC tienen un gran impacto en la sociedad, evidencia de ello es la utilización de las mismas a diario por las personas y también por las empresas, como se ha mencionado antes a través de los sitios web. Por eso en el país se ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las TIC y lograr una cultura digital, lo que facilitaría a la sociedad acercarse más hacia el objetivo de un desarrollo sostenible. Se han desarrollado múltiples programas encaminados a lograr la informatización de la sociedad; que están relacionados con el proceso de utilización ordenada y masiva de las TIC para satisfacer las necesidades de información y conocimiento de la sociedad lo que constituye una de las metas que tiene Cuba, en el presente y para los próximos años. (Carbó)

Parte importante del éxito de este proceso de informatización de la sociedad es la creación de la Universidad de las Ciencias Informáticas (UCI) en el año 2002 con el objetivo fundamental de la formación recursos humanos en este campo; del Ministerio de la informática y las Comunicaciones. La UCI además de formar profesionales capacitados de ejercer esta labor en las empresas a lo largo y ancho del país, cuenta con 15 centros que desarrollan actividades de I+D+i<sup>1</sup>, encargados de las aplicaciones informáticas, el desarrollo tecnológico y las investigaciones asociadas para sustentar la informatización de la sociedad. Algunos de estos centros se encuentran vinculados a las diferentes facultades que forman parte del campus universitario, como es el caso del Centro de Innovación y Desarrollo para Internet (CIDI), perteneciente a la Facultad 1. (Lima, 2020)

CIDI es un centro de desarrollo que tiene entre sus funciones principales el desarrollo de portales web e intranets. A día de hoy el número de solicitudes que recibe el centro están entre 10 y 20 al año, de las cuales, en ocasiones, no todas puedan ser atendidas. Esto es provocado por dos causas fundamentales, la primera está relacionada con la cantidad de personal vinculado al proceso productivo y la segunda con el tiempo de duración de desarrollo del proyecto. Una de las actividades más importantes dentro del desarrollo del proyecto, es el aseguramiento de la calidad del sistema desde el punto de vista funcional. Actualmente las pruebas funcionales a los portales e intranets desarrolladas se hacen de forma manual.

---

En una encuesta realizada para el análisis del proceso de ejecución de las pruebas funcionales de forma manual en el centro se obtuvieron los siguientes resultados a considerar:

El 80 % del personal del centro participa en el proceso de prueba.

El 78% coinciden en que el personal del centro no es suficiente, por este motivo los jefes de proyecto, analistas y desarrolladores participan en el proceso de pruebas.

El 25 % del personal implicado en este proceso considera que no cuentan con las herramientas necesarias.

El 75% considera que las detecciones de las no conformidades son elevadas lo que hace que se realicen más de una iteración de prueba antes de liberar el producto. Se detecta que esto se debe al hecho de que las pruebas funcionales son realizadas manualmente, lo que implica un factor subjetivo dado por la apreciación de cada probador. Además, se identifica, que los artefactos a probar son realizados por humanos, que además están en varios proyectos a la vez desarrollando varias tareas y desempeñando varios roles, lo que implica que se cometan errores en su elaboración.

El 58% considera que el esfuerzo empleado en este proceso es excesivo porque en ocasiones se encuentran probando más de un producto los cuales hay que aplicarle más de una iteración para detectar todas las no conformidades para que luego sean corregidas.

El 80% considera que la documentación cuenta con la calidad requerida pero no en cuanto a al formato

Teniendo en cuenta los roles, el nivel de implicación en el proceso de pruebas se comporta de la siguiente manera: se involucra el 25 % de los jefes de proyecto, el 25 % de los analistas, el 41 % de los desarrolladores y el 33 % de los probadores.

De manera general lo anteriormente expuesto refleja el esfuerzo que por parte de los trabajadores se emplea en el centro CIDI para realizar las pruebas de software.

Este proceso trae los siguientes inconvenientes:

-Los artefactos son desarrollados por una persona distinta a la que desarrolla y por lo general se hacen en paralelo, para disminuir el tiempo de duración del proyecto. Esto significa que usualmente los documentos se terminan antes que la aplicación, provoca que, si el desarrollador realiza cambios necesarios durante la implementación, los documentos deban actualizarse una y otra vez, incurriendo en un gasto de esfuerzo.

Sería un problema menor, sino fuese por el hecho de que, por la escasez de personal en el centro, un mismo analista puede estar en varios proyectos a la vez, lo que puede dilatar el tiempo de desarrollo o afectar la calidad de los documentos.

-Las personas encargadas de realizar las pruebas son comunes para todos los proyectos. Esto significa que en una misma semana o incluso en un mismo día pueden coincidir más de un proyecto a probar. Lo que conlleva a, en primer lugar, a que, el cronograma de cada proyecto individual se retrase y en segundo, que, al tener tantos proyectos a la vez, las pruebas puedan no salir con la calidad requerida.

-Al realizarse las pruebas funcionales manuales, hay un factor subjetivo provocado por la persona que prueba. Esto significa que en ocasiones hay NC detectadas que son provocadas por la mala interpretación o el desconocimiento de quien prueba. Añadido a esto ocurre además que muchas NC detectadas son producto de elementos que se encuentran de una manera que puede no agradar al probador. Esta realidad produce un número elevado de NC en cada prueba, que dilatan el proceso y además no siempre tienen el impacto que se les da.

A pesar de existir todos los inconvenientes mencionados, las pruebas funcionales se continúan desarrollando manualmente. Esto es provocado por dos factores principales, uno es que la mayoría de los desarrolladores no tienen experiencia en ese tipo de prueba y otro que debido a la cantidad de solicitudes por atender y la poca cantidad de especialistas, el tiempo es primordial y debe aprovecharse al máximo en el desarrollo. Por lo tanto, se hace necesario buscar el modo más efectivo para realizar las pruebas funcionales, de manera que permita disminuir el esfuerzo implicado hoy en este proceso.

Por lo anteriormente descrito se define como **problema de investigación** ¿Cómo disminuir el esfuerzo invertido en la realización de pruebas funcionales a los sistemas desarrollados con Drupal en CIDI?

El **objeto de estudio** está centrado en el proceso de pruebas funcionales durante el desarrollo de un portal web y el **campo de acción** en la realización de pruebas funcionales a portales web desarrollados en Drupal en el centro CIDI.

El **objetivo general** de la investigación es: definir, a partir de un estudio comparativo entre las pruebas funcionales automatizadas que trae Drupal 8 y las manuales que se realizan hoy en el centro, una estrategia de pruebas funcionales para el desarrollo de portales web e intranets con Drupal en CIDI, que permita disminuir el esfuerzo invertido en la realización de esta actividad.

Para guiar el cumplimiento del objetivo planteado, se formulan las siguientes **preguntas científicas**:

- 1- ¿Cuáles son los referentes teóricos que sustentan la realización de pruebas funcionales en portales web e intranets?
- 2- ¿Cuál es el estado y tendencia actual en la realización de pruebas funcionales a aplicaciones web desarrolladas con el CMS Drupal?
- 3- ¿Qué elementos deben tenerse en cuenta para llevar a cabo el análisis y diseño desde el punto de vista ingenieril del portal web para la Empresa de Ingeniería y Proyectos Azucareros (IPROYAZ)?
- 4- ¿Cómo materializar, en términos de componentes y código fuente, los elementos especificados para el portal web IPROYAZ?
- 5- ¿Qué resultados se obtendrán al aplicar el estudio comparativo entre las pruebas funcionales automatizadas que brinda Drupal y las pruebas funcionales manuales que se realizan hoy en el centro CIDI al portal web IPROYAZ?

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

1. Estudio del estado actual de la realización de pruebas funcionales a portales webs e intranets desarrolladas con Drupal.
2. Caracterización de la realización de pruebas funcionales automatizadas en aplicaciones desarrolladas con Drupal 8.
3. Completamiento del análisis y diseño del sistema a partir de la propuesta de diseño de interfaz del portal web.
4. Definición los estándares de codificación y los componentes del sistema.
5. Implementación del portal web IPROYAZ.
6. Aplicación de las pruebas funcionales manuales.
7. Aplicación las pruebas funcionales automatizadas
8. Realización del estudio comparativo.
9. Definición la estrategia a aplicar en la realización de las pruebas funcionales que disminuya el esfuerzo singular en la realización de la tarea.

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

### Teóricos:

- **Histórico-Lógico:** permitió estudiar los conceptos, elementos distintivos y características de como se procede actualmente durante el proceso de pruebas funcionales que se les aplican a los portales web. Estos elementos permitieron seleccionar las herramientas apropiadas para darle cumplimiento al objetivo general de la investigación.
- **Analítico-Sintético:** Empleado para realizar el estudio teórico de la investigación facilitando el análisis de documentos y la extracción de los elementos más importantes relacionados con el proceso de pruebas del sistema manuales y automatizadas que hacen posible la selección de criterios y características para fundamentar las conclusiones relacionadas con el objeto de estudio.

### Empíricos:

- **La Encuesta:** Se emplea para recolectar datos en la realización de un cuestionario para conocer las necesidades de los especialistas a la hora de hacer las pruebas e identificar como se hace el proceso de pruebas del sistema en el centro CIDI. Ver anexo (Encuesta)
- **La Observación:** Se emplea para obtener conocimiento acerca de la aplicación de las pruebas funcionales y detectar carencias que pueden ser corregidas para disminuir el esfuerzo.

### Estructuración de la investigación por capítulos

El presente documento está formado por tres capítulos:

#### **Capítulo 1: “Fundamentación teórica del uso de las pruebas de software.”**

Abarca los referentes teóricos que sustentan la realización de pruebas funcionales en portales web e intranets, el estado y tendencia actual en la realización de pruebas funcionales a aplicaciones web desarrolladas con el CMS Drupal. Describe las herramientas, tecnologías y metodología empleadas para el desarrollo del portal web. Brinda las conclusiones que fueron obtenidas respecto al tema durante el proceso de investigación.

#### **Capítulo 2: “Propuesta de la estrategia para las pruebas de software.”**

En este capítulo se documenta todo el proceso de elaboración de la estrategia de pruebas de software más detallada. Se describen los requerimientos de acuerdo a lo establecido en la metodología utilizada, las técnicas y enfoques para el diseño de los casos de pruebas, que permitirán hacer el diseño de las pruebas del sistema de forma manual, así como los test PHPUnit y Behat para las pruebas automatizadas.

Provee conclusiones sobre la incidencia de los elementos antes mencionados en el proceso de prueba de software.

### **Capítulo 3: “Validación de la propuesta para las pruebas de software.”**

En este capítulo se detalla la propuesta de solución al problema planteado. Se describe la organización y diseño de la estrategia de prueba del sistema y se especifican los estándares de codificación a utilizar. Además, se aplica la estrategia de pruebas definida para el portal web IPROYAZ y se emiten conclusiones sobre estos procesos.

Con la realización de la presente investigación, se pretende obtener una estrategia de pruebas funcionales para los portales web desarrollados en el centro de desarrollo CIDI.

## **Capítulo 1: “Fundamentación teórica del uso de las pruebas de software.”**

### **1.1 Introducción.**

Decir calidad de software es cumplir con las expectativas del cliente. Para asegurar la calidad del producto se realizan una serie de pruebas que revisan, supervisan y examinan la forma en que se desenvuelve el producto a la vez que se comprueba la exactitud y confiabilidad de los procesos. Es posible contar con un informe final en el que se establece si el software cumple o no con lo que espera el usuario, según el motivo por el cual fue hecho. En este capítulo se exponen los elementos relacionados con el marco teórico metodológico que respaldan la investigación. Se definen los conceptos esenciales asociados a las pruebas de software, así como los tipos de pruebas que existen. Se realiza un diagnóstico del estado y tendencia actual en la realización de pruebas funcionales a aplicaciones web desarrolladas con el CMS Drupal. Se selecciona la metodología AUP en su versión UCI y el entorno de desarrollo adecuado para la implementación en lo referente a herramientas, tecnologías y despliegue.

### **1.2 Conceptos asociados al dominio del problema de la investigación.**

#### **1.2.1 Pruebas de software.**

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su puesta en producción. Lo interesante de las pruebas es que se pueden ejecutar de manera automática, para determinar en cualquier momento si tenemos una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin que nos demos cuenta.(campusmvp, 2020)

#### **1.2.2 Estrategia de prueba de software.**

El proceso de ejecución de Pruebas de sistema debe ser considerado durante todo el ciclo de vida de un Proyecto, para así obtener un producto de alta calidad. Su éxito dependerá del seguimiento de una Estrategia de Prueba adecuada. Una estrategia de prueba integra técnicas de casos de prueba en una serie de pasos planeados y bien definidos que llevan a la construcción exitosa del software. Se describen

los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar estos pasos, y cuántos recursos se requieren.(Toll Palma & Ril Gil, s. f.)

### **1.2.2.1 Descripción de la estrategia de prueba de software.**

Cualquier estrategia de prueba debe incorporar la planeación, el diseño de casos de prueba, la ejecución y evaluación de las mismas, es decir, los datos resultados finales.

Una estrategia de prueba debe ser lo suficientemente flexible para promover la creatividad y la adecuación necesaria a los sistemas de software, y lo suficientemente rígida para promover una planeación razonable y un seguimiento administrativo a medida que el proyecto progresa.

La estrategia de pruebas debe estar dirigida a encontrar los errores más importantes con la mayor rapidez y el menor costo posible. Al involucrar más pruebas y medidas de detección, se puede definir mejor la estrategia. Las duplicidades u omisiones involuntarias que tengan lugar entre diferentes pruebas pueden ser evitadas coordinando a los probadores y las actividades de pruebas, así como fijando el alcance de la prueba.(Toll Palma & Ril Gil, s. f.)

### **1.2.3 Técnicas de diseño de pruebas.**

Las técnicas de Diseño de Pruebas están dadas por la imposibilidad de realizar la mayor cantidad de Prueba, pues se deben seguir determinados criterios para seleccionar los casos de prueba. Es por eso que el objetivo de las técnicas para el Diseño de Pruebas es garantizar con el mayor grado de confianza posible la detección de los defectos del software.

Los Casos de Prueba especifican una forma de probar el sistema, incluyendo los datos de entrada y resultados esperados. Un Caso de Prueba puede derivarse de un Caso de Uso del Modelo de Casos de Uso o de una realización de un Caso de Uso del Modelo de Diseño.(Toll Palma & Ril Gil, s. f.)

**Caso de Prueba (CP):** conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

**Caso de Uso (CU):** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.

En la práctica lo que se prueba puede estar dado por un requisito o varios requisitos del sistema cuya implementación justifica una prueba que es posible realizar y resulta poco costosa.

### 1.2.3.1 Enfoques principales para el diseño de casos de prueba.

1. El enfoque estructural o de caja blanca: este tipo de enfoque es para realizar las pruebas de unidad las cuales se le realizan al código del software.
2. El enfoque funcional o de caja negra: en este enfoque funcional las pruebas están centradas en los requisitos funcionales del software y para elegir bien un Caso de Prueba se debe tener en cuenta los siguientes requisitos:
  - ✓ Que cubra un conjunto extenso de otros casos posibles.
  - ✓ Indique algo acerca de la ausencia o la presencia de defectos en el conjunto específico de entradas que prueba, así como de otros conjuntos similares.
  - ✓ Reduzca el número de otros casos necesarios para que la prueba sea razonable.
  - ✓ Implique que el caso ejecute el máximo número de posibilidades de entrada diferentes para así reducir el total de casos.
3. El enfoque aleatorio consiste en utilizar modelos (en muchas ocasiones estadísticos) que representen las posibles entradas al programa para crear a partir de ellos los casos de prueba.

### 1.2.3.2 Diseño de las pruebas del sistema.

Las pruebas de sistema se dividen en dos tipos: las pruebas funcionales y las pruebas no funcionales.

Las pruebas funcionales se encargan de verificar la funcionalidad del sistema basado en la especificación de casos de usos o requisitos funcionales.

Las pruebas funcionales requieren el uso y la aplicación de una técnica de caja negra, siguiendo esta técnica se confeccionan los casos de pruebas para probar las condiciones de entrada tanto válidas como inválidas.

Las pruebas no funcionales son las encargadas de verificar el rendimiento, el estrés, configuración, seguridad, entre otras del sistema, los cuales conforman los requisitos no funcionales específicos que debe cumplir una aplicación determinada.

El diseño de las pruebas se debe describir para cada uno de los tipos de pruebas de sistema con los siguientes datos:

Para requisitos funcionales y no funcionales:

**Nombre de la Prueba:** Todo manual o estrategia de prueba debe llevar el nombre de la prueba que se le aplicó para poder identificar el tipo de error encontrado a la hora de recopilar en la tabla de no conformidades y así saber hacia dónde se debe enfocar el trabajo para remediar el error.

**Tipo de Prueba:** El tipo de prueba se refiere en este caso al nivel de prueba, es decir, si es prueba de unidad, si es prueba de integración, prueba de sistema o si es prueba de aceptación.

**Entrada:** En la entrada se explica el cómo se introducirán los datos que aparecen en el diccionario de datos, los cuales son necesarios solamente para las pruebas de funcionalidad del sistema, es decir, para confección de los casos de pruebas que ayudarán a verificar la funcionalidad de la aplicación en cuestión, aunque para las pruebas no funcionales pueden hacerse casos de pruebas. Todo depende de la especificación de los requisitos no funcionales del sistema. (ver Anexo 1)

**Técnica:** Del método de caja negra la técnica de particiones de equivalencia es la utilizada para diseñar los casos de prueba que se describen. (ver Anexo 2)

**Tabla de no conformidades (NC):** recopilar todos los errores encontrados, para luego corregirlos. (ver Anexo 3 para requisitos funcionales, ver Anexo 5 para requisitos no funcionales)

No es de obligatorio cumplimiento la elaboración de casos de pruebas para requisitos no funcionales, ya que los requisitos no funcionales de una aplicación determinada, valida el rendimiento, el stress, la seguridad, la disponibilidad de la red, la fiabilidad, compatibilidad, entre otras que no requieren de una condición de entrada específica, sino de explicar el procedimiento del cómo se va a llevar a cabo en la aplicación.

### **1.3 Pruebas funcionales automatizadas en aplicaciones desarrolladas con Drupal 8.**

La automatización de pruebas, es una práctica idónea que permite optimizar el tiempo de ejecución en las distintas etapas del desarrollo de software. Adicionalmente permite realizar una cobertura mayor, cuando se agregan las pruebas técnicas, como carga, volumen, estrés. Lo que hace, que este sea progresivo y funcional de acuerdo a las necesidades de cada empresa. La automatización de pruebas brinda eficiencia en la identificación de fallas, reduciendo las probabilidades de errores humanos, entre otros beneficios. Aunque hay algunas pruebas que se realizan de forma manual, como las pruebas de usabilidad y

aceptación se recomienda que las regresiones de gran volumen sean siempre automatizadas para garantizar la calidad de software.

Drupal 8 viene con varias opciones de prueba automatizadas para elegir. La mayoría de ellos son parte del marco de prueba central. Y las funcionalidades enriquecidas y la interfaz de usuario las proporciona el nuevo marco de prueba *PHPUnit*. Las pruebas se pueden realizar en diferentes capas para confirmar la calidad y la reacción del código en casos extremos:

- Pruebas unitarias.
- Pruebas de kernel.
- Pruebas funcionales.

También es posible utilizar un marco externo como Behat con escenarios en sintaxis Gherkin.

### 1.3.1 Configuración para ejecutar pruebas PHPUnit

Asegúrese de que las dependencias de Composer estén instaladas

**Si lo instaló desde un paquete de Drupal.org, tiene tres opciones:**

1. Instale Composer y ejecute la instalación de composer (o intente actualizar composer en una raíz D8 o D9 existente).

Utilice una instantánea de desarrollo (por ejemplo, 9.0.x-dev) en lugar de una versión etiquetada para su sitio de desarrollo.

Instale las dependencias de desarrollo que necesita manualmente en el directorio de proveedores de Drupal o en otro lugar.

2. Si lo instaló usando la plantilla Composer de drupal / recommended-project, las dependencias de desarrollo se pueden instalar requiriendo drupal / core-dev como una dependencia en su proyecto.
3. Si instaló usando git, instale Composer y ejecute la instalación de composer.

### Configurar PHPUnit

PHPUnit almacena la configuración en el archivo phpunit.xml. Drupal viene con una versión de muestra de este, core / phpunit.xml.dist, que debe copiar para comenzar.

El lugar donde coloque esto depende de su flujo de trabajo:

El ejecutable PHPUnit espera encontrar el archivo phpunit.xml en el directorio actual. Esto se puede anular con la opción -c.

Si está utilizando Composer para administrar el núcleo de Drupal, la actualización del núcleo sobrescribirá el núcleo / carpeta y eliminará su archivo phpunit.xml.

### En phpunit.xml realizar los siguientes cambios:

- ✓ Establezca la variable SIMPLETEST\_BASE\_URL en la URL de su sitio.
- ✓ Configure la variable SIMPLETEST\_DB para que apunte a la URL de su base de datos Drupal.
- ✓ Si está colocando phpunit.xml en otro lugar que no sea el núcleo, cambie el valor del atributo 'bootstrap' de la etiqueta phpunit para reflejar la nueva ubicación.
- ✓ Para pruebas funcionales y de kernel, configure BROWSETEST\_OUTPUT\_DIRECTORY.

El resultado debería verse así:

```
<phpunit bootstrap="tests/bootstrap.php" colors="true"
    beStrictAboutTestsThatDoNotTestAnything="true"
    beStrictAboutOutputDuringTests="true"
    beStrictAboutChangesToGlobalState="true"
    printerClass="\Drupal\Tests\Listeners\HtmlOutputPrinter">
<php>
  <!-- Set error reporting to E_ALL. -->
  <ini name="error_reporting" value="32767"/>
  <!-- Do not limit the amount of memory tests take to run. -->
  <ini name="memory_limit" value="-1"/>
  <!-- Example SIMPLETEST_BASE_URL value: http://localhost -->
```

```
<env name="SIMPLETEST_BASE_URL" value="http://drupal8.localhost"/>

<!-- Example SIMPLETEST_DB value:
mysql://username:password@localhost/databasename#table_prefix -->

<env name="SIMPLETEST_DB" value="mysql://drupal8:drupal8@localhost/drupal8"/>

<!-- Example BROWSERTEST_OUTPUT_DIRECTORY value:
/path/to/webroot/sites/simpletest/browser_output -->

<env name="BROWSERTEST_OUTPUT_DIRECTORY"
value="/var/www/sites/simpletest/browser_output"/>

</php>

</phpunit>
```

Una vez hecho todo esto, puede ejecutar cualquier prueba que desee sin tener que volver a realizar toda esta configuración.

Busque y reemplace rápidamente los valores con sed desde la línea de comando. Se supone que el archivo de configuración phpunit.xml se encuentran en el directorio principal. Ajústelo para que coincida con su configuración:

```
$ cd core
```

```
$ cp phpunit.xml.dist phpunit.xml
```

```
$ sed -i 's|name="SIMPLETEST_BASE_URL" value=""|name="SIMPLETEST_BASE_URL"
value="http://d8dev\.\ndo\.\site"|g' phpunit.xml
```

```
$ sed -i 's|name="SIMPLETEST_DB" value=""|name="SIMPLETEST_DB"
value="mysql://drupal8:drupal8@database/drupal8"|g' phpunit.xml
```

```
$ sed -i 's|name="BROWSERTEST_OUTPUT_DIRECTORY"
value=""|name="BROWSERTEST_OUTPUT_DIRECTORY" value="..\sites/simpletest/browser_output"|g'
phpunit.xml
```

### Cree un directorio para la salida HTML

Las pruebas funcionales pueden generar las páginas HTML que ve el código de ejecución de la prueba. Estos se generan como archivos HTML sin formato. Para proporcionar una ubicación en la que escribirlos, cree un directorio llamado `sites / simpletest` y asegúrese de que el servidor web pueda escribirlo. Está bien hacer que este directorio sea "escribible en todo el mundo", es decir:

Localice el binario PHPUnit

La ubicación relativa del binario PHPUnit depende de cómo instale Drupal.

- Si instaló Drupal usando un paquete y luego ejecute la actualización del compositor, el directorio del proveedor está dentro de la raíz de Drupal (adyacente al 'núcleo'). Las instrucciones asumen que esta es su configuración.
- Si instaló usando `composer require drupal / drupal`, terminará con un directorio de proveedor fuera de la raíz de Drupal (arriba de 'core'). Es posible que deba ajustar la ruta a PHPUnit en los comandos que se dan: `vendor / bin / phpunit` se convierte en `../vendor/bin/phpunit` y `../vendor/bin/phpunit` se convierte en `../../vendor/bin / phpunit`.

Dependiendo de su flujo de trabajo de desarrollo, puede resultarle útil vincular suavemente PHPUnit desde `/usr/local/bin` (`cd /usr/local/bin; ln -s /var/www/html/vendor/bin/phpunit`), o similar, o agregue `vendor / bin` a su PATH.

### 1.3.2 Ejecutando pruebas

Todos los comandos de ejemplo en este documento asumen que el archivo de configuración `phpunit.xml` están ubicados en el directorio principal:

```
cd core
```

Para ejecutar `\Drupal\Tests\datetime\Unit\Plugin\migrate\field\DateFieldTest`, por ejemplo, su comando se vería así:

```
../vendor/bin/phpunit modules / datetime / tests / src / Unit / Plugin / migrate / field / DateFieldTest.php
```

### Ejecute todas las pruebas unitarias de PHPUnit

### **Para ejecutar las pruebas unitarias en OS X, Linux u otros sistemas \* nix:**

```
../vendor/bin/phpunit --testsuite = unidad
```

Todas las pruebas, incluidas las que se encuentran en [drupalroot] / modules o [drupalroot] / sites / \* / modules, se ejecutan desde la carpeta principal con el comando ../vendor/bin/phpunit

Tenga en cuenta también que no necesita tener una instalación de Drupal en funcionamiento para ejecutar pruebas unitarias basadas en PHPUnit de esta manera. Las pruebas de PHPUnit están aisladas de Drupal y no las necesitan para ejecutarse.

### **En Windows, el enlace simbólico almacenado en vendor / bin / phpunit no funcionará. Debe usar la ruta completa al ejecutable PHPUnit:**

```
../vendor/phpunit/phpunit/phpunit
```

### **Ejecute la prueba del kernel y las pruebas del navegador**

Para las pruebas del kernel, necesita una conexión de base de datos que funcione y para las pruebas del navegador, su instalación de Drupal debe ser accesible a través de un servidor web. Sin embargo, nunca debe instalar Drupal usted mismo, ya que la prueba lo maneja automáticamente.

### **Problemas de permisos**

Las pruebas funcionales deben invocarse con un usuario del mismo grupo que el usuario del servidor web. Puede configurar Apache para que se ejecute como su propio usuario del sistema o ejecutar pruebas como un usuario privilegiado.

Para desarrollar localmente, un enfoque sencillo, pero también menos seguro, es ejecutar pruebas como su propio usuario del sistema. Para lograrlo, cambie el usuario Apache predeterminado para que se ejecute como su usuario del sistema. Normalmente, necesitaría modificar `/etc/apache2/envvars` en Linux o `/etc/apache2/httpd.conf` en Mac.

### **Ejecute una prueba específica**

Simplemente especifique el nombre del archivo, ejemplo; `../vendor/bin/phpunit tests / Drupal / Tests / Core / Password / PasswordHashingTest.php`

## Ejecutar grupos de pruebas

Para facilitar la ejecución de pruebas específicas, los autores de pruebas utilizan anotaciones para colocar sus pruebas en uno o más grupos.

Enumere todos los grupos de pruebas:

```
../vendor/bin/phpunit --list-groups
```

## Ejecute un grupo de pruebas:

```
../vendor/bin/phpunit --group Nombre de grupo
```

## Ejecute varios grupos de pruebas:

```
../vendor/bin/phpunit --group Grupo1, Grupo2
```

## Excluir un grupo de pruebas:

```
../vendor/bin/phpunit --exclude-group nombre de grupo
```

## Ejecutar un método específico

```
../vendor/bin/phpunit --filter = MyMethodTest
```

## Genere un informe de cobertura de código

```
../vendor/bin/phpunit --coverage-html / tmp / report
```

Y luego abra `/tmp/report/index.html` en su navegador para revisar.

Para obtener una descripción completa de las opciones de la línea de comandos al ejecutar pruebas, consulte *The Command-Line Test Runner de PHPUnit.* (Drupal, 2021)

### 1.3.3 Automatizando test funcionales con Behat.

Behat es una herramienta de BDD (Behaviour Driven Development) que se utiliza para comprobar el comportamiento de una aplicación desde el punto de vista de un final. Es muy popular el uso de esta herramienta para pruebas de automatización de casos, utilizando escenarios legibles para los humanos.

Para escribir los test se usa el lenguaje Gherkin, de forma que se puedan escribir los test de la forma "Teniendo en cuenta que... Entonces debería...". Se puede además extender escribiendo funciones PHP personalizadas. Se define las condiciones para confirmar el correcto funcionamiento de un software mediante pasos que no requieren conocimientos de programación para su entendimiento.

Behat ayuda a cumplir con las especificaciones y requisitos del cliente porque funciona con test que describen escenarios de posibles comportamientos del usuario en la web. Estos test pueden ser creados y mantenidos por cualquier persona, ya sea un gerente de proyecto, un desarrollador o cualquier otra parte interesada en el proyecto. Los escenarios describen la funcionalidad que queremos testear, tal y como si fuese un usuario final. (ateneatech, 2019)

Los test automatizados de Behat pueden ayudar a:

- Comprobar datos y contenido estático en una web.
- Comprobar acciones sobre botones, enlaces y campos.
- Comprobar formularios.
- Comprobar Flujos de trabajo como registros o procesos de compra.
- Comprobar que no haya regresiones en el código.

Behat no puede ayudar:

- Comprobar datos dinámicos.
- Procesos sobre imágenes.
- Códigos de respuesta de enlaces de un sitio web.

### 1.3.4 Pasos para hacer un test Behat:

Requerimientos:

- ✓ Instalación de Drupal 8.
- ✓ También se puede crear una instalación de prueba utilizando `drush qd test_behat --core=drupal-8.x -y`
- ✓ Composer.
- ✓ Drush
- 1- Instalar dependencias

Instalar un conjunto de librerías las cuales se obtendrán mediante el uso de Composer ejecutando los siguientes comandos

```
composer require behat/behat
```

```
composer require drupal/drupal-extension
```

```
composer require drupal/drupal-driver
```

- 2- Comprobar que las dependencias estén ahora disponibles en la carpeta /vendor del proyecto.
- 3- Comprobar que la instalación de Behat sea satisfactoria ejecutando vendor/behat/behat/bin/behat – version
- 4- Crear el archivo behat.yml donde se indicará la configuración del Behat para el proyecto en cuestión.

Debería verse así:

```
default:
```

```
suites:
```

```
default:
```

```
contexts:
```

- FeatureContext
- Drupal\DrupalExtension\Context\DrupalContext
- Drupal\DrupalExtension\Context\DrushContext
- Drupal\DrupalExtension\Context\MinkContext

```
extensions:
```

```
Behat\MinkExtension:
```

```
goutte: ~
```

```
base_url: http://localhost:8888/ # Reemplazar con el URL del sitio
```

*Drupal\DrupalExtension:*

*blackbox: ~*

*api\_driver: 'drupal'*

*drush:*

*alias: 'local'*

Basado en *behat.yml.dist*

- 5- Ejecutar `vendor/behat/behat/bin/behat --init` el cual creara la carpeta `features/` y creara un fichero base donde se pueden agregar pasos personalizados.
- 6- Crear el test behat.
  - 6.1 En la raíz del proyecto crear la carpeta `/ features / tests /`
  - 6.2 Crear un archivo llamado: `nombre_test.feature` donde con el lenguaje Gherkin se definen las condiciones para probar el correcto funcionamiento del software. Cada archivo define una característica específica y puede tener distintos escenarios para la misma.
- 7- Correr el test de behat mediante el uso del siguiente comando: `vendor/behat/behat/bin/behat features/tests/nombre_test.feature`

Una vez terminado de correr el test al final del mismo se brinda la información de la cantidad de escenario correcto e incorrecto que se recorrieron y en la cantidad de pasos que se hizo, además del tiempo en que se recorrió el test.

### **1.4 Tipos de pruebas.**

- 1- Prueba de Funcionalidad:

Objetivo: Verificar la función del sistema al fijar la tensión en la validación de las funciones, métodos, servicios.

Metas: Validar que la aplicación:

- Cumpla con los requisitos funcionales especificados en el diseño de la solución.
- Cumpla con los requisitos No funcionales especificados en el diseño de la solución.

- Cumpla con las restricciones de entrada y salida de la información especificada en el diccionario de Datos.
- Cumpla íntegramente con la estructura referencial especificada en el Mapa de Navegación.

Las pruebas de funcionalidad también abarcan:

### 1.2- Pruebas unitarias:

Las pruebas unitarias son las que aseguran que cada célula del código desarrollado en un componente brinde los resultados adecuados. En estas pruebas los desarrolladores observan la interfaz y la especificación de un componente, proporcionando la documentación del desarrollo del código se prueba exhaustivamente, claro que de forma independiente antes de pasar a otra unidad.

Las pruebas unitarias admiten pruebas funcionales al ejercer el código que es más probable que se rompa. Por ello, si usas pruebas funcionales sin pruebas unitarias, puedes experimentar algunas dificultades para diagnosticar pruebas fallidas.

### 1.3- Prueba de componentes:

Las pruebas de componentes se ejecutan de forma independiente para comprobar que el resultado sea el requerido. Su objetivo es verificar las funcionalidades y/o usabilidades de los componentes, aunque no solo se limite a eso.

Para ilustrarla mejor, un ejemplo de esta prueba puede ser cualquier elemento que tenga entrada y deba generar alguna salida. Puede ser el módulo de código, página web, pantallas e incluso un sistema dentro de un sistema más grande, en un componente. Aquí algunos usos de los componentes que puedes probar:

- ✓ Prueba de UI para usabilidad y accesibilidad.
- ✓ Prueba de carga para asegurar el rendimiento.
- ✓ Inyección de SQL a través de componentes de UI para asegurar la seguridad.
- ✓ Prueba de login con credenciales válidas e inválidas.

### 1.4- Prueba de humo:

Las pruebas de humo se realizan para verificar si las funcionalidades más significativas de la aplicación funcionan o no. De forma que lo más básico del software se ejecute de forma correcta con pruebas sencillas y rápidas.

Es una de las pruebas funcionales más importantes y debería ser la primera que se realice en una nueva compilación. La prueba de humo es común y aunque a veces no se tiene claro su concepto, no se trata de realizar pruebas exhaustivas sino de verificar que la funcionalidad crítica del sistema realmente funciona bien.

Si la prueba es exitosa será entonces una compilación estable. Luego se realizarán pruebas funcionales para las características o funcionalidades recién agregadas posteriormente o pruebas de regresión según la situación. Por otro lado, si esta no es estable y falla la compilación lo común es que se devuelva al equipo de desarrollo para solucionar los problemas de compilación y crear una nueva.

### 1.5- Pruebas de integración:

La prueba de integración es uno de los tipos de prueba funcional más común y se realiza de forma automatizada. Se realizan para probar componentes individuales con el objetivo de verificar cómo los módulos, que trabajan de forma individual, funcionan cuando estén integrados.

Usualmente nos ayuda a identificar problemas en las operaciones de la interfaz de usuario, formatos de datos, invocar API, acceso a bases datos, entre otras.

Algunas de las verificaciones que se realizan en las pruebas de integración son:

Prueba de interfaz: en la comprobación de las transferencias de datos entre dos componentes.

Prueba de interfaces como servicios web, API, entre otros. Se realiza para verificar que los componentes estén sincronizados entre sí. Ayudan a determinar que diferentes funciones, como la transferencia de datos entre los diferentes elementos del sistema, se realizan de acuerdo con la forma en que fueron diseñadas.

### 1.6- Pruebas de regresión:

Es normal que los desarrolladores modifiquen y mejoren las funcionalidades de su desarrollo. Por ello existe una gran posibilidad de que puedan causar 'efectos' inesperados en su comportamiento. Estas pruebas de regresión se realizan para asegurar que los cambios o adiciones no hayan alterado ni eliminado las funcionalidades existentes.

El objetivo de las pruebas de regresión es encontrar errores que puedan haber sido introducidos accidentalmente en la compilación existente y así garantizar que los errores eliminados continúen así.

### 1.7- Prueba de cordura:

Si tienes una compilación con modificaciones menores, en vez de ejecutar las pruebas de regresión, realizamos una prueba de cordura.

Con ella podemos determinar que las modificaciones realmente hayan solucionado los problemas. Y que dichas correcciones no hayan generado ningún problema. Usualmente estas pruebas son subpruebas de la de 'Regresión' ya que están relacionadas con los cambios realizados en el producto. No confundas estas las pruebas de humo con las de cordura por una simple razón:

Las pruebas de humo se inician en la compilación desde el inicio y se inspeccionan las funcionalidades más importantes. Mientras que las de cordura analizan profundamente las compilaciones de software. Es decir, las primeras confirman la estabilidad del producto, mientras que las segundas aseguran la racionalidad del producto.

### 2. Pruebas de aceptación del usuario:

Cuando ya hemos seguido e implementado las pruebas que requerimos para nuestro producto, hacemos las pruebas de aceptación. Estas hacen parte de la última fase de este proceso de testing. Aquí los usuarios reales del software lo usan para verificar que cumpla con las tareas requeridas en un ambiente 'real'. En ocasiones se realiza cuando se hace la entrega del producto "como punto de control final entre todos los tipos de pruebas funcionales".

Desde el inicio hasta la implementación, el software deberá someterse a varios tipos de pruebas. El objetivo siempre será asegurar la calidad para evitar reprocesos y garantizar las funcionalidades de la aplicación, tanto para el usuario final, como para el cliente. (*trycore*, s. f.)

### 3. Prueba de Seguridad:

Objetivo: Verificar que los mecanismos de protección incorporados en el sistema realmente lo protegerán de accesos impropios.

Metas: Validar que en la aplicación:

- Los datos y funciones del sistema solo pueden ser accesibles por los autores debidamente autorizados.
- Las funciones que atenten contra la integridad de los datos de negocios sean debidamente impedidas.

### 4. Prueba de Rendimiento o Carga:

Objetivo: Verificar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado y así como la verificación de la capacidad del sistema para manejar volúmenes de datos extremos de acuerdo a la velocidad que se especifique para el sistema.

Metas: Validar en la aplicación:

- Comprobar los tiempos de respuesta del sistema en una cantidad limitada de escenarios de trabajo (a nivel de número de usuarios y número de transacciones), bajo una configuración de hardware y software constante.
- Comprobar el tiempo de respuesta al realizar una función.
- Comprobar el tiempo de respuesta al realizar accesos concurrentes a una determinada información.
- Atender múltiples solicitudes de parte de los actores que acceden a un mismo recurso.

Las pruebas de rendimiento o carga también abarcan:

#### 4.1- Prueba de Resistencia o Stress:

Objetivo: Verificar cómo se comporta el sistema bajo condiciones anormales.

Metas: Validar en la aplicación:

- Carencia de sistemas externos con los que interactúa el sistema.
- Aplicar carga excesiva de trabajo al sistema (extremas sobrecarga).
- Hardware no disponible.
- Recursos compartidos no disponibles.
- Además, verifica qué hace el sistema cuando el usuario no hace lo que supuestamente debe hacer.

4.2- Prueba de Fiabilidad:

Objetivo: Verificar la probabilidad de que ese sistema funcione o desarrolle una cierta función, bajo condiciones fijadas y durante un período de tiempo determinado.

Metas: Validar en la aplicación:

- El sistema deberá estar disponible todo el tiempo.
- El tiempo permitido para que el sistema puede estar fuera de operación después de un fallo no debe exceder 2 días.

4.3- Prueba de Disponibilidad y Red:

Objetivo: Verificar el comportamiento de la aplicación cambiando la infraestructura de red al aplicar diferentes configuraciones, o retardos.

Metas: Validar que en la aplicación:

- No se reduzca la disponibilidad de los sistemas, debido a la actividad de alguna persona o sistema, ya sea, accidental o malintencionado.
- 

5. Prueba de Compatibilidad:

Objetivo: Verificar el funcionamiento del sistema sobre diferentes componentes de software.

Metas: Validar en la aplicación:

- Sistemas Operativos.
- Navegadores.

6. Prueba de Usabilidad:

Objetivo: Determinar si la organización de los contenidos y las funcionalidades que se ofrecen desde el Sitio Web son entendidas y utilizadas por los usuarios de manera simple y directa.

Metas: Validar en la aplicación:

- Para poder ver las páginas adecuadamente necesita utilizar un navegador compatible con estándares Web.
- Proporcionar al usuario información relacionada con el estado actual del sistema. (Toll Palma & Ril Gil, s. f.)

**1.5 Diagnóstico inicial de la situación actual de la ejecución de las pruebas funcionales.**

Los datos que a continuación se muestran reflejan los resultados de las pruebas de liberación realizadas a los proyectos de desarrollo en el trimestre octubre, noviembre y diciembre del 2019.

**Iteraciones según artefacto a liberar.**

En el caso del centro tres aplicaciones fueron liberadas en la 3ra iteración de la aplicación, dos en la segunda iteración de la documentación y una en la primera iteración de la documentación.

**No conformidades por iteraciones por tipo de error según su clasificación.**

En los productos liberados por el centro en las pruebas internas se pudo apreciar que de las NC detectadas en las aplicaciones en las tres iteraciones la mayoría fueron de correspondencia, validación y, error de idioma y funcionales. En la documentación fueron de correspondencia.

*Tabla 1NC de aplicaciones*

<b>Tipo de error</b>	<b>1ra iteración</b>	<b>2da iteración</b>	<b>3ra iteración</b>
Funcionalidad	16	11	2
Correspondencia	111	75	-
Validación	17	9	-

Error de idioma	22	11	2
Visibilidad	9	2	-
Error técnico	7	6	4
Redacción	3	4	1
Ortografía	3	3	1
Estética y diseño	1	2	-
Recomendación	1	-	1
Consistencia y estándares.	5		
Formato	2	-	-
Opciones que no funcionan	1	-	-
Total	198	123	11

Tabla 2NC de documentación.

Tipo de error	1ra iteración	2da iteración	3ra iteración
Correspondencia	8	-	-
Error técnico	1	-	-
Redacción	2	-	-
Ortografía	1	-	-
Formato	4	-	-
Total	16		

**Estado de las NC por iteraciones:**

De las 332 NC de aplicación encontradas solo 5 no procedieron y de las 16 NC encontradas en la documentación todas procedieron.

En los tres proyectos se encontraron más de 100 NC en la aplicación y en la documentación en el proyecto 3 no se encontraron NC.

Tabla 3 NC en la aplicación por iteración en cada proyecto.

Tipo de error	Proyecto 1			Proyecto 2			Proyecto 3		
	1ra	2da	3ra	1ra	2da	3ra	1ra	2da	3ra
Funcionalidad	2	2	-	4	7	-	10	2	2
Correspondencia	41	27	-	34	34	-	36	14	-
Validación	6	3	-	8	3	-	3	3	-
Error de idioma	7	2	-	2	6	-	13	3	2
Visibilidad	-	2	-	4	-	-	5	-	-
Error técnico	3	-	-	-	4	-	4	2	4
Redacción	1	-	-	-	1	-	2	3	1
Ortografía	3	2	-	-	1	-	-	-	1
Estética y diseño	1	-	-	-	-	-	-	2	-
Recomendación	-	-	-	1	-	-	-	-	-
Consistencia y estándares.	1	-	-	2	-	-	2	-	-
Formato	2	-	-	-	-	-	-	-	-
Opciones que no funcionan	-	-	-	-	-	-	1	-	-
Total por iteración	67	38	-	55	56	-	76	29	10
Total por proyecto	105			111			115		

Tabla 4 NC en la documentación por cada proyecto.

Tipo de error	Proyecto 1			Proyecto 2			Proyecto 3		
	1ra	2da	3ra	1ra	2da	3ra	1ra	2da	3ra
Correspondencia	4	-	-	4	-	-	-	-	-
Error técnico	-	-	-	1	-	-	-	-	-
Redacción	1	-	-	1	-	-	-	-	-

Ortografía	1	-	-	-	-	-	-	-	-	-
Formato	-	-	-	4	-	-	-	-	-	-
Total por iteración	6	-	-	10	-	-	-	-	-	-
Total por proyecto	6			10			-			

## 1.6 Tecnologías para el modelado y el desarrollo que serán utilizadas.

### 1.6.1 Metodología de desarrollo.

La Metodología de Desarrollo de Software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. En un proyecto de desarrollo de software la metodología ayuda a definir: quién, cuándo y cómo debe hacerlo. La metodología para el desarrollo de software un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende actividades a seguir para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Este proceso detalla un fuerte énfasis en la planificación de las actividades para el desarrollo. AUP UCI es una metodología de desarrollo de software basada en AUP y adecuada a las particularidades del proceso productivo de la UCI y utilizada en el centro de desarrollo CIDI. Esta variante propone tres fases: Inicio, Ejecución y Cierre, que guían el ciclo de desarrollo de una aplicación informática.

En la etapa de Inicio se llevan a cabo las tareas relacionadas con la planeación del proyecto. En la Ejecución se tienen en cuenta las actividades requeridas para el desarrollo, los requisitos y la arquitectura. En el Cierre se analizan los resultados del proyecto. AUP UCI define 7 disciplinas elementales: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación. (Lima, 2020)

### 1.6.2 Herramientas para el desarrollo.

Para construir la presentación *online* de la empresa que se adapte a las necesidades tanto de la organización como de los clientes es necesario dedicar grandes cantidades de tiempo y esfuerzo. Por fortuna los Sistemas de Gestión de Contenidos (*Content Management Systems* CMS por sus siglas en inglés) han tomado un auge por sus variadas herramientas y técnicas para construir la web ideal. Un CMS en su conjunto provee una buena arquitectura web, diseño y elaboración del contenido ajustada a la parte del mercado donde se proponga incidir una empresa.

Es una aplicación desarrollada para facilitar la organización y el manejo de todas las funciones necesarias para crear un proyecto digital. Simplifica la edición, estructuración, creación y montaje del contenido a un nivel muy sencillo de manejar. Con los CMS la creación de portales web es mucho más sencilla y rápida porque centraliza en lo posible todas las herramientas necesarias tanto para el lado del cliente como del lado del servidor.

Para llevar a cabo el desarrollo de portales webs e intranets solicitados por los clientes, el área del centro CIDI encargada de ofrecer este servicio, emplea el *CMS Drupal como tecnología base*. Debido a que la presente solución tiene como fin además de aumentar la visibilidad del centro de cara a Internet, mostrar a sus posibles clientes a partir de su propia imagen online en la red de redes, un ejemplo de resultado que se asemeje a los productos y servicios que se desarrollan, un elemento que constituye un requisito fundamental en la implementación de la solución, exigido así por el propio centro, es el empleo para el desarrollo del mismo ambiente tecnológico usado por ellos para desarrollar. En el siguiente apartado se describe este ambiente tecnológico.

### **CMS Drupal 8**

Drupal es un CMS o sistema de gestión de contenidos que se utiliza para crear sitios web dinámicos y con gran variedad de funcionalidades. Drupal es un software libre, escrito en PHP, que cuenta con una amplia y activa comunidad de usuarios y desarrolladores que colaboran conjuntamente en su mejora y ampliación. Esta ampliación es posible gracias a que se trata de un sistema modular con una arquitectura muy consistente, que permite que los módulos creados por cualquier desarrollador puedan interactuar con el núcleo del sistema y con los módulos creados por otros miembros de la comunidad. Con Drupal es

posible implementar una gran variedad de sitios web: un blog personal o profesional, un portal corporativo, una tienda virtual, una red social o comunidad virtual. Integra de manera global todos los elementos necesarios para el desarrollo web del lado del cliente (*HTML, CSS, JavaScript, Bootstrap*) y del lado del servidor (*PHP, MySQL/PostgreSQL*). (Buitrago, 2017)

### **HTML5**

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript). Utiliza "marcas" para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web.

"Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «World Wide Web» (Red Informática Mundial). («MDN Web Doc», s. f.)

### **CSS3**

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios. Es uno de los lenguajes base de la Open Web y posee una especificación estandarizada por parte del W3C. Anteriormente, el desarrollo de varias partes de las especificaciones de CSS era realizado de manera sincrónica, lo que permitía el versionado de las recomendaciones. Desde CSS3, el alcance de las especificaciones se incrementó de forma significativa y el progreso de los diferentes módulos de CSS comenzó a mostrar varias diferencias, lo que hizo más efectivo desarrollar y publicar recomendaciones separadas por módulos. («MDN Web Doc», s. f.)

### **Bootstrap v4.1.3**

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como Responsive Design o Diseño Adaptativo.

Este Framework te abstrae de tener que preocuparte por las media *queries* y los porcentajes en tus CSS para hacer una web Responsive, facilitando la programación del sitio. Además, se basa en la simplicidad de sus interfaces, lo cual es una tendencia del mercado, en las que tiende a diseño plano, botones grandes, etc... para facilitar la usabilidad en los dispositivos más pequeño.(«Instituto de nuevas tecnologías», 2016)

### **Twig v2.0**

Twig es un motor de creación de plantillas para utilizar con PHP. Se ocupa de brindar una solución al tratamiento de las cuestiones visuales alrededor de una aplicación desarrollada. Usar Twig para definir el layout de un sitio web hace que la experiencia de implementar un diseño sea mucho más limpio y menos engorroso. Permite separar por completo el diseño de la lógica, evitando complejizar por demás el desarrollo de la propia plataforma. Al independizar ambas capas (la de diseño y la de lógica) también brinda flexibilidad absoluta en relación con la imagen del sitio.(Esper, 2020)

### **PHP v7**

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo". Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.(php, s. f.)

### **Gestor de bases de datos MySQL v5.0.15**

MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle. Se utilizan actualmente para dar forma y facilitar la comunicación entre webs y servidores. Una de las principales características de MySQL es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente. MySQL sirve para almacenar toda la información que se desee en bases de datos relacionales, como también para administrar todos estos datos sin apenas complicaciones gracias a su interfaz visual y a todas las opciones y herramientas de las que dispone. Es algo esencial, sobre todo en webs que cuentan con la opción de registrar usuarios para que inicien sesión. (*NeoAttack*, s. f.)

### **Apache v2.2**

Apache es un servidor web HTTP de código abierto. Está desarrollado y mantenido por una comunidad de usuarios en torno a la Apache Software Foundation. La funcionalidad principal de este servicio web es servir a los usuarios todos los ficheros necesarios para visualizar la web. Las solicitudes de los usuarios se hacen normalmente mediante un navegador (Chrome, Firefox, Safari, etc.). Apache tiene una estructura basada en módulos, que permite activar y desactivar funcionalidades adicionales, por ejemplo, módulos de seguridad como `mod_security`, módulos de caché como Varnish, o de personalización de cabeceras como `mod_headers`. También permite ajustar los parámetros de PHP de tu hosting de forma personalizada mediante el fichero `.htaccess`.

Las principales ventajas de usar este el servicio web son las siguientes:

- ✓ De código abierto y gratuito, con una gran comunidad de usuarios.
- ✓ Parches de seguridad regulares y actualizados con frecuencia.
- ✓ Estructura basada en módulos.
- ✓ Multiplataforma. Está disponible en servidores Windows y Linux.
- ✓ Personalización mediante `.htaccess` independiente en cada hosting.
- ✓ Compatible con los principales CMS y tiendas online y plataformas e-learning. (*dinahosting*, s. f.)

### **phpMyAdmin v4.9.7**

Es una herramienta de software libre escrito en *PHP*, cuya intención es facilitar la administración del servidor *MySQL* a través de la web. Las operaciones más frecuentes del servidor (gestión de bases de

datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc.) se realizan a través de la interfaz de usuario, pero también ofrece la capacidad de ejecutar directamente cualquier sentencia SQL para las operaciones de mantenimiento y consulta de datos.

El gestor de base de datos *phpMyAdmin* puede administrar un servidor *MySQL* completo con todas sus bases de datos (se necesita una cuenta de súper-usuario o *root*), así como una base de datos única. Para este último caso se necesita configurar un usuario de *MySQL* que pueda leer / escribir sólo sobre la base de datos deseada. Viene con una amplia gama de documentación, dispone de páginas wiki donde se comparten ideas y tutoriales para la realización de diversas operaciones. Para facilitar el uso de una amplia gama de personas, está siendo traducido a 72 idiomas y es compatible con los idiomas LTR y RTL. (Lima, 2020)

Entre las operaciones que se pueden realizar con *phpMyAdmin* se encuentran:

- Usar y borrar bases de datos, tablas, vistas, columnas e índices.
- Mostrar múltiples conjuntos de resultados a través de los procedimientos o consultas almacenadas.
- Crear, copiar, borrar, renombrar y modificar bases de datos, tablas, columnas e índices.
- Realizar labores de mantenimiento del servidor, bases de datos y tablas, dando consejos sobre la configuración del servidor.
- Ejecutar, editar y marcar cualquier sentencia SQL, incluyendo consultas en lote.
- Cargar tablas con el contenido de archivos de texto.

### 1.6.3 Herramientas para pruebas

#### **Jmeter v4.2**

JMeter fue diseñado para realizar pruebas de carga en servidores o aplicativos Web por medio del protocolo HTTP, pero debido a su gran popularidad, se expandió para incluir otros protocolos de comunicación. A pesar de que JMeter es una herramienta diseñada para generar carga, también se puede medir el rendimiento, por medio de la colaboración de herramientas externas tales como APM (Application Performance Monitor) o aplicaciones de monitoreo de rendimiento, esta combinación puede ser sumamente importante para establecer y configurar alertas o umbrales críticos de consumo. (Jmeter, s. f.)

#### **Acunetix Web Vulnerability Scanner v13**

Se emplea para escanear el portal web para así identificar vulnerabilidades del mismo. *Acunetix Web Vulnerability Scanner* es una herramienta de seguridad de aplicaciones Web automatizada. *Acunetix WVS* es capaz de escanear cualquier sitio Web o aplicación Web que es accesible a través del protocolo HTTP / HTTPS. Sin embargo, no todas las pruebas se pueden realizar de forma automática, y por lo tanto *Acunetix WVS* proporciona herramientas de penetración manuales para pruebas particulares. (acunetix, s. f.)

### 1.6.4 Lenguaje y herramienta para el modelado.

#### Lenguaje de Modelado Unificado (UML)

Es un lenguaje para visualizar, especificar, construir y documentar un sistema. Está respaldado por el *OMG* (del inglés, *Object Management Group*). Se puede aplicar en el desarrollo de *software* con gran variedad de formas para dar soporte a una metodología de desarrollo de *software*. Un modelo UML está compuesto por 3 clases de bloques de construcción: los elementos (representaciones abstractas de cosas reales o ficticias), las relaciones y los diagramas; los cuales muestran diferentes aspectos de las entidades representadas. («uml», s. f.) El lenguaje UML se utilizará en todo el proceso de elaboración de diagramas de ingeniería de *software*.

#### *Visual Paradigm v8*

Es una herramienta *CASE*, multiplataforma. Utiliza modelado UML para la representación de las etapas por las que transita un producto de *software*. Además, propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. (Pressman, s. f.) La herramienta *Visual Paradigm* se utilizará en la elaboración de diagramas de ingeniería de *software*.

### 1.7 Conclusiones.

- La definición del marco teórico aportó todos los elementos teóricos que sustentan la solución del problema.
- El diagnóstico inicial de la situación actual de la ejecución de las pruebas funcionales permitió identificar la cantidad de iteraciones y no conformidades que se encontraron durante el proceso de prueba de tres proyectos durante un trimestre demostrando el esfuerzo que se dedica actualmente en esta tarea.

- Establecer la metodología AUP-UCI como guía para el proceso de desarrollo y la base tecnológica a emplear en la solución que se propone; seleccionándose para el desarrollo, Drupal 8 como CMS y PHP 7 como lenguaje de programación y Visual Paradigm para el modelado. Como sistema gestor de bases de datos se escogió MySQL 5.05.15 por ser un sistema seguro, libre y potente, para la gestión de la base de datos del portal web.

## Capítulo 2

### 2.1 Introducción

En el presente capítulo se describe una propuesta de estrategia a seguir para aplicar las pruebas funcionales al portal web IPROYAZ una vez este sea implementado; dicha estrategia se encuentra definida en dos partes una la que se seguirá para realizar las pruebas de forma manual y otra para las pruebas de forma automatizadas. El objetivo del capítulo es presentar los componentes que conforman la solución propuesta y explicar su funcionamiento y relación. Se presentan los requisitos funcionales y no funcionales, las historias de usuarios y los artefactos que servirán de base para la implementación.

### 2.2 Propuesta de solución

Utilizando la información recopilada en el capítulo preliminar se plantea definir una estrategia para la aplicación de las pruebas funcionales al portal web IPROYAZ de forma manual y otra de forma automatizada aprovechando las ventajas que brinda Drupal 8 una vez este sea implementado. En el caso de las pruebas manuales, para el diseño de casos de pruebas se utilizó la técnica de caja negra porque permite probar escenarios de entradas tanto válidas como inválidas, para poder verificar la funcionalidad del sistema basado en este caso en la especificación de los requisitos funcionales. Para las pruebas automatizadas se utilizarán los test PHPUnit y los test Behat.

El portal web a implementar cuenta con varias secciones en la que se presenta de forma organizada los contenidos. Algunos de estos contenidos los usuarios podrán comentarlos. Dispondrá de un buscador que permite buscar por letras, palabras o frases, obteniendo los resultados de forma inmediata. Tendrá un filtro donde se pueden filtrar contenidos en dependencia de la categoría de estos. Brindará un foro en el cual se podrá debatir. También pondrá a disposición de los usuarios multimedia, proyectos y servicios de la entidad, así como información de los directivos además de sus correos y números de teléfonos. Algunos usuarios podrán publicar noticias.

### 2.3 Estructura del portal:

El portal web IPROYAZ estará conformado por un conjunto de páginas dinámicas que se irán incluyendo a medida que se vaya creando contenido en el portal. El acceso a las diferentes vistas del portal lo realizarán los usuarios, a los que se les asignarán roles que previamente tendrán permisos asignados.

Cada página quedará construida por una cabecera que será el menú principal. Poseerá una región para mostrar el contenido principal de la página que se está visitando y un *footer* que contendrá los principales enlaces de interés. Además, contará con una página inicial (*home*) que estará dividida en la cabecera, por un logo y los enlaces a las redes sociales. A continuación, presentará un menú principal que se estará viendo en todas las páginas, una región donde se mostrará los contenidos añadidos, una foto estática, continuado otra región dividida en seis bloques *IPROYAZ*, servicios que prestamos, nuestros proyectos, noticias, multimedia, contacta con nosotros. Por último, un *footer* que estará divididos en dos bloques, *enlace de interés* en uno y en el otro: preguntas frecuentes, mapa del sitio, políticas de privacidad, términos y condiciones de uso.

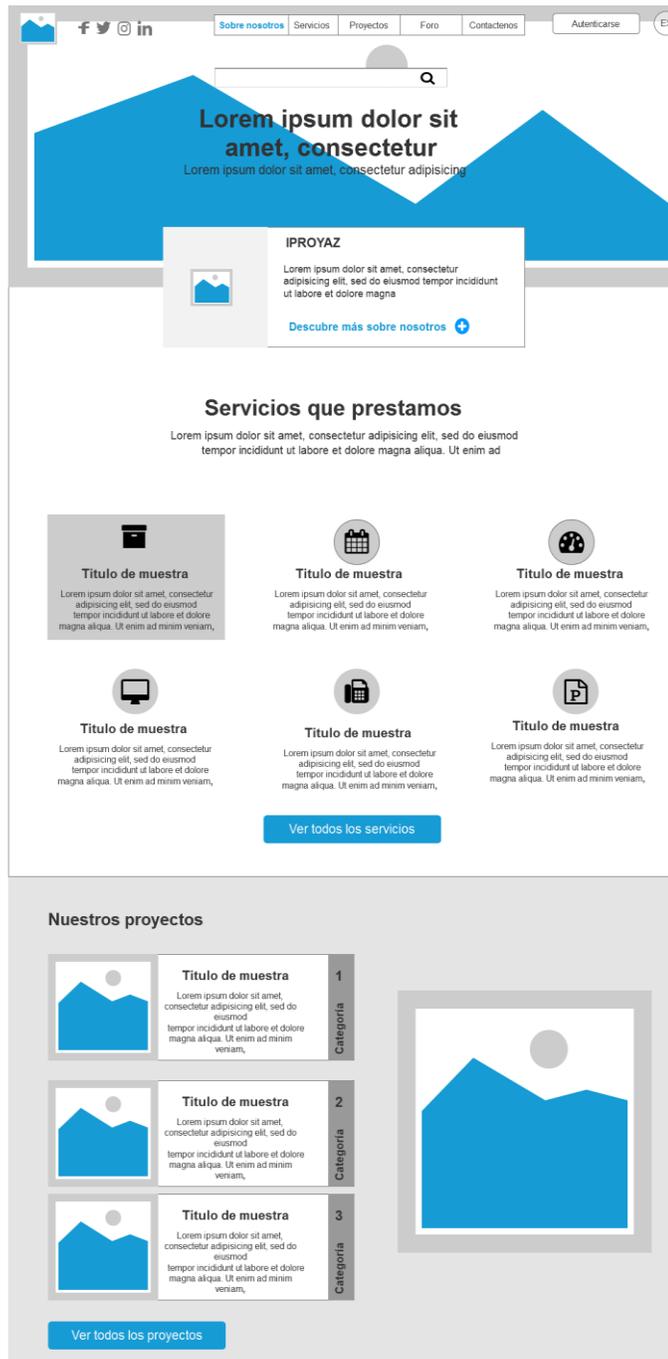


Figura: 1 Arquitectura de Información para la página de 1er nivel. Ejemplo: Home sección 1

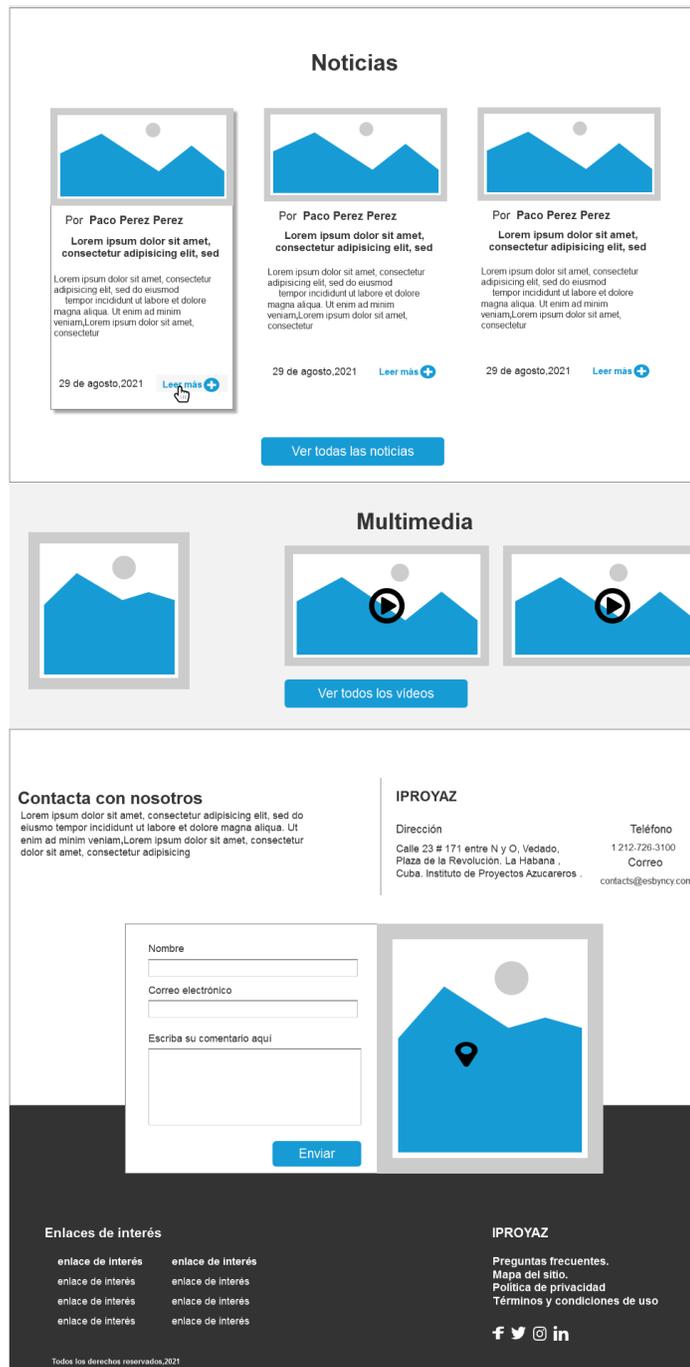


Figura 1.1 Arquitectura de Información para la página de 1er nivel. Ejemplo: Home sección 2



Figura: 2 Arquitectura de Información para la página de 1er nivel. Ejemplo: Home sección 2

## 2.4 Especificación de los requisitos.

Los requisitos son la base para la actividad de la estimación de software, ya que proporciona la necesaria abstracción del problema para que los ingenieros de software puedan realizar un análisis adecuado del problema. Los requisitos deben ser identificados y gestionados en todo el proyecto con el fin de reflejar las necesidades reales de los clientes. (Cardozzo, 2016)

### 2.4.1 Requisitos funcionales

Los requisitos funcionales para un sistema describen lo que debe hacer el sistema. Estos requisitos dependen del tipo de software que se está desarrollando, los usuarios esperados del software y el enfoque general adoptado por la organización al redactar los requisitos. Cuando se expresan como

requisitos del usuario, los requisitos funcionales deben estar escritos en lenguaje natural para que los usuarios y administradores del sistema puedan entenderlos. Los requisitos funcionales del sistema amplían los requisitos del usuario y están escritos para desarrolladores de sistemas. Deben describir las funciones del sistema, sus entradas y salidas, y las excepciones en detalle.(Sommerville, 2011)

*Tabla5 Requisitos funcionales*

<b>No.</b>	<b>Requerimiento</b>	<b>Prioridad</b>
RF1	Autenticar usuario	Alta
RF2	Crear usuario	Alta
RF3	Editar usuario	Media
RF4	Mostrar datos de usuario	Baja
RF5	Mostrar listado de usuarios	Baja
RF6	Filtrar usuario	Baja
RF7	Cancelar cuenta de usuario	Baja
RF8	Bloquear cuenta de usuario	Baja
RF9	Desbloquear cuenta de usuario	Baja
RF10	Cambiar contraseña	Baja
RF11	Eliminar contenidos	Baja
RF12	Mostrar listado de contenidos	Baja
RF13	Filtrar contenidos	Baja
RF14	Publicar contenidos	Baja
RF15	Retirar contenido de la publicación	Baja
RF16	Crear servicio	Alta
RF17	Editar servicio	Media
RF18	Mostrar servicio	Alta
RF19	Editar páginas básicas	Media
RF20	Mostrar página básica	Media
RF21	Crear noticia	Alta
RF22	Editar noticia	Media
RF23	Mostrar noticia	Alta
RF24	Mostrar enlace de interés	Alta
RF25	Crear banner promocional	Alta
RF26	Editar banner promocional	Media
RF27	Mostrar banner promocional	Alta
RF28	Crear pregunta frecuente	Alta
RF29	Editar pregunta frecuente	Media
RF30	Mostrar pregunta frecuente	Baja
RF31	Crear proyecto	Alta
RF32	Editar proyecto	Media
RF33	Mostrar proyecto	Alta
RF34	Crear marco legal	Alta

RF35	Editar marco legal	Media
RF36	Mostrar marco legal	Alta
RF37	Crear directivo	Alta
RF38	Editar directivo	Media
RF39	Mostrar directivo	Alta
RF40	Crear vídeo	Alta
RF41	Editar vídeo	Media
RF42	Mostrar vídeo	Alta
RF43	Crear foro	Alta
RF44	Editar foro	Alta
RF45	Mostrar foro	Alta
RF46	Crear Oficina	Alta
RF47	Editar Oficina	Alta
RF48	Mostrar Oficina	Alta
RF49	Cambiar idioma	Baja
RF50	Crear traducción de contenido	Alta
RF51	Editar traducción de contenido	Media
RF52	Mostrar traducción de contenido	Alta
RF53	Crear comentario	Alta
RF54	Editar comentario	Alta
RF55	Mostrar comentario	Baja
RF56	Eliminar comentario	Baja
RF57	Publicar comentario	Baja
RF58	Retirar los comentarios seleccionados de la publicación.	Baja
RF59	Mostrar listados de comentarios publicados.	Baja
RF60	Mostrar listados de comentarios sin aprobar	Baja
RF61	Crear términos de taxonomía	Alta
RF62	Editar término de taxonomía	Media
RF63	Mostrar término de taxonomía	Baja
RF64	Eliminar término de taxonomía	Baja
RF65	Realizar búsqueda simple	Baja
RF66	Realizar búsqueda avanzada	Baja
RF67	Contactar a administradores del sistema	Baja
RF68	Mostrar información del contacto	Baja
RF69	Mostrar ubicación geográfica	Baja
RF70	Mostrar enlaces de redes sociales	Baja
RF71	Compartir contenido en las redes sociales	Baja
RF72	Mostrar mapa del sitio.	Baja

### 2.4.2 Requisitos no funcionales

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

#### RNF 1: Usabilidad

**RNF 1.1:** El sistema debe presentar un diseño coherente de la web y sus enlaces y ser intuitiva para el usuario.

**RNF 1.2:** El sistema debe tener visibilidad en todas las versiones de los principales navegadores web como *Chrome v5, Firefox v4, Safari v4.0.5*.

**RNF 1.3:** El sistema podrá ser visualizado en dispositivos desde las resoluciones 320x480, 768x1024, 1024x980 y 1325x980.

#### RNF 2: Eficiencia

**RNF 2.1:** El tiempo de demora del sistema en cada transición debe ser menor de cinco (5) segundos aproximadamente.

**RNF 2.2:** El sistema debe permitir que 100 usuarios interactúen con él de manera simultánea

#### RNF 3: *Hardware*

**RNF 3.1:** El servidor de base de datos debe poseer una capacidad mínima de 80 GB.

**RNF 3.2:** El servidor de aplicaciones web debe poseer una capacidad mínima de 20 GB.

**RNF 3.3:** Los servidores web y de base de datos deben poseer como mínimo 1 GB de memoria RAM.

**RNF 4: Seguridad**

**RNF 4.1:** En caso de que el sistema presente alguna falla, los errores deben mostrar la menor cantidad de detalles posible, de forma tal, que se evite dar información que comprometa la seguridad e integridad del sistema. Sólo se mostrarán detalles ampliados del error a usuarios con privilegios de administración.

**RNF 4.2:** Se asignarán los permisos de acceso, escritura, lectura en dependencia del rol que desempeñe cada usuario del sistema.

**RNF 4.3:** Se podrá acceder a las páginas de administración del portal web a través del protocolo HTTPS, y a las páginas de usuarios por el protocolo HTTP.

**RNF 4.4:** Se garantizará la integridad de la información mediante mecanismos de control de acceso utilizando usuarios, contraseñas y niveles de accesos para cada usuario, de manera que cada uno pueda tener disponible solamente las opciones que se encuentran en correspondencia con su actividad.

**RNF 5: Software**

**RNF 5.1:** Servidor web Apache en su versión 2.2.

**RNF 5.2:** Servidor de base de datos MySQL en su versión 5.0.15 o superior.

**RNF 5.3:** Lenguaje de programación PHP 7 y como sistema de gestión de contenidos Drupal 8.6

**RNF 6: Legales**

**RNF 6.1:** Uso de licencia GNU/GPL para el CMS Drupal.

**RNF 6.2:** Uso de licencia BSD de MySQL.

**RNF 6.3:** Uso de licencia PHP *License*.

**2.5 Descripción de requisitos de software (Historias de Usuario)**

*Tabla 9 Crear servicio*

--

<b>Número:</b> 16	<b>Nombre del requisito:</b> Crear Servicio
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgo	<b>Tiempo Real:</b>
<p><b>Descripción:</b></p> <p>El sistema permite a los usuarios con rol webmaster o editor <b>crear</b> contenidos de tipo servicio en el sistema, llenando los siguientes campos:</p> <ul style="list-style-type: none"> <li>• <b>Título:</b> Campo de texto, obligatorio. Admite todos los caracteres. Longitud máxima 255 caracteres.</li> <li>• <b>Categoría:</b> Lista de selección, obligatorio. Valores (Términos de la taxonomía Categoría de servicio).</li> <li>• <b>Imagen:</b> Obligatorio. Campo de tipo archivo. Tamaño máximo 512 KB. Permite solo los archivos con las extensiones png, jpg, jpeg.</li> <li>• <b>Título de la imagen:</b> Opcional. Campo de texto. Permite todos los caracteres. Longitud máxima 1024. Este campo debe mostrarse al hacer clic en el botón Subir al servidor.</li> <li>• <b>Texto alternativo:</b> Campo de texto, opcional. Admite todos los caracteres. Este campo se muestra cuando se presiona el botón Subir al servidor.</li> <li>• <b>Descripción:</b> Área de texto, obligatorio. Admite todos los caracteres.</li> <li>• <b>Publicado:</b> Opcional. Campo de selección. Valor por defecto (marcado).</li> </ul>	
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. Si el usuario introduce los datos de forma correcta, el sistema emitirá un mensaje confirmando la culminación exitosa de la acción.</li> <li>2. Si el usuario introduce los datos de forma incorrecta, el sistema emitirá un mensaje notificando el error.</li> <li>3. Si el usuario deja campos obligatorios vacíos, el sistema emitirá un mensaje notificando el error.</li> </ol>	
<p><b>Prototipo elemental de interfaz gráfica de usuario:</b> No aplica.</p>	

Tabla 10 Editar servicio

<b>Número:</b> 17	<b>Nombre del requisito:</b> Editar Servicio
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgo	<b>Tiempo Real:</b>
<p><b>Descripción:</b></p> <p>El sistema permite a los usuarios con rol webmaster o editor <b>editar</b> contenidos de tipo servicio existentes en el sistema, modificando todos sus campos:</p> <ul style="list-style-type: none"> <li>• <b>Título:</b> Campo de texto, obligatorio. Admite todos los caracteres. Longitud máxima 255 caracteres.</li> <li>• <b>Categoría:</b> Lista de selección, obligatorio. Valores (Términos de la taxonomía Categoría de servicio).</li> <li>• <b>Imagen:</b> Obligatorio. Campo de tipo archivo. Tamaño máximo 512 KB. Permite solo los archivos con las extensiones png, jpg, jpeg.</li> <li>• <b>Título de la imagen:</b> Opcional. Campo de texto. Permite todos los caracteres. Longitud máxima 1024. Este campo debe mostrarse al hacer clic en el botón Subir al servidor.</li> <li>• <b>Texto alternativo:</b> Campo de texto, opcional. Admite todos los caracteres. Este campo se muestra cuando se presiona el botón Subir al servidor.</li> <li>• <b>Descripción:</b> Área de texto, opcional. Admite todos los caracteres.</li> <li>• <b>Publicado:</b> Opcional. Campo de selección. Valor por defecto (marcado).</li> </ul>	
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. Si el usuario introduce los datos de forma correcta, el sistema emitirá un mensaje confirmando la culminación exitosa de la acción.</li> <li>2. Si el usuario introduce los datos de forma incorrecta, el sistema emitirá un mensaje notificando el error.</li> <li>3. Si el usuario deja campos obligatorios vacíos, el sistema emitirá un mensaje notificando el error.</li> </ol>	

**Prototipo elemental de interfaz gráfica de usuario:** No aplica.

Tabla 11 Mostrar Servicio

<b>Número:</b> 18	<b>Nombre del requisito:</b> Mostrar Servicio
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgo	<b>Tiempo Real:</b>
<p><b>Descripción:</b></p> <p>Los usuarios autenticados pueden visualizar los servicios en el módulo Administrar/Contenido de la interfaz de administración. Además, al hacer clic en la sección Servicios, del menú principal ubicado en el encabezado. Además de la vía anterior puede hacer clic en la sección Servicios que prestamos, ubicado en la página principal. De estos se muestran las categorías, se selecciona la deseada y el sistema muestra una ventana emergente con los servicios de esa categoría.</p>	
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. El sistema debe mostrar todos los servicios al seleccionar el enlace Ver todos los servicios.</li> </ol>	
<p><b>Prototipo elemental de interfaz gráfica de usuario:</b></p> <p>Arquitectura de Información/ sección Servicio</p>	

## 2.6 Modelado de diseño

### 2.6.1 Diagramas de clases del diseño con estereotipos web (DCD)

Un diagrama de clases del diseño con estereotipos *web* tiene el mismo objetivo o propósito que un diagrama de clases tradicional, con la particularidad de que se emplea para el modelado de aplicaciones *web*. (Pressman, s. f.). A continuación, se muestra el diagrama de clases del diseño para historia de usuario gestionar servicio.

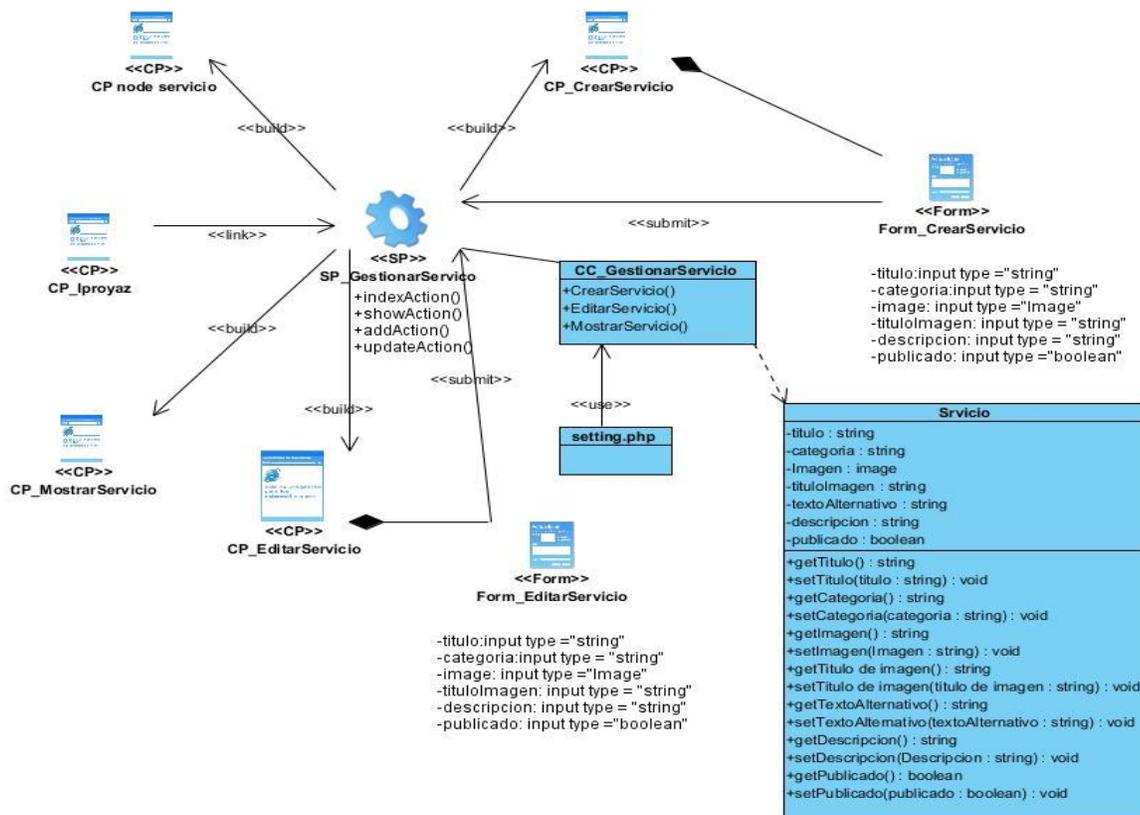


Figura 1 Diagrama de clases del diseño para historia de usuario gestionar servicio

Para gestionar servicio la página servidora SP\_GestionarServicio construye las páginas clientes CP\_MostrarServicio, CP\_EditarServicio, CP\_CrearServicio, las cuales son las encargadas de mostrar, crear y editar y los servicios, las dos primeras integradas por un formulario donde se insertarán los datos requeridos, seguidamente se realiza un *submit* que dirige a la SP\_GestionarServicio. El acceso a los datos se realiza por medio de la clase controladora CC\_GestionarServicio.

### 2.6.2 Diagramas de Secuencia

El Diagrama de Secuencia tiene como objetivo describir como colaboran los distintos objetos entre sí para conseguir un objetivo a lo largo del tiempo. Está directamente relacionado con el Diagrama de Comunicación ya que el objetivo es el mismo, pero tiene la particularidad de estar obligatoriamente ordenado en el tiempo.(Olivieri, s. f.). A continuación, se muestra el diagrama de secuencia para historia de usuario gestionar servicio.

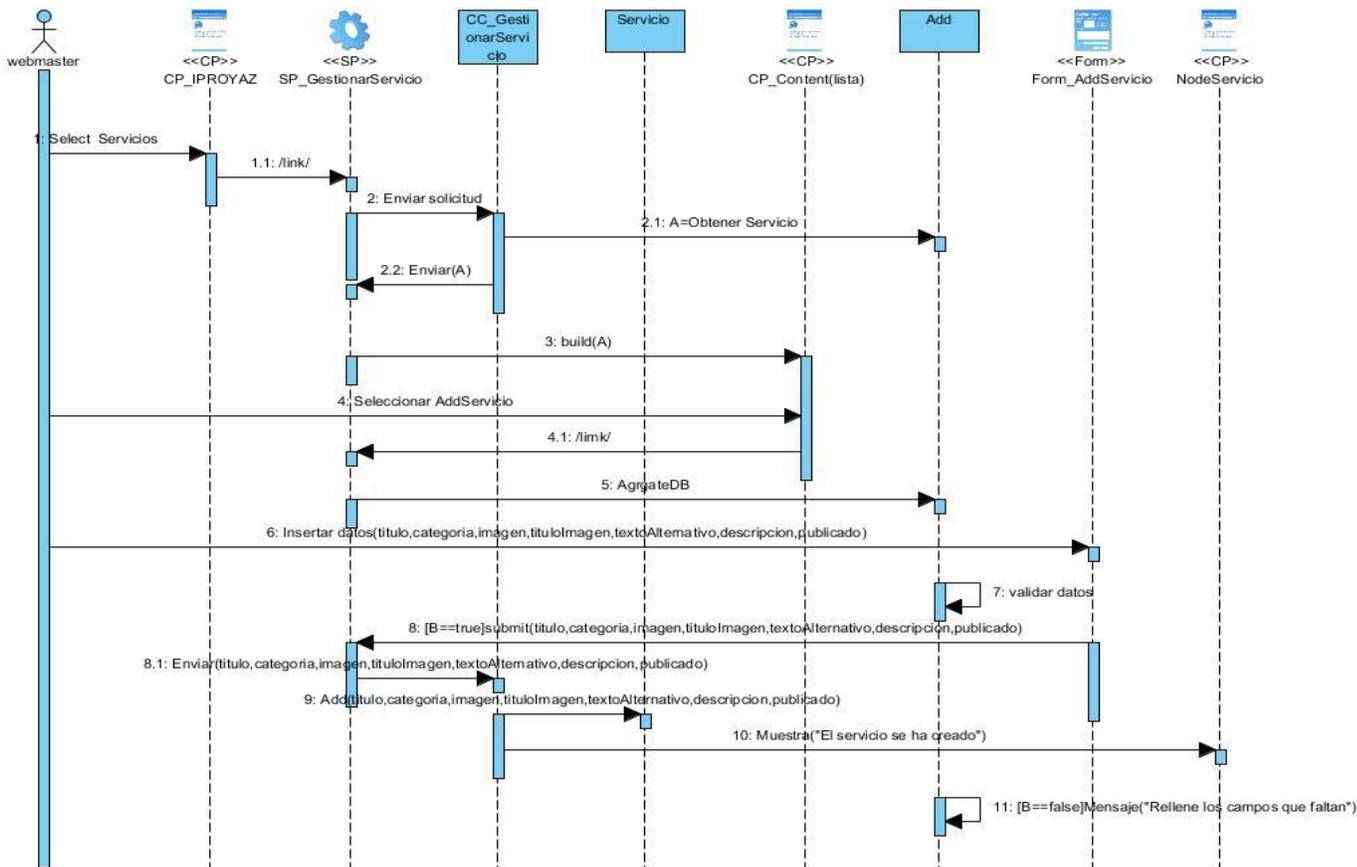


Figura 2 Diagrama de secuencia para historia de usuario adicionar servicio

Para crear un artículo el webmaster realiza un link desde la página cliente CP\_IPROYAZ a la página servidora SP\_GestionarServicio la cual envía la solicitud a la clase controladora CC\_AddServicio y esta obtiene campos a rellenar enviándoselo a la SP\_GestionarServicio y este al seguido lo coloca en la CP\_Content(lista) donde este manda el artículo para SP\_GestionarServicio al ser seleccionado por el webmaster y este inserta los datos en el formulario donde estos se validan y se actualizan. Si todos los datos están bien hacen un *submit* a la SP\_GestionarServicio y este los envía a la CC\_AddServicio donde este adiciona los campos y muestra un mensaje donde dice “El servicio se ha creado” y si falta algún campo obligatorio por rellenar muestra un mensaje donde dice “Rellene los campos que faltan”

## 2.7 Estrategia para aplicar las pruebas funcionales al portal.

### 2.7.1 Estrategia para las pruebas funcionales manualmente.

La Estrategia de Pruebas de Software propuesta está compuesta por las siguientes actividades:

1-Planificación de las pruebas.

Las pruebas que se realizarán serán funcionales donde se comprobará el correcto funcionamiento del software de acuerdo a los requisitos funcionales obtenidos.

2- Diseño de las pruebas.

En esta actividad se diseñan los casos de pruebas a utilizar.

**Casos de pruebas:**

**Crear servicios:**

Descripción general

El sistema permite crear servicios.

Condiciones de ejecución

El sistema debe estar disponible y el usuario con privilegios de administración debe estar autenticado en el sistema.

Escenario	Descripción	Título	Descripción	Imagen	Título de la imagen	Texto alternativo	Categoría	Publicado	Respuesta del sistema	Flujo central
EC 16.1 Crear servicio de forma correcta.	El sistema crea servicios de forma correcta.	V Título del servicio	V Descripción del servicio	V Subir una imagen que cumpla con las restricciones	V Título de la Imagen	V Este es el texto alternativo de la imagen	V ServicioX	V marcado	El sistema almacena la información y muestra un mensaje de confirmación de la acción: <b>"Contenido Tipo-Contenido Título ha sido"</b>	1.-El usuario accede al bloque superior <b>Administrar/Contenido/ +Añadir contenido o Atajos/ Agregar contenido</b> y el sistema muestra todos los tipos de contenidos que pueden ser añadidos. 2.-El usuario selecciona el tipo de contenido Servicio. 3.-El sistema muestra un formulario para que el usuario introduzca la información. 4.- El usuario introduce la información y presiona el botón: <b>"Guardar"</b> .
EC 16.2 Crear servicio de forma incorrecta.	El sistema no crea servicios de forma incorrecta.	NA Título del servicio	NA Descripción del servicio	I Subir un archivo que no cumpla con las restricciones de tamaño I Subir un archivo que no cumpla con las restricciones de formato I Subir un archivo con formato	NA	NA	NA	NA	El sistema no almacena la información. <b>"El archivo especificado TÍTULO-IMAGEN no se pudo subir al servidor. El archivo ocupa TAMAÑO-IMAGEN"</b> <b>"El archivo especificado TÍTULO-IMAGEN no se pudo subir al servidor. Solo se permiten archivos con las siguientes extensiones: png, jpg, gif"</b> <b>"El archivo especificado TÍTULO-IMAGEN no se pudo subir al servidor. El sistema no almacena la información y muestra los siguientes mensajes de error: "</b>	
EC 16.3 Crear servicio dejando campos	El sistema no crea servicios dejando campos	I	I	I	NA	NA	I	NA	El sistema no almacena la información y muestra los siguientes mensajes de error: "	
EC 16.4 Cancelar operación	El sistema no realiza ninguna operación	NA	NA	NA	NA	NA	NA	NA	No realiza ninguna operación y regresa a la página anterior.	Para cancelar la operación de crear el contenido, el usuario sigue el flujo de acción en el escenario

Figura: 3 CP Crear servicio

## Editar Servicios

### Descripción general

El sistema permite editar servicios.

### Condiciones de ejecución

El sistema debe estar disponible y el usuario con privilegios de administración debe estar autenticado en el sistema. Además, debe existir algún contenido de tipo Servicio insertado en el sistema. En caso de que existan traducciones de este contenido el botón Guardar del formulario para editar se comportará como Guardar (esta traducción).

Escenario	Descripción	Título	Descripción	Imagen	Título de la imagen	Texto alternativo	Categoría	Publicado	Respuesta del sistema	Flujo central
EC 17.1 Editar servicio de forma correcta.	El sistema edita servicios de forma correcta.	V	V	V	V	V	V	V	El sistema almacena la información y muestra un mensaje de confirmación de la acción: <b>"Contenido Tipo-Contenido Título ha sido actualizado."</b>	1-El usuario se autentica en el sistema y accede a la opción <b>Editar</b> del contenido por una de las siguientes vías: Vía 1 a) Accede en la página principal al módulo <b>Administrar/ Contenido</b> ubicado en la barra de administración. b) Se muestra una página con el listado de todos los contenidos existentes en el sistema. c) Selecciona de entre los contenidos de este tipo, la opción <b>"Editar"</b> del que desea editar, que aparece a la derecha del mismo. (Si el usuario desea que solo se le muestre el contenido específico que quiere editar, puede realizar un filtrado de la información, ver CPT3 Filtrar contenido). Vía 2 a) Accede en la página principal al módulo <b>Administrar/ Contenido</b> ubicado en la barra de administración. b) Se muestra una página con el listado de todos los contenidos existentes en el sistema. c) Da clic en el título del contenido de este tipo que quiere editar. d) El sistema muestra una página con toda la información del contenido.
EC 17.2 Editar servicio de forma incorrecta.	El sistema no edita servicios de forma incorrecta.	NA	NA	I	NA	NA	NA	NA	El sistema no almacena la información. <b>"El archivo especificado TÍTULO-IMAGEN no se pudo subir al servidor. El archivo ocupa TAMAÑO-IMAGEN excediendo el límite de tamaño."</b> <b>"El archivo especificado TÍTULO-IMAGEN no se pudo subir al servidor. Solo se permiten archivos con las siguientes extensiones: png, jpg."</b>	
EC 17.3 Editar servicio dejando campos vacíos.	El sistema no edita servicios dejando campos vacíos.	I	I	I	NA	NA	I	NA	El sistema no almacena la información y muestra los siguientes mensajes de error: <b>"El campo Título es obligatorio"</b>	
EC 17.4 Cancelar operación.	El sistema no realiza ninguna operación.	NA	NA	NA	NA	NA	NA	NA	No realiza ninguna operación y regresa a la página anterior.	Para cancelar la operación de editar el contenido, el usuario elige alguna de las vías descritas en el escenario anterior y presiona el botón <b>Cancelar</b> .

Figura: 4 CP Editar servicio

## Mostrar servicio

### Descripción general

El sistema muestra los datos de los servicios.

Condiciones de ejecución.

El sistema debe estar disponible. No necesariamente el usuario debe estar autenticado el sistema.

Tabla 5CP Mostrar servicio

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 18.1 Mostrar servicio.	El sistema muestra los datos de un servicio seleccionado cuando el usuario tiene permiso de administración, de ser un usuario con o sin permiso de administración el sistema muestra una vista de todos los servicios.	Al seleccionar el título de servicio deseado el sistema muestra una nueva interfaz con la información del servicio cuando el usuario tiene permiso de administración, de ser un usuario con o sin permiso de administración el sistema muestra una vista de todos los servicios.	1-Los usuarios autenticados pueden visualizar todos los contenidos de este tipo existentes, dirigiéndose al módulo <b>Administrar/ Contenido</b> ubicado en la barra de administración 1.2- El sistema muestra una página con el listado de todos los contenidos existentes en el mismo. 1.3- El usuario selecciona el título del deseado (Si el usuario desea que solo se le muestre el contenido específico, puede realizar un filtrado de la información, ver CP13 Filtrar contenido). 1.4- El sistema muestra una interfaz con los datos referentes al seleccionado. ----- -----2.-El usuario se desplaza hasta la sección <b>Servicios que prestamos</b> en la página principal. 2.1.- El usuario selecciona Ver todos los servicios. 2.2- El sistema muestra una interfaz con los servicios----- 3.-El usuario se desplaza hasta la sección <b>Servicios</b> en el menú de la página principal. 3.1- El sistema muestra una interfaz con los servicios. -----

Tabla 6 Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	<b>Título</b>	Campo de texto	No	Admite todos los caracteres. Longitud máxima 255 caracteres.
2	<b>Descripción</b>	Área de texto	No	Admite todos los caracteres.
3	<b>Imagen</b>	Campo de tipo archivo	No	Tamaño máximo 512 KB. Permite solo los archivos con las extensiones png, jpg, jpeg.
4	<b>Título de la imagen</b>	Campo de texto	Si	Permite todos los caracteres. Longitud máxima 1024. Este campo debe mostrarse al hacer clic en el botón Subir al servidor.
5	<b>Texto alternativo</b>	campo de texto	Si	Admite todos los caracteres. Este campo se muestra cuando se presiona el botón Subir al servidor.
6	<b>Categoría</b>	Lista de selección	No	Valores (Términos de la taxonomía Categoría de servicio).
7	<b>Publicado</b>	campo de selección	No	Valor por defecto(Marcado).

3-Ejecución de las pruebas.

Se procede a la ejecución de la prueba en cuestión en los navegadores y plataformas correspondientes.

4- Evaluación de las pruebas.

Recopilar en la tabla de no conformidades todos los errores encontrados, para luego corregirlos.

### 2.7.2 Estrategia para pruebas funcionales automatizadas en Drupal.

Se utilizarán los test PHPUnit y los test Behat.

#### 1. Creación de test PHPUnit.

Configuración para ejecutar pruebas de PHPUnit:

- Instalar Drupal con composer: `Composer create-project drupal /IPROYAZ: ^8 drupaltest`
- Descargar las dependencias para PHPUnit
- En la carpeta raíz se ejecutará el siguiente comando: `composer require --dev `drupal/core-dev: ^8.6``

- Abrir el proyecto en un editor de código en este caso se usará visual estudio code. Buscamos en el archivo raíz dentro de **vendor / bin** el archivo ejecutable de phpunit
- Buscar dentro del **core /phpunit.xml.dist** y renombrar a **phpunit.xml**
- Actualizar la información correspondiente a la URL del proyecto y a la base de datos.
- Habilitar **BROWSERTEST\_OUTPUT\_DIRECTORY** e **printerClass** en su archivo **phpunit.xml**
- Crear un archivo llamado **sites/simpletest/browser\_output**
- Comprobar si la configuración de las pruebas es la correcta: **vendor/bin/phpunit --version**

Se usará *BrowserTestBase* para la creación de los test porque ofrece una forma de probar comportamientos e interacciones basados en la web y satisface la necesidad de realizar pruebas de alto nivel basadas en solicitudes de integraciones entre varios subsistemas en Drupal. Cada prueba se ejecuta en una instancia de Drupal completamente nueva, que se crea desde cero para la prueba.

Para crear los escenarios de pruebas se usará el módulo *Rules* porque proporciona una interfaz de usuario para los administradores donde pueden crear reglas. Consiste en múltiples páginas y formularios que se quieren probar, los cuales están protegidos con permisos de usuario para que solo los administradores puedan acceder a ellos. El escenario de prueba más simple está en `rules / tests / src / Functional / UiPageTest.php`.

Hay cuatro pasos básicos involucrados en la creación de una prueba:

- Crear la estructura (simplemente crear una clase que herede de `\Drupal\Tests\BrowserTestBase` o una clase de prueba de navegador similar)
- Inicializar el caso de prueba con cualquier creación o configuración de usuario que deba realizarse
- Crear pruebas reales dentro del caso de prueba.
- Intentar desesperadamente averiguar por qué nuestra prueba no funciona de la manera que esperamos y depurar la prueba.

```
namespace Drupal\Tests\rules\Functional;
```

```
use Drupal\Tests\BrowserTestBase;
```

```
class UiPageTest extends BrowserTestBase {
```

Siguiente paso, debemos especificar la lista de módulos que deben habilitarse para la ejecución de prueba. En nuestro caso, esto es, por supuesto, las Rules.

```
protected static $modules = ['node', 'rules'];
```

Luego viene el setUp () opcional. Aquí es donde debemos hacer todo lo que sea necesario para que esta instancia de Drupal funcione como queremos. Si implementa el método setUp (), comience ejecutando el método parent :: setUp ()

```
protected function setUp(): void {
```

```
parent::setUp();
```

Crear pruebas específicas para ejercitar el módulo. Simplemente se crearán métodos de la clase de prueba, cada uno de los cuales realiza una prueba en particular. Todos los métodos deben comenzar con *test* para que sean reconocido automáticamente por PHPUnit y se ejecutarán cuando se soliciten.

```
public function testPrueba(){
```

```
    // código del caso de prueba a ejecutar //
```

```
}
```

Para hacer los métodos se hacen solicitudes GET muy simples. La mayoría de las pruebas seguirán este patrón:

`$this->drupalGet(some/path)` para ir a una página.

`$this->clickLink()` para navegar por enlaces de la página.

`$this->getSession()->getPage()->fillField()` para completar los campos del formulario.

`$this->getSession()->getPage()->pressButton()` o `$this->submitForm()` para enviar formularios.

### 2. Creación de test Behat.

Instalación e implementación de Behat en Drupal

La forma oficial de instalar Behat en su proyecto Drupal es a través de Composer.

```
$ composer require behat/behat
```

A continuación, ejecute el comando `composer install` en la terminal.

- Cuando haya terminado, también verá un nuevo directorio `bin /` con un archivo Behat en él. Este es el ejecutable de Behat y se usará para ejecutar las pruebas y obtener información de depuración.
- A continuación, crear un archivo llamado `behat.yml` dentro de la carpeta raíz del proyecto. Cuando se ejecuta Behat, buscar un archivo `behat.yml`, que utiliza para su configuración.

Escribir historias y ejecutar pruebas con Behat

Nuestro objetivo es describir las historias de la forma más eficaz. Las historias están escritas en un lenguaje legible por humanos, Gherkin, con la extensión `.feature` en el directorio `features /`.

Cada test puede tener muchos escenarios, que describen el comportamiento específico de la característica a probar. Cada escenario tendrá secciones:

- **Given:** detalla el estado inicial del sistema.
- **When:** incluye la acción que realiza el usuario.
- **Then:** describe lo que ve el usuario después de realizar una acción.
- **And** - para mantener la conectividad en el escenario.

### 3. Ejecutando los test

Para PHPUnit:

```
cd core
```

```
../vendor/bin/phpunit --filter testPrueba
```

```
../modules/rules/tests/src/Functional/UiPageTest.php
```

Para Behat:

```
vendor / bin / behat
```

Si desea ejecutar un archivo de características específico

```
vendor / bin / behat features /nombrePrueba.feature
```

Ejecutar un único escenario simplemente agregando un número de línea donde se define el escenario:

```
vendor / bin / behat features / prueba.feature: 11.
```

Para aplicar las pruebas automatizadas se debe:

- 1) Hacer las instalaciones y configuraciones previas de acuerdo con el tipo de test que se vaya a hacer.
- 2) Crear los test necesarios en dependencia de los casos de prueba.
- 3) Ejecutar los test creados.
- 4) Ver los resultados de los test y en caso de haber algún error localizar donde es para que sea corregido.

### 2.8 Conclusiones

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

- El análisis de las características del sistema permitió identificar los principales requisitos funcionales y no funcionales, los que fueron agrupados y categorizados en sus respectivas historias de usuarios.
- El diseño de los diagramas de clases, permitió obtener el enfoque en cuanto a composición lógica y física del portal web IPROYAZ, así como permitió una mejor comprensión sobre cómo debe comportarse el mismo.

- El diseño de los casos de prueba como elemento principal para poder aplicar las pruebas funcionales de forma manual lo que permitirá detectar las no conformidades por iteración que deberán resolverse para cumplir con la calidad del software.
- El análisis de los elementos fundamentales para poder ejecutar las pruebas automatizadas en drupal mediante la creación de los test PHPUnit y test Behat lo que permitirá detectar los errores que incumplan con los requisitos funcionales y a su vez con la calidad del software.

## **Capítulo 3: “Validación de la propuesta para las pruebas de software.”**

### **3.1 Introducción.**

Antes de escribir una sola línea de código, es fundamental haber comprendido bien el problema que se pretende resolver y haber aplicado principios básicos de diseño que permitan construir un sistema de calidad. Una vez que se sabe qué funciones debe desempeñar el sistema (análisis) y se ha decidido cómo organizar sus distintos componentes (diseño), es el momento de pasar a la etapa de implementación

En el presente capítulo se abordarán las actividades que se llevan a cabo durante la fase de implementación y prueba. Se describe la organización del módulo en un diagrama de componentes y se especifican los estándares de codificación a utilizar. Se aplicarán las estrategias de pruebas propuestas para determinar cuál conlleva menos esfuerzos durante el proceso lo que permitirá definir una estrategia final y evaluar la calidad del producto desarrollado y garantizar que este cumpla con las funcionalidades requeridas.

### **3.2 Modelo de despliegue.**

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

A continuación, se describen los elementos que componen el diagrama de despliegue para el portal web IPROYAZ

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

**Dispositivos:** Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.

**Conectores:** Expresa el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

**LDAP:** Es la estación de trabajo que brinda la posibilidad de comprobar las credenciales de los usuarios que se autentican en el sistema.

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

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

**servidor\_aplicacion:** Es la estación de trabajo que hospeda el código fuente de la aplicación, y que les brinda a los usuarios las interfaces de la misma para realizar los procesos definidos por cada uno de los roles del sistema. Esta estación se comunica con el servidor de base de datos donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo *TCP/IP*. Además, se comunica con el servidor *LDAP* que le brinda la posibilidad de comprobar las credenciales de los usuarios que se autentican en el sistema.

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

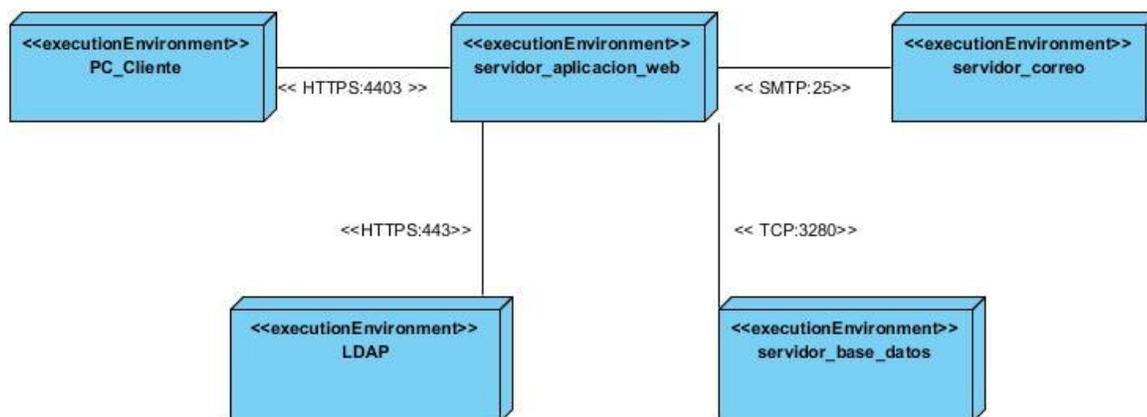


Figura: 5 Diagrama de despliegue

### Diagrama de componentes

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

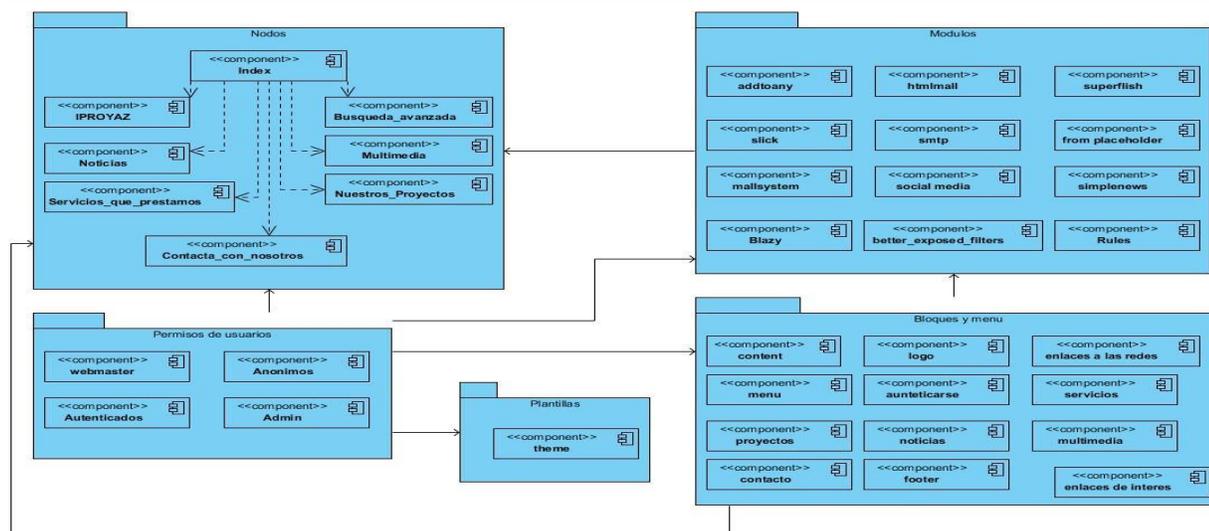


Figura: 6Diagrama de Componente

### 3.3 Estándares de codificación de Drupal

Los estándares de codificación definidos por los desarrolladores de *Drupal* para desarrollar un sistema de tal manera que a otros programadores se les facilite entender el código (como identificar las variables, las funciones o métodos, etc.) son:

**Indentación:** La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos (Drupal, 2020).

```

jQuery('#block-formulariodebusqueda').hover(
  function() {
    jQuery('input#edit-keys').css({
      "width": "112px",
      "background-color": "#fff"
    });
  },
  function() {
    jQuery('input#edit-keys').css({
      "width": "0px",
      "background-color": "black"
    });
  }
);

```

**Etiquetas de apertura y cierre de PHP:** Cuando se utiliza el lenguaje *PHP*, siempre se deben utilizar las etiquetas `<? php` y `?>`, y en ningún caso la versión corta `<?` y `?>`. En general se omite la etiqueta de cierre de *PHP* (`?>`) al final de los archivos *.module* y *.inc*. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre `?>`), que serían

identificados como salida *HTML* y podrían provocar un error muy típico, "Cannot modify header information - headers already sent by...". (Drupal, 2020)

```
<?php

function pa_form_search_block_form_alter(&$form, &$form_state)
{
    $form['actions']['submit']['#value'] = html_entity_decode('&#xf349;');
}

function pa_preprocess_comment(&$variables){
    $comment = $variables['elements']['#comment'];
    $variables['createddate'] = \Drupal::service('date.formatter')->format($comment->getCreatedTime(), 'fecha');
}
```

**Operadores:** Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, **\$numero = 3**, en lugar de \$numero=3. Esto se aplica a operadores como +, -, \*, /, =, ==, !=, >, <, . (Concatenación de cadenas),.=, +=, -=, etc. Los operadores unarios como ++, -- no deben tener separación. Por ejemplo, **\$numero++** (Drupal, 2020).

```
foreach (\Drupal::moduleHandler()->getModuleList() as $module => $filename) {
    $output .= " * - $module\n";
}
```

**Uso de comillas:** Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<h1>\${title}</h1>". También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple (Drupal, 2020).

```
function pa_preprocess_comment(&$variables){
    $comment = $variables['elements']['#comment'];
    $variables['createddate'] = \Drupal::service('date.formatter')->format($comment->getCreatedTime(), 'fecha');
}
```

**Estructuras de control:** Con respecto a las estructuras de control, hay que tener en cuenta las siguientes normas (Drupal, 2020):

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

Las estructuras *else* y *elseif* se escribirán en la línea siguiente al cierre de la sentencia anterior.

```
foreach (\Drupal::moduleHandler()->getModuleList() as $module => $filename) {  
  $output .= " * - $module\n";  
}
```

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

Cuando la línea de declaración del array supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos (Drupal, 2020).

**Variables globales:** Aunque el uso de variables globales está desaconsejado, en caso de necesitarse, éstas se declararán utilizando un guion bajo inicial, seguido del nombre del módulo o tema y otro guion bajo antes del nombre de la variable (Drupal, 2020).

La llave de apertura { se situará en la misma línea que la definición de la estructura, separada por un espacio.

```
function() {
    jQuery('input#edit-keys').css({
        "width": "112px",
        "background-color": "#fff"
    });
},
```

- Se recomienda usar siempre las llaves {} aún en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control).
- Las estructuras *else* y *elseif* se escribirán en la línea siguiente al cierre de la sentencia anterior.

### 3.4 Aplicación de la estrategia de validación del portal web IPROYAZ.

#### 3.4.1 Resultados de las pruebas no funcionales:

##### Pruebas de Rendimiento.

Al Portal web se le realizará la prueba de rendimiento de tipo carga y estrés. Se utilizará la herramienta Apache JMeter en su versión 4.2, diseñado para pruebas de cargas de comportamientos funcionales y la medición del rendimiento. El ambiente de prueba estuvo conformado por:

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

- Sistema Operativo: Linux Mint 18
- Microprocesador: Intel(R) Core (TM) i3-4100U CPU @2.40GHz 2.40GHz
- Memoria RAM: 4.00 GB
- Disco Duro: 500 GB

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

- Sistema Operativo: Windows v.10
- Microprocesador: Intel(R) Core (TM) i5-5200U CPU @2.20GHz 2.20GHz
- Memoria RAM: 8.00 GB
- Disco Duro: 1024 GB
- Tipo de Sistema: Sistema operativo de 64 bits, procesador x64

**Software instalado en ambas PC:**

- Tipo de servidor web: Apache 2.4 (PC cliente).
- Plataforma: SO Linux (PC servidor) y SO Windows (PC cliente).
- Servidor de BD: MySQL 5.7.24 (PC cliente).

Con la prueba de rendimiento se pretende analizar el comportamiento del portal cuando existan 25,50 y 100 usuarios conectados simultáneamente evaluando los siguientes parámetros:

Luego de haber definido el *hardware* se configuran los parámetros del Apache JMeter logrando un total de 25 usuarios conectados concurrentemente.

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

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

*Tabla 7 Resultados de la prueba de carga y estrés.*

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

De los resultados obtenidos en las pruebas se determina que la aplicación cumple con los requisitos que se presentan. El informe de Apache JMeter arroja que el portal web es capaz de responder 100 peticiones

de 25 usuarios conectados simultáneamente en un tiempo promedio de 3039 milisegundos con un error de 0.0% esto evidencia que el portal puede responder correctamente las peticiones realizadas.

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

### Pruebas de Seguridad

Se emplea la herramienta *Acunetix WVS* con el fin de evaluar la seguridad del portal web la cual arroja los siguientes resultados luego de haber realizado una primera iteración se obtiene un total de 9 no conformidades, divididas en 5 de nivel medio, 3 de nivel bajo e 1 de nivel informacional. Se destaca en el nivel medio el uso de credenciales de usuarios enviadas en texto plano. Por otra parte, en el nivel bajo destaca campos de contraseña con auto completamiento activado. Y de carácter informacional fueron detectados campos de usuario y contraseña mostrados. Estas deficiencias fueron corregidas en la primera iteración, en una segunda no se identificaron nuevas no conformidades, obteniéndose así una herramienta que cumple con los requisitos de seguridad. A continuación, se muestran los resultados descritos.

### Pruebas de usabilidad:

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

Tabla 8 Chequeo de usabilidad

Categoría de los indicadores	Indicadores	Proceden	Correctos	Incorrectos
Visibilidad del sistema	17	17	16	1
Lenguaje común entre sistema y usuario	11	7	7	0
Libertad y control por parte del usuario	29	22	21	0
Consistencia y estándares	33	23	23	0

Estética y diseño minimalista	18	6	6	0
Prevención de errores	8	3	2	1
Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	11	6	6	0
Ayuda y documentación	11	7	7	0
Flexibilidad y eficiencia de uso	6	6	6	0
<b>Total</b>	<b>144</b>	<b>97</b>	<b>94</b>	<b>2</b>

Como se observa en la tabla, de los 144 parámetros originales de la lista de chequeo, solo proceden 97, se evaluaron como correctos 94 parámetros, identificando 2 no conformidades las cuales fueron resueltas, para un 100 % de usabilidad.

### 3.4.2 Resultados de las pruebas funcionales:

A continuación, se muestran los resultados obtenidos en tres de las iteraciones de pruebas realizadas al portal web Iproyaz, la Figura 7 brinda información del total de no conformidades encontradas y las que fueron resueltas por cada iteración. Con un total de 72 requisitos funcionales se detectaron 27 NC en la primera iteración resolviéndose satisfactoriamente 24, en la segunda iteración se detectaron 6 NC resolviéndose todas satisfactoriamente y en la tercera iteración no se detectaron NC.



Figura: 7 Resultado de las pruebas de funcionales por casos de prueba.

Tabla 9 Tabla de NC de una de las NC encontradas.

<b>Tipo de Prueba</b>	Funcional
<b>Nombre del Probador</b>	Daylin Hernández Pérez
<b>Nº</b>	2
<b>Elementos</b>	Contenido
<b>CP</b>	Crear servicio
<b>No conformidad</b>	Al subir una imagen que sobrepasa el tamaño disponible el mensaje que muestra es "El archivo <i>x.jpg</i> no se pudo guardar porque sobrepasa los 2 MB, el tamaño máximo para las subidas."
<b>Localización de la no Conformidad</b>	Escenario 16.2
<b>Sugerencias</b>	Debe mostrar el mensaje: "El archivo especificado TÍTULO-IMAGEN no se pudo subir al servidor. El archivo ocupa TAMAÑO-IMAGEN excediendo el máximo permitido por archivo que es 512KB."
<b>Atributos asociados</b>	campo imagen: inválido

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

- Errores de validación.
- Errores ortográficos.
- Errores de correspondencia entre artefactos.

### 3.4.3 Pruebas automatizadas:

```

Area de trabajo
account_registration.feature
FeatureContext.php -- bin/feature
index.php
example.gitignore
composer.lock
composer.json -- pw_iproyaz-mast
autoload.php
Message.php
MessagePlugin.php
pw_iproyaz-master
  bin
  features
    bootstrap
    test
  behat
    behat.bat
  composer
  doctrine
  drupal
    core-composer-scaffold
    core-project-message
    core-vendor-hardening
    drupal-driver
    drupal-extension
    doc
      _static
      snippets
        aliases.drushrc.php
        api.feature
        apitag.feature
        apitag.output
        behat-1.yml
        behat-api.yml
1 Feature: Crear servicio
2 In order to create an servicio:
3 As an webmaster
4 I want to acces /node/add
5 So that I can create a page
6
7 Scenario: Complete the create service form
8 Given I am logged in as a user whit the "webmaster" role
9 when I go to "/node/add/servicio"
10 And I enter "Asunto 1" for "eedit-title-0-value"
11 And I select "Investigaciones aplicadas" for "edit-field-ca"
12 And I enter "Lorem ipsum dolor sit amet" for "edit-body-0-value"
13 And I press the "edit-submit"
14 Then I should see "El campo imagen es obligatorio"
15
16 Scenario: Complete the create service empty form
17 Given I am logged in as a user whit the "webmaster" role
18 when I go to "/node/add/servicio"
19 And I enter "" for "eedit-title-0-value"
20 And I select "" for "vedit-field-ca"
21 And I enter "" for "edit-body-0-value"
22 And I press the "edit-submit"
23 Then I should see "El campo titulo es obligatorio, el campo sleccion categoria es obligatorio,el campo imagen es obligatorio"
24
25
26 Escenario: Complete the edit service form
27 Given I am logged in as a user whit the "webmaster" role
28 when I go to "/content/"
29 And I see my content of type:Servicio whit tile: Evaluacion económica y financiera.
30 Then I should can content type:Servicio
31 And I press "edit dropdown-action"
32 And I go to "/admin/content/"
33 And I enter "Otro servicio" for "eedit-title-0-value"
34 And I select "Investigaciones aplicadas" for "edit-field-ca"
35 And I enter "" for "edit-body-0-value"
36 And I press the "edit-submit"
37 Then I should see "El campo titulo es obligatorio, el campo sleccion categoria es obligatorio,el campo imagen es obligatorio"
38
39
40 Escenario: Escenario: Complete the edit service empty form
41 Given I am logged in as a user whit the "webmaster" role
42 when I go to "/content/"
43 And I see my content of type:Servicio whit tile: Evaluacion económica y financiera.
44 Then I should can content type:Servicio
45 And I press "edit dropdown-action"
46 And I go to "/admin/content/"
47 And I enter "Otro servicio" for "eedit-title-0-value"
48 And I select "Investigaciones aplicadas" for "edit-field-ca"
49 And I enter "" for "edit-body-0-value"
50 And I press the "edit-submit"
51 Then I should see "El campo titulo es obligatorio, el campo sleccion categoria es obligatorio,el campo imagen es obligatorio"
52

```

Figura: 8 Test Behat creado.

Este test cuenta con 4 escenarios donde se evalúa que los campos título, categoría funcionen correctamente cuando los mismos se llenan tanto con datos correctos e incorrectos.

El campo imagen no se evalúa en este test porque los test behat no permiten el procesamiento de imagen por lo que este campo se evaluara mediante test Phpunit que se harán.

El código de este test se reutilizó para la creación de otros test donde se evaluaron campos parecidos.

### 3.5 Propuesta de estrategia final:

Después de haber aplicado las dos estrategias propuestas se define que la estrategia final será un híbrido entre las pruebas manuales y las pruebas automatizadas la cual contará con los siguientes pasos:

1. Planificación de las pruebas.

2. Diseño de los casos de pruebas.
3. Hacer las instalaciones y configuraciones previas de acuerdo con el tipo de test que se vaya a hacer.
4. Crear los test necesarios en dependencia de los casos de prueba.
5. Ejecutar los test creados.
6. Ver los resultados de los test y en caso de haber algún error localizar donde es para que sea corregido.
7. Ejecución de las pruebas: Se procede a la ejecución de la prueba en cuestión en los navegadores y plataformas correspondientes.
8. Evaluación de las pruebas.  
 Recopilar en la tabla de no conformidades todos los errores encontrados, para luego corregirlos.

Esta estrategia disminuirá el esfuerzo empleado actualmente en el proceso de prueba del sistema porque el tiempo que se emplea en hacer las instalaciones y configuraciones previas para crear los test proporciona mejoras en el desarrollo debido a que se obtiene mayor seguridad a la hora de escribir nuevas funcionalidades, además no habrá que preocuparse de escribir pensando en se pueda estropear algún código que haya implementado otra persona porque serán los test los que avisarán y así actuar entonces, permiten disminuir las NC porque los errores serán corregidos antes de ejecutar las pruebas manuales. Si disminuyen las no conformidades pueden disminuir la cantidad de iteraciones que se realicen de las pruebas así como el tipo de error de la NC. Las pruebas manuales permiten probar los escenarios a los que no se le apliquen test automatizados.

*Tabla 10 Ventajas y desventajas de las pruebas manuales y las pruebas automatizadas.*

	<b>PRUEBAS MANUALES</b>	<b>PRUEBAS AUTOMATIZADAS.</b>
<b>VENTAJAS</b>	<ul style="list-style-type: none"> <li>- Permite crear diversos escenarios que permiten comprobar que el sistema cumpla con todos los requisitos funcionales.</li> <li>-Se adaptan fácilmente a las necesidades y realidades de cada proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>-Garantiza eficiencia y calidad del software obtenido.</li> <li>-Los test que tratan con tarea repetitiva lo harán de forma más rápido.</li> <li>-Se pueden ejecutar cada vez que sea necesario.</li> <li>-Evalúan la usabilidad y</li> </ul>

	<p>-Garantiza eficiencia y calidad del software obtenido.</p>	<p>experiencia de usuario. -Elimina los típicos errores causados por los desarrolladores cuando se realizan las pruebas manuales.</p>
<p><b>DESVENTAJAS</b></p>	<p>-Se pueden encontrar NC del mismo tipo en diferentes escenarios.</p>	<p>- Escribir las funciones automatizadas puede llevar tiempo. -No todos los aspectos visuales se pueden probar de esta manera.</p>

### 3.5 Interfaces del portal web Iproyaz:



Figura: 9 Banner del portal web Iproyaz.



Figura: 10 Bloque de servicios del portal web Iproyaz.

### 3.6 Conclusiones parciales del capítulo:

- La confección del modelo de despliegue nos permite describir la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.
- La confección del diagrama de componente permitió la integración de los componentes de software.
- Al aplicar los estándares de codificación facilita la comprensión y legibilidad del código fuente, permitiendo desarrollar el portal en un menor tiempo y garantizando la facilidad y calidad de un futuro mantenimiento.
- La realización de las pruebas de seguridad, funcionalidad, carga y estrés, guían la calidad de la aplicación y correcta ejecución de las funcionalidades del sistema.
- La realización de las pruebas automatizadas permitió crear una nueva estrategia de pruebas funcionales que garantiza la disminución de las NC y a su vez garantiza la calidad del software.

## **Conclusiones Generales**

- A partir del estudio realizado de los referentes teóricos relacionados con la estrategia de pruebas funcionales y creación de los test automatizados se seleccionaron las características y funcionalidades para la propuesta de solución de acuerdo a las necesidades existentes. Así como las herramientas para la implementación del portal web Iproyaz.
- La utilización de las tecnologías y herramientas seleccionadas, permitió analizar y describir los procesos que se debían ejecutar para la implementación del portal.
- Con la definición de la arquitectura y los estándares de codificación se logró el empleo de buenas prácticas en el desarrollo del portal.
- La definición de la estrategia de pruebas funcionales manuales y de pruebas automatizadas, permitió analizar y describir los procesos que se debían ejecutar para el desarrollo de la propuesta de solución.
- Con la utilización de las validaciones de pruebas al portal web Iproyaz se garantiza la corrección de las no conformidades detectadas, demostrándose que el sistema constituye una solución funcional.
- La utilización de los test Behat permitió corregir errores que habían sido determinados como no conformidades lo que ayudó a establecer una nueva estrategia para realizar las pruebas funcionales siendo la misma un híbrido entre las pruebas funcionales manuales y las automatizadas lo que garantizará una mayor calidad de software.

## **Recomendaciones**

Para dar continuidad a la presente investigación se recomienda:

Crear los test Phpunit para el procesado de imágenes.

Realizar las pruebas de aceptación.

Utilizar otros módulos como Google Optimize A/B Test UI y Integración Crazy Egg de Drupal para crear otro test automatizado para probar otras funcionalidades del portal.

## Referencias Bibliográficas

Acuentix. (s. f.). <https://www.acunetix.com/>

Ateneatech. (2019, septiembre 19). *Automatizando test funcionales con Behat y Drupal: Instalacion y puesta en marcha*. <https://ateneatech.com/blog/automatizando-test-funcionales-con-behat-y-drupal-instalacion-y-puesta-en-marcha-desde-cero>

Bibián. (2017). *PRUEBAS DE CALIDAD APLICADAS AL SITIO WEB ALLISON. COLIMA-MEXICO: Secretaria de Educacion Pública*.

Biblioteca Medica Nacional. (s. f.). *Que son las TIC*. <http://www.bmnns.sld.cu/que-son-las-tic>.

Buitrago, J. (2017, mayo 11). *Drupal Groups*. <https://groups.drupal.org/node/148379>

Campusmvp. (2020, marzo 10). *Qué son las pruebas de software*. <https://www.campusmvp.es/recursos/post/que-son-las-pruebas-de-software.aspx>

Carbó, Y. (s. f.). *Cuba y el impacto de las TIC en la informatizacion de la sociedad*. <https://www.monografias.com/trabajos109/cuba-y-impacto-tic-informatizacion-sociedad/cuba-y-impacto-tic-informatizacion-sociedad.shtml>.

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

Cardozzo. (2016). *Desarrollo de Software: Requisitos, Estimaciones y Análisis*. (2 Edición. S.l.: IT Campus Academy.).

Dinahosting. (s. f.). *Que es apache y para que sirve*. <https://dinahosting.com/ayuda/que-es-apache-y-para-que-sirve/>

Drupal. (2020, junio). *Drupal*. <https://www.drupal.org/>

*Drupal*. (2021, marzo 22). <https://www.drupal.org/docs/automated-testing/phpunit-in-drupal/running-phpunit-tests>

Esper, L. (2020, noviembre). <https://www.tiendanube.com/blog/que-es-twig-como-usarlo/>

Instituto de nuevas tecnologias. (2016, enero 20). *Bootstrap ,ventajas y desventajas*. <https://tecnologiaenvivo.com/bootstrap-ventajas-y-desventajas/>

Jimenez-Gonzales. (2018). *APLICACIÓN WEB PARA LA GESTIÓN DE LA INFORMACIÓN EN LA ESCUELA DE CAPACITACIÓN DE LA CONSTRUCCIÓN DE PINAR DEL RÍO, CUBA* (UNESUM-Ciencias: Revista Científica Multidisciplinaria .).

*Jmeter*. (s. f.). <https://jmeterenespanol.org/>

- Lima, J. (2020). *Portal web para el centro de desarrollo CIDI*. Universidad de las Ciencias Informatica.
- MDN Web Doc. (s. f.). *HTML: Lenguaje de etiquetas de hipertexto*. <https://developer.mozilla.org/es/docs/Web/HTML>
- NeoAttack. (s. f.). <https://neoattack.com/neowiki/mysql/>
- Olivieri. (s. f.). *Hernán Olivieri*. <http://www.hernanolivieri.com>
- Php. (s. f.). *Que es PHP*. <https://www.php.net/manual/es/intro-what-is.php>
- Pressman, R. (s. f.). *Ingengeria de software, un enfoque práctico* (Quinta edición). McGraw-Hill Companies.
- Sommerville. (2011). *Software engineering* (9th Edition).
- Toll Palma, Y. del C., & Ril Gil, Y. (s. f.). *Propuesta de manual de procedimiento para pruebas de sistema*.
- Trycore. (s. f.). [Tipos de pruebas funcionales]. <https://trycore.com/transformacion-digital/tipos-de-pruebas-funcionales/>
- Uml. (s. f.). *Introduction to OMG'S Unified Modeling Language*. OMG. <https://www.uml.org/what-is-uml.htm>.
- Universidad Latina de Costa Rica. (s. f.). *Que son las TIC y para que sirven*. <https://www.ulatina.ac.cr/blog/qu3-son-las-tic-y-para-que-sirven>.

## Tablas

Tabla 11 Descripción de las clases válidas y clases inválidas.

Condición de Entrada	Clases válidas	Clases inválidas	Resultado Obtenido	Resultado de la prueba	Observaciones
[Es la condición especificada en el diccionario de datos del CU]	[Será la que cumpla con la especificación de un rango de valores para los datos de entrada] [Datos válidos]	[Será la que no cumpla con la especificación de un rango de valores para los datos de entrada] [Datos no válidos o erróneos]	[El resultado que devolvió el sistema como respuesta según la entrada]	[Satisfactorio o insatisfactorio]	[Cualquier sugerencia]

Tabla 12 Descripción del Caso de Prueba.

<b>Tipo de Prueba</b>	[Nombre del tipo de prueba de sistema: funcional]
<b>Nombre del Caso de Uso (1...n)</b>	[Nombre del CU que se probará]
<b>Descripción</b>	[Describir brevemente los pasos a seguir para llegar al CU sometido a prueba ]
<b>Nombre del Caso de Prueba (1....n)</b>	[Nombre del CP y especificar el escenario donde se encuentra] Ejemplo: El nombre del caso de prueba está dado según la cantidad de flujos tanto básicos como alternos
<b>Condición de Entrada</b>	[Descripción textual de lo que ocurra en el mundo real que hace necesario ejecutar el caso de prueba, precisando los datos de entrada y los comandos a dar por el actor. Descripción textual del estado de la información almacenada.]
<b>Salida Esperada</b>	[Descripción textual del estado en que queda la información y las alertas que pueden generarse, una vez ejecutado el CU con los valores y el estado especificado en la entrada.]

Tabla 13 Descripción de la tabla de No Conformidades para pruebas funcionales.

<b>Tipo de Prueba</b>	[Nombre de la prueba]
<b>Nombre del Probador</b>	[Nombre del probador]
<b>Nº</b>	[Número de la no conformidad]
<b>Elementos</b>	[Tipo de elemento, ejemplo: interfaz]
<b>CU</b>	[Nombre del CU correspondiente a la no conformidad encontrada]
<b>No conformidad</b>	[Describir la no conformidad encontrada]
<b>Localización de la no Conformidad</b>	[El escenario, sección o pantalla del CU asociado, donde se encontró]
<b>Sugerencias</b>	[Comentario que desee poner sobre la no conformidad encontrada]
<b>Atributos asociados</b>	[Especificar la subcaracterística asociada tanto para las no conformidades, como para las sugerencias, por separado. Ejemplo: Completitud (no conformidad), Corrección (sugerencia)]

Tabla 14 Descripción de la tabla de No Conformidades para Pruebas no Funcionales.

<b>Tipo de Prueba</b>	[Nombre de la prueba]
<b>Nombre del Probador</b>	[Nombre del probador]
<b>Nº</b>	[Número de la no conformidad]
<b>Elementos</b>	[Tipo de elemento, ejemplo: interfaz]
<b>Nombre del Requisito No Funcional</b>	[Nombre del requisito no funcional correspondiente a la no conformidad encontrada]
<b>No conformidad</b>	[Describir la no conformidad encontrada]
<b>Localización de la no Conformidad</b>	[El escenario, sección o pantalla del CU asociado, donde se encontró]
<b>Sugerencias</b>	[Comentario que desee poner sobre la no conformidad encontrada]
<b>Atributos asociados</b>	[Especificar la subcaracterística asociada tanto para las no conformidades, como para las sugerencias, por separado. Ejemplo: Completitud (no conformidad), Corrección (sugerencia)]

## Historias de usuario

Tabla 15 Crear usuario

<b>Número:</b> 2	<b>Nombre del requisito:</b> Crear usuario
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgos	<b>Tiempo Real:</b>
<p><b>Descripción:</b></p> <p>El sistema permite a los usuarios con rol webmaster<sup>2</sup> o anónimos <b>crear</b> usuarios en el sistema llenando los siguientes campos:</p> <ul style="list-style-type: none"> <li>• <b>Usuario:</b> Campo de texto, obligatorio. Longitud entre 5 y 60 caracteres. Permiten los caracteres: (a-z, A-Z, 0-9). Debe comenzar con (a-z, A-Z).</li> <li>• <b>Correo electrónico:</b> Campo de texto, obligatorio. Longitud máxima 254 caracteres. Permite direcciones de correo estructuralmente válidas (Estructura: usuario@subdominios.dominio)</li> <li>• <b>Contraseña:</b> Campo de texto, obligatorio. Longitud entre 6 y 128 caracteres. Permite todos los caracteres. Para que sea una contraseña válida debe tener a-z, A-Z, 0-9 y signos de puntuación, al menos una vez.</li> <li>• <b>Confirmar contraseña</b> Campo de texto, obligatorio. Longitud entre 5 y 60 caracteres. Permite todos los caracteres. Para que sea una contraseña válida debe tener a-z, A-Z, 0-9 y signos de puntuación, al menos una vez. Debe coincidir con la contraseña del campo Contraseña.</li> <li>• <b>Estado:</b> Campo de selección, obligatorio. Valores (Bloqueado, Activo). Valor por defecto (Activo)).</li> </ul>	

<ul style="list-style-type: none"> <li>• <b>Rol:</b> Campo de selección múltiple, opcional. Valores (webmaster, editor, forista).</li> <li>• <b>Notificar al usuario acerca de su nueva cuenta:</b> Campo de selección, opcional. Valor por defecto (Desmarcado).</li> </ul>
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. Si el usuario introduce los datos de forma correcta, el sistema emitirá un mensaje confirmando la culminación exitosa de la acción.</li> <li>2. Si el usuario introduce los datos de forma incorrecta, el sistema emitirá un mensaje notificando el error.</li> <li>3. Si el usuario deja campos obligatorios vacíos, el sistema emitirá un mensaje notificando el error.</li> <li>4. Si el usuario ya existe, el sistema emite un mensaje notificando que el usuario ya está en el sistema.</li> <li>5. Si la cuenta es creada por un usuario anónimo, esta se crea por defecto con rol forista, sin posibilidad de seleccionar ningún otro rol.</li> <li>6. Si la cuenta es creada por un usuario con rol webmaster autenticado en el sistema, este puede seleccionar del listado de rol, el que desea para la cuenta.</li> </ol>
<p><b>Prototipo elemental de interfaz gráfica de usuario:</b> No aplica.</p>

*Tabla 16 Editar usuario*

<b>Número:</b> 3	<b>Nombre del requisito:</b> Editar usuario
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgos	<b>Tiempo Real:</b>
<p><b>Descripción:</b> El sistema permite a los usuarios con rol webmaster editar las cuentas existentes en el sistema.</p>	

Si un usuario desea modificar los campos Dirección de correo electrónico y Contraseña de su cuenta tendrá además que llenar el campo Contraseña actual:

- **Usuario:** Campo de texto, obligatorio. Longitud entre 5 y 60 caracteres. Permiten los caracteres: (a-z, A-Z, 0-9). Debe comenzar con (a-z, A-Z).
- **Contraseña actual:** Campo de texto, opcional. Longitud entre 6 y 128 caracteres. Permite todos los caracteres. El campo solo se muestra en el **editar** y solo se llena si el usuario autenticado desea **editar** su propia cuenta, y en caso de que vaya a modificar el correo o la contraseña.
- **Correo electrónico:** Campo de texto, opcional. Longitud máxima 254 caracteres. Permite direcciones de correo estructuralmente válidas (Estructura: usuario@subdominios.dominio) (Si el usuario a modificar tiene creado anteriormente correo electrónico, entonces este campo es obligatorio)
- **Contraseña:** Campo de texto, obligatorio. Longitud entre 6 y 128 caracteres. Permite todos los caracteres. Para que sea una contraseña válida debe tener a-z, A-Z, 0-9 y signos de puntuación, al menos una vez.
- **Confirmar contraseña** Campo de texto, obligatorio. Longitud entre 5 y 60 caracteres. Permite todos los caracteres. Para que sea una contraseña válida debe tener a-z, A-Z, 0-9 y signos de puntuación, al menos una vez. Debe coincidir con la contraseña del campo Contraseña.
- **Estado:** Campo de selección, obligatorio. Valores (Bloqueado, Activo). Valor por defecto (Activo)).
- **Rol:** Campo de selección múltiple, opcional. Valores (webmaster, editor, forista).

**Observaciones:**

1. Si el usuario introduce los datos de forma correcta, el sistema emitirá un mensaje confirmando la culminación exitosa de la acción.
2. Si el usuario introduce los datos de forma incorrecta, el sistema emitirá un mensaje notificando el error.
3. Si el usuario deja campos obligatorios vacíos, el sistema emitirá un mensaje notificando el error.
4. El Campo Contraseña actual solo se requiere si el usuario autenticado desea modificar los campos Correo electrónico y Contraseña de su propia cuenta.

**Prototipo elemental de interfaz gráfica de usuario:** No aplica.

Tabla 17 Mostrar listado de usuarios

<b>Número:</b> 5	<b>Nombre del requisito:</b> Mostrar listado de usuarios
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgos	
<b>Descripción:</b> Los usuarios con rol webmaster pueden ver el listado de todos los usuarios existentes al seleccionar en el bloque de administración, el módulo Administrar/Usuarios. De los usuarios se muestra Nombre de usuario, Estado, Roles , Miembro desde hace, Último acceso y Operaciones.	
<b>Observaciones:</b> 1. Si no existen usuarios en el sistema el listado se mostrará vacío.	
<b>Prototipo elemental de interfaz gráfica de usuario:</b> No aplica.	

Tabla 18 Cancelar cuenta de usuario

<b>Número:</b> 7	<b>Nombre del requisito:</b> Cancelar cuenta de usuario.
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgo	<b>Tiempo Real:</b>
<b>Descripción:</b> Los usuarios que tengan rol webmaster podrán eliminar usuarios en el sistema cancelando una o varias cuentas de usuario, Para ello debe seleccionar la opción cancelar cuenta del usuario deseado en el listado de usuarios existentes o Cancelar las cuentas de usuarios seleccionadas	

<p>del siguiente campo y presionar el botón Aplicar a los elementos seleccionados.</p> <p><b>Acción:</b> Obligatorio. Lista desplegable. Valores (Lista de opciones).</p> <p>Al hacer clic en el botón Aplicar a los elementos seleccionados el sistema debe permitir:</p> <ul style="list-style-type: none"> <li>• Desactivar la cuenta y mantener su contenido.</li> <li>• Desactivar la cuenta y retirar de la publicación su contenido.</li> <li>• Eliminar la cuenta y atribuir todo su contenido al usuario Anónimo.</li> <li>• Eliminar la cuenta y su contenido.</li> </ul>
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. Para que el sistema realice la funcionalidad es obligatorio que seleccione al menos un usuario.</li> </ol>
<p><b>Prototipo elemental de interfaz gráfica de usuario:</b> No aplica.</p>

Tabla 19 Crear noticia

<b>Número:</b> 21	<b>Nombre del requisito:</b> Crear noticia
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgo	<b>Tiempo Real:</b>
<p><b>Descripción:</b></p> <p>Los usuarios autenticados con privilegios de administración podrán <b>crear</b> noticias en el sistema, para ello deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> <li>• <b>Título:</b> Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 255.</li> <li>• <b>Imagen/Añadir archivo nuevo:</b> Obligatorio. Campo de tipo archivo. Tamaño máximo 512 KB. Permite solo los archivos con las extensiones png, gif, jpg, jpeg.</li> <li>• <b>Título de la imagen:</b> Opcional. Campo de texto. Permite todos los caracteres. Longitud máxima 1024. Este campo debe mostrarse al hacer clic en el botón Subir al servidor.</li> <li>• <b>Texto alternativo:</b> Campo de texto, opcional. Admite todos los caracteres. Este campo</li> </ul>	

<p>se muestra cuando se presiona el botón Subir al servidor.</p> <ul style="list-style-type: none"> <li>• <b>Resumen:</b> Obligatorio. Área de texto. Permite todos los caracteres.</li> <li>• <b>Descripción:</b> Obligatorio. Área de texto. Admite todos los caracteres.</li> <li>• <b>Fecha:</b> Calendario desplegable, obligatorio. Formato DD/MM/AAAA. Valor por defecto (Fecha actual).</li> <li>• <b>Autor:</b> Campo de texto, opcional. Admite todos los caracteres. Longitud máxima 255 caracteres.</li> <li>• <b>Imágenes/Añadir archivo nuevo:</b> Campo de tipo archivo, opcional. Admite archivos de hasta 512 KB y con los formatos png, gif, jpeg y jpg.</li> <li>• <b>Fuente:</b> Opcional. Campo de texto. Permite todos los caracteres. Longitud máxima 255.</li> <li>• <b>Publicado:</b> Opcional. Campo de selección. Valor por defecto (marcado).</li> </ul>
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. Si el usuario introduce los datos de forma correcta, el sistema emitirá un mensaje confirmando la culminación exitosa de la acción.</li> <li>2. Si el usuario introduce los datos de forma incorrecta, el sistema emitirá un mensaje notificando el error.</li> <li>3. Si el usuario deja campos obligatorios vacíos, el sistema emitirá un mensaje notificando el error.</li> </ol>
<p><b>Prototipo elemental de interfaz gráfica de usuario:</b> No aplica.</p>

Tabla 22 Editar noticia

<b>Número:</b> 22	<b>Nombre del requisito:</b> Editar noticia
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgo	<b>Tiempo Real:</b>
<b>Descripción:</b>	

Los usuarios autenticados con privilegios de administración podrán **editar** noticias en el sistema, para ello deben llenar los siguientes campos:

- **Título:** Obligatorio. Campo de texto. Permite todos los caracteres. Longitud máxima 255.
- **Imagen/Añadir archivo nuevo:** Obligatorio. Campo de tipo archivo. Tamaño máximo 512 KB. Permite solo los archivos con las extensiones png, gif, jpg, jpeg.
- **Título de la imagen:** Opcional. Campo de texto. Permite todos los caracteres. Longitud máxima 1024. Este campo debe mostrarse al hacer clic en el botón Subir al servidor.
- **Texto alternativo:** Campo de texto, opcional. Admite todos los caracteres. Este campo se muestra cuando se presiona el botón Subir al servidor.
- **Resumen:** Obligatorio. Área de texto. Permite todos los caracteres.
- **Descripción:** Obligatorio. Área de texto. Admite todos los caracteres.
- **Fecha:** Calendario desplegable, obligatorio. Formato DD/MM/AAAA. Valor por defecto (Fecha actual).
- **Autor:** Campo de texto, opcional. Admite todos los caracteres. Longitud máxima 255 caracteres.
- **Imágenes/Añadir archivo nuevo:** Campo de tipo archivo, opcional. Admite archivos de hasta 512 KB y con los formatos png, gif, jpeg y jpg.
- **Fuente:** Opcional. Campo de texto. Permite todos los caracteres. Longitud máxima 255.
- **Publicado:** Opcional. Campo de selección. Valor por defecto (marcado).

**Observaciones:**

1. Si el usuario introduce los datos de forma correcta, el sistema emitirá un mensaje confirmando la culminación exitosa de la acción.
2. Si el usuario introduce los datos de forma incorrecta, el sistema emitirá un mensaje notificando el error.
3. Si el usuario deja campos obligatorios vacíos, el sistema emitirá un mensaje notificando el error.

**Prototipo elemental de interfaz gráfica de usuario:** No aplica.

Tabla 20 Mostrar noticia.

<b>Número:</b> 23	<b>Nombre del requisito:</b> Mostrar noticia
<b>Programador:</b> Daylin Hernández Pérez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2
<b>Riesgo en Desarrollo:</b> Gespro/Plan de riesgo	<b>Tiempo Real:</b>
<p><b>Descripción:</b></p> <p>Los usuarios autenticados pueden visualizar las noticias registradas en el sistema, en el módulo Contenido de la interfaz de administración. Además, se pueden visualizar en la sección de Noticias de la página principal. De estas se muestran, un resumen de las tres últimas publicadas y un enlace para ver una página con todas las noticias. Para ver la noticia ampliada se selecciona la deseada y el sistema muestra una ventana con todo el contenido.</p>	
<p><b>Observaciones:</b></p> <p>1. El sistema debe mostrar todas las noticias al seleccionar el enlace Ver todas las noticias.</p>	
<p><b>Prototipo elemental de interfaz gráfica de usuario:</b></p> <p>Arquitectura de Información/ Sección Noticia</p>	

## Figuras

Diagrama de clases del diseño para historia de usuario.

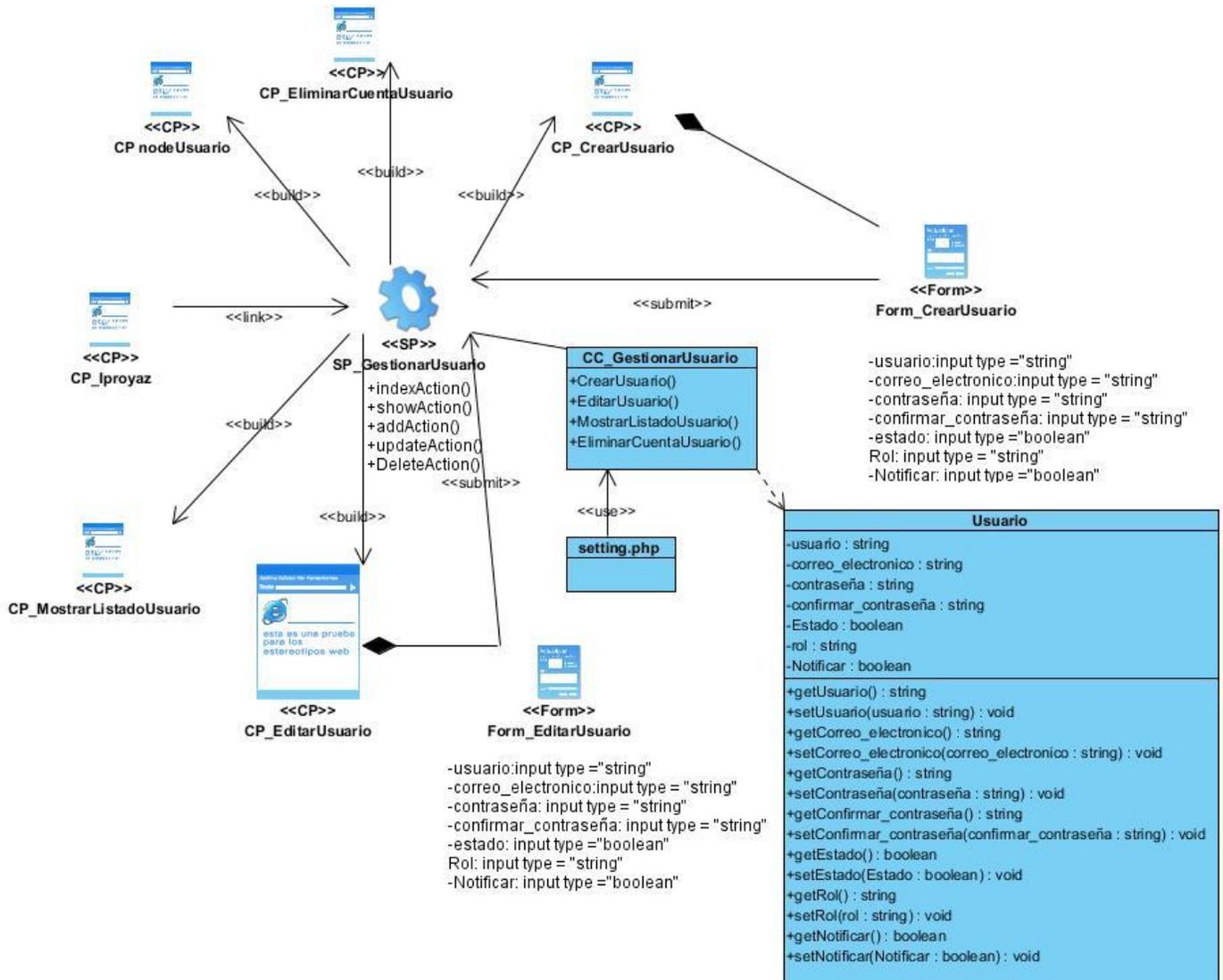


Figura: 11 Gestionar usuario



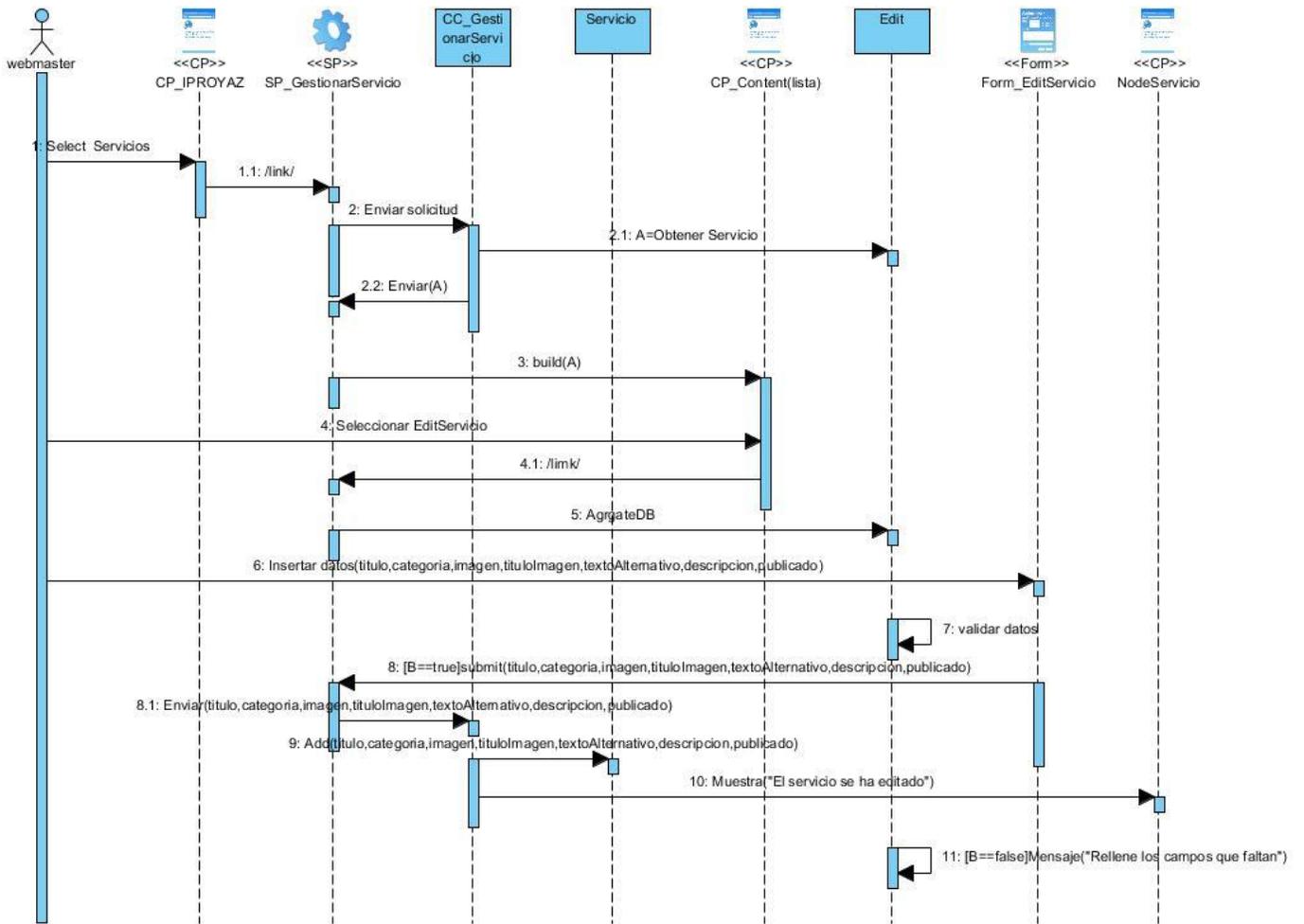


Figura: 13 Editar servicio

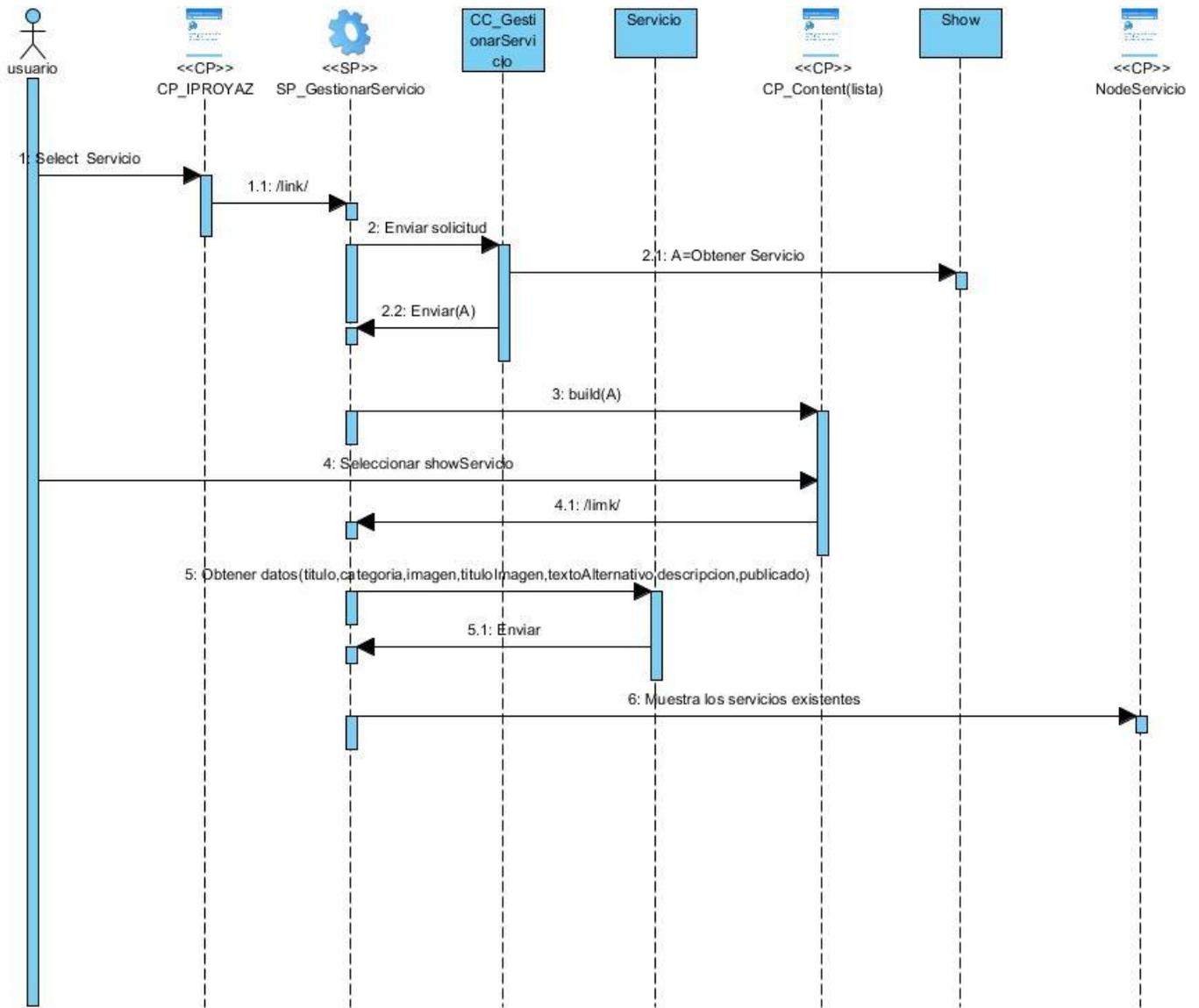


Figura: 14 Mostrar servicio

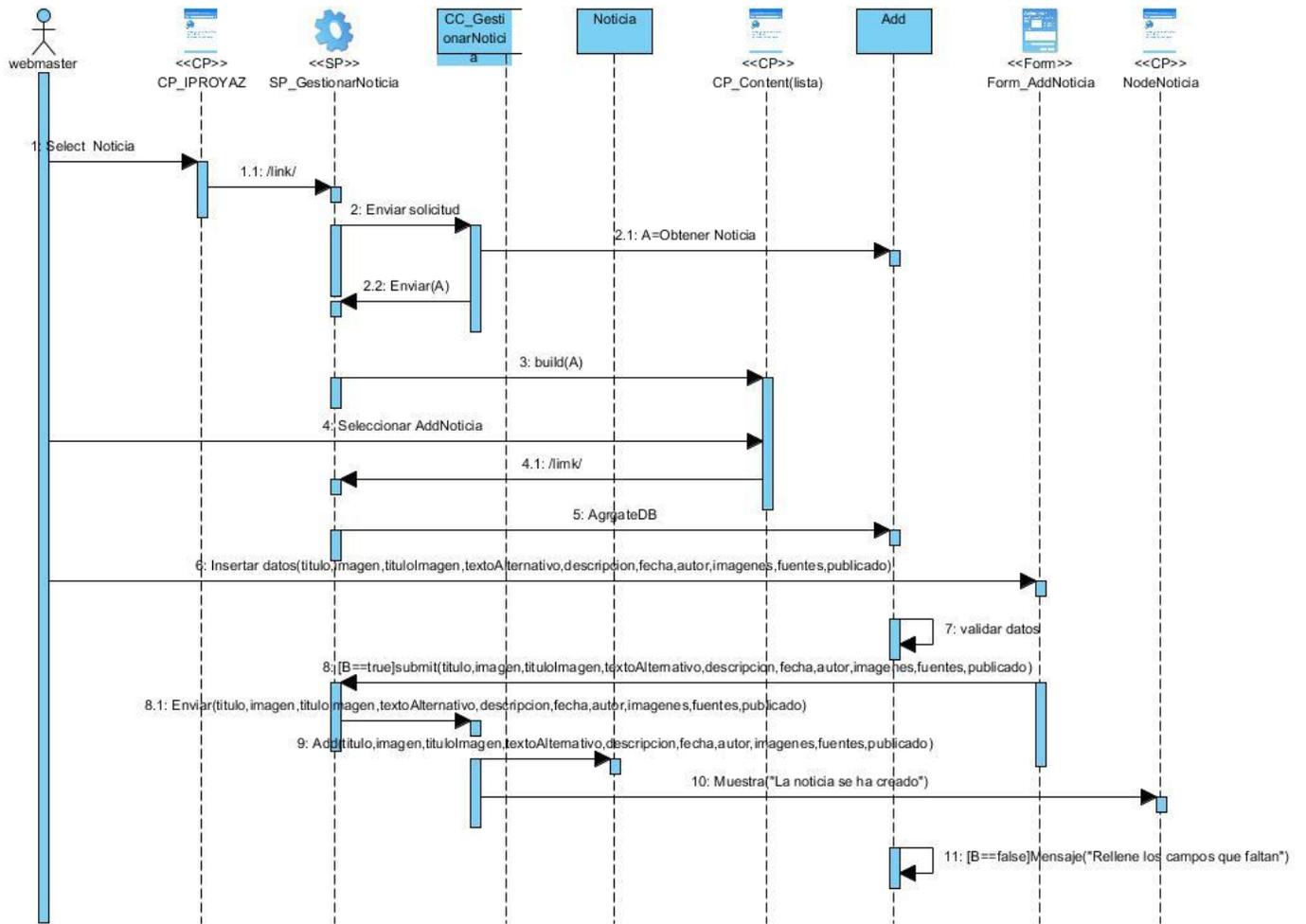


Figura: 15 Crear noticia

Casos de Prueba

Escenario	Descripción	Usuario	Dirección de correo electrónico	Contraseña	Confirmar contraseña	Estado	Roles	Notificar al usuario acerca de su nueva cuenta	Respuesta del sistema	Flujo central
EC 2.1 Insertar usuario de forma correcta.	El sistema inserta un usuario de forma correcta.	V usuariox	V usuariox@u ci.cu	V Contraseña 123*.	V Contraseña 123*.	V activo	V editor	V Seleccionad o	El sistema adiciona el usuario y muestra el siguiente mensaje: <b>"Un mensaje de bienvenida con instrucciones adicionales ha sido enviado al nuevo usuario usuariox."</b> Cuando el campo <b>Notificar al</b> Mensaje del correo electrónico: Dirección: IPROYAZ Asunto: Un administrador ha creado una cuenta para usted en IPROYAZ Mensaje: "usuarioX,	El usuario accede al bloque superior 1.- Selecciona la opción: <b>Administrar/Usuarios</b> . 2.- El sistema muestra el listado de los usuarios existentes. 3.-El usuario selecciona la opción: <b>"Añadir usuario"</b> . 4.- El sistema habilita un formulario para introducir la información. 5.-El usuario introduce la información y presiona el botón: <b>"Crear nueva cuenta"</b> .
EC 2.2 Insertar usuario de forma incorrecta.	El sistema no inserta un usuario de forma incorrecta.	I User,1	I usuario.sub dominios.do minio	I vb	I kguk	NA	NA	NA	El sistema no adiciona al usuario y emite el siguiente mensaje de error: <b>"La dirección de correo electrónico Dircorreo.cu no es válido."</b> <b>"El nombre de usuario contiene un carácter no permitido."</b>	
EC 2.3 Insertar usuario dejando	El sistema no inserta un usuario dejando	I	I	I	I	NA	I	NA	El sistema no adiciona al usuario y emite los siguientes mensajes de error: <b>"El campo Usuario es obligatorio."</b> <b>"El campo Contraseña es obligatorio."</b> <b>"El campo Roles es obligatorio"</b> <b>"El campo Dirección de correo electrónico es obligatorio"</b>	
EC 2.4 Insertar usuario con valores existentes.	El sistema no inserta un usuario con valores existentes.	I Administrador	I admin@cu	NA	NA	NA	NA	NA	El sistema no adiciona al usuario y emite los siguientes mensajes de error: <b>"El nombre de usuario nombreuser ya está en uso."</b> <b>" La dirección de correo Dirreccioncorreo ya está en uso."</b>	
EC 2.5 Cancelar operación.	El sistema no realiza ninguna operación.	NA	NA	NA	NA	NA	NA	NA	No realiza ninguna operación y regresa a la página anterior.	Para cancelar la operación de crear el contenido, el usuario sigue el flujo descrito en el escenario anterior y presiona el botón <b>"Cancelar"</b> .

Figura: 16 Crear usuario

Escenario	Descripción	Usuario	Contraseña actual	Dirección de correo electrónico	Contraseña	Confirmar Contraseña	Roles	Estado	Respuesta del sistema	Flujo central
EC 3.1 Editar usuario de forma correcta.	El sistema edita un usuario de forma correcta.	V usuarioX	V Contraseña 123*	V usuario@ucj.cu	V Contraseña 123**	V Contraseña 123**	V webmaster	V activo/bloqueado	El sistema edita el usuario y muestra el siguiente mensaje: "Se han guardado los cambios." Si el usuario desea modificar su	Se puede editar por las siguientes vías: El usuario accede al bloque superior: 1.- Selecciona la opción: <b>Administrar/Usuarios</b> .
EC 3.2 Editar usuario de forma incorrecta.	El sistema edita un usuario de forma incorrecta.	I User,1	V Contraseña 123*	I usuario.subdominios.dominio	I 1234	NA 123	NA	NA	El sistema no edita al usuario y emite el siguiente mensaje de error: "La dirección de correo electrónico Dircorreo.cu no es válido." "El nombre de usuario contiene un carácter	1.1.-El sistema muestra un listado con los usuarios existentes. 1.2.-El usuario selecciona aquel usuario que desea modificar y presiona la operación "Editar" 1.3.-El sistema muestra un formulario para modificar la información. 1.4.-El usuario modifica la información y presiona el botón: "Guardar".
EC 3.3 Editar usuario dejando campos vacíos.	El sistema no edita un usuario dejando campos obligatorios vacíos.	I	NA	I	NA	NA	I	NA	El sistema no adiciona al usuario y emite los siguientes mensajes de error: "El campo Usuario es obligatorio." "El campo Roles es obligatorio." "El campo Dirección de correo	2.-El usuario selecciona <b>usuarioX/Editar perfil o usuarioX/Ver perfil/ Editar</b> que se encuentra en el bloque superior. 2.1.- El sistema muestra un formulario para modificar la información. 2.2.- El usuario modifica la información y presiona el botón:
EC 3.4 Editar usuario con valores existentes.	El sistema no edita un usuario con valores existentes.	I administrador	NA	I admin@cu	NA	NA	NA	NA	El sistema no adiciona al usuario y emite los siguientes mensajes de error: " El nombre de usuario nombreuser ya está en uso." "La dirección de correo Dirreccioncorreo va está en uso "	Para cancelar la operación de editar el contenido, el usuario elige alguna de las vías descritas en el escenario anterior y presiona
EC 3.5 Cancelar operación.	El sistema no realiza ninguna operación.	NA	NA	NA	NA	NA	NA	NA	No realiza ninguna operación y regresa a la página anterior.	

Figura: 17 Editar usuario

## *Tablas y figuras*

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 5.1 Mostrar listado de usuario.	El sistema muestra el listado de usuarios.	El sistema muestra el listado de los usuarios existentes. De cada usuario muestra Usuario, Estado, Roles , Miembro desde hace, Último acceso y Operaciones.	El usuario accede al bloque superior. 1.-Selecciona la opción: <b>Administrar/Usuarios</b> . 1.2.-El sistema muestra un listado con los usuarios existentes.
EC 5.2 Mostrar listado de usuario vacío.	El sistema muestra el listado de usuarios vacío.	<b>No hay usuarios disponibles.</b>	

*Figura: 18 Mostrar listado de usuario*

---

**Encuesta**

Estimado especialista, se necesita su colaboración para conocer su valoración y grado de satisfacción durante al proceso de prueba del sistema a los portales e intranets desarrolladas en el centro. Esto ayudará a la mejoría de las prestaciones de acuerdo a sus necesidades.

1- ¿Qué rol desempeña en el centro CIDI?

Jefe de Proyecto

Analista

Desarrollador

Probador

2- ¿Ha participado en más de un proyecto al mismo tiempo?

Si ¿Por qué? \_\_\_\_\_

No

3- ¿Qué actividades realiza durante el proceso de prueba del sistema?

\_\_\_\_\_

4- ¿Cuenta con las herramientas necesarias para la aplicación de las pruebas?

Si

No ¿Cómo le afecta a la hora de realizar las pruebas?

5- Mediante la ejecución de los diferentes tipos de prueba se pueden encontrar diversos errores que se registran en un documento como no conformidad. ¿Considera que las detecciones de estas no conformidades son elevadas en cada prueba?

Si

No

¿A qué se debe esto?

\_\_\_\_\_

¿En que se basa principalmente para determinar una no conformidad?

\_\_\_\_\_

6- ¿Una vez ejecutadas las pruebas como se procede con los resultados obtenidos?

7- ¿Cuenta con la experiencia suficiente para realizar el proceso de prueba del sistema?

- Si  
 No

8- Teniendo en cuenta que los proyectos cuentan con un tiempo de duración que debe ser cumplido y que el aseguramiento de la calidad del sistema desde el punto de vista funcional es primordial. ¿Considera que el hecho de realizar el proceso de prueba del sistema manualmente implica un exceso de esfuerzo dedicado a ello?

- Si  
 No

9- ¿Considera que la documentación que se emplea mediante el proceso de prueba del sistema cuenta con la calidad requerida?

- Si  
 No