

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



AUTOMATIZACIÓN DE PLATAFORMA DE GESTIÓN
DOCUMENTARÍA VIRTUAL PARA LA COMISIÓN PERMANENTE
DE FISCALIZACIÓN – UNA - PUNO, 2017

TESIS

PRESENTADA POR:

Bach. MILTON VLADIMIR MAMANI CALISAYA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO

PUNO – PERÚ

2017

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

**Automatización de plataforma de gestión documentaria virtual
para la Comisión Permanente de Fiscalización – UNA Puno, 2017**

TESIS PRESENTADA POR:

Bach. MILTON VLADIMIR MAMANI CALISAYA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO




APROBADA POR EL JURADO REVISOR CONFORMADO POR:

PRESIDENTE

:


M.Sc. ERNESTO NAYER TUMI FIGUEROA

PRIMER MIEMBRO

:


Dr. LEONEL COYLA IDME

SEGUNDO MIEMBRO

:

Dr. REYNALDO SUCARI LEON

DIRECTOR / ASESOR

:


Dr. BERNABÉ CANQUI FLORES

Área

:

Informática

Tema

:

Sistema de Información

Fecha de Sustentación

:

15/12/2017

DEDICATORIA

Y pensar que todo empezó ayer, como un anhelo; que día a día fue tomando fuerza y hoy se hace realidad, veo hoy el fruto de mi dedicación y mi esfuerzo realizado, en esta hermosa carrera, por lo que dedico este gran logro a:

A mis padres Julián y Juana con mucho cariño y respeto, la cual agradezco por su sacrificio, apoyo incondicional, confianza, moral y económico. A mis hermanos por permitirme este logro a pesar de alguna de nuestras diferencias.

A mi gran Amor Yasbel, la ayuda que me has brindado ha sido sumamente importante, ya que estuviste conmigo en los momentos más complicados, no fue sencillo, pero siempre fuiste muy alentadora recordándomela que lo lograría. Gracias.

Milton Vladimir.

AGRADECIMIENTO

A la Facultad de Ingeniería Estadística e Informática de la Universidad Nacional del Altiplano - Puno, por brindarme la oportunidad de poder cumplir mi sueño.

Un agradecimiento muy grande también a los jurados M.Sc. Ernesto Nayer Tumi Figueroa, Dr. Leonel Coyla Idme, Dr. Reynaldo Sucari León y en especial a mi director de tesis Dr. Bernabe Canqui Flores a que con su dirección y apoyo eficaz hicieron posible la conclusión de la presente tesis.

ÍNDICE GENERAL

RESUMEN	11
ABSTRACT	12
CAPITULO I INTRODUCCIÓN	13
1.1. Planteamiento del Problema	14
1.2. Formulación del Problema	15
1.3. Hipótesis de la Investigación	16
1.4. Justificación de la Investigación	16
1.5. Objetivos de la Investigación.....	17
Objetivo General	17
Objetivos Específicos	18
CAPITULO II REVISIÓN DE LITERATURA	19
2.1. Marco Teórico	19
2.1.1. Antecedentes de la Investigación	19
2.2. Marco Conceptual	21
2.2.1.Objetivos y Funciones de la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano.....	21
2.2.2.Sistema de Información	22
2.2.3.Aplicaciones Web	23
2.2.4.Maquetado Web	24
2.2.5 Lenguaje De Programación.....	25
2.2.6 Gestor De Base De Datos	28
2.2.7 Biblioteca Javascript.....	29
2.2.8.Tecnología AJAX.....	30
2.2.9 Biblioteca de Javascript JQuery.....	31
2.2.10 Análisis y Diseño Orientado a Objetos.....	32

2.2.11. Programación Orientada a Objetos	33
2.2.12. Elementos de la POO	34
2.2.13. Ingeniería de Software	34
2.2.14. Modelos de Desarrollo de Software	35
2.2.15. Algunos de los Modelos de Desarrollo	36
2.2.16. Metodología de Desarrollo de Software	36
2.2.17. XP	37
2.2.18. Fases de la Metodología XP	38
2.2.19. Patrones de Diseño	42
2.2.20. Versionamiento Semántico	44
2.2.21. Diseño Web Adaptable	45
2.2.22. Ventajas e Inconvenientes del Empleo de Responsive Web Design	46
CAPITULO III MATERIALES Y MÉTODOS	48
3.1. Ubicación Geográfica del Estudio	48
3.2. Población y Muestra del Estudio	48
3.3. Operacionalización De Variables	49
3.4. Recopilación de la Información	49
3.5. Modelo y Metodología de Desarrollo de Software	50
3.6. Análisis de Planificación para el Desarrollo	53
3.7. Diagrama del Proceso Detallado	54
CAPITULO IV RESULTADOS Y DISCUSIÓN	58
4.1. Resultados	58
4.2. Diseño de la Plataforma	60
4.3. Modelamiento de la Plataforma Mediante UML7	60
4.4. Versiones e Iteraciones	61

4.5.Codificación.....	65
4.6.Pruebas.....	65
4.7.Instalación.....	66
4.8.Resultados de la Prueba Según El ISO – 9126.....	66
4.9.Gráficos de Satisfacción de Usuario.....	67
4.10.Contraste de Hipótesis.....	68
4.11.Decisión.....	70
CAPITULO V CONCLUSIONES.....	71
CAPITULO VI RECOMENDACIONES.....	72
CAPITULO VII REFERENCIAS.....	73
ANEXOS.....	77

ÍNDICE DE FIGURAS

FIGURA N° 1 Componentes de un Sistema de Información.....	23
FIGURA N° 2 Interacción de Componentes del patrón de diseño MVC.....	43
FIGURA N° 3 Comparación Tiempo y Costo de Metodologías de desarrollo. .	51
FIGURA N° 4 Diagrama del proceso de registro y activación de usuarios.....	54
FIGURA N° 5 Diagrama del proceso de los trámites	54

ÍNDICE DE TABLAS

TABLA N° 1 Ventajas y Desventajas de Lenguajes de Programación	27
TABLA N° 2 Tabla comparativa de Ventajas y Desventajas de Gestores de Bases de Datos	29
TABLA N° 3 Ventajas y Desventajas de Patrones de Diseño MVC y MVP	44
TABLA N° 4 Operacionalización de variables	49
TABLA N° 5 Diferencias entre Metodologías Ágiles y Metodologías Tradicionales	50
TABLA N° 6 Backlog	53
TABLA N° 7 Cuadro de decisiones.....	66
TABLA N° 8 Ficha de Evaluación ISO - 9126.....	78
TABLA N° 9 Escala valorativa (Escala de Likert)	81
TABLA N° 10 Cuadro de decisiones ISO 9126.....	81
TABLA N° 11 FICHAS DE EVALUACIÓN REALIZADAS A LOS USUARIOS ISO 9126.....	84
TABLA N° 12 FICHAS DE EVALUACIÓN REALIZADAS A LOS USUARIOS ISO 9126.....	89

ÍNDICE DE ACRÓNIMOS

PLAGEDO	Plataforma de Gestión Documentaria
ISO	Organización de Estándares Internacionales
MVC	Modelo Vista Controlador
HTML	Lenguaje de Marcado de Hiper Texto
PHP	Pre-procesador de hiper-texto
AJAX	Acrónimo de Asynchronous JavaScript And XML
Jquery	Biblioteca multiplataforma de JavaScript
Sublime Text	Sublime Text es un editor de texto y código
MySQL	Sistema de gestión de bases de datos relacionales
GPL	Licencia Pública General
XAMPP	Servidor web multi plataforma constituido por un servidor servidor
Scrum	Metodología de desarrollo agil e incremental
eXtreme Programming	La programación extrema
WORKBENCH	Herramienta visual de diseño de bases de datos
CodeIgniter	Marco de desarrollo [framework] rápido para PHP

RESUMEN

Los trámites documentarios se han tornado en procedimientos deficientes provocando confusiones de datos y reincidencias de información y el hecho de no contar con esta plataforma que ayuden a automatizar los procedimientos de la oficina de Comisión Permanente de Fiscalización. La investigación tuvo como objetivo automatizar una plataforma de gestión documentaria virtual para la gestión documentaria de la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano, de esta manera la herramienta presentada automatiza los procesos de gestión de documentaria, fiscalización interna, emisión y recepción de documentos y reportes necesarios de manera eficiente. La plataforma se basó en la metodología de desarrollo eXtreme Programming (XP) y la metodología de desarrollo ágil SCRUM, que fue la que se adecuo por tratarse de una metodología ágil y que permite la programación a bajo costo. Para el modelado de la plataforma, se ha utilizado la metodología UML (Unified Modeling Lenguaje) y para la validación del sistema se hizo el uso de la ficha de evaluación de calidad del software estándar ISO- 9126, evaluación realizada a los usuarios directos de la Comisión Permanente de Fiscalización. Por lo que se concluye la automatización la mejora de los procesos que se llevan a cabo en la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano – Puno, también se validó el sistema bajo el estándar de evaluación de calidad del software estándar ISO- 9126, el cual indica el cumplimiento de los requisitos con un calificativo promedio de 110 puntos según la escala dada por el estándar.

Palabras clave: Automatización, Fiscalización, Framework, Gestión virtual, Scrum.

ABSTRACT

Documentary procedures have become deficient procedures causing data confusion and information reoccurrence and the fact of not having this platform to help automate the procedures of the Office of Permanent Control Commission. The objective of the research was to automate a virtual document management platform for the document management of the Permanent Commission of Control of the National University of the Altiplano, in this way the presented tool automates the processes of documentary management, internal control, emission and reception of documents and necessary reports efficiently. The platform was based on the development methodology eXtreme Programming (XP) and the agile development methodology SCRUM, which was the one that was adapted because it is an agile methodology that allows programming at low cost. For the modeling of the platform, the UML (Unified Modeling Language) methodology was used and for the validation of the system the use of the quality evaluation form of the standard software ISO-9126 was made, an evaluation made to the direct users of the Permanent Commission of Control. So the automation concludes the improvement of the processes that are carried out in the Permanent Commission of Control of the National University of the Altiplano - Puno, also the system was validated under the quality evaluation standard of the ISO-9126 standard software , which indicates compliance with the requirements with an average qualification of 110 points according to the scale given by the standard.

Keywords: Automation, Inspection, Framework, Virtual management, Scrum.

CAPITULO I

INTRODUCCIÓN

El presente trabajo propone el diseño e implementación de una Plataforma Virtual para la recepción y control de los diferentes tipos de documentos enviados y recepcionados por la Comisión Permanente de Fiscalización, enfocando el trabajo a la optimización de los procesos de trámites, registros, seguimientos, y finalización de estos documentos basados en los hechos rutinarios.

Para el diseño de la misma se realizó un proceso previo la recolección de información de las autoridades responsables de las oficinas de gestión de rectorado que participan en estos procesos. Una vez recolectada la información se procedió a la implementación de la plataforma para lo cual se usó tecnologías web como es PHP, MYSQL para los módulos de operatividad de la plataforma, HTML, CSS, JavaScript para la interacción con los usuarios de la plataforma.

Ya para poner y aplicar la plataforma en los procesos de trámites, control y finalización del documento tramitados por estas autoridades, se realizaron capacitaciones previas para el uso de la plataforma, las cuales ayudaron a las

autoridades de las oficinas de la gestión del rectorado de optimizar el tiempo de aprendizaje sobre el uso de la plataforma.

En el Estado II se presenta el documento en la cual está incluida los tipos de documentos con el respectivo contenido del documento las cuales podrán ser enviados a las oficinas correspondientes de destino de acuerdo al organigrama de la Universidad Nacional del Altiplano – Puno. En el estado II se tiene el documento creado y enviando al destino del responsable de la oficina, la cual tendrá en su bandeja, podrá leerla y dar acciones sobre ella. En el estado III el jefe de la oficina podrá derivar o finalizar el trámite.

En el Capítulo I se presenta la Introducción en el cual están incluidos el problema, descripción del problema, la formulación del problema, los objetivos e hipótesis. En el Capítulo II se tiene el Marco Teórico el cual está conformado por: los antecedentes, base teórica, definición de términos básicos y la 11 operacionalización de variables. En el Capítulo III se muestra la metodología. En el Capítulo IV están los resultados. Y para finalizar se encuentra las conclusiones, recomendaciones y sugerencias.

1.1. Planteamiento del Problema

En la actualidad los sistemas de información que funcionan en los institutos y en las universidades son muy utilizados, esto es gracias a su facilidad de acceso desde la nube, control, organización y como también se puede interactuar con múltiples usuarios concurrentes, esto es posible ya que solo requiere una aplicación para la gestión y administración de dichos documentos.

La Comisión Permanente de Fiscalización no cuenta con una plataforma que permita el control de la información de los diferentes documentos emitidos y recepcionados de las distintas autoridades (gestión de documentos), es por esta razón que planteamos la siguiente solución la “Automatización de Plataforma virtual para la Comisión Permanente de Fiscalización – Universidad Nacional del Altiplano Puno”, plataforma permitirá el apoyo operacional y el control de la gestión administrativa, con el fin de agilizar el tiempo de respuesta en la emisión de documentos como, expedientes, resoluciones u otros documentos la Comisión Permanente de Fiscalización.

La plataforma contará con el acceso desde la web, el mismo que tendrá un módulo de seguridad, en el cual se administrará las cuentas y las claves necesarias para el acceso a la plataforma mediante las aplicaciones desarrolladas según necesidad, con esto podemos transparentar los trámites de cada usuario.

Sin esta Plataforma se ocasiona confusión de datos, reincidencia de información, pérdida de datos al momento de realizar el intercambio de información al utilizar dispositivos de almacenamiento y archivos físicos. De acuerdo a la problemática anterior se formula la siguiente interrogante:

1.2. Formulación del Problema

¿Mejorará la Plataforma de Gestión Documentaría los procesos de trámites y la administración documentaría en la Universidad Nacional del Altiplano?

1.3. Hipótesis de la Investigación

La implementación de la plataforma de gestión documentaria virtual mejorará la gestión documentaria de la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano Puno, 2017.

1.4. Justificación de la Investigación

La Comisión Permanente de Fiscalización no cuenta con una plataforma para dichos procesos de la información de los diferentes documentos emitidos y recepcionados de las distintas autoridades, es por esta razón que planteamos la siguiente solución, la “Automatización de Plataforma virtual para la Comisión Permanente de Fiscalización – Universidad Nacional del Altiplano Puno”, la plataforma permitirá la gestión de documentos y administrativa de los diferentes procesos que elabora la Comisión Permanente de Fiscalización.

Para el diseño de la misma se realizó un proceso previo la recolección de información de las autoridades responsables de las oficinas de la comisión permanente de fiscalización que participan en estos procesos. Una vez recolectada la información se procedió a la implementación de la plataforma para lo cual se usó tecnologías web como es PHP, MYSQL para los módulos de operatividad de la plataforma, HTML, CSS, JavaScript para la interacción con los usuarios de la plataforma.

La plataforma web para los trámites de documentos administrativos de la UNA Puno 2017, busca gestionar los documentos de tal manera garantizar la seguridad de toda la información que se obtiene de estos

procesos de la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano Puno, además de poder monitorear(fiscalizar) todo este procedimiento en forma detallada, transparente y automatizada a cada uno de los usuarios de la plataforma.

La Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano para hacer la organización de los trámites cuenta en cada una de las dependencias de los miembros y jefes, la cual tiene como funciones de vigilar la gestión administrativa, actualmente los procesos que involucran los documentos, tienen un registro a través de documentos creados por la plataforma, por lo que se propuso una solución de acuerdo a la actualidad y con tecnología que está a nuestro alcance.

Con la automatización se permite mejorar los problemas mencionados de la calidad de servicio al usuario cumpliendo con las normas establecidas para el manejo de la gestión de los documentos, se requiere empezar solucionando las actualizaciones de los perfiles de cada miembro de las oficinas ya que es de gran interés para el usuario porque se podrá realizar un buen reporte que será de mucha importancia para la unidad. Teniendo en cuenta las necesidades y expectativas.

1.5. Objetivos de la Investigación

Objetivo General

Desarrollar e implementar la plataforma de gestión documentaria virtual que mejore la gestión de los diferentes procesos documentarios de la Comisión Permanente de Fiscalización de la Universidad Nacional del

Altiplano Puno 2017.

Objetivos Específicos

- ✓ Implementar la plataforma basándose en los requerimientos de la Comisión Permanente de Fiscalización.
- ✓ Implementar la seguridad adecuada del acceso a la información.
- ✓ Determinar las ventajas de la Plataforma con su implementación.
- ✓ Implementar la categorización de la información en las bases de datos para su posterior preservación.

CAPITULO II

REVISIÓN DE LITERATURA

2.1. Marco Teórico

2.1.1. Antecedentes de la Investigación

Tesis Internacionales

Mendoza (2011) En su investigación su objetivo general señala: Implementar un sistema automatizado que optimice la gestión de los procesos administrativos del área de servicios médicos. Concluye que la comunicación con el cliente represento una clave fundamental para poder validar los requisitos y cumplir con sus necesidades o requerimientos que se da a partir de cada una de las iteraciones a lo largo del proceso de desarrollo, facilitando la labor de muchas tareas e impactando de manera positiva en el tiempo como análisis y diseño.

Tesis Nacionales

Aranda Font (2013) Con los sistemas de gestión documental se asegura la conservación de los documentos, las evidencias, un servicio

más eficiente, el control y organización de la documentación que conllevará a obtener la mejora continua de la institución, la excelencia, que no es más que la calidad.

Pezo Villacorta (2014) Cuando se implanto Sistema de Tramite Documentarlo se hicieron las pruebas respectivas con la misma, y se tuvo una gran reducción con más del 70% en la reducción del tiempo en los procesos realizados por la misma, ya que esto beneficia en la forma de manejar a los usuarios con las entradas de los documentos.

Tesis Locales

Torres Cruz (2016) Se ha planificado los tiempos para reducir los procesos redundantes y automatizarlos como la generación de memorandos, generación automática de constancias y envío de notificaciones por correo electrónico, una vez realizado el registro del proyecto, así como la aprobación del director/asesor, teóricamente puede realizarse todos los trámites en menos de una semana, si los jurados y directores están atentos y completamente diestros en la plataforma.

Jihuallanca Ccoa (2016) El tiempo de demora redujo notablemente con la implementación del sistema de consulta de perfiles de la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno. Se realizó un análisis estadístico utilizando la prueba t-student para datos relacionados:

2.2. Marco Conceptual

2.2.1. Objetivos y Funciones de la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano

Art.54°. Constituyen órganos especiales y consultivos de la Asamblea Universitaria, las comisiones permanentes, especiales o transitorias que se eligen en ella, siendo estas las siguientes:

- ✓ Comité Electoral Universitario
- ✓ Tribunal de Honor Universitario
- ✓ Comisión Permanente de Fiscalización
- ✓ Defensoría Universitaria
- ✓ Comisiones Especiales y transitorias

Art.58°. La Comisión Permanente de Fiscalización, es el órgano encargado de vigilar la gestión académica, investigativa, administrativa y económica de la Universidad, cuenta con amplias facultades para solicitar información a toda instancia interna de la Universidad. Está obligada a guardar la debida confidencialidad de la información proporcionada, bajo responsabilidad. Los miembros de esta comisión no podrán ser autoridades ni pertenecer a ningún órgano de gobierno, de ejecución, de asesoramiento o apoyo de la Universidad y asumirá la presidencia el profesor principal más antiguo en la categoría. Para su organización y cumplimiento de su comisión, elaborará su reglamento que entre otros aspectos deberá incluir, régimen de sesiones, periodicidad, reglas de votación, reglas de

aprobación de acuerdos, coordinaciones y responsabilidades.

- a. Fiscalizar la gestión académica de la Universidad Nacional del Altiplano – Puno.
- b. Fiscalizar la gestión investigativa de la Universidad Nacional del Altiplano – Puno.
- c. Fiscalizar la gestión administrativa de la Universidad Nacional del Altiplano – Puno.
- d. Fiscalizar la gestión económica de la Universidad Nacional del Altiplano – Puno.

2.2.2. Sistema de Información

Es el producto útil de la manipulación de datos organizados y estructurados que se emplean en las organizaciones para la toma de decisiones.

La información se produce a partir de los datos que son símbolos que describen un objeto, condición y situación.

Un sistema de información es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad u objetivo. (<http://es.wikipedia.org/>, 2013).

Figura N° 1 Componentes de un Sistema de Información.



Según los autores Laudon & Laudon, profesores de Administración de Empresas, un sistema de información es un organismo que recolecta, procesa, almacena y distribuye información. Son indispensables para ayudar a los gerentes a mantener ordenada su compañía, a analizar todo lo que por ella pasa y a crear nuevos productos que coloquen en un buen lugar a la organización. (<https://es.slideshare.net>, 2011).

2.2.3. Aplicaciones Web

Cobo (2005) Una aplicación web es cualquier aplicación que es accedida vía web por una red como internet o una intranet.

En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML5); confiándose en el navegador web para que reproduzca (renderice) la aplicación.

Una de las ventajas de las aplicaciones web cargadas desde internet

(u otra red) es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes. También la posibilidad de ser ejecutadas en múltiples plataformas.

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

2.2.4. Maquetado Web

Iglesias (2012) La maquetación se conoce como la distribución de los elementos en páginas web, algunos elementos, como encabezados, columnas, menús laterales, formularios entre otros.

Hace unos años, la maquetación de las páginas web se realizaba utilizando tablas (<table>). Una vez entendido este proceso podía resultar sencillo, aunque si no se dominaban las tablas, podía convertirse en algo tedioso. Además, algunos buscadores encontraban problemas al analizar la estructura de la página.

Actualmente, la maquetación con tablas ha caído en desuso, y se realiza utilizando capas (<div>), también llamadas divisiones o contenedores. La colocación de las capas en la página se realiza a través de CSS3. Esto permite, por ejemplo, que podamos pasar de un diseño con

un menú lateral a otro con el menú en la parte superior, sólo cambiando la hoja de estilos.

HTML (HyperTextMarkupLanguage)

Graham (1995) Es el lenguaje que todos los programas navegadores usan para presentar información en la World Wide Web (WWW). Este es un lenguaje muy sencillo que se basa en el uso de etiquetas, consistentes en un texto ASCII encerrado dentro de un par de paréntesis angulares (<...>). (Lenguaje

HTML 2000

Es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

2.2.5. Lenguaje De Programación

Spona (2010) Debido a que se requiere desarrollar una solución que no implique costos por adquisición de licencias y sea fácil de manejar, se ha optado por utilizar un lenguaje de programación orientado a objetos y que, adicionalmente, sea compatible con otras tecnologías de software libre.

PHP

Es un acrónimo recursivo que significa “PHP Hypertext Preprocessor”. Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor.

Características:

- ✓ Su código no necesita ser compilado para ejecutarse.
- ✓ Para su funcionamiento necesita tener instalado el servidor Apache o IIS con las librerías de PHP.
- ✓ La mayor parte de su sintaxis ha sido tomada de los lenguajes C, Java y Perl con algunas características específicas.
- ✓ No requiere de declaración de variables.
- ✓ Los archivos cuentan con la extensión php.
- ✓ Soporta en cierta medida las clases y la herencia.
- ✓ Es multiplataforma.
- ✓ Por su diseño dinámico, no puede ser compilado y es difícil de optimizar.
- ✓ Favorece la creación de código desordenado y complejo de mantener.

En la siguiente tabla se muestra una comparación que revela las ventajas y desventajas de algunos lenguajes de programación, según ciertas características comunes, el nivel de aprendizaje requerido, el nivel de experiencia del equipo, entre otras cosas.

Tabla N° 1 Ventajas y Desventajas de Lenguajes de Programación

Características	Ruby	PHP	Java	Python
Multiplataforma	Si	Si	Si	Si
Nivel de aprendizaje	Fácil	Fácil, Intermedio	Fácil	Fácil
Software libre	Si	Si	Si	Si
Documentación existente	Regular	Bastante	Bastante	Regular
Nivel de experiencia	-	Intermedio	Básico	-
Organización por capas	Muy difícil	Difícil	Fácil	Fácil
Extensiones	Si	Si	Si	Si
Rendimiento	Los scripts podrían resultar lentos cuando se trata con muchos datos.	Muy eficiente para aplicaciones web que demanden gran cantidad de solicitudes del servidor.	Penalización en tiempo debido a características propias del lenguaje.	Lentitud por ser un lenguaje interpretado

De los lenguajes de programación presentados, y a pesar que se requiere mayor aprendizaje, se ha optado por utilizar PHP como lenguaje de programación para el desarrollo de la solución. Esto se justifica primordialmente en el nivel de experiencia que tiene el equipo de desarrollo, la documentación existente que se puede encontrar y la facilidad que ofrece PHP para el desarrollo de aplicaciones en capas.

2.2.6. Gestor De Base De Datos

Duarte-García (2007) Considerando la cantidad de conexiones concurrentes y el tamaño de la aplicación, se propuso usar un gestor de base de datos relacional que brinde un buen rendimiento, que sea software libre y que se integre muy bien con las tecnologías antes mencionadas.

MySQL

Se trata de un sistema de gestión de bases de datos relacional, bajo la licencia GPL de GNU. Fue desarrollado con la idea de brindar mayor velocidad de respuesta, aunque se sacrificaron algunas características esenciales. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente.

Este gestor de bases de datos se puede considerar como el más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esto puede deberse adicionalmente a que existe gran cantidad de librerías y herramientas que permiten su uso a través de diferentes lenguajes de programación, además de su fácil instalación y configuración.

Se recomienda su uso en aplicaciones que no tengan una gran cantidad de accesos concurrentes a la base de datos, puesto que el rendimiento puede bajar considerablemente.

En la siguiente tabla se muestra una comparación que revela las ventajas y desventajas de algunas tecnologías.

Tabla N° 2 Tabla comparativa de Ventajas y Desventajas de Gestores de Bases de Datos

Característica	PostgreSQL	MySQL
Usabilidad	Buena	Muy buena
Robustez	Muy buena	Regular
Instalación y configuración	Regular	Fácil
Apariencia	Buena	Muy Buena
Nivel de experiencia	Básico	Intermedio
Velocidad de respuesta	Buena	Muy buena
Control de acceso y autenticación de usuarios	Difícil	Fácil
Soporte de triggers y procedimientos almacenados	Mejor	Deficiente
Consumo de recursos	Mucho	Poco

Por todo lo mostrado en las definiciones y el cuadro comparativo, se ha optado por utilizar MySQL ya que, aunque no ofrece robustez como PostgreSQL, brindaría el rendimiento adecuado para las pocas conexiones concurrentes que se plantea tiene la aplicación web.

2.2.7. Biblioteca Javascript

Gauchat (2012) Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el C.

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipo) y extendiendo su funcionalidad.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con un web se provee al lenguaje JavaScript de una implementación del DOM.

2.2.8. Tecnología AJAX

Powel (2008) AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Las tecnologías que forman AJAX son:

- ✓ XHTML y CSS, para crear una presentación basada en estándares.
- ✓ DOM, para la interacción y manipulación dinámica de la presentación.
- ✓ XML, XSLT y JSON, para el intercambio y la manipulación de información. XMLHttpRequest, para el intercambio asíncrono de información.
- ✓ JavaScript, para unir todas las demás tecnologías.

La característica fundamental de AJAX es permitir actualizar parte de una página con información que se encuentra en el servidor sin tener que refrescar completamente la página. De modo similar podemos enviar información al servidor. (Wikipedia).

2.2.9. Biblioteca de Javascript JQuery

Silva, M.S. (2008) Es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp

NYC. JQuery es la biblioteca de JavaScript más utilizada.

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. (<http://jquery.com/>)

2.2.10. Análisis y Diseño Orientado a Objetos

Suárez (2016) Detalla los siguientes puntos:

Objeto: Es aquello que tiene estado (Propiedades más valores), comportamientos (acciones y reacciones a mensajes) e identidad (propiedad que lo distingue de los demás objetos).

Clase: Es un conjunto de objetos que comparten una estructura y comportamiento común.

Clase de Objetos: Una clase de objetos describe un grupo de objetos con propiedades (atributos) similares, con relaciones comunes con otros y una semántica común. Los objetos de una clase comparten un propósito semántico común, más allá de los requisitos de la comunidad de atributos y de comportamiento. Cada objeto conoce su clase, la mayoría de los lenguajes de programación orientados a objetos puede determinar la clase de objeto en el momento de la ejecución. La clase de objetos es una propiedad implícita de sí mismo.

Orientado a Objetos: Superficialmente, el término “Orientado a Objetos” significa que el software se organiza como una colección de objetos discretos que contienen tanto estructuras de datos como un comportamiento.

El desarrollo orientado a objetos es una forma de pensar acerca del software basándose en abstracciones que existen en el mundo real, la palabra desarrollo hace alusión a la parte inicial del ciclo de vital del software (análisis, diseño e implementación).

Atributo: están asociados a clases y objetos, y estos describen la clase o el objeto de alguna manera.

2.2.11. Programación Orientada a Objetos

García-Beltrán et al. (2016) Define a POO como un paradigma de programación (o técnica de programación) que utiliza objetos e interacciones en el diseño de un sistema.

Está basado en varias técnicas, y presenta distintas características tales como herencia, abstracción, polimorfismo y encapsulamiento.

Sierra (1991) Indica que un lenguaje de programación que soporte programación orientada a objetos beneficia el desarrollo de software, ya que ofrece una forma más natural de desarrollo de modelos, utilizando clases.

2.2.12. Elementos de la POO

Clase: Una clase es un modelo que se utiliza para crear objetos que comparten un mismo comportamiento, estado e identidad.

Objeto: Es una entidad provista de métodos o mensajes a los cuales responde (comportamiento); atributos con valores concretos (estado); y propiedades (identidad).

Método: Es el algoritmo asociado a un objeto que indica la capacidad de lo que éste puede hacer.

Evento y Mensaje: Un evento es un suceso en el sistema mientras que un mensaje es la comunicación del suceso dirigida al objeto.

Propiedades y atributos: Las propiedades y atributos, son variables que contienen datos asociados a un objeto.

2.2.13. Ingeniería de Software

Pressman y Troya (1988) La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza.

En esta definición, existen dos frases clave:

Disciplina de la ingeniería. Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de

descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones.

Todos los aspectos de producción de software. La ingeniería del software no sólo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

2.2.14. Modelos de Desarrollo de Software

Pons, C., Giandini, R. S., & Pérez, G. (2010) Un modelo es una representación de la realidad en la que se toma en cuenta solo los detalles relevantes con la finalidad de obtener la solución de un problema.

Entonces un modelo del desarrollo del software es una representación abstracta de un proceso del software. Cada modelo de proceso representa un proceso desde una perspectiva particular, y así proporciona sólo información parcial sobre ese proceso. En esta sección, se introducen varios modelos de proceso muy generales (algunas veces llamados paradigmas de proceso) y se presentan desde una perspectiva arquitectónica.

Esto es, vemos el marco de trabajo del proceso, pero no los detalles de actividades específicas.

Estos modelos generales no son descripciones definitivas de los

procesos del software. Más bien, son abstracciones de los procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo de software. Puede pensarse en ellos como marcos de trabajo del proceso que pueden ser extendidos y adaptados para crear procesos más específicos de ingeniería del software.

2.2.15. Algunos de los Modelos de Desarrollo

El modelo en cascada. Considera las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etcétera.

Desarrollo evolutivo. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones abstractas. Este se refina basándose en las peticiones del cliente para producir un sistema que satisfaga sus necesidades.

Ingeniería del software basada en componentes. Este enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en integrar estos componentes en el sistema más que en desarrollarlos desde cero.

2.2.16. Metodología de Desarrollo de Software

Canós & Letelier (2012) Las metodologías imponen un proceso sobre el desarrollo de software con el fin de hacerlo más eficiente. Este

proceso tiene un fuerte énfasis en la planificación.

Las metodologías tradicionales han estado presentes durante mucho tiempo y no se han distinguido precisamente por ser muy exitosas. Es por ello que hacen su aparición metodologías ágiles, las cuales son estrategias de desarrollo de software centradas en los equipos y orientadas hacia prestaciones y hacia la entrega.

Entre las metodologías tradicionales más conocidas se encuentra RUP y por el lado de las ágiles sobresale la metodología de desarrollo de software XP.

2.2.17. XP

Burke et al. (2013) Define Extreme Programming como la metodología ágil más estudiada hasta el momento. Esta metodología se basa en el desarrollo de los casos de prueba e historias de usuario y se recomienda para aumentar la velocidad de desarrollo de un producto. Se basa en valores, principios y prácticas esenciales. Los cuatro valores son la comunicación, la simplicidad, la retroalimentación y la valentía.

Canós y Letelier (2012) Señala que XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de

negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

2.2.18. Fases de la Metodología XP

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Fase I: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo

tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto

tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

Fase III: Iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo.

Las iteraciones son también utilizadas para medir el progreso del proyecto.

Una iteración terminada sin errores es una medida clara de avance.

Fase IV: Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta

fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

Fase V: Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Fase VI: Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema.

Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

2.2.19. Patrones de Diseño

APA Aubry, C. (2012) Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. (<https://goo.gl/wS1Tm4>, 2015)

Dado que existe gran cantidad de patrones de diseño, a continuación, se describe uno de los patrones de diseño más conocidos.

Modelo – Vista – Controlador (MVC)

Bahit (2011) Modelo-Vista-Controlador (o en inglés, model-view-controller), que es uno de los tantos patrones de arquitectura de software.

MVC es el patrón de diseño arquitectural recomendado para aplicaciones interactivas. Entre sus características se tiene que separar los conceptos de diseño, y por lo tanto disminuye la duplicación de código, la centralización del control y hace que la aplicación sea más extensible.

Los componentes en los que se separan los datos de una aplicación, la interfaz de usuario y la lógica de control son los siguientes:

Modelo: Encapsula los datos y las funcionalidades del negocio. Accede a la capa de acceso a datos.

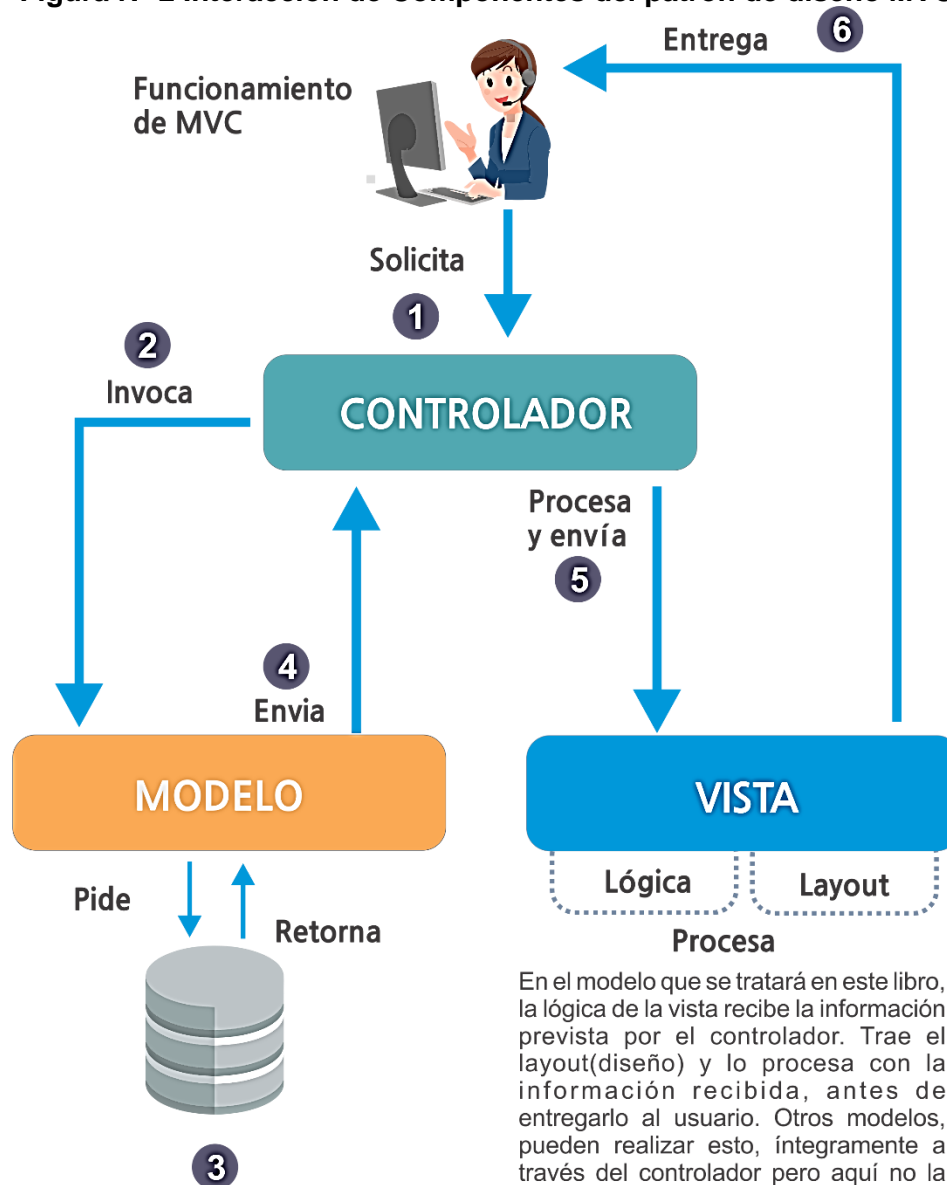
Vista: Se encarga de mostrar la información recibida al usuario por

medio de la interfaz gráfica de usuario. Así como también notifica al Controlador que se ha producido algún evento.

Controlador: Recibe los eventos de entrada y se encarga de realizar peticiones de actualización al modelo o a la vista según sea el caso.

En la siguiente figura se muestra la interacción de dichos componentes.

Figura N° 2 Interacción de Componentes del patrón de diseño MVC



En el modelo que se tratará en este libro, la lógica de la vista recibe la información prevista por el controlador. Trae el layout(diseño) y lo procesa con la información recibida, antes de entregarlo al usuario. Otros modelos, pueden realizar esto, íntegramente a través del controlador pero aquí no la trataremos por considerar que resta mantenibilidad al sistema

En la tabla siguiente se muestra una comparación que revela las principales ventajas y desventajas del uso del MVC frente a Modelo – Vista – Presentador (MVP).

Tabla N° 3 Ventajas y Desventajas de Patrones de Diseño MVC y MVP

Característica	MVC	MVP
Clara separación de conceptos	Si	Si
Reutilización de código	Si	Si
Flexibilidad	Si	Si
Facilidad de probar	Regular	Fácil
Facilidad de manejo de múltiples vistas	Fácil	Regular
Nivel de aprendizaje	Fácil	Regular
Nivel de conocimientos previos	Regular	-
Documentación existente	Bastante	Poca

2.2.20. Versionamiento Semántico

Muñoz Zolotoochin, J. E. (2011) En el mundo de la gestión de software existe el temor de caer en algún momento en el llamado “infierno de las dependencias”. Mientras más grande crece tu sistema y mientras más paquetes integras en tu software, más probable es que te encuentres, un día, en este pozo de la desesperación.

(Preston-Werner s. f.) Propone un set simple de reglas y requerimientos que dictan cómo asignar y cómo aumentar los números de

versión. Para que este sistema funcione, tienes que declarar primero un API pública. Esto puede consistir en documentación o ser explicitado en el código mismo. De cualquier forma, es importante que esta API sea clara y precisa. Una vez que identificaste tu API pública, comunicas cambios a ella con aumentos específicos al número de versión. Considera un formato de versión del tipo X.Y.Z (Major.Minor.Patch) Los arreglos de bugs que no cambian el API incrementan el patch, los cambios y adiciones que no rompen la compatibilidad de las dependencias anteriores incrementan el minor, y los cambios que rompen la compatibilidad incrementan el major.

2.2.21. Diseño Web Adaptable

APA Hassan Montero, Y., & Martín Fernández, F. J. (2013) El diseño web adaptable o adaptativo, conocido por las siglas RWD (del inglés, Responsive Web Design) es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visualizarla. Hoy día las páginas web se visualizan en multitud de tipos de dispositivos como tabletas, smartphones, libros electrónicos, portátiles, PC. Además, aún dentro de cada tipo, cada dispositivo tiene sus características concretas: Tamaño de pantalla, resolución, potencia de CPU, capacidad de memoria, ... Esta tecnología pretende que, con un solo diseño web, tengamos una visualización adecuada en cualquier dispositivo. (<https://goo.gl/64puv5>, 2015)

A nivel implementación Responsive Web Design tiene tres conceptos claves. El primero de ellos es el uso de los Media Queries que nos ofrece CSS3 permitiéndonos aplicar estilos condicionalmente teniendo

en cuenta parámetros de la pantalla. El segundo trata del diseño web fluido, se trata de layouts definidos en porcentajes que se ajustan a los anchos de la pantalla. y por último el tercer concepto se trata de los elementos fluidos dentro de estos layouts, como las fuentes o elementos multimedia.

Indica que al crear un sitio con Responsive Web Design solo necesitamos una única versión de HTML y CSS que funcionará adecuadamente en cualquier tipo de dispositivo y resolución. Con Responsive Web Design debemos de dejar de ofuscarnos en que nuestra web se vea idéntica en cada dispositivo.

2.2.22. Ventajas e Inconvenientes del Empleo de Responsive Web

Design

Alonso Vega (2013) Define las siguientes ventajas y desventajas acerca del RWD:

Ventajas

Reducción de costos en el desarrollo, ya que el objetivo es único y claro con una sola implementación válida tanto para dispositivos de escritorio como dispositivos móviles.

Otra ventaja que aporta es la gran eficiencia a la hora de realizar los procesos de mantenimiento y actualización del contenido, ya que con un solo cambio se podrá ver reflejado en todos los dispositivos.

Desventajas

Un diseño Responsive son los tiempos de carga, ya que el usuario

debe descargar todo el código fuente, aún sin ser necesario, e incluso si el diseño no está optimizado con imágenes adaptadas a distintas resoluciones, debe descargar imágenes de gran resolución sin ser necesario.

Aunque aporta ventajas para el desarrollador web, también este debe de emplear más tiempo en optimizar el diseño para las distintas resoluciones respecto de un desarrollo web no Responsive.

CAPITULO III

MATERIALES Y MÉTODOS

3.1. Ubicación Geográfica del Estudio

El desarrollo del presente trabajo se llevó a cabo en la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano - Puno.

3.2. Población y Muestra del Estudio

Población

Nuestra población para efectos de investigación está integrada por todos los usuarios de la Plataforma de Gestión Documentaria comprendida por los miembros de la Comisión Permanente de Fiscalización, autoridades y jefes de unidades orgánicas comprende entre 93 individuos.

3.3. Operacionalización De Variables

Tabla N° 4 Operacionalización de variables

Variable	Dimensiones	Indicadores	Escala de Calificación	Instrumentos
Plataforma de Gestión Documentaria	Creación de los documentos	<ul style="list-style-type: none"> ✓ Selección del tipo de documento. ✓ Fecha de registro ✓ Hora de registro ✓ Asunto 	<ul style="list-style-type: none"> ✓ Inaceptable. ✓ Mínimament e aceptable. ✓ Aceptable. ✓ Cumple los requisitos. ✓ Excede los requisitos. 	Plataforma de Gestión Documentaria Ficha de evaluación de la calidad del producto estándar ISO-9126
	Elección de destino	<ul style="list-style-type: none"> ✓ Selección de la unidad orgánica de envío. 		
	Atención del destino	<ul style="list-style-type: none"> ✓ Revisar ✓ Observar ✓ Finalizar ✓ Derivar 		
	Validación de la Plataforma de Gestión Documentaria	<ul style="list-style-type: none"> ✓ Funcionalidad ✓ Fiabilidad ✓ Usabilidad ✓ Eficiencia ✓ Mantenibilidad ✓ portabilidad 		

3.4. Recopilación de la Información

Para la recolección de información de requerimientos de la plataforma de gestión documentaria, se realizó entrevistas y encuestas a los usuarios que hacen uso de la plataforma de gestión documentaria.

Para la recopilación de datos de usabilidad de la plataforma se utilizó los instrumentos que fueron aplicados en los usuarios de la Plataforma de Gestión Documentaria para la Comisión Permanente de Fiscalización en la Universidad Nacional del Altiplano Puno, como: La ficha de evaluación del ISO-9126 que ve la calidad del software.

3.5. Modelo y Metodología de Desarrollo de Software

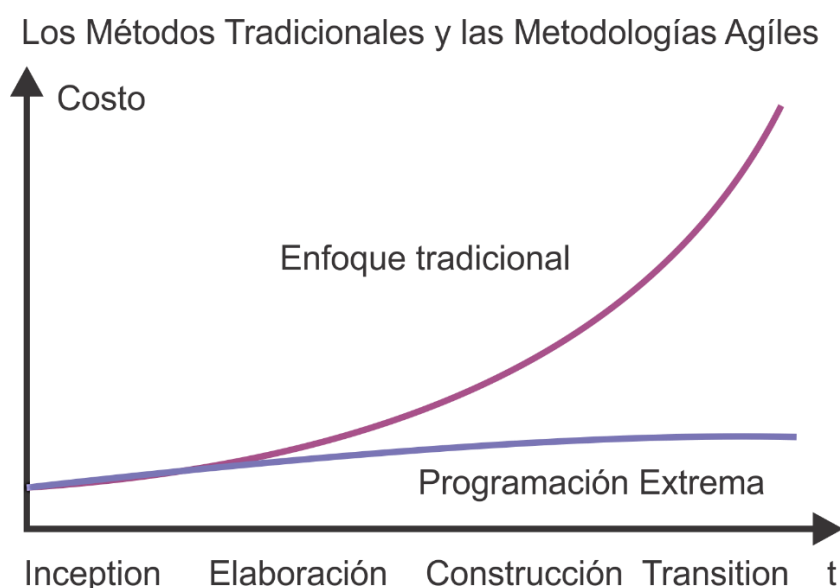
El propósito no es escoger una metodología porque sea mejor, pues el empleo de una u otra es de acuerdo al tipo de proyecto, a los recursos que serán utilizados y a la facilidad de interacción con el usuario final

Tabla N° 5 Diferencias entre Metodologías Ágiles y Metodologías Tradicionales.

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

La investigación se centra en el desarrollo de un sistema de información y los recursos que priman son los relacionados a los factores tiempo y costo. En cuanto a los costos lo deseable es que sea lo más reducido posible y, en relación al tiempo, se desea que la mayor inversión de éste sea para la construcción del sistema.

Figura N° 3 Comparación Tiempo y Costo de Metodologías de desarrollo.



El costo de Introducir cambios crece conforme avanza la vida del proyecto

3.5.1. Modelo de Desarrollo

Nos adaptamos a un híbrido modelo de desarrollo optando por dos metodologías XP y SCRUM por ágiles además de adaptarse a este tipo de software que se va a diseñar ya que se da por tareas específicas que permiten que el trabajo se desarrolle simultáneamente, colaborativamente y permitiendo repartir el tiempo y trabajo de forma equitativa y a la vez corrigiendo los errores cometidos en las primeras fases del desarrollo de la plataforma. Otro motivo es porque genera menores costos y porque es uno

de los más utilizados actualmente lo que nos da seguridad de estos modelos de desarrollo.

Ya que también el modelo de desarrollo SCRUM es organizado y a la vez es riguroso. Además este es utilizado como base en otros modelos, este se define por su enfoque metodológico que ordena rigurosamente en los SPRINTS, de forma tal que el inicio de una nueva tarea se pueden ir desarrollando otras tareas en simultaneo (Sprint Planning Meeting), sin descuidar una de la otra de esta forma cualquier error de diseño detectado se puede corregir inmediatamente antes de finalizar la tarea (Scrum Daily Meeting) de esta manera controlando todo el proceso con revisiones al finalizar cada tarea encomendada (Sprint Review), dándole la seguridad, optimización y calidad de desarrollo observando cada detalle de las etapas de desarrollo y dando la oportunidad reiniciar el ciclo para mejorar la calidad del producto (Sprint Retrospective) así como también las historias de usuario y el tablero de kanban.

3.6. Análisis de Planificación para el Desarrollo

3.6.1. Dueño de Producto:

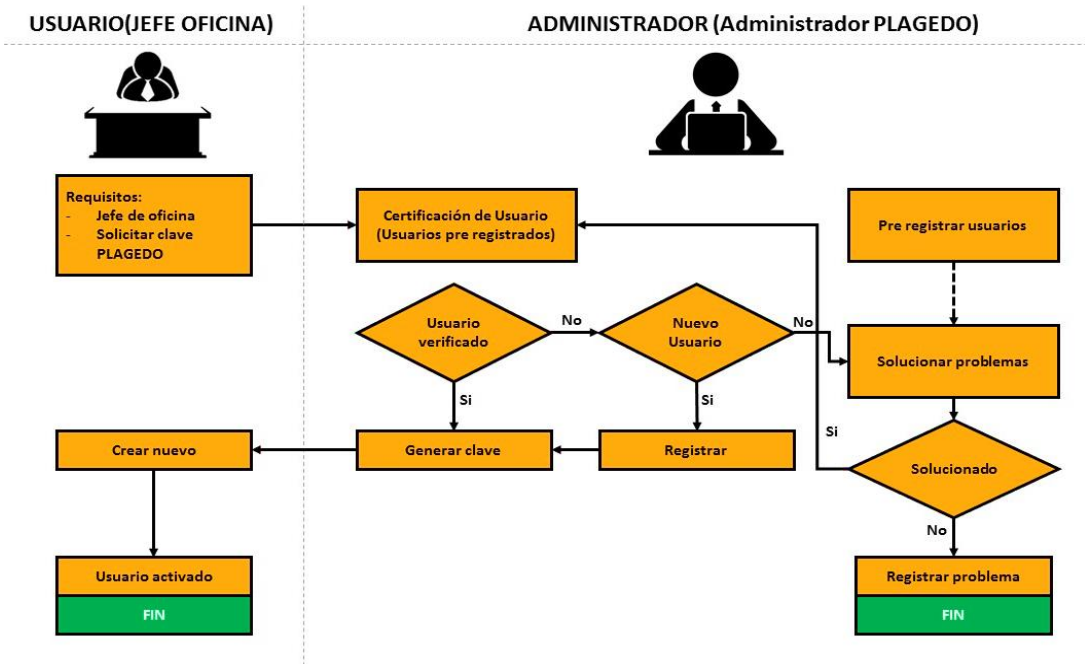
Tabla N° 6 Backlog

N°	USUARIO	TAREAS
1	Los jefes de las unidades deberán registrarse y validar sus datos en la Plataforma desde cualquier punto de conexión a internet (celular, Tablet, laptop, computadora).	✓ Diseño e implementación de tablas la base de datos. ✓ Diseño de una interfaz de validación. ✓ Elegir una herramienta de desarrollo de la aplicación.
2	El tramitante desde la unidad orgánica de origen deberá registrar sus trámite.	✓ Diseño de la base de datos e interfaz de registro de información del documento.
3	El tramitante desde la unidad orgánica deberá leer el documento y debe dar acciones sobre los documentos recepcionados.	✓ Diseño e implementación de los módulos de la bandeja de trámites recibidos.
4	Si el jefe de la unidad orgánica de destino acepta y da solución sobre el documento recepcionado habrá finalizado el documento.	✓ Diseño e implementación de las listas de observaciones en la base de datos
5	La plataforma debe de administrar todos los documentos tramitados automáticamente y categorizar por cada unidad orgánica Categoría - Asunto - Número de expediente - Unidad orgánica de origen - Unidad orgánica de destino	Diseño e implementación de los módulos de administración de los documentos finalizados o archivados por los motivos de los estados del documento.
6	Los documentos serán conservados para así dar fe sobre los hechos.	Se deben diseñar e implementar los módulos para la administración de los documentos.
7	Para controlar los estados en el panel de administración se deberán mostrar reportes.	Construir la base de datos además de los interfaces.

3.7. Diagrama del Proceso Detallado

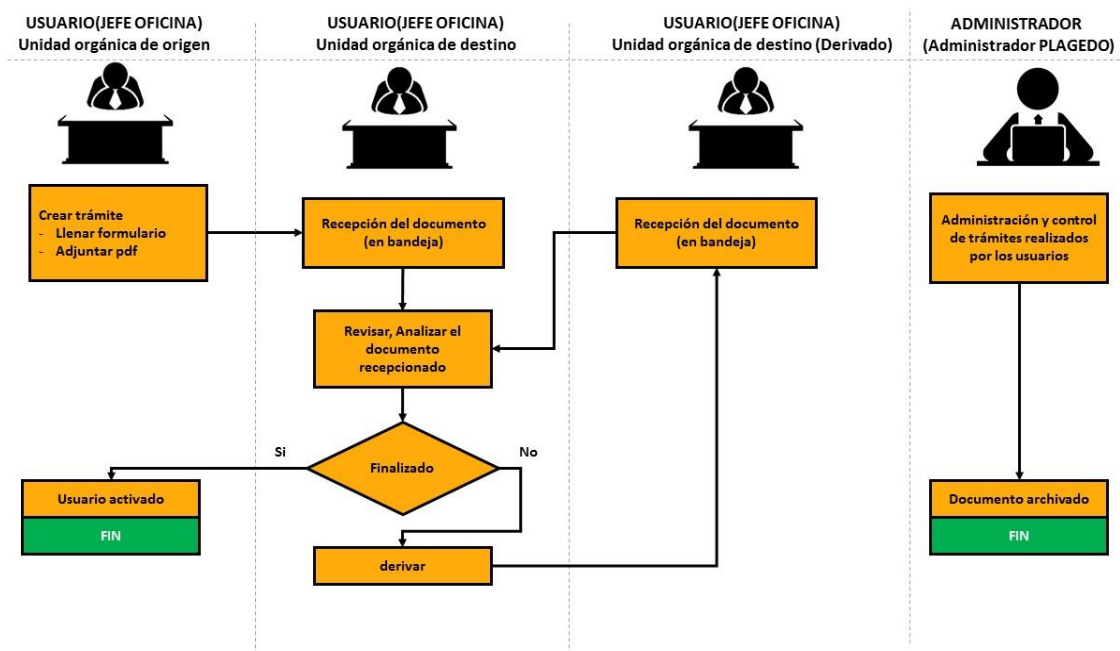
3.7.1. Registro y Activación de Usuarios

Figura N° 4 Diagrama del proceso de registro y activación de usuarios



3.7.2. Proceso de Trámites

Figura N° 5 Diagrama del proceso de los trámites



3.7.3. Requerimientos de la Plataforma

Requerimientos Funcionales

Se definieron para la plataforma, los siguientes puntos más relevantes que el software debe poder realizar:

- R1.** Deben existir una validación de datos para el acceso a la plataforma, este le dará mayor seguridad a la información concebida por los usuarios – Los usuarios solicitaran una clave hacer el uso de la Plataforma.
- R2.** Módulo de validación de los registros deberá realizarse con la previa solicitud de cuenta PLAGEDO, para verificar los requisitos.
- R3.** La Plataforma debe estar en la capacidad de activar nuevas cuentas de los miembros de la Comisión Permanente de Fiscalización.
- R4.** La Plataforma debe estar en la capacidad de poder editar datos como de los usuarios, si en caso hubiese errores de registro.
- R5.** Por necesidad debe existir la posibilidad de consultar datos como el seguimiento de los expedientes realizados.
- R6.** Por necesidad hacer el seguimiento desde su trámite hasta la ser atendidos por las unidades de destino.
- R7.** Módulo para la creación de los documentos.
- R8.** Módulo para la administración de los documentos enviados.
- R9.** Módulo para la administración de los documentos recibidos.
- R10.** Módulo de lectura y acciones sobre el documento recibido.

- R11. Módulo de comentario o derivaciones sobre el documento recibido.
- R12. Módulo para la generación de los documentos en formato digital (pdf).
- R13. Módulo para generar Reportes de los estados de los documentos.

Requerimientos no Funcionales:

- ✓ Aplicación multiplataforma.
- ✓ Interfaz agradable para un fácil entendimiento del software.
- ✓ Disponibilidad de la plataforma de encontrarse disponible todos los días.
- ✓ Estabilidad la plataforma soporta varios usuarios a la vez.
- ✓ Portabilidad estará diseñado en un lenguaje multiplataforma. ü
- Rendimiento la plataforma brindará un servicio óptimo ya que es diseñado para que funcione en un ambiente web.
- ✓ Mantenimiento y Escalabilidad diseñado pensando en el crecimiento de la plataforma.

Requerimientos técnicos:**Hardware**

Utilice para el desarrollo del presente trabajo de investigación el

Hardware con las siguientes características:

- ✓ 03 Computadora HP CORE i7 / 2.5Ghz.
- ✓ Servicio de Internet 4 Mbps - Claro.
- ✓ 4 Gb de memoria RAM DDR3.
- ✓ Monitor LG IPS 24MP88

- ✓ 01 Impresora HP COLOR LASER JET CP3525dn.
- ✓ Disco duro externo 1Tb
- ✓ 02 Smartphone Samsung Galaxy S7 edge.
- ✓ Como mínimo para el funcionamiento de la plataforma se utilizará una computadora Core i7.

Software

- ✓ Lenguaje de programación PHP.
- ✓ Gestor de Bases de Datos MySQL.
- ✓ Herramienta visual de diseño de bases de datos (MySQL Workbench).
- ✓ Servidor Web, Apache.
- ✓ Sublime Text.
- ✓ Microsoft Office.
- ✓ Google Chrome.
- ✓ Mozilla Firefox
- ✓ Adobe Photoshop
- ✓ Rational Rose.
- ✓ Dia Diagram Editor
- ✓ Sistema Operativo Windows 10.
- ✓ Frameworks (Bootstrap, JQuery, Codeigniter).

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1. Resultados

4.1.1.Ámbito del Problema

La Comisión Permanente de Fiscalización tenía como premisa que el trámite de documentos realizados por los responsables de las oficinas se llevaría a cabo con el registro manual de los documentos a nivel de todas oficinas de la gestión del rectorado, lo que ocasiona en un momento dado aglomeraciones, y pérdida de tiempo por parte de los usuarios, además de generar confusiones y archivamiento rápido sin dar solución sobre ella ya que todos los trámites no se manejan bien.

A. Análisis de Viabilidad

Viabilidad Operativa: La Comisión Permanente se encuentra en la capacidad de administrar y controlar todos los trámites, así como las oficinas de la gestión del rectorado está en las condiciones técnicas para poder coadyuvar a todos estos procedimientos aquí planteados.

Viabilidad Técnica: La comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano Puno, cuenta con algunos equipos para la implementación del software, será necesario instalar un servidor web para poder correr la aplicación.

B. Análisis de Roles

Hay que tener en cuenta que los desarrolladores del proyecto eran dos personas por lo que los roles definidos en XP fueron ocupados por mi persona, en algunos casos por el asesor del proyecto y el director de proyecto.

- ✓ Programador: El código fuente fue hecho por mi persona.
- ✓ Arquitecto de Software: Mi asesor de tesis, quien apoyo en el modelado de la Plataforma.
- ✓ Cliente: El desarrollador del proyecto y los encargados de administrar la plataforma fueron los que hicieron las pruebas funcionales para validar su implementación.
- ✓ Encargado de pruebas (Tester): El personal del Vicerrectorado de Investigación quien ayudó a escribir las pruebas funcionales. Ejecutó las pruebas regularmente, e informó los resultados y apreciaciones al desarrollador.

- ✓ Encargado de seguimientos (Tracker): El personal del Vicerrectorado de investigación quien proporcionó realimentación al equipo y realizó el seguimiento del progreso de cada iteración.
- ✓ Entrenador (Coach): Mi persona fue el responsable del proceso global también de realizar las guías para las prácticas XP y SCRUM además responsable de que se siguiera el proceso correctamente.

Algunos roles definidos en XP y SCRUM no fueron tomados en cuenta para este proyecto, bien porque no fuese necesario o porque el tamaño tan reducido del equipo de trabajo no lo permita.

4.2. Diseño de la Plataforma

Para el diseño de la aplicación el equipo de trabajo siguió las recomendaciones de la metodología XP y SCRUM, tratando de evitar las soluciones complejas, y se trabajó en varias iteraciones, previendo las modificaciones futuras para la optimización de la plataforma.

Otro aspecto importante en el diseño, fue la constante reestructuración del código. El principal objetivo de la reestructuración fue evitar la duplicación del código poniendo en práctica la normalización de datos simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Esto se hizo constantemente en la programación de la aplicación.

4.3. Modelamiento de la Plataforma Mediante UML7

Para entender el modelamiento en el diseño de la plataforma se ha

utilizado el lenguaje unificado de Modelado UML, la misma que nos ha permitido obtener los siguientes diagramas:

- ✓ Diagrama de Casos de uso.
- ✓ Diagrama de secuencia.

4.4. Versiones e Iteraciones

Versiones 0.1

Iteración 1: El diseño de la base de datos es la parte primordial para empezar con el desarrollo de cualquier aplicación, ya que un buen diseño de esta, es la base fundamental para el óptimo funcionamiento y éxito de cualquier software.

Para definir el diseño se contó siempre con la ayuda del personal del Vicerrectorado de Investigación, quienes informaban que datos serian relevantes para el proyecto en varias iteraciones se logró diseñar las siguientes tablas.

- ✓ Tabla **documento**: Esta tabla almacena todos los contenidos de los documentos creados por los usuarios.
- ✓ Tabla **area**: Esta tabla almacena todas las unidades orgánicas de origen y destino asociada con los respectivos usuarios.
- ✓ Tabla **estado**: Esta tabla almacena el estado en el cual se encuentra el trámite realizado.
- ✓ Tabla **mensajes**: Esta tabla almacena todas las observaciones sobre el documento asociada con la tabla **estado**.
- ✓ Tablas **facultad, escuela**: Almacena las escuelas asociadas a sus respectivas facultades.

De esta manera se terminó la versión 0.1, cumpliendo con las tareas registradas en el backlog.

Esta versión se les enseñó a los miembros de la Comisión Permanente de Fiscalización, los cuales dieron su visto bueno al sentirse conformes con los avances presentados.

Versión 0.2

Iteración 1: Como la aplicación debía estar en la nube para poder ser consultada desde cualquier ubicación geográfica con conexión a internet, se seleccionó PHP 5.6.30 una versión estable, ya que es uno de los lenguajes de desarrollo para aplicaciones web más utilizadas y de gran difusión, por estas razones y por otras que nombramos en el apartado de herramientas, se consideró una herramienta ideal para el desarrollo de esta aplicación.

Iteración 2: Como se buscaba tener un diseño amigable, intuitivo y de fácil uso para la aplicación se creó una página principal (máster page), la cual es una plantilla de la que heredaran todas las demás páginas de la aplicación web. De esta forma se tiene una estructura fija para todo el sitio web.

En la página principal plantilla máster se definió la estructura común para toda la aplicación web dándole los estilos y distribución correspondiente con BOOTSTRAP, en cuanto su optimización y agilidad con AJAX, herramientas de JQUERY y su validación respectiva de los formularios con JavaScript y el menú de navegación de la aplicación.

De esta manera se terminó la versión 0.2, cumpliendo con el 1er

backlog del usuario.

Versión 0.3

Iteración 1: Se implementó los módulos de registro de datos haciendo uso del MVC (Modelo Vista Controlador) para poder tener un código ordenado y óptimo.

En este módulo se hizo la aplicación vía web para el registro y validación de información de los usuarios, así como también se elaboró un manual de usuario para el registro de datos.

De esta manera se concluyó con la versión 0.3.

Versión 0.4.

Iteración 1: En esta versión se implementó el módulo para la edición de datos y reportes de todo el proceso que se realizó.

Este módulo fue muy similar al módulo de ingresar datos, porque por medio de los controles que proporciona las herramientas web para poder editar los datos que existían por el error que se da por parte de los usuarios.

Iteración 2: En esta iteración se implementó el módulo de revisión del director y revisión del jurado, por tener una similitud en el procedimiento de revisión corrección y dictamen, el cual se etiquetó con estados del trámite para la identificación de los mismos.

Iteración 3: Para esta iteración se implementó los estados de los trámites que realizan los usuarios de la plataforma.

Iteración 4: Para esta iteración se implementó los reportes de los

trámites realizados por los mismos usuarios tramitantes.

De esta manera se terminó la versión 0.4, en la que se cumple con los requerimientos del backlog del usuario, estando la plataforma lista para las retrospectivas de los sprints, para realizar las mejoras correspondientes a la plataforma.

Esta versión se le enseñó a los clientes, los cuales dieron su visto bueno al sentirse conformes con los avances presentados además de la puesta en funcionamiento de la plataforma.

Versión 0.5 hasta la 0.9

Desde la versión 0.5 hasta la versión 0.9 se trabajó en el diseño estético de la aplicación.

En cada una de las iteraciones que involucraba la entrada de datos por parte de los usuarios a la aplicación, se diseñaron e implementaron los diferentes tipos de validaciones respectivas para cada uno de los campos que lo requerían.

En este punto fue primordial validar los campos requeridos, los cuales no debían quedar vacíos ya que representarían un error al momento de ser guardados en la base de datos y emitir su documento.

Además de estar listos la plataforma para que pueda ser instalado de una manera fácil y poder ser controlado desde cualquier sitio remotamente por el administrador de la plataforma ya que todo está en red.

4.5. Codificación

Toda la transcripción de algoritmos y métodos utilizados para el software, se han realizado en PHP, esto hace al software independiente de la plataforma que se esté utilizando, sin exigir las características de hardware, únicamente un navegador web (browser) porque lo más importante para el software es el cumplimiento de sus funciones para las cuales fue desarrollado, así como para la administración del mismo.

Cliente siempre disponible

La Metodología XP recomienda como factor de éxito que el cliente esté involucrado en toda la etapa de desarrollo, esto se cumplió satisfactoriamente.

Estándares de codificación

La metodología XP aconseja seguir estándares de codificación para que cualquier integrante del equipo de desarrollo, pueda entender y asimilar fácilmente el código fuente del programa por otro integrante.

Seguir esta recomendación fue fácil para el desarrollador ya que venía trabajando de tiempo atrás en otros desarrollos y esta buena práctica siempre se tuvo presente.

4.6. Pruebas

Al momento de las ejecuciones reiteradas de la plataforma no se tuvo inconvenientes sin embargo no está demás explicar que como todo software este se encuentra predispuesto a cualquier tipo de cambio que el usuario requiera con la finalidad de poder optimizar del mismo, esto incluye

también las actualizaciones que por tiempo de vida de uso se deben de realizar periódicamente.

4.7. Instalación

Durante la instalación no se presentó ningún tipo de problemas, porque esta plataforma fue desarrollada en un entorno web.

Lo que si debemos de tener en cuenta de que el tráfico de información el cual aumentará progresivamente por la cantidad de usuarios en la Universidad Nacional del Altiplano Puno, también el manual de usuario se encuentra en el ANEXO III.

4.8. Resultados de la Prueba Según El ISO – 9126

PROMEDIO GENERAL DE LAS FICHAS DE EVALUACIÓN ISO-9126

Tabla N° 7 Cuadro de decisiones

Clasificación	Intervalo	Decisión
A) Inaceptable	[27 - 54 >	
B) Mínimamente aceptable	[54 – 81 >	
C) Aceptable	[81 – 95 >	
D) Cumple los requisitos	[95 – 122 >	110
E) Excede los requisitos	[122 – 135]	

Fuente: Cuadro de decisiones ISO - 9126

Según los resultados el promedio de 114, 113, 117, 107, 106, 103 y

110 nos resultó 110, indicando que cumple con los requisitos según el ISO - 9126. Todo esto se muestra en el ANEXO II.

4.9. Gráficos de Satisfacción de Usuario

Gráfico N° 1: ¿Cómo califica los servicios prestados a través de la PLATAFORMA PLAGEDO?

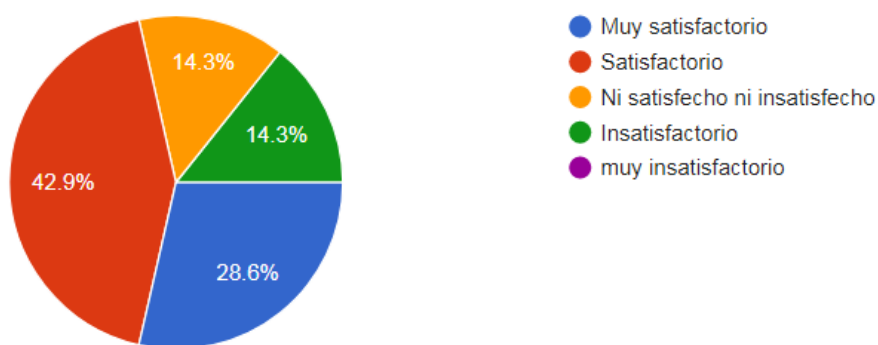


Gráfico N° 2: ¿Cómo considera el acceso a la PLATAFORMA PLAGEDO?

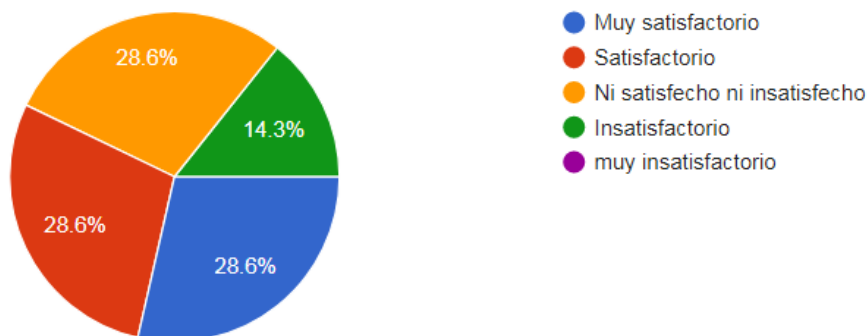


Gráfico N° 3: ¿Para remitir el documento y/o para su recepción, se hizo de la forma correcta la primera vez de acuerdo a la capacitación antecedida?

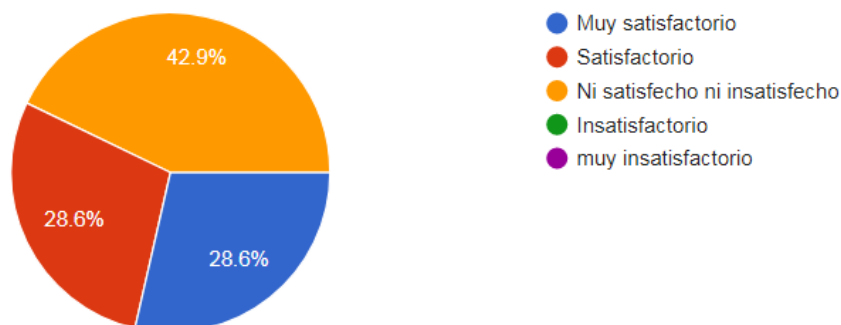


Gráfico N°4: ¿La PLATAFORMA PLAGEDO muestra la información de los documentos con claridad?

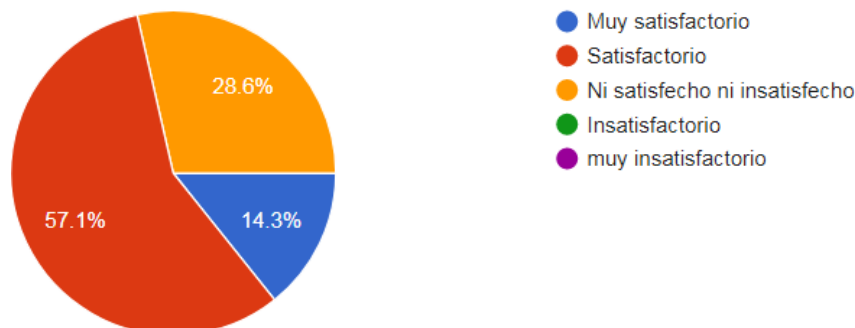


Gráfico N°5: ¿Cómo usted considera el tiempo de respuesta de los tramites?

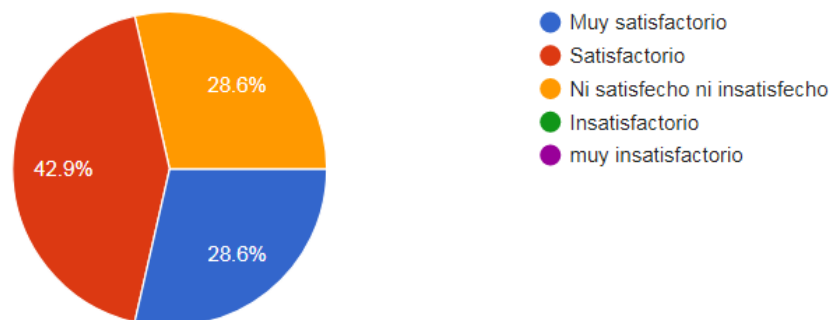
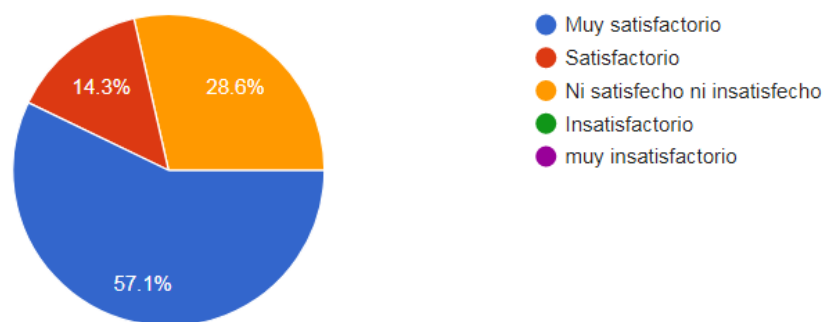


Gráfico N°6: ¿Cómo considera usted la seguridad de los tramites?



4.10. Contraste de Hipótesis

Los datos son provenientes de una población conformada por 93 usuarios de la Plataforma de Gestión Documentaria Virtual para la Comisión Permanente de Fiscalización. El porcentaje poblacional para la implementación de la plataforma PLAGEDO es del 69%.

Hipótesis Nula (H0):

La implementación de la plataforma de gestión documentaria virtual no mejorará la gestión documentaria de la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano Puno, 2017.

Alternativa (Ha):

La implementación de la plataforma de gestión documentaria virtual mejorará la gestión documentaria de la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano Puno, 2017.

i) Nivel de Significancia

Se considera un nivel de 95% de confiabilidad y una significancia del ($\alpha = 0.05$) para establecer relación entre las variables evaluadas. con IBM SPSS para 93 datos de la población con 3 valores perdidos.

ii) Prueba estadística

CORRELACIÓN DE PEARSON DEL NIVEL DE SATISFACCIÓN Y EL TIEMPO DE PROCESAMIENTO DE LOS DATOS EN EL AÑO 2017.

		NIVEL DE SATISFACCION DEL SISTEMA WEB	TIEMPO EN EL PROCESO DE LOS TRÁMITES
NIVEL DE SATISFACCIO N DEL SISTEMA WEB	Correlación de Pearson Sig. (bilateral) N	1 90	,669** ,000 90
TIEMPO DE PROCESAMIE NTO DE LOS DATOS	Correlación de Pearson Sig. (bilateral) N	,669** ,000 90	1 90

** La correlación es significativa en el nivel 0,01 (bilateral)
FUENTE: Elaboración en SPSS (Versión 22).

Como el P (**0.000**) muy inferior o menor al nivel de significancia α (0.050) entonces la prueba es significativa, por lo tanto, determinamos que existe una relación significativa entre el nivel de satisfacción de usuarios y el tiempo de procesamiento de los datos en la implementación de la plataforma de gestión documentaria virtual en la Universidad Nacional del Altiplano en el año 2017, con una confiabilidad del 95%. Existe una correlación de Pearson de 0.669 lo cual nos indica que existe una correlación directa y muy buena, los usuarios están muy satisfechos con la plataforma. También se denota que el tiempo de procesamiento de los datos es eficaz y buena.

iii) Regla de decisión

Si $Z_c > Z_t = 1.00 > 0.05$, entonces se rechaza la H_0 y se acepta la H_a .

iv) Conclusión

De acuerdo a los resultados obtenidos, se puede concluir que la implementación de la Plataforma de Gestión Documentaria virtual para la Comisión Permanente de Fiscalización de la UNA – PUNO 2017, mejorará la gestión documentaria de la comisión permanente de Fiscalización en la Universidad Nacional del Altiplano 2017

4.11. Decisión

De acuerdo a los resultados de la calidad del software se concluyó que la Plataforma de Gestión Documentaria de la Comisión Permanente de Fiscalización, mejora la gestión documentaria virtual y cumple los requisitos de software con un promedio de 110 puntos del total de 135 puntos que se considera en el cuadro de decisiones del ISO - 9126.

CAPITULO V

CONCLUSIONES

Del análisis de los resultados se concluye que:

- ✓ El desarrollo y la implementación de la Plataforma de Gestión Documentaria mejora la gestión documentaria de la Comisión Permanente de Fiscalización de la Universidad Nacional del Altiplano Puno, 2017.
- ✓ La implementación de la Plataforma de Gestión Documentaria cumple los requisitos establecidos por la ficha de evaluación la cual se aplicó la evaluación de calidad del software con el ISO-9126 se obtuvo 110 puntos cumple los requisitos establecidos por esta ficha de evaluación.
- ✓ Se determinó las principales ventajas de la implementación de la Plataforma de Gestión Documentaria Virtual: Ágil, Ordenado, Tiempo, de acuerdo a los requisitos establecidos por la ficha de evaluación ISO-9126.
- ✓ Se realizó la implementación de los módulos de activación de usuarios, trámites, bandejas de entrada y salida, reporte del tiempo de trámites, estados de los documentos.

CAPÍTULO VI

RECOMENDACIONES

- ✓ Se recomienda la implementación de programas de capacitación de asistencia para cada usuario del sistema y así mejorar la adaptación de los actores del servicio de los diferentes tipos de documento se recomienda implementar plataformas con la oficina del vicerrectorado académico y vicerrectorado de investigación, los institutos, facultades y otros en su totalidad.
- ✓ Se sugiere la automatización de los procedimientos mecánicos de los trámites puesto que se transformarán en una forma usual de comunicación y reducción de procesos.
- ✓ Para la vigente investigación se tomó como herramienta de desarrollo el software libre, sin embargo, existen otros lenguajes de programación con licencia que se pueden utilizar para futuros proyectos y así incorporar nuevas mejoras que fortalezcan su potencia y rapidez.
- ✓ Se recomienda que la presente investigación sea utilizada como documento de referencia para investigadores en el área de la Informática, Ciencias de la Computación y Sistemas en otras universidades de la región.

CAPITULO VII

REFERENCIAS

Alonso Vega, Adrián. (2013). «Responsive Web Design interfaces web adaptables al dispositivo empleando HTML5 y CSS3».

Aranda Font, Odalys. (2013). «Implementación de un Sistema de Gestion Documental en la Universidad Central Marta Abreu de las Villas, Cuba: Facultad de Ciencias de la Información y de la Educación. Universidad Central Marta Abreu de las Villas». <https://goo.gl/Nj8Tv2>.

Burke, Edmund K et al. (2013). «Hyper-heuristics: A survey of the state of the art». *Journal of the Operational Research Society* 64(12): 1695-1724.

Canós, José H, y MaCarmen Penadés Patricio Letelier. (2012). «Metodologías ágiles en el desarrollo de software».

Cobo, Ángel. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos.

Garcia-Beltrán, Ángel, Raquel Martínez, José Alberto Jaén, y

Santiago Tapia. (2016). «La autoevaluación como actividad docente en entornos virtuales de aprendizaje/enseñanza». Revista de Educación a Distancia (50).

Graham, Ian S. (1995). The HTML sourcebook. John Wiley & Sons, Inc.

Iglesias, Fran. (2012). «El futuro de la maquetación web con CSS3». Macworld España: la revista para los usuarios del Macintosh. (220): 56-62.

Jihuallanca Ccoa, Nelida Sonia. (2016). «sistema de consulta de la unidad de pensiones y liquidaciones de la Universidad Nacional del Altiplano Puno». http://repositorio.unap.edu.pe/bitstream/handle/UNAP/2091/Jihuallanca_Ccoa_Nelida_Sonia.pdf?sequence=1&isAllowed=y.

Mendoza, Lolimar. (2011) «Implementación de un sistema automatizado que optimice la gestión de los procesos administrativos del área servicios médicos.» Maturin. Venezuela.

Pezo Villacorta, Edwin Augusto. (2014). «Sistema informático de trámite documentario para la Universidad Privada de la Selva Peruana - Iquitos». Universidad Privada de la Selva Peruana - Iquitos. <https://goo.gl/RFGT2r>.

Pressman, Roger S, y Jose Maria Troya. (1988). «Ingeniería del software».

Preston-Werner, Tom. «Versionamiento Semántico». <https://goo.gl/Y4UzH5>.

Suárez, Oscar Jardey. (2016). «Aproximación al origen de la noción de objeto de aprendizaje: revisión histórico-bibliográfica». INGE CUC 12(2): 26-40.

Torres Cruz, Fred. (2016). «Plataforma web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado UNA Puno 2016». http://repositorio.unap.edu.pe/bitstream/handle/UNAP/4848/Torres_Cruz_Fred.pdf?sequence=1&isAllowed=y.

Spona, H. (2010). Programación de bases de datos con MYSQL y PHP. Marcombo.

Duarte-García, E. (2007). Gestores personales de bases de datos de referencias bibliográficas: características y estudio comparativo.

Silva, M. S. (2008). JQuery: a biblioteca do programador JavaScript. São Paulo: Novatec Editora.

Gauchat, J. D. (2012). El gran libro de HTML5, CSS3 y Javascript. Marcombo.

Powell, T. (2008). Ajax: the complete reference. McGraw-Hill, Inc..

Pons, C., Giandini, R. S., & Pérez, G. (2010). Desarrollo de software dirigido por modelos. ISO 690

Aubry, C. (2012). HTML5 y CSS3-Revolucione el diseño de sus sitios web. Ediciones ENI.

Bahit, E. (2011). Poo y mvc en php. El paradigma de la Programación.

Hassan Montero, Y., & Martín Fernández, F. J. (2013). Propuesta de adaptación de la metodología de diseño centrado en el usuario para el desarrollo de sitios web accesibles. ISO 690

Muñoz Zolotoochin, J. E. (2011). Implementación de una Red Semántica de Archivos.

Canós, J. H., & Letelier, M. C. P. P. (2012). Metodologías ágiles en el desarrollo de software.

ANEXOS

ANEXO – I

FICHA DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO ESTÁNDAR

ISO-9126

Tabla N° 8 Ficha de Evaluación ISO - 9126

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
1. FUNCIONALIDAD					
Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					
Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.					
Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas especificados					
Seguridad: referido a la capacidad del producto software para proteger la información y los datos.					
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad.					
2. FIABILIDAD					
Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software.					
Tolerancia a fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.					

Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.					
Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.					
3. USABILIDAD					
Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.					
Facilidad de aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación.					
Atracción: la capacidad del producto software para atraer al usuario.					
Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.					
Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle.					
4. EFICIENCIA					
Comportamiento temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.					
Utilización de recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.					

Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					
5. MANTENIBILIDAD					
Analizabilidad: Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.					
Cambiabilidad: Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.					
Estabilidad: Capacidad del producto software de evitar los efectos inesperados de las modificaciones.					
Facilidad de prueba: Capacidad del producto software de permitir validar las partes modificadas.					
Conformidad: Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.					

Ficha de Evaluación ISO - 9126

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
6. PORTABILIDAD					
Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.					
Facilidad de instalación: la capacidad del producto software para ser instalado en un ambiente determinado.					
Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.					

Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.					
Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.					
SUB TOTALES					
TOTAL					

Fuente: Ficha de evaluación estándar ISO-9126

Tabla N° 9 Escala valorativa (Escala de Likert)

Indicador Cualitativo	Valor
Deficiente	1
Malo	2
Regular	3
Bueno	4
Muy bueno	5

Fuente: Escala de Likert

Tabla N° 10 Cuadro de decisiones ISO 9126

Clasificación	Intervalo	Decisión
A) Inaceptable	[27 - 54 >	
B) Mínimamente aceptable	[54 – 81 >	
C) Aceptable	[81 – 95 >	
D) Cumple los requisitos	[95 – 122 >	
E) Excede los requisitos	[122 – 135]	

Fuente: Cuadro de decisiones ISO 9126.

ANEXO – II

ENCUESTA DE SATISFACCIÓN DE USUARIO

Esta encuesta está dirigida a todos los USUARIOS registrados en la Plataforma de Gestión Documentaria (PLAGEDO).

Nos interesa tu opinión y queremos brindarte un servicio ágil, eficiente, seguro y amigable.

1. ¿Cómo califica los servicios prestados a través de la PLATAFORMA PLAGEDO?

- Muy satisfactorio
- Satisfactorio
- Ni satisfecho ni insatisfecho
- Insatisfactorio
- Muy insatisfactorio

2. ¿Cómo considera el acceso a la PLATAFORMA PLAGEDO?

- Muy satisfactorio
- Satisfactorio
- Ni satisfecho ni insatisfecho
- Insatisfactorio
- Muy insatisfactorio

3. ¿Para remitir el documento y/o para su recepción, se hizo de la forma correcta la primera vez de acuerdo a la capacitación antecedida?

- Muy satisfactorio
- Satisfactorio
- Ni satisfecho ni insatisfecho
- Insatisfactorio
- Muy insatisfactorio

4. ¿La PLATAFORMA PLAGEDO muestra la información de los documentos con claridad?

- Muy satisfactorio
- Satisfactorio
- Ni satisfecho ni insatisfecho
- Insatisfactorio
- Muy insatisfactorio

5. ¿Cómo usted considera el tiempo de respuesta de los tramites?

- Muy satisfactorio
- Satisfactorio
- Ni satisfecho ni insatisfecho
- Insatisfactorio
- Muy insatisfactorio

6. ¿Cómo considera usted la seguridad de los tramites?

- Muy satisfactorio
- Satisfactorio
- Ni satisfecho ni insatisfecho
- Insatisfactorio
- Muy insatisfactorio

Tabla N° 11 FICHAS DE EVALUACIÓN REALIZADAS A LOS USUARIOS ISO 9126.

N°	¿Cómo califica los servicios prestados a través de la PLATAFORMA PLAGEDO?	¿Cómo considera el acceso a la PLATAFORMA PLAGEDO?	¿Para remitir el documento y/o para su recepción, se hizo de la forma correcta la primera vez de acuerdo a la capacitación antecedida?	¿La PLATAFORMA PLAGEDO muestra la información de los documentos con claridad?	¿Cómo usted considera el tiempo de respuesta de los tramites?	¿Cómo considera usted la seguridad de los tramites?
1	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Muy satisfactorio	Satisfactorio
2	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho
3	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	insatisfecho	Ni satisfecho ni insatisfecho
4	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio	Muy satisfactorio	Muy satisfactorio
5	Satisfactorio	Insatisfecho	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio	Muy satisfactorio
6	Insatisfecho	Satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio
7	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio
8	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio
9	Satisfactorio	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio
10	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio
11	Muy satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio
12	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho
13	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio
14	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio
15	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio
16	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio
17	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio
18	Insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio
19	Muy satisfactorio	Satisfactorio	Satisfactorio	Insatisfecho	Insatisfecho	Muy insatisfecho
20	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio

21	Muy satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Muy insatisfecho
22	Muy satisfactorio	Satisfactorio	Satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho
23	Muy satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho
24	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Insatisfecho
25	Muy satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio
26	Satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio
27	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio
28	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio
29	Satisfactorio	Insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Insatisfecho
30	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho
31	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho
32	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio
33	Ni satisfecho ni insatisfecho	Insatisfecho	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho
34	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho
35	Satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	Muy satisfactorio
36	Muy satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Satisfactorio	
37	Muy satisfactorio	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Insatisfecho	Satisfactorio
38	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio
39	Muy satisfactorio	Insatisfecho	Muy insatisfecho	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Insatisfecho

40	Muy satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio
41	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho
42	Muy satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Muy satisfactorio	Ni satisfecho ni insatisfecho
43	Muy satisfactorio	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho
44	Insatisfactorio	Insatisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Muy satisfactorio
45	Satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio
46	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho
47	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio
48	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio
49	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio
50	Satisfactorio	Muy satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio
51	Ni satisfecho ni insatisfecho	Muy satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho
52	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho
53	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Insatisfactorio
54	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio
55	Muy satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Muy satisfactorio
56	Muy satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio
57	Muy satisfactorio	Satisfactorio	Insatisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio

58	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio
59	Satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio
60	Muy satisfactorio	Satisfactorio	Insatisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio
61	Satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio
62	Muy satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio
63	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio
64	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio
65	Satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio
66	Ni satisfecho ni insatisfecho	Muy satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio
67	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho
68	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Insatisfactorio
69	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio
70	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio	Muy satisfactorio
71	Muy satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio
72	Muy satisfactorio	Satisfactorio	Insatisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio
73	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio
74	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio
75	Muy satisfactorio	Satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio
76	Satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio

77	Muy satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio
78	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio
79	Ni satisfecho ni insatisfecho	Satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio
80	Satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio
81	Satisfactorio	Muy satisfactorio	Satisfactorio	Insatisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho
82	Muy satisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Ni satisfecho ni insatisfecho	Ni satisfecho ni insatisfecho
83	Muy satisfactorio	Satisfactorio	Satisfactorio	Insatisfactorio	Ni satisfecho ni insatisfecho	Insatisfactorio
84	Satisfactorio	Satisfactorio	Insatisfactorio	Satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio
85	Muy satisfactorio	Muy satisfactorio	Ni satisfecho ni insatisfecho	Satisfactorio	Satisfactorio	Muy satisfactorio
86	Muy satisfactorio	Ni satisfecho ni insatisfecho	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio	Muy satisfactorio
87	Muy satisfactorio	Muy satisfactorio	Insatisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio
88	Satisfactorio	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio
89	Satisfactorio	Muy satisfactorio	Satisfactorio	Muy satisfactorio	Satisfactorio	Satisfactorio
90	Muy satisfactorio	Satisfactorio	Insatisfactorio	Satisfactorio	Muy satisfactorio	Muy satisfactorio

ANEXO - III

Tabla N° 12 FICHAS DE EVALUACIÓN REALIZADAS A LOS USUARIOS ISO 9126.

INDICADORES	Puntuación (1)	Puntuación (2)	Puntuación (3)	Puntuación (4)	Puntuación (5)	Puntuación (6)	Puntuación (7)
FUNCIONALIDAD							
Adecuación	5	5	5	3	3	3	4
Exactitud	5	4	3	4	5	4	5
Interoperabilidad	4	4	5	3	4	5	3
Seguridad	5	5	5	5	4	3	4
Conformidad	4	3	5	4	3	3	3
FIABILIDAD							
Madurez	5	3	5	5	3	4	3
Tolerancia a fallos	5	5	5	4	3	5	4
Recuperabilidad	4	5	4	3	5	4	5
Conformidad	5	4	5	5	3	3	4
USABILIDAD							
Comprensibilidad	3	5	5	3	5	5	3
Facilidad de aprendizaje	3	3	5	4	4	3	4
Atracción	4	4	4	3	3	5	3
Conformidad	4	4	4	4	4	3	4
Operabilidad	5	4	5	4	4	5	5
EFICIENCIA							
Comportamiento temporal	3	3	4	4	3	4	5
Utilización de recursos	4	3	4	3	3	3	4
Conformidad	5	5	3	5	4	4	5
MANTENIBILIDAD							
Analizabilidad	3	5	5	3	3	4	4
Cambiabilidad	4	3	3	5	4	3	5
Estabilidad	4	5	3	4	5	3	3
Facilidad de prueba	4	5	5	5	3	4	5
Conformidad	5	5	5	5	4	4	4
PORTABILIDAD							
Adaptabilidad	5	3	3	5	5	3	5
Facilidad de instalación	4	5	4	3	4	3	4
Coexistencia	3	5	3	3	5	5	4
Reemplazabilidad	4	4	5	4	5	5	5
Conformidad	5	4	5	4	5	3	3
SUB TOTALES	114	113	117	107	106	103	110
TOTAL	770						
PUNTOS (Promedio)	110						

ANEXO – IV

UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

MANUAL DE USUARIO

En este documento se presenta el Manual de Usuario de la plataforma de Gestión de documentos para la simplificación de trámites documentarios, que tiene como objetivo proporcionar una guía práctica a los usuarios de la Comisión permanente de Fiscalización, para el dominio y aplicación adecuada del sistema, también se precisa cada una de las alternativas del menú principal según el rol de cada usuario, así como las directrices necesarias y las acciones a realizar en cada pantalla.

La Plataforma de Comisión Permanente de Fiscalización – PLAGEDO, ha sido elaborada en la Comisión Permanente de Fiscalización en el marco de las implementaciones de la ley universitaria 30220, con el propósito de mejorar la Gestión documentaria asistiendo de esta forma a los miembros de la Comisión Permanente de Fiscalización de la universidad nacional del altiplano, facilitándoles el tiempo de los procesos y funciones que los integrantes de la Comisión Permanente de Fiscalización realizan.

La principal finalidad de PLAGEDO es contribuir con los integrantes de la Comisión Permanente de Fiscalización evitando innecesarios desplazamientos que no suelen concretarse debidamente.

Reducir tiempo, burocracia y el gasto excesivo de papel, PAGEDO es EcoAmigable. pues no requiere la solicitud de información a toda instancia interna de la universidad puesto que ya dispondríamos toda la información en la plataforma.

VENTAJAS DE PAGEDO

Acceder a PAGEDO

- ✓ desde cualquier punto de acceso de internet, siempre listos para asistirlos.

PAGEDO es Multidispositivo

- ✓ PAGEDO tiene una interfaz que se adecua a cualquier dispositivo como: Laptops, Tablets y Smartphones facilitando a los integrantes de la Comisión permanente de fiscalización desde cualquier lugar y en el dispositivo que se sienta más cómodo o disponible en su momento.





INGRESO AL SISTEMA

1.1. Activación de cuenta

Para activar una cuenta, el usuario deberá estar anticipadamente registrado en la plataforma, para ello el administrador debe ingresar los datos del usuario en el panel de control previa validación de requisitos.

Al intentar ingresar a la plataforma, lo primero que se le mostrará será la ventana de inicio de sesión en donde deberá ingresar la cuenta de usuario y la contraseña.

CREAR TRAMITE

1.2. Ventana de Panel Control

Esta ventana permite visualizar todas las opciones disponibles para el respectivo usuario.

PLAGEDO
Plataforma de Gestión Documentaria

INICIO PREGUNTAS FRECUENTES QYASBEL@GMAIL.COM

COMISIÓN PERMANENTE DE FISCALIZACIÓN

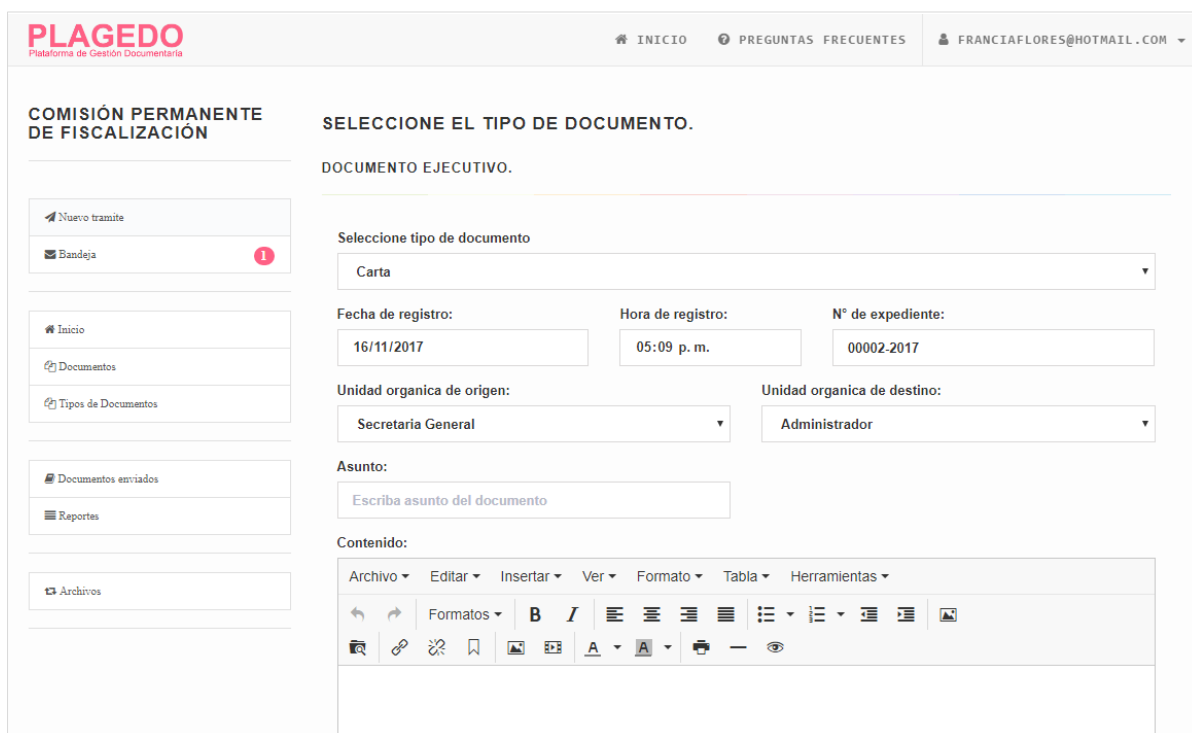
TODOS LOS EXPEDIENTES RECIBIDOS

#	De:	Tipo	Asunto	Estado	Fecha	Opciones

[Nuevo tramite](#)
[Bandeja](#)
[Inicio](#)
[Documentos](#)
[Tipos de Documentos](#)
[Documentos enviados](#)
[Reportes](#)
[Archivos](#)

1.3. TRAMITE DE DOCUMENTARIO

La opción, permite a los miembros crear nuevos trámites para las diferentes oficinas. Para ello clicar en nuevo trámite, seguidamente elaborar tu documento. Como se muestra en la siguiente imagen.



Seguidamente seleccionar el tipo de documento a realizar los campos fecha, hora y número de expediente son automáticos, aunque se podría crear en casos contrario no exista ese número en los registros también se deberá elegir la unidad orgánica de destino para terminar llenar el asunto y el respectivo contenido del documento para finalizar clic sobre el botón

REALIZAR TRÁMITE.

The screenshot shows the 'COMISIÓN PERMANENTE DE FISCALIZACIÓN' interface. It includes a sidebar with navigation options like 'Nuevo tramite', 'Bandeja', 'Inicio', 'Documentos', 'Tipos de Documentos', 'Documentos enviados', 'Reportes', and 'Archivos'. The main area is titled 'SELECCIONE EL TIPO DE DOCUMENTO.' and contains a form for 'DOCUMENTO EJECUTIVO.' with fields for 'Oficio', 'Fecha de registro', 'Hora de registro', 'N° de expediente', 'Unidad organica de origen', 'Unidad organica de destino', and 'Asunto'. A rich text editor is present for the 'Contenido:' field. At the bottom, there is a 'REALIZAR TRÁMITE' button and copyright information for the 'Plataforma de Gestión Documentaria'.

1.4. TRÁMITES REALIZADOS

Ya el documento registrado con sus respectivos datos se podrá gestionar todos los documentos enviados en la ventana de administración de documento las cuales tendrá las opciones visualizar, editar (por un tiempo limitado de 5 minutos para evitar errores propios y éste se desactivará) e imprimir el documento.

TODOS LOS EXPEDIENTES REALIZADOS							Para buscar ingrese número de expediente
Exp. #	De:	Para:	Documento	Asunto	Estado	Fecha	Opciones
4	Michael, Cutipa Kallo	Trámite documentario	Oficio	Requerimiento de dispositivos móviles	Enviado	2017-11-18 00:39:00	
2	Michael, Cutipa Kallo	Abastecimiento	Oficio	respecto al Informe	Enviado	2017-11-13 13:11:00	

1.5. BANDEJA DE ENTRADA DE LOS DOCUMENTOS

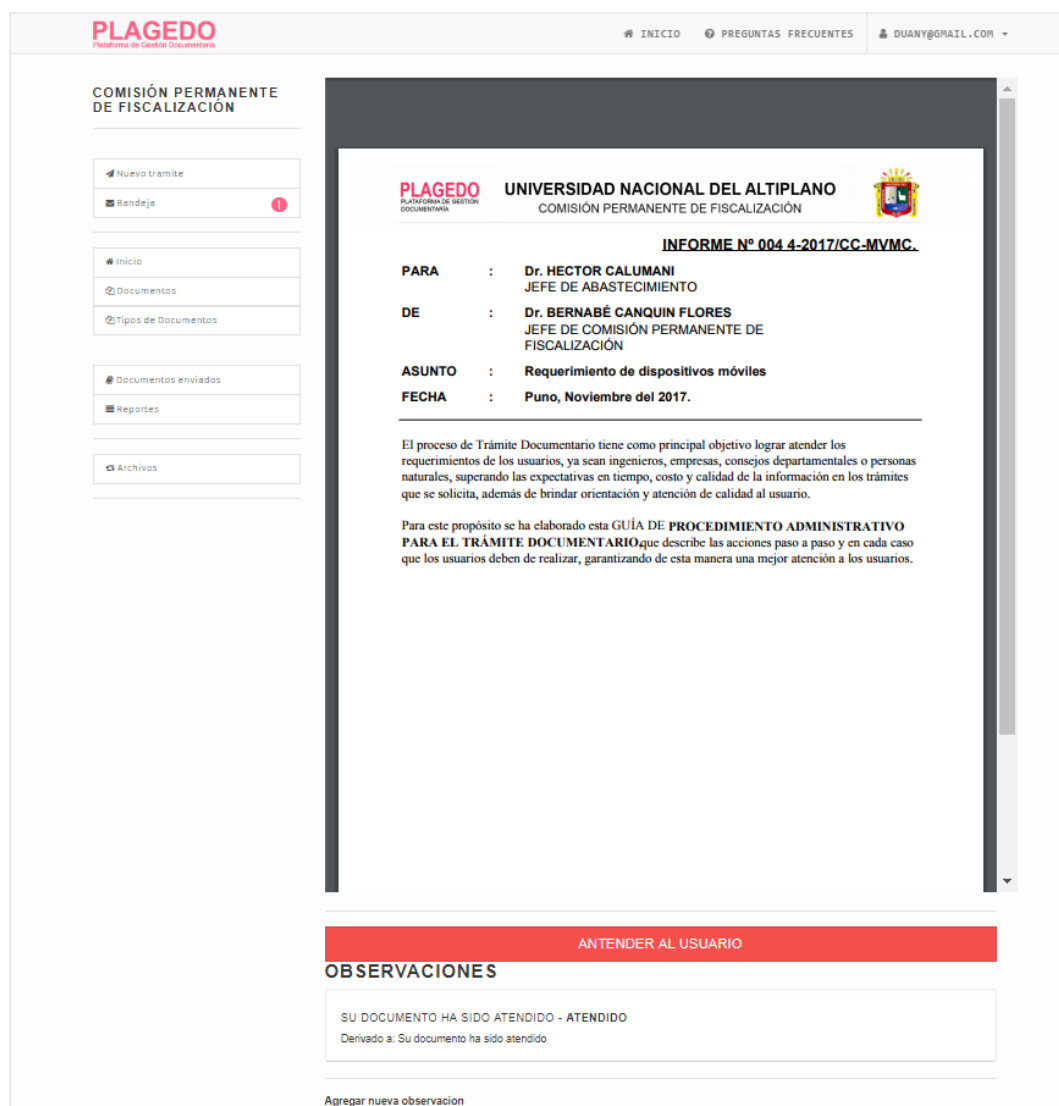
Documentos recibidos para ello el usuario tomará acciones sobre el documento recibido para atender al usuario tramitante se dispone de dos botones visualizar y derivar directamente.

The screenshot shows the PLAGEDO web application interface. The header includes the logo 'PLAGEDO Plataforma de Gestión Documentaria', navigation links for 'INICIO' and 'PREGUNTAS FRECUENTES', and a user profile for 'MAICKOL@GMAIL.COM'. The main content area is titled 'COMISIÓN PERMANENTE DE FISCALIZACIÓN' and 'TODOS LOS EXPEDIENTES RECIBIDOS'. A search bar is present with the placeholder text 'Para buscar ingrese número de expediente'. Below the search bar is a table with the following data:

#	De:	Tipo	Asunto	Estado	Fecha	Opciones
1	Milton Vladimir Calisaya - Abastecimiento	Informe	Informe de actividades realizadas corresspondientes al mes de noviembre	Recibido	13/11/2017, 06:50:00	LEER DERIVAR

On the left side of the interface, there is a sidebar menu with the following items: 'Nuevo tramite', 'Bandeja' (with a red notification badge '1'), 'Inicio', 'Documentos', 'Tipos de Documentos', 'Documentos enviados', 'Reportes', and 'Archivos'.

Visualización del documento en el siguiente pantallazo.



PLAGEDO
Plataforma de Gestión Documentaria

INICIO PREGUNTAS FRECUENTES DUANY@GMAIL.COM

COMISIÓN PERMANENTE DE FISCALIZACIÓN

Nuevo trámite
Bandeja 1
Inicio
Documentos
Tipos de Documentos
Documentos enviados
Reportes
Archivos

PLAGEDO UNIVERSIDAD NACIONAL DEL ALTIPLANO
COMISIÓN PERMANENTE DE FISCALIZACIÓN

INFORME N° 004 4-2017/CC-MVMC.

PARA : Dr. HECTOR CALUMANI
JEFE DE ABASTECIMIENTO

DE : Dr. BERNABÉ CANQUIN FLORES
JEFE DE COMISIÓN PERMANENTE DE FISCALIZACIÓN

ASUNTO : Requerimiento de dispositivos móviles

FECHA : Puno, Noviembre del 2017.

El proceso de Trámite Documentario tiene como principal objetivo lograr atender los requerimientos de los usuarios, ya sean ingenieros, empresas, consejos departamentales o personas naturales, superando las expectativas en tiempo, costo y calidad de la información en los trámites que se solicita, además de brindar orientación y atención de calidad al usuario.

Para este propósito se ha elaborado esta **GUÍA DE PROCEDIMIENTO ADMINISTRATIVO PARA EL TRÁMITE DOCUMENTARIO** que describe las acciones paso a paso y en cada caso que los usuarios deben de realizar, garantizando de esta manera una mejor atención a los usuarios.

ANTENDER AL USUARIO

OBSERVACIONES

SU DOCUMENTO HA SIDO ATENDIDO - ATENDIDO
Derivado a: Su documento ha sido atendido

Agregar nueva observacion

1.6. ESTADOS

Los estados son para el control de los trámites éstos cuentan con 5 estados.

- ✓ Enviado
- ✓ Recibido
- ✓ Leído
- ✓ Derivado
- ✓ Finalizado(atendido)

Esto se podrá visualizar en la lista de documentos atendidos

PLAGEDO
Plataforma de Gestión Documentaria

INICIO PREGUNTAS FRECUENTES DUANY@GMAIL.COM

COMISIÓN PERMANENTE DE FISCALIZACIÓN

TODOS LOS EXPEDIENTES RECIBIDOS

Para buscar ingrese número de expediente

Nuevo tramite

Bandeja **1**

#	De:	Tipo	Asunto	Estado	Fecha	Opciones
4	Michael Cutipa Kallo - Comisión Permanente de Fiscalización	Oficio	Requerimiento de dispositivos móviles	Finalizado	18/11/2017, 00:39:00	LEER

PLAGEDO
Plataforma de Gestión Documentaria

INICIO PREGUNTAS FRECUENTES DUANY@GMAIL.COM

COMISIÓN PERMANENTE DE FISCALIZACIÓN

TODOS LOS EXPEDIENTES RECIBIDOS

Para buscar ingrese número de expediente

Nuevo tramite

Bandeja **1**

#	De:	Tipo	Asunto	Estado	Fecha	Opciones
4	Michael Cutipa Kallo - Comisión Permanente de Fiscalización	Oficio	Requerimiento de dispositivos móviles	Derivado	18/11/2017, 00:39:00	LEER

ANEXO – V

CÓDIGO FUENTE

```
01 <?php
02 /*****
03 *   Plataforma de Gestión Documentaria
04 *   version : 1.1
05 *   Programador: Bach. Milton Vladimir Mamani Calisaya
06 *   Architect: Bach. Milton Vladimir Mamani Calisaya
07 *   Revisor: Dr. Bernabé Canqui Flores
08 *
09 *****/
10
11 define('ENVIRONMENT', 'development');
12
13 if (defined('ENVIRONMENT'))
14 {
15     switch (ENVIRONMENT)
16     {
17         case 'development': error_reporting(E_ALL); break;
18         case 'testing':
19         case 'production': error_reporting(0); break;
20         default: exit('The application environment.');
```

```
28 }
29 if (realpath($system_path) !== FALSE)
30 {
31     $system_path = realpath($system_path).'/' ; }
32
33 // ensure there's a trailing slash
34 $system_path = rtrim($system_path, '/').'/';
35
36 // Is the system path correct? if ( ! is_dir($system_path))
37 {
38     exit("Your system folder path does not appear to
39 be set correctly. Please open the following file
40 correct: ".pathinfo(__FILE__, PATHINFO_BASENAME));
41 } require_once BASEPATH.'core/CodeIgniter.php';
42 /* End of file index.php */
43 /* Location: ./index.php */
44
45
46
47
48
49
50
51
52
53
```

Implementación del modelo de la creación de los documentos.

```
01 <?php
02 defined('BASEPATH') OR exit('No direct script access allowed');
03 class Bandeja_model extends CI_Model{
04     public function __construct(){
05     }
06     public function get_tramites($url = FALSE, $limit = FALSE,
07     $offset = FALSE)
08     {
09         if($limit){
10             $this->db->limit($limit, $offset);
11         }
12         if($url === FALSE){
13             $this->db->order_by('documento.id_documento',
14             'DESC');
15             $this->db->join('tipo_documento',
16             'tipo_documento.id_tipo_documento =
17             documento.tipo_documento_id_tipo_documento');
18             $this->db->join('area', 'area.id_area =
19             documento.area_id_area');
20             $this->db->join('estado', 'estado.id_estado =
21             documento.estado_id_estado');
22             $this->db->join('usuarios', 'usuarios.idusuarios =
23             documento.usuarios_idusuarios');
24             $query = $this->db->get('documento');
25             return $query->result_array();
26         }
27         $query = $this->db->get_where('documento',
28         array('id_documento' => $url));
29         return $query->row_array();
30     }
31     public function create_tramite(){
32         $documento_registro = url_title($this->input-
33         >post('registro'));
```

```
30         $data = array(
31             'documento_nombre' => $this->input-
32 >post('nombre'),
33             'documento_registro' => $documento_registro,
34             'documento_hora' => $this->input->post('hora'),
35             'documento_fecha' => $this->input-
36 >post('fecha'),
37             'documento_contenido' => $this->input-
38 >post('contenido'),
39             // id usuario destino
40             'to_area_id' => $this->input->post('to_a'),
41             // id area destino
42             'aread_id' => $this->input->post('destino'),
43             // id area origen
44             'area_id_area' => $this->session-
45 >userdata('user_id'),
46             'tipo_documento_id_tipo_documento' => $this-
47 >input->post('tipo'),
48             'estado_id_estado' => 1,
49             'usuarios_idusuarios' => $this->session-
50 >userdata('user_id')
51         );
52         return $this->db->insert('documento', $data);
53     }
54
55     public function get_tipo(){
56         $this->db->order_by('tipo_doc_nombre');
57         $query = $this->db->get('tipo_documento');
58         return $query->result_array();
59     }
60
61     public function get_area(){
62         $this->db->order_by('area_nombre');
63         $query = $this->db->get('area');
64     }
```

```
60         return $query->result_array();
61     }
62     public function get_origen(){
63         $this->db->order_by('area_nombre');
64         $query = $this->db->get('area');
65         return $query->result_array();
66     }
67     public function
68 get_tramites_by_tipo($tipo_documento_id_tipo_documento){
69         $this->db->order_by('documento.id_documento',
70 'DESC');
71         $this->db->join('tipo_documento',
72 'tipo_documento.id_documento =
73 documento.tipo_documento_id_tipo_documento');
74         $query = $this->db->get_where('documento',
75 array('tipo_documento_id_tipo_documento' =>
76 $tipo_documento_id_tipo_documento));
77         return $query->result_array();
78     }
79 }
80
81 /* End of file tramite_model.php */
82
83 /* Location:
84 ./application/models/trami
85 te_model.php */
```