

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



“Plataforma web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado UNA Puno 2016.”

**TESIS**

**PRESENTADA POR:**

**Bach. FRED TORRES CRUZ**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ESTADÍSTICO E INFORMÁTICO**

**PUNO – PERÚ**  
**2016**

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

“PLATAFORMA WEB BASADA EN CLOUD COMPUTING PARA EL  
SEGUIMIENTO DE PROYECTOS DE TESIS DE PREGRADO UNA PUNO – 2016”

TESIS

PRESENTADA POR:

Bach. FRED TORRES CRUZ

Para optar el Título Profesional de:

INGENIERO ESTADÍSTICO E INFORMÁTICO

APROBADA POR:

PRESIDENTE:

Dr. BERNABÉ CANQUI FLORES

PRIMER MIEMBRO:

M.Sc. EMMA AZAÑERO DE AGUIRRE

SEGUNDO MIEMBRO:

Dr. REYNALDO SUCARI LEÓN

DIRECTOR:

M.Sc. ERNESTO NAYER TUMI FIGUEROA

ASESOR:

M.Sc. RAMIRO PEDRO LAURA MURILLO

ÁREA: Informática

TEMA: Sistemas de Información

## DEDICATORIA

A mis Padres por darme la vida y estar ahí cuando más los necesité; y sobre todo por su amor, cariño, paciencia, ayuda, y constante cooperación.

A mi hermano Edward para quien espero ser la competencia que el deberá superar.

Y por sobre todo a aquellos que esperaban el fracaso de PILAR como mi proyecto de grado.

## AGRADECIMIENTOS

Agradezco a la Universidad Nacional del Altiplano Puno, y a la Facultad de Ingeniería Estadística e Informática por la Formación Profesional que me brindó en sus claustros de Educación Superior. Así como los profesores Nayer Tumi Figueroa, Bernabe Canqui Flores y Wencesalo Medina Espinoza, cada uno reúne características que han contribuido en mi formación como persona y ahora como profesional del área de tecnologías de la información.

A Ramiro Pedro y Romel Percy por ser muy buenos amigos, colegas y consejeros resaltando que todo con una sonrisa y un bailecito puede estar mucho mejor.

A los compañeros de CINTIA (Centro de Investigación en Tecnología e Innovación del Altiplano) por su interés y motivación para realizar más investigación.

Sin dejar de mencionar a mis compañeros de aula y amigos quienes me acompañaron durante mi estancia en esta casa superior de estudios.

## INDICE

RESUMEN .....	8
ABSTRACT .....	9
INTRODUCCIÓN .....	10
CAPÍTULO I <u>PROBLEMÁTICA DE LA INVESTIGACIÓN</u> .....	12
1.1 PLANTEAMIENTO DEL PROBLEMA .....	12
1.2 JUSTIFICACIÓN DE LA INVESTIGACIÓN .....	13
1.3 OBJETIVOS DE LA INVESTIGACIÓN .....	15
1.3.1 OBJETIVO GENERAL .....	15
1.3.2 OBJETIVOS ESPECÍFICOS .....	15
1.4 HIPOTESIS DE LA INVESTIGACIÓN .....	15
CAPÍTULO II <u>BASE TEÓRICA REFERENCIAL</u> .....	16
2.1 ANTECEDENTES DE LA INVESTIGACIÓN .....	16
2.2 BASE TEÓRICA .....	17
2.2.1 COMPUTACION EN LA NUBE .....	17
2.2.2 IMPLEMENTACIÓN DE APLICACIONES WEB .....	18
2.2.3 TRANSFERENCIA REMOTA DE ARCHIVOS .....	18
2.2.4 SOFTWARE AS A SERVICE (SaaS) .....	19
2.2.5 INFRAESTRUCTURA AS A SERVICE (IaaS) .....	19
2.2.6 PLATAFORMA AS A SERVICE (PaaS) .....	20
2.2.7 SISTEMAS DE INFORMACIÓN .....	20
2.2.8 SISTEMA VIRTUAL .....	22
2.2.9 PROTOTIPO DE SISTEMA .....	22
2.2.10 MODELO DE PROCESO DEL SOFTWARE .....	23
2.2.11 LENGUAJE UNIFICADO DE MODELADO O UML .....	27
2.2.12 PRUEBAS O MÉTRICAS DEL SOFTWARE .....	32
2.2.13 ORGANIZACIÓN INTERNACIONAL DE ESTANDARIZACION 9126 .....	33
2.2.14 BASE DE DATOS .....	38
2.2.15 METODOLOGÍA DE PROGRAMACIÓN .....	51
2.2.16 WORLD WIDE WEB O WWW .....	56

2.2.17	MYSQL .....	57
2.2.18	HTML .....	58
2.2.19	PHP .....	59
2.2.20	AJAX.....	60
2.2.21	JQUERY .....	61
2.2.22	JAVASCRIPT .....	63
2.2.23	PHPMYADMIN .....	64
2.2.24	FRAMEWORK.....	64
2.2.25	BOOTSTRAP.....	65
2.2.26	CODEIGNITER.....	66
2.3	DEFINICIÓN DE TÉRMINOS BÁSICOS .....	66
CAPÍTULO III MATERIALES Y MÉTODOS.....		68
3.1.	LOCALIZACIÓN .....	68
3.2.	POBLACIÓN Y MUESTRA.....	68
3.3.	OPERACIONALIZACIÓN DE VARIABLES.....	70
3.4.	RECOPIACIÓN DE LA INFORMACIÓN .....	71
3.5.	MODELO Y METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	71
3.5.1.	MODELO DE DESARROLLO .....	71
3.6.	ANÁLISIS DE PLANIFICACIÓN PARA EL DESARROLLO .....	73
3.7.	DIAGRAMA DEL PROCESO DETALLADO.....	74
3.7.1.	REQUERIMIENTOS DE LA PLATAFORMA .....	74
CAPÍTULO IV RESULTADOS Y DISCUSIONES.....		78
4.1.	ANÁLISIS Y DISEÑO DE LA PLATAFORMA .....	78
4.1.1.	DESCRIPCIÓN DEL PLATAFORMA DE INSCRIPCIÓN INICIAL .....	78
4.2.	DISEÑO DE LA PLATAFORMA.....	80
4.3.	MODELAMIENTO DE LA PLATAFORMA MEDIANTE UML7.....	80
4.4.	VERSIONES E ITERACIONES.....	81
4.5.	CODIFICACIÓN .....	85
4.6.	PRUEBAS.....	86
4.7.	INSTALACIÓN .....	86
4.8.	RESULTADOS DE LA PRUEBA SEGÚN EL ISO – 9126.....	87
4.9.	CUADRO COMPARATIVO SEGÚN EL TIEMPO .....	88

<b>CONCLUSIONES .....</b>	<b>94</b>
<b>RECOMENDACIONES Y SUGERENCIAS .....</b>	<b>95</b>
<b>BIBLIOGRAFÍA.....</b>	<b>96</b>
<b>ANEXOS.....</b>	<b>100</b>
<b>ANEXO – I.....</b>	<b>101</b>
<b>FICHA DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO ESTÁNDAR .....</b>	<b>101</b>
<b>ANEXO– III.....</b>	<b>107</b>
<b>ANEXO– IV .....</b>	<b>115</b>

## RESUMEN

Es tendencia en el desarrollo de Tecnologías de la Información, la creación de plataformas basadas en los principios del Cloud Computing (Computación en la Nube) porque nos ofrece muchas ventajas de administración, mantenimiento, escalabilidad y adaptabilidad a la hora de crear nuevos servicios. Además por el problema del acceso a la información, la burocracia y el tiempo que le toma a un tesista el proceso de inscripción, registro, sorteo, revisión, corrección y dictamen de proyectos de tesis, en el Vicerrectorado de Investigación de la Universidad Nacional del Altiplano Puno nace la necesidad de desarrollar e implementar una plataforma web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado para poder brindar un servicio ágil, eficiente y seguro en todo el procedimiento de los proyectos de tesis. El desarrollo e implementación de la plataforma se centró en la aplicación de la metodología de desarrollo de software, se utilizó las metodologías SCRUM y XP las cuales se adecuaron mejor a la plataforma por su agilidad y bajo costo. Para el modelado de la plataforma se ha empleado la metodología UML (Lenguaje Unificado de Modelado) y para ver la calidad de la plataforma que se desarrolló, se utilizó la ficha de evaluación de calidad del producto estándar ISO-9126, además de una encuesta de satisfacción de usuarios. Se concluyó que la implementación de la plataforma web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado agilizó el proceso de inscripción, registro, sorteo, revisión, corrección y dictamen de proyectos de tesis de pregrado y como consecuencia la satisfacción de los tesistas que han acelerado el proceso de sus proyectos.

**Palabras claves:** Cloud, Plataforma Virtual, Scrum, Digitalización, web.

## ABSTRACT

It is the trend in the development of Information Technologies, the creation of platforms based on the principles of Cloud Computing because they offer many advantages of administration, maintenance, scalability and adaptation when creating new services. In addition to the problem of access to information, bureaucracy and the time taken by a panel of the registration process, registration, draw, review, correction and opinion of thesis projects, at the Vice-Rectorate of Research of the National University of the Altiplano Puno is born the need to develop and implement a web platform based on cloud computing for the monitoring of undergraduate thesis projects to be able to provide a service, efficient and secure throughout the procedure of thesis projects. The development and implementation of the platform focused on the application of software development methodology, using SCRUM and XP methodologies which were better adapted to the platform due to its agility and low cost. For the platform model, the UML (Unified Modeling Language) methodology has been used and to see the quality of the platform that was developed, the ISO-9126 product quality assessment sheet was used, as well as a Satisfaction Survey Of users. It was concluded that the implementation of the web platform based on cloud computing for the follow-up of undergraduate thesis projects streamlined the registration, registration, draw, review, correction and opinion process of undergraduate thesis projects and as a consequence the satisfaction of the Tesistas who have accelerated the process of their projects.

**Keywords:** Cloud, Virtual Platform, Scrum, Digitization, web.

## INTRODUCCIÓN

En el presente trabajo propone el diseño e implementación de una Plataforma Web Basada en Cloud Computing para el Seguimiento de Proyectos de Tesis, enfocando el trabajo a la optimización de los procesos de inscripción, registro, sorteo, revisión, corrección y dictamen de los proyectos de tesis, haciendo uso de la teoría de Cloud Computing, (SaaS, IaaS, PaaS).

Por lo que para el diseño de la misma se realizó un proceso previo de recolección de información de las autoridades, coordinadores, docentes y estudiantes que participan en este procedimiento de presentación y aprobación de proyectos de tesis 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 inscripción registro, sorteo, revisión, corrección y dictamen de proyectos de tesis, se realizaron capacitaciones previas al uso de la plataforma, lo que ayudaron a que los docentes y estudiantes puedan optimizar el tiempo de aprendizaje además del tiempo en cada uno de estos procesos en la presentación de proyectos de tesis.

En el Capítulo I se presenta el Plan de Investigación 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

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.

## CAPÍTULO I

### PROBLEMÁTICA DE LA INVESTIGACIÓN

#### 1.1 PLANTEAMIENTO DEL PROBLEMA

En el marco de la nueva Ley Universitaria 30220, el Vicerrectorado de Investigación es el organismo de más alto nivel en la universidad en el ámbito de la investigación. Está encargado de orientar, coordinar y organizar los proyectos y actividades que se desarrollan a través de las diversas unidades académicas. Organiza la difusión del conocimiento y promueve la aplicación de los resultados de las investigaciones, así como la transferencia tecnológica y el uso de las fuentes de investigación, integrando fundamentalmente a la universidad, la empresa y las entidades del Estado. (Ley N° 30220, 2014, Art. 50)

Otra vertiente del problema es el acceso de la información por los egresados de la Universidad Nacional del Altiplano Puno quienes aún no iniciaron el

trámite, ya sea por situación laboral, residencia y otras razones que impiden el trámite por ser presencial.

Por todo ello, no existe en la actualidad ninguna plataforma que pueda clasificar a estos docentes y estudiantes en sus respectivas líneas y así puedan estos interactuar con el Vicerrectorado de Investigación desde cualquier ubicación geográfica para continuar el trámite del título profesional, lo que otorga a la investigación un valor añadido al tratar de responder a todas las preguntas que rodean a este problema desde un punto de vista novedoso, innovador y multidisciplinario.

De acuerdo a la problemática anterior se formula la siguiente interrogante:

¿Agilizará la Plataforma de Seguimiento de Tesis el Proceso de inscripción, registro, sorteo, revisión, corrección y dictamen de los proyectos y borradores en la Universidad Nacional del Altiplano?

## **1.2 JUSTIFICACIÓN DE LA INVESTIGACIÓN**

La plataforma web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado UNA Puno 2016, busca administrar de manera ágil, eficiente y segura toda la información que se obtiene de la inscripción, registro, sorteo, revisión y dictamen de los proyectos de tesis en la Universidad Nacional del Altiplano Puno, además de poder monitorear todo este procedimiento en forma detallada, transparente y automatizada a cada uno de los usuarios de la plataforma.

El Vicerrectorado de Investigación de la Universidad Nacional del Altiplano para hacer la organización de proyectos y actividades cuenta en cada una de las dependencias de las Escuelas Profesionales una Coordinación de Investigación, la cual tiene como funciones administrar el proceso de presentación y sustentación de tesis, informes y examen de suficiencia profesional, actualmente los procesos que involucran proyectos de tesis e informes, tienen un registro a través de hojas de cálculo y archivos físicos, por lo que se propuso una solución de acuerdo a la actualidad y con tecnología que esté a nuestro alcance.

La realidad universitaria en nuestro día a día en el ámbito de la tecnologías de la información actualmente a la hora de afrontar un proyecto en el área de investigación, está basada en un modelo en el cuál un estudiante o un docente esta propenso a estar afectado por la burocracia, el cual es un escenario en el cuál se pone la solución encima de la mesa, esta debe ser a medida, y que esta misma esté orientada al docente y al estudiante, la cual finalmente agilice el tiempo desde la presentación hasta la aprobación del proyecto de tesis y en lo posterior también el borrador de tesis así como también brinde transparencia en cada uno de estos procesos.

Y a todo esto se añade que docentes y estudiantes puedan elegir sus propias áreas, líneas y temas de investigación de acuerdo a la temática del proyecto a ser presentado, al mismo tiempo que los docentes que conformen parte del jurado dictaminador sean especialistas en la línea de investigación a la cual

está referida el trabajo de investigación de esta manera garantizar la calidad del trabajo de investigación que se está realizando.

### **1.3 OBJETIVOS DE LA INVESTIGACIÓN**

#### **1.3.1 OBJETIVO GENERAL**

Desarrollar e implementar una plataforma web basada en cloud computing para agilizar el proceso de inscripción, registro, sorteo, revisión, corrección y dictamen de los proyectos de tesis de pregrado en la Universidad Nacional del Altiplano Puno 2016 .

#### **1.3.2 OBJETIVOS ESPECÍFICOS**

- ✓ Agilizar el procedimiento de inscripción, registro, sorteo, revisión, corrección y dictamen de los proyectos de tesis.
- ✓ Implementar la seguridad adecuada del acceso a la información
- ✓ Implementar la seguridad adecuada del acceso a la información.
- ✓ Determinar las ventajas de la Plataforma con su implementación.

### **1.4 HIPOTESIS DE LA INVESTIGACIÓN**

La implementación de la Plataforma de web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado agilizará el proceso de inscripción, registró, sorteo, revisión, corrección y dictamen de los proyectos de tesis de pregrado en la Universidad Nacional del Altiplano 2016.

## CAPÍTULO II

### BASE TEÓRICA REFERENCIAL

#### 2.1 ANTECEDENTES DE LA INVESTIGACIÓN

Andrade (2001), Eficacia de sistemas de informação e percepção de mudança organizacional: un estudio de caso, uno de sus objetivos fue verificar en qué medida la evaluación de la eficacia del sistema de información; desde el punto de vista de los usuarios se asocia con los niveles de mejora en la calidad percibida de la atención al paciente. Y concluye que SI considera eficaz, a medida que se percibe como un elemento que contribuye al alcance de los objetivos organizacionales. De ahí la importancia de la visión de las reales necesidades de informatización debe de ser compartida por todos y que, desde ahí sean definidos objetivos claros con transparencia y sobre todo con coherencia.

Martín (2010), Filosofía Lean aplicada a la Ingeniería del Software, Cuyo objetivo define la exposición del estado del arte actual de la aplicación de los principios de la filosofía Lean a la Ingeniería de Software y servir de base para futuros trabajos: de ampliación práctica más allá de la exposición de la

situación actual y que pueda servir para la elaboración de una guía de implementación. Y concluye que la aplicabilidad de las técnicas y prácticas Lean en esta disciplina se pone de manifiesto y los resultados obtenidos en las experiencias notablemente son positivos.

Quispe (2014), Sistema Informático de Gestión Administrativa para la coordinación de Investigación de la Facultad de Ingeniería Estadística e Informática de la UNA PUNO 2014, Cuyo objetivo fue, la implementar un sistema para la gestión administrativa de la coordinación de investigación, de la FINESI. Y concluye que la implementación del sistema SIGACI permite realizar los procesos descritos en menor tiempo.

## **2.2 BASE TEÓRICA**

### **2.2.1 COMPUTACION EN LA NUBE**

El almacenamiento en nube o almacenamiento en la nube (del inglés cloud storage), es un modelo de almacenamiento de datos basado en redes, ideado en los «años 1960», donde los datos están alojados en espacios de almacenamiento virtualizados, por lo general aportados por terceros. Las compañías de alojamiento operan enormes centros de procesamiento de datos. Los usuarios que requieren estos servicios compran o alquilan la capacidad de almacenamiento necesaria. Los operadores de los centros de datos, a nivel servicio, virtualizan los recursos según los requerimientos del cliente. Solo exhiben los entornos con los recursos requeridos. Los clientes administran el almacenamiento y el funcionamiento de los archivos, datos o

aplicaciones. Físicamente los recursos pueden estar repartidos en múltiples servidores físicos.

### **2.2.2 IMPLEMENTACIÓN DE APLICACIONES WEB**

HTML5 está reemplazando previos complementos o plugins, como Flash o Java applets, por nuevas y mejores tecnologías ahora incluso WebGL y Canvas para representación gráfica de datos. El desarrollo web contribuyen a la simplificación de la interfaz de desarrollo sobre el problema real de transportabilidad de la programa final, como el caso particular de un correo electrónico o página web que únicamente requiere ejecutarse sobre un navegador web, dejando al desarrollador concentrarse en el algoritmo y el objetivo planteado. Las aplicaciones web están diseñadas para apoyar el desarrollo de sitios web dinámicos, aplicaciones y servicios web. Las aplicaciones Web intentan aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web. Por ejemplo, muchos de ellos proporcionan bibliotecas para acceder a bases de datos, administración general de archivos y carpetas, mantenimiento de ellas, estructuras para plantillas y gestión de sesiones, y con frecuencia facilitan la reutilización de código.

### **2.2.3 TRANSFERENCIA REMOTA DE ARCHIVOS**

El desarrollo de la banda ancha es un fenómeno en crecimiento, observable a diario. Los Proveedores de Servicios en Internet, cuya sigla en inglés es ISP, están continuamente haciendo mejoras en el ancho de banda a costos cada

vez más competitivos. A manera de ejemplo, lo que se pagaba hace una década por un ancho de banda de 56 kpbs, la veinteava parte de 1Mb, vía teléfono y un módem dedicado, con tarifa variable de acuerdo con el uso del teléfono, es muy similar a lo que se paga actualmente por un enlace de 1Mb, con tarifa plana, vía módem ADSL, no dedicado, es decir, con la línea de teléfono libre para hacer llamadas telefónicas. Esta situación tenderá a mejorar en el futuro, ya que frecuentemente los ISP aumentan el ancho de banda y mantienen los costos en el mismo nivel. La capacidad de acceso a la Internet. La transferencia de archivos sobre un servicio de internet básico de 2MB sobre un archivo de 4MB no toma más de 4Minutos, garantizando de este modo la administración de información de gran tamaño, como Planos, libros o compendios escaneados.

#### **2.2.4 SOFTWARE AS A SERVICE (SaaS)**

En español Software como Servicio. Modelo de distribución de software donde una empresa sirve el mantenimiento, soporte y operación que usará el cliente durante el tiempo que haya contratado el servicio. El cliente usará el sistema alojado por esa empresa, la cual mantendrá la información del cliente en sus sistemas y proveerá los recursos necesarios para explotar esa información.

#### **2.2.5 INFRAESTRUCTURE AS A SERVICE (IaaS)**

En español Infraestructura como Servicio. Modelo de distribución de infraestructura de computación como un servicio, normalmente mediante una plataforma de virtualización. En vez de adquirir servidores, espacio en un

centro de datos o equipamiento de redes, los clientes compran todos estos recursos a un proveedor de servicios externo. Una diferencia fundamental con el hosting virtual es que el provisionamiento de estos servicios se hace de manera integral a través de la web.

### 2.2.6 PLATAFORMA AS A SERVICE (PaaS)

En español Plataforma como Servicio. Aunque suele identificarse como una evolución de SaaS, es más bien un modelo en el que se ofrece todo lo necesario para soportar el ciclo de vida completo de construcción y puesta en marcha de aplicaciones y servicios web completamente disponibles en la Internet. Otra característica importante es que no hay descarga de software que instalar en los equipos de los desarrolladores. PaaS ofrece múltiples servicios, pero todos provisionados como una solución integral en la web.

### 2.2.7 SISTEMAS DE INFORMACIÓN

Un sistema de Información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

**Entrada de Información:** Es el proceso mediante el cual el sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas. Las unidades

típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáneres, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

**Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

**Procesamiento de Información:** Es la capacidad del sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

**Salida de Información:** La salida es la capacidad de un sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales,

diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un sistema de Información puede constituir la entrada a otro sistema de Información o módulo. En este caso, también existe una interface automática de salida.

### **2.2.8 SISTEMA VIRTUAL**

En Informática, una computadora es el sistema formado por el hardware y software y su sistema operativo. Conjunto integrado de aplicaciones usuario-máquina para proveer la información que apoye a las operaciones, a la administración y las funciones de toma de decisiones en una empresa. El sistema usa equipo de computación y software.<sup>1</sup>

### **2.2.9 PROTOTIPO DE SISTEMA**

Surgen a partir de la realización del análisis, lo cual implica aplicar una o más técnicas tales como: factores críticos para el éxito, análisis de datos, árboles de decisión, diagramas de flujos de datos, diagramas orientados a objetos, grupos de trabajo, diagramas de efectos y correspondencias. Todos estos permiten al usuario verificar los resultados que han sido generados por el prototipo<sup>2</sup>.

Los prototipos son modelos que permiten a las personas necesitadas, crear un modelo de software implantando una de las tres formas siguientes:

---

<sup>1</sup>GORDON D. DAVIS: "Sistema de información Gerencial" Pág. Nº 9 -10 -2001

<sup>2</sup>Graham, I. "Métodos Orientados a Objetos". Pág. 125 2° Edición Edit. Addison-Wesley Iberoamericana S.A, 1996

- Un prototipo en papel o basado en un computador en el que se describa la interacción hombre-máquina, de modo que facilite al usuario la comprensión de cómo se producirá la interacción.
- Un prototipo que implemente algunas operaciones deseadas para la solución de requerimientos de información.
- Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

#### **2.2.10 MODELO DE PROCESO DEL SOFTWARE**

##### **METODOLOGÍA SCRUM**

Scrum es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. Las reglas de Scrum relacionan los eventos, roles y artefactos,

gobernando las relaciones e interacciones entre ellos. Scrum se basa en la teoría de control de procesos empírica. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación

Transparencia.- Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo.

Inspección.- Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.

Adaptación.- Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.

Scrum prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación, tal y como se describen en la sección Eventos de Scrum del presente documento.

Reunión de Planificación del Sprint (Sprint Planning Meeting)

Scrum Diario (Daily Scrum)

Revisión del Sprint (Sprint Review)

Retrospectiva del Sprint (Sprint Retrospective)

**Fig. N° 1 Roles, artefactos y eventos principales de SCRUM**



Fuente: Deemer et al., (2009)

**Ingeniería y Análisis del Sistema:**

Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

**Análisis de los requisitos del software:**

El proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El Ingeniero de Software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

**Diseño:**

El diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

**Codificación:**

El diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

**Prueba:**

Una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

**Mantenimiento:**

El software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debido a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

En principio, el resultado de cada fase es uno o más documentos aprobados. La siguiente fase no debe empezar hasta que la fase previa haya finalizado. En la práctica, estas etapas se suponen y proporcionan información a las otras. Durante el diseño se identifican los problemas con los requerimientos durante el diseño del código se encuentran problemas, y así sucesivamente. El proceso del software no es un módulo lineal simple, sino que implica una serie de iteraciones de las actividades de desarrollo.

Las ventajas del modelo en cascada son que la documentación se produce en cada fase y que este cuadra con otros modelos del proceso de ingeniería. Su principal problema es su inflexibilidad al dividir el proyecto en distintas etapas. Se deben hacer compromisos en las etapas iniciales, los que hace difícil responder a los cambios en los requerimientos del cliente.

**2.2.11 LENGUAJE UNIFICADO DE MODELADO O UML**

El UML es un lenguaje de modelado, y no un método. La mayor parte de los métodos consisten, al menos en principio, en un lenguaje y en un proceso para

modelar. El **lenguaje de modelado** es la notación (principalmente gráfica) de que se valen los métodos para expresar los diseños. El **proceso** es la orientación que nos dan sobre los pasos a seguir para hacer el diseño.

Es el sucesor de la oleada de métodos de análisis y diseño orientados a objetos (OOA & D) que surgió a finales de la década de 1980 y principios de la siguiente. El UML unifica, sobre todo, los métodos de Booch, Rumbaugh(OMT) y Jacobson.<sup>3</sup>

El Lenguaje Unificado de Modelado (UML), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura de decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo

---

<sup>3</sup> Fowler Martin y Scout kendall – UML Gota a Gota–Pág. 1 - 1999

interactivo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.<sup>4</sup>

UML, también contiene construcciones organizativas para agrupar los modelos en paquetes, lo que permite a los equipos de software dividir grandes sistemas en piezas de trabajo, para entender y controlar las dependencias entre paquetes, UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de Lenguajes de Programación, así como construir modelos por ingeniería inversa a partir de programas existentes. El nombre que se le da a un conjunto de métodos unificado como el de Booch, Rumbaugh (OMT) y Jacobson, es un lenguaje de modelado, y no un método, que combina estos tres puntos de vista para el desarrollo del sistema. Para verlo más claro; un panorama del proceso es el siguiente, en orden sucesivo; la concepción, elaboración, construcción y transmisión.

**Concepción:** Establece la razón de ser el proyecto y se determina a su alcance.

**Elaboración:** Reúne requisitos más detallados, se hacen análisis y diseños de alto nivel, a fin de establecer una arquitectura base.

**Construcción:** Confecciona el sistema a lo largo de una serie de iteraciones. Cada iteración es un mini proyecto. Se hace el análisis, diseño, codificación,

---

<sup>4</sup> El Lenguaje de Modelado, Manual de referencia, J. Rumbaugh, Ivar Jacobson Grady Booch. Pag. 3

pruebas e integración de los casos de uso asignados a cada iteración. Ésta termina con el usuario con una demostración y haciendo pruebas del sistema, con el fin de confirmar que se han construido correctamente los casos de uso.

El propósito de este proceso es reducir los riesgos. Los riesgos surgen con frecuencia debido a que las cuestiones difíciles se posponen para el final del proyecto. Las pruebas y la integración son tareas mayores y generalmente tardan más de lo que se cree.

**Transición:** De lo que se trata en el desarrollo iterativo es de hacer todo el proceso de desarrollo consistente, de tal modo que el equipo del desarrollo se acostumbre.

Resulta útil modelar un sistema desde tres puntos de vista distintos, aunque relacionados, cada uno de los cuales captura aspectos importantes del sistema, pero siendo todos ellos necesarios para su descripción completa.

Los conceptos y modelos de UML pueden agruparse en las siguientes áreas conceptuales:

**Estructura estática:** Cualquier modelo preciso debe primero definir su universo, esto es, los conceptos clave de la aplicación, sus propiedades internas, y las relaciones entre cada una de ellas. Este conjunto de construcciones es la estructura estática. Los conceptos de la aplicación son modelados como clases, cada una de las cuales describe un conjunto de objetos que almacenan información y se comunican para implementar un

comportamiento. La información que almacena es modelada como atributos. La estructura estática se expresa con diagramas de clases y puede usarse para generar la mayoría de las declaraciones de estructuras de datos de un programa.

**Comportamiento dinámico:** Hay dos formas de modelar el comportamiento, una es la historia de la vida de un objeto y la forma como interactúa con el resto del mundo y la otra es por los patrones de comunicación de un conjunto de objetos conectados, es decir la forma en que interactúan entre sí. La visión de un objeto aislado es una máquina de estados, muestra la forma en que el objeto responde a los eventos en función de su estado actual. La visión de la interacción de los objetos se representa con los enlaces entre objetos junto con el flujo de mensajes y los enlaces entre ellos. Este punto de vista unifica la estructura de los datos, el control de flujo y el flujo de datos.

**Gestión de modelo:** Representa los aspectos, transformaciones de las funciones del sistema.

Cada uno de estos tres modelos van evolucionando durante el ciclo de desarrollo, el análisis de un modelo del dominio de la aplicación se construye sin tener en cuenta la implementación que se efectuará eventualmente.

En el Lenguaje Unificado de Modelado (UML) cada iteración contiene las etapas usuales del ciclo de vida: análisis, diseño, implementación y experimentación del software, este proceso de desarrollo iterativo y gradual en el sentido de que el software no se libera de un solo gran golpe al final del

proyecto, sino que, al contrario, se desarrolla y se libran las partes. La etapa de construcción consta de muchas iteraciones, donde cada iteración construye un software de calidad para producción, probado e integrado. En un enfoque la construcción del software orientado a objetos que difiere de los enfoques tradicionales del desarrollo del software, y en última instancia al producto del software en sí.

### **2.2.12 PRUEBAS O MÉTRICAS DEL SOFTWARE**

La prueba de software es un elemento crítico para la garantía de calidad del software y representa una visión final de las especificaciones del diseño y de la codificación.

La creciente inclusión del software como un elemento más de muchos sistemas y la importancia de los costos adicionales a un fallo del mismo, están motivando la creación de pruebas minuciosas y bien planificadas. La métrica intenta obtener un conjunto de medidas indirectas que dan lugar a métricas que proporcionan una indicación de la calidad de algún tipo de representación del software.

El objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolos con la menor cantidad de tiempo y de esfuerzo.

Las normas que pueden servir para objetos de prueba son:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.

- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si se descubre un error no detectado hasta entonces.

Si la prueba se lleva a cabo con éxito (de acuerdo con el objetivo anteriormente establecido) descubrirá errores en el software. Como ventaja secundaria, la prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento.<sup>5</sup>

### 2.2.13 ORGANIZACIÓN INTERNACIONAL DE ESTANDARIZACION 9126

**ISO 9126** es un estándar internacional para la evaluación de la calidad del software. Está reemplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos<sup>6</sup>.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido.

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

---

<sup>5</sup> Ingeniería del Software, Roger Pressman, Pág. 302

<sup>6</sup> [http://es.wikipedia.org/wiki/ISO/IEC\\_9126](http://es.wikipedia.org/wiki/ISO/IEC_9126)

- **Funcionalidad.**- Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.
  
- Idoneidad.
- Exactitud.
- Interoperabilidad.
- Seguridad.
- Cumplimiento de normas.
  
- **Fiabilidad.**- Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.
  
- Madurez.
- Recuperabilidad.
- Tolerancia a fallos.
  
- **Usabilidad.**- Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
  
- Aprendizaje.
- Comprensión.
- Operatividad.

- Atractividad.
- **Eficiencia.**- Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.
- Comportamiento en el tiempo.
- Comportamiento de recursos.
- **Mantenibilidad.**- Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.
- Estabilidad.
- Facilidad de análisis.
- Facilidad de cambio.
- Facilidad de pruebas.
- **Portabilidad.**- Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde un sistema a otra.
- Capacidad de instalación.
- Capacidad de reemplazamiento.
- Adaptabilidad.
- Co-Existencia.

La sub-característica Conformidad no está listada arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad.

Cada sub-característica (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así. Como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).

Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales.

Idealmente, la calidad interna no necesariamente implica calidad externa y esta a su vez la calidad en el uso.

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación.

### 2.2.14 BASE DE DATOS

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. A continuación le presentamos una guía que le explicará el concepto y características de las bases de datos.<sup>7</sup>

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una **base de datos** se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más **columnas** y **filas**. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

---

<sup>7</sup> <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

### **Definición de Base de Datos**

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

### **Características**

Entre las principales características de los sistemas de base de datos podemos mencionar:

Independencia lógica y física de los datos.

Redundancia mínima.

Acceso concurrente por parte de múltiples usuarios.

Integridad de los datos.

Consultas complejas optimizadas.

Seguridad de acceso y auditoría.

Respaldo y recuperación.

Acceso a través de lenguajes de programación estándar.

### **Sistema de Gestión de Base de Datos (SGBD)**

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz

entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

### **Control sobre la redundancia de datos**

Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

### **Consistencia de datos**

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

### **Compartición de datos**

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la

base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

### **Mantenimiento de estándares**

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

### **Mejora la integridad de datos**

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

### **Mejora la seguridad**

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

### **Mejora en la accesibilidad de los datos**

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

### **Mejora la productividad**

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

### **Mejora en el mantenimiento**

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

### **Aumento de concurrencia**

En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

### **Mejora en los servicios de copias de seguridad**

Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

## **Desventajas de las bases de datos**

### **Complejidad**

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

### **Coste del equipamiento adicional**

Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

### **Vulnerable a fallos**

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

### **Tipos de campos**

Cada Sistema de Base de Datos posee tipos de campos que pueden ser similares o diferentes. Entre los más comunes podemos nombrar:

**Numérico:** entre los diferentes tipos de campos numéricos podemos encontrar enteros “sin decimales” y reales “decimales”.

**Booleanos:** Poseen dos estados: Verdadero “Si” y Falso “No”.

**Memos:** Son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados.

**Fechas:** Almacenan fechas facilitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.

**Alfanuméricos:** Contienen cifras y letras. Presentan una longitud limitada (255 caracteres).

**Autoincrementables:** Son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta: Servir de identificador ya que resultan exclusivos de un registro.

### **Tipos de bases de datos**

Entre los diferentes tipos de base de datos, podemos encontrar los siguientes:

**MySql:** Es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.

**PostgreSql y Oracle:** Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.

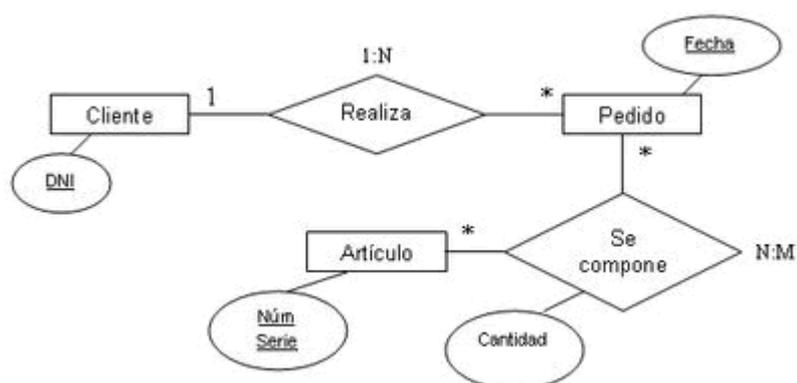
**Access:** Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa access, el cual crea un archivo .mdb con la estructura ya explicada.

**Microsoft SQL Server:** es una base de datos más potente que Access, desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

### Modelo Entidad - Relación

Los diagramas o modelos entidad-relación (denominado por su siglas, ERD “Diagram Entity Relationship”) son una herramienta para el modelado de datos de un Sistema de Información. Estos modelos expresan entidades relevantes para un Sistema de Información, sus inter-relaciones y propiedades.

Fig. N° 2 Modelo Entidad - Relación



### Cardinalidad de relaciones

El diseño de relaciones entre las tablas de una base de datos puede ser la siguiente:

**Relaciones de uno a uno:** una instancia de la entidad A se relaciona con una y solamente una de la entidad B.

**Relaciones de uno a muchos:** cada instancia de la entidad A se relaciona con varias instancias de la entidad B.

**Relaciones de muchos a muchos:** cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B.

### Estructura de una base de datos

Una base de datos, a fin de ordenar la información de manera lógica, posee un orden que debe ser cumplido para acceder a la información de manera coherente. Cada base de datos contiene una o más tablas, que cumplen la función de contener los campos.

Por consiguiente una base de datos posee el siguiente orden jerárquico:

Tablas

Campos

Registros

Lenguaje SQL

El lenguaje SQL es el más universal en los plataformas de base de datos. Este lenguaje nos permite realizar consultas a nuestras bases de datos para mostrar, insertar, actualizar y borrar datos.

### **ARQUITECTURA CLIENTE – SERVIDOR**

Es importante Indicar que hace más de una década, a la programación tradicional en un entorno compartido basado en el uso de un Servidor, se le denominaba arquitectura centralizada. Dentro de ésta, el manejador de bases de datos (DBMS) y los datos físicos residían en un computador central junto con el programa de aplicación que acepta entradas desde terminales de usuario y muestra los datos en la pantalla del usuario. En este caso el servidor cumplía dos funciones:

Servidor de datos.

Servidor de aplicaciones.

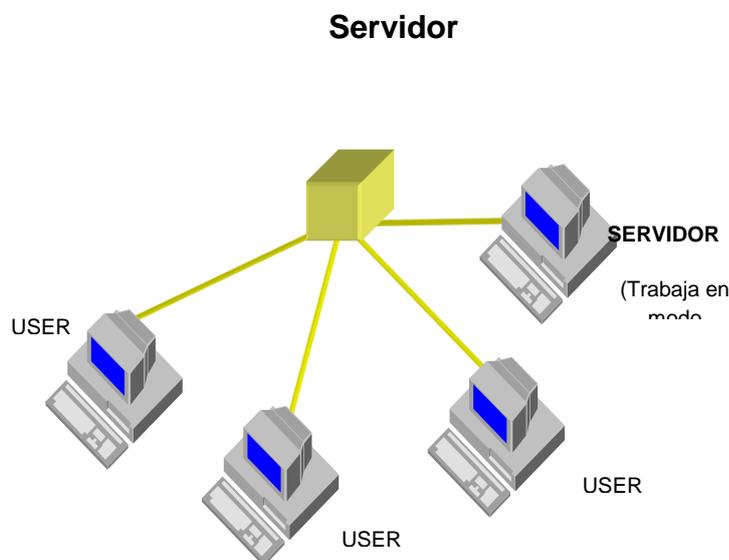
Ésta arquitectura centralizada hacia que los usuarios utilicen el plataforma en modo compartido, lo cual ocasionaba la degradación del rendimiento del plataforma debido a una sobrecarga del uso del plataforma.<sup>8</sup>

La velocidad de acceso y respuesta del servidor, dependía de sus recursos, para entonces esperar unos minutos no eran tan significativos como ahora.

---

<sup>8</sup> BUSTAMANTE, C. “Desarrollo de Plataformas Cliente Servidor”. Pág. 24. Lima. 2001

Fig. Nº 3 Arquitectura de las primeras plataformas Cliente-



En la actualidad la arquitectura Cliente-Servidor, hace que las computadoras personales integradas en una red de área local puedan interactuar con un servidor de base de datos, en la cual almacena las bases de datos que se desean compartir. Las funciones del DBMS están divididas en dos partes:

**a) Los frontales de base de datos (front - end)**, tales como herramientas de consulta interactiva, entornos de pequeñas aplicaciones, los cuales se ejecutan en un computador personal.

**b) La máquina de soporte (back-end)**, actúa sobre la bases de datos que almacena y gestiona los datos. En el servidor se ejecutan instrucciones S.Q.L.

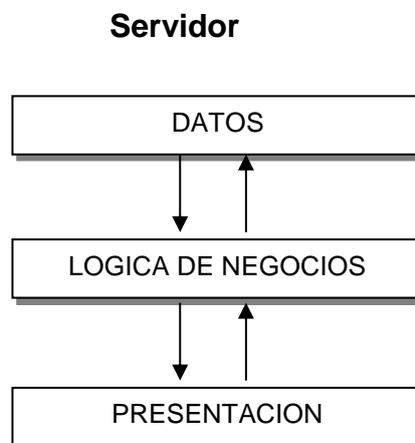
El S.Q.L. se ha convertido en el lenguaje de base de datos estándar para establecer comunicación entre herramientas frontales y la máquina de soporte en esta arquitectura.

Una computadora preferentemente bien equipada puede operar como servidor con mejor rendimiento, que las máquinas con procesadores de poca velocidad. El entorno gráfico basado en Windows, así como la programación basada en eventos ahora permite implementar soluciones que hagan uso de información y multimedia. Además de ello los clientes ahora dejan de ser tontos, ya que como cualquier computadora, poseen su propio cerebro, es decir trabajaban con su propio programa por lo que el servidor solamente gestiona datos.

### Arquitectura del software

Veamos de qué manera se despliega en software en el modelo Cliente-Servidor:

Fig. Nº 4 Despliegue de software en el modelo Cliente -



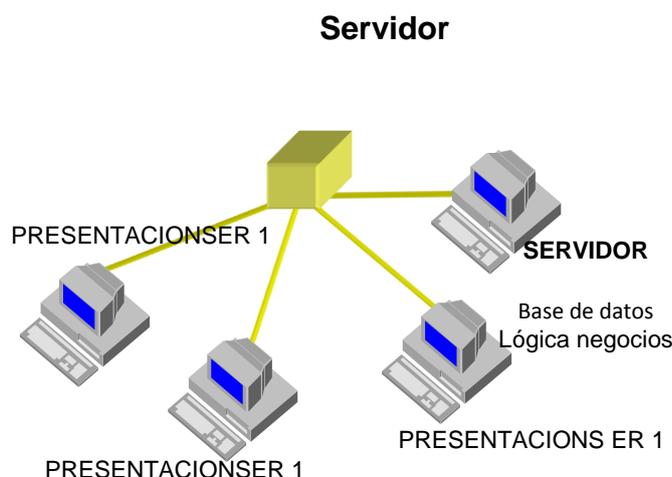
Dónde:

**Presentación:** Corresponde a la serie de ventanas que utiliza el usuario para acceder a la aplicación.

**Datos:** Representan la información que se encuentra en el servidor de base de datos.

**Lógica de negocios:** Son todas las reglas propias de una organización o de un sistema de organización determinado.

**Fig. Nº 5 Arquitectura actual de las plataformas Cliente –**



### 2.2.15 METODOLOGÍA DE PROGRAMACIÓN

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. 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.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP en sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación presentaremos las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas.

### **a) LAS HISTORIAS DE USUARIO**

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el plataforma debe poseer, sean requisitos funcionales o no funcionales.

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

Respecto de la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción o sólo una descripción, más quizás una estimación de esfuerzo en días. Beck en su libro presenta un ejemplo de ficha (customerstory and taskcard) en la cual pueden reconocerse

los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado cosas por terminar y comentarios.

Una de las interrogantes (que también se presenta cuando se utilizan casos de uso) es ¿cuál es el nivel de granularidad adecuado para una historia de usuario? La respuesta no es tajante. Jeffries dice que depende de la complejidad del plataforma, debe haber al menos una historia por cada característica importante, y propone realizar una o dos historias por programador por mes. Si se tienen menos, probablemente sea conveniente dividir las historias, si se tienen más lo mejor es disminuir el detalle y agruparlas. Para efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración).

No hay que preocuparse si en un principio no se identifican todas las historias de usuario. Al comienzo de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración.

Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

## b) ROLES XP

Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado describiremos los roles de acuerdo con la propuesta original de Beck.

**Programador:** El programador escribe las pruebas unitarias y produce el código de la plataforma.

Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

**Cliente:** El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el plataforma.

**Encargado de pruebas (Tester):** El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

**Encargado de seguimiento (Tracker):** El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es

verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.

También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

**Entrenador (Coach):** Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

**Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

**Gestor (Big boss):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

### c) PROCESO XP

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.
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 plataforma tenga el mayor valor de negocio posible con cada iteración.

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

#### **2.2.16 WORLD WIDE WEB O WWW**

En Informática, la **World Wide Web (WWW)** o **Red informática mundial**<sup>9</sup> es una plataforma de distribución de información basada en hipertexto o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden

---

<sup>9</sup><http://www.fundeu.es/vademecum-W-www-928.html>

contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

La Web fue creada alrededor de 1989 por el inglés Tim Berners-Lee con la ayuda del belga Robert Cailliau mientras trabajaban en el CERN en Ginebra, Suiza, y publicada en 1992. Desde entonces, Berners-Lee ha jugado un papel activo guiando el desarrollo de estándares Web (como los lenguajes de marcado con los que se crean las páginas web), y en los últimos años ha abogado por su visión de una Web semántica.

### **2.2.17 MYSQL**

Es un plataforma de gestión de bases de datos relacionales (SGBD) multiusuario, multiplataforma y de código abierto. Fue originalmente desarrollado por compañía sueca MySQLAB.

Este software es muy popular en aplicaciones web o distribuidas y es también componente de las plataformas LAMP, MAMP, WAMP, entre otras ya que como herramienta multiusuario brinda soluciones de base de datos SQL (StructuredQueryLanguage) multi-threaded. Es rápido, robusto y fácil de utilizar. Dentro de sus características principales podemos destacar los siguientes<sup>10</sup>:

Aprovecha la potencia de plataformas multiprocesador, gracias a su implementación multihilo.

---

<sup>10</sup> MySQL [www.mastermagazine.info/mysql3.htm](http://www.mastermagazine.info/mysql3.htm)

Soporta gran cantidad de tipos de datos para las columnas.

Dispone de API's en gran cantidad para: C, C++, Java, PHP, etc.

Gran portabilidad entre plataformas.

Soporta hasta 32 índices por tabla.

Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

### **2.2.18 HTML**

HTML es el acrónimo inglés de HyperTextMarkupLanguage, que se traduce al español como Lenguaje de Etiquetas de Hipertexto. Es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores como Internet Explorer, Opera, Firefox, Netscape o Safari, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para web.<sup>11</sup>

En realidad HTML no es un lenguaje de programación si no que son sentencias -etiquetas- de las cuales me indican que operaciones se van a realizar con el texto o con los atributos que se estén manejando con una sentencia-etiqueta, la verdad estas secuencias son muy necesarias para el diseño de una página

---

<sup>11</sup><http://darkub.wordpress.com/2008/03/03/%C2%BFque-significa-html-y-para-que-sirve/>

web ya que cada una de esas sentencias le indican a internet como está compuesta la estructura de cualquier Web

### 2.2.19 PHP

PHP, también denominado PHP HipertextPreprocessor, es un lenguaje de programación interpretado de alto nivel para internet, muy similar en su sintaxis al lenguaje C, con algunas diferencias, no compila como al igual que C, ya que es un intérprete, por lo tanto cada vez que se debe ejecutar un programa, lo interpreta verificando toda su sintaxis<sup>12</sup>.

El objetivo de PHP es brindarles a los creadores de sitios webs la posibilidad de desarrollar sitios dinámicos en forma sencilla y rápida, aunque en verdad veremos que las posibilidades y funcionalidades de PHP son muy superiores al simple hecho de solo hacer una página web dinámica.

Todos sabemos que en internet ya no basta con tener un sitio Estático porque estamos limitados para introducir cambios en su contenido en tiempo real, es decir, que cada vez que queremos introducir modificaciones debemos crear la página web para hacerlo, y luego publicarla (subiendo la página por medio de un FTP). En un sitio Dinámico, esta operatoria cambia radicalmente, ya que la información del sitio, generalmente está contenida en una base de datos. Cada vez que mostramos la página, como por ejemplo una página de noticias, buscamos en la base de datos las últimas noticias que tenemos ingresadas

---

<sup>12</sup> <http://www.blogdelaweb.com/%C2%BFque-es-php%C2%BFpara-que-sirve/>

para mostrar en el navegador del visitante. Ahora bien, si queremos que la página muestre noticias nuevas, simplemente cargamos lo mismo en la base de datos, por ejemplo, a través de un formulario y nuestra página cambiará automáticamente.

### 2.2.20 AJAX

Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). 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, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones<sup>13</sup>.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) 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.

---

<sup>13</sup> <http://es.wikipedia.org/wiki/AJAX>

Ajax es una técnica válida para múltiples plataformas y utilizable en muchas plataformas operativas y navegadores dados que está basado en estándares abiertos como JavaScript y DocumentObjectModel (DOM).

Ajax sirve para diseñar y programar interfaces de usuario superando las limitaciones que la sincronía supone y abre una nueva puerta que nos permite desarrollar aplicaciones web.

Permite acceder a información de manera remota pero siempre con una interface similar a la que podríamos usar en nuestro escritorio y un numeroso grupo de utilidades que revalorizarán el producto convirtiendo cualquier desarrollo en algo más que una página web.

### **2.2.21 JQUERY**

jQuery es un frameworkJavaScript, pero quizás muchos de los lectores se preguntarán qué es un framework. Pues es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, framework son unas librerías de código que contienen procesos o rutinas ya listos para usar. Los programadores utilizan los frameworks para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar.<sup>14</sup>

---

<sup>14</sup> <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>

jQuery es un framework para el lenguaje Javascript, luego será un producto que nos simplificará la vida para programar en este lenguaje. Como probablemente sabremos, cuando un desarrollador tiene que utilizar Javascript, generalmente tiene que preocuparse por hacer scripts compatibles con varios navegadores y para ello tiene que incorporar mucho código que lo único que hace es detectar el browser del usuario, para hacer una u otra cosa dependiendo de si es Internet Explorer, Firefox, Opera, etc. jQuery es donde más nos puede ayudar, puesto que implementa una serie de clases (de programación orientada a objetos) que nos permiten programar sin preocuparnos del navegador con el que nos está visitando el usuario, ya que funcionan de exacta forma en todas las plataformas más habituales.

Así pues, este framework Javascript, nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con jQuery obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Cuando programemos Javascript con jQuery tendremos a nuestra disposición una interfaz para programación que nos permitirá hacer cosas con el navegador que estemos seguros que funcionarán para todos nuestros visitantes. Simplemente debemos conocer las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la consecución de nuestros objetivos.

### 2.2.22 JAVASCRIPT

JavaScript es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web.<sup>15</sup>

Para entender lo que es JavaScript consideremos lo siguiente. Un usuario escribe una dirección web en su navegador, por ejemplo <http://www.aprenderaprogramar.com>. El servidor recibe la petición y como respuesta a esa petición envía al ordenador del usuario código HTML junto a código JavaScript. El código HTML se encarga de que en la pantalla se muestre algo, por ejemplo una imagen, un menú, etc. El código JavaScript se puede encargar de crear efectos dinámicos en respuesta a acciones del usuario, por ejemplo que se despliegue un menú tipo acordeón cuando el usuario pasa el ratón por encima de un elemento del menú.

La ventaja de JavaScript es que al estar alojado en el ordenador del usuario los efectos son muy rápidos y dinámicos. Al ser un lenguaje de programación permite toda la potencia de la programación como uso de variables, condicionales, bucles, etc. También podemos citar algún inconveniente: por

---

<sup>15</sup> [http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=590:ique-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192](http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=590:ique-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192)

ejemplo si el usuario tiene desactivado JavaScript en su navegador, no se mostrarán los efectos. No obstante, hoy día la mayoría de los usuarios navegan por la web con JavaScript activado.

### **2.2.23 PHPMYADMIN**

PhpMyAdmin es un software de código abierto, diseñado para manejar la administración y gestión de bases de datos MySQL a través de una interfaz gráfica de usuario. Escrito en PHP, PhpMyAdmin se ha convertido en una de las más populares herramientas basadas en web de gestión de MySQL. PhpMyAdmin viene con una documentación detallada y está siendo apoyado por un gran multi-idioma de la comunidad. PhpMyAdmin es cada vez mayor lista de características soporta todas las operaciones de uso común tales como la navegación, pasando, crear, modificar las bases de datos MySQL, las tablas, campos e índices. Además, PhpMyAdmin le permite administrar usuarios MySQL y privilegios de usuario. Otra característica común es PhpMyAdmin su función de importación. Con PhpMyAdmin, importar base de datos MySQL de copia de seguridad es fácil y se puede importar un volcado SQL o CSV con unos clics del ratón. También, usted puede exportar su base de datos en formato CSV, SQL,XML, Excel y otros

### **2.2.24 FRAMEWORK**

La palabra inglesa "framework" (infraestructura, armazón, marco) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como

referencia, para enfrentar y resolver nuevos problemas de índole similar<sup>16</sup>. En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

#### **2.2.25 BOOTSTRAP**

Twitter Bootstrap es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.<sup>17</sup>

Es el proyecto más popular en GitHub<sup>1</sup> y es usado por la NASA y la MSNBC junto a demás organizaciones

---

<sup>16</sup> Riehle, Dirk (2000), Framework Design: A Role Modeling Approach, Swiss Federal Institute of Technology

<sup>17</sup> <http://getbootstrap.com/getting-started/>

### 2.2.26 CODEIGNITER

CodeIgniter es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas.<sup>18</sup>

## 2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS

**Plataforma:** es el conjunto de elementos relacionados entre sí, en función de un objetivo común, actuando en determinado entorno y con capacidad de autocontrol. Se define también como un conjunto de componentes que interaccionan entre sí para un cierto objetivo. Es una colección de elementos o medios que están relacionados y que pueden ser descritos en términos de sus partes componentes.

**Director de Proyecto:** Es el docente que coadyuva a la elaboración del proyecto de tesis quien además para poder interactuar como tal deberá estar registrado en la plataforma y ser un docente ordinario de la Universidad Nacional del Altiplano.

**Jurado Dictaminador:** Son docentes de la Universidad Nacional del Altiplano contratados o nombrados que laboren actualmente y además estén

---

<sup>18</sup> <http://www.codeigniter.com/docs>

registrados en la plataforma de investigación, el jurado dictaminador es seleccionado de acuerdo a la línea de investigación.

**Sorteo Aleatorio:** Es el procedimiento que realiza la plataforma implementada utilizando un algoritmo que nos permita aplicar las respectivas restricciones.

**Línea de Investigación:** Las líneas de investigación están definidas por cada una de las escuelas profesionales de la Universidad Nacional del Altiplano, las cuales los docentes registrados en plataforma seleccionaron de acuerdo a su área de especialidad.

## CAPÍTULO III

### MATERIALES Y MÉTODOS

#### 3.1. LOCALIZACIÓN

El desarrollo del presente trabajo se llevó a cabo en la Universidad Nacional del Altiplano Puno en el Vicerrectorado de Investigación.

#### 3.2. POBLACIÓN Y MUESTRA

##### POBLACIÓN

Nuestra población estará integrada por todos los usuarios de la Plataforma de Investigación, quienes comprenden los estudiantes o egresados, docentes, y coordinadores de investigación.

En el momento de la aplicación de la ficha de evaluación del ISO 9126 y de la encuesta de satisfacción de usuarios 120 proyectos registrados, y consideramos usuarios a todos los que hacen uso del sistema por cada

proyecto se tiene al tesista, director de proyecto y 3 jurados dictaminadores, lo que hace una población de 600 usuarios.

**MUESTRA**

Se realizará un muestreo aleatorio simple entre todos los usuarios actuales en la Plataforma de Investigación. Para lo cual aplicamos la fórmula de (MARTINEZ BENCARDINO, 2012).

$$n = \frac{Z^2 PQN}{Z^2 PQ + Ne^2}$$

Tamaño Poblacional:	<b>N =</b>	<b>600</b>
Error (probabilidad):	<b>e =</b>	<b>0.03</b>
Ingrese la proporción (probabilidad):	<b>p =</b>	<b>0.89</b>
	<b>adicional</b>	<b>5</b>

Nivel de confianza (%)	muestra proyecto	Razón muestreo	muestra campo
<b>90</b>	<b>197</b>	<b>32.9</b>	<b>207</b>
91	206	34.3	216
92	214	35.7	225
93	224	37.3	235
94	234	39.1	246
<b>95</b>	<b>246</b>	<b>41.1</b>	<b>259</b>
96	260	43.3	273
97	276	46.1	290
98	297	49.5	312
<b>99</b>	<b>328</b>	<b>54.6</b>	<b>&lt;</b>

### 3.3. OPERACIONALIZACIÓN DE VARIABLES

VARIABLE	DIMENSION	INDICADORES	ESCALA
Eficacia de la Plataforma de Investigación	Utilidad de la Plataforma (ISO)	1. Mejoría en el flujo de trabajo. 2. Reducción del Tiempo de Procesos. 3. Calidad de las Interfaces. 4. Reducción de errores. 5. Disponibilidad de la Plataforma. 6. Consistencia de Correcciones. 7. Presenta la información necesaria. 8. Optimización de Recurso. 9. Prevención de Errores. 10. Velocidad de la Plataforma.	<ul style="list-style-type: none"> <li>• Inaceptable</li> <li>• Mínimamente aceptable</li> <li>• Aceptable</li> <li>• Cumple los requisitos</li> <li>• Excede los requisitos</li> </ul>
	Satisfacción de Usuario (ENCUESTA)	1. Navegación de la Plataforma 2. Acceso al la Plataform 3. Seguridad de la Plataforma 4. Claridad de los mensaje 5. Precisión de la información 6. Consistencia en los datos de ingreso 7. Calidad del Soporte 8. Pantallas de ayuda 9. Calidad de los manuales 10. Tiempo de Respuesta de la plataforma.	<ul style="list-style-type: none"> <li>- Deficiente</li> <li>- Mala</li> <li>- Regular</li> <li>- Bueno</li> <li>- Muy Bueno</li> </ul>

**Cuadro N° 01: Operacionalización de variables**

### **3.4. RECOPIACIÓN DE LA INFORMACIÓN**

Para la recolección de información de requerimientos de la plataforma se realizó mediante entrevistas y encuestas a los usuarios así como los estudiantes, docentes y docentes que hacen uso de la plataforma de investigación.

Para la recopilación de datos de usabilidad de la plataforma se utilizó los instrumentos que fueron aplicados en los usuarios de la Plataforma de Investigación del Vicerrectorado de Investigación en la Universidad Nacional del Altiplano Puno, como:

- ✓ La ficha de evaluación del ISO-9126 que ve la calidad del software.
- ✓ Encuesta de Satisfacción del Usuario.

### **3.5. MODELO Y METODOLOGÍA DE DESARROLLO DE SOFTWARE**

#### **3.5.1. MODELO DE DESARROLLO**

Decidimos adaptarnos a un híbrido en el modelo de desarrollo optando por dos (2) 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 etapas. Otro motivo es porque nos generó menores costos y porque es uno de los más utilizados actualmente lo que nos da seguridad de estos modelos de desarrollo.

También porque 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 (Daily Scrum) 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

## DUEÑO DE PRODUCTO:

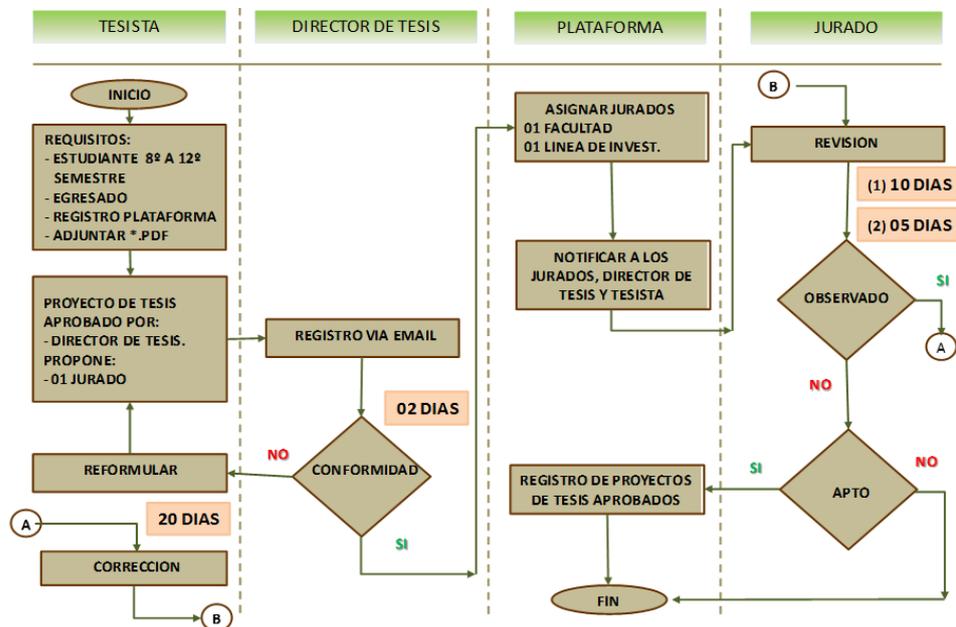
Tabla N° 1: Backlog

N°	USUARIO	TAREAS
1	El tesista deberá 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 tesista deberá registrar la información de su proyecto.	- Diseño de la base de datos e interfaz de registro de información de proyecto.
3	El director de proyecto debe aprobar el proyecto en la Plataforma.	- Diseño e implementación de los módulos de consulta de proyectos asignados.
4	Si el director aprueba el proyecto sortea Jurados.	Diseño e implementación de estados en la base de datos
5	La plataforma debe de sortear automáticamente los jurados de acuerdo a los criterios de sorteo: - Cateogria - Grado Académico - Antigüedad en la Categoría - Antigüedad en la Docencia	Diseño e implementación de los módulos de edición de datos y Sorteo de Jurados
6	Los jurados sorteaos deberán revisar el proyecto mediante la plataforma.	Se deben diseñar e implementar las validaciones
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

El programador mostró los módulos principales, los cuales contuvieron los procesos que permiten su funcionamiento, el detalle de cada uno de los procesos es el siguiente:

**Fig. Nº 6 Diagrama de Secuencia de Procedimientos de la Plataforma**



#### 3.7.1. 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 que este maneje.

R2: Módulo de validación de los registros deberá realizarse con el envío de un correo electrónico con un enlace de verificación de la cuenta.

R3: El plataforma debe estar en la capacidad de registrar nuevos tesis y docentes.

R4: El plataforma debe estar en la capacidad de poder editar datos como de los tesis y docentes si hubiese errores de registro.

R5: Por necesidad debe existir la posibilidad de consultar datos.

R6: Por necesidad hacer el seguimiento desde su inscripción hasta la aprobación del proyecto de un tesis.

R7: Módulo para el registro de líneas de investigación.

R8: Módulo para el registro de grados académicos.

R9: Módulo para la constancia de inscripción de proyecto.

R10: Módulo para la generación de Acta de aprobación del proyecto.

R11: Módulo para generar Reportes de los estados de la Plataforma.

#### **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.
- ✓ 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:

- 04 Computadora HP CORE i5 / 2.5Ghz.
- 4 Gb de memoria RAM DDR3.
- HD de 720 Gb.
- 01 Monitor LED 18.5
- 01 Impresora HP630.
- Disco duro externo 1Tb

Como mínimo para el funcionamiento de la plataforma se utilizaría una computadora Core i7.

#### ➤ **Software**

- Lenguaje de programación para Internet PHP.
- Gestor de Bases de Datos MySQL.
- Servidor Web, Apache.

- Sublime Text.
- Microsoft Office.
- Google Chrome.
- Rational Rose.
- Sistema Operativo Windows 8.1.
- Frameworks (Bootstrap, Codeigniter).

## CAPÍTULO IV

### RESULTADOS Y DISCUSIONES

#### 4.1. ANÁLISIS Y DISEÑO DE LA PLATAFORMA

##### 4.1.1. DESCRIPCIÓN DEL PLATAFORMA DE INSCRIPCIÓN INICIAL

El Vicerrectorado de Investigación tenía como premisa que el trámite de proyectos de tesis se llevara a cabo con el registro manual de los proyectos a nivel de todas las coordinaciones y Sub coordinaciones de Investigación, lo que ocasionaba en un momento dado aglomeraciones, y pérdida de tiempo por parte de los tesisistas, además de un descontrol total si no se maneja bien toda esta información.

##### A. ANALISIS DE VIABILIDAD

**Viabilidad Operativa:** El Vicerrectorado de Investigación se encuentra en la capacidad de administrar y controlar todos los proyectos de tesis, así como las coordinaciones de investigación está en las condiciones técnicas para poder coadyuvar a todos estos procedimientos aquí planteados.

**Viabilidad Técnica:** El Vicerrectorado de Investigación de la Universidad Nacional del Altiplano Puno, cuenta con equipos adecuados para la implementación del software, será necesario instalar AppServ 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 solo 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

**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 DOCENTES: Esta tabla almacena a todos los docentes de la plataforma.
- Tabla TESISTAS: Esta tabla almacena todos los tesis de la plataforma que ya cuentan con información validada.
- Tabla PROYECTOS: Esta tabla almacena la fecha de inscripción del proyecto.
- Tabla TRÁMITES: Esta tabla almacena el estado en el cual se encuentra el proyecto.
- Tabla DATOS\_TESISTA: Almacena la Información del Tesista y del trámite como son los Jurados sorteados y línea de Investigación del Proyecto.

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

## - 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 5.2 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 toda 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 del tesista, así como también se elaboró un manual de usuario para el registro de datos.

Esta versión se brindó una capacitación a los usuarios, los cuales dieron su visto bueno al sentirse conformes con los avances presentados.

### - 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 dio por parte de los tesistas.

**Iteración 3:** 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 4:** Para esta iteración se implementó las constancias y actas de aprobación de los proyectos con consultas a la base de datos de los estados así como los datos de los tesis.

**Iteración 5:** Para esta iteración se implementó los reportes de Líneas de Investigación así como de los usuarios de la Plataforma si como también de los estados de los mismos.

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 tesis o docentes 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 constancia de inscripción y su respectivo carné.

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**

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 el 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 estudiantes 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 Nº 2: Cuadro de decisiones ISO - 9126

Clasificación	Intervalo	Decisión
A) Inaceptable	[ 27 - 54 >	
B) Mínimamente aceptable	[ 54 – 81 >	
C) Aceptable	[ 81 – 95 >	
<b>D) Cumple los requisitos</b>	<b>[ 95 – 122 &gt;</b>	<b>102</b>
E) Excede los requisitos	[ 122 – 135 ]	

Fuente: Cuadro de decisiones ISO - 9126

Según los resultados el promedio de 103, 102, 108, 100, 98, 107 y 99 nos resulta 102, indicando que cumple con los requisitos según el ISO - 9126. Todo esto se muestra en el ANEXO II.

**DECISIÓN:** De acuerdo a los resultados de la calidad del software se concluyó que la Plataforma Web Basada en Cloud Computing para el seguimiento de Proyectos de Tesis de pregrado UNA Puno – 2016, cumple los requisitos con un promedio de 110 puntos del total de 135 puntos que se considera en el cuadro de decisiones del ISO - 9126.7.

#### 4.9. CUADRO COMPARATIVO SEGÚN EL TIEMPO

##### RESULTADOS UTILIZANDO LA PLATAFORMA Y SIN LA PLATAFORMA

Tabla N° 3. Cuadro comparativo expresado en minutos con y sin plataforma.

	CON PLATAFORMA	SIN PLATAFOR MA
Registro de Información	1 min.	3 horas.
Inscripción de Proyecto	2 min.	20 min.
Sorteo de Jurados	1 min.	1 hora.
Revisión por Jurados	3 min.	2 días.
Correcciones de Proyecto	1 min.	1 semana.
Dictamen de Proyecto	10 min	2horas
<b>TOTAL</b>	<b>18 min.</b>	<b>2semanas.</b>

Fuente propia: Cuadro comparativo expresado en minutos con y sin plataforma.

#### Decisión:

De acuerdo al cuadro comparativo expresado en minutos de una inscripción con plataforma y sin plataforma se aprecia que utilizando la plataforma de inscripción demora 9 min. Promedio por persona mientras que sin la plataforma se duplica el tiempo por persona.

4.10. GRAFICOS DE SATISFACCIÓN DE USUARIO

GRAFICO N° 1: ¿CÓMO CONSIDERA USTED LA NAVEGACIÓN EN PILAR?

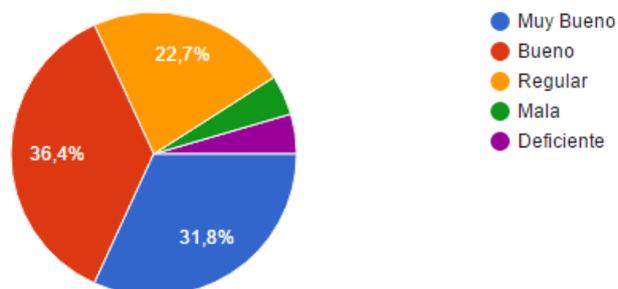


GRAFICO N° 2: ¿COMO CONSIDERA USTED EL ACCESO A PILAR (Login )?

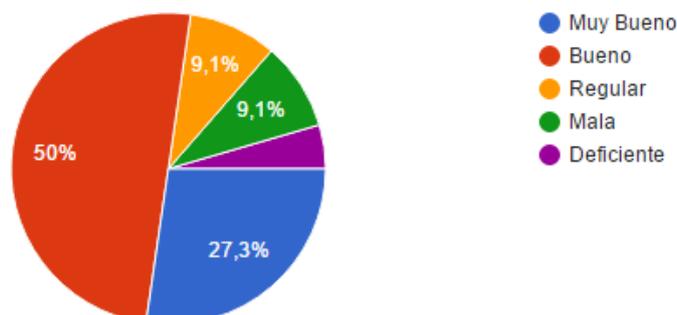
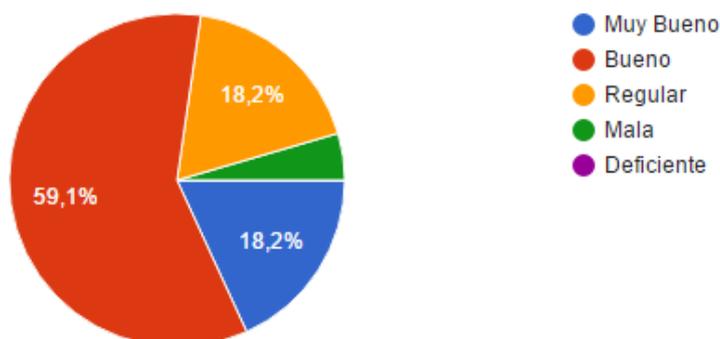
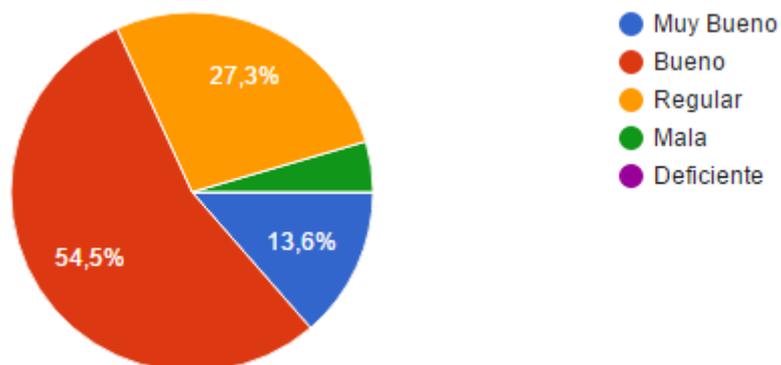


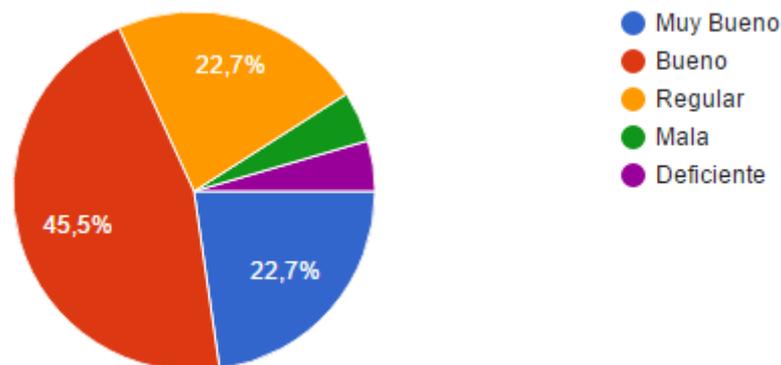
GRAFICO N° 3: ¿CÓMO CONSIDERA USTED LA SEGURIDAD DE PILAR?



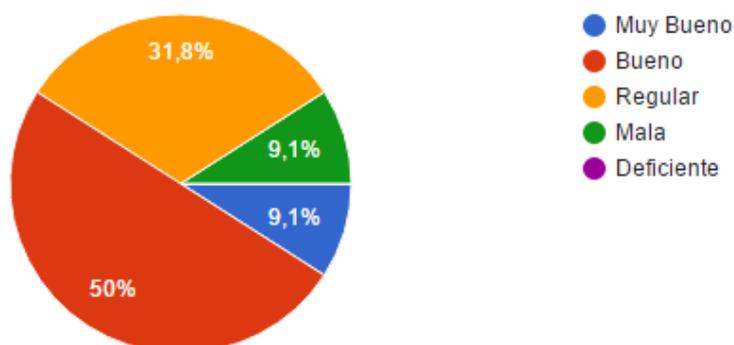
**GRAFICO N° 4: ¿PILAR MUESTRA INFORMACIÓN CON CLARIDAD?**



**GRAFICO N° 5: ¿CÓMO USTED CONSIDERA EL TIEMPO DE RESPUESTA DE PILAR?**



**GRAFICO Nº 6: ¿CÓMO CONSIDERA USTED LA CALIDAD DE SOPORTE  
(Apoyo técnico)?**



#### 4.11. CONTRASTE DE HIPÓTESIS

Los datos son provenientes de una población conformada por 600 usuarios y la muestra de estudio son 240 usuarios del sistema de plataforma web. El porcentaje poblacional para la implementación de la plataforma web es del 98% y el porcentaje muestral es de 95%.

##### **Hipótesis Nula (H0):**

La implementación de la Plataforma web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado no agilizará el proceso de inscripción, registro, sorteo, revisión, corrección y dictamen de los proyectos de tesis de pregrado en la Universidad Nacional del Altiplano 2016.

**Hipótesis Alternativa (Ha):**

La implementación de la Plataforma de web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado agilizará el proceso de inscripción, registró, sorteo, revisión, corrección y dictamen de los proyectos de tesis de pregrado en la Universidad Nacional del Altiplano 2016.

**i) Probabilidad de significancia.**

La probabilidad de significancia para esta prueba de hipótesis estadística es de  $\alpha = 0.05$ , y como la muestra de estudio es mayor a 30 datos, se utiliza la prueba estadística de la distribución normal estándar Z, en donde el valor tabulado para una prueba estadística es igual a 1.96

$$(Z\alpha = Z_{0.05} = 1.96)$$

**ii) Prueba estadística.**

La prueba estadística según el tamaño de muestra para este estudio es mayor a 30 datos, en tal caso, se utiliza la prueba de la distribución normal estándar Z.

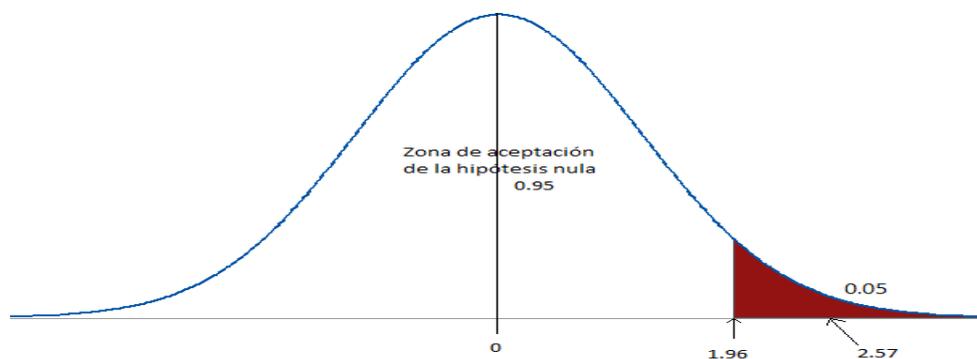
$$Z_c = \frac{P_0 - p}{\sqrt{\frac{P_0 Q_0}{n} \left( \frac{N-n}{N-1} \right)}} = \frac{0.98 - 0.95}{\sqrt{\frac{(0.98)(0.02)}{240} \left( \frac{600-240}{600-1} \right)}} = 2.57$$

**iii) Nivel de decisión.**

Como el valor de la distribución estadística normal Z calculado  $Z_c = 2.57$  es superior al valor de la distribución normal estándar tabulado  $Z\alpha = 1.96$ , entonces se acepta la hipótesis alternativa.

Tal como se puede observar en el gráfico correspondiente:

**Gráfico N° 7: Distribución Normal de Aceptación**



### **Conclusión.**

De acuerdo a los resultados obtenidos, se puede concluir que la implementación de la Plataforma de web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado agilizará el proceso de inscripción, registro, sorteo, revisión, corrección y dictamen de los proyectos de tesis de pregrado en la Universidad Nacional del Altiplano 2016.

## CONCLUSIONES

- Se desarrolló una plataforma ágil demostrando que haciendo uso de la plataforma se aminora el tiempo de inscripción, registro, sorteo, revisión, corrección y dictamen de los Proyectos de tesis de pregrado optimizando de esta manera todos los procedimientos.
- La eficiencia de la plataforma fue calificada por los usuarios quienes afirman que PILAR agilizó óptimamente los procedimientos. Como también la seguridad de la plataforma es la indicada para evitar cualquier vulnerabilidad de la información aquí consignada.
- Se ha planificados los tiempos para reducir los procesos redundantes y automatizarlos como la generación de memorandos, generación automática de constancias y envió de notificaciones por correo electrónico, una vez realizado el registros 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.

## RECOMENDACIONES Y SUGERENCIAS

- Se recomienda la implementación de programas de capacitación docente y de investigadores y/o tesis para una mejor adaptación de los actores del servicio de seguimiento del trámite del proyecto de tesis.
- Se recomienda la automatización de los procesos mecánicos de los trámites puesto que se convertirán en una forma habitual de comunicación y reducción de procesos.
- Se recomienda la integración con las plataformas de CONCYTEC como DINA, ALICIA, REGINA, para la comprobación de la información de los docentes y la comunicación con el repositorio nacional, así tener en tiempo real una tesis a penas termine de sustentarse, facilitando así el acceso a la información y la modernización de procesos.

**BIBLIOGRAFÍA**

ARTICULO N° 50 [ÓRGANO UNIVERSITARIO DE INVESTIGACIÓN] (9 de Julio de 2014). Ley Universitaria.[ Ley 30220 del 2014 ].

ANDRADE ,DAVI Y FALK J. A. (2001) *Eficacia de sistemas de informação e percepção de mudança organizacional: un estudo de caso. Revista de Administração contemporânea*, 5(3):53-84

BALESTRINI, M. (2006).*Cómo se elabora el Proyecto de investigación. Quinta Edición*. Caracas: Editorial Consultores Asociados.

BEN, L (2005).*Software libre, PHP y MySql .Tecnologías para el desarrollo de aplicaciones web*. España: Ediciones Díaz de Santos.

BOOCH, G. (1999). *El lenguaje Unificado de Modelado*. Primera Edición. Editorial Addison Wesley.

DE LA CRUZ V., Joel (2007). *Webs Dinámicas con CSS, DHTML y AJAX*. Lima – Perú: Grupo editorial Megabyte.

DA ROSA, FERNANDO Y HEINZ, FEDERICO (2007). *Guía Práctica sobre Software Libre, su selección y aplicación local en América Latina y el Caribe*.

KENDALL & KENDALL KENNETH Y JULIE. (2005). *Análisis y Diseño de Plataformas*. México: Tercera Edición. Editorial Prentice Hall.

LAUDON, KENNETH C.; LAUDON, JANE P (1996). *Administración de los Plataformas de Información*, México: 3ra. Edición-Editorial McGraw Hill.

LAUDON y LAUDON (2006). *Plataformas de Información Gerencial* (p.7)

MARTIN, M. Á. M. (2010) *Filosofía lean aplicada a la ingeniería de software.*

*Universidad de Sevilla*

NICOLAS, H. (2006). *Plataformas de Información Gerencial*

QUISPE, G A (2014). *Sistema informático de gestión administrativa para la coordinación de investigación de la Facultad de Ingeniería Estadística e Informática de la UNA – PUNO*

STALLMAN, R. (2004). *Software libre para una sociedad libre.* Edición, Traficantes de Sueños. Madrid. pp. 45-47.

SAMUELSON, P. Y NORDHAUS, W. (2002), *Economía.* 17 Edición, McGraw Hill, Madrid.

TAMAYO Y TAMAYO, M. (2003). *El Proceso de la Investigación Científica.* Editorial Limusa, Mexico.

VALDERRAMA M., Santiago (2006). *Pasos para elaborar proyectos y tesis de investigación científica.* Lima – Perú. Editorial san Marcos.

## REFERENCIAS DE INTERNET

*Definición de información* accesado el 11 de diciembre de 2011 y disponible en

<http://www.alegsa.com.ar/Definicion/de/informacion.php>

*Definición de Plataforma operativo* Recuperado el 15 de diciembre del 2011 y

disponible en <http://www.alegsa.com.ar/Dic/plataforma%20operativo.php>

Base de Datos accesado el 02 de enero del 2012 y disponible en

<http://es.kioskea.net/contents/bdd/bddintro.php3>

DAMIÁN, P. (2007). *Que son las bases de datos*. Disponible en

<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de->

[datos/](http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/) consultado el 18 de enero del 2012

Miguel, A. (2001). *¿Qué es PHP?* accesado el 01 de enero del 2012 y disponible

en <http://www.desarrolloweb.com/articulos/392.php>

*Definición de Javascript* accesado el 20 de enero de 2012 y disponible en

[http://www.librosweb.es/javascript/pdf/introduccion\\_javascript.pdf](http://www.librosweb.es/javascript/pdf/introduccion_javascript.pdf)

*Browser, explorador, navegador web* accesado el 19 de enero del 2012 y

disponible en <http://es.scribd.com/doc/33556783/Browser-Definicion-y-Tipos>

*Definición de Informática* accesado el 29 de diciembre del 2011 y disponible

en [http://www.ucla.edu.ve/dac/Departamentos/coordinaciones/informaticai/docu-](http://www.ucla.edu.ve/dac/Departamentos/coordinaciones/informaticai/documentos/Resumen%20tema2.pdf)

[mentos/Resumen%20tema2.pdf](http://www.ucla.edu.ve/dac/Departamentos/coordinaciones/informaticai/documentos/Resumen%20tema2.pdf)

*Métricas de calidad* accesado el 02 de febrero y disponible en

<http://ldc.usb.ve/~abianc/materias/ci4712/metricas.pdf>

*La norma ISO/IEC 9126* accesado el 05 de febrero de 2012 y disponible en

<http://iso25000.com/index.php/iso-iec-9126.html>

*Administración de la cobranza* accesado el 15 de febrero del 2012 de <http://www.buenastareas.com/ensayos/Administraci%C3%B3n-De-La-Cobranza/170004.html>

*Definición de Archivo* accesado el 16 de febrero de 2012 y disponible en <http://archivonorma.com/news/showByTitle/title/Archivosinformaticos/src/@random4d62d110f1dc3>

*Cuota mensual* accesado el 18 de febrero de 2012 y disponible en [http://orientacion.sunat.gob.pe/index.php?option=com\\_content&view=article&id=852:05-cuota-mensual&catid=65:fraccionamiento-y-aplazamiento&Itemid=110](http://orientacion.sunat.gob.pe/index.php?option=com_content&view=article&id=852:05-cuota-mensual&catid=65:fraccionamiento-y-aplazamiento&Itemid=110)

*Definición de Control* accesado el 17 de febrero de 2012 y disponible en <http://www.monografias.com/trabajos14/control/control.shtml#def>

*Cuenta de usuario* accesado el 19 de febrero y disponible en [de http://es.wikipedia.org/wiki/Cuenta\\_de\\_usuario](http://es.wikipedia.org/wiki/Cuenta_de_usuario)

*El control como fase del proceso administrativo* accesado el 20 de febrero de 2012 y disponible en <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger1/controladmin.htm>

*Interfaz de usuario* accesado el 17 de febrero de 2012 y disponible en [http://es.wikipedia.org/wiki/Interfaz\\_de\\_usuario](http://es.wikipedia.org/wiki/Interfaz_de_usuario)

# ANEXOS

**ANEXO – I  
FICHA DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO ESTÁNDAR  
ISO-9126**

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
<b>1. FUNCIONALIDAD</b>					
<b>Adecuación:</b> la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					
<b>Exactitud:</b> la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.					
<b>Interoperabilidad:</b> la capacidad del producto software para interactuar con uno o más plataformas especificados					
<b>Seguridad:</b> referido a la capacidad del producto software para proteger la información y los datos.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad.					
<b>2. FIABILIDAD</b>					
<b>Madurez:</b> la capacidad del producto software para evitar fallos provocados por errores en el software.					
<b>Tolerancia a fallos:</b> la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.					
<b>Recuperabilidad:</b> la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.					
<b>3. USABILIDAD</b>					
<b>Comprensibilidad:</b> 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.					
<b>Facilidad de aprendizaje:</b> la capacidad del producto software para permitir al usuario aprender su aplicación.					
<b>Atracción:</b> la capacidad del producto software para atraer al usuario.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.					
<b>Operabilidad:</b> la capacidad del producto software para permitir que el usuario lo opere y lo controle.					
<b>4. EFICIENCIA</b>					
<b>Comportamiento temporal:</b> la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.					
<b>Utilización de recursos:</b> la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					
<b>5. MANTENIBILIDAD</b>					
<b>Analizabilidad:</b> Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.					
<b>Cambiabilidad:</b> Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.					
<b>Estabilidad:</b> Capacidad del producto software de evitar los efectos inesperados de las modificaciones.					
<b>Facilidad de prueba:</b> Capacidad del producto software de permitir validar las partes modificadas.					
<b>Conformidad:</b> Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.					
<b>INDICADORES</b>	<b>PUNTUACION</b>				
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>6. PORTABILIDAD</b>					
<b>Adaptabilidad:</b> 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.					
<b>Facilidad de instalación:</b> la capacidad del producto software para ser instalado en un ambiente determinado.					
<b>Coexistencia:</b> la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.					
<b>Reemplazabilidad:</b> la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad					
<b>SUB TOTALES</b>					
<b>TOTAL</b>					

Fuente: Ficha de evaluación ISO-9126

Tabla N° 3. Escala valorativa. (Escala de Likert)

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

Fuente: Escala valorativa. (Escala de Likert)

Tabla N° 4. 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.

**Escala valorativa. (Escala de Likert)**

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

Fuente: Escala valorativa ISO-9126

**Cuadro de decisiones ISO - 9126**

Clasificación	Intervalo	Decisión 1	Decisión 2	Decisión 3	Decisión 4	Decisión 5	Decisión 6	Decisión 7
A) Inaceptable	[ 27 - 54 >							
B) Mínimamente aceptable	[ 54 - 81 >							
C) Aceptable	[ 81 - 95 >							
D) Cumple los requisitos	[ 95 - 122 >	103	102	108	100	98	107	99
E) Excede los requisitos	[ 122 - 135 ]							

Fuente: Cuadro de decisiones ISO - 9126

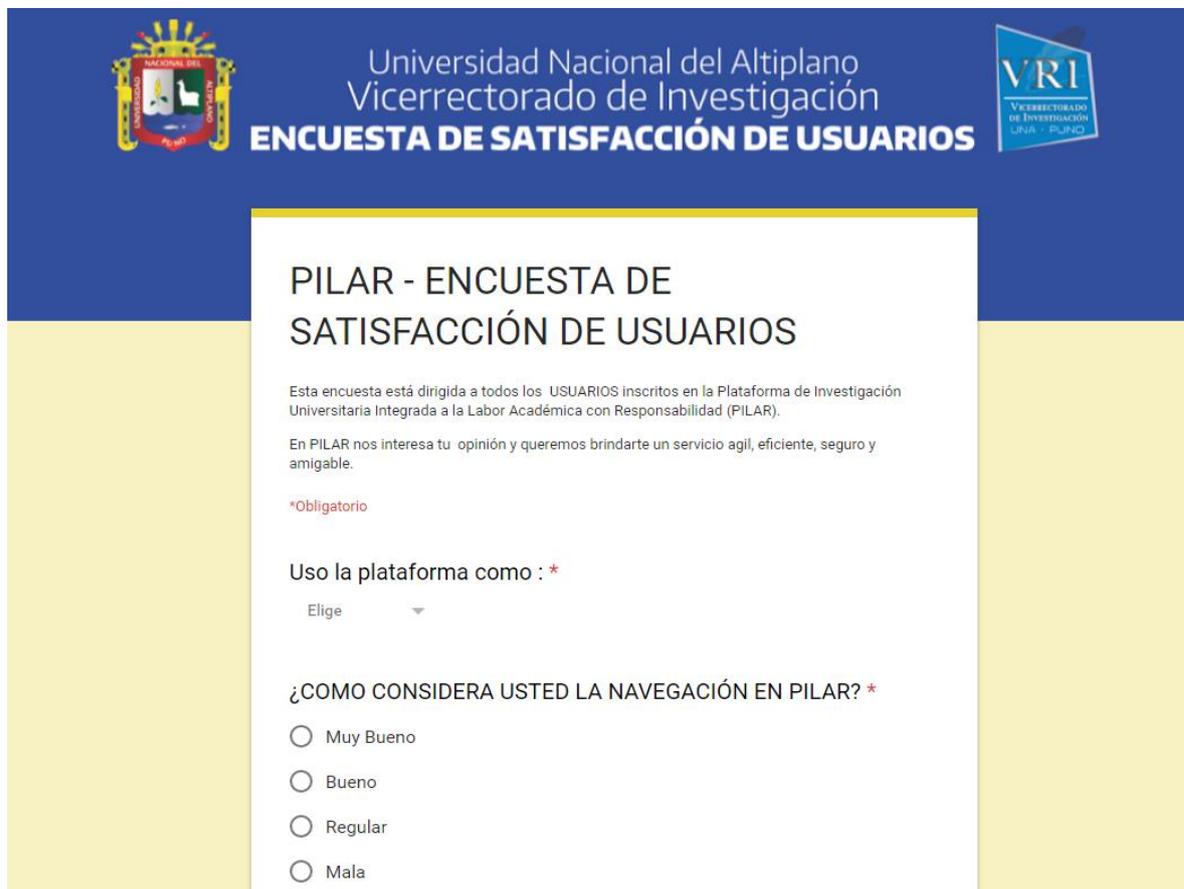
## ANEXO– II

## ENCUESTA DE SATISFACCIÓN DE USUARIO

Esta encuesta está dirigida a todos los USUARIOS inscritos en la Plataforma de Investigación Universitaria Integrada a la Labor Académica con Responsabilidad (PILAR).

En PILAR nos interesa tu opinión y queremos brindarte un servicio agil, eficiente, seguro y amigable. \*Obligatorio

1. USO LA PLATAFORMA COMO? : \*
  - Tesista
  - Docente
  - Coordinador
  
2. ¿COMO CONSIDERA USTED LA NAVEGACIÓN EN PILAR? \*
  - Muy Bueno
  - Bueno
  - Regular
  - Mala
  - Deficiente
  
3. ¿COMO CONSIDERA USTED EL ACCESO A PILAR ( Login )? \*
  - Muy Bueno
  - Bueno
  - Regular
  - Mala
  - Deficiente
  
4. ¿COMO CONSIDERA USTED LA SEGURIDAD DE PILAR? \*
  - Muy Bueno
  - Bueno
  - Regular
  - Mala
  - Deficiente
  
5. ¿PILAR MUESTRA INFORMACIÓN CON CLARIDAD? \*
  - Muy Bueno
  - Bueno
  - Regular
  - Mala
  - Deficiente
  
6. ¿COMO USTED CONSIDERA EL TIEMPO DE RESPUESTA DE PILAR? \*
  - Muy Bueno
  - Bueno
  - Regular
  - Mala
  - Deficiente
  
7. ¿COMO CONSIDERA USTED LA CALIDAD DE SOPORTE (Apoyo técnico)? \*
  - Muy Bueno
  - Bueno
  - Regular
  - Mala
  - Deficiente



 Universidad Nacional del Altiplano  
Vicerrectorado de Investigación  
**ENCUESTA DE SATISFACCIÓN DE USUARIOS** 

### PILAR - ENCUESTA DE SATISFACCIÓN DE USUARIOS

Esta encuesta está dirigida a todos los USUARIOS inscritos en la Plataforma de Investigación Universitaria Integrada a la Labor Académica con Responsabilidad (PILAR).

En PILAR nos interesa tu opinión y queremos brindarte un servicio ágil, eficiente, seguro y amigable.

**\*Obligatorio**

Uso la plataforma como : \*

Elige ▼

¿COMO CONSIDERA USTED LA NAVEGACIÓN EN PILAR? \*

Muy Bueno

Bueno

Regular

Mala

**Dirección web:** <https://goo.gl/forms/Tm6cyOOutGxqtGf33>

## ANEXO– III

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

---

## MANUAL DE USUARIO

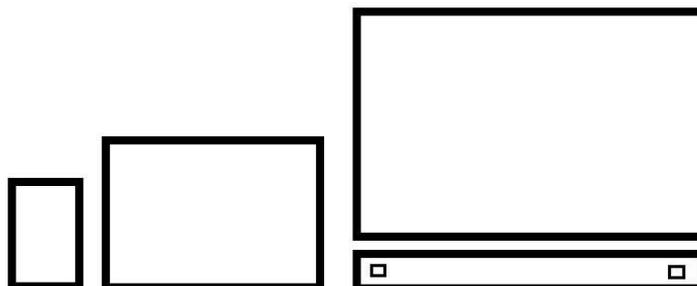
La Plataforma de Investigación Universitaria Integrada a la Labor Académica con Responsabilidad - PILAR, ha sido creada en el Vicerrectorado de Investigación de la Universidad Nacional del Altiplano Puno en el marco de la implementación de la ley universitaria 30220 y con la finalidad de optimizar el proceso de seguimiento y revisión de los proyectos de tesis, contribuyendo de este modo con los estudiantes de semestres finales y egresados que están iniciando el trámite para la obtención de sus títulos profesionales, facilitándoles un medio electrónico para la comunicación con los jurados de tesis, mediante una plataforma web que permite la calificación anónima del proyecto, tal y como se realiza un artículo científico.

La principal finalidad de PILAR es contribuir con tesis y docentes evitando innecesarios desplazamientos y reuniones que no suelen concretarse en su momento, la revisión, corrección se realiza cómodamente una vez leído el documento en formato PDF y debidamente mapeado para ubicar con rapidez las correcciones.

Reducir tiempo, burocracia y el gasto excesivo de papel, PILAR es EcoAmigable. Pues no requiere la impresión de ejemplares para cada jurado, sino que realiza la revisión mediante un navegador web.

### **Ventajas de Pilar**

- **Acceder a PILAR**  
Desde cualquier punto del planeta puede uno acceder gracias a la interconexión de Internet encuéntrenos siempre listos para ayudarlo en <http://vriunap.pe/pilar>
- **PILAR es Multidispositivo**  
PILAR tiene una interfaz que se adapta a cualquier dispositivo como Laptops, Tablets y Smartphones facilitando el acceso de docentes y tesis desde cualquier lugar y en el dispositivo que se sienta más cómodo o disponible en su momento.





### Resumiendo las Ventajas

- Es fácil e intuitivo de usar
- El proyecto de tesis sigue los lineamiento en líneas OCDE
- El jurado y el tesis se comprenden en la línea de investigación
- Las correcciones se realizan en los plazos establecidos.
- Al ser anónimo el tesis no será etiquetado ni victimizado por el jurado.
- Se comunica a los jurados por correo y mensajes de texto.
- Reduce el tiempo de trámite
- No hay ningún costo por la inscripción ni proceso.
- No hay gasto innecesario de papel para las revisiones.
- Los formatos son electrónicos.
- Se capacita a los docentes y tesis para el uso adecuado de PILAR.



## REGISTRO DE NUEVOS TESISISTAS

**Paso 1:** Ingresar a <http://vriunap.pe/pilar> Click en Tesistas.

**Paso 2:** digitar Código de Matricula y DNI.

**Paso 3:** Completar el **formulario** y grabar.

Recomendamos un correo que usa como investigador y profesional.  
[mario.barrios@yahoo.es](mailto:mario.barrios@yahoo.es)

[doloretc92@hotmail.com](mailto:doloretc92@hotmail.com)

No recomendado (Poco profesional)

[florcita3000x@gmail.com](mailto:florcita3000x@gmail.com)

[nodiox100pre@hotmail.com](mailto:nodiox100pre@hotmail.com)

Nro de Telefono Celular de 9 Digitos



**Registrar nuevo Tesista**

<b>FACULTAD</b>	<input type="text" value="Ingeniería de Alimentos y Nutrición"/>
<b>Escuela Profesional</b>	<input type="text" value="Ingeniería de Alimentos y Nutrición"/>
<b>Nombres Completos</b>	<input type="text" value="MARIO PASCAL ALVARO GARCIA"/>
<b>Último Semestre reg.</b>	<input type="text" value="2012-I (DECIMO)"/>
<b>Numero de Telef. Celular</b>	<input type="text" value="Ej. 999001122"/>
<b>Ingrese su Correo personal</b>	<input type="text" value="e-mail"/>
<b>Crear una contraseña</b>	<input type="password" value="*****"/> <small>debe contener numeros y letras</small>
<b>Repita su contraseña</b>	<input type="password" value="*****"/> <small>se repite para minimizar errores</small>

**Paso 4:** Se le envía un correo de Validación, una vez concluido puede ingresar siempre como tesista registrado.

**Acceso de Tesistas Registrados**

<b>Su correo</b>	<input type="text" value="ipalvar@una.edu.pe"/>
<b>Contraseña</b>	<input type="password" value="*****"/>

Si olvida su clave, acerquese a Plataforma



## REGISTRAR MI TESIS (en formato PDF)

**Paso 1:** Descargar el Formato, borre las etiquetas guía, mantenga el número de líneas, converse previamente con su Director de Tesis, debe ser un docente Nombrado, por la responsabilidad que conlleva su Proyecto de Investigación.



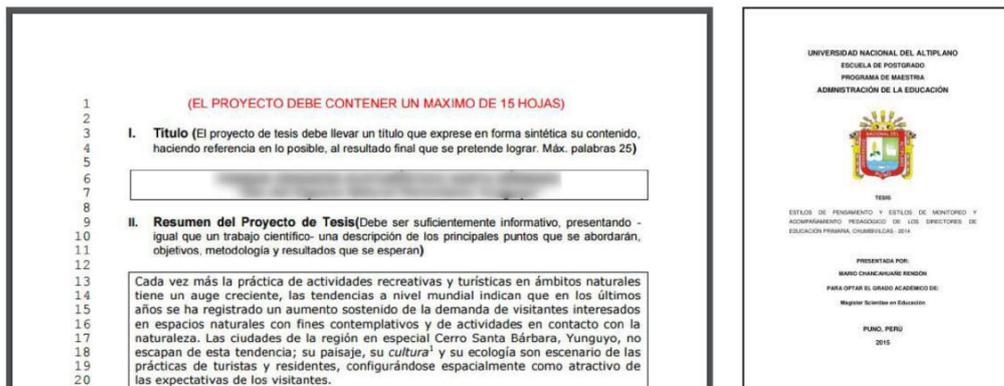
**Paso 2:** Revisar las líneas de Investigación y Docentes registrados en sus respectivas carreras mediante plataforma.

Preparado para enviar Proyecto de Tesis

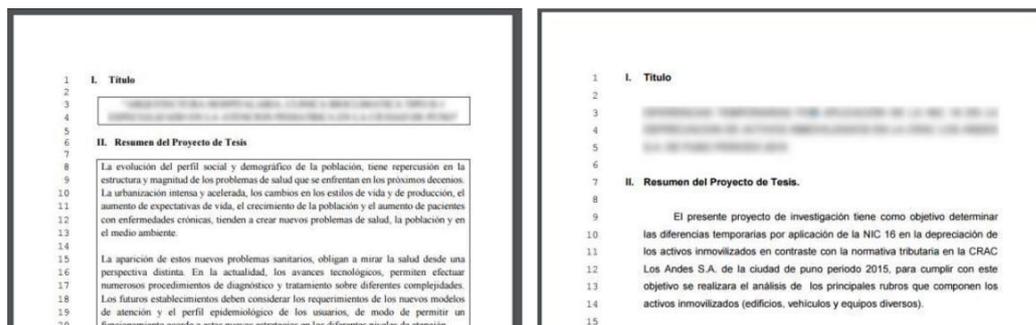
Escuela Profesional	INGENIERÍA ECONÓMICA
Codigo Universitario	00000
Nombres Completos	INGENIEROS ALUMNOS Y PROFESORES
Título de Proyecto (sin comillas)	<input type="text"/>
<b>Linea de Investigación</b>	<div style="border: 1px solid orange; padding: 5px;">                 Seleccione un elemento                  Seleccione un elemento                  ECONOMÍA DE LA EMPRESA Y MERCADOS - ( 34 )                  ECONOMÍA DE RECURSOS NATURALES Y DEL MEDIO AMBIENTE - ( 18 )                  ECONOMÍA REGIONAL Y LOCAL - ( 31 )                  POLÍTICAS PÚBLICAS Y SOCIALES - ( 42 )             </div>
Director de Tesis	
Jurado de sugerencia	Seleccione un elemento

**Paso 3:** Ejemplos de presentación de Proyectos

**Ejemplos Incorrectos:** debe borrarse las guías, el proyecto no debe contener caratula ni nombres, sin nombre de carrera, completamente anónimo. Se califica el Proyecto de Investigación, no al tesista.



**Ejemplo Aceptables:** Numero de líneas y estructura de acuerdo al formato.



**Paso 4:** Secuencia de estados del tesista.

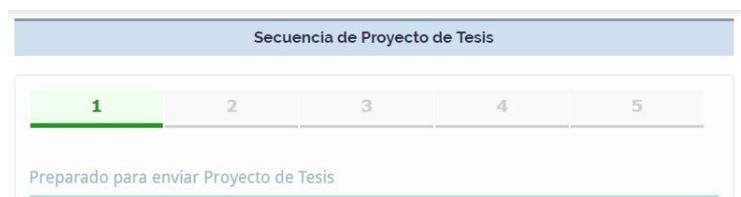
Preparado para subir proyecto.

Proyecto enviado a su Director.

En espera de Sorteo.

Proyecto Remitido a Los Jurados (subida de corrección y dictaminación).

Proyecto Aprobado.



**Paso 5:** Modelo de Acta de Aprobación, desde el día de aprobación debe contarse tres meses o 90 días.

UNIVERSIDAD NACIONAL DEL ALTIPLANO	VICERRECTORADO DE INVESTIGACIÓN	PLATAFORMA DE INVESTIGACIÓN UNIVERSITARIA ENFOCADA A LA LABOR ACADÉMICA CON RESPONSABILIDAD
2016-263		
<b>ACTA DE APROBACIÓN DE PROYECTO DE TESIS</b>		
<p>En la Ciudad Universitaria, a los 06 días del mes de JULIO del 2016 siendo horas 11:26:40 los miembros del Jurado, declaran APROBADO POR REGLAMENTO el PROYECTO DE INVESTIGACIÓN DE TESIS titulado:</p>		
<b>CINÉTICA DE LIXIVIACIÓN DE MINERAL DE MANGANESO EN SOLUCIÓN DE ÁCIDO SULFÚRICO UTILIZANDO AGENTE REDUCTOR MAZORCA DE MAÍZ Y TRIGO</b>		
Presentado por el(los) Bachiller:		
ROSALUZ ARGOTE PAREDES		
De la Escuela Profesional de:		
INGENIERÍA QUÍMICA		
Siendo el Jurado dictaminador, conformado por:		
Presidente	: MOSES FEREZ CAPA	
Primer Miembro	: M.Sc. JAQUETE ROSARIO RAMOS PINEDA	
Segundo Miembro	: Ing. MARLENI YOVANNA VALENCIA PACHO	
Director/Asesor	: GERMAN QUILLE CALZAYA	
<p>Para dar fe de este proceso electrónico, el Vicerrectorado de Investigación de la Universidad Nacional del Altiplano - Puno, mediante la Plataforma de Investigación se le asigna la presente constancia y a partir de la presente fecha queda expedito para la ejecución de su PROYECTO DE INVESTIGACIÓN DE TESIS.</p>		
Puno, JULIO de 2016		
<p>VIR UNA Puno - 2016 Código: 2016-263</p>	 DE FANCELLO MEDINA ESPINOZA VICERRECTOR DE INVESTIGACIÓN	
<p>Vicerrectorado de Investigación Teléfono 051-862264 www.una.edu.pe www.vicir.una.edu.pe</p>		

## ANEXO– IV

## CÓDIGO FUENTE

```

01 <?php
02 /*****
03 * Plataforma de Investigación seguimiento de Tesis
04 * Fecha de Producción : Febrero de 2016
05 * version : 2.0
06 *
07 *
08 *   Programador: Fred Torres Cruz
09 *   Architect: Ramiro Pedro Laura Murillo
10 *   Revisor: Nayer Tumi Figueroa
11 *
12 *****/
13
14 define('ENVIRONMENT', 'development');
15
16 if (defined('ENVIRONMENT'))
17 {
18     switch (ENVIRONMENT)
19     {
20         case 'development': error_reporting(E_ALL); break;
21         case 'testing':
22         case 'production': error_reporting(0); break;
23         default: exit('The application environment.');
```

- Implementación basada en Clases PHP con Objetos

```

01
02 <?PHP if ( ! defined('BASEPATH')) exit('No direct script access
03 allowed');
04
05 include("../absmain/mlLibrary.php");
06
07 class Docentes extends CI_Controller {
08
09     public function __construct() {
10         parent::__construct();
11
12         $this->lang->load('generales');
13         $this->load->model('ado');
14         $this->load->library("session");
15     }
16
17     //-----
18     public function login()
19     {
20         $user = mlSecurePost("user");
21         $pass = mlSecurePost("pass");
22         if( !$user or !$pass ) exit;
23         //-----
24         $pass = sqlPassword( $pass ); // SHA1
25         $temp = $this->ado->getSnapView('vwConsulDocentes');
26         if( $temp->validate() == true ) {
27             $row = $temp->row();
28             $this->session->set_userdata( array(
29                 'utype'     => "docentes",
30                 'userId'    => $row->IdDocente,
31             ) );
32         }
33
34         // en ambos casos van: internamente se valida
35         redirect( base_url($this->lang->lang()."/docentes") );
36     }
37     //-----
38     public function logout()
39     {
40         $this->session->sess_destroy();
41         redirect( base_url($this->lang->lang()."/docentes") );
42     }
43     //-----
44     function onAreWeLogged()
45     {
46         if( $this->isActiveSess() == false )
47             $this->logout();
48     }
49     //-----
50     function isActiveSess()
51     {
52         return ( $this->session->userdata('pilarSes') );

```

```
53     }
54
55
56
57     public function index()
58     {
59         $this->correctURL();
60
61         // activado desde e-amil
62         if( $this->docenteRegValida(mlSecureRequest("reg")) )
63             return;
64
65         //
66         // si estamos logeados
67         //
68         $this->load->view("webHead");
69         $this->load->view("webLogo", array('logoCarru'=>false) );
70
71         if( $this->isActiveSess() )
72         {
73             $this->load->view("doc/procRevis", $args );
74         }
75         else
76         {
77             $this->load->view("doc/inicio");
78         }
79
80         $this->load->view("webFoot");
81     }
82 }
```