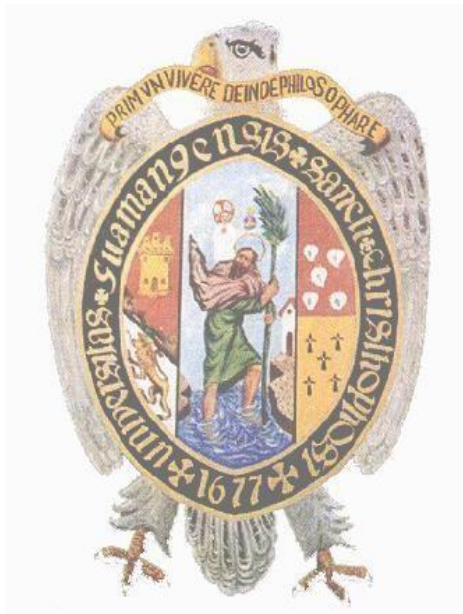


UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA  
FACULTAD DE INGENIERÍA DE MINAS GEOLOGÍA Y CIVIL  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“APLICACIÓN WEB PARA EL SISTEMA DE TUTORÍA DE LA  
UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA,  
AYACUCHO 2016”**

Tipo de investigación: aplicada Área de investigación: tecnologías web

EJECUTOR: Bach. Quispe Gómez, Ayde  
ASESOR: Mg. Ing. Manuel A. Lagos Barzola

AYACUCHO – PERÚ 2016

## **DEDICATORIA**

A mis padres por su apoyo incondicional que me brindaron para salir adelante, a mi hijo por iluminarme con la paz de su sonrisa, cambiarme la vida y ser el motor que me empuja día a día.

## **AGRADECIMIENTOS**

A la Universidad Nacional de San Cristóbal de Huamanga y a los docentes que guiaron mi aprendizaje para la vida profesional.

# ÍNDICE

<i>AGRADECIMIENTOS</i> .....	3
<i>RESUMEN</i> .....	4
<i>INTRODUCCIÓN</i> .....	9

## CAPITULO I

### PLANTEAMIENTO DE LA INVESTIGACIÓN

<i>1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA</i> .....	10
<i>1.2. DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN</i> .....	11
<i>1.2.1. PROBLEMA PRINCIPAL</i> .....	11
<i>1.2.2. PROBLEMAS SECUNDARIOS</i> .....	11
<i>1.3. OBJETIVOS DE LA INVESTIGACIÓN</i> .....	11
<i>1.3.1. OBJETIVO GENERAL</i> .....	11
<i>1.3.2. OBJETIVOS ESPECÍFICOS</i> .....	12
<i>1.4. JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN</i> .....	12
<i>1.4.1. IMPORTANCIA DEL TEMA</i> .....	12
<i>1.4.2. JUSTIFICACIÓN</i> .....	12
<i>1.4.3. DELIMITACIÓN DE LA INVESTIGACIÓN</i> .....	13
<i>DELIMITACIÓN TEMPORAL</i> .....	13
<i>DELIMITACIÓN TEÓRICA</i> .....	13

## CAPITULO II

### MARCO DE REFERENCIA DE LA INVESTIGACIÓN

<i>2.1. ANTECEDENTES DE LA INVESTIGACIÓN</i> .....	14
<i>2.2. MARCO TEÓRICO</i> .....	15
<i>2.2.1. APLICACIÓN WEB</i> .....	15
<i>2.2.2. CICLO DE VIDA DEL SOFTWARE</i> .....	15

2.2.3.	<i>INGENIERÍA DEL SOFTWARE</i> .....	17
2.2.4.	<i>PROGRAMACIÓN ORIENTADA A OBJETOS</i> .....	17
2.2.5.	<i>SISTEMA DE GESTION DE BASE DE DATOS</i> .....	18
2.2.6.	<i>PROGRAMACIÓN EXTREMA</i> .....	18
2.2.6.1.	<i>VALORES EN LA PROGRAMACIÓN EXTREMA</i> .....	19
2.2.6.2.	<i>CICLO DE VIDA IDEAL DEL XP</i> .....	20
2.2.6.3.	<i>HISTORIAS DE USUARIO</i> .....	23
2.2.6.4.	<i>TARJETAS C.R.C.</i> .....	24
2.2.6.5.	<i>PRUEBAS DE ACEPTACIÓN</i> .....	25
2.2.7.	<i>TUTORÍA UNIVERSITARIA</i> .....	26
2.2.8.	<i>SISTEMA TUTORIAL UNIVERSITARIO</i> .....	26

### CAPITULO III

#### METODOLOGÍA DE LA INVESTIGACIÓN

3.1	<i>TIPO DE INVESTIGACIÓN</i> .....	32
3.2	<i>NIVEL DE INVESTIGACIÓN</i> .....	32
3.3	<i>DISEÑO DE LA INVESTIGACIÓN</i> .....	32
3.4	<i>POBLACIÓN Y MUESTRA</i> .....	32
	<i>POBLACIÓN</i> .....	32
	<i>MUESTRA</i> .....	32
3.5	<i>VARIABLES E INDICADORES</i> .....	33
3.6	<i>TÉCNICAS E INSTRUMENTOS TÉCNICAS</i> .....	33
	<i>INSTRUMENTOS</i> .....	34
3.7	<i>HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN</i> 35	
3.8	<i>TÉCNICAS PARA APLICAR XP</i> .....	36

CAPÍTULO IV  
ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

4.1	<i>RESULTADOS DE LA APLICACIÓN DEL PROCESO AGIL XP43</i>	
4.1.1	<i>FASE DE EXPLORACION .....</i>	<i>43</i>
4.1.2	<i>ARQUITECTURA TÉCNICA INICIAL .....</i>	<i>48</i>
4.1.3	<i>PLAN DE ALTO NIVEL .....</i>	<i>48</i>
4.2	<i>FASE DE PLANEACIÓN .....</i>	<i>49</i>
4.2.1	<i>HISTORIA USUARIO POR PRIORIDAD .....</i>	<i>49</i>
4.2.2	<i>ESTIMACIÓN DE ESFUERZO .....</i>	<i>49</i>
4.2.3	<i>PLAN DE ENTREGA .....</i>	<i>51</i>
4.3	<i>FASE DE ITERACIÓN A LA PRIMERA VERSIÓN .....</i>	<i>53</i>
4.3.1	<i>ARQUITECTURA TECNICA .....</i>	<i>53</i>
	<i>DIAGRAMA DE COMPONENTES .....</i>	<i>53</i>
	<i>DIAGRAMA DE DESPLIEGUE .....</i>	<i>53</i>
4.3.2	<i>TAREAS DE INGENIERÍA .....</i>	<i>54</i>
4.3.3	<i>PLAN DE ITERACIÓN .....</i>	<i>81</i>
4.3.4	<i>CASOS DE PRUEBA DE ACEPTACIÓN .....</i>	<i>86</i>
4.3.5	<i>INTERFAZ GRÁFICA DE USUARIO (GUI) .....</i>	<i>87</i>
4.3.6	<i>TARJETA CRC .....</i>	<i>93</i>
	<i>BASE DE DATOS FÍSICA .....</i>	<i>99</i>
4.3.7	<i>CÓDIGO FUENTE .....</i>	<i>87</i>
4.3.8	<i>REPORTE DE PRUEBA UNITARIA .....</i>	<i>95</i>
4.3.9	<i>REPORTE DE PRUEBAS DE ACEPTACIÓN .....</i>	<i>96</i>

CAPITULO V  
CONCLUSIONES Y RECOMENDACIONES

*BIBLIOGRAFIA*.....102

## RESUMEN

En nuestra universidad, pese a la existencia del Reglamento del Sistema de Tutoría y a la Directiva de Lineamientos para la Implementación de dicho sistema, así como a las iniciativas de implementación de ciertas escuelas; la universidad en su conjunto carece de una herramienta tecnológica que de soporte a las actividades del sistema de tutoría tanto para los tutores como para tutorados y la parte administrativa encargada de la gestión del proceso de seguimiento y evaluación tutorial.

El objetivo fundamental que se persiguió en la investigación, fue desarrollar una aplicación web para el Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga y describir si la aplicación web brinda un soporte eficiente a sus actividades; Ayacucho 2016.

La investigación se desarrolló dentro de los ambientes de la Universidad Nacional de San Cristóbal de Huamanga.

La investigación desarrollada es de tipo aplicada, con un nivel descriptivo y diseño no experimental.

Las técnicas utilizadas para el cumplimiento de los objetivos fueron las entrevistas, encuestas y el análisis documental.

Se lograron los resultados del Análisis, Diseño, Implementación y Pruebas del Sistema Tutorial de la Universidad Nacional de San Cristóbal de Huamanga.

**Palabras clave:** Aplicación web, Sistema de tutoría, Tutorado, Tutor.



## INTRODUCCIÓN

La Programación Extrema (XP, eXtreme Programming) es un modelo de proceso de software que toma los principios y prácticas aceptadas, y las lleva a niveles extremos. Tiene como objetivo reducir el riesgo en el ciclo de vida del software mediante grupos de desarrollo pequeños. Considera que la mejor manera de tratar la falta de requisitos estables en un sistema, es mediante la agilidad de un grupo pequeño de desarrollo (Weitzenfeld, 2005).

Una aplicación web (web based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (Hiper Text Transfer Protocol (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones (Luján, 2001).

El Sistema de Tutorial Universitario, es una forma de dar respuesta a la creciente exigencia social sobre los rendimientos y resultados del sistema universitario, en ese sentido la acción tutorial se articula como un recurso para que los estudiantes puedan recibir una asistencia personalizada, para configurar mejor su formación personalizada, para optimizar su rendimiento académico, de tal manera que para mejorar la calidad universitaria es pertinente la acción tutorial universitaria (Directiva VRAC-UNSCH, 2013).

Los alumnos de la Universidad se encuentran viviendo en un mundo que cambia constantemente, es importante elaborar propuestas que los integre al mundo moderno y cambiante en el que vivimos, es de vital importancia que un Tutor realice actividades para acompañar entender y promover a los universitarios.

El objetivo principal es la de desarrollar una Aplicación Web de Apoyo al Sistema de Tutoría de la Universidad Nacional de San Cristobal de Huamanga, 2016, utilizando metodología de desarrollo agil.

## **CAPITULO I**

### **PLANTEAMIENTO DE LA INVESTIGACIÓN**

#### **1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA**

El 28 de diciembre del 2012, fue emitida la Resolución de Consejo Universitario N° 2012-UNSCH-cu, que resuelve aprobar con vigencia a partir de la fecha, el Reglamento del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga, el que consta de cinco capítulos, diecinueve artículos y dos disposiciones finales.

Así mismo, el vicerrectorado académico de la Universidad Nacional de San Cristóbal de Huamanga, generó con fecha agosto del 2013 la Directiva SN- 2013-VRAC-UNSCH, el documento de gestión “Lineamientos para la implementación del sistema de tutoría en las escuelas de formación profesional”, cuya finalidad establecida es que la Universidad Nacional de San Cristóbal de Huamanga implemente el Sistema de Tutoría de acuerdo al modelo del CONEAU en cada una de las Escuelas de Formación Profesional. El ámbito de acción de la directiva SN-2013-VRAC-UNSCH establece que las disposiciones establecidas son de cumplimiento obligatorio según competencia, por las escuelas de formación profesional, departamentos académicos, facultades, oficinas involucradas, vicerrectorado académico y otras dependencias que tienen relación con el cumplimiento del Sistema Tutorial.

El punto c) del Artículo 9°, del reglamento indicado, establece como obligaciones de los tutores el de “registrar los datos del estudiante y las entrevistas en la base de datos del Sistema Tutorial con la finalidad que le permita contar con información necesaria para realizar el seguimiento y la evaluación de su desempeño académico”

Por otro lado, como centro de la eficacia de este sistema, se indica en el punto f) del Artículo 14°: “disponer de un sistema de control y evaluación para tutores y alumnos”, así mismo el punto b) del Artículo 17° con respecto al proceso de seguimiento y evaluación tutorial establece que el seguimiento y evaluación comprenderá: “las actividades desarrolladas por el tutor respecto a información, orientación y atención a alumnos, conforme al sistema tutorial”.

Se observa entonces, que de acuerdo al Reglamento del Sistema de Tutoría y a la directiva

sobre los lineamientos para la implementación del sistema de tutoría en las escuelas de formación profesional; la universidad tiene como propósito firme la implementación del sistema de tutoría. Así, algunas de las Escuelas de Formación Profesional vienen acatando ya las disposiciones de las autoridades con respecto a la implementación del sistema de tutoría.

Finalmente, pese a la existencia del reglamento del sistema de tutoría y a la directiva de lineamientos para la implementación de dicho sistema, así como a las iniciativas de implementación de ciertas escuelas; la universidad en su conjunto carece de una herramienta tecnológica que de soporte a las actividades de este sistema de tutoría tanto para los tutores como para tutorados y la parte administrativa encargada de la gestión del proceso de seguimiento y evaluación tutorial.

## **1.2. DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN**

### **1.2.1. PROBLEMA PRINCIPAL**

¿Una aplicación web, permite gestionar eficazmente las actividades del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga, 2016?

### **1.2.2. PROBLEMAS SECUNDARIOS**

- a. ¿Cuáles son los resultados del análisis del desarrollo de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga?
- b. ¿Cuáles son los resultados del diseño del desarrollo de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga?
- c. ¿Cuáles son los resultados de la implementación del desarrollo de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga?
- d. ¿Cuáles son los resultados de las pruebas en el desarrollo de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga?

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

### **1.3.1. OBJETIVO GENERAL**

Desarrollar una aplicación web de apoyo al Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga, Ayacucho 2016. Mediante la aplicación del

proceso ágil de desarrollo de software Programación Extrema (XP), la programación orientada a objetos, el uso de un gestor de base de datos relacional, con el propósito de brindar una herramienta de soporte a las actividades del sistema de tutoría y la finalidad de automatizar actividades del sistema tutorial.

### **1.3.2. OBJETIVOS ESPECÍFICOS**

- a. Obtener los resultados del análisis de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga.
- b. Obtener los resultados del diseño de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga.
- c. Obtener los resultados de la implementación de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga.
- d. Obtener los resultados de las pruebas de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga.

## **1.4. JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN**

### **1.4.1. IMPORTANCIA DEL TEMA**

El sistema tutorial universitario es una forma de dar respuesta a la creciente exigencia social sobre los rendimientos y resultados del sistema universitario, en ese sentido la acción tutorial se articula como un recurso para que los estudiantes puedan recibir asistencia personalizada tanto en el aspecto personal como aspectos académicos, para optimizar su rendimiento. Por otro lado es evidente que para mejorar la calidad universitaria es pertinente la acción tutorial universitaria, de acuerdo a la experiencia en muchas instituciones universitarias.

### **1.4.2. JUSTIFICACIÓN**

El sistema tutorial requiere de una herramienta tipo Software que permita la interacción de los entes participantes en el sistema de tutoría. Como por ejemplo para los docentes es necesario registrar los datos del estudiante y las entrevistas con ellos en la base de datos del Sistema Tutorial, con la finalidad que estas acciones le permitan al docente contar con información necesaria para realizar el seguimiento y la evaluación de su desempeño académico.

Dentro de la eficacia del Sistema de Tutoría es necesario disponer de un sistema de control y evaluación para tutores y alumnos; y con respecto al proceso de seguimiento y evaluación

tutorial es necesario considerar las actividades desarrolladas por el tutor respecto a información, orientación y atención a alumnos, conforme al Sistema Tutorial.

### **1.4.3. DELIMITACIÓN DE LA INVESTIGACIÓN**

#### **DELIMITACIÓN ESPACIAL**

El trabajo de investigación se realizó en el Proyecto Hatun Ñan de la Universidad Nacional de San Cristóbal de Huamanga.

#### **DELIMITACIÓN TEMPORAL**

Para realizar la presente investigación se tomó como referencia datos que se recolectaron en la Universidad Nacional de San Cristóbal de Huamanga, durante el año 2016.

#### **DELIMITACIÓN TEÓRICA**

La investigación abarcó los siguientes temas: Sistema de tutoría y Aplicación

## **CAPITULO II**

### **MARCO DE REFERENCIA DE LA INVESTIGACIÓN**

#### **2.1. ANTECEDENTES DE LA INVESTIGACIÓN**

Gómez (2005), en su trabajo de investigación titulado “Sistema Inteligente Para La Tutoría Académica de La Uaem: Análisis de Aplicación en la Facultad de Ciencias Políticas y Administración Pública”, concluyó que La incorporación de las Tecnologías de la Información y la Comunicación a las actividades de tutoría permitirán la posibilidad de intercambio de información y flexibilización de tiempos y horarios, pues la “conciencia en el espacio y tiempo de tutor y tutorado” se modificará cuando se haya generalizado el uso del SITUAEMEX. Sin embargo, la incorporación del sistema automatizado no tiene como meta “desaparecer” por completo la relación cara a cara entre tutorados y tutores, solo es un medio que le permite a los tutores generar información para tomar decisiones, elaborar un plan de trabajo o generar informes de manera sencilla y para los tutorados se garantiza un espacio en el que puedan dialogar con su tutor; a través del correo electrónico interno en el que el tutor puede concertar alguna cita con su tutorado o bien para preguntarle sobre algún tópico en particular. En el caso del SITUAEMEX, la introducción de las TIC como una herramienta de gestión permite a los tutorados participar de las tareas de tutoría, pues desde la página del tutorado suben al sistema las unidades de aprendizaje que cursa en cada periodo y al tutor le permite dar un mayor seguimiento del desempeño del tutorado durante el semestre.

Rodriguez (2003), en su tesis de postgrado titulado: “Diseño y Experimentación de un Sistema de Tutoría para la FIQ-UNAC”, concluyó que las necesidades de orientación de los tutorados, para apoyar su formación profesional; están relacionadas con las necesidades de orientación de tipo académico, social, familiar, económico y de salud.

También manifiesta que el sistema de tutoría diseñado está compuesto por siete elementos: personal (organización y funciones), instrumentos de orientación y control (reglamento), infraestructura (oficinas y mobiliario), materiales de escritorio y archivo, capacitación continua, la comunicación en el sistema y el presupuesto.

Ávila (2001), en su tesis de maestría titulado “Sistema automatizado para el control de los expedientes de tutoría en la sede universitaria de Florencia”, concluyó que las tecnologías

de información y comunicaciones constituyen en esta nueva universidad medios que facilitan la misión de la misma. La automatización de los expedientes de tutoría constituye un ejemplo práctico de tecnologías al servicio del proceso docente educativo.

## **2.2. MARCO TEÓRICO**

### **2.2.1. APLICACIÓN WEB**

Una aplicación web (web based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (Hiper Text Transfer Protocol (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones (Luján, 2001).

“Los sistemas y aplicaciones basados en Web hacen posible que una población extensa de usuarios finales dispongan de una gran variedad de contenido y funcionalidad. La ingeniería Web no es un clónico perfecto de la Ingeniería del software, pero toma prestado muchos de los conceptos básicos de la Ingeniería del software, dando importancia a las mismas actividades técnicas y de gestión. Existen diferencias sutiles en la forma en que se llevan a cabo estas actividades, pero la filosofía primordial es idéntica dado que dicta un enfoque disciplinado para el desarrollo de un sistema basado en computadora” (Pressman, 2002).

Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor a través de internet o de una intranet. Las aplicaciones web son populares debido a la sencillez del navegador web como cliente ligero. La facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad (Barry y Lang, 2001).

Una aplicación web es un sitio web donde la navegación es a través del sitio y la entrada de datos por parte de un usuario afecta el estado de la lógica del negocio. En esencia, una aplicación web usa un sitio web como entrada (front-end) a una aplicación típica (Senn, 1997).

### **2.2.2. CICLO DE VIDA DEL SOFTWARE**

#### **a. ANALISIS**

Su objetivo es definir con detalle las necesidades de información que tendrá que resolver el software, sin tener en cuenta, por el momento, los medios técnicos con los que se tendrá que llevar a término el desarrollo del software (Campderrich, 2003).

Es el proceso de comprensión y definición de que servicios se requieren del sistema y de identificación de las restricciones de funcionamiento y desarrollo del mismo, es una etapa crítica ya que los errores en esta etapa originan inevitablemente problemas posteriores en el diseño e implementación del sistema (Sommerville, 2005).

Tiene como objetivo construir una arquitectura capaz de resolver el problema bajo condiciones ideales. Esto significa desarrollar una estructura lógica del sistema, la cual debe ser estable y extensible, el análisis se enfoca en que debe hacer el sistema, en lugar de como se supone que lo hará (Weitzenfeld, 2005).

#### **b. DISEÑO**

Si el análisis especifica el problema o “qué tiene que hacer el software”, el diseño especifica la solución a este problema o “cómo el software tiene que hacer su función”. Del software hay que diseñar varios aspectos diferenciados: Su arquitectura general, las bases de datos, la especificación de cada programa y las interfaces con el usuario (Campderrich, 2003).

El propósito del modelo de diseño es extender la arquitectura de análisis. Debe definir todo lo necesario para alcanzar el código final concentrándose en dos aspectos principales: Diseño de objetos y Diseño de sistemas (Weitzenfeld, 2005).

El diseño es una representación significativa de ingeniería de algo que se va a construir. Se puede hacer el seguimiento basándose en los requisitos del cliente, y al mismo tiempo la calidad se puede evaluar y cotejar con el conjunto de criterios predefinidos para obtener un diseño bueno (Pressman, 2002).

#### **c. IMPLEMENTACIÓN**

La etapa de implementación del software es el proceso de convertir una especificación del sistema en un sistema ejecutable. Siempre implica los procesos de diseño y programación de software, mientras que un diseño de software es una descripción de la estructura del software que se va a implementar, los datos que son parte del sistema, las interfaces entre los componentes del sistema y, algunas veces, los algoritmos utilizados (Sommerville, 2005).

El modelo de implementación toma el resultado del modelo de diseño para generar el código final. Esta traducción debe ser relativamente sencilla y directa, ya que las decisiones mayores han sido tomadas durante las etapas previas. Durante la etapa de implementación se adapta al lenguaje de programación y/o a la base de datos, según la especificación del diseño (Weitzenfeld, 2005).



#### **d. PRUEBA**

Consiste en probar el software desde distintos puntos de vista de una manera planificada y, naturalmente, localizar y corregir dentro del software y su documentación los errores que se detecten (Campderrich, 2003).

Se utiliza para mostrar que el sistema se ejecuta a su especificación y que cumple las expectativas del usuario que lo comprobará (Sommerville, 2005).

Es el responsable de revisar la calidad del sistema. Este modelo consiste en la evaluación del sistema o prueba de especificación y la verificación o prueba de resultado. De manera tradicional, el modelo de pruebas combina pruebas de unidad y pruebas de integración (Weitzenfeld, 2005).

#### **2.2.3. INGENIERÍA DEL SOFTWARE**

En general, a cada tipo de producto industrial corresponde un tipo de ingeniería, entendida como un conjunto de métodos, técnicas y herramientas que se utiliza tanto para desarrollar el producto (es decir, elaborar el proyecto o prototipo) como para fabricarlo (afinando más se puede decir que existen, pues, dos ingenierías para cada tipo de productos: la del producto y la del proceso) (Campderrich, 2003).

La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza (Sommerville, 2005).

La ingeniería del software es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo (Pressman, 2002).

#### **2.2.4. PROGRAMACIÓN ORIENTADA A OBJETOS**

La Programación Orientada a Objetos (POO) constituye una nueva forma de organización del conocimiento en la que las entidades centrales son los objetos. En un objeto se unen una serie de datos con una relación lógica entre ellos, a los que se denomina variables de instancia, con las rutinas necesarias para manipularlos, a las que se denomina métodos. Los objetos se comunican unos con otros mediante interfaces bien definidas a través de paso de mensajes (Joyanes, y Zahonero, 2002).

“La Programación Orientada a Objetos es un paradigma increíblemente poderoso y natural para crear programas que sobreviven a los cambios inevitables que acompañan al crecimiento y envejecimiento de cualquier sistema. Debido a que se entiende

la función de cada objeto, se tiene interfaces limpias y fiables entre los objetos, se pueden extraer partes de un sistema antiguo con facilidad y sin miedo” (Patrick, 1996, p.16).

La programación orientada a objetos es, realmente, otra técnica para implementar el famoso dicho de la programación: "divide y vencerás". La idea es encapsular datos y métodos en objetos, de forma que cada objeto sea semiautónomo, encerrando métodos y datos privados (es decir, internos) y salvándolos del desorden general que les rodea. Así, el objeto puede interactuar con el resto del programa por medio de una interfaz bien definida por sus métodos públicos (es decir, se les puede invocar desde fuera) (Holzner, s.f.).

### **2.2.5. SISTEMA DE GESTIÓN DE BASE DE DATOS**

“Sistema de Gestión de Base de Datos (SGBD), conjunto de programas que permiten la implantación, acceso y mantenimiento de la base de datos. El SGBD, junto con la base de datos y con los usuarios, constituye el Sistema de Base de Datos” (Pantigoso, 2008, pp. 22-23).

Según Batini (1994), son paquetes de software para la gestión de base de datos en particular, para almacenar, manipular y recuperar datos en la computadora.

Los sistemas de gestión de bases de datos (en inglés Data Base Management System, abreviado DBMS), son un conjunto de programas que maneja la base de datos y controla el acceso a los datos guardados en éstas. El sistema administrador de base de datos permite compartir los datos de la base de datos entre múltiples aplicaciones y usuarios (Rob y Coronel, 2004).

### **2.2.6. PROGRAMACIÓN EXTREMA**

La Programación Extrema (XP, eXtreme Programming) es un modelo de proceso de software que toma los principios y prácticas aceptadas, y las lleva a niveles extremos. Tiene como objetivo reducir el riesgo en el ciclo de vida del software mediante grupos de desarrollo pequeños. Considera que la mejor manera de tratar la falta de requisitos estables en un sistema, es mediante la agilidad de un grupo pequeño de desarrollo (Weitzenfeld, 2005).

La Programación Extrema (XP) es posiblemente el método ágil más conocido y ampliamente utilizado, el enfoque fue desarrollado utilizando buenas prácticas reconocidas, como el desarrollo iterativo, y con la participación del cliente en niveles

extremos. En la programación extrema, todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales implementa directamente como una serie de tareas. Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integre al sistema (Sommerville, 2005).

“XP es tanto una filosofía del trabajo de programación como un conjunto de guías para acometer esta tarea. Algunas de estas guías se reflejan en otras metodologías recientes, pero las dos contribuciones más distintivas e importantes en mi opinión son "escribir las pruebas en primer lugar" y "la programación a pares". Aunque Beck discute bastante todo el proceso en sí, señala que si se adoptan únicamente estas dos prácticas, uno mejorará enormemente su productividad y nivel de confianza” (Eckel, 2002, p. 45).

### **2.2.6.1. VALORES EN LA PROGRAMACIÓN EXTREMA**

Según Beck (1999), la Programación Extrema se basa en cuatro valores, que deben estar presentes en el equipo de desarrollo para que el proyecto tenga éxito, siendo los siguientes:

#### **a. COMUNICACIÓN**

Muchos de los problemas que existen en proyectos de software, se deben a problemas de comunicación entre las personas. La comunicación permanente es fundamental en XP, dado que los artefactos son pocos, el diálogo frontal, cara a cara, entre desarrolladores, administrador y el cliente es el medio básico de comunicación. Una buena comunicación se debe mantener durante todo el proyecto.

#### **b. SIMPLICIDAD**

El proceso XP, es una metodología ágil, que apuesta por la sencillez, en su máxima expresión. Sencillez en diseño, en código, en los procesos, etc. La sencillez es fundamental para que todos entiendan el código y, se trata de mejorar mediante recodificaciones continuas.

#### **c. RETROALIMENTACIÓN (FEEDBACK)**

La retroalimentación debe practicarse en forma permanente. El cliente debe brindar retroalimentación de las historias de usuario desarrolladas, a fin de considerar sus comentarios para la siguiente iteración, y para entender, cada vez más, sus necesidades. Los resultados de las pruebas unitarias, son también una retroalimentación permanente que tienen los desarrolladores acerca de la calidad de la aplicación.

#### **d. CORAJE**

Cuando se encuentran problemas serios en el diseño, o en cualquier fase del ciclo de XP, se debe tener el coraje suficiente para encontrar la solución, sin importar que tan difícil sea. Si es necesario cambiar completamente parte del código, hay que hacerlo, sin importar cuánto tiempo se ha invertido en desarrollar el código a cambiar.

#### **2.2.6.2. CICLO DE VIDA IDEAL DEL XP**

Según DSIIC (2006), El ciclo de vida ideal de XP consiste de seis fases: exploración, planificación de la entrega (release), iteraciones, producción, mantenimiento y muerte del proyecto.

#### **a. FASE DE EXPLORACIÓN**

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

A partir del punto citado y la revisión de otras bibliografías se resume el siguiente cuadro, haciendo mención las tareas, artefactos, técnica y responsables del desarrollo.

*Tabla 2.1: Fase de exploración*

<b>TAREA</b>	<b>ARTEFACTO</b>	<b>TECNICA</b>	<b>RESPONSABLES</b>
<b>Escribir historias de usuario</b>	Historia de usuario	Describir brevemente la historia de usuario con la regla del negocio (lo que el sistema debe hacer)	Cliente
<b>Probar las tecnologías a utilizar</b>	Lista tecnologías a utilizar.	Explorar posibilidades de uso de tecnologías.	Programador Entrenador
<b>Establecer Arquitectura inicial</b>	Arquitectura técnica inicial.	Establecer las tecnologías a usar.	Programador Entrenador

## **b. FASE DE PLANIFICACIÓN DE LA ENTREGA**

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

A partir del punto citado y la revisión de otras bibliografías se resume el siguiente cuadro, haciendo mención las tareas, artefactos, técnica y responsables del desarrollo.

*Tabla 2.2: Fase de planificación*

<b>TAREA</b>	<b>ARTEFACTO</b>	<b>TECNICA</b>	<b>RESPONSABLES</b>
<b>Establecer prioridad de las historias de usuario</b>	Historias de usuario de prioridad	Seleccionar las historias de usuario que tiene mayor prioridad	Cliente
<b>Estimar esfuerzo para historias de usuarios</b>	Estimación de esfuerzo	Estimar y asignar esfuerzo (semana) para cada historia de Usuario	Programador
<b>Elaborar plan de la entrega</b>	Plan de la entrega	Realizar el cronograma para el plan de entrega.	Programador

## **c. FASE DE ITERACIÓN**

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable.

A partir del punto citado y la revisión de otras bibliografías se resume el siguiente cuadro, haciendo mención las tareas, artefactos, técnica y responsables del desarrollo.

*Tabla 2.3: Fase de iteración*

<b>TAREA</b>	<b>ARTEFACTO</b>	<b>TECNICA</b>	<b>RESPONSABLE</b>
<b>Definir la arquitectura técnica</b>	Arquitectura técnica	Actualizar la técnica inicial	Cliente Programador Entrenador
<b>Escribir tareas de ingeniería</b>	Tarea de ingeniería	Dividir cada historia de usuario en tareas, describir usando reglas del negocio cada tarea de ingeniería.	Cliente Programador
<b>Formular el plan de iteraciones</b>	Plan de iteración	Estimar y asignar esfuerzo para desarrollar una tarea de ingeniería.	Programador
		Asignar una tarea de ingeniería al programador	Entrenador Programador
		Utilizar el plan de versión Actualizar el plan con tareas de ingeniería de la siguiente iteración.	Programador Entrenador
<b>Crear pruebas de aceptación</b>	Caso de pruebas de aceptación	Escribir pruebas de aceptación para cada historia de usuario por iteración	Cliente Encargado de pruebas

<b>Implementar las interfaces</b>	GUI	Diseñar con precisión la GUI relacionada a cada historia de usuario	Cliente Programador
<b>Escribir tarjetas CRC.</b>	Tarjeta CRC	Identificar atributos. Identificar colaboración Identificar responsabilidades	Cliente Programador
<b>Implementar código fuente</b>	Código fuente	Implementa y Codificar, una tarea de ingeniería. Hacer refactoring Mover programadores	Programador supervisor
<b>Ejecutar pruebas unitarias</b>	Reporte de pruebas unitarias	Ejecutar el modulo de cada prueba unitaria. Modificar código fuente si la prueba unitaria muestra un resultado incorrecto	Programador
<b>Ejecutar pruebas de aceptación</b>	Reporte de pruebas de aceptación	Correr la última versión de una iteración Utilizar los casos de prueba de <u>aceptación</u>	Cliente Encargado de pruebas

### 2.2.6.3. HISTORIAS DE USUARIO

Según Wake (2000), las historias de usuarios sirven el mismo propósito como los casos de uso, pero no son los mismos. Se utilizan para crear estimaciones de tiempo para la reunión de planificación de la liberación. También se utilizan en lugar de un gran documento sobre los requisitos. Las historias de usuarios están escritas por los clientes como

las cosas que el sistema tiene que hacer para ellos. Son similares a los escenarios de uso, salvo que no se limitan a describir una interfaz de usuario. Ellos están en el formato de alrededor de tres frases del texto escrito por el cliente sin la terminología técnico-sintaxis. Las historias de los usuarios también impulsan la creación de las pruebas de aceptación. Uno o más pruebas automatizadas de aceptación deben ser creados para verificar las historias de usuario, si han sido correctamente implementados.

*Tabla 2.4: Plantilla historia de usuario*

---

**HISTORIA DE USUARIO**

---

**Número:**                      **Nombre:**

---

**Usuario:**

---

**Modificación de Historia Numero:**    **Iteración asignada:** Alto

---

**Prioridad de Negocio:**                      **Riesgo en desarrollo:**  
 (Alta/Media/Baja)                                      (Alta/Media/Baja)

---

**Descripción:**

---

**Observaciones:**

---

#### **2.2.6.4. TARJETAS C.R.C.**

Según Castillo (2010), el uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos, olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

*Tabla 2.5: Plantilla tarjeta CRC*

---

**TARJETA CRC**

---

**Numero:**    **Escenario:**

---

**Nombre CRC:**

---

**Responsabilidades**                                      **Colaboradores**                                      **Atributos**

---



---

**Observaciones:**

---

### **2.2.6.5. PRUEBAS DE ACEPTACIÓN**

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo; el cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada; las pruebas de aceptación son consideradas como “pruebas de caja negra”, los clientes son responsables de verificar que los resultados de éstas pruebas sean correctos. Asimismo, en caso que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no está terminada mientras no pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

*Tabla 2.6: Plantilla prueba de aceptación*

---

#### **PRUEBA DE ACEPTACIÓN**

**Caso de Prueba:**

**Numero de prueba:**

**Numero Historia de Usuario:**

**Nombre de Caso de Prueba:**

**Descripción:**

---

**Condiciones de Ejecución:**

---

**Entradas:**

---

**Resultados esperados:**

---

**Evaluación:**

---

### **2.2.7. TUTORÍA UNIVERSITARIA**

Es la acción de intervención formativa destinada al seguimiento de los estudiantes que se ocupa de la información, la formación y la orientación personalizada, grupal y virtual, además se centra su atención en facilitar la adaptación a la universidad, apoyar el proceso de aprendizaje, mejorar el rendimiento académico, brindar soporte emocional, social y profesional (Directiva VRAC-UNSCH,2013).

Se perfila como uno de los factores necesarios de apoyo y asesoramiento didáctico-curricular y psicopedagógico que favorece la calidad e innovación del aprendizaje, potencia el desarrollo formativo y profesional de los estudiantes y mejora las relaciones y acciones de acogida, bienestar académico y social entre éstos, los profesores, egresados y la propia institución.

### **2.2.8. SISTEMA TUTORIAL UNIVERSITARIO**

Es una forma de dar respuesta a la creciente exigencia social sobre los rendimientos y resultados del sistema universitario, en ese sentido la acción tutorial se articula como un recurso para que los estudiantes puedan recibir una asistencia personalizada, para configurar mejor su formación personalizada, para optimizar su rendimiento académico, de tal manera que para mejorar la calidad universitaria es pertinente la acción tutorial universitaria (Directiva VRAC-UNSCH, 2013).

#### **a. TUTORÍA**

Es un acompañamiento y guía al estudiante de carácter personal, académico, social y profesional en su formación integral, el cual tiene por finalidad realizar una orientación sistemática a lo largo de toda la vida estudiantil para que el estudiante mejore su rendimiento académico, solucione sus dificultades universitarias y desarrolle una actitud proactiva con capacidad de reflexión crítica y convivencia social y se inserte con liderazgo al campo laboral (Directiva VRAC-UNSCH, 2013).

#### **b. TUTOR**

Es el docente que realiza el acompañamiento tutorial desarrollando contenidos técnicos y científicos de su campo de conocimiento, facilitador del aprendizaje, consejero brindando orientaciones certeras y promoviendo la independencia, la autoestima, la resiliencia u la definición, el manejo de habilidades sociales logrando obtener un proyecto de vida personal (Directiva VRAC-UNSCH, 2013).

**c. TUTORADO**

Es un estudiante dinámico de aprendizaje, en un proceso de formación integral con responsabilidad social y con gran interés de desarrollarse personal, académica y profesionalmente (Directiva VRAC- UNSCH, 2013).

**d. MODALIDADES DE TUTORÍA**

El VRAC-UNSCH, define las siguientes modalidades:

Tutoría Individual: Se realiza en función a sus características y necesidades particulares, la cual se realiza mediante diferentes técnicas como la entrevista, consejería y otros.

Tutoría Grupal: Se realiza en función de los criterios de agrupación, se orienta a abrir un espacio de comunicación, conversación e intercambio grupal en el cual se revisan y discuten temas que sean de interés, inquietud y preocupación.

Tutoría virtual: Se realiza a distancia mediante la utilización de redes sociales.

Tutoría de Pares: Dirigidos por los tutores y conformado por grupos de estudiantes de diferentes series formando pares

**e. ACTIVIDADES DE LA ACCIÓN TUTORIAL**

El VRAC-UNSCH, define las siguientes actividades de la acción tutorial:

- a Elaboración del perfil académico, familiar y social.
- b Elaboración del plan tutorial individual y grupal
- c Inducción al estudiante ingresante.
- d Orientación en la matrícula.
- e Acompañamiento en el área personal, académica, social y profesional.
- f Planeamiento de estrategias tutoriales.
- g Evaluación del sistema tutorial.
- h Presentación del informe de la acción tutorial.

**f. JUSTIFICACIÓN DE LAS ACCIONES ORIENTADAS Y DE TUTORÍA EN LA UNIVERSIDAD**

Es ampliamente compartido el falso supuesto de que la madurez alcanzada por los individuos en edad universitaria les hace prácticamente autónomos y como necesitados de ayuda.

Sin embargo diversas estadísticas aplicadas a estudiantes de muchas universidades muestran que éstos presentan necesidades formativas y de orientación que van más allá de los contenidos tratados en las asignaturas de sus respectivas titulaciones (Roig, 1984).

Se debe precisar también si el alumnado no demanda más orientación es, en parte, por que difícilmente puede pedir lo que no conoce.

Rincón (2000), afirma que la universidad, además de procurar la idoneidad científica de sus egresados, debe incidir en sus procesos globales de formación y en aquellos que se relacionan con un aprendizaje más maduro, consciente y autónomo. Todo ello es necesario para cualquier desempeño profesional y aún para el propio éxito académico.

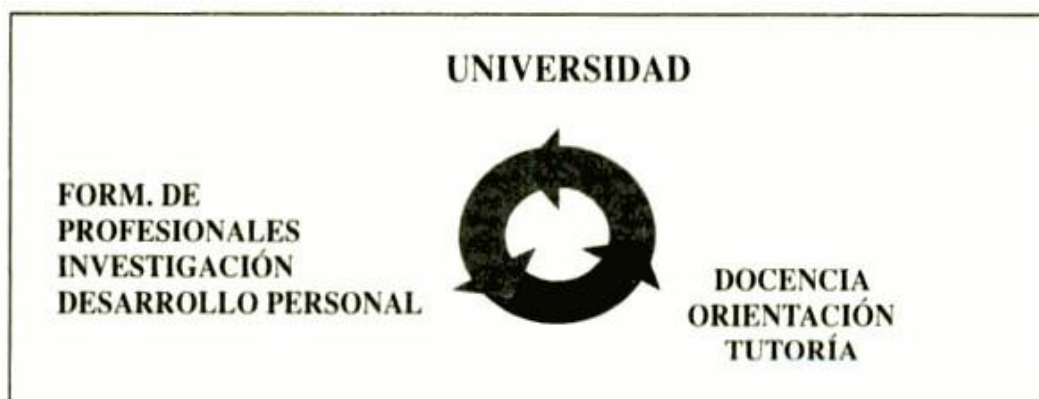


Figura 2.1: Funciones de la universidad

En el marco de la orientación se inscribe la atención tutorial personalizada como actividad no especializada pero de crítica importancia para el logro de los objetivos de la universidad, como se muestra en la figura anterior. En su sentido más académico, ligada a los contenidos científicos, la función tutorial posee una larga tradición en algunas universidades extranjeras.

Según Alvares (2002), el profesor universitario también debe poseer una faceta como tutor y orientador, dirigida a desarrollar de forma integral a su alumnado. Además de ser un profesional de la materia que imparte, debe conocer el proceso de desarrollo de la carrera profesional y ser capaz de planificar y poner en práctica intervenciones educativas que faciliten el logro de los objetivos establecidos.

Y Ferrer (2003) afianza la afirmación anterior indicando que la tutoría universitaria es una actividad de carácter formativo que incide en el desarrollo integral de los estudiantes en su dimensión intelectual, académica, profesional y personal.

#### **g. TUTORÍA DE ENSEÑANZA – APRENDIZAJE**

Es el tipo de tutoría implementable en universidades con un número reducido de estudiantes, como lo precisa muy bien Rincón (2000): “es la más habitual entre nosotros aunque, no se encuentra suficientemente explotada, tiene como principal objetivo apoyar

los procesos de aprendizaje e investigación de los estudiantes y constituye la gran oportunidad de individualización de la enseñanza, con más posibilidades en aquellas universidades menos masificadas. En cualquier caso, podemos decir que es una forma de docencia que ofrece la posibilidad de enfatizar las estrategias de estudio propias de cada asignatura. Además, constituye una vía para la individualización, una medida eficaz de atención a las dificultades de aprendizaje y un camino para conseguir autonomía en la adquisición de los contenidos académicos”.

#### **h. RELACIÓN TUTOR – ALUMNO**

La labor del tutor va dirigida hacia el alumno, la relación que se establece entre el tutor y el alumno es el encuentro de dos personas, en el que cada uno llega con su propia historia personal, con sus características, pensamientos, sentimientos, expectativas y preocupaciones. Cada uno se relaciona con el otro como un todo, se pone en juego la persona integral de cada uno y pueden, por lo tanto, presentarse situaciones de mutua influencia.

En la relación tutor-alumno, de acuerdo a Zavala (2003) se debe considerar:

- Tratar de conocer a sus alumnos, su mundo, sus inquietudes, preocupaciones, temores y dudas.
- Cada uno de los alumnos es una persona independiente y autónoma, que tiene su propia individualidad y personalidad.
- No pretender que el alumno haga las cosas de acuerdo al criterio del tutor, ayudarlo a encontrar sus mejores decisiones.
- No pretender cambiarle la vida, ayudarlo a vivir de la mejor manera su realidad.
- Ser sinceros.

#### **i. MODELOS DE ACCIÓN TUTORIAL UNIVERSITARIOS**

La concepción actual de tutoría dentro del campo universitario, puede entenderse desde diferentes perspectivas o enfoques teóricos, dando lugar a un tipo de actuación que desemboca en un modelo u otro de intervención. Siguiendo a Rodríguez (2004), en la actualidad y en el contexto universitario, conviven cuatro grandes modelos de tutoría:

**Modelo académico:** en el que la acción docente se limita a informar u orientar sobre su asignatura. Este modelo está vinculado a la tradición continental, en un ambiente de máxima autonomía por parte del profesor y libertad de todos los miembros de la comunidad

universitaria.

**Modelo de desarrollo personal:** relacionado con el modelo anglosajón, en el que el objetivo fundamental es el desarrollo integral del estudiante. La acción tutorial, en la línea de este modelo, comprende ámbitos que van más allá del estrictamente académico, para adentrarse en cuestiones profesionales y personales.

**Modelo de desarrollo profesional:** en el que la figura del tutor en la universidad se ve complementada por la del tutor en el lugar de trabajo, donde se desarrollan las prácticas en situaciones reales, y en las que el principal objetivo es el desarrollo de destrezas y competencias personales, académicas y profesionales para que el perfil del estudiante se ajuste al máximo a las exigencias del puesto profesional que va a desempeñar.

**Modelo de formación integral:** este modelo de tutoría es el ideal para desarrollar en el contexto universitario. Es posiblemente el modelo más completo al impulsar el desarrollo integral del alumno, en sus facetas: intelectual, afectiva y profesional. No obstante es un modelo que requiere, por un lado, la formación del profesorado en acciones de orientación debido a la amplitud de su actuación y a la diversidad y complejidad de los roles que ha de desempeñar, y por el otro la colaboración de los equipos de apoyo, como los gabinetes de orientación, para establecer vínculos y redes con centros de educación e instituciones profesionales.

Como vemos, cada modelo engloba objetivos, implicaciones y requisitos diferentes, consideramos que en gran medida complementarios y mantienen exigencias muy diversas. Los beneficios y exigencias de cada uno de los ellos podrán ser analizados para su posible aplicación de modo que cada centro o titulación podrá escoger aquella que mejor se ajuste a sus circunstancias. De todas formas no se trata de establecer o proponer un modelo, por muy completo que sea, si luego no existen voluntades o posibilidades de aplicación y desarrollo. Se trata de escoger aquel que estratégicamente sea factible de aplicar.

De acuerdo a Rodríguez (2004), las modalidades de tutoría realizada podrían

clasificarse como se muestra en la siguiente tabla.

Criterios	Modelos de Tutoría
<b>1. En relación con los servicios de orientación</b>	<ul style="list-style-type: none"> <li>· <u>Modelo tutorial puro</u>: el programa de tutorías funciona de forma independiente a los servicios de orientación de la institución.</li> <li>· <u>Modelo mixto</u>: el modelo tutorial se organiza con apoyo de los servicios de orientación.</li> </ul>
<b>2. Según el contenido de las tutorías</b>	<ul style="list-style-type: none"> <li>· <u>Tutoría académica</u>: orienta al estudiante sobre contenidos disciplinares de la materia.</li> <li>· <u>Tutoría de prácticas</u>: tiene como finalidad el proporcionar al alumno en prácticas habilidades y herramientas de cara a desarrollar las prácticas.</li> <li>· <u>Tutoría de proyecto</u>: asesora y orienta el proyecto del alumno (en aquellas carreras que se demanda un proyecto final).</li> <li>· <u>Tutoría de asesoramiento personal</u>: se trata de una tutoría especializada para abordar determinadas circunstancias personales de algunos estudiantes.</li> </ul>
<b>3. En relación al tiempo</b>	<ul style="list-style-type: none"> <li>· <u>Tutoría de curso</u>: el seguimiento del alumno se hace en un tramo de su trayecto formativo, un curso académico.</li> <li>· <u>Tutoría de carrera o de itinerario académico</u>: el seguimiento del alumno se realiza a lo largo de los estudios universitarios, y aborda temas generales relativos a itinerarios curriculares, salidas profesionales, etc.</li> </ul>
<b>4. En cuanto a la figura del tutor</b>	<ul style="list-style-type: none"> <li>· <u>Profesor-tutor</u>: el tutor es un profesor.</li> <li>· <u>Tutoría entre iguales (peer tutoring)</u>: los alumnos de cursos superiores son los encargados de orientar a sus compañeros de primeros cursos.</li> </ul>
<b>5. En cuanto a los destinatarios</b>	<ul style="list-style-type: none"> <li>· <u>Tutoría individual</u>: consiste en una acción personalizada.</li> <li>· <u>Tutoría grupal</u>: permite abordar temas comunes, intercambiar experiencias... y posibilita la acción sobre más alumnos en menor tiempo.</li> </ul>

Tabla 2.8: Clasificación de las modalidades de tutoría realizada

## **CAPITULO III**

### **METODOLOGÍA DE LA INVESTIGACIÓN**

#### **3.1 TIPO DE INVESTIGACIÓN**

El tipo de investigación es aplicada, se desarrolló un producto software “Aplicación Web para el Sistema de Tutoría de la Universidad Nacional de San Cristóbal De Huamanga, Ayacucho 2016”.

#### **3.2 NIVEL DE INVESTIGACIÓN**

El nivel de investigación es descriptivo; desde el punto de vista científico una investigación descriptiva muestra, narra, reseña o identifica hechos, situaciones, rasgos, características de un objeto de estudio o se diseñan productos, modelos, prototipos, guías, etcétera. Pero no se dan explicaciones o razones del porque de las situaciones, los hechos, los fenómenos, etc. (Bernal, 2006).

#### **3.3 DISEÑO DE LA INVESTIGACIÓN**

Según Carrasco (2005), el diseño de investigación transversal descriptivo, se emplea para analizar y conocer las características, rasgos, propiedades y cualidades de un hecho realizado en un momento determinado de tiempo.

Según al punto citado, el diseño de investigación para el presente trabajo de tesis es el diseño transversal descriptivo simple, debido a que se buscó y recogió la información de forma directa, teniéndose como objetivo básicamente el de precisar y describir la manera adecuada de hacer tutoría en la Universidad Nacional San Cristóbal de Huamanga.

#### **3.4 POBLACIÓN Y MUESTRA**

##### **POBLACIÓN**

La población de estudio estuvo compuesto por todas las actividades de tutoría que se realizan en el Proyecto Hatun Ñan de la Universidad Nacional de San Cristóbal de Huamanga, 2016.

##### **MUESTRA**

Se realizó muestreo por juicio de expertos de las actividades reasignables al software web del Proyecto Hatun Ñan de la Universidad Nacional de San Cristóbal de Huamanga, 2016.



### **3.5 VARIABLES E INDICADORES**

#### **a. DEFINICIÓN CONCEPTUAL DE LAS VARIABLES**

##### **VARIABLE DE INTERÉS**

**Sistema de tutoría.** Es un acompañamiento y guía al estudiante de carácter personal, académico, social y profesional en su formación integral, el cual tiene por finalidad realizar una orientación sistemática a lo largo de toda la vida estudiantil para que el estudiante mejore su rendimiento académico, solucione sus dificultades universitarias y desarrolle una actitud proactiva con capacidad de reflexión crítica y convivencia social y se inserte con liderazgo al campo laboral (Directiva VRAC-UNSCH, 2013).

##### **VARIABLES DESCRIPTIVAS DE LA VARIABLE DE ESTUDIO**

**Actividades tutoriales.** Según el (VRAC-UNSCH), define como actividades tutoriales a la elaboración del perfil académico, familiar y social del tutorado, la elaboración del plan tutorial individual y grupal, el proceso de inducción al estudiante ingresante, la orientación en el proceso de matrícula, el acompañamiento en el área personal, académica, social y profesional; al planeamiento de estrategias tutoriales, a la evaluación del sistema tutorial y la presentación del informe de la acción tutorial.

**Proceso de seguimiento.** Correspondiente a las actividades por parte del tutor y personal administrativo orientadas a obtener información del tutorado(o los tutorados), con el objetivo de verificar que se cumplan los objetivos del sistema de tutoría.

#### **b. DEFINICIÓN OPERACIONAL DE LAS VARIABLES**

##### **PRIMERA VARIABLE DE INTERÉS**

X: Aplicación web

##### **VARIABLES DESCRIPTIVAS DE LA VARIABLE DE ESTUDIO**

X1: Análisis X2: Diseño

X3: Implementación X4: Pruebas

##### **SEGUNDA VARIABLE DE INTERÉS**

Y: Sistema de tutoría

##### **VARIABLES DESCRIPTIVAS DE LA VARIABLE DE ESTUDIO**

Y1: Actividades tutoriales Y2: Proceso de seguimiento

### **3.6 TÉCNICAS E INSTRUMENTOS TÉCNICAS**

**Entrevista.** Técnica utilizada para recopilar información que brindó el Director del Proyecto Hatun Ñan.

**Encuesta.** Técnica utilizada para recopilar información aplicada a los tutores.

## **INSTRUMENTOS**

Los instrumentos utilizados fueron:

- Guía de entrevista.
- Cuestionario para encuesta

Tabla 3.1

---

<b>Entrevista no estructurada</b>
1. Quiénes son los actores participantes del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga.
2. Qué actividades debe realizar el profesor en el Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga.
3. Qué funciones tiene el estudiante dentro del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga.
4. Qué actividades realiza el personal administrativo del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga.

---

<b>Encuesta de la Funcionalidad del Software</b>
1. ¿El software permite registrar a los tutores?
2. ¿El software permite modificar los datos de un tutor?
3. ¿El software permite registrar a los tutorados?
4. ¿El software permite modificar los datos de los tutorados?
5. ¿El software permite al administrador registrar los tutorados asignados a un tutor?
6. ¿El software permite el registro de la evaluación de objetivos del tutor?
7. ¿El software permite registrar los objetivos propuestos por el tutor para un determinado semestre académico?
8. ¿El software permite modificar los objetivos propuestos por el tutor para un determinado semestre académico?
9. ¿El software permite al tutor registrar sus actividades y talleres planteados para el semestre académico?

---

- 
10. ¿El software permite al tutor modificar sus actividades y talleres planteados para el semestre académico?
11. ¿El software permite al tutor registrar notas de la participación de los estudiantes en los respectivos talleres?
12. ¿El software permite al tutorado registrar sus cursos matriculados durante el semestre actual?
- 
- 

### **3.7 HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN**

Los datos obtenidos mediante la aplicación de los instrumentos antes mencionados fueron analizados y tratados con programas computarizados, tales como los aplicativos de Microsoft Office.

Tabla 3.2

Herramientas para el tratamiento de datos de las encuestas y/o entrevistas

<b>SOFTWARE</b>	<b>FABRICANTE</b>	<b>SERVICIO</b>
Microsoft Excel 2013	Desarrollado por Microsoft Corp.	Excel es un programa informático que permite realizar tareas contables y financieras gracias a sus funciones, desarrolladas específicamente para ayudar a crear y trabajar con hojas de cálculo.
Microsoft Word 2013	Desarrollado por Microsoft Corp.	Microsoft Word 2013 es un programa de procesamiento de textos diseñado para ayudarle a crear documentos de calidad profesional. Word le ayuda a organizar y escribir sus documentos de forma más eficaz.

---

En cuanto a las herramientas tecnológicas utilizadas para el desarrollo del software, en su totalidad son de software libre, y seleccionadas en función al objetivo que cumplieron en el proyecto.

### 3.8 TÉCNICAS PARA APLICAR XP

Tabla 3.3

Fase de exploración

<b>TAREA</b>	<b>ARTEFACTO</b>	<b>TÉCNICA</b>	<b>RESPONSABLE</b>
<b>Escribir historias de usuario</b>	Historia de usuario	Describir brevemente la historia de usuario con la regla del negocio (lo que el sistema debe hacer)	Cliente
		Dividir historia de usuario grandes	
<b>Probar las tecnologías a utilizar</b>	Arquitectura técnica inicial	Explorar posibilidades de uso de tecnologías	Cliente
		Probar el rendimiento de las tecnologías	Programador
		Definir las tecnologías a usar	Entrenador
<b>Estimar esfuerzo para historia de usuario</b>	Plan de alto nivel	Conocer previamente la historia de usuario	Programador
		Hacer una implementación rápida de historia de usuario	
		Estimar esfuerzo (semana) para desarrollar la historia de usuario	

Fuente: (Porras, 2010)

Tabla 3.4

Fase de planificación

<b>TAREA</b>	<b>ARTEFACTO</b>	<b>TÉCNICA</b>	<b>RESPONSABLE</b>
<b>Rescribir las historias de usuario</b>	Historia de usuario	Describir detalladamente la historia de usuario con la regla del negocio	Cliente
		Introducir nuevos requisitos del software.  Definir prioridad para cada historia de usuario por necesidad del negocio	Cliente
<b>Formular el plan de versiones</b>	Plan de versión (una iteración)	Utilizar técnicas de elaboración del plan de alto nivel  Estimar y asignar esfuerzo (semana) para cada historia de usuario en función a tiempo para planear, diseñar, implementar y probar.  Estimar y asignar riesgo a cada historia de usuario en función	Programador

---

a situación que afecta  
la estimación del  
esfuerzo.

Actualizar tarjeta de  
historia de usuario.

---

Fuente: (Porras, 2010)

Tabla 3.5

Fase de iteración

<b>TAREA</b>	<b>ARTEFACTO</b>	<b>TÉCNICA</b>	<b>RESPONSABLE</b>
<b>Definir la arquitectura técnica</b>	Arquitectura Técnica	<p>Actualizar la arquitectura técnica inicial.</p> <p>Usar características del negocio</p> <p>Utilizar arquitectura por capas</p> <p>Integrar frameworks.</p>	<p>Cliente</p> <p>Programador</p> <p>Entrenador</p>
<b>Escribir tareas de ingeniería</b>	Tarea de ingeniería	<p>Dividir cada historia de usuario en tareas, describir usando reglas del negocio cada tarea de ingeniera</p>	<p>Cliente</p> <p>programador</p>
		<p>Estimar y asignar esfuerzo para desarrollar una tarea de ingeniería</p>	<p>Programador</p>
		<p>Asignar una tarea de ingeniería al programador</p>	<p>Entrenador</p> <p>Programador</p>
<b>Formular el plan de iteraciones</b>	Plan de iteración	<p>Utilizar el plan de versión</p> <p>Actualizar el plan con tareas de</p>	

ingeniera de la siguiente iteración.  
 Actualizar el cuándo fallo prueba de aceptación.  
 Actualizar el plan con tareas no concluidas.  
 Actualizar las tarjetas de tarea de ingeniera

<b>Crear pruebas de aceptación</b>	Caso de prueba de aceptación	Escribir pruebas de aceptación para cada historia de usuario por iteración	Cliente Encargado de pruebas
<b>Implementar las interfaces</b>	GUI	Diseñar con precisión la GUI relacionada a cada historia de usuario Generar código para la interface usando herramienta	Cliente Programador
<b>Escribir tarjetas CRC</b>		Diseñar para una tarea de ingeniería de forma simple Rediseñar por falla	Cliente



<b>para cada tarea de ingeniería</b>	Tarjetas CRC	de pruebas de aceptación una tarea.	Programador
--------------------------------------	--------------	-------------------------------------	-------------

Identificar responsabilidades

Identificar colaboración

Identificar atributos

<b>Implementar la base de datos física</b>	Base de datos física	Escribir script usando tarjeta CRC. Ejecutar script usando DBMS	Programador
--	----------------------	--	-------------

<b>Implementar código para clase entidad</b>	Código fuente	Escribir código fuente o generar con una herramienta usando tarjetas CRC.	Programador
--	---------------	---	-------------

<b>Crear pruebas unitarias para las</b>	Prueba unitaria	Escribir código fuente para una prueba unitaria, usando una	programador
---	-----------------	---	-------------

<b>clases control</b>		herramienta	
-----------------------	--	-------------	--

<b>Implementar código fuente</b>	Código fuente	Codificar una tarea de ingeniería Hacer refactoring Mover	Programador Supervisor
----------------------------------	---------------	---	---------------------------

		programadores	
<b>Ejecutar pruebas unitarias</b>	Reporte de prueba unitaria	Ejecutar el módulo de cada prueba unitaria Modificar código fuente si la prueba unitaria muestra resultado incorrecto	Programador
<b>Realizar integración continua</b>	Código fuente	Integrar las tareas para una historia de usuario Mantener sistema integrador todo el tiempo	Programador
<b>Ejecutar pruebas de integración para una historia de usuario</b>	Reporte de pruebas de integración	Integrar continuamente al concluir las tareas de una historia de usuario Verificar que las pruebas de integración pasan al 100%	Programador
<b>Ejecutar pruebas de aceptación</b>	Reporte de pruebas de aceptación	Correr la última versión de una iteración Utilizar los casos de prueba de aceptación	Cliente Encargado de pruebas

## CAPÍTULO IV

### ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

#### 4.1 RESULTADOS DE LA APLICACIÓN DEL PROCESO AGIL XP

##### 4.1.1 FASE DE EXPLORACION

##### a. HISTORIA DE USUARIO

Tabla N° 4.1

Historia de usuario – Administrar tutor

---

#### HISTORIA DE USUARIO

---

**Número: 01**    **Nombre:** Administrar tutor

---

**Usuario:** Administrador

---

**Modificación de historia N°:** NA    **Iteración asignada:** 1

---

**Prioridad en negocio:** Alta    **Riesgo en desarrollo:** Medio

---

Descripción:

Como administrador registro, edito, visualizo y doy de baja los datos de los tutores; para poder tener información necesaria y actualizada para la distribución de tutorados para un determinado semestre académico.

**Observaciones:**

Los datos de los tutores se registran una vez los docentes son acreditados como tutores del proyecto.

Tabla N° 4.2

Historia de usuario – Administrar tutorado.

---

#### HISTORIA DE USUARIO

---

**Número: 02**    **Nombre:** Administrar tutorado

---

**Usuario:** Administrador

---

**Modificación de historia N°:** NA    **Iteración asignada:** 1

---

**Prioridad en negocio:** Alta    **Riesgo en desarrollo:** Alto

---

Descripción:

Como administrador registro, edito, visualizo y doy de baja los datos de los

---

---

alumnos aprobados como tutorados, con los cuales se llevará a cabo las labores de tutoría.

---

Observaciones:

El registro se realiza previa una evaluación del tutorado

---

### Tabla N° 4.3

#### Historia de usuario – Administrar objetivos del tutor.

---

HISTORIA DE USUARIO

---

Número: **03** Nombre: Administrar objetivos del tutor

---

Usuario:

**Tutor**

---

Modificación de historia N°: **NA** Iteración asignada: **2**

---

Prioridad en negocio: **Alta** Riesgo en desarrollo: **Medio**

---

Descripción:

Como tutor debo plantear y registrar los objetivos a cumplir mediante la tutoría, con la finalidad de poder evaluar el grado de cumplimiento de los mismos al que se llegó en el semestre académico.

---

Observaciones:

Ninguno.

---

### Tabla N° 4.4

#### Historia de usuario – Administrar actividades y talleres del tutor.

---

HISTORIA DE USUARIO

---

Número: **04** Nombre: Administrar actividades y talleres del tutor

---

Usuario:

**Tutor**

---

Modificación de historia N°: **NA** Iteración asignada: **2**

---

Prioridad en negocio: **Alta** Riesgo en desarrollo: **Alta**

---

Descripción:

Como Tutores del proyecto Hatun Ñan, coordinamos determinamos y configuramos las fechas de inicio y finalización de las actividades a realizarse durante un semestre

---

---

académico, acorde al cronograma pre establecido de la UNSCH. Con la finalidad de hacer cumplir el cronograma del semestre académico previamente establecido.

---

Observaciones:

Las actividades se registran previa coordinación y validación con los responsables del proyecto.

---

#### **Tabla N° 4.5**

##### **Historia de usuario – Asignación de tutorados.**

---

HISTORIA DE USUARIO

---

Número: **05**    **Nombre:** Asignación de tutorados

---

Usuario: **Alumnos, docentes, administrativos, egresados y grupos de interés**

---

Modificación de historia N°: **NA**    **Iteración asignada: 2**

---

Prioridad en negocio: **Alta**    **Riesgo en desarrollo: Alta**

---

Descripción:

Como usuarios (alumnos, docentes, administrativos, egresados y grupos de interés), queremos responder a las preguntas de las encuestas y cuestionarios en línea. Para poder tener una mayor participación de todos los actores.

---

Observaciones:

Los cuestionarios están únicamente diseñados para los docentes.

Las encuestas están diseñadas para todos los usuarios.

---

#### **Tabla N° 4.6**

##### **Historia de usuario – Registrar cursos del tutorado.**

---

HISTORIA DE USUARIO

---

Número: **06**    **Nombre:** Registrar cursos del tutorado

---

Usuario: **Tutorado**

---

Modificación de historia N°: **NA**    **Iteración asignada: 3**

---

Prioridad en negocio: **Alta**    **Riesgo en desarrollo: Alta**

---

<p>Descripción:</p> <p>Como tutorado debo poder registrar, editar y visualizar los cursos que llevo en un determinado semestre académico, con la finalidad de que esta información pueda facilitar realizar el seguimiento de los tutores de forma adecuada.</p>
<p>Observaciones:</p> <p>Los cursos se encuentran registrados en la base de datos de acuerdo al plan de estudios vigente de una escuela de formación profesional.</p>

**Tabla N° 4.7**

**Historia de usuario – Registrar notas del taller.**

HISTORIA DE USUARIO	
Número: <b>07</b>	Nombre: Registrar notas del taller
Usuario:	
<b>Tutor</b>	
Modificación de historia N°: <b>NA</b>	Iteración asignada: <b>3</b>
Prioridad en negocio: <b>Alta</b>	Riesgo en desarrollo: <b>Media</b>
Descripción:	
Como tutor debo registrar, actualizar y publicar las notas por taller. Todo ello con la finalidad de tener un control adecuado y el seguimiento de la evolución de los tutorados.	
Observaciones	
:	

**Tabla N° 4.8**

**Historia de usuario – Registrar evaluación de objetivos.**

HISTORIA DE USUARIO	
Número: <b>08</b>	Nombre: Registrar evaluación de objetivos
Usuario: <b>Comité interno</b>	
Modificación de historia N°: <b>NA</b>	Iteración asignada: <b>4</b>
Prioridad en negocio: <b>Alta</b>	Riesgo en desarrollo: <b>Alta</b>

<p>Descripción:</p> <p>Como miembros del directorio (administrador) evaluamos los objetivos planteados por cada uno de los tutores, de acuerdo a dicha evaluación damos por concluido, determinamos si se cumplió satisfactoriamente o con observaciones, con la finalidad de tener un historial de cumplimiento de objetivos y determinar el grado de cumplimiento de los docentes tutores.</p>
<p>Observaciones</p> <p>:</p> <p><b>Ninguno</b></p>

**Tabla N° 4.9**

**Historia de usuario – Administrar usuarios.**

HISTORIA DE USUARIO
Número: <b>09</b> <b>Nombre:</b> Administrar usuarios
Usuario: <b>Administrador</b>
Modificación de historia N°: <b>NA</b> <b>Iteración asignada:</b> 4
Prioridad en negocio: <b>Media</b> <b>Riesgo en desarrollo:</b> Media
<p>Descripción:</p> <p>Como administrador, podemos crear y editar sus cuentas de usuario para acceder al sistema. A si mismo pueden dar de baja a los usuarios que incurran en faltas que deriven en la alteración de resultados de la tutoría. Esto con la finalidad de poder tener un control adecuado.</p>
Observaciones:

**Tabla N° 4.10**

**Historia de usuario – Iniciar sesión.**

HISTORIA DE USUARIO
Número: <b>10</b> <b>Nombre:</b> Iniciar sesión
Usuario: <b>Administrador, Tutor, Tutorado</b>
Modificación de historia N°: <b>NA</b> <b>Iteración asignada:</b> 4

---

Prioridad en negocio: **Media**      **Riesgo en desarrollo:** Media

---

Descripción:

Como usuarios (administrador, tutor, tutorado), debemos ingresar al sistema mediante el uso de nuestra propia cuenta de usuario y contraseña; con la finalidad de poder tener un control adecuado de las operaciones que se realicen dentro del sistema y ser responsables de las mismas.

---

Observaciones:

Cada uno de los usuarios tendrá acceso a ciertas funcionalidades del software, dependiendo de las funciones y roles que estos tengan.

---

### 4.1.2 ARQUITECTURA TÉCNICA INICIAL



Figura N° 4.8: Arquitectura técnica inicial

### 4.1.3 PLAN DE ALTO NIVEL

Tabla N° 4.12 Plan de alto nivel

N°	HISTORIA DE USUARIO	ESFUERZO(días)
01	Administrar tutor	15
02	Administrar tutorado	11
03	Administrar objetivos del tutor	11
04	Administrar actividades y talleres del tutor	15
05	Asignación de tutorados	11



06	Registrar cursos del tutorado	11
07	Registrar notas del taller	14
08	Registrar evaluación de objetivos	11
09	Administrar usuarios	8
10	Iniciar sesión	5

Fuente: Elaboración propia.

## 4.2 FASE DE PLANEACIÓN

### 4.2.1 HISTORIA USUARIO POR PRIORIDAD

Tabla N° 4.13:

#### Historias de usuario por prioridad

N°	HISTORIA DE USUARIO	PRIORIDAD
01	Administrar tutor	ALTA
02	Administrar tutorado	ALTA
03	Administrar objetivos del tutor	ALTA
04	Administrar actividades y talleres del tutor	ALTA
05	Asignación de tutorados	MEDIA
06	Registrar cursos del tutorado	MEDIA
07	Registrar notas de taller	MEDIA
08	Registrar evaluación de objetivos	ALTA
09	Administrar usuarios	ALTA
10	Iniciar sesión	ALTA

### 4.2.2 ESTIMACIÓN DE ESFUERZO

Tabla N° 4.14: Estimación de esfuerzo

N°	HISTORIA DE USUARIO	TIEMPO ESTIMADO		
		SEMANAS	DÍAS ESTIMADOS	HORAS
		ESTIMAD		ESTIMAD
		AS		AS
01	Administrar tutor	3	15	60
02	Administrar tutorado	2.2	11	44

03	Administrar objetivos del tutor	2.2	11	44
04	Administrar actividades y talleres del tutor	3	15	60
05	Asignación de tutorados	2.2	11	44
06	Registrar cursos del tutorado	2.2	11	44
07	Administrar notas de taller	2.8	14	56
08	Registrar evaluación de objetivos	2.2	11	44
09	Administrar usuarios	1.6	8	32
10	Iniciar sesión	1	5	20
	Tiempo estimado total	<b>22.4</b>	<b>112</b>	<b>448</b>

### 4.2.3 PLAN DE ENTREGA

Tabla N° 4.15: Plan de entrega.

N° HISTORIA DE USUARIO	ESFUERZO DE DESARROLLO			ITERACIÓN ASIGNADA				ENTREGA ASIGNADA			
	SEMANAS	DÍAS	HORAS	1	2	3	4	V1	V2	V3	V4
	ESTIMADAS	ESTIMADOS	ESTIMADAS								
01 Administrar tutor	3	15	60	X				X			
02 Administrar tutorado	2.2	11	44	X				X			
03 Administrar objetivos del tutor	2.2	11	44		X				X		
04 Administrar actividades y talleres del tutor	3	15	60		X				X		
05 Asignación de tutorados	2.2	11	44		X				X		
06 Registrar cursos del tutorado	2.2	11	44			X				X	
07 Registrar notas del taller	2.8	14	56			X				X	
08 Registrar evaluación de objetivos	2.2	11	44				X			X	
09 Registrar usuarios	1.6	8	32				X			X	
10 Iniciar sesión	1	5	20				X			X	
	<b>Total semanas</b>			<b>5.2</b>	<b>7.4</b>	<b>5</b>	<b>4.8</b>			<b>22.4</b>	

### 4.3 FASE DE ITERACIÓN A LA PRIMERA VERSIÓN

#### 4.3.1 ARQUITECTURA TECNICA

##### DIAGRAMA DE COMPONENTES

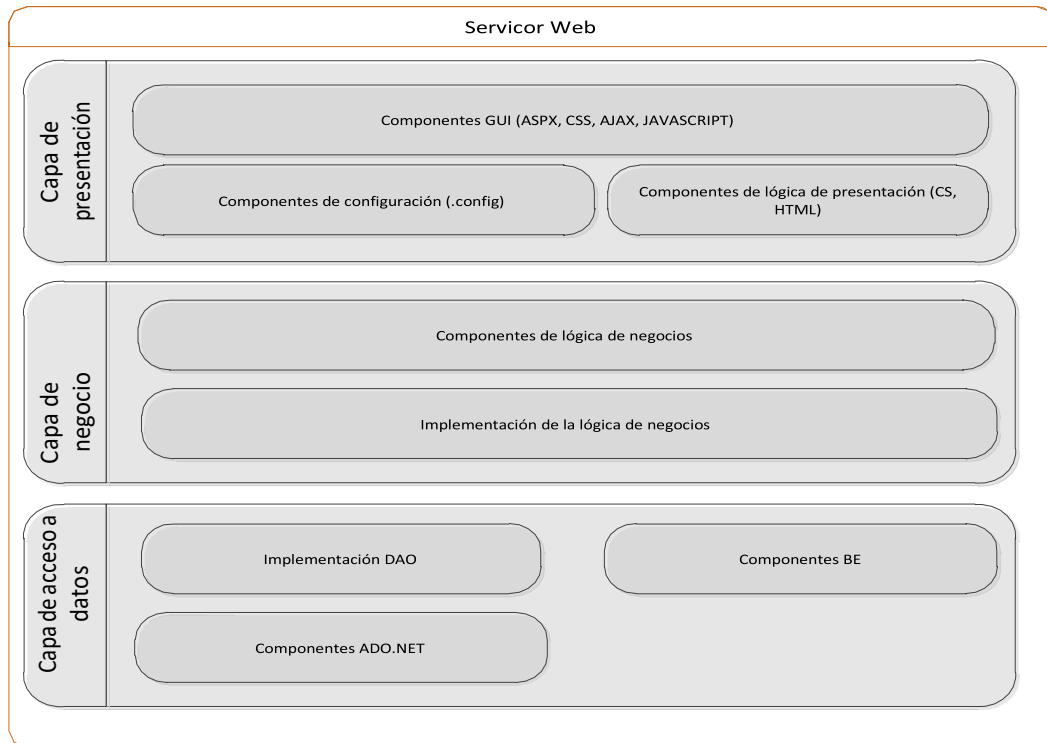


Figura N° 4.1

Diagrama de componentes Fuente: Elaboración propia

##### DIAGRAMA DE DESPLIEGUE

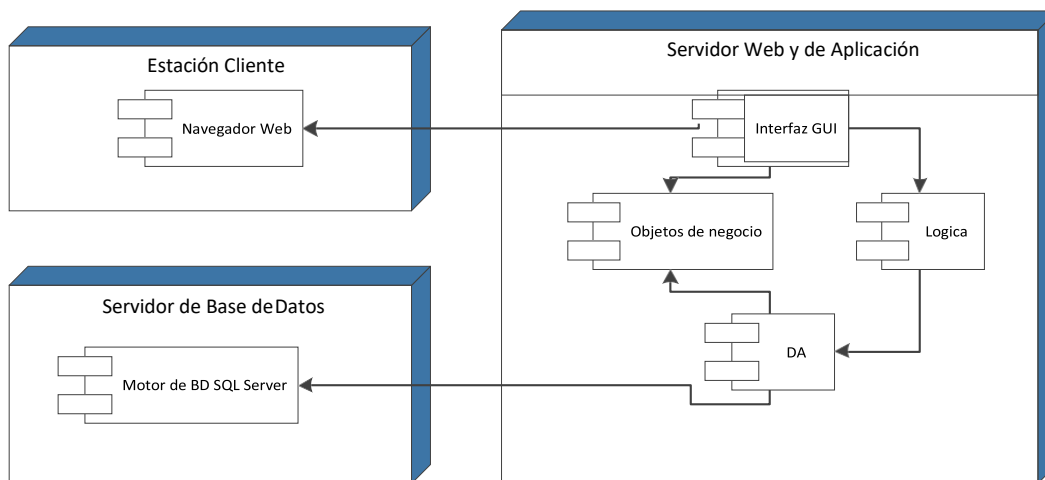


Figura N° 4.2

Diagrama de despliegue Fuente: Elaboración propia

### 4.3.2 TAREAS DE INGENIERÍA

#### TAREAS DE INGENIERÍA PARA LA PRIMERA ITERACIÓN

HU: Administrar tutor

##### Tabla N° 4.16

Tarea de ingeniería – Diseñar base de datos para administrar tutor.

---

TAREA DE INGENIERÍA	
Numero de tarea: <b>1</b>	Numero de historia de usuario: <b>1</b>
Nombre de tarea: <b>Diseñar base de datos para administrar tutor</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>05/04/2016</b>	Fecha fin: <b>06/04/2017</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Realizar el análisis para posteriormente crear las tablas de base de datos que permitan la administración de los datos de los tutores, teniendo en cuenta sus campos y relaciones con otras tablas.	

---

##### Tabla N° 4.17

Tarea de ingeniería – Diseñar las tarjetas CRC para administrar tutores.

---

TAREA DE INGENIERÍA	
Numero de tarea: <b>2</b>	Numero de historia de usuario: <b>1</b>
Nombre de tarea: <b>Diseñar las tarjetas CRC para administrar tutores</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.2</b>
Fecha de inicio: <b>07/04/2016</b>	Fecha fin: <b>07/04/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Realizar un análisis minucioso para identificar los objetos, sus atributos, colaboración y responsabilidades para finalmente ser traducidas en clases, métodos y relaciones.	

---

**Tabla N° 4.18****Tarea de ingeniería – Diseñar las interfaces para administrar tutores.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>3</b>	Numero de historia de usuario: <b>1</b>
Nombre de tarea: <b>Diseñar las interfaces para administrar tutores</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>08/04/2016</b>	Fecha fin: <b>11/04/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Diseñar las interfaces graficas de usuario (GUI), necesarias que permita registrar, actualizar, listar, buscar y eliminar; los datos de los tutores; todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.	

**Tabla N° 4.19****Tarea de ingeniería – Registrar tutor.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>4</b>	Numero de historia de usuario: <b>1</b>
Nombre de tarea: <b>Registrar tutor</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: <b>0.8</b>
Fecha de inicio: <b>12/04/2016</b>	Fecha fin: <b>15/04/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El sistema muestra la página mantener tutor con los controles deshabilitados y la grilla donde se visualizan los tutores registrados.	
El Usuario hace clic en el botón nuevo y automáticamente se habilitan los controles para su inserción.	
El Usuario ingresa los campos necesarios para insertar y a continuación hace clic en el botón guardar.	
El sistema valida la información ingresada por el usuario y a continuación muestra mensaje de confirmación de registro.	

**Tabla N° 4.20**

### Tarea de ingeniería – Mantener tutor.

---

#### TAREA DE INGENIERÍA

---

Numero de tarea: **5**

**Numero de historia de usuario: 1**

---

Nombre de tarea: **Mantener tutor**

---

Tipo de tarea: **Desarrollo**

**Puntos estimados: 0.8**

---

Fecha de inicio: **18/04/2016**

**Fecha fin: 21/04/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

#### Descripción:

El sistema muestra la página mantener tutor con los controles deshabilitados y la grilla donde se visualizan los tutores registrados.

El usuario hace clic en el icono mantener de la grilla y automáticamente se visualizan los datos en los controles.

El usuario modifica los datos del tutor y luego hace clic en Actualizar.

El sistema valida la información ingresada por el usuario y a continuación muestra un mensaje de confirmación de la actualización.

---

### Tabla N° 4.21

### Tarea de ingeniería – Realizar pruebas de aceptación.

---

#### TAREA DE INGENIERÍA

---

Numero de tarea: **6**

**Numero de historia de usuario: 1**

---

Nombre de tarea: **Realizar pruebas de aceptación**

---

Tipo de tarea: **Prueba**

**Puntos estimados: 0.4**

---

Fecha de inicio: **22/04/2016**

**Fecha fin: 25/04/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

#### Descripción:

El cliente junto al encargado de pruebas realizaron las pruebas de aceptación sobre las funcionalidades involucradas en la historia de usuario, para garantizar el correcto funcionamiento del registro, actualización, búsqueda y listado de los datos de los tutores.

---

**HU:** Administrar tutorado

**Tabla N° 4.22**

**Tarea de ingeniería – Diseñar base de datos para administrar tutorado.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: <b>7</b>	Numero de historia de usuario: <b>2</b>
---------------------------	---

---

Nombre de tarea: **Diseñar base de datos para administrar tutorado**

---

Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.4</b>
------------------------------	------------------------------

---

Fecha de inicio: <b>26/04/2016</b>	Fecha fin: <b>27/04/2016</b>
------------------------------------	------------------------------

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar el análisis para posteriormente crear las tablas de base de datos que permitan la administración de los datos de los tutorados, teniendo en cuenta sus campos y relaciones con otras tablas.

**Tabla N° 4.23**

**Tarea de ingeniería – Diseñar las tarjetas CRC para administrar tutorados.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: <b>8</b>	Numero de historia de usuario: <b>2</b>
---------------------------	---

---

Nombre de tarea: **Diseñar las tarjetas CRC para administrar tutorados**

---

Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.2</b>
------------------------------	------------------------------

---

Fecha de inicio: <b>28/04/2016</b>	Fecha fin: <b>28/04/2016</b>
------------------------------------	------------------------------

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar un análisis minucioso para identificar los objetos, sus atributos, colaboración y responsabilidades involucrados en la administración de tutorados, para que finalmente estos puedan ser traducidas en clases, métodos y relaciones.

**Tabla N° 4.24**

**Tarea de ingeniería – Diseñar las interfaces para administrar tutorados.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: <b>9</b>	Numero de historia de usuario: <b>2</b>
---------------------------	---

---

Nombre de tarea: **Diseñar las interfaces para administrar tutorados**

---



Tipo de tarea: <b>Diseño</b>	<b>Puntos estimados: 0.6</b>
Fecha de inicio: <b>29/04/2016</b>	<b>Fecha fin: 03/05/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Diseñar las interfaces graficas de usuario (GUI), necesarias que permita registrar, actualizar, listar, buscar y eliminar; los datos de los tutorados; quienes se inscriben en las diversas actividades del proyecto, todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.	

**Tabla N° 4.25**

**Tarea de ingeniería – Registrar tutorado.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>10</b>	<b>Numero de historia de usuario: 2</b>
Nombre de tarea: <b>Registrar tutorado</b>	
Tipo de tarea: <b>Desarrollo</b>	<b>Puntos estimados: 0.4</b>
Fecha de inicio: <b>04/05/2016</b>	<b>Fecha fin: 05/05/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El sistema muestra la página mantener tutorado con los controles deshabilitados y la grilla donde se visualizan los tutorados registrados.	
El Usuario hace clic en el botón nuevo y automáticamente se habilitan los controles para su inserción.	
El Usuario ingresa los campos necesarios para insertar y a continuación hace clic en el botón guardar.	
El sistema valida la información ingresada por el usuario y a continuación muestra un _mensaje de confirmación de registro.	

**Tabla N° 4.26**

**Tarea de ingeniería – Mantener tutorado.**

TAREA DE INGENIERÍA
---------------------

Numero de tarea: <b>11</b>	Numero de historia de usuario: <b>2</b>
Nombre de tarea: <b>Mantener tutorado</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>06/05/2016</b>	Fecha fin: <b>09/05/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El sistema muestra la página mantener tutorado con los controles deshabilitados y la grilla donde se visualizan los tutores registrados.	
El usuario hace clic en el icono actualizar de la grilla y automáticamente se visualizan los datos en los controles.	
El usuario modifica los datos del tutorado y luego hace clic en Actualizar.	
El sistema valida la información ingresada por el usuario y a continuación muestra un mensaje de confirmación de la actualización.	

**Tabla N° 4.27**

**Tarea de ingeniería – Realizar pruebas de aceptación.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>12</b>	Numero de historia de usuario: <b>2</b>
Nombre de tarea: <b>Realizar pruebas de aceptación</b>	
Tipo de tarea: <b>Prueba</b>	Puntos estimados: <b>0.2</b>
Fecha de inicio: <b>10/05/2016</b>	Fecha fin: <b>10/05/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El cliente junto al encargado de pruebas realizaron las pruebas de aceptación sobre las funcionalidades involucradas en la historia de usuario, para garantizar el correcto funcionamiento del registro, actualización, búsqueda y listado de los <u>datos de los tutorados.</u>	

**TAREAS DE INGENIERÍA PARA LA SEGUNDA ITERACIÓN**

**HU:** Administrar objetivos del tutor

**Tabla N° 4.28**

**Tarea de ingeniería – Diseñar base de datos para administrar objetivos del tutor.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: <b>13</b>	Numero de historia de usuario: <b>3</b>
----------------------------	---

---

Nombre de tarea: **Diseñar base de datos para administrar objetivos del tutor**

---

Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.4</b>
------------------------------	------------------------------

---

Fecha de inicio: <b>11/05/2016</b>	Fecha fin: <b>12/05/2016</b>
------------------------------------	------------------------------

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar el análisis para posteriormente crear las tablas de base de datos que permitan la administración de los objetivos trazados por el tutor, teniendo en cuenta sus campos y relaciones con otras tablas.

**Tabla N° 4.29**

**Tarea de ingeniería – Diseñar las tarjetas CRC para administrar objetivos.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: <b>14</b>	Numero de historia de usuario: <b>3</b>
----------------------------	---

---

Nombre de tarea: **Diseñar las tarjetas CRC para administrar objetivos**

---

Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.2</b>
------------------------------	------------------------------

---

Fecha de inicio: <b>13/05/2016</b>	Fecha fin: <b>13/05/2016</b>
------------------------------------	------------------------------

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar un análisis minucioso para identificar los objetos, sus atributos, colaboración y responsabilidades involucrados en la administración de objetivos del tutor, para que finalmente estos puedan ser traducidos en clases, métodos y relaciones.

**Tabla N° 4.30**

**Tarea de ingeniería – Diseñar las interfaces para administrar objetivos.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: <b>15</b>	Numero de historia de usuario: <b>3</b>
Nombre de tarea: <b>Diseñar las interfaces para administrar objetivos</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.6</b>
Fecha de inicio: <b>16/05/2016</b>	Fecha fin: <b>18/05/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Diseñar las interfaces graficas de usuario (GUI), necesarias que permita registrar, actualizar, listar, buscar y eliminar; los objetivos que se traza cada tutor, todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.	

**Tabla N° 4.31**

**Tarea de ingeniería – Registrar objetivos.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>16</b>	Numero de historia de usuario: <b>3</b>
Nombre de tarea: <b>Registrar objetivos del tutor</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>19/05/2016</b>	Fecha fin: <b>20/05/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El sistema muestra la página Administrar objetivo con los controles deshabilitados y la grilla donde se visualizan los objetivos con los que cuenta actualmente.	
El Usuario hace clic en el botón nuevo y automáticamente se habilitan los controles para su inserción.	
El Usuario ingresa los campos necesarios para insertar y a continuación hace clic en el botón guardar.	
El sistema valida la información ingresada por el usuario y a continuación muestra un mensaje de confirmación de registro.	

**Tabla N° 4.32**

**Tarea de ingeniería – Mantener objetivos.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>17</b>	Numero de historia de usuario: <b>3</b>

Nombre de tarea: <b>Mantener objetivos del tutor</b>	
Tipo de tarea: <b>Desarrollo</b>	<b>Puntos estimados:</b> 0.4
Fecha de inicio: <b>23/05/2016</b>	<b>Fecha fin:</b> 24/05/2016
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El sistema muestra la página administrar objetivo con los controles deshabilitados y la grilla donde se visualizan los objetivos registrados.	
El usuario hace clic en el icono mantener de la grilla y automáticamente se visualizan los datos en los controles correspondientes.	
El usuario modifica los datos del objetivo y luego hace clic en Actualizar.	
El sistema valida la información ingresada por el usuario y a continuación muestra un mensaje de confirmación de la actualización.	

**Tabla N° 4.33**

**Tarea de ingeniería – Realizar pruebas de aceptación.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>18</b>	<b>Numero de historia de usuario:</b> 3
Nombre de tarea: <b>Realizar pruebas de aceptación</b>	
Tipo de tarea: <b>Prueba</b>	<b>Puntos estimados:</b>
Fecha de inicio: <b>25/05/2016</b>	<b>Fecha fin:</b> 25/05/2016
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El cliente junto al encargado de pruebas realizaron las pruebas de aceptación sobre las funcionalidades involucradas en el registro, actualización, eliminación y listado de los objetivos del tutor.	

**HU:** Administrar actividades y talleres del tutor

**Tabla N° 4.34**

**Tarea de ingeniería – Diseñar base de datos para administrar actividades y talleres del tutor.**

TAREA DE INGENIERÍA
---------------------

Numero de tarea: <b>19</b>	Numero de historia de usuario: <b>4</b>
<b>señar base de datos para administrar actividades y talleres del tutor</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.6</b>
Fecha de inicio: <b>26/05/2016</b>	Fecha fin: <b>30/05/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Realizar el análisis para posteriormente crear las tablas de base de datos que permitan la administración de las actividades y talleres del tutor, teniendo en cuenta sus campos y relaciones con otras tablas.	

#### **Tabla N° 4.35**

#### **Tarea de ingeniería – Diseñar las tarjetas CRC para administrar actividades y talleres del tutor.**

<b>TAREA DE INGENIERÍA</b>	
Numero de tarea: <b>20</b>	Numero de historia de usuario: <b>4</b>
<b>diseñar las tarjetas CRC para administrar actividades y talleres del tutor</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>31/05/2016</b>	Fecha fin: <b>01/06/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Realizar un análisis minucioso para identificar los objetos, sus atributos, colaboración y responsabilidades involucrados en la administración de actividades y talleres del tutor, para que finalmente estos puedan ser traducidos en clases, métodos y relaciones	

#### **Tabla N° 4.36**

#### **Tarea de ingeniería – Diseñar las interfaces para administrar actividades y talleres del tutor.**

---

**TAREA DE INGENIERÍA**

---

**Numero de tarea: 21****Numero de historia de usuario: 4**

---

**Diseñar las interfaces para administrar actividades y talleres del tutor**

---

**Tipo de tarea: Diseño****Puntos estimados: 0.6**

---

**Fecha de inicio: 02/06/2016****Fecha fin: 06/06/2016**

---

**Programador responsable: Ayde Quispe Gómez**

---

Descripción:

Diseñar las interfaces graficas de usuario (GUI), necesarias que permita registrar, actualizar, listar, buscar y eliminar; las actividades y talleres que realiza cada tutor, todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.

---

**Tabla N° 4.37****Tarea de ingeniería – Registrar actividades y talleres.**

---

**TAREA DE INGENIERÍA**

---

**Numero de tarea: 22****Numero de historia de usuario: 4**

---

**Nombre de tarea: Registrar actividades y talleres del tutor**

---

**Tipo de tarea: Desarrollo****Puntos estimados: 0.6**

---

**Fecha de inicio: 07/06/2016****Fecha fin: 09/06/2016**

---

**Programador responsable: Ayde Quispe Gómez**

---

Descripción:

El sistema muestra la página administrar actividad y administrar taller con los controles deshabilitados y la grilla donde se visualizan los actividades y talleres registrados.

El Usuario hace clic en el botón nuevo y automáticamente se habilitan los controles para su registro.

El Usuario ingresa los campos necesarios para registrar y a continuación hace clic en el botón guardar.

El sistema valida la información ingresada por el usuario y a continuación muestra un mensaje de confirmación de registro.

**Tabla N° 4.38**

**Tarea de ingeniería – Mantener actividades y talleres.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>23</b>	Numero de historia de usuario: <b>4</b>
Nombre de tarea: <b>Mantener actividades y talleres del tutor</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>10/06/2016</b>	Fecha fin: <b>13/06/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
<p>El sistema muestra la página administrar actividades y talleres con los controles deshabilitados y la grilla donde se visualizan las actividades y talleres registrados. El usuario hace clic en el icono mantener de la grilla y automáticamente se visualizan los datos en los controles correspondientes.</p> <p>El usuario modifica los datos de las actividades y talleres y luego hace clic en Actualizar. El sistema valida la información ingresada por el usuario y a continuación muestra un mensaje de confirmación de la actualización.</p>	

**Tabla N° 4.39**

**Tarea de ingeniería – Realizar pruebas de aceptación.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>24</b>	Numero de historia de usuario: <b>4</b>
Nombre de tarea: <b>Realizar pruebas de aceptación</b>	
Tipo de tarea: <b>Prueba</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>14/06/2016</b>	Fecha fin: <b>15/06/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	



---

Descripción:

El cliente junto al encargado de pruebas realizaron las pruebas de aceptación sobre las funcionalidades involucradas en el registro, actualización, eliminación y listado de las actividades y talleres del tutor.

---

**HU:** Asignación de tutorados

**Tabla N° 4.40**

**Tarea de ingeniería – Diseñar base de datos para asignar tutorados**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **25**

Numero de historia de usuario: **5**

---

Nombre de tarea: **Diseñar base de datos para asignar tutorados**

---

Tipo de tarea: **Diseño**

Puntos estimados: **0.6**

---

Fecha de inicio: **16/06/2016**

Fecha fin: **20/06/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar el análisis para posteriormente crear las tablas de base de datos que permitan la de tutorados a un tutor, teniendo en cuenta sus campos y relaciones con otras tablas.

**Tabla N° 4.41**

**Tarea de ingeniería – Diseñar las tarjetas CRC para asignar tutorados**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **26**

Numero de historia de usuario: **5**

---

Nombre de tarea: **Diseñar las tarjetas CRC para asignar tutorados**

---

Tipo de tarea: **Diseño**

Puntos estimados: **0.2**

---

Fecha de inicio: **21/06/2016**

Fecha fin: **21/06/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

---

Descripción:

Realizar un análisis minucioso para identificar los objetos, sus atributos, colaboración y responsabilidades involucrados en de los tutorados, para que finalmente estos puedan ser traducidos en clases, métodos y relaciones.

**Tabla N° 4.42**

**Tarea de ingeniería – Diseñar las interfaces para asignar tutorados.**

---

TAREA DE INGENIERÍA	
Numero de tarea: <b>27</b>	Numero de historia de usuario:
Nombre de tarea: <b>Diseñar las interfaces para asignar tutorados</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: 0.6
Fecha de inicio: <b>22/06/2016</b>	Fecha fin: 24/06/2016
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Diseñar las interfaces graficas de usuario (GUI), necesarias que permita la asignación de los tutorados a su respectivo tutor, todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.	

**Tabla N° 4.43**

**Tarea de ingeniería – Registrar tutorados.**

---

TAREA DE INGENIERÍA	
Numero de tarea: <b>28</b>	Numero de historia de usuario: 5
Nombre de tarea: <b>Registrar asignación de tutorados</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: 0.6
Fecha de inicio: <b>27/06/2016</b>	Fecha fin: 29/06/2016
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	

El sistema muestra la página asignar tutorado con los controles deshabilitados y la grilla donde se visualizan los tutores asignados para el tutor.

El Usuario hace clic en el botón nuevo y automáticamente se habilitan los controles para la asignación del tutorado.

El Usuario hace la búsqueda del tutorado a ser asignado al tutor e ingresa algunos campos necesarios para registrar y a continuación hace clic en el botón guardar.

El sistema valida la información ingresada por el usuario y a continuación muestra un mensaje de confirmación de registro.

### **Tarea de ingeniería – Realizar pruebas de aceptación.**

---

#### TAREA DE INGENIERÍA

---

Numero de tarea: **29**

Numero de historia de usuario: **5**

Nombre de tarea: **Realizar pruebas de aceptación**

Tipo de tarea: **Prueba**

Puntos estimados: **0.2**

Fecha de inicio: **30/06/2016**

Fecha fin: **30/06/2016**

Programador responsable: **Ayde Quispe Gómez**

Descripción:

El cliente junto al encargado de pruebas realizaron las pruebas de aceptación sobre las funcionalidades involucradas en la asignación de los tutorados para su respectivo tutor.

### **TAREAS DE INGENIERÍA PARA LA TERCERA ITERACIÓN**

**HU:** Registrar cursos del tutorado

#### **Tabla N° 4.45**

### **Tarea de ingeniería – Diseñar base de datos para registrar cursos del tutorado**

---

#### TAREA DE INGENIERÍA

---

Numero de tarea: **30**

Numero de historia de usuario: **6**

Nombre de tarea: **Diseñar base de datos para registrar cursos del tutorado**

Tipo de tarea: **Diseño**

Puntos estimados: **0.4**

Fecha de inicio: **01/07/2016**

Fecha fin: **04/07/2016**

---

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar el análisis para posteriormente crear las tablas de base de datos que permitan el registro de cursos del tutorado para un determinado semestre académico, teniendo en cuenta sus campos y relaciones con otras tablas.

---

#### **Tabla N° 4.46**

##### **Tarea de ingeniería – Diseñar las tarjetas CRC para registrar cursos**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **31**

Numero de historia de usuario: **6**

Nombre de tarea: **Diseñar las tarjetas CRC para registrar cursos del tutorado**

Tipo de tarea: **Diseño**

Puntos estimados: **0.2**

Fecha de inicio: **05/07/2016**

Fecha fin: **05/07/2016**

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar un análisis minucioso para identificar los objetos, sus atributos, colaboración y responsabilidades involucrados en el proceso de registro de los cursos para un determinado semestre académico, para que finalmente estos puedan ser traducidos en clases, métodos y relaciones.

#### **Tabla N° 4.47**

##### **Tarea de ingeniería – Diseñar las interfaces para registrar cursos.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **32**

Numero de historia de usuario: **6**

Nombre de tarea: **Diseñar las interfaces para registrar cursos del tutorado**

Tipo de tarea: **Diseño**

Puntos estimados: **0.6**

Fecha de inicio: **06/07/2016**

Fecha fin: **08/07/2016**

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Diseñar las interfaces graficas de usuario (GUI), necesarias que permita el registro

---

de todos los cursos del tutorado en el semestre académico, todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.

**Tabla N° 4.48**

**Tarea de ingeniería – Registrar cursos.**

---

TAREA DE INGENIERÍA	
Numero de tarea: <b>33</b>	Numero de historia de usuario: <b>6</b>
Nombre de tarea: <b>Registrar cursos del tutorado</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>11/07/2016</b>	Fecha fin: <b>12/07/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El sistema muestra la página registrar curso con los controles deshabilitados y la grilla donde se visualizan los cursos registrados para el tutorado.	
El Usuario hace clic en el botón nuevo y automáticamente se habilitan los controles para la el registro del curso.	
El Usuario hace la búsqueda del curso mediante la sigla, esta debe ser retornada de la base de datos con todos los datos que le corresponda como el plan de estudios, sus requisitos, el número de créditos y a que ciclo pertenece.	
Una vez cargada la data y llenada los capos el tutorado hace clic en el botón guardar. El sistema valida la información ingresada por el usuario y a continuación muestra un mensaje de confirmación de registro.	

**Tabla N° 4.49**

**Tarea de ingeniería – Mantener cursos.**

---

TAREA DE INGENIERÍA	
Numero de tarea: <b>34</b>	Numero de historia de usuario: <b>6</b>
Nombre de tarea: <b>Mantener cursos del tutorado</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>13/07/2016</b>	Fecha fin: <b>14/07/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	

---

---

Descripción:

El sistema muestra la página registrar curso con los controles deshabilitados y la grilla donde se visualizan los cursos registrados para el tutorado.

El Usuario hace clic en el icono de edición de datos de los cursos registrados, el sistema le muestra los datos del curso en sus respectivos campos con los controles habilitados.

El Usuario hace las actualizaciones que este estime por conveniente y hace clic en el botón Actualizar, el sistema valida la información, registra y muestra un mensaje de confirmación de la actualización realizada, y muestra dicha información en la tabla de cursos registrados.

**Tabla N° 4.50**

**Tarea de ingeniería – Realizar pruebas de aceptación.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **35**

**Numero de historia de usuario: 6**

---

Nombre de tarea: **Realizar pruebas de aceptación**

---

Fecha de inicio: **15/07/2016**

**Fecha fin: 15/07/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

El cliente junto al encargado de pruebas realizaron las pruebas de aceptación sobre las funcionalidades involucradas en el registro de cursos del tutorado.

**HU:** Registrar notas del taller

**Tabla N° 4.51**

**Tarea de ingeniería – Diseñar base de datos para administrar notas del taller**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **36**

**Numero de historia de usuario: 7**

---

Nombre de tarea: **Diseñar base de datos para administrar notas del taller**

---

Tipo de tarea: **Diseño**

**Puntos estimados: 0.4**

---

Fecha de inicio: **18/07/2016**

**Fecha fin: 19/07/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

---

Descripción:

Realizar el análisis para posteriormente crear las tablas de base de datos que permitan el registro de las notas de los talleres a los cuales un tutorado se encuentra registrado, teniendo en cuenta sus campos y relaciones con otras tablas.

**Tabla N° 4.52**

**Tarea de ingeniería – Diseñar las tarjetas CRC para registrar notas**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **37**

**Numero de historia de usuario: 7**

---

Nombre de tarea: **Diseñar las tarjetas CRC para registrar notas de taller**

---

Tipo de tarea: **Diseño**

**Puntos estimados: 0.6**

---

Fecha de inicio: **20/07/2016**

**Fecha fin: 22/07/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar un análisis minucioso para identificar los objetos, sus atributos, colaboración y responsabilidades involucrados en el proceso de registro de las notas de los talleres del tutorado por parte del tutor, para que finalmente estos puedan ser traducidos en clases, métodos y relaciones.

**Tabla N° 4.53**

**Tarea de ingeniería – Diseñar las interfaces para registrar notas.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **38**

**Numero de historia de usuario: 7**

---

Nombre de tarea: **Diseñar las interfaces para registrar notas de taller**

---

Tipo de tarea: **Diseño**

**Puntos estimados: 0.6**

---

Fecha de inicio: **25/07/2016**

**Fecha fin: 27/07/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Diseñar las interfaces graficas de usuario (GUI), necesarias que permita el registro y la visualización de todas las notas de los talleres para cada uno de los registrados a dicho taller, todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.

**Tabla N° 4.54****Tarea de ingeniería – Registrar notas.**

---

TAREA DE INGENIERÍA	
Numero de tarea: <b>39</b>	Numero de historia de usuario: <b>7</b>
Nombre de tarea: <b>Registrar notas de talleres</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: <b>0.6</b>
Fecha de inicio: <b>28/07/2016</b>	Fecha fin: <b>01/08/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
El sistema muestra la página nota taller donde el tutor realiza la búsqueda de un taller por denominación, el sistema muestra el taller con todos los inscritos en una lista.	
El tutor selecciona un tutorado al cual se le asignará su nota de acuerdo a su participación, el sistema habilita los campos para asignarle una nota y una observación. El tutor ingresa la nota y su respectiva observación (opcional) y hace clic en el botón grabar, el sistema confirma el registro mediante un mensaje emergente y muestra la nota asignada en la lista de todos los participantes de dicho taller.	

**Tabla N° 4.55****Tarea de ingeniería – Mantener notas.**

---

TAREA DE INGENIERÍA	
Numero de tarea: <b>40</b>	Numero de historia de usuario: <b>7</b>
Nombre de tarea: <b>Mantener notas de talleres</b>	
Tipo de tarea: <b>Desarrollo</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>02/08/2016</b>	Fecha fin: <b>03/08/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	

---



---

Descripción:

El sistema muestra la página nota taller donde el tutor realiza la búsqueda de un taller por denominación, el sistema muestra el taller con todos los inscritos en una lista con sus respectivas notas.

El tutor selecciona un tutorado al cual se le asignará o actualizará su nota de acuerdo a su participación, el sistema habilita los campos para actualizar su nota y una observación. El tutor ingresa la nota y su respectiva observación (opcional) y hace clic en el botón grabar, el sistema confirma la actualización mediante un mensaje emergente y muestra la nota actualizada en la lista de todos los participantes de dicho taller.

**]Tabla N° 4.56**

**Tarea de ingeniería – Realizar pruebas de aceptación.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **41**

**Numero de historia de usuario: 7**

---

Nombre de tarea: **Realizar pruebas de aceptación**

---

Tipo de tarea: **Prueba**

**Puntos estimados: 0.2**

---

Fecha de inicio: **04/08/2016**

**Fecha fin: 04/08/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

El cliente junto al encargado de pruebas realizaron las pruebas de aceptación sobre las funcionalidades involucradas en el registro de notas de los talleres del tutorado.

**TAREAS DE INGENIERÍA PARA LA CUARTA ITERACIÓN**

**HU:** Registrar evaluación de objetivos

**Tabla N° 4.57**

**Tarea de ingeniería – Diseñar base de datos para registrar evaluar objetivos.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **42**

**Numero de historia de usuario: 8**

---

Nombre de tarea: **Diseñar base de datos para registrar evaluación de objetivos**

---

Tipo de tarea: **Diseño**

**Puntos estimados: 0.6**

Fecha de inicio: <b>05/08/2016</b>	Fecha fin: 09/08/2016
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Realizar el análisis para posteriormente crear las tablas de base de datos que permitan el registro de las de la evaluación de los objetivos de los tutores, teniendo en cuenta sus campos y relaciones con otras tablas.	

#### **Tabla N° 4.58**

#### **Tarea de ingeniería – Diseñar las tarjetas CRC para registrar evaluación de objetivos**

TAREA DE INGENIERÍA	
Numero de tarea: <b>43</b>	Numero de historia de usuario: 8
<b>Diseñar las tarjetas CRC para registrar evaluación de objetivos</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: 0.6
Fecha de inicio: <b>10/08/2016</b>	Fecha fin: 12/08/2016
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción:	
Realizar un análisis minucioso para identificar los objetos, sus atributos, colaboración y responsabilidades involucrados en el proceso de registro de la evaluación de los objetivos de los tutores, para que finalmente estos puedan ser traducidos en clases, métodos y relaciones.	

#### **Tabla N° 4.59**

#### **Tarea de ingeniería – Diseñar las interfaces para registrar evaluación de objetivos.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>44</b>	Numero de historia de usuario: 8
Nombre de tarea: <b>Diseñar las interfaces para registrar evaluación de objetivos</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: 0.4
Fecha de inicio: <b>15/08/2016</b>	Fecha fin: 16/08/2016

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Diseñar las interfaces graficas de usuario (GUI), necesarias que permita el registro de la calificación para los objetivos y la visualización de todas las calificaciones que obtuvieron los objetivos de los tutores, todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.

#### **Tabla N° 4.60**

##### **Tarea de ingeniería – Registrar evaluación de objetivos.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **45**

**Número de historia de usuario: 8**

---

Nombre de tarea: **Registrar evaluación de objetivos del tutor**

---

Tipo de tarea: **Desarrollo**

**Puntos estimados: 0.4**

---

Fecha de inicio: **17/08/2016**

**Fecha fin: 18/08/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

El administrador ingresa a la página evaluación, el sistema muestra la página con la información de los objetivos planteados por los tutores que estén concluidos.

El administrador del sistema selecciona el objetivo a evaluar, el sistema muestra la información de dicho objetivo en sus campos correspondientes.

El administrador le asigna una calificación de acuerdo al logro obtenido para dicho objetivo, siendo las calificaciones; concluido, satisfactorio, y observado, el sistema valida la información y muestra un mensaje de confirmación de la calificación realizada.

#### **Tabla N° 4.61**

##### **Tarea de ingeniería – Realizar pruebas de aceptación.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **46**

**Numero de historia de usuario: 8**

---

Nombre de tarea: **Realizar pruebas de aceptación**

---

Tipo de tarea: **Prueba**

**Puntos estimados:**

---

Fecha de inicio: **19/08/2016**

**Fecha fin: 19/08/2016**

---

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

El cliente junto al encargado de pruebas realizaron las pruebas de aceptación sobre las funcionalidades involucradas en el registro de evaluación de objetivos trazados para un ciclo determinado.

**HU:** Administrar usuarios

**Tabla N° 4.62**

**Tarea de ingeniería – Diseñar base de datos para registrar cuenta de usuario.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **47**

Numero de historia de usuario: **9**

Nombre de tarea: **Diseñar base de datos para registrar cuenta de usuario**

Tipo de tarea: **Diseño**

Puntos estimados: **0.4**

Fecha de inicio: **22/08/2016**

Fecha fin: **23/08/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar el análisis para posteriormente crear las tablas de base de datos que permita registrar usuarios y sus respectivas cuentas, teniendo en cuenta sus campos y relaciones con otras tablas.

**Tabla N° 4.63**

**Tarea de ingeniería – Diseñar las tarjetas CRC para registrar cuenta de usuario.**

---

TAREA DE INGENIERÍA

---

Numero de tarea: **48**

Numero de historia de usuario: **9**

Nombre de tarea: **Diseñar las tarjetas CRC para registrar cuenta de usuario**

Tipo de tarea: **Diseño**

Puntos estimados: **0.4**

Fecha de inicio: **24/08/2016**

Fecha fin: **25/08/2016**

---

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Realizar el análisis para identificar los objetos, sus atributos, colaboración y responsabilidades para finalmente ser traducidas en clases, métodos y relaciones.

**Tabla N° 4.64**

### Tarea de ingeniería – Diseñar las interfaces para registrar cuenta de usuario.

---

#### TAREA DE INGENIERÍA

---

Numero de tarea: **49**

Numero de historia de usuario: **9**

Nombre de tarea: **Diseñar las interfaces para registrar cuenta de usuario**

Tipo de tarea: **Diseño**

Puntos estimados: **0.2**

Fecha de inicio: **26/08/2016**

Fecha fin: **26/08/2016**

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

Diseñar las interfaces necesarias que permita solicitar una cuenta de acceso, para ello debe permitir registrar los datos personales y de contacto de los usuarios; todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.

### Tabla N° 4.65

### Tarea de ingeniería –Registrar cuenta de usuario.

---

#### TAREA DE INGENIERÍA

---

Numero de tarea: **50**

Numero de historia de usuario: **9**

Nombre de tarea: **Registrar cuenta de usuario**

Tipo de tarea: **Desarrollo**

Puntos estimados: **0.4**

Fecha de inicio: **29/08/2016**

Fecha fin: **30/08/2016**

Programador responsable: **Ayde Quispe Gómez**

---

Descripción:

El usuario se encuentra en la interfaz de inicio de sesión y hace clic en el botón “Regístrese Aquí”, el sistema muestra la interfaz “Registrar nuevo usuario” con los campos (Nombres, Apellido paterno, Apellido materno, N° DNI, Genero, Fecha de nacimiento, Teléfono, Correo electrónico, Nombre de usuario, Contraseña).

El usuario ingresa los datos según las validaciones y hace clic en el botón “Guardar”, el sistema almacena la información en la base de datos y muestra un mensaje de

confirmación.

#### **Tabla N° 4.66**

##### **Tarea de ingeniería – Realizar pruebas de aceptación**

TAREA DE INGENIERÍA	
Numero de tarea: <b>51</b>	Numero de historia de usuario: <b>9</b>
Nombre de tarea: <b>Realizar pruebas de aceptación</b>	
Tipo de tarea: <b>Prueba</b>	Puntos estimados: <b>0.2</b>
Fecha de inicio: <b>31/08/2016</b>	Fecha fin: <b>31/08/2016</b>
Programador responsable: <b>Ayde Quispe Gómez</b>	
Descripción: El cliente junto al encargado de pruebas realizarán las pruebas de aceptación sobre la interfaces involucrada en la historia de usuario, para determinar el correcto funcionamiento de la historia de usuario.	

**HU:** Iniciar sesión

#### **Tabla N° 4.67**

##### **Tarea de ingeniería – Diseñar las interfaces para iniciar sesión.**

TAREA DE INGENIERÍA	
Numero de tarea: <b>52</b>	Numero de historia de usuario: <b>10</b>
Nombre de tarea: <b>Diseñar las interfaces para iniciar sesión</b>	
Tipo de tarea: <b>Diseño</b>	Puntos estimados: <b>0.4</b>
Fecha de inicio: <b>01/09/2016</b>	Fecha fin: <b>02/09/2016</b>
Programador responsable:	
Descripción: Diseñar las interfaces necesarias que permita el inicio de sesión de los usuarios, mediante el ingreso de sus datos de cuenta respectivos; todo ello teniendo en cuenta aspectos estéticos, funcionales y validaciones correspondientes.	

#### **Tabla N° 4.68**

##### **Tarea de ingeniería –Iniciar sesión.**

---

**TAREA DE INGENIERÍA**

---

**Numero de tarea: 53****Numero de historia de usuario: 10**

---

**Nombre de tarea: Iniciar sesión**

---

**Tipo de tarea: Desarrollo****Puntos estimados: 0.4**

---

**Fecha de inicio: 05/09/2016****Fecha fin: 06/09/2016**

---

**Programador responsable: Ayde Quispe Gómez**

---

**Descripción:**

El usuario (Administrador, tutor y tutorado) se encuentra en la interfaz principal del sistema e ingresa el nombre de usuario y la clave de usuario y hace clic en el botón "Ingresar", el sistema verifica la existencia del nombre de usuario, inmediatamente valida la clave del usuario y muestra la interfaz "principal" de acuerdo al rol del usuario.

---

**Tabla N° 4.69****Tarea de ingeniería – Realizar pruebas de aceptación**

---

**TAREA DE INGENIERÍA**

---

**Numero de tarea: 54****Numero de historia de usuario: 10**

---

**Nombre de tarea: Realizar pruebas de aceptación**

---

**Tipo de tarea: Prueba****Puntos estimados: 0.2**

---

**Fecha de inicio: 07/09/2016****Fecha fin: 07/09/2016**

---

**Programador responsable:**

---

**Descripción:**

El cliente junto al encargado de pruebas realizarán las pruebas de aceptación sobre la interfaces involucrada en la historia de usuario, para determinar el correcto funcionamiento de la historia de usuario.

---

---

### **4.3.3 PLAN DE ITERACIÓN**

**Plan para la primera iteración**

**Tabla N° 4.70**



	Tareas de ingeniería	Días	Comienzo	Fin
1	<b>HU-01: Administrar tutor</b>	<b>15 días</b>	<b>mar 05/04/16</b>	<b>lun 25/04/16</b>
2	Diseñar base de datos para administrar tutor	2 días	mar 05/04/16	mié 06/04/16
3	Diseñar las tarjetas CRC para administrar tutores	1 día	jue 07/04/16	jue 07/04/16
4	Diseñar las interfaces para administrar tutores	2 días	vie 08/04/16	lun 11/04/16
5	Registrar tutor	4 días	mar 12/04/16	vie 15/04/16
6	Mantener tutor	4 días	lun 18/04/16	jue 21/04/16
7	Realizar pruebas de aceptación	2 días	vie 22/04/16	lun 25/04/16
8	Fin historia	0 días	lun 25/04/16	lun 25/04/16
9	<b>HU-02: Administrar tutorado</b>	<b>11 días</b>	<b>mar 26/04/16</b>	<b>mar 10/05/16</b>
10	Diseñar base de datos para administrar tutorado	2 días	mar 26/04/16	mié 27/04/16
11	Diseñar las tarjetas CRC para administrar tutorados	1 día	jue 28/04/16	jue 28/04/16
12	Diseñar las interfaces para administrar tutorados	3 días	vie 29/04/16	mar 03/05/16
13	Registrar tutorado	2 días	mié 04/05/16	jue 05/05/16
14	Mantener tutorado	2 días	vie 06/05/16	lun 09/05/16
15	Realizar pruebas de aceptación	1 día	mar 10/05/16	mar 10/05/16
16	Fin historia	0 días	mar 10/05/16	mar 10/05/16

**Plan para la primera iteración Fuente: Elaboración propia**

## Plan para la segunda iteración

Tabla N° 4.71

	Tareas de ingeniería	Días	Comienzo	Fin
17	<b>HU-03: Administrar objetivos del tutor</b>	<b>11 días</b>	<b>mié 11/05/16</b>	<b>mié 25/05/16</b>
18	Diseñar base de datos para administrar objetivos del tutor	2 días	mié 11/05/16	jue 12/05/16
19	Diseñar las tarjetas CRC para administrar objetivos	1 día	vie 13/05/16	vie 13/05/16
20	Diseñar las interfaces para administrar objetivos	3 días	lun 16/05/16	mié 18/05/16
21	Registrar objetivos	2 días	jue 19/05/16	vie 20/05/16
22	Mantener objetivos	2 días	lun 23/05/16	mar 24/05/16
23	Realizar pruebas de aceptación	1 día	mié 25/05/16	mié 25/05/16
24	Fin historia	0 días	mié 25/05/16	mié 25/05/16
25	<b>HU-04: Administrar actividades y talleres del tutor</b>	<b>15 días</b>	<b>jue 26/05/16</b>	<b>mié 15/06/16</b>
26	Diseñar base de datos para administrar actividades y talleres del tutor	3 días	jue 26/05/16	lun 30/05/16
27	Diseñar las tarjetas CRC para administrar actividades y talleres del tutor	2 días	mar 31/05/16	mié 01/06/16
28	Diseñar las interfaces para administrar actividades y talleres del tutor	3 días	jue 02/06/16	lun 06/06/16
29	Registrar objetivos	3 días	mar 07/06/16	jue 09/06/16
30	Mantener actividades y talleres	2 días	vie 10/06/16	lun 13/06/16
31	Realizar pruebas de aceptación	2 días	mar 14/06/16	mié 15/06/16
32	Fin historia	0 días	mié 15/06/16	mié 15/06/16
33	<b>HU-05: Asignación de tutorados</b>	<b>11 días</b>	<b>jue 16/06/16</b>	<b>jue 30/06/16</b>
34	Diseñar base de datos para asignar tutorados	3 días	jue 16/06/16	lun 20/06/16
35	Diseñar las tarjetas CRC para asignar tutorados	1 día	mar 21/06/16	mar 21/06/16
36	Diseñar las interfaces para asignar tutorados	3 días	mié 22/06/16	vie 24/06/16
37	Registrar tutorados	3 días	lun 27/06/16	mié 29/06/16
38	Realizar pruebas de aceptación	1 día	jue 30/06/16	jue 30/06/16
39	Fin historia	0 días	jue 30/06/16	jue 30/06/16

Plan para la segunda iteración Fuente: Elaboración propia

## Plan para la tercera iteración

Tabla N° 4.72

	Tareas de ingeniería ▼	Días ▼	Comienzo ▼	Fin ▼
40	▲ <b>HU-06: Registrar cursos del tutorado</b>	<b>11 días</b>	<b>vie 01/07/16</b>	<b>vie 15/07/16</b>
41	Diseñar base de datos para registrar cursos del tutorado	2 días	vie 01/07/16	lun 04/07/16
42	Diseñar las tarjetas CRC para registrar cursos	1 día	mar 05/07/16	mar 05/07/16
43	Diseñar las interfaces para registrar cursos	3 días	mié 06/07/16	vie 08/07/16
44	Registrar cursos	2 días	lun 11/07/16	mar 12/07/16
45	Mantener cursos	2 días	mié 13/07/16	jue 14/07/16
46	Realizar pruebas de aceptación	1 día	vie 15/07/16	vie 15/07/16
47	Fin Historia	0 días	vie 15/07/16	vie 15/07/16
48	▲ <b>HU-07: Registrar notas del taller</b>	<b>14 días</b>	<b>lun 18/07/16</b>	<b>jue 04/08/16</b>
49	Diseñar base de datos para administrar notas del taller	2 días	lun 18/07/16	mar 19/07/16
50	Diseñar las tarjetas CRC para registrar notas	3 días	mié 20/07/16	vie 22/07/16
51	Diseñar las interfaces para registrar notas	3 días	lun 25/07/16	mié 27/07/16
52	Registrar notas	3 días	jue 28/07/16	lun 01/08/16
53	Mantener notas	2 días	mar 02/08/16	mié 03/08/16
54	Realizar pruebas de aceptación	1 día	jue 04/08/16	jue 04/08/16
55	Fin historia	0 días	jue 04/08/16	jue 04/08/16

Plan para la tercera iteración Fuente: Elaboración propia

## Plan para la cuarta iteración

Tabla N° 4.73

	Tareas de ingeniería	Días	Comienzo	Fin
56	<b>HU-08: Registrar evaluación de objetivos</b>	<b>11 días</b>	<b>vie 05/08/16</b>	<b>vie 19/08/16</b>
57	Diseñar base de datos para evaluar objetivos	3 días	vie 05/08/16	mar 09/08/16
58	Diseñar las tarjetas CRC para registrar evaluación de objetivos	3 días	mié 10/08/16	vie 12/08/16
59	Diseñar las interfaces para registrar evaluación de objetivos	2 días	lun 15/08/16	mar 16/08/16
60	Registrar evaluación de objetivos	2 días	mié 17/08/16	jue 18/08/16
61	Realizar pruebas de aceptación	1 día	vie 19/08/16	vie 19/08/16
62	Fin historia	0 días	vie 19/08/16	vie 19/08/16
63	<b>HU-09: Administrar usuarios</b>	<b>8 días</b>	<b>lun 22/08/16</b>	<b>mié 31/08/16</b>
64	Diseñar base de datos para registrar cuenta de usuario	2 días	lun 22/08/16	mar 23/08/16
65	Diseñar las tarjetas CRC para registrar cuenta de usuario	2 días	mié 24/08/16	jue 25/08/16
66	Diseñar las interfaces para registrar cuenta de usuario	1 día	vie 26/08/16	vie 26/08/16
67	Registrar cuenta de usuario	2 días	lun 29/08/16	mar 30/08/16
68	Realizar pruebas de aceptación	1 día	mié 31/08/16	mié 31/08/16
69	Fin historia	0 días	mié 31/08/16	mié 31/08/16
70	<b>HU-10: Iniciar sesión</b>	<b>5 días</b>	<b>jue 01/09/16</b>	<b>mié 07/09/16</b>
71	Diseñar las interfaces para iniciar sesión	2 días	jue 01/09/16	vie 02/09/16
72	Iniciar sesión	2 días	lun 05/09/16	mar 06/09/16
73	Realizar pruebas de aceptación	1 día	mié 07/09/16	mié 07/09/16
74	Fin historia	0 días	mié 07/09/16	mié 07/09/16

Figura N°: Plan para la cuarta iteración Fuente: Elaboración propia

#### 4.3.4 CASOS DE PRUEBA DE ACEPTACIÓN

Tabla N° 4.74

##### Casos de prueba de aceptación para la primera iteración

N°	HISTORIA DE	N°	CASO DE PRUEBA
HU	USUARIO	CP	
01	Administrar tutor	01	El usuario ingresa los datos del tutor, el sistema valida los campos del formulario y permite la búsqueda, registro, listado y actualización de los datos del tutor.
02	Administrar tutorado	02	El usuario ingresa los datos del tutorado, el sistema valida los campos del formulario y permite la búsqueda, registro, listado y <u>actualización de los datos del tutorado.</u>

Tabla N° 4.75

##### Casos de prueba de aceptación para la segunda iteración

N°	HISTORIA DE USUARIO	N°	CASO DE PRUEBA
HU		CP	
03	Administrar objetivos del tutor	03	El usuario ingresa los objetivos planteados por cada tutor, el sistema valida los campos del formulario y permite visualizar, buscar, registrar y actualizar los objetivos.
04	Administrar actividades y talleres del tutor	04	Se ingresa las actividades y talleres a desarrollar en un semestre académico, el sistema valida los datos ingresados y permite registrar, buscar, y actualizar.
05	Asignación de tutorados	05	El usuario hace la búsqueda de los tutorados inscritos y se asigna, el sistema valida los campos del formulario y realiza <u>el registro del tutorado.</u>

**Tabla N° 4.76**

**Casos de prueba de aceptación para la tercera iteración**

N°	HISTORIA DE USUARIO	N°	CASO DE PRUEBA
HU		CP	
06	Registrar cursos del tutorado	06	El sistema valida los campos de los formularios y permite registrar a los tutorados en los cursos de reforzamiento que se habilitaran para el periodo vigente.
07	Registrar notas del taller	07	El usuario registra las notas de cada taller para cada uno de los tutorados y las visualiza.

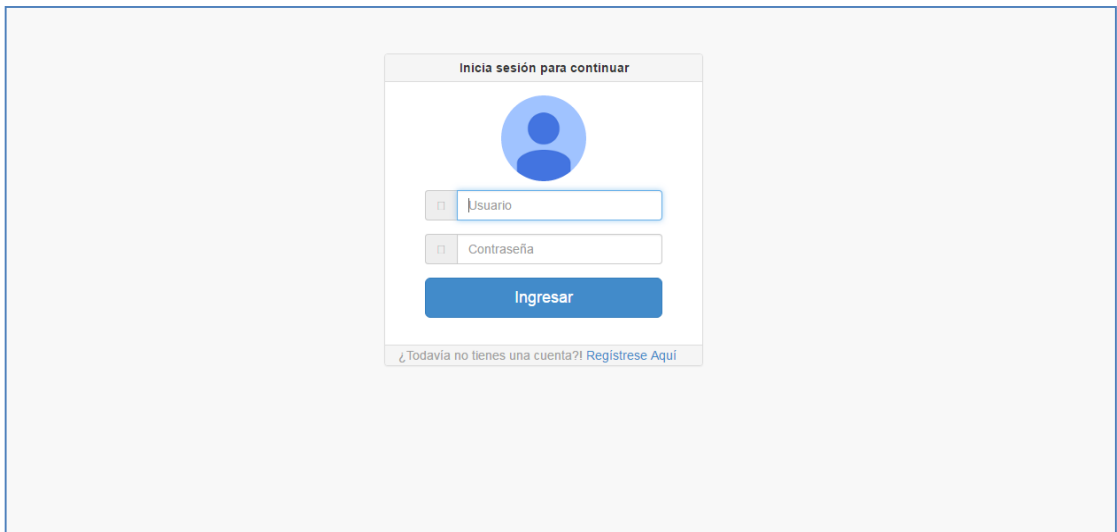
**Tabla N° 4.77**

**Casos de prueba de aceptación para la cuarta iteración**

N°	HISTORIA DE USUARIO	N°	CASO DE PRUEBA
HU		CP	
08	Registrar evaluación de objetivos	08	El sistema permita registrar el grado de cumplimiento de los objetivos planteados para cada tutor.
09	Administrar usuarios	09	El sistema permite una adecuada administración de las cuentas de usuario.
10	Iniciar sesión	10	El sistema permite acceso al sistema de acuerdo al rol que cumpla en el proyecto Hatun Ñan.

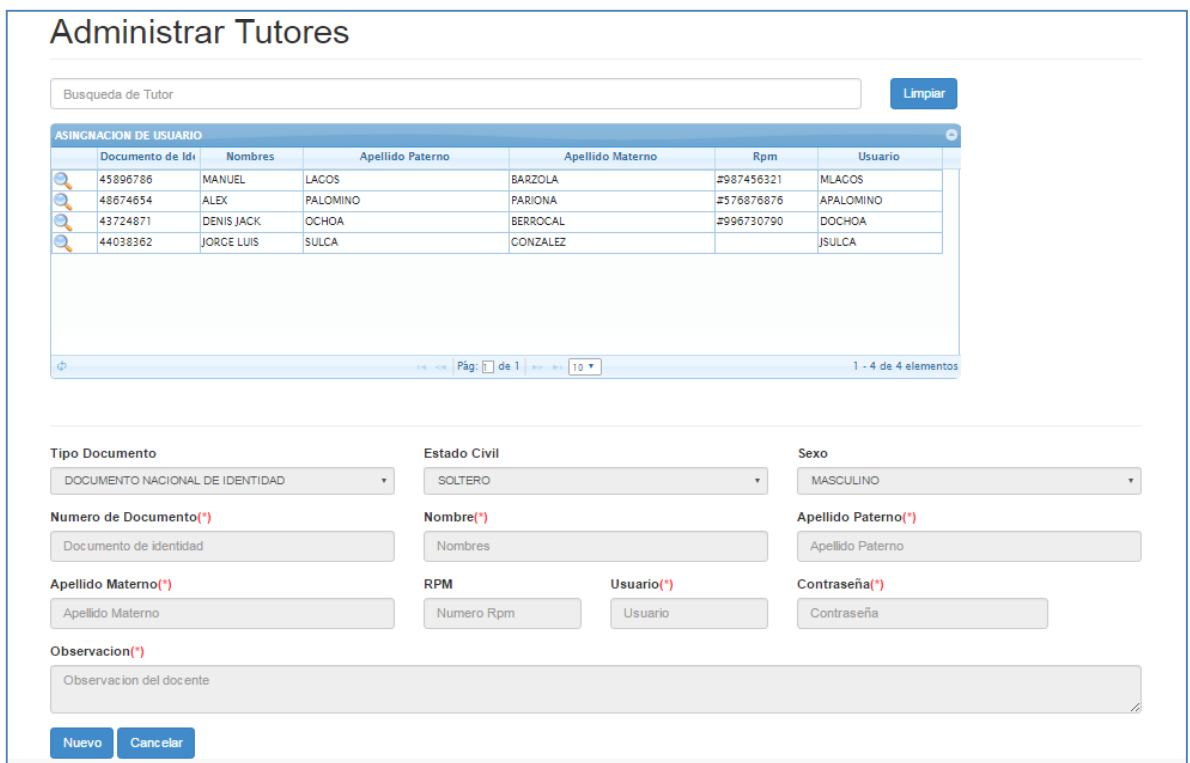
**4.3.5 INTERFAZ GRÁFICA DE USUARIO (GUI)**

Para el diseño de las interfaces gráficas de usuario de la aplicación web se consideró estándares de usabilidad, teniendo en cuenta que el objetivo de la tesis sí considera su evaluación; los factores para los estándares utilizados son: botones, títulos de páginas, menús, fuentes, tamaño de fuente, colores; que permitan tener interfaces amigables y fáciles de usar.



**Figura N° 4.2 Interfaz de acceso**

**Fuente: Elaboración propia**



**Figura N° 4.3**

**Interfaz administrar tutores Fuente: Elaboración propia**

### Metas del Tutor

EVALUACION DE OBJETIVOS						
	Tutor	Objetivo	Fecha Inicio	Fecha Fin	Observacion	Estado
	MANUEL LAGOS BARZOLA	TERCER REGISTRO	05/01/2017	31/01/2017	ninguno	CONCLUIDO
	ALEX PALOMINO PARIONA	OBJETIVO N 01EEEE	31/03/2015	26/03/2015	kekekake	CONCLUIDO
	MANUEL LAGOS BARZOLA	OBJETIVO X	05/01/2017	31/01/2017	ninguno	CONCLUIDO
	ALEX PALOMINO PARIONA	RRRRRR	05/09/2015	11/09/2015	DDDD	CONCLUIDO

1 - 4 de 4 elementos

---

**Tutor**

**Objetivo**

Concluido?    
  satisfactorio?    
  Observado?

**Observacion(\*)**

**Figura N° 4.4**  
**Interfaz registro metas del tutor Fuente: Elaboración propia**

### Administrar mis objetivos

OBJETIVOS					
	Objetivo	Fecha Inici	Fecha Fin	Observacion	Estado
	LOGRAR EL EMPODERAMIENTO DE IDENTIDAD CULTURAL	05/01/2017	31/01/2017	NINCUNA	CONCLUIDO
	LOGRAR LA APROBACION DE TUTORADOS EN LOS CURSOS DE LA	05/01/2017	31/01/2017	NINCUNO	CONCLUIDO

1 - 2 de 2 elementos

---

**Tutor**

**Denominacion(\*)**

**Fecha Inicio(\*)**

**Fecha Fin**

**Observacion(\*)**

**Figura N° 4.5**  
**Interfaz administrar objetivos Fuente: Elaboración propia**



### Administrar Mis Actividades

ACTIVIDADES					
	Actividad	Fecha Inicio	Fecha Fin	Observacion	Estado
	Organización de cursos de reforzamiento de acuerdo a la	20/03/2017	28/07/2017	Fechas sujetas a modificaciones	EN PROCESO

Pág. 1 de 1 10 1 - 1 de 1 elementos

**Tutor**

**Nombre de Actividad(\*)**

**Observacion(\*)**

**Fecha Inicio(\*)**

**Fecha Fin**

**Figura N° 4.6**

**Interfaz administración de actividades Fuente: Elaboración propia**

### Administrar mis Talleres

TALLERES					
	Taller	Fecha Inicio	Fecha Fin	Observacion	Estado
	Sexualidad Responsable	03/01/2015	03/05/2015	sexualidad responsable	CONCLUIDO

Pág. 1 de 1 10 1 - 1 de 1 elementos

**Tutor**

**Denominacion(\*)**

**Fecha Inicio(\*)**

**Fecha Fin**

**Observacion(\*)**

**Figura N° 4.7**

**Interfaz administración de talleres Fuente: Elaboración propia**

## Asistencia de taller

Sexualidad Responsable  Limpiar

ASISTENCIA DE TALLER		
Tutorado	Fecha Asignación	Observación
JHON JAVIER HUARACA PAREDES	03/01/2015	segunda prueba

Pág: 1 de 1 11 1 - 1 de 1 elementos

---

**Tutor**

**Tutorado(\*)**  
 Limpiar

**Observacion(\*)**

Nuevo Cancelar

**Figura N° 4.8**

**Interfaz registro de asistencia por taller Fuente: Elaboración propia**

## Asignacion de tutorados

TUTORADOS ASIGNADOS		
Tutorado	Fecha Asignación	Observación
PEDRO CALVEZ NOA	06/01/2014	SIN OBSERVACION

Pág: 1 de 1 11 1 - 1 de 1 elementos

**Tutor(\*)**  **Semestre(\*)**

**Tutorado(\*)**

**Observacion(\*)**

Nuevo Cancelar

**Figura N° 4.9**

**Interfaz registrar asignación de tutorados Fuente: Elaboración propia**

## Registrar Nota de Taller

Sexualidad Responsable Limpiar

ASISTENCIA A TALLER		
	Tutorado	Nota
🔍	JHON JAVIER HUARACA PAREDES	10

Pág: 1 de 1 10 ▼ 1 - 1 de 1 elementos

---

**Tutorado** **Nota(\*)**

JHON JAVIER HUARACA PAREDES

**Observacion(\*)**

**Figura N° 4.10**

Interfaz listado por estándar de las preguntas de un cuestionario

Fuente: Elaboración propia

## SEMESTRE 2014-I

Cursos							
	Sigla	Nombre Asignatura	Creditos	Nota	Fecha Entrega	Modalidad	

Pág: 1 de 0 10 ▼ No hay resultados

---

**Tutorado** **Semestre**

Tutorado 2014-I ▼

**Sigla(\*)** **Denominacion(\*)** **Tipo de Matricula(\*)**

Ingrese la sigla del curso Ingrese la denominacion del curso REGULAR ▼

**Serie** **Plan de Estudios** **Requisito 1**

**Nro de Créditos** **Tipo de Curso** **Requisito 2**

Nro de Creditos del curso

**Observacion(\*)**

Ingrese alguna observacion sobre el curso

**Figura N° 4.11**

Interfaz para responder a cada una de las preguntas de un cuestionario

Fuente: Elaboración propia

### 4.3.6 TARJETA CRC

Para un diseño simple, se procede a definir tarjetas CRC, que permitan identificar las clases, sus responsabilidades y las colaboraciones que realiza para una historia de usuario y las tareas correspondientes.

**Tabla N° 4.76**

#### **Tarjeta CRC - Persona**

TARJETA CRC		
Número: 01	Escenario: Administrar persona	
Nombre CRC: Persona		
Responsabilidades	Colaboradores	Atributos
Registrar persona		nombre
Actualizar persona		apePaterno
Listar persona		apeMaterno
Buscar persona		razonSocial
		numDocumento
		<u>domiProcesal</u>
		rpm
		email
		imagen
		fechaNacimiento
		observación
		estado

Observaciones

Ninguno.

**Tabla N° 4.77 Tarjeta CRC - Tutor**

TARJETA CRC		
Número: 02	Escenario: Administrar tutor	
Nombre CRC: Tutor		
Responsabilidades	Colaboradores	Atributos

Registrar tutor	Persona	idPersona
Obtener la tutor	Escuela	idEscuela
Actualizar tutor		nnumeroDocumento
Buscar tutor		observacion fechaRegistro estado fechaVigencia

---

Observaciones

El registro de los datos de la persona son implícitos.

---

**Tabla N° 4.78**

**Tarjeta CRC - Tutorado**

TARJETA CRC		
Número: 03	Escenario: Administrar tutorado	
Nombre CRC: Tutorado		
Responsabilidades	Colaboradores	Atributos
Registrar tutorado	Persona	idPersona
Buscar tutorado	Escuela	idEscuela
Actualizar tutorado	Dependencia	idDependencia
Listar tutorado	Ocupación	idOcupación
	<input type="checkbox"/>	numeroDocumento observación fechaRegistro

---

Observaciones

Los tutorados se registran siempre y cuando hayan superado la fase de entrevista personal.

---

**Tabla N° 4.79**

**Tarjeta CRC – Registro objetivo**

---

TARJETA CRC

---

---

Número: 04	Escenario: Administrar objetivos	
Nombre CRC: RegObjetivo		
Responsabilidades	Colaboradores	Atributos
Registrar objetivo	Tutor	idRegObjetivo
Mantener objetivo		idTutor
Listar objetivos		denominacion
Eliminar objetivos		fechaInicio
		fechaFin
		observacion

---

Observaciones

Ninguno

**Tabla N° 4.80**

**Tarjeta CRC – Registro Actividad**

---

TARJETA CRC

---

---

Número: 05	Escenario: Administrar actividad	
Nombre CRC: RegActividad		
Responsabilidades	Colaboradores	Atributos
Registrar actividad	Tutor	idRegActividad
Mantener actividad		idTutor
Listar actividad		denominacion
Eliminar actividad		fechaInicio
		fechaFin
		observacion

---

Observaciones

Ninguno

---

**Tabla N° 4.81****Tarjeta CRC – Registro Taller**

TARJETA CRC		
Número: 06	Escenario: Administrar taller	
Nombre CRC: RegTaller		
Responsabilidades	Colaboradores	Atributos
Registrar taller	Tutor	idRegTaller
Mantener taller		idTutor
Listar Taller		denominacion
Eliminar taller		fechaInicio
		fechaFin
		observacion
<u>Observaciones</u>		
<u>Ninguno</u>		

**Tabla N° 4.82****Tarjeta CRC – AsignacionTutorado**

TARJETA CRC		
Número: 07	Escenario: Asignar tutorados	
Nombre CRC: RegAsignacion		
Responsabilidades	Colaboradores	Atributos
Registrar asignación de tutorado	Tutor Tutorado	idRegAsignacion idTutor
Ver Tutorados asignados	Semestre	idTutorado
Buscar tutores asignados		idSemestre observacion fechaRegistro
<u>Observaciones</u>		
<u>Los tutorados son asignados previo registro</u>		

**Tabla N° 4.83****Tarjeta CRC – RegistroCurso**

TARJETA CRC		
Número: 08	Escenario: Administrar curso	
Nombre CRC: RegCurso		
Responsabilidades	Colaboradores	Atributos
Registrar cursos	PlanEstudio	idRegCurso
Buscar cursos	Serie	idPlanEstudio
Listar cursos	TipoCurso	idSerie
Actualizar cursos		idTipoCurso
		sigla
		denominacion
		credito
		observacion
		requisitos
<u>Observaciones</u>		
Ninguno		

**Tabla N° 4.84****Tarjeta CRC – Registro notas taller**

TARJETA CRC		
Número: 09	Escenario: Administrar notas	
Nombre CRC: RegNotasTaller		
Responsabilidades	Colaboradores	Atributos
Registrar notas	Taller	idRegNotasTaller
Actualizar notas	Tutorado	idTaller
Buscar notas	Tutor	idTutorado
Listar notas		nota
<u>Observaciones</u>		
Las notas se registran para cada taller de todos sus participantes.		



**Tabla N° 4.85**

**Tarjeta CRC – Registro evaluación de objetivos**

---

TARJETA CRC

---

Número: 10                      Escenario: Administrar objetivos

---

Nombre CRC: RegEvaluacionObjetivos

---

Responsabilidades	Colaboradores	Atributos
-------------------	---------------	-----------

---

Registrar evaluación	Objetivo	idRegEvaluacionObjetivos
----------------------	----------	--------------------------

Actualizar calificación de Tutor		idObjetivo
----------------------------------	--	------------

evaluación		idTutor
------------	--	---------

Ver evaluaciones		calificacion
------------------	--	--------------

		observacion
--	--	-------------

---

Observaciones

Las notas se registran para cada taller de todos sus participantes.

---



## 4.3.7 CÓDIGO FUENTE

### Código fuente para la clase entidad

```
1 namespace UNSCH.TUTORIA.BE
2 {
3     public class BaseBE
4     {
5         public int? iOpSp { get; set; }
6         public string nvObservacion { get; set; }
7         public int? iIdRegUsuaRegistra { get; set; }
8         public System.DateTime? dtFechRegistra { get; set; }
9         public int? iCantidadModifica { get; set; }
10        public bool? bEstado { get; set; }
11    }
12 }

1 using System;
2 namespace UNSCH.TUTORIA.BE.TUTO
3 {
4     public class RegTutorBE : BaseBE
5     {
6         public float? biIdRegTutor { get; set; }
7         public float? biIdRegPersona { get; set; }
8         public string cCodRegEscuela { get; set; }
9         public string nvNumDocumento { get; set; }
10        public DateTime? dtFechaVigencia { get; set; }
11    }
12 }
```

Figura N°4.13

### Código fuente para el DA

```
1 namespace UNSCH.TUTORIA.DA.TUTO
2 {
3     #region Transaccional
4     public class RegTutorDAT : GenericDA<RegTutorDAT>
5     {
6         /// <summary>
7         /// fInsertarRegTutor, es un metodo que Inserta a la tabla RegTutor
8         /// </summary>
9         /// <param name="poRegTutorBE">Objeto correspondiente a RegTutor</param>
10        /// <returns>Retorna un numero entero positivo cuando es exito, numero cero cuando no hubo insercion alguna (la BD decidio no insertar)
11        public int fInsertarRegTutor(RegTutorBE poRegTutorBE)
12        {
13            SqlCommand oCommand;
14            SqlConnection cn = HELPER.MethodCommonHelp.GetConnectionDataBase();
15
16            try
17            {
18                cn.Open();
19
20                oCommand = new SqlCommand(System.Data.CommandType.StoredProcedure, "tuto.usp_RegTutor_ins", cn, null);
21
22                oCommand.AgregarParametro("@biIdRegPersona", SqlDbType.BigInt, ParameterDirection.Input, poRegTutorBE.biIdRegPersona);
23                oCommand.AgregarParametro("@cCodRegEscuela", SqlDbType.Char, 5, ParameterDirection.Input, poRegTutorBE.cCodRegEscuela);
24                oCommand.AgregarParametro("@nvNumDocumento", SqlDbType.Char, 8, ParameterDirection.Input, poRegTutorBE.nvNumDocumento);
25                oCommand.AgregarParametro("@nvObservacion", SqlDbType.NVarChar, 500, ParameterDirection.Input, poRegTutorBE.nvObservacion);
26                oCommand.AgregarParametro("@iIdRegUsuaRegistra", SqlDbType.Int, ParameterDirection.Input, poRegTutorBE.iIdRegUsuaRegistra);
27                oCommand.AgregarParametro("@dtFechRegistra", SqlDbType.DateTime, ParameterDirection.Input, poRegTutorBE.dtFechRegistra);
28                oCommand.AgregarParametro("@iCantidadModifica", SqlDbType.Int, ParameterDirection.Input, poRegTutorBE.iCantidadModifica);
29                oCommand.AgregarParametro("@bEstado", SqlDbType.Bit, ParameterDirection.Input, poRegTutorBE.bEstado);
30                oCommand.AgregarParametro("@ValueReturn", SqlDbType.Int, ParameterDirection.ReturnValue, 0);
31
32                oCommand.ExecuteNonQuery();
33
34                return Convert.ToInt32(oCommand.strObtenerParametro("@ValueReturn"));
35            }
36            catch (Exception)
37            {
38            }
39        }
40    }
41 }
```

Figura N°4.14

```

40     {
41         throw;
42         //MethodCommonHelp.WriteLog(ex, "RegTutorDAT", MessageSystemHelp.CAPA_DATOS, "fInsertarRegTutor", "Usuario");
43     }
44     finally
45     {
46         cn.Close();
47         oCommand = null;
48     }
49 }
50
51 /// <summary>
52 /// fActualizarRegTutor, es un metodo que Actualiza a la tabla RegTutor
53 /// </summary>
54 /// <param name="poRegTutorBE">Objeto correspondiente a RegTutor</param>
55 /// <returns>Retorna un numero entero positivo cuando es exito, número cero cuando no hubo actualizacion alguna (la BD decidio no actualizar)
56 public int fActualizarRegTutor(RegTutorBE poRegTutorBE)
57 {
58     SqlHelp oCommand;
59     SqlConnection cn = HELPER.MethodCommonHelp.GetConnectionDataBase();
60
61     try
62     {
63         cn.Open();
64
65         oCommand = new SqlHelp(System.Data.CommandType.StoredProcedure, "tuto.usp_RegTutor_upd", cn, null);
66
67         oCommand.AgregarParametro("@biIdRegTutor", SqlDbType.BigInt, ParameterDirection.Input, poRegTutorBE.biIdRegTutor);
68         oCommand.AgregarParametro("@biIdRegPersona", SqlDbType.BigInt, ParameterDirection.Input, poRegTutorBE.biIdRegPersona);
69         oCommand.AgregarParametro("@oCodRegEscuela", SqlDbType.Char, 5, ParameterDirection.Input, poRegTutorBE.oCodRegEscuela);
70         oCommand.AgregarParametro("@nvNumDocumento", SqlDbType.Char, 8, ParameterDirection.Input, poRegTutorBE.nvNumDocumento);
71         oCommand.AgregarParametro("@nvObservacion", SqlDbType.NVarChar, 500, ParameterDirection.Input, poRegTutorBE.nvObservacion);
72         oCommand.AgregarParametro("@iIdRegUsuaRegistra", SqlDbType.Int, ParameterDirection.Input, poRegTutorBE.iIdRegUsuaRegistra);
73         oCommand.AgregarParametro("@dtFechRegistra", SqlDbType.DateTime, ParameterDirection.Input, poRegTutorBE.dtFechRegistra);
74         oCommand.AgregarParametro("@iCantidadModifica", SqlDbType.Int, ParameterDirection.Input, poRegTutorBE.iCantidadModifica);
75         oCommand.AgregarParametro("@bEstado", SqlDbType.Bit, ParameterDirection.Input, poRegTutorBE.bEstado);
76         oCommand.AgregarParametro("@ValueReturn", SqlDbType.Int, ParameterDirection.ReturnValue, 0);
77
78         oCommand.ExecuteNonQuery();
79
80         return Convert.ToInt32(oCommand.strObtenerParametro("@ValueReturn"));
81     }
82     catch (Exception)
83     {
84         throw;
85         //MethodCommonHelp.WriteLog(ex, "RegTutorDAT", MessageSystemHelp.CAPA_DATOS, "fActualizarRegTutor", "Usuario");
86     }
87     finally
88     {
89         cn.Close();
90         oCommand = null;
91     }
92 }
93
94 #endregion
95
96 }

```

**Figura N°4.15**

## Código fuente para la capa de negocio

```
1 namespace UNSCH.TUTORIA.BL.TUTO
2 {
3
4
5     #region Transaccional
6     public class RegTutorBLT : GenericBL<RegTutorBLT>
7     {
8
9         /// <summary>
10        /// fInsertarRegTutor, es un metodo que Inserta a RegTutor
11        /// </summary>
12        /// <param name="poRegTutorBE">Objeto correspondiente a RegTutor</param>
13        /// <returns>Retorna un numero entero positivo cuando es exito, numero cero cuando no hubo insercion alguna (la BD decidio no insertar) y numero -1
14        public int fInsertarRegTutor(RegTutorBE poRegTutorBE)
15        {
16            try
17            {
18                using (TransactionScope scope = new TransactionScope())
19                {
20                    int resultado = 0;
21
22                    resultado = RegTutorDAT.GetInstance.fInsertarRegTutor(poRegTutorBE);
23
24                    if (resultado >= 0)
25                    {
26                        scope.Complete();
27                    }
28                    else
29                    {
30                        new Exception("Error en transaccion - Insertar");
31                    }
32
33                    return resultado;
34                }
35            }
36            catch (Exception)
37            {
38                throw;
39                //MethodCommonHelp.WriteLog(ex, "RegTutorBLT", MessageSystemHelp.CAPA_LOGICO, "fInsertarRegTutor", "Usuario");
40            }
41        }
42    }
43
44    /// <summary>
45    /// fActualizarRegTutor, es un metodo que Actualiza a RegTutor
46    /// </summary>
47    /// <param name="poRegTutorBE">Objeto correspondiente a RegTutor</param>
48    /// <returns>Retorna un numero entero positivo cuando es exito, numero cero cuando no hubo actualizacion alguna (la BD decidio no actualizar) y numero -1
49    public int fActualizarRegTutor(RegTutorBE poRegTutorBE)
50    {
51        try
52        {
53            using (TransactionScope scope = new TransactionScope())
54            {
55                int resultado = 0;
56
57                resultado = RegTutorDAT.GetInstance.fActualizarRegTutor(poRegTutorBE);
58
59                if (resultado >= 0)
60                {
61                    scope.Complete();
62                }
63                else
64                {
65                    new Exception("Error en transaccion - Actualizar");
66                }
67
68                return resultado;
69            }
70        }
71        catch (Exception)
72        {
73            throw;
74            //MethodCommonHelp.WriteLog(ex, "RegTutorBLT", MessageSystemHelp.CAPA_LOGICO, "fActualizarRegTutor", "Usuario");
75        }
76    }
77 }
```

Figura N°4.16

```
40
41
42
43    /// <summary>
44    /// fActualizarRegTutor, es un metodo que Actualiza a RegTutor
45    /// </summary>
46    /// <param name="poRegTutorBE">Objeto correspondiente a RegTutor</param>
47    /// <returns>Retorna un numero entero positivo cuando es exito, numero cero cuando no hubo actualizacion alguna (la BD decidio no actualizar) y numero -1
48    public int fActualizarRegTutor(RegTutorBE poRegTutorBE)
49    {
50        try
51        {
52            using (TransactionScope scope = new TransactionScope())
53            {
54                int resultado = 0;
55
56                resultado = RegTutorDAT.GetInstance.fActualizarRegTutor(poRegTutorBE);
57
58                if (resultado >= 0)
59                {
60                    scope.Complete();
61                }
62                else
63                {
64                    new Exception("Error en transaccion - Actualizar");
65                }
66
67                return resultado;
68            }
69        }
70        catch (Exception)
71        {
72            throw;
73            //MethodCommonHelp.WriteLog(ex, "RegTutorBLT", MessageSystemHelp.CAPA_LOGICO, "fActualizarRegTutor", "Usuario");
74        }
75    }
76 }
```

Figura N°4.17

## Código fuente para la capa de presentación

```
1 @{\n2   ViewBag.Title = "PerfilTutor";\n3   Layout = "~/Views/Shared/_LayoutTutor.cshtml";\n4 }\n5 \n6 <html>\n7 <head>\n8   <title></title>\n9   <script src="../../Scripts/jquery-1.10.2.js"></script>\n10  <script src="../../Scripts/jqueryUpload/jquery.ui.widget.js"></script>\n11  <script src="../../Scripts/jqueryUpload/jquery.fileupload.js"></script>\n12 \n13  <script type="text/javascript">\n14    var imagen = '../Images/photo.png';\n15 \n16    $(document).ready(function () {\n17      loadPage();\n18    });\n19 \n20    $(function () {\n21      $('#file-upload').fileinput({\n22        initialPreview: ["<img src='" + imagen + "' class='file-preview-image'/>"],\n23        'showPreview': true,\n24        'allowedFileExtensions': ['jpg', 'png', 'gif'],\n25        'elErrorContainer': '#errorBlock'\n26      });\n27 \n28      $('#file-upload').fileupload({\n29        dataType: 'json',\n30        url: '@Url.Action("fnSendImage", "PerfilTutor")',\n31        autoUpload: true,\n32        done: function (e, data) {\n33          //alert(JSON.stringify(data));\n34        }\n35      }).on('fileuploadprogressall', function (e, data) {\n36        var progress = parseInt(data.loaded / data.total * 100, 10);\n37      });\n38 \n39      $('#kv-fileinput-upload').click(function (e) {\n40        $.ajaxApp('@Url.Action("fnUpdatePerfil", "PerfilTutor")', {\n41          pnvUrl: $('#file-caption-name').text()\n42        }, false, function (result) {\n43          if (result == 1) {\n44            getPerfil();\n45          } else {\n46            jQuery.MessageError(MENSAJEOPERACION.INCORRECTO.msje);\n47          }\n48        });\n49      });\n50      $('#btnGrabar').click(function () {\n51        savePersona();\n52      });\n53      $('#btnCancelar').click(function () {\n54        clearControl();\n55      });\n56    });\n57 \n58    function clearControl() {\n59      $('#txtDomiProcesal').val("");\n60      $('#txtRpm').val("");\n61      $('#txtEmail').val("");\n62      $('#txtPass').val("");\n63      $('#txtObservacion').val("");\n64      getPerfil();\n65    }\n66    function loadPage() {\n67      getPerfil();\n68    }\n69 \n70    function getPerfil() {\n71      $.ajaxApp('@Url.Action("fnGetPerfil", "PerfilTutor")', {\n72      }, false, function (result) {\n73        $('#txtNombre').val(result.nvNombre);\n74        $('#txtApePaterno').val(result.nvApePaterno);\n75        $('#txtApeMaterno').val(result.nvApeMaterno);\n76        $('#txtUser').val(result.cCodTipoDocumento);\n77        $('#txtNumDocumento').val(result.nvNumDocumento);\n78        $('#txtDomiProcesal').val(result.nvDomiProcesal);\n
```

Figura N°4.18

```

79     $("#txtRpm").val(result.nvRpm);
80     $("#txtEmail").val(result.nvEmail);
81     //$("#txtObservacion").val("");
82     $("#txtObservacion").val(result.nvObservacion);
83     $("#txtUser").data("url", result.nvUrlImagen);
84     if (result.nvUrlImagen == null || result.nvUrlImagen == "") {
85         imagen = '../Images/photo.png';
86     } else {
87         imagen = "../Images/ImgPerfil/" + result.nvUrlImagen;
88     }
89     $(".file-preview-image").attr("src", imagen);
90     });
91 }
92
93 function savePersona() {
94     var resultValidacion = 0;
95     resultValidacion = $.ValidarData('#form');
96     switch (resultValidacion) {
97         case 0:
98             save();
99             break;
100        case -1:
101            jQuery.MessageAlertApp(MENSAJEOPERACION.REQUERIDO.msje);
102            break;
103        case -2:
104            jQuery.MessageError(MENSAJEOPERACION.INCORRECTO.msje);
105            break;
106    }
107 }
108
109 function save() {
110     var oPersona = {
111         //cCodTipoDocumento: $("#txtPass").val(),
112         //nvNombre: $("#txtNombre").val(),
113         nvNombre: $("#txtUser").val(),
114         //nvApePaterno: $("#txtApePaterno").val(),
115         //nvApeMaterno: $("#txtApeMaterno").val(),
116         //nvRazoSocial: $("#txtUser").val(),
117         nvRazoSocial: $("#txtPass").val(),
118         //nvNumDocumento: $("#txtNumDocumento").val(),
119         nvDomiProcesal: $("#txtDomiProcesal").val(),
120         nvRpm: ($("#txtRpm").val() == null ? '' : $("#txtRpm").val()),
121         nvEmail: ($("#txtEmail").val() == null ? '' : $("#txtEmail").val()),
122         nvUrlImagen: ($("#txtUser").data("url")),
123         //dtFechnacimiento: $("#txtFechnacimiento").val(),
124         nvObservacion: ($("#txtObservacion").val())
125     };
126
127     $.ajaxApp('@Url.Action("fnUpdatePersona", "PerfilTutor")', {
128         oRegPersonaBE: oPersona
129     }, false, function (result) {
130         switch (result) {
131             case 1:
132                 jQuery.MessageInformationApp(MENSAJEOPERACION.GRABAR.msje);
133                 clearControl();
134                 break;
135             case -1:
136                 jQuery.MessageError(MENSAJEOPERACION.ERROR.msje);
137                 break;
138         }
139     });
140 }
141
142 <style type="text/css">
143     .btn btn-default .fileinput-remove .fileinput-remove-button {
144         visibility: hidden;
145     }
146 </style>
147 </head>
148 <body>
149     <div class="row">
150         <div class="col-lg-12">
151             <h1 class="page-header">Mi Perfil</h1>
152         </div>
153     </div>
154     <div class="row">
155         <div id="form">
156             <div class="col-lg-8">
157                 <div class="form-group">

```

Figura N°4.19

```

157     <label>Imagen</label>
158     <input id="file-upload" type="file">
159     <span id="errorBlock"></span>
160   </div>
161 </div>
162 <div class="col-lg-4">
163   <div class="form-group">
164     <label>Nombre</label>
165     <input id="txtNombre" class="form-control" placeholder="Nombres" obligatorio disabled uppercase>
166   </div>
167 </div>
168 <div class="col-lg-4">
169   <div class="form-group">
170     <label>Apellido Paterno</label>
171     <input id="txtApePaterno" class="form-control" placeholder="Nombres" obligatorio disabled uppercase>
172   </div>
173 </div>
174 <div class="col-lg-4">
175   <div class="form-group">
176     <label>Apellido Materno</label>
177     <input id="txtApeMaterno" class="form-control" placeholder="Nombres" obligatorio disabled uppercase>
178   </div>
179 </div>
180 <div class="col-lg-4">
181   <div class="form-group">
182     <label>Numero de Documento</label>
183     <input id="txtNumDocumento" class="form-control" maxlength="8" placeholder="Documento de identidad" obligatorio uppercase disabled>
184   </div>
185 </div>
186 <div class="col-lg-8">
187   <div class="form-group">
188     <label>Domicilio Procesal</label><label class="msgResultado"> (*)</label>
189     <input id="txtDomiProcesal" class="form-control" placeholder="Domicilio Procesal" obligatorio uppercase>
190   </div>
191 </div>
192 <div class="col-lg-2">
193   <div class="form-group">
194     <label>RPM</label><label class="msgResultado"> (*)</label>
195     <input id="txtRpm" class="form-control bh-number" maxlength="10" min="6" data-min="000000" data-max="999999999" placeholder="Numero Rpm"
196   </div>
197 </div>
198 <div class="col-lg-2">
199   <div class="form-group">
200     <label>Email</label><label class="msgResultado"> (*)</label>
201     <input id="txtEmail" type="email" class="form-control" placeholder="correo Electronico" obligatorio required uppercase>
202   </div>
203 </div>
204 <div class="col-lg-2">
205   <div class="form-group">
206     <label>Usuario</label>
207     <input id="txtUser" class="form-control" placeholder="Usuario" required disabled uppercase>
208   </div>
209 </div>
210 <div class="col-lg-3">
211   <div class="form-group">
212     <label>Contraseña</label><label class="msgResultado"> (*)</label>
213     <input id="txtPass" type="password" class="form-control" placeholder="Contraseña" required obligatorio>
214   </div>
215 </div>
216 <div class="col-lg-12">
217   <div class="form-group">
218     <label>Comentario</label><label class="msgResultado"> (*)</label>
219     <textarea id="txtObservacion" class="form-control" placeholder="Mis expectativas del programa" obligatorio required uppercase></textarea>
220   </div>
221 </div>
222 <div class="col-lg-12"></div>
223 <div class="col-lg-12">
224   <button id="btnGrabar" type="button" class="btn btn-primary">Grabar</button>
225   <button id="btnCancelar" type="button" class="btn btn-primary">Cancelar</button>
226 </div>
227 </div>
228 </div>
229 </body>
230 </html>

```

Figura N°4.20



## Código fuente para la capa de control

```
15 namespace UNSCH.TUTORIA.WEB.Controllers
16 {
17     public class TutorController : Controller
18     {
19         //
20         // GET: /Tutor/
21
22         public ActionResult Index()
23         {
24             return View();
25         }
26
27         public ActionResult UpdateTutor()
28         {
29             return View();
30         }
31
32         public ActionResult AsignarTutorado()
33         {
34             return View();
35         }
36
37         public ActionResult AsistenciaTaller()
38         {
39             return View();
40         }
41
42
43
44         public JsonResult fGetTutor()
45         {
46             try
47             {
48                 RegTutorBE oRegPersonaBE = new RegTutorBE();
49                 oRegPersonaBE.iOpSp = 12;
50                 oRegPersonaBE.biIdRegPersona = SessionUsuario.GetIdPersona();
51                 return Json(RegTutorBLNT.GetInstance.fGetRegTutor(oRegPersonaBE), JsonRequestBehavior.AllowGet);
52             }
53             catch (Exception ex)
54             {
55                 MethodCommonHelp.WriteLog(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fGetTutor", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
56                 return Json("", JsonRequestBehavior.AllowGet);
57             }
58         }
59
60
61         public JsonResult fGetTaller()
62         {
63             try
64             {
65                 RegTutorBE oRegPersonaBE = new RegTutorBE();
66                 oRegPersonaBE.iOpSp = 12;
67                 oRegPersonaBE.biIdRegPersona = SessionUsuario.GetIdPersona();
68                 return Json(RegTutorBLNT.GetInstance.fGetRegTutor(oRegPersonaBE), JsonRequestBehavior.AllowGet);
69             }
70             catch (Exception ex)
71             {
72                 MethodCommonHelp.WriteLog(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fGetTutor", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
73                 return Json("", JsonRequestBehavior.AllowGet);
74             }
75         }
76
77
78         public JsonResult fnGetSemestre()
79         {
80             try
81             {
82                 MaeSemestreBE oSemestreBE = new MaeSemestreBE();
83                 oSemestreBE.iOpSp = 1;
84                 List<ItemBE> result = MaeSemestreBLNT.GetInstance.fGetListCodAndValorMaeSemestre(oSemestreBE);
85                 return Json(result, JsonRequestBehavior.AllowGet);
86             }
87             catch (Exception ex)
88             {
89                 MethodCommonHelp.WriteLog(ex, "InscripcionController.cs", ParamSystem.CAPA_VISTA, "fnGetTratamiento", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
90                 return Json(-1, JsonRequestBehavior.AllowGet);
91             }
92         }
93     }
94 }
```

Figura N°4.21

```

93
94
95
96 public JsonResult fnSaveAsistencia(RegAsistenciaBE oRegAsistenciaBE)
97 {
98     try
99     {
100
101         oRegAsistenciaBE.IldRegUsuaRegistra = SessionUsuario.GetIdUsuario();
102         //oRegAsistenciaBE.deNota = Convert.ToDecimal(oRegAsistenciaBE.deNota);
103         //oRegAsistenciaBE.iOpSp = 1;
104         int result = RegAsistenciaBLT.GetInstance.fInsertarRegAsistencia(oRegAsistenciaBE);
105         return Json(result, JsonRequestBehavior.AllowGet);
106     }
107     catch (Exception ex)
108     {
109         MethodCommonHelp.WriteLine(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fnSaveAsistencia", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
110         return Json(-1, JsonRequestBehavior.AllowGet); ;
111     }
112 }
113
114 public JsonResult fnSaveAsignacion(RegAsignacionBE oRegAsignacionBE)
115 {
116     try
117     {
118
119         oRegAsignacionBE.IldRegUsuaRegistra = SessionUsuario.GetIdUsuario();
120
121
122         int result = RegAsignacionBLT.GetInstance.fInsertarRegAsignacion(oRegAsignacionBE);
123         return Json(result, JsonRequestBehavior.AllowGet);
124     }
125     catch (Exception ex)
126     {
127         MethodCommonHelp.WriteLine(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fnSaveAsignacion", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
128         return Json(-1, JsonRequestBehavior.AllowGet); ;
129     }
130 }
131
132 public JsonResult fnVerificarAsignacion(RegAsignacionBE oRegAsignacionBE)
133 {
134     try
135     {
136         oRegAsignacionBE.iOpSp = 1;
137         oRegAsignacionBE = RegAsignacionBLMT.GetInstance.fGetRegAsignacion(oRegAsignacionBE);
138         return Json(oRegAsignacionBE, JsonRequestBehavior.AllowGet);
139     }
140     catch (Exception ex)
141     {
142         MethodCommonHelp.WriteLine(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fnVerificarAsignacion", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
143         return Json(-1, JsonRequestBehavior.AllowGet); ;
144     }
145 }
146
147 public JsonResult fnVerificarAsistencia(RegAsistenciaBE oRegAsistenciaBE)
148 {
149     try
150     {
151         oRegAsistenciaBE.iOpSp = 1;
152         oRegAsistenciaBE = RegAsistenciaBLMT.GetInstance.fGetRegAsistencia(oRegAsistenciaBE);
153         return Json(oRegAsistenciaBE, JsonRequestBehavior.AllowGet);
154     }
155     catch (Exception ex)
156     {
157         MethodCommonHelp.WriteLine(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fnVerificarAsistencia", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
158         return Json(-1, JsonRequestBehavior.AllowGet); ;
159     }
160 }
161
162 public ActionResult autoCompleteTutorado(string pnvDenominacion)
163 {
164     RegTutoradoBE oRegTutorBE = new RegTutoradoBE();
165     oRegTutorBE.iOpSp = 15;
166     oRegTutorBE.biIdRegTutorado = SessionUsuario.GetIdPersona();
167     oRegTutorBE.nvObservacion = pnvDenominacion;
168     List<ItemBE> list = RegTutoradoBLMT.GetInstance.fGetListIdAndValorRegTutorado(oRegTutorBE);
169     return Json(list, JsonRequestBehavior.AllowGet);
170 }
171
172 public ActionResult autoCompleteTaller(string pnvDenominacion)
173 {
174     RegTallerBE oRegTallerBE = new RegTallerBE();
175     oRegTallerBE.iOpSp = 5;
176     oRegTallerBE.biIdRegTutor = SessionUsuario.GetIdPersona();
177     oRegTallerBE.nvDenominacion = pnvDenominacion;
178     List<ItemBE> list = RegTallerBLMT.GetInstance.fGetListIdAndValorRegTaller(oRegTallerBE);
179     return Json(list, JsonRequestBehavior.AllowGet);
180 }
181
182
183 public JsonResult fnjqGridAsistencia(int pPaginaActual, int pTamanoPagina, int biIdRegTaller, int biIdRegTutor)
184 {
185     try
186     {
187         RegAsistenciaBE oRegAsistenciaBE = new RegAsistenciaBE();
188         oRegAsistenciaBE.iOpSp = 7;
189         oRegAsistenciaBE.biIdRegTaller = Convert.ToInt32(biIdRegTaller);
190         oRegAsistenciaBE.biIdRegActividad = Convert.ToInt32(biIdRegTutor);
191         string listGrid = Newtonsoft.Json.JsonConvert.SerializeObject(RegAsistenciaBLMT.GetInstance.fGetJqGridRegAsistencia(oRegAsistenciaBE, pPaginaActual,
192         return Json(listGrid, JsonRequestBehavior.AllowGet);
193     }
194     catch (Exception ex)
195     {
196         MethodCommonHelp.WriteLine(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fnjqGridAsistencia", SessionUsuario.GetNombreUsuario(), SessionUsuario.
197         return Json("", JsonRequestBehavior.AllowGet);
198     }
199 }
200
201 public JsonResult fnjqGridAsignacion(int pPaginaActual, int pTamanoPagina, int biIdRegTutor)
202 {
203     try
204     {
205         RegAsignacionBE oRegAsignacionBE = new RegAsignacionBE();
206         oRegAsignacionBE.iOpSp = 6;
207         oRegAsignacionBE.biIdRegTutor = Convert.ToInt32(biIdRegTutor);
208         string listGrid = Newtonsoft.Json.JsonConvert.SerializeObject(RegAsignacionBLMT.GetInstance.fGetJqGridRegAsignacion(oRegAsignacionBE, pPaginaActual,
209         return Json(listGrid, JsonRequestBehavior.AllowGet);

```

Figura N°4.22

```

210 }
211 catch (Exception ex)
212 {
213     MethodCommonHelp.WriteLine(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fnjgGridAsignacion", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
214     return Json("", JsonRequestBehavior.AllowGet);
215 }
216 }
217
218
219 public JsonResult fnGetPerfil()
220 {
221     RegPersonaBE oRegPersonaBE = new RegPersonaBE();
222     try
223     {
224         oRegPersonaBE.biIdRegPersona = SessionUsuario.GetIdPersona();
225         oRegPersonaBE.iOpSp = 3;
226         return Json(RegPersonaBLNT.GetInstance.fnGetRegPersona(oRegPersonaBE), JsonRequestBehavior.AllowGet);
227     }
228     catch (Exception ex)
229     {
230         MethodCommonHelp.WriteLine(ex, "TutorController.cs", ParamSystem.CAPA_VISTA, "fnGetPerfil", SessionUsuario.GetNombreUsuario(), SessionUsuario.GetIp());
231         return null;
232     }
233 }
234
235 public FilePathResult Image()
236 {
237     string filename = Request.Url.AbsolutePath.Replace("/Images", "");
238     string contentType = "";
239     var filePath = new FileInfo(Server.MapPath("~/Images/ImgPerfil") + filename);
240
241     var index = filename.LastIndexOf(".") + 1;
242     var extension = filename.Substring(index).ToUpperInvariant();
243
244     // Fix for IE not handling jpg image types
245     contentType = string.Compare(extension, "JPG") == 0 ? "image/jpeg" : string.Format("image/{0}", extension);
246
247     return File(filePath.FullName, contentType);
248 }
249
250 public JsonResult fnSendImge()
251 {
252     var r = new List<UploadFilesResult>();
253     foreach (string file in Request.Files)
254     {
255         HttpPostedFileBase hpf = Request.Files[file] as HttpPostedFileBase;
256         if (hpf.ContentLength == 0)
257             continue;
258
259         string savedFileName = Path.Combine(Server.MapPath("~/Images/ImgPerfil"), Path.GetFileName(hpf.FileName));
260         hpf.SaveAs(savedFileName);
261
262         r.Add(new UploadFilesResult()
263         {
264             Name = hpf.FileName,
265             Length = hpf.ContentLength,
266             Type = hpf.ContentType
267         });
268     }
269     return Json(r[0].Name, JsonRequestBehavior.AllowGet);
270 }
271
272
273 public JsonResult fnUpdatePersona(RegPersonaBE oRegPersonaBE)
274 {
275     try
276     {
277         oRegPersonaBE.iOpSp = 1;
278         oRegPersonaBE.biIdRegPersona = SessionUsuario.GetIdPersona();
279         oRegPersonaBE.iIdRegUsuaRegistra = SessionUsuario.GetIdUsuario();
280
281         string pss = encriptarPassword(oRegPersonaBE.cCodTipoDocumento, oRegPersonaBE.nvRazoSocial.ToUpper());
282
283
284         return Json(RegPersonaBLI.GetInstance.fnActualizarRegPersona(oRegPersonaBE), JsonRequestBehavior.AllowGet);
285     }
286     catch (Exception ex)
287

```

Figura N°4.23

### 4.3.8 REPORTE DE PRUEBA UNITARIA

Tabla N° 4.86

#### Reporte de pruebas unitarias

Nº HU	NOMBRE DE CLASE/MÉTODO	RESULTADO
1	Tutor/Registrar	Satisfactorio
1	Tutorado/Registrar	Satisfactorio
3	Objetivos/Registrar	Satisfactorio

4	Talleres/Registrar	Satisfactorio
5	Asignación de tutorados/Registrar	Satisfactorio
7	Registrar notas/Registrar	Satisfactorio

### 4.3.9 REPORTE DE PRUEBAS DE ACEPTACIÓN

Tabla N° 4.87

#### Reporte de pruebas de aceptación – Registrar Tutor

PRUEBA DE ACEPTACIÓN	
Caso de prueba: El sistema valida los campos del formulario y realiza un registro correcto de un tutor	
Numero de prueba: 1	Numero historia de usuario: 1
Nombre de caso de prueba: Registrar correctamente los datos del tutor	
Descripción: Se realiza el registro de datos del tutor.	
Condiciones de ejecución: Autenticarse en el sistema como usuario administrador	
Entradas: El usuario accede al sistema como administrador El usuario llena los campos correspondientes al tutor El usuario hace clic en el botón Guardar	
Resultados esperados: Se registra la información correspondiente a un tutor. El sistema despliega un mensaje de confirmación del registro. El sistema muestra los datos registrados en una tabla de resultados.	
Evaluación: El tutor fue registrado con éxito.	

**Tabla N° 4.88****Reporte general de pruebas de aceptación**

N° HU	HISTORIA DE USUARIO	N° C.P	RESULTADOS ESPERADOS	RESULTADO DE LA PRUEBA
1	Administrar tutor	1	Registro y mantenimiento de tutores	Exitosa
2	Administrar tutorado	2	Registro y mantenimiento de tutorados	Exitosa
3	Administrar objetivos del tutor	3	Registro y mantenimiento de objetivos del tutor	Exitosa
4	Administrar actividades y talleres del tutor	4	Registro y mantenimiento de actividades y talleres	Exitosa
5	Asignación de tutorados	5	Asignación de tutorados para un tutor.	Exitosa
6	Registrar cursos del tutorado	6	El tutorado se inscribe en los cursos	Exitosa
7	Registrar notas del taller	7	Registro de notas de los talleres	Exitosa
8	Registrar evaluación de Objetivos	8	Registrar calificación de objetivos según cumplimiento	Exitosa
9	Administrar usuarios	9	Registro y	Exitosa

---

mantenimiento  
de usuarios

---

#### 4.3.10 RESULTADOS DE LA FUNCIONALIDAD DEL SOFTWARE

La funcionalidad de la aplicación Web para el sistema de Tutoría de la universidad, se evaluó mediante la prueba a los actores del sistema tutorial.

**Tabla N° 4.90**

##### **Resultados de la evaluación de la funcionalidad del software**

Pregunta	Cantidad		Porcentaje	
	Sí	No	Sí	No
¿El software le permite registrar a los tutores?	10	0	100%	0%
¿El software le permite modificar los datos de un tutor?	10	0	100%	0%
¿El software le permite registrar a los tutorados?	10	0	100%	
¿El software le permite modificar los datos de los tutorados?	10	0	100%	
¿El software le permite registrar los tutorados asignados a un tutor ?	10	0	100%	
¿El software le permite el registro de la evaluación de objetivos del tutor?	10	0	100%	

¿El software le permite registrar sus objetivos propuestos para un determinado semestre académico?	10	0	100%	0%
¿El software le permite modificar sus objetivos propuestos para un determinado semestre académico?	10	0	100%	0%
¿El software le permite registrar sus actividades y talleres planteados para el semestre académico?	10	0	100%	0%
¿El software le permite modificar sus actividades y talleres planteados para el semestre académico?	10	0	100%	0%
¿El software le permite registrar notas de la participación de los estudiantes en los respectivos talleres?	10	0	100%	0%
¿El software le permite registrar sus cursos matriculados durante el semestre actual?	10	0	100%	0%

¿El software le permite visualizar información publicada por su tutor?	0	10	100%	0%
¿Hay más funcionalidades que quisiera agregar al software?	6	4	60%	40%
¿Qué funcionalidades quisiera agregar al software?	Un chat en línea con el tutor, poder subir imágenes o archivos para compartir con mis compañeros, poder conversar con los otros tutorados mediante el programa.			



## **CAPITULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 CONCLUSIONES**

- a Se logró el objetivo específico, los artefactos presentados que se muestran en las tablas del 2.1 al 2.6
- b Se logró el objetivo específico de obtener resultados del análisis de la aplicación web mediante las fases de la Programación Extrema.
- c Se logró determinar los resultados del diseño de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga, los cuales se muestran en las Figuras 4.1, 4.2; las Tablas 4.16 a la Tabla 4.69, las Tablas 4.70 a la Tabla 4.77.
- d Se logró determinar los resultados de la implementación de la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga, los cuales se muestran en las Figuras 4.2 a la Figura 4.11 y las Figuras 4.13 a la Figura 4.23.
- e Se logró determinar los resultados de las pruebas a la aplicación web para la gestión del Sistema de Tutoría de la Universidad Nacional de San Cristóbal de Huamanga, los cuales se muestran en las Tablas 4.86, 4.87 y Tabla 4.88.

#### **5.2 RECOMENDACIONES**

- a Desarrollar una aplicación móvil orientado a brindar información sobre la Tutoría al Estudiante cuyas funcionalidades abarcan a los docentes Tutores y Estudiantes Tutorados.
- b Se recomienda continuar con el Proyecto Hatun Ñan, sin embargo no existe presupuesto para la continuidad, entonces hacer un estudio que permita obtener Recursos Humanos y Financiamiento.

## BIBLIOGRAFIA

- Álvarez, P. (2002) La función tutorial en la Universidad. Una apuesta por la mejora de la calidad de la enseñanza. Madrid: EOS.
- Barry, C. y Lang, M. (2001). A Survey of Multimedia and Web Development Techniques and Methodology Usage. IEEE Multimedia.
- Batini, N (1994). Diseño Conceptual de Base de Datos. Primera Edición, Addison-Wesley Iberoamericana, S.A, UUEE.
- Beck, K. (1999) Extreme programming explained. Primera edición. Editorial Addison-Wesley urb Co. Estados Unidos.
- Campderrich, B. (2003). Ingeniería del software. Barcelona, España: Editorial UOC.
- Campderrich, B. (2003). Ingeniería del software. Barcelona, España: Editorial UOC.
- Bernal, C. (2006). Metodología de la investigación. Para administración, economía, humanidades y ciencias sociales (2da. Ed.). México: Pearson educación.
- Carrasco, S. (2005). Metodología de la investigación científica: Pautas metodológicas para diseñar y elaborar el proyecto de investigación, Aplicaciones en educación y otras ciencias sociales (2da. Ed.). Lima, Perú. Editorial San
- Castillo I., Dick A. (2010) Sistema de gestión integral vía web en empresas de transporte público en Lima Metropolitana subsistema caja- operaciones-almacén. Universidad Nacional Mayor de San Marcos, Lima, Perú.
- Directiva S/N 2013-VRAC-UNSCH. Lineamientos Para la Implementación del Sistema de Tutoría en las Escuelas de Formación Profesional. Universidad Nacional de San Cristóbal de Huamanga. Ayacucho, Perú 2013.
- DSIIC-Universidad Politécnica de Valencia. (2006). Metodologías Ágiles para el Desarrollo de Software: Extreme Programming (XP). Valencia, España.
- Eckel, B. (2002). Piensa en Java (2da. Ed.). Madrid, España: Pearson Educación S.A.
- Ferrer, V. (2003). La acción tutorial en la Universidad. En Michavila, A y García Delgado, J. (coord.). La tutoría y los nuevos modos de aprendizaje en la Universidad (pp.. 67-84). Madrid: Consejería de Educación de la Comunidad de Madrid y Cátedra UNESCO de Gestión política Universitaria de la Universidad Politécnica de Madrid.
- Gómez C., Martha E. (2005) Sistema Inteligente para la Tutoría Académica de la Uaem
- Holzner, S. (s.f.). Java 2. Coriolis.

- Joyanes, L. y Zahonero, I. (2002). Programación en Java 2. Algoritmos, Estructura de Datos y Programación Orientada a Objetos. Madrid, España: McGraw-Hill/Interamericana S.A.U.
- Luján, S. (2001). Programación en internet: Clientes WEB. España. Editorial Club Universitario.
- Pantigoso, J. (2008). Base de Datos con SQL Server 2008. Grupo Editorial Megabyte S.A.C.
- Patrick, N. (1996). Manual Java. Primera Edición. Mcgraw-Hill Interamericana De España, S.A. Madrid, España.
- Pressman, R. (2002). Ingeniería del Software. Un enfoque práctico (5ta. Ed.). Madrid, España: McGraw-Hill/Interamericana S.A.U.
- Rincón Igea, Benito(2000), Tutorías personalizadas en la universidad, , Ediciones de la Universidad de Castilla – La Mancha, Cuenca.
- Rob, P. y Coronel, C. (2004). Sistema de Base de Datos, Diseño Implementación y Administración. Primera Edición. México. Editorial Thomson Editores S.A.
- Rodríguez E., S. et al. (2004). Manual de tutoría universitaria. Recursos para la acción. Barcelona: Octaedro- ICE U. de Barcelona
- Rodriguez T., Oscar J. (2003), Diseño y Experimentación de un Sistema de Tutoría para la FIQ-UNAC
- Senn, J. (1997). Análisis y Diseño de Sistemas de Información (2da Ed.). Colombia: McGraw-Hill.
- Sommerville, I. (2005). Ingeniería del software (7ma. Ed.). Madrid, España: Pearson Educación S.A.
- Wake, W. C. (2000) Extreme Programming Explored.
- Weitzenfeld, A. (2005). Ingeniería de software orientado a objetos con UML, Java e Internet. México. Thomson Editores S.A.
- Zavala P, C.(2003), Guía práctica para la tutoría grupal, César, Ediciones Universidad Autónoma de Aguascalientes. México.