

**Universidad de las Ciencias Informáticas**

**Facultad 15**



**Título: “Evaldes”**

**Aplicación informática de apoyo al Sistema de Pago  
Adicional Centrado en la Evaluación por Desempeño.**

**Trabajo de Diploma para optar por el título de Ingeniero Informático.**

**Autor: Pavel Pérez Sierra.**

**Tutor: Lic. Rolan Rober Bullain Dieguez**

**Co-tutor: Ing. Abdiel Matos Nieto**

Ciudad de La Habana, Cuba

Junio de 2010

## Pensamiento

---



*“Educar no es dar carrera para vivir, sino templar el alma para las dificultades de la vida”*

*Pitágoras*

# Declaración

---

## DECLARACIÓN DE AUTORÍA

Yo Pavel Pérez Sierra, declaro ser el único autor del trabajo titulado:

“Evaldes”, Aplicación informática de apoyo al *Sistema de Pago Adicional Centrado en la Evaluación por Desempeño*.

y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Pavel Pérez Sierra

---

Lic. Rolan Rober Bullain Dieguez

---

Ing. Abdiel Matos Nieto



# Agradecimientos

---

## AGRADECIMIENTOS

*Quiero agradecer primeramente a la Revolución por haberme permitido realizar uno de mis principales sueños, ser un profesional que pueda aportar mis conocimientos en el bien de la sociedad.*

*Por otra parte quisiera agradecer a mi tutor Rolan que cuando todo parecía poco probable me demostró que si podía terminar mis estudios ahora junto a mis compañeros, a mi co-tutor Abdiel que también estuvo pendiente de mi progreso en el trabajo, a mi madre que junto a mi esposa e hijo han sido la fuente de mis fuerzas para seguir adelante, a mis amigos que juntos hemos vivido esta etapa importante de nuestras vidas.*

*A todos muchas gracias...*

## Dedicatoria

---

### DEDICATORIA

*Dedico este Trabajo de Diploma a toda mi familia que me ha apoyado en todos los momentos de mi vida, mi madre que tanto se ha sacrificado por mí y por todos mis hermanos, a mi tío que se ha portado como un padre para todos nosotros, a mi esposa Dayra que ha sabido esperar por mí todos estos años y me ha apoyado incondicionalmente, a mi hermano que pronto estará discutiendo su trabajo de diploma, a mi padre que aunque no está presencialmente conmigo siempre lo llevo en mis pensamientos y en mi corazón, a mi abuelita que tanto la quiero.*

*También quisiera agradecer a todas las personas que alguna vez me cerraron las puertas pues hicieron que por cada golpe que recibí en la vida supiera levantar la cabeza, seguir adelante y llegar hasta donde estoy hoy.*

# Resumen

---

## RESUMEN

Una de las características singulares que posee la Universidad de Ciencias Informáticas (UCI, en lo adelante) es el sistema de estimulación a los profesores y trabajadores, para la cual fue diseñado un sistema de pago adicional centrado en la evaluación por desempeño de las actividades asignadas a cada profesor o trabajador. Este sistema tiene la deficiencia de ser implementado por los evaluadores de distintas formas, para lo cual tienen en cuenta su experiencia y rigor al evaluar. No existe una estandarización en la forma de controlar las evaluaciones a las tareas, actividades y responsabilidades.

El trabajo que se presenta a continuación es una aplicación de apoyo a este proceso, el cual persigue el objetivo de contribuir en la gestión de la evaluación de desempeño de los profesores y trabajadores de la facultad 15, proporcionándoles a los jefes de área una evaluación por cada profesor y permitiéndoles que puedan aceptar o rechazar dicha evaluación. Para ello se realiza un análisis de los sistemas encontrados con características similares a la solución propuesta, permitiendo agilizar el proceso de desarrollo del producto.

# Índice

---

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>4</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>7</b>
1.1- EVALUACIÓN DEL DESEMPEÑO .....	7
1.1.1 Métodos de evaluación del desempeño .....	8
1.2- APLICACIONES WEB .....	10
1.3- ANÁLISIS DE SOLUCIONES EXISTENTES .....	11
1.4- METODOLOGÍAS DE DESARROLLO.....	12
1.4.1- Metodologías ágiles .....	13
1.4.1.1- Extreme Programming .....	14
1.4.1.2- SCRUM.....	15
1.5- SISTEMA GESTOR DE CONTENIDOS .....	17
1.6.1- Drupal .....	17
1.6.2- Joomla .....	18
1.6- LENGUAJES .....	19
1.5.1- PHP 5.0 .....	19
1.5.2- Javascript .....	20
1.5.3- UML.....	20
1.5.4- SQL .....	20
1.7- HERRAMIENTAS CASE.....	21
1.7.1- Rational Rose.....	21



# Índice

---

1.7.2- Visual Paradigm .....	22
1.8- SISTEMA GESTOR DE BASE DE DATOS .....	22
1.8.1- MySQL.....	23
1.9- ENTORNO DE DESARROLLO INTEGRADO PARA APLICACIONES WEB.....	23
1.9.1- Zend Studio .....	24
1.9.2- Aptana Studio.....	24
CONCLUSIONES PARCIALES.....	25
<b>CAPÍTULO 2: LA APLICACIÓN INFORMÁTICA “EVALDES” .....</b>	<b>26</b>
2.1- REQUISITOS DEL SISTEMA. ....	26
2.1.1- Requisitos funcionales .....	26
2.1.2- Requisitos no funcionales.....	30
2.2- PROPUESTA DEL SISTEMA. ....	31
2.2.1- Personas relacionadas con el sistema. ....	32
2.2.2- Prototipo de Interfaz.....	33
2.3- HISTORIAS DE USUARIOS.....	33
2.4- PLANIFICACIÓN DE LAS ITERACIONES (SPRINT).....	36
2.5- DISEÑO.....	37
2.5.1- Componentes de Joomla. ....	37
2.5.2- Componentes implementados. ....	38
2.6- PATRONES DE DISEÑO UTILIZADOS. ....	38
2.6.1- Modelo-Vista-Controlador.....	39





# Índice

---

2.7- DESARROLLO DE LOS COMPONENTES DEL SISTEMA.....	41
2.7.1- <i>Tareas de Ingeniería</i> .....	41
2.7.2- <i>Diagrama Modelo de Datos</i> .....	46
2.7.3- <i>Diagrama de Despliegue</i> .....	47
2.8- BENEFICIOS DE LA IMPLEMENTACIÓN DEL SISTEMA.....	47
2.9- VALIDACIÓN DEL SISTEMA.....	47
2.9.1- <i>Pruebas unitarias</i> .....	48
2.9.2- <i>Pruebas de aceptación</i> .....	49
2.9.2- <i>Evaluación por criterio de expertos</i> .....	50
CONCLUSIONES PARCIALES.....	52
<b>CONCLUSIONES GENERALES.....</b>	<b>53</b>
<b>RECOMENDACIONES.....</b>	<b>54</b>
<b>BIBLIOGRAFÍA.....</b>	<b>55</b>
<b>ANEXOS.....</b>	<b>59</b>
<i>Anexo A: Comparación de las metodologías ágiles y las metodologías tradicionales</i> .....	59
<i>Anexo B: Pruebas de Aceptación</i> .....	60
<i>Anexo C: Instrumento aplicado para la validación mediante Criterio de Expertos</i> .....	69
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>74</b>



# Capítulo 1

---

## INTRODUCCIÓN

El mundo avanza cada día a una velocidad mayor, las nuevas tecnologías quedan obsoletas en breves periodos de tiempo, soluciones que antes hubiesen llevado a la humanidad miles de años para su sustitución hoy apenas dan cabida para actualizarse.

El uso irremplazable de las Tecnologías de la Informática y las Comunicaciones (TIC) se está haciendo cada vez más evidente. El desarrollo de las mismas permite que los procesos de la humanidad se realicen con más calidad, control y velocidad, siendo posible acortar innumerables situaciones complicadas en la sociedad y aumentar la fiabilidad y efectividad de los bienes y servicios que brindan en la actualidad el uso de las TIC.

Dentro de estas nuevas tecnologías juegan un papel fundamental los sistemas informáticos que se encargan de gestionar, mostrar y actualizar la información, entre las cuales se encuentran las aplicaciones Web que cada vez tienen una exigencia mayor debido a su constante evolución. Muchos de estos sistemas están orientados a la esfera educacional, la cual junto al desarrollo de software vive cambios vertiginosos y se adapta a ellos.

El país no se queda atrás en esta nueva visión social, y a pesar de ser un país bloqueado por más de 50 años ha logrado adentrarse en la nueva era, la "era de la informática". La misión de informatizar al país es cada vez más prioritaria, y sobre esta base se creó la Universidad de Ciencias Informáticas (UCI), para la cual la formación y la superación personal tanto de sus estudiantes como de sus profesores y trabajadores, es fundamental.

La universidad posee características muy singulares las cuales la hacen una universidad reconocida en todo el territorio cubano. Dentro de estas características una de la más notable es la forma en que se estimula el trabajo y resultados de sus profesores y trabajadores para que su desempeño sea cada vez mejor y este a la altura de los docentes que necesita una universidad de excelencia. Para ello la UCI ha definido un sistema de pago adicional centrado en la evaluación por desempeño, el cual a grosso modo consiste en proporcionar un pago adicional al salario del profesor dependiendo de cómo se ha desempeñado en las actividades que tiene que realizar de acuerdo a su plan de trabajo.

# Capítulo 1

---

El sistema de pago adicional (PA, en lo adelante) define tres categorías y varios grupos que están estrechamente relacionados:

*Deficiente:* el profesor no tiene derecho al PA.

*Adecuado:* el profesor tiene derecho al PA y la cantidad dependerá del grupo en el que se encuentra.

*Superior:* el profesor tiene derecho al PA mucho mayor que el pago recibido cuando el profesor recibe la categoría de adecuado, la cuantía también dependerá del grupo en el que se encuentra.

Debido a que el sistema de PA es un estímulo a los profesores y trabajadores de la universidad teniendo en cuenta su aporte a la economía del país, el trabajo que tienen que realizar y la cantidad de actividades de acuerdo a su plan de trabajo mensual, se decidió por la dirección general de la universidad definir un conjunto de requisitos los cuales pueden invalidar que el profesor obtenga el PA de una u otra categoría.

Para ello el sistema de PA demanda un estricto sistema de control para que cumpla el objetivo con el que fue creado: una herramienta de estímulo al buen desempeño. Pero la realidad es que no se ha logrado cumplir con el objetivo, pues como la evaluación es un proceso cualitativo han existido múltiples discrepancias por las diferentes evaluaciones emitidas por los evaluadores y jefes de área, esto viene dado según el análisis de las encuestas realizadas, por:

- Inexperiencia de algunos evaluadores.
- Diversidad de criterios, control y exigencia por parte de los evaluadores.
- Especulación de algunos profesores sobre favoritismo de la evaluación.
- Inexistencia de registro histórico del desempeño profesoral
- Diferencia en la carga individual del profesor y su desempeño.

Teniendo en cuenta la situación problemática antes planteada se define el siguiente **problema**: el proceso de evaluación de desempeño para el pago adicional es impreciso, presenta divergencia en su aplicación y es susceptible a errores.

# Capítulo 1

---

Para darle solución al problema antes planteado se identifica como **objeto de estudio**: el proceso de desarrollo de software enmarcado en el **campo de acción** de las aplicaciones informáticas de apoyo a la toma de decisiones para la evaluación de desempeño.

Definiendo como **objetivo general**: desarrollar una aplicación informática que asista el pago adicional mediante la evaluación de desempeño de los profesores y trabajadores de la facultad 15.

**Hipótesis**: el desarrollo de una aplicación informática que gestione, registre, estandarice parámetros y apoye la toma de decisiones contribuirá a mitigar los problemas que hoy presenta el pago adicional mediante la evaluación de desempeño.

Para el desarrollo del mismo se cuenta con una serie de **tareas** que ayudarán a cumplimentar la construcción del mismo:

- Análisis de otros sistemas con características similares al sistema a desarrollar.
- Documentación de las herramientas y tecnologías a utilizar.
- Selección y fundamentación de una metodología de desarrollo acorde a la magnitud del proyecto.
- Levantamiento de requisitos funcionales del sistema.
- Desarrollo del sistema propuesto.
- Validación del sistema propuesto.

El documento se estructuró en dos capítulos para dar solución al problema planteado en la investigación.

**Capítulo No 1**: constituye la base teórica del presente trabajo. Se realiza un estudio del estado del arte, se describen y justifican las tecnologías, paradigmas de programación, herramientas, metodologías y lenguajes a utilizar para el proceso de desarrollo.

**Capítulo No 2**: constituye el desarrollo de la aplicación “Evaldes” y se describen los elementos esenciales que permitieron la elaboración de la aplicación y su validación mediante el criterio de expertos.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.**

### **Introducción**

En el desarrollo de este capítulo se realizará un estudio de las diferentes aplicaciones web utilizadas tanto en el ámbito internacional como dentro del país, se abordará sobre las soluciones existentes que tienen semejanza con nuestro sistema, paradigmas de programación actuales, además de las tendencias y tecnologías a utilizar, así como la metodología de desarrollo, las herramientas y el lenguaje de modelado que serán utilizados.

### **1.1- Evaluación del Desempeño**

Para un mejor entendimiento de la importancia del sistema que se quiere desarrollar para la evaluación del desempeño de los profesores de las diferentes áreas, se dan respuestas a diferentes interrogantes que pueden surgir en cuanto al tema en cuestión, además se destacarán algunos métodos de evaluación por desempeño.

#### **¿Qué es evaluación del desempeño?**

Según el criterio de Fuchs (Fuchs, 1997) un sistema de evaluación de desempeño es el conjunto de mecanismos que permite definir el grado en que las personas contribuyen al logro de los estándares requeridos para el cargo o puesto que ocupan en la organización, tanto como para los objetivos de la empresa. Facilita las acciones necesarias para su desarrollo profesional y personal, así como para aumentar su aporte futuro.

En el ámbito educacional, Valdés (Váldez, 2000) señala que la evaluación de desempeño de un profesor es un proceso sistemático de obtención de datos válidos y fiables, con el objetivo de comprobar y valorar el efecto educativo que produce en los alumnos el despliegue de sus capacidades pedagógicas, su emocionalidad, responsabilidad laboral y la naturaleza de sus relaciones interpersonales con alumnos, padres, directivos, colegas y representantes de las instituciones de la comunidad.

# Capítulo 1

---

## **¿Por qué evaluar el desempeño?**

Según Solaas (Solaas, 2010) una evaluación del desempeño justa y fundamentada es una necesidad para toda persona que trabaja como empleado en una organización. Además, la evaluación del desempeño y la comunicación de la misma es un excelente instrumento para el desarrollo de los subordinados en sus puestos de trabajo actuales. Los ayuda a comprender tanto sus fortalezas como sus debilidades, y el modo en que pueden superar a estas últimas.

Según Aylwin (Aylwin, 2001) es imprescindible evaluar el sistema educativo con detenimiento externo e internamente, sus resultados y sus procesos, para poder tomar las medidas oportunas y mejorar cualitativamente su funcionamiento y sus resultados. Para conseguir que los conocimientos de las personas sean explícitos y compartidos y generen valor para la organización escolar, es decir, sean el “Capital Intelectual” de la organización es necesario crear un ambiente en el que la organización escolar y las personas se comprometen en un proceso de aprendizaje continuo.

### **1.1.1 Métodos de evaluación del desempeño**

A continuación se explicarán dos grupos de métodos para la evaluación del desempeño con el objetivo que puedan servir de base para el desarrollo del sistema en cuestión.

#### **Métodos de evaluación basados en el desempeño durante el pasado.**

Según el criterio del equipo de trabajo de RV (RV,2009) estos métodos tienen la ventaja de versar sobre algo que ya ocurrió y que puede, hasta cierto punto, ser medido. Su desventaja radica en la imposibilidad de cambiar.

1. Escalas de puntuación: el evaluador debe conceder una evaluación subjetiva del desenvolvimiento del empleado en una escala que vaya de bajo a alto.
2. Lista de verificación: requiere que la persona que otorga la calificación seleccione oraciones que describan el desenvolvimiento del empleado y sus características, el evaluador suele ser el supervisor inmediato.

# Capítulo 1

---

3. Método de selección forzada: obliga al evaluador a seleccionar la frase más descriptiva del desempeño del empleado en cada par de afirmaciones que encuentra.
4. Método de registro de acontecimientos críticos: requiere que el evaluador lleve una bitácora diaria, el evaluador consigna las acciones más destacadas que lleva a cabo el evaluado.
5. Escalas de calificación conductual: utilizan el sistema de comparación del desempeño del empleado con determinados parámetros conductuales específicos.
6. Método de verificación de campo: un representante calificado del personal participa en la puntuación que conceden los supervisores a cada empleado. El representante del departamento de personal solicita información sobre el desempeño del empleado al supervisor inmediato.
7. Métodos de evaluación en grupos: los enfoques de evaluación en grupos pueden dividirse en varios métodos que tienen en común la característica de que se basan en la comparación entre el desempeño del empleado y el de sus compañeros de trabajo.
8. Método de categorización: lleva al evaluador a colocar a sus empleados en una escala de mejor a peor.
9. Método de distribución forzada: se pide a cada evaluador que ubique a sus empleados en diferentes clasificaciones.
10. Método de comparación por parejas: el evaluador debe comparar a cada empleado contra todos los que están evaluados en el mismo grupo.

## **Métodos de evaluación basados en el desempeño a futuro.**

Se centran en el desempeño venidero mediante la evaluación del potencial del empleado o el establecimiento de objetivos de desempeño.

El equipo de trabajo de RV (RV,2009) plantea que los métodos de evaluación basados en el desempeño al futuro se pueden categorizar en:

1. Auto evaluaciones: llevar a los empleados a efectuar una auto evaluación puede constituir una técnica muy útil, cuando el objetivo es alentar el desarrollo individual.

# Capítulo 1

---

2. Administración por objetivos: consiste en que tanto el supervisor como el empleado establecen conjuntamente los objetivos de desempeño deseables.

3. Evaluaciones psicológicas: cuando se emplean psicólogos para las evaluaciones, su función esencial es la evaluación del potencial del individuo y no su desempeño anterior.

4. Métodos de los centros de evaluación: son una forma estandarizada para la evaluación de los empleados que se basa en tipos múltiples de evaluación y múltiples evaluadores.

Luego de analizar los dos grupos de métodos de evaluación, se verificó que el sistema de evaluación de desempeño implementado por la universidad utiliza el método de “Escalas de Puntuación”, teniendo en cuenta de que la evaluación ocurre luego de que el profesor haya culminado las tareas asignadas, y que además las categorías están determinadas por un nivel de puntuación.

## 1.2- Aplicaciones Web

Una aplicación web según Ailon (Ailon, 2010) es una página web con contenido dinámico, es decir, cambia su contenido según la interacción de cada usuario que la visita.

### **Tipos de aplicaciones web:**

Según el criterio de Álvarez (Álvarez, 2004) las aplicaciones web se dividen en tres grupos fundamentales: Informativas, de negocio y portales.

- *Las aplicaciones web informativas:* son las más sencillas de realizar. En las que sólo se muestra información de un tema determinado, donde la página principal es la de mayor importancia a la hora de su elaboración, ya que es la que da la idea del contenido que se encontrará en la aplicación.
- *Las aplicaciones web de negocio:* incluyen las aplicaciones informativas y opciones donde el cliente selecciona lo que necesita de un producto o servicio, en esta el intercambio de información es más complejo, en su forma más sencilla, luego de la selección de los elementos que el usuario necesita, el sistema le pide los datos y los almacena en la base de datos con toda la información ingresada por el usuario.



# Capítulo 1

---

- *Los portales*: son el grupo más complejo de las aplicaciones web. Un portal de Internet es conocido como “Puerta o puerto”, es decir, es un sitio web que sirve como punto de partida para la navegación en la red. Posibilitando de forma fácil y sencilla, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse facilidades para la localización de sitios como son los buscadores, noticias, buzones de correo, directorios telefónicos y foros de discusión con temas de interés general.

El sistema que se quiere desarrollar se encuentra dentro de las aplicaciones web de negocio debido a que lo que se quiere es realizar una aplicación web que permita gestionar el cumplimiento de las actividades y evaluar a los profesores en las diferentes áreas, donde las personas designadas puedan poner una evaluación de acuerdo al desempeño del profesor, y que el jefe de área pueda obtener a través de la aplicación una posible evaluación general del trabajador.

## 1.3- Análisis de soluciones existentes

En la actualidad ya son varios los sectores que utilizan aplicaciones informáticas para controlar la gestión de la información. En ello las aplicaciones web juegan un papel fundamental y están desplazando a las aplicaciones de escritorio debido a las ventajas que poseen para este tipo de negocio.

A continuación se realiza un análisis de varios sistemas informáticos que realizan actividades semejantes a las que se quiere automatizar.

*Sistema de Evaluación del Desempeño (SED)*: es un sistema creado para la Secretaría de Hacienda y Crédito Público (SHCP) de México, el cual según Jácome (Jácome, 2009) “es una estrategia para configurar un esquema eficaz para la toma de decisiones presupuestarias que considera información objetiva respecto del diseño, pertinencia, estrategia, operación y resultados de las políticas y programas públicos. Los principales componentes del SED serán un conjunto de evaluaciones a los programas, a las instituciones y a las políticas públicas, así como un agregado de acciones estratégicas para mejorar el quehacer de la administración pública, en donde ambos elementos tomen como referente fundamental el beneficio y satisfacción de la sociedad”.

# Capítulo 1

---

Este sistema posee características ideales para evaluar el desempeño de las instituciones pero no está orientado a la evaluación profesoral, razón por la cual queda descartado como posible solución a la problemática planteada.

*Sistema para el control de la información y evaluación de los profesores en los departamentos docentes de la facultad 8:* consiste en un software donde se controla la información de los profesores y medios básicos que tiene cada departamento de la facultad 8, dándole la posibilidad a los jefes de departamento de evaluar a sus profesores de una forma más eficiente y fácil.

Este sistema nos brinda algunas de las características deseadas, pero tiene como dificultad que es una herramienta en la cual no existen diferentes áreas sino está implementado para el uso de los jefes de departamentos, por otra parte no brinda la posibilidad de que el evaluador conozca los indicadores que debe tener en cuenta al evaluar un aspecto determinado, además de no posibilitar que los profesores puedan conocer la evaluaciones de los demás compañeros de trabajo. Debido a lo planteado anteriormente, este sistema solo será tomado como objeto de estudio permitiendo agilizar algunos procesos.

## **1.4- Metodologías de desarrollo**

Una metodología de desarrollo es un medio de estandarización que cubre completamente el proceso de desarrollo de un software. Posibilita el movimiento en doble sentido entre las fases del ciclo de desarrollo para verificar trabajos realizados y corrección de los errores detectados. Provee un entendimiento efectivo entre los analistas, programadores, clientes, usuarios y gestores para una correcta toma de decisiones (Mendoza, 2004).

Las metodologías tradicionales según Pressman (Pressman, 2008) imponen una disciplina de trabajo sobre el proceso de desarrollo de software, con el objetivo de conseguir un software más eficiente y predecible. Para ello, se hace un especial hincapié en la planificación total de todo el trabajo a realizar y una vez que esté todo detallado, comienza el ciclo de desarrollo del producto software.

Este tipo de metodología se basa en realizar el proceso completo del desarrollo de software por lo que hace que sea un proceso más lento. Considerando lo antes expuesto y teniendo en presente que el

# Capítulo 1

---

tiempo que se tiene para realizar el producto es en un período menor de un año, no se considera este tipo de metodología para modelar nuestro producto.

Para una mejor comprensión del argumento planteado anteriormente véase **Tabla 1** en el **Anexo A**.

## 1.4.1- Metodologías ágiles

Estas metodologías se caracterizan por tener un desarrollo incremental para producir tempranamente pequeñas entregas en ciclos rápidos, predisposición para el cambio y la adaptación continúa. Estas metodologías por lo general se centran en desarrollar productos funcionales más que en conseguir una buena documentación (Letelier, 2004).

Dentro de estas metodologías se encuentran Programación Extrema (XP, por sus siglas en inglés), SCRUM, Crystal, Método de Desarrollo de Sistemas Dinámicos (DSDM, por sus siglas en inglés). Para el desarrollo de este trabajo solo serán analizadas XP y SCRUM por el éxito y prestigio alcanzado dentro de esta categoría.

Las características de las metodologías ágiles según Qualitrain (Qualitrain, 2010) son:

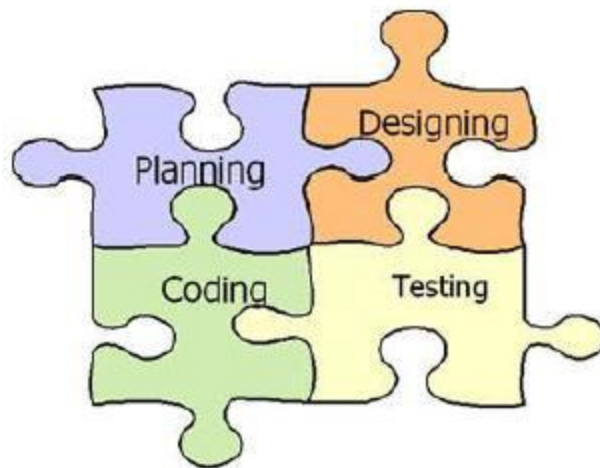
- Proceso iterativo e incremental.
- Mitigación del riesgo mediante iteraciones fijas.
- Mejora continua.
- Calidad desde el primer día.
- Priorización de requerimientos de acuerdo a su valor.
- Equipos dedicados y auto-gestionados.
- Incorporar al cambio.
- Prácticas de desarrollo modernas.

# Capítulo 1

---

## 1.4.1.1- Extreme Programming

*Figura 1 Metodología XP*



*Fuente: Tomado de (Fowler, 2000).*

La Programación Extrema XP (Extreme Programming, en inglés), es una metodología ágil nacida en el verano de 1996 de la mano de Kent Beck cuando éste se incorpora a Chrysler Corporation para desarrollar una aplicación de nóminas, donde las condiciones existentes conducían a este tipo de metodologías no convencionales que él propuso aplicar. La misma se encuentra centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Le da prioridad a las tareas que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación (Fowler, 2000).

Según Mendoza (Mendoza, 2004) XP propone:

# Capítulo 1

---

- Empezar en pequeño y añadir funcionalidades con retroalimentación continua.
- Manejar el cambio para que se convierta en parte sustantivo del proceso.
- Independizar el costo del cambio de la fase o etapa.
- No introducir funcionalidades antes que sean necesarias.
- Convertir el cliente o el usuario en miembro del equipo.

XP posee características ideales para una rápida programación, características deseadas y necesarias para el desarrollo de este trabajo. Considerando el alcance de la aplicación y el período que se tiene para el desarrollo de la misma, se decide seleccionar esta metodología como guía para el desarrollo del producto.

## 1.4.1.2- SCRUM

Es una metodología ágil de administración de proyectos, entre ellos, proyectos de desarrollo de software. Con Scrum los proyectos progresan a través de una serie de iteraciones que duran entre una o dos semanas y un mes, llamadas "Sprint". No es una metodología de análisis, ni de diseño, como podría ser RUP, es una metodología de gestión del trabajo (García, 2007).

Según García (García, 2007) *los artefactos que genera esta metodología son:*

### **Sprint.**

- Es la base del desarrollo Scrum.
- Su duración máxima es de 30 días.
- Se llevan a cabo las tareas pre-establecidas y no se puede modificar el trabajo acordado en el backlog.
- Sólo el ScrumMaster puede abortar un sprint si lo considera no viable por alguna de las siguientes razones:
  - Las circunstancias del negocio han cambiado.
  - La tecnología acordada no funciona.
  - El equipo ha tenido interferencias.

# Capítulo 1

---

## **Product Backlog.**

- Crea un listado con los requisitos de los usuarios o propietarios del sistema para planificar el proyecto.
- No es una lista completa y definitiva. Es sólo una estimación inicial de los requisitos.
- Es un documento dinámico que incorpora las constantes necesidades del sistema y se mantiene durante todo el ciclo de vida (hasta la retirada del sistema).

## **Sprint Backlog.**

- Especifica la serie de tareas que se van a desarrollar según los requisitos señalados.
- Estas tareas tienen una duración de entre 4 y 16 hrs. de trabajo.
- Las de mayor duración intentar descomponerlas en Sub-Tareas dentro de ese rango de tiempo.
- Al final del sprint se busca un incremento en la funcionalidad

Debido a que la aplicación deseada tiene gran importancia, y que Scrum brinda una buena gestión de la información, se decide tomar esta metodología como guía para el desarrollo de la misma.

Finalmente, analizando la posibilidad de que estas dos metodologías se pueden unir, podemos conformar una metodología ágil más completa. Scrum nos brindaría una mejor gestión del trabajo y XP nos ayudaría para que la programación sea más rápida, de esta manera garantizando que el producto tenga una mejor calidad.

La unión de estas metodologías está dividida en cuatro fases, estas son:

- Planificación-Definición
- Desarrollo
- Entrega
- Mantenimiento

Cada una de estas fases está compuesta por una serie de actividades que son las que generan los artefactos que quedan incluidos en el nuevo expediente de proyecto.

- Historias de Usuarios.

# Capítulo 1

---

- Tareas de Ingeniería.
- Pruebas.

## 1.5- Sistema Gestor de contenidos

Un Sistema de gestión de contenidos (CMS, por sus siglas en inglés) es un software que permite crear una estructura de soporte (framework) para la creación y administración de contenidos por parte de los participantes principalmente en páginas web (García, 2007)

Actualmente existen muchos CMS según sus funciones o finalidad: noticias, blog, comercio electrónico, entre otras. Según García (García, 2007):

*Joomla*: está programado en PHP para Bases de Datos MySQL.

*Mambo*: programado en PHP para Bases de Datos MySQL.

*Drupal*: está programado en PHP para Bases de Datos MySQL y PostgreSQL.

*Plone*: Plone sobre Zope, programado en Python para todo tipo de BD.

*Muravey-Web*: programado en Ruby sobre Rails para todo tipo de BD.

*Typo3*: programado en PHP4 para Bases de Datos MySQL

La aplicación a desarrollar requiere de tecnología multiplataforma y de código abierto para garantizar su utilidad posterior. Por lo anterior señalado se ha considerado solo el análisis de aquellos que brindan soporte para el desarrollo de aplicaciones web, que sean de código abierto. Resultaron los CMS a analizar: Drupal y Joomla.

### 1.6.1- Drupal

Drupal es un sistema de gestión de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos (Reyero, 2005).

Reyero (Reyero, 2005) define que algunas de las características fundamentales que posee este CMS son:

# Capítulo 1

---

**Módulos:** La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como “página de categorías” autenticación mediante jabber, mensajes privados, bookmarks, y otros.

**Personalización:** Posee un robusto entorno de personalización implementado en su núcleo. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.

**Multiplataforma:** Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X.

## **1.6.2- Joomla**

Es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL. Es usado para publicar en Internet e intranets utilizando una base de datos MySQL. Posee un número considerable de facilidades, entre ellas: hacer caché de páginas para mejorar el rendimiento, indexar web, feed RSS, versiones imprimibles de páginas, flash con noticias, blogs, foros, encuestas, calendarios, búsqueda en el sitio web, e internacionalización del lenguaje. (LeBlanc, 2007)

A continuación se ofrecen algunas de las características fundamentales definidas por LeBlanc (LeBlanc, 2007):

**Organización del sitio web:** Joomla permite organizar eficientemente los contenidos de un sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Esto también forma parte de la optimización para buscadores.

**Publicación de Contenidos:** Con su uso, se pueden crear páginas ilimitadas. Permitiendo su edición desde un sencillo editor (similar al editor de MS Word) que permite darle a los textos el formato e imágenes deseadas. Los contenidos son totalmente editables y modificables.



# Capítulo 1

---

Escalabilidad e implementación de nuevas funcionalidades: Joomla ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos. Lo que permitirá agregar servicios de valor para los usuarios finales, por ejemplo: galerías de imágenes, foros, clasificados, entre otros.

Administración de usuarios: Joomla permite almacenar datos de usuarios registrados y también te da la posibilidad de enviar e-mails masivos a todos los usuarios. La administración de usuarios es jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades/permisos dentro de la gestión y administración del sitio.

Diseño y aspecto estético: Es posible cambiar todo el aspecto del sitio web tan solo con un par de clicks, gracias al sistema de templates que utiliza Joomla.

Plataformas compatibles: Es compatible con las plataformas Linux, Windows y Mac OSX.

Tras un análisis de los CMS Drupal y Joomla y las ventajas que ambos poseen en cuanto a su portabilidad, gestión de usuarios, de contenidos y disposición de módulos, se concluye que el uso de cualquiera de los mismos, permite el desarrollo de un sistema a la altura de los requerimientos definidos para este trabajo. Se seleccionó Joomla debido a su gran interfaz gráfica configurable, facilidad de aprendizaje y personalización.

A continuación se abordaran las herramientas y lenguajes utilizados para el desarrollo de la aplicación, los cuales fueron descritos teniendo en cuenta el CMS utilizado.

## 1.6- Lenguajes

Según Quintero (Quintero, 2000) un lenguaje de programación es un conjunto de instrucciones entendibles directamente o traducibles al lenguaje del ordenador con el que se trabaje; combinando estas instrucciones realizamos programas.

### 1.5.1- PHP 5.0

Castillo (Castillo, 2005) define que PHP es un lenguaje de propósito general usado para el desarrollo de aplicaciones a web y puede ser embebido dentro de código HTML. Es ejecutado en un web, creando

# Capítulo 1

---

páginas web a partir del código PHP que toma como entrada. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Entre las ventajas fundamentales que este lenguaje posee según Castillo (Castillo, 2005), se encuentran su alto rendimiento, es un lenguaje multiplataforma que posee capacidad de conexión con gran parte de los manejadores de datos existentes (especialmente con MySQL), brinda además la posibilidad de expandir su potencial utilizando los módulos (llamados extensiones), no requiere definición de variables y además posee manejo de excepciones (desde la versión que se va utilizar, php5)

Su condición de lenguaje para el diseño de aplicaciones web, junto a sus facilidades de uso, seguridad y su excelente combinación con herramientas como la base de datos MySQL, además que es el lenguaje de programación que utiliza Joomla, justifican su selección para el desarrollo del presente trabajo.

## **1.5.2- Javascript**

Valdés (Valdés, 2007) precisa que es un lenguaje interpretado creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. El mismo, a pesar de su semejanza con Java, se diferencia de este en que no dispone de herencia. La mayoría de los navegadores en sus últimas versiones interpretan código Java script.

## **1.5.3- UML**

Según Booch (Booch, 2008) el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

## **1.5.4- SQL**

SQL (Structured Query Language, en inglés) es un lenguaje de consulta estructurado y declarativo de acceso a bases de datos relacionales que permite realizar diversas operaciones sobre las mismas. Posee

# Capítulo 1

---

manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma (Casares, 2004).

Se hace uso de este lenguaje para acceder a la base de datos del sistema, dotando el acceso de las ventajas que el mismo posee.

## 1.7- Herramientas Case

Se pueden definir las Herramientas CASE (Computer Aided Software Engineering) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software (Vivas White, y otros, 2009).

Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras (INEI, 1999).

### 1.7.1- Rational Rose

Según la compañía Rational Software Corporation (Rational, 2003) es una herramienta CASE que cubre todo el ciclo de ayuda de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases. Permite trazabilidad real entre modelo (análisis y diseño) y el código.

Entre sus principales características se encuentran que mantiene la consistencia de los modelos de software, diseño centrado en casos de uso y enfocado al negocio, genera documentación automáticamente, genera código a partir de modelos y además tiene capacidad de ingeniería inversa.

Esta herramienta case tiene excelentes características para el modelado de los productos software, pero a su vez posee una inconveniente, su adquisición no es libre por lo que queda descartado como herramienta a utilizar para modelar el sistema.

# Capítulo 1

---

## 1.7.2- Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que da soporte al análisis y diseño orientados a objetos, construcción, pruebas y despliegue del ciclo de desarrollo de un software. Permite la rápida construcción de aplicaciones de calidad y a un menor coste. Brinda la posibilidad de crear todos los tipos de diagramas de clases, código inverso (SlideShare, 2009).

SlideShare (SlideShare, 2009) define que las características generales de Visual Paradigm son:

- Soporte de UML versión 2.1
- Interoperabilidad con modelos UML2 (meta modelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).
- Generación de código: Modelo a código, diagrama a código.
- Generación de bases de datos: Transformación de diagramas de Entidad- Relación en modelos de base de datos.
- Ingeniería inversa de bases de datos: Desde sistemas gestores de bases de datos (SGBD) existentes a diagramas de entidad-relación.
- Generador de informes para generación de documentación.

Esta herramienta se encuentra disponible en múltiples plataformas y en múltiples versiones. Sus características lo hacen una opción viable ante herramientas como el Rational Rose, que es una herramienta muy recomendada y además profesional, pero obliga al usuario a desarrollar en máquinas con el sistema operativo Windows, mientras que el Visual Paradigm está disponible para varios sistemas operativos como Windows, Linux, Unix. Por tales motivos y por la productividad alcanzada mediante la misma, resulta ser Visual Paradigm la herramienta utilizada para el modelado del presente trabajo.

## 1.8- Sistema Gestor de Base de Datos.

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos (Ramos, 2008).

# Capítulo 1

---

Teniendo en cuenta de que el CMS seleccionado utiliza MySQL como gestor de base de datos, a continuación se describe su funcionamiento.

## 1.8.1- MySQL

Según Gallego (Gallego, 2003) MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso (Welling, 2003).

Entre las características que hacen posible su gran aceptación se encuentra la existencia de ininidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Entre las características principales definidas en el Manual de Referencia MySQL 5.0 por Sun Microsystems, Inc. se encuentran: alto rendimiento, aprovecha la potencia de sistemas multiprocesador; velocidad, es mucho más rápido que la gran parte de sus rivales; portabilidad, se ejecuta en la mayoría de los sistemas operativos; gestión de usuarios y contraseñas, mantiene un buen nivel de seguridad en los datos; facilidad de configuración y aprendizaje; funcionalidad, posee compatibilidad completa con ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad).

## 1.9- Entorno de Desarrollo Integrado para Aplicaciones Web

Un IDE (Integrated Development Environment) es un entorno de programación que se compone un editor de código, un compilador, un depurador y en ocasiones un constructor de interfaz gráfica GUI, que han sido empaquetados como un programa de aplicación.

Los IDEs pueden ser aplicaciones que brindan funcionalidades propias o que sirven de apoyo a aplicaciones existentes. Disponen de un editor de código que además, puede servir para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación.

# Capítulo 1

---

## 1.9.1- Zend Studio

Zend Studio o Zend Development Environment es un entorno de desarrollo integrado para el trabajo con el lenguaje de programación PHP. Se encuentra implementado en Java y disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux (Alvarez, 2008).

Según Alvarez (Alvarez, 2008) las características más relevantes que posee este IDE son: autocompletado de código, posee ayuda de código y lista de parámetros de funciones y métodos de clase; detección de errores de sintaxis en tiempo real; funciones de depuración, posee botones de depuración y traza, marcadores, puntos de paradas, seguimientos de variables y mensajes de error del intérprete de PHP; tiene soporte para control de versiones usando CVS o Subversión.

A pesar de las potencialidades y facilidades que brinda, no se ajusta a las características definidas para el sistema a desarrollar por su condición de software propietario, por lo que queda descartado como herramienta para desarrollar la aplicación.

## 1.9.2- Aptana Studio

Aptana Studio es un entorno de desarrollo integrado para la elaboración de aplicaciones web dinámicas que empleen PHP, Ruby, Ruby on Rails y Python (Aptana Inc, 2009).

Es multiplataforma y de código abierto, enfocado al desarrollo de interfaces de usuario Web dinámicas ideal para maquetar páginas Web. Soporta las librerías más populares: Prototype, ExtJS, MochiKit, Yahoo UI, Aflax, JQuery, HTML, CSS y JavaScript (Valdés, 2007).

Posee un asistente de código que ayuda al programador en la escritura de los diferentes lenguajes, CSS y Javascript entre ellos. Aptana Studio contiene también información de soporte para los principales navegadores web: IE, Firefox, Opera, Netscape y Safari (Aptana Inc, 2009).

Según la Corporación de Aptana (Aptana Inc, 2009) otras características interesantes de Aptana son las siguientes:

- Explorador de código en forma de árbol.

# Capítulo 1

---

- Librerías populares AJAX/Javascript.
- Extensión de funcionalidad mediante macros y acciones.
- Visor de errores y advertencias.
- Servidor local para probar el código.

Su condición de software libre y multiplataforma junto a sus facilidades de uso, lo hacen útil para el desarrollo del presente trabajo según las funcionalidades definidas para el mismo.

## Conclusiones Parciales

Luego del análisis de cada uno de los elementos contemplados en el desarrollo del capítulo nos permite llegar a las siguientes conclusiones parciales:

- La aplicación deseada debe asistir el proceso de evaluación de desempeño de los profesores de una manera más justa, contribuyendo con el mejoramiento intelectual de los profesores.
- Las principales limitaciones que poseen las aplicaciones informáticas que apoyan el sistema de evaluación de desempeño es que no brindan la posibilidad de que los evaluadores conozcan los indicadores que deben tener en cuenta para evaluar cada aspecto, además no posibilita que los profesores conozcan su historial de evaluaciones durante un curso.
- De manera general, como parte de la investigación fueron descritas las herramientas, metodologías y tecnologías a utilizar, las cuales permitieron realizar la aplicación informática de forma eficiente y rápida: Scrum-XP fue la metodología de desarrollo seleccionada, como marco de trabajo se escogió Joomla, la cual permitió definir el lenguaje de programación utilizado (php5) y el gestor de base de datos (MySQL), y por último el entorno de desarrollo a utilizar es Aptana Studio.

# CAPÍTULO 2: LA APLICACIÓN INFORMÁTICA “EVALDES”.

## Introducción

En este capítulo se describen los requerimientos funcionales y no funcionales, se definen las personas que están relacionadas con el sistema, además para que se entienda correctamente lo que debe hacer la aplicación se describen las características deseadas del sistema a través de las historias de los usuarios. Se construye el diseño de acuerdo las funcionalidades que brinda el CMS Joomla, y son especificadas las historias de usuarios agregadas para el correcto funcionamiento del sistema deseado, teniéndose como resultado un diseño que cumpla con las necesidades de los clientes. Por otra parte se desarrollan las tareas de ingeniería para lograr que se cumplan todas las historias de usuarios definidas para su correcto funcionamiento.

## 2.1- Requisitos del sistema.

### 2.1.1- Requisitos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Identifican lo que el sistema debe hacer y se mantienen invariables sin importar con que propiedades o cualidades estén relacionadas (JACOBSON, y otros, 2000).

#### **RF1:** Autenticar Usuario.

El sistema permitirá que los usuarios que se encuentran registrados en la base de datos de la aplicación puedan autenticarse. Para ello el usuario deberá introducir su usuario y contraseña.

##### **1.1 :** Cambiar contraseña.

El usuario podrá cambiar su contraseña, para la cual debe proporcionar el usuario, la contraseña que quiere tener y luego debe confirmarla.

##### **1.2 :** Recuperar contraseña.

El usuario podrá recuperar su contraseña en caso de olvido, para ello debe suministrar su correo electrónico.



## Capítulo 2

---

### 1.3 : Recuperar usuario.

El usuario podrá recuperar su usuario en caso de olvido, para ello debe suministrar su correo electrónico.

### **RF2:** Gestionar Usuarios.

El administrador podrá gestionar los usuarios que tendrán permisos para interactuar con la aplicación.

#### 2.1: Insertar usuario.

El administrador podrá insertar los usuarios al sistema, para ello debe introducir el nombre completo, el nombre de usuario que tendrá, el correo electrónico, la contraseña y luego verificarla, y por último el rol que tendrá el usuario.

#### 2.2: Modificar datos de usuario.

El administrador podrá modificar los datos de los usuarios del sistema. (Los datos se describen en el requisito 2.1)

#### 2.3: Eliminar usuario.

El administrador podrá eliminar el usuario que desee.

### **RF3:** Gestionar Desempeño.

Todos los requisitos que se describen a continuación tienen como escenario el menú ***“Gestionar Desempeño”***.

#### 3.1: Registrar Profesor.

El administrador podrá registrar los profesores a la aplicación, accediendo a la opción ***“Registrar Profesor”*** y proporcionando la información siguiente: nombres, apellidos, usuario categoría docente y departamento.

#### 3.2: Eliminar Profesor.

El administrador podrá eliminar el profesor que desee accediendo a la opción ***“Eliminar Profesor”***.

#### 3.3: Insertar Área.

El administrador podrá insertar las áreas a la aplicación mediante la opción ***“Insertar Área”***, donde debe proporcionar los siguientes datos: nombre del área, área superior, si es un área administrativa, el jefe del área.

## Capítulo 2

---

### 3.4: Eliminar Área.

El administrador podrá eliminar el área que desee solamente accediendo a la opción **“Eliminar Área”** y seleccionando el área que desee eliminar.

### 3.5: Evaluar Actividad.

El evaluador podrá evaluar a los profesores que les son asignados por el área, para ello debe acceder a la opción **“Evaluar Actividad”** y proporcionar los siguientes datos: aspecto a evaluar, profesor a evaluar, si cumplió con la actividad, fecha de realización, factor de calidad y por último un comentario.

### 3.6: Actualizar Evaluación de Desempeño.

El jefe de área podrá actualizar la evaluación proporcionada por la aplicación, para ello debe acceder a la opción **“Actualizar Evaluación”** y luego llenar los siguientes datos: nombre del área, período evaluativo, profesor a evaluar, evaluación final y un comentario.

### 3.7: Agregar Profesores.

El jefe de área podrá agregar al área los profesores que pertenecen a la misma, para lo cual debe seleccionar la opción **“Agregar Profesores”**.

### 3.8: Agregar Evaluadores.

El jefe de área podrá agregar al área los profesores que pertenecen a la misma, para lo cual debe seleccionar la opción **“Agregar Evaluadores”**.

### 3.9: Asignar Aspectos a Profesores.

El jefe de área podrá asignar los aspectos a medir a cada uno de los profesores que pertenecen su área, para ello debe acceder a la opción **“Asignar Aspectos a Profesores”**.

### 3.10: Asignar Aspectos a Evaluadores.

El jefe de área podrá asignar los aspectos a medir a cada uno de los evaluadores que pertenecen su área, para ello debe acceder a la opción **“Asignar Aspectos a Evaluadores”**.

### RF4: Gestionar Aspectos e Indicadores.

El administrador podrá gestionar los aspectos e indicadores, para ello debe acceder a la opción **“Aspectos e Indicadores”**.

## Capítulo 2

---

### 4.1: Insertar Indicadores.

El administrador podrá insertar indicadores mediante la opción **“Insertar Indicador”**, para ello debe proporcionar los siguientes datos: nombre del indicador y descripción.

### 4.2: Insertar Aspectos.

El administrador podrá insertar aspectos mediante la opción **“Insertar Aspecto”**, para ello debe proporcionar los siguientes datos: nombre del aspecto y grado de relevancia.

### 4.3: Mostrar relación Indicadores-Aspectos.

El administrador podrá relacionar los indicadores con los aspectos, para lo cual debe acceder a la opción **“Relación Aspecto – Indicador”**.

### 4.4: Eliminar Indicadores.

El administrador podrá eliminar el indicador que desee accediendo a la opción **“Eliminar Indicador”**.

### 4.5: Eliminar Aspectos.

El administrador podrá eliminar el aspecto que desee accediendo a la opción **“Eliminar Aspecto”**.

### RF5: Gestionar Período Evaluativo.

El administrador podrá gestionar el período evaluativo, para ello debe acceder a la opción **“Período evaluativo”**.

#### 5.1: Insertar Período Evaluativo.

El administrador podrá insertar un período evaluativo accediendo a la opción **“Insertar Período”** y proporcionando los siguientes datos: fecha de inicio y fecha fin.

#### 5.2: Modificar Período Evaluativo.

El administrador podrá modificar un período evaluativo accediendo a la opción **“Modificar Período”** y proporcionando los siguientes datos: período a modificar, fecha de inicio y fecha fin.

### RF6: Mostrar Reportes.

Los usuarios podrán conocer las evaluaciones de desempeño de todos los profesores en un período determinado, para ello debe seleccionar la opción **“Reportes”** y seleccionar el período evaluativo.

## Capítulo 2

---

### **RF7: Mostrar Historial de Evaluaciones.**

Los usuarios podrán conocer el historial de evaluaciones de desempeño de todos los profesores, para ello debe seleccionar la opción “**Mostrar Historial de Evaluaciones**” y seleccionar el profesor que desee.

### **2.1.2- Requisitos no funcionales.**

Los requerimientos no funcionales son capacidades o cualidades que el producto debe tener, o sea, características que hagan al producto atractivo, rápido, usable o confiable. Están estrechamente vinculados a los requisitos funcionales, puesto que una vez que está definido lo que el sistema debe hacer, es necesario especificar cómo ha de hacerlo. Pueden llegar a marcar la diferencia entre un producto bien aceptado por los clientes y usuarios o uno de poca o ninguna calidad y aceptación (JACOBSON, y otros, 2000).

#### **RNF 1 Software.**

- Debe disponerse del sistema operativo Windows 98 o superior, o con el sistema operativo Linux.

#### **RNF 2 Apariencia o Interfaz Externa.**

- Diseño orientado a llamar la atención del usuario y con una navegación sencilla.
- El sistema debe tener una interfaz sencilla y fácil de usar.

#### **RNF 3 Seguridad.**

- Identificar al usuario antes de que pueda realizar cualquier acción sobre el contenido del sistema.
- Garantizar que la información sea editada únicamente por quien tiene derecho a editarla.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de acceso del usuario activo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones).

#### **RNF 4 Disponibilidad.**

- El sistema deberá estar disponible las 24 horas del día.

## Capítulo 2

---

### **RNF 5 Confiabilidad.**

- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.
- Se deben establecer distintos roles que limiten el nivel de accesibilidad de los usuarios.

### **RNF 6 Portabilidad.**

El sistema será portable y multiplataforma, cumpliendo con los siguientes requisitos:

- Hará uso de tecnologías rápidas, livianas y multiplataforma.
- No requerirá de acciones adicionales para la migración de una plataforma a otra.
- No necesitará de la instalación de herramientas adicionales en el cliente, a parte de un navegador web.

### **RNF 7 Hardware.**

El sistema debe ser capaz de correr con prestaciones mínimas para el servidor:

- Pentium II de 256 MB de RAM, preferiblemente 512 MB de RAM a 1GHz para el servidor.

### **RNF 8 Legales.**

- El sistema será desarrollado bajo el uso de herramientas no propietarias.

### **RNF 9 Rendimiento.**

- Tiempos de respuesta no mayor de 2 segundos, al igual que la velocidad de procesamiento de la información.
- Estará implementado con tecnología Web para facilitar el uso a través de la red.

## **2.2- Propuesta del sistema.**

El sistema que se desea desarrollar tiene como objetivo servir como herramienta para evaluar el desempeño de los profesores de las diferentes áreas. El sistema estará diseñado completamente con el

## Capítulo 2

---

uso de las tecnologías, el sistema Web se implementará en PHP para facilitar la interacción con el sistema y la mejor obtención de los reportes generado.

Con el fin de cumplir con los objetivos del trabajo, el sistema debe poseer cuatro niveles de acceso: el usuario registrado que son los profesores que deben ser evaluados, los cuales van a tener acceso a la información pública, que no es más que los Reportes donde el profesor podrá ver las evaluaciones de todos los profesores en el período evaluativo; el evaluador, el cual tendrá los mismos permisos del usuario registrado además de poder gestionar el desempeño de los profesores, así como los aspectos e indicadores, además de determinar el período evaluativo; el Jefe de área, el cual va a tener los mismos permisos del evaluador pero además es el que determina cual será la evaluación de desempeño final de cada profesor; por último, el administrador del sistema el cual tendrá todos los permisos del sistema, será el encargado de gestionar los usuarios y de velar porque el sistema este funcionando adecuadamente.

### 2.2.1- Personas relacionadas con el sistema.

*Tabla 2.1 Personas relacionadas con el sistema*

Personas	Descripción
Usuario	Es la generalización entre evaluadores, jefes de áreas y administrador, ya que todos deben autenticarse para acceder al sistema para realizar acciones específicas dependiendo del rol determinado.
Evaluador.	Son los encargados de gestionar el desempeño de los profesores, de gestionar los aspectos e indicadores a evaluar y de gestionar el período evaluativo.
Jefe de área.	Son los encargados de dar la evaluación de desempeño a todos los profesores de su área, además gestionar el historial de evaluaciones.
Administrador.	Es el encargado de administrar el sistema, permitiendo que este funcione adecuadamente las 24 horas del día.

## Capítulo 2

### 2.2.2- Prototipo de Interfaz.

Figura2 Interfaz Principal de la aplicación “Evaldes”.



### 2.3- Historias de usuarios.

Las Historias de Usuarios (HU, en lo adelante) según Beck (Beck, y otros, 2000) conducen al proceso de creación de los test de aceptación, los cuales servirán para verificar que estas historias se han implementado correctamente. Otra de sus características es que solamente proporcionan los detalles sobre la estimación del riesgo y cuánto tiempo conllevará su implementación. Su nivel de detalle debe ser el mínimo posible, de manera que permita hacerse una ligera idea de cuánto costará implementar el sistema.

## Capítulo 2

---

**Tabla2.2 HU Autenticar Usuario.**

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre de Historia de Usuario:</b> Autenticar Usuario.
<b>Modificación de Historia de Usuario:</b> (-)	
<b>Usuario (s):</b> usuario	<b>Iteración asignada:</b> 1
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b>
<b>Descripción:</b> El usuario para realizar cualquier operación en el sistema debe autenticarse, además podrá realizar las siguientes operaciones: <ul style="list-style-type: none"><li>• Recuperar usuario.</li><li>• Cambiar contraseña.</li><li>• Recuperar contraseña.</li></ul>	
<b>Observaciones:</b> El usuario tiene que estar previamente registrado en la base de datos.	

**Tabla2.3 HU Gestionar Usuario.**

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre de Historia de Usuario:</b> Gestionar Usuario
<b>Modificación de Historia de Usuario:</b> (-)	
<b>Usuario (s):</b> administrador	<b>Iteración asignada:</b> 1
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b>
<b>Descripción:</b> El administrador podrá realizar las siguientes operaciones: <ul style="list-style-type: none"><li>• Insertar usuario.</li><li>• Modificar datos de usuario.</li><li>• Eliminar usuario.</li></ul>	



## Capítulo 2

**Observaciones:** -

**Tabla2.4 HU Gestionar Desempeño.**

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre de Historia de Usuario:</b> Gestionar Desempeño
<b>Modificación de Historia de Usuario:</b> (-)	
<b>Usuario (s):</b> evaluador, jefe de área, administrador	<b>Iteración asignada:</b> 1
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 3
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b>
<b>Descripción:</b> El evaluador podrá realizar las siguientes operaciones: <ul style="list-style-type: none"><li>• Evaluar actividad del profesor.</li></ul> El jefe de área además de las operaciones descritas anteriormente, podrán realizar: <ul style="list-style-type: none"><li>• Agregar profesores.</li><li>• Agregar evaluadores.</li><li>• Agregar aspectos a evaluadores.</li><li>• Agregar aspectos a profesores.</li><li>• Actualizar evaluación de desempeño.</li></ul> El administrador del sistema además de las operaciones descritas para el evaluador y el jefe de área podrá realizar: <ul style="list-style-type: none"><li>• Insertar profesor.</li><li>• Eliminar profesor.</li><li>• Insertar área.</li><li>• Eliminar área.</li></ul>	
<b>Observaciones:</b> El evaluador, el jefe de área y el administrador deben tener los permisos necesarios para realizar las operaciones descritas.	

Las otras HU se encuentran la carpeta Artefactos.

## Capítulo 2

---

### 2.4- Planificación de las iteraciones (sprint).

En Scrum todo el desarrollo ocurre en iteraciones de duración fija. Estas iteraciones de duración fija se denominan Sprint. La duración del Sprint es de entre 2 y 4 semanas, o 5 semanas en algunos casos se establece al comienzo del proyecto, y se mantiene durante toda la vida del mismo. Lo anteriormente definido es importante para hacer uso de las métricas que provee Scrum.

Durante cada Sprint el equipo desarrolla una porción del sistema. Lo destacable del Sprint es que el desarrollo incremental que potencia es con calidad productiva, cumpliendo con algún requerimiento del usuario (por más pequeño que sea). De esta manera, al finalizar cada Sprint se tiene una porción del producto con calidad, que el usuario puede decidir poner en producción.

#### Definición de los Sprint del proyecto:

**Sprint 1:** En este sprint es donde se va a obtener la primera versión del producto. Puede ser el sprint de mayor duración, ya que, al principio cuesta más trabajo obtener un ejecutable. Las HU de alta prioridad serán implementadas ofreciéndole al sistema las primeras funcionalidades.

**Sprint 2:** En este sprint serán analizadas otras HU de alta y media prioridad respectivamente, que no fueron analizadas en la primera iteración. Con la terminación de la misma se habrán implementado todas las HU de alta prioridad y gran parte de las necesidades del cliente.

**Sprint 3:** En este sprint serán implementadas las restantes HU y se tendrá finalmente todas las funcionalidades implementadas. Comenzaría entonces la fase de pruebas.

*Tabla 2.9 Plan de duración de los sprint.*

Sprint.	HU a integrar al sistema.	Duración total de los Sprints.
Sprint 1	1. Autenticar Usuario. 2. Gestionar Usuario. 3. Gestionar Desempeño.	5 semanas
Sprint 2	1. Gestionar Período Evaluativo. 2. Gestionar Aspecto e	4 semanas

## Capítulo 2

---

	Indicador.	
Sprint 3	1. Mostrar Reportes. 2. Mostrar Historial de Evaluaciones.	2 semanas

### 2.5- Diseño

Para una mejor comprensión de este trabajo se hace una breve explicación del funcionamiento de los componentes del CMS Joomla, así como de los componentes implementados para obtener la solución final.

#### 2.5.1- Componentes de Joomla.

La instalación estándar de Joomla según White y colegas (White, y otros, 2006) contiene los siguientes componentes:

Banners: permite mostrar banners (anuncios) en su sitio. También permite gestionar algunas características avanzadas como impresiones de los banners y código personalizado.

Contactos: posibilita el gestionar una lista de contactos para su sitio.

Correo Masivo: permite enviar un mensaje de correo electrónico a uno o más grupos de usuarios.

Encuestas: permite gestionar y crear diferentes encuestas para sus visitantes. Las cuestiones planteadas pueden ser útiles si se utilizan con fines informativos.

Enlaces Web: permite contar el número de clics para cada enlace y los Usuarios Registrados pueden enviar sus enlaces web desde el Front-end.

Noticias Externas: posibilita la capacidad para gestionar Noticias Externas, situándolas en diferentes categorías, y seleccionar los enlaces para recibir Noticias Externas desde diferentes proveedores.

Sindicación: La Sindicación del sitio permite que los Artículos de Contenido asignados al Componente Página de Inicio del sitio estén disponibles como archivos XML (estándar para el intercambio de

## Capítulo 2

---

información estructurada), para su distribución pública mediante los servicios de suscripción a sitios de noticias.

### 2.5.2- Componentes implementados.

Para darle las funcionalidades requeridas al sistema se desarrollaron los componentes que se describen a continuación:

Gestionar evaldes: permite que se realicen las funcionalidades de la HU Gestionar Desempeño.

Aspecto e Indicador: permite que se realicen las funcionalidades de la HU Gestionar Aspectos e Indicadores.

Periodo evaluativo: permite que se realicen las funcionalidades de la HU Gestionar Período Evaluativo.

Reportes: permite que se realicen las funcionalidades de la HU Mostrar Reportes.

Gestionar Historial: permite que se realicen las funcionalidades de la HU Mostrar Historial de Evaluaciones.

### 2.6- Patrones de diseño utilizados.

Una arquitectura orientada a objetos bien estructurada está llena de patrones. La calidad de un sistema orientado a objetos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles (Booch, 2008).

Según el equipo de trabajo de Joomla (Joomlaos, 2010) los patrones de diseño se encuentran comunmente en el código. Joomla utiliza comunmente los siguientes patrones:

**Iterator**: este modelo describe cómo realizar una tarea varias veces con un bucle. Joomla! utiliza numerosos patrones de diseño, muchos de los cuales son más complejos que el patrón iterador.

**Fabrica abstracta**: es un patrón que se utiliza para construir y devolver objetos. El patrón de la fábrica se utiliza en los casos en donde diferentes clases, por lo general derivan de una clase abstracta, son

## Capítulo 2

---

instanciadas dependiendo de los parámetros. Joomla! proporciona la clase estática JFactory, que implementa el patrón de la fábrica. Esta clase permite acceder de forma sencilla a instancias de los objetos globales.

Este ejemplo muestra cómo podemos acceder a algunos de los objetos globales utilizando JFactory.

```
$db = &JFactory::getDBO();  
$user = &JFactory::getUser();  
$document = &JFactory::getDocument();
```

**Singleton:** se utiliza para permitir la creación de sólo un único objeto de una clase específica. Esto se consigue haciendo el constructor privado o protegido y usando un método estático para crear instancias de la clase.

Muchas de las clases de Joomla utilizan un modelo pseudo-singleton que nos permite crear instancias de objetos y de acceso. Para lograr esto, Joomla a menudo utiliza un método estático llamado getInstance(); en algunos casos JFactory actúa como una interfaz a través de este método.

### 2.6.1- Modelo-Vista-Controlador.

El patrón de diseño Modelo-Vista-Controlador (MVC, en lo adelante) según el criterio de Reynoso (Reynoso, 2005) es materializada mediante la implementación del patrón arquitectónico que lleva el mismo nombre, este patrón separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

En el caso específico de Joomla el uso del patrón MVC no requiere de un fichero de configuración, lo cual puede ser una ventaja o una desventaja según como se mire. Joomla proporciona el mapeo las acciones del controlador a través del nombre que se le tiene que dar a cada directorio y fichero del componente, un error en los nombres y Joomla no podrá encontrar los ficheros requeridos implicando el mal funcionamiento del componente (LeBlanc, 2007).

Específicamente el CMS Joomla se implementa utilizando este patrón a través de 3 clases principales: JModel, JView, JController.

## Capítulo 2

---

**JController** es una clase abstracta que proporciona la funcionalidad básica para las clases de su propio controlador que crear para sus componentes. Tiene un método para cada tarea con la que el controlador tenga que tratar: mostrar un elemento, crear un elemento, borrar un elemento. En estos métodos el controlador analiza la entrada del usuario y lleva a cabo los pasos necesarios para generar la respuesta (Dique, 2010).

**JModel** es una clase abstracta que proporciona la funcionalidad básica de los objetos modelo concreto en relación con el patrón de Joomla MVC. Los modelos que amplían la clase JModel se definen a menudo en los componentes de Joomla y definir la lógica de negocio del componente específico. Poseen distintos métodos para realizar distintas acciones sobre los datos (Dique, 2010).

**JView** es una clase abstracta que proporciona la funcionalidad básica para el patrón de diseño MVC de Joomla. Se necesita para escribir sus propias clases de vista concreta, a hacer pleno uso de la funcionalidad (Dique, 2010).

Según Enso (Enso, 2010) las ventajas del patrón MVC son:

- Clara separación entre interfaz, lógica de negocio y de presentación, que además provoca parte de las ventajas siguientes.
- Sencillez para crear distintas representaciones de los mismos datos.
- Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas (TDD).
- Reutilización de los componentes.
- Simplicidad en el mantenimiento de los sistemas.
- Facilidad para desarrollar prototipos rápidos.
- Los desarrollos suelen ser más escalables

## Capítulo 2

---

### 2.7- Desarrollo de los componentes del sistema.

En la fase de codificación o implementación de la metodología Scrum-XP se genera toda la documentación para los términos de difícil entendimiento para los clientes, así como las tareas de ingeniería a realizar durante la implementación, es generado el código fuente y la documentación relacionada con las pruebas.

#### 2.7.1- Tareas de Ingeniería.

Teniendo en cuenta la planificación que se hizo anteriormente de los Sprints, se decidió realizar una iteración de desarrollo para cada sprint obteniéndose como finalidad un producto con las restricciones características deseadas para la utilización del sistema. Debido a esto, a continuación se detallan cada una de las iteraciones.

#### Tareas asignadas al Sprint 1.

*Tabla2.10 Desarrollar el formulario para la inserción de profesores.*

Tarea de Ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de Historia de Usuario:</b> HU3
<b>Nombre de tarea:</b> Desarrollar el formulario para la inserción de profesores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha Inicio:</b> 13 de marzo 2010	<b>Fecha Fin:</b> 14 de marzo 2010
<b>Programador responsable:</b> Pavel Pérez Sierra.	
<b>Descripción:</b> Se genera el código que permite crear el formulario para la inserción de profesores al sistema.	

## Capítulo 2

---

**Tabla2.11 Desarrollar el formulario para la inserción de áreas.**

Tarea de Ingeniería	
<b>Número de Tarea:</b> 2	<b>Número de Historia de Usuario:</b> HU3
<b>Nombre de tarea:</b> Desarrollar el formulario para la inserción de áreas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha Inicio:</b> 14 de marzo 2010	<b>Fecha Fin:</b> 15 de marzo 2010
<b>Programador responsable:</b> Pavel Pérez Sierra.	
<b>Descripción:</b> Se genera el código que permite crear el formulario para la inserción de áreas al sistema.	

**Tabla2.18 Desarrollar el formulario para evaluar actividad.**

Tarea de Ingeniería	
<b>Número de Tarea:</b> 9	<b>Número de Historia de Usuario:</b> HU3
<b>Nombre de tarea:</b> Desarrollar el formulario para evaluar actividad.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha Inicio:</b> 21 de marzo 2010	<b>Fecha Fin:</b> 22 de marzo 2010
<b>Programador responsable:</b> Pavel Pérez Sierra.	
<b>Descripción:</b> Se genera el código que permite crear el formulario para evaluar las actividades de los profesores de las áreas del sistema.	



## Capítulo 2

---

**Tabla2.19 Desarrollar el formulario para evaluar el desempeño de los profesores.**

Tarea de Ingeniería	
<b>Número de Tarea:</b> 10	<b>Número de Historia de Usuario:</b> HU3
<b>Nombre de tarea:</b> Desarrollar el formulario para evaluar el desempeño de los profesores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha Inicio:</b> 22 de marzo 2010	<b>Fecha Fin:</b> 23 de abril 2010
<b>Programador responsable:</b> Pavel Pérez Sierra.	
<b>Descripción:</b> Se genera el código que permite crear el formulario para evaluar el desempeño de los profesores de las diferentes áreas del sistema.	

**Tabla2.20 Implementar la funcionalidad para la inserción de profesores**

Tarea de Ingeniería	
<b>Número de Tarea:</b> 11	<b>Número de Historia de Usuario:</b> HU3
<b>Nombre de tarea:</b> Implementar la funcionalidad para la inserción de profesores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha Inicio:</b> 23 de marzo 2010	<b>Fecha Fin:</b> 24 de marzo 2010
<b>Programador responsable:</b> Pavel Pérez Sierra.	
<b>Descripción:</b> Se genera el código que permite implementar la funcionalidad para la inserción de profesores al sistema.	

## Capítulo 2

---

**Tabla2.21 Implementar la funcionalidad para la inserción de áreas.**

Tarea de Ingeniería	
<b>Número de Tarea:</b> 12	<b>Número de Historia de Usuario:</b> HU3
<b>Nombre de tarea:</b> Implementar la funcionalidad para la inserción de áreas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha Inicio:</b> 24 de marzo 2010	<b>Fecha Fin:</b> 25 de marzo 2010
<b>Programador responsable:</b> Pavel Pérez Sierra.	
<b>Descripción:</b> Se genera el código que permite implementar la funcionalidad para la inserción de áreas al sistema.	

**Tabla2.28 Implementar la funcionalidad para evaluar actividad.**

Tarea de Ingeniería	
<b>Número de Tarea:</b> 19	<b>Número de Historia de Usuario:</b> HU3
<b>Nombre de tarea:</b> Implementar la funcionalidad para evaluar actividad.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha Inicio:</b> 31 de marzo 2010	<b>Fecha Fin:</b> 1 de abril 2010
<b>Programador responsable:</b> Pavel Pérez Sierra.	
<b>Descripción:</b> Se genera el código que permite implementar la funcionalidad para evaluar las actividades de los profesores de las áreas del sistema.	

## Capítulo 2

---

**Tabla2.29 Implementar la funcionalidad para evaluar el desempeño de los profesores.**

Tarea de Ingeniería	
<b>Número de Tarea:</b> 20	<b>Número de Historia de Usuario:</b> HU3
<b>Nombre de tarea:</b> Implementar la funcionalidad para evaluar el desempeño de los profesores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha Inicio:</b> 1 de abril 2010	<b>Fecha Fin:</b> 2 de abril 2010
<b>Programador responsable:</b> Pavel Pérez Sierra.	
<b>Descripción:</b> Se genera el código que permite implementar la funcionalidad para evaluar el desempeño de los profesores de las diferentes áreas del sistema.	

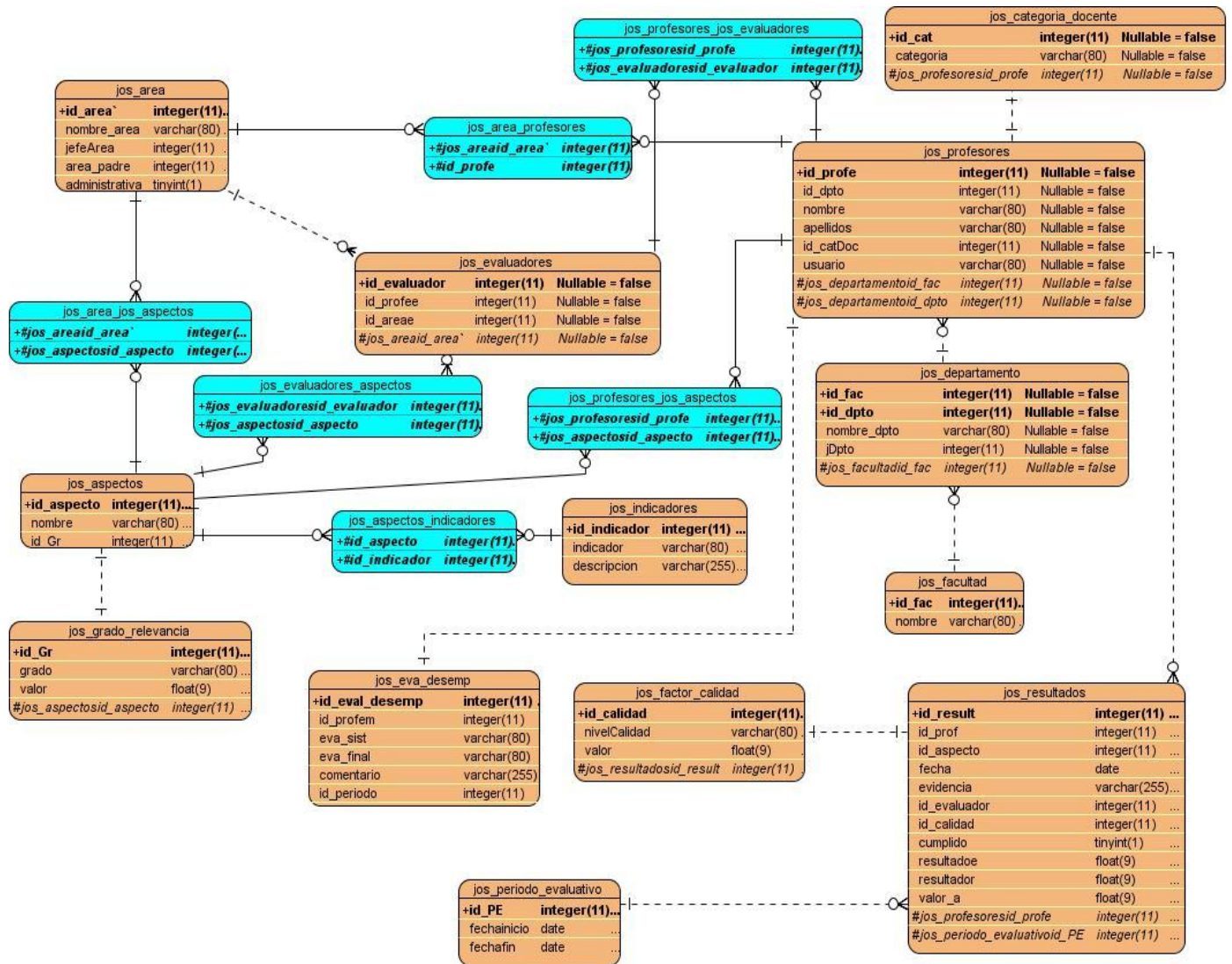
Nota: un día se tomo como 0.14 unidades aproximadamente.

Las otras tareas de la iteración 1 y las tareas de las iteraciones 2 y 3 se encuentran en la carpeta Artefactos.

# Capítulo 2

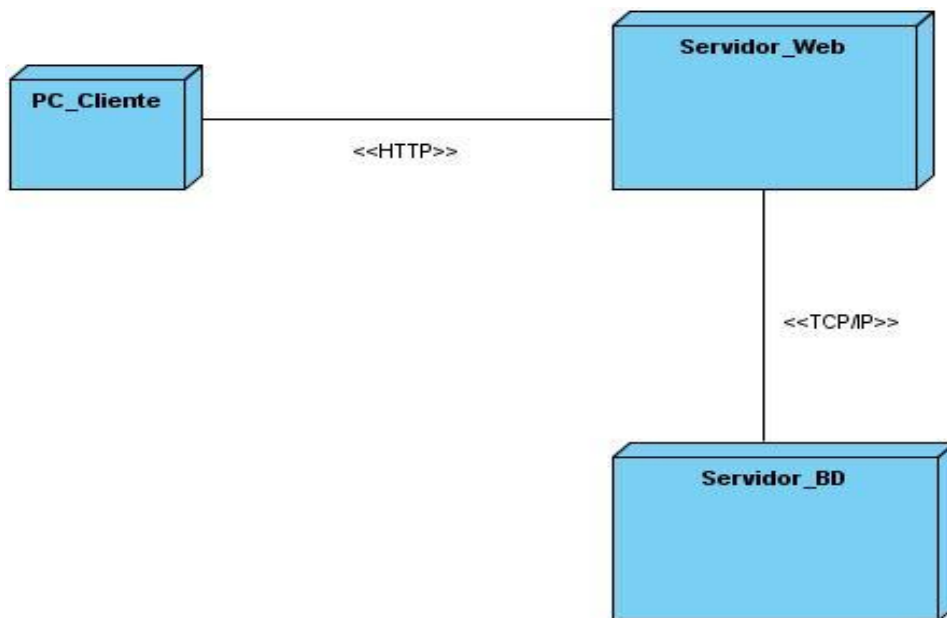
## 2.7.2- Diagrama Modelo de Datos.

Figura 3 Modelo de Datos de la aplicación.



### 2.7.3- Diagrama de Despliegue.

Figura 4 Diagrama de Despliegue.



### 2.8- Beneficios de la implementación del sistema.

Antes de la implementación del Sistema de Pago Adicional Centrado en la Evaluación de Desempeño, se hacía muy engorrosa la gestión de la evaluación de desempeño de cada profesor de las diferentes áreas de la facultad, debido a que no existían factores que midieran de forma neutral el desempeño de los profesores. Esta aplicación informática posibilita una forma más justa de otorgar la evaluación de desempeño y a su vez le brinda a los jefes de área la posibilidad de controlar de las actividades, proponiendo una evaluación que puede ser aceptada o rechazada por el jefe de área.

### 2.9- Validación del sistema.

Uno de los pilares fundamentales de XP es el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados así como el tiempo entre la introducción de este en el sistema y su detección. Todo esto

## Capítulo 2

---

contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones.

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente (Crispin, y otros, 2002).

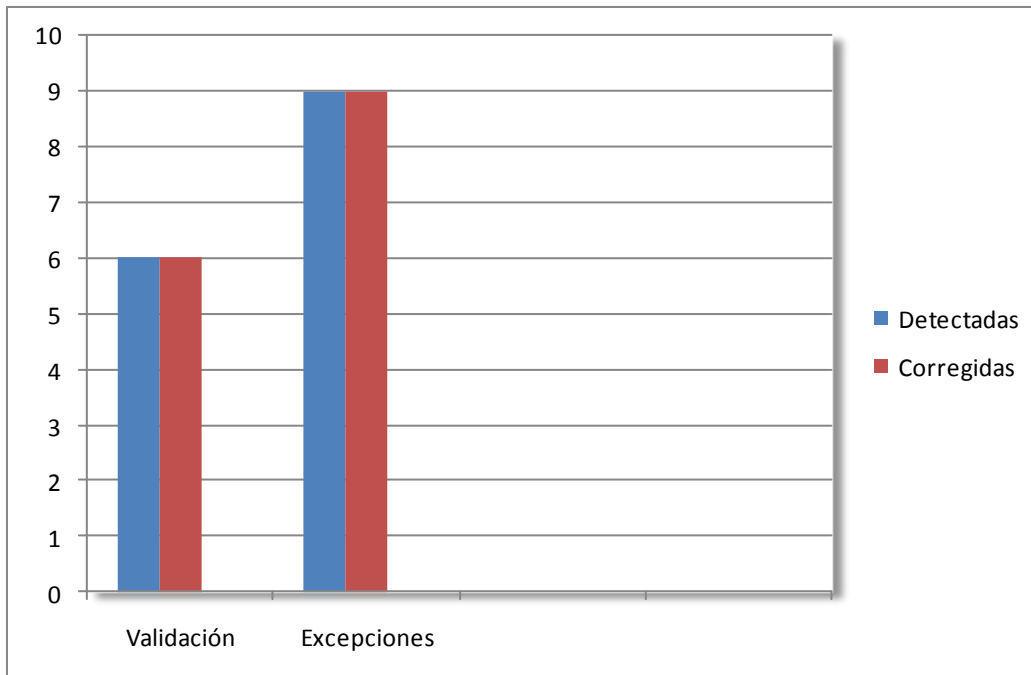
### **2.9.1- Pruebas unitarias.**

Dentro de las pruebas se encuentran dos fundamentales: Prueba de Caja Negra y de Caja Blanca. La prueba de la caja blanca del software comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, a través de los casos de prueba se demuestra que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene (Reynoso, 2004).

Luego de realizar la primera iteración de pruebas de caja negra a la aplicación se obtuvieron errores de validación y excepciones como se muestra en la **Figura 5** , siendo liberado el software en la segunda iteración.

## Capítulo 2

**Figura 5 : Cantidad de no conformidades encontradas en la primera iteración.**



**Fuente:** *Elaboración propia.*

### 2.9.2- Pruebas de aceptación.

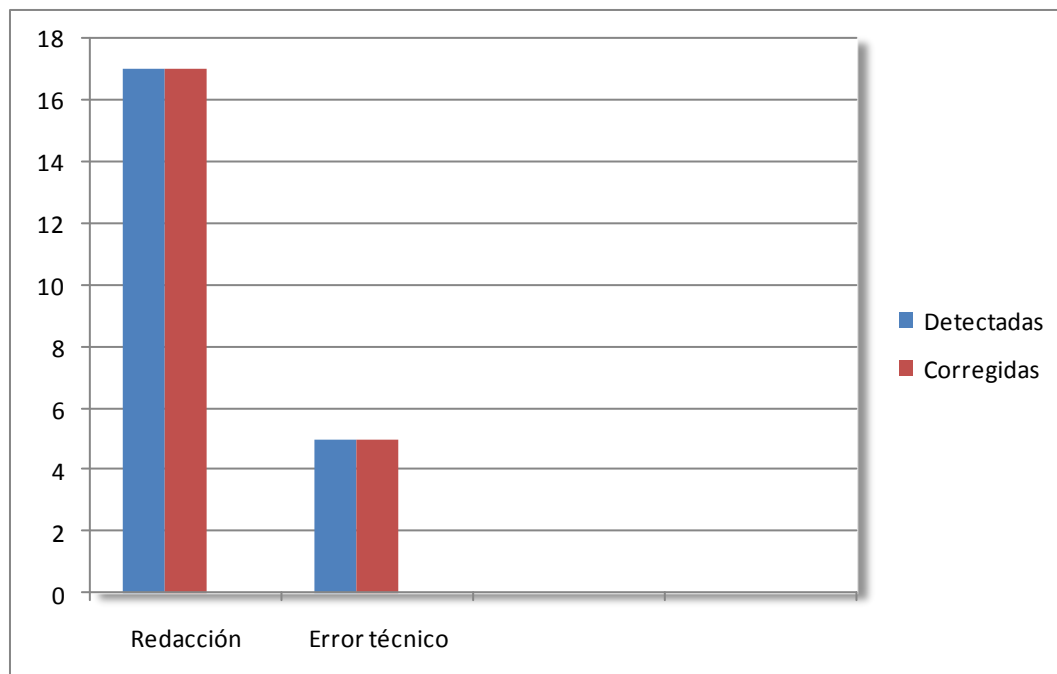
El objetivo de estas pruebas es verificar los requisitos, por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación. Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones a cerca de las mismas (Reynoso, 2004).

En el **Anexo B** pueden analizarse los casos de prueba de aceptación que se utilizaron para comprobar que las funcionalidades daban cumplimiento a los requisitos funcionales descritos en la investigación. Al

## Capítulo 2

concluir la primera iteración de las pruebas de aceptación se obtuvieron solamente no conformidades de redacción y técnicas, las cuales no son consideradas como críticas.

**Figura 5 : Cantidad de no conformidades encontradas en la primera iteración.**



**Fuente:** *Elaboración propia.*

### 2.9.2- Evaluación por criterio de expertos.

Además de las pruebas realizadas para verificar el correcto funcionamiento de la aplicación, se realizó una valoración por criterios de expertos mediante una encuesta a un total de 7 personas que se desempeñan como evaluadores o jefes de área, para la cual se aplicó una encuesta (Ver Instrumento Aplicado en el **Anexo C**).

El coeficiente de competencia de los expertos encuestados obtenido fue de un 28.57 % medio y de un 72.43 % alto y el coeficiente general fue alto con un valor de 0.82, un conocimiento promedio de 7.86, un coeficiente de conocimiento de 0.79 y un coeficiente de aceptación de 0.81.

La siguiente tabla (**Tabla 2.66**) muestra el criterio final obtenido en cada uno de los aspectos que fueron valorados por los encuestados (Ver las **Tabla 2.67 – Tabla 2.70**). En el 85.71% de los aspectos en



## Capítulo 2

---

cuestión se obtuvo un criterio de muy adecuado y el criterio de bastante adecuado representa 14.29%, corroborándose de esta manera el cumplimiento de la hipótesis planteada en la investigación.

**Tabla 2.66: Criterio obtenido para cada uno de los aspectos**

No	Aspectos	Criterio
1	Calidad de la investigación.	Muy adecuado
2	Novedad Científica.	Muy adecuado
3	Aporte Científico.	Bastante adecuado
4	Necesidad del empleo de la nueva aplicación.	Muy adecuado
5	Satisfacción de las necesidades para gestionar el desempeño de los profesores y trabajadores.	Muy adecuado
6	Contribución al proceso de gestión de la evaluación de desempeño de los profesores y trabajadores de la facultad.	Muy adecuado
7	Facilidades de utilización como medio de apoyo a los evaluadores y jefes de área.	Muy adecuado
8	Posibilidad de uso en las demás facultades de la universidad.	Muy adecuado
9	Posibilidades de extensión de la aplicación mediante la inclusión de otras funcionalidades.	Muy adecuado
10	Posibilidades de utilización de la aplicación informática.	Muy adecuado

*Fuente: Elaboración propia*

Los datos que se muestran a continuación contribuyen a corroborar la validación de la hipótesis:

- i. El 43% considera que la calidad de la investigación es excelente, el 29% que es buena y el 28% que la calidad es aceptable.
- ii. El 29% considera de que la novedad científica de la aplicación es excelente el 43% que es buena y el 28% que es aceptable.
- iii. El 15% considera de que el aporte científico es excelente, el 57% que es bueno y el 28% que es aceptable.
- iv. El 29% considera de que es muy necesario el uso de la nueva aplicación, el 57% considera que es necesario y el 14% considera de que es poco necesario.

## Capítulo 2

---

- v. El 57% de los encuestados estuvo totalmente de acuerdo con la utilización de la aplicación, el 29% estuvo de acuerdo y el 14% no estuvo de acuerdo.
- vi. El 72% plantea que es excelente la contribución de la aplicación al proceso de gestión de la evaluación de desempeño y el otro 28% considera de que es buena.
- vii. El 57% considera que las facilidades de utilización como medio de apoyo para la evaluación de desempeño es excelente, mientras que el otro 43% considera de que es buena.
- viii. El 86% de los encuestados plantea que están totalmente de acuerdo de la posibilidad de uso del de la aplicación en las demás facultades de la universidad, el 14% restante está de acuerdo.
- ix. El 100% de los encuestados están totalmente de acuerdo de las posibilidades de extensión de la aplicación mediante otras funcionalidades.
- x. El 57% considera que las posibilidades de la utilización de la aplicación es excelente, mientras que el resto considera que es buena.

### **Conclusiones Parciales.**

Luego de describir el desarrollo de la aplicación informática Evaldes, se pueden arribar a las siguientes conclusiones parciales:

- La especificación de los requisitos funcionales y no funcionales permitió definir las historias de usuarios y las tareas ingenieriles de la aplicación.
- El uso de nuevos componentes permitieron un correcto progreso de la aplicación.
- El uso de la unión de las metodologías Scrum y XP permitió generar todos los artefactos necesarios para desarrollar la aplicación deseada.
- Se corroboró mediante la valoración de criterios de expertos, la pertinencia de la investigación realizada y la aplicación desarrollada.

### CONCLUSIONES GENERALES

Al concluir el desarrollo de la aplicación informática Evaldes, se considera haber cumplido con el objetivo trazado al comienzo de la investigación, pudiendo arribar a las siguientes conclusiones:

- El análisis de las soluciones existentes permitió identificar algunas características que tenían en común con la aplicación deseada, lo que a su vez contribuyó agilizar el proceso de desarrollo del software.
- La integración de los nuevos componentes permitió darle cumplimiento a los requisitos funcionales y no funcionales del sistema.
- El desarrollo de la aplicación “Evaldes” contribuirá a mitigar los problemas que hoy presenta el pago adicional mediante la evaluación de desempeño.
- La realización de la validación de la aplicación informática “Evaldes” arrojó resultados satisfactorios concluyendo de esta forma que el objetivo general planteado en el comienzo de la investigación fue cumplido satisfactoriamente.

## RECOMENDACIONES

Una vez concluida el desarrollo de la aplicación se recomienda:

- Trabajar en nuevas mejoras del diseño de la aplicación.
- Agregar nuevas funcionalidades que le permita al administrador poder saber que acciones realizan los usuarios en la aplicación.
- Agregar nuevas funcionalidades que permitan gestionar la pérdida de requisitos.

## BIBLIOGRAFÍA

**Álvarez, Miguel Angel. 2008.** [En línea] 2008.

**Alvarez, Miguel Angel. 2008.** Definición y ejemplos de Sistemas CMS. [En línea] noviembre de 2008.

**Álvarez, Miguel Angel. 2004.** Desarrollo Web. [En línea] 16 de noviembre de 2004. [Citado el: 25 de marzo de 2010.] <http://www.desarrolloweb.com/articulos/1696.php>..

**Álvarez, Sofía y Hernandez, Anaisa. 2000.** *Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML*. Habana : s.n., 2000.

**2009.** Apache.org. [En línea] 2009. <http://www.apache.org>.

**2009.** Apachefriends.org. [En línea] 2009. <http://www.apachefriends.org/es/xampp.html>.

**Aptana Inc. 2009.** [En línea] 2009.

**Beck, K. y M.Fouler. 2000.** *Extreme Programming Explained*. s.l. : Addison Wesley, 2000.

**Booch, G., Rumbaugh, J. y Jacobson. 1999.** *El lenguaje Unificado de Modelado*. 1999.

**Booch, Grady. 2008.** *Object Oriented Analysis and Design with application*. California, Estados Unidos : Addison-Wesley, 2008.

**Casares. 2004.** 2004.

**Castillo, Á. D. 2005.** Webs dinámicas con PHP. [En línea] 2005. [Citado el: 2010 de marzo de 26.] <http://mailxmail.com/curso/informatica/phpprogramacionweb/capitulo4.htm>.

**2009.** codenova.net. [En línea] 2009. [Citado el: 2010 de marzo de 28.] <http://www.codenova.net>.

**Crispin, L y House, T. 2002.** *Testing Extreme Programming*. s.l. : Addison Wesley, 2002.

## Bibliografía

---

**Cristopher, Alexander. 2007.** [En línea] 2007. [Citado el: 12 de abril de 2010.] <http://www.sei.edu/architecture/definitions.html>. IEEE 1471-2000.

Desarrolloweb.com. [En línea] <http://www.desarrolloweb.com/articulos/1178.php>.

**2008.** dev.mysql.com. [En línea] 2008. [http://dev.mysql.com/doc/refman/5.0/es/what\\_is.html](http://dev.mysql.com/doc/refman/5.0/es/what_is.html).

**Dique, Raúl Gonzalez. 2010.** Mundo week. [En línea] 12 de 02 de 2010. <http://mundogeek.net/archivos/2010/02/12/>.

**2008.** drupal.org. [En línea] 2008. [Citado el: 2010 de marzo de 28.] <http://www.drupal.org>.

—. **2010.** Prestashop 5 Estrellas. [En línea] 29 de marzo de 2010. [Citado el: 12 de abril de 2010.] <http://prestashop5estrellas.wordpress.com/2010/03/29/el-patron-mvc-modelo-vista-controlador/>.

**Fowler, Beck. 2000.** *Planeando en programación extrema*. 2000.

**García, Joaquín. 2007.** Patrones de diseño. *Diseño de software orientado a objetos*. [En línea] 27 de mayo de 2007. [Citado el: 14 de abril de 2010.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.

**GARCÍA, JULIA CUERDA. 2007.** Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto. [En línea] 2007. [Citado el: 2010 de marzo de 29.] <http://mosaic.uoc.edu/articulos/cms1204.html>.

**Group, The PHP. 2009.** Conceptos básicos. [En línea] 2009. [Citado el: 2010 de marzo de 26.] <http://www.php.net/manual/es/getting-started.php>.

**JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.

## Bibliografía

---

- Jácome, Dionisio Pérez. 2009.** siteresources.worldbank.org. [En línea] 2009. [Citado el: 18 de abril de 2010.] <http://siteresources.worldbank.org/INTMEXICOINSPANISH/Resources/4-1DionisioPerezJacomeFinal.pdf>.
- James Rumbaugh, I.J., Grady Booch.** *El lenguaje Unificado de Modelado. Manual de Referencia.vol.1.3.* Joomlaos.net. [En línea] <http://www.joomlaos.net/caracteristicas-de-joomla.php>.
- Larman, Craig. 2003.** *UML y Patrones. Edición2.* s.l. : Pearson, 2003.
- LEBLANC, JOSEPH. 2007.** "Learning Joomla! 1.5 Extension Development". 2007.
- LeBlanc, Joseph. 2007.** *Learning Joomla! 1.5Extension Development Creating Modules, Components and Plug-ins with PHP.* Mumbai : Pack Publishing Ltd, 2007.
- Letelier, P. 2004.** *Metodologías ágiles para el desarrollo.* 2004.
- Mendoza, María A. 2004.** *Metodologías de Desarrollo de Software.* 2004.
- Potencier, Fabien y Francois, Zaninotto. 2007.** *Simfony, la guía definitiva.* 2007.
- Qualitrain.** [En línea] [Citado el: 29 de marzo de 2010.] <http://www.qualitrain.com.mx/Metodologias-Agiles-de-Desarrollo-de-Software-Primera-Parte.html>.
- Rational Software Corporation. 2003.** s.l. : s.l. Rational Software Corporation , 2003.
- Reyero, J. A. 2005.** *Características de Drupal.* [En línea] 2005. [Citado el: 2010 de marzo de 29.] <http://drupal.org.es/caracteristicas>.
- Reynoso, Carlos Billy. 2005.** MSDN webcast No2. *Profundizando en Estilos de Arquitectura de Software.* [En línea] 2005. [Citado el: abril de 23 de 2010.] <http://www.microsoft.com/spanish/msdn/latam/mediacenter/webcast/architect.aspx>.

## Bibliografía

---

**Reynoso, Carlos. 2004.** Metodologías de Desarrollo de Software Ágiles. [En línea] 2004. [Citado el: 15 de mayo de 2010.] <http://www.sel.unsl.edu.ar/ApuntesMaes/2004/Metodologias%20Agiles.doc...>

**RV.** El Rincon del Vago. [En línea] [Citado el: 18 de abril de 2010.] [http://html.rincondelvago.com/administracion\\_evaluacion-de-desempeno.html](http://html.rincondelvago.com/administracion_evaluacion-de-desempeno.html).

**Sánchez, M.A.M. 2004.** *Metodologías De Desarrollo De Software*. 2004.

**Serrano, Yelenis y Perdigón, Ekaterina. 2007.** Proceso de desarrollo de aplicaciones Web. Ciudad de la Habana : s.n., 2007.

**Solaas, Harald. 2010.** solaas.com.ar. [En línea] 2010. [Citado el: 19 de abril de 2010.] <http://solaas.com.ar/harald/es/evaldes>.

**Valdés, Damián Pérez. 2007.** ¿Qué es Javascript? *Maestros del Web*. [En línea] 2007. [Citado el: 2010 de marzo de 25.] <http://www.maestrosdelweb.com/editorial/%C2%BF-que-es-javascript/>.

**Vivas White, Pedro y López, Emilio. 2009.** *Cuerpo de Profesores de Enseñanza Secundaria. Informática. Temario Vol. III*. 2009.



## ANEXOS

### Anexo A: Comparación de las metodologías ágiles y las metodologías tradicionales.

*Tabla 1 Diferencias entre las metodologías tradicionales*

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

*Fuente: Tomado de (Fowler, 2000).*

## Anexos

---

### Anexo B: Pruebas de Aceptación.

Caso de Prueba de Aceptación	
<b>Código:</b> HU1_P1	<b>Historia de Usuario:</b> Autenticar Usuario
<b>Descripción:</b> Prueba para la funcionalidad autenticar usuario.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado en el sistema.	
<b>Entrada/ Pasos de ejecución:</b> El usuario inserta los datos necesarios y se ejecuta la acción <b>Iniciar Sesión</b>	
<b>Resultado esperado:</b> El usuario se autentica correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.48 Prueba para la funcionalidad autenticar usuario.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU1_P2	<b>Historia de Usuario:</b> Autenticar Usuario
<b>Descripción:</b> Prueba para la funcionalidad cambiar de contraseña.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado en el sistema.	
<b>Entrada/ Pasos de ejecución:</b> El usuario inserta los datos necesarios y se ejecuta la acción <b>Guardar</b> .	
<b>Resultado esperado:</b> El usuario cambia la contraseña correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.49 Prueba para la funcionalidad cambiar contraseña.*

## Anexos

---

Caso de Prueba de Aceptación	
<b>Código:</b> HU1_P3	<b>Historia de Usuario:</b> Autenticar Usuario
<b>Descripción:</b> Prueba para la funcionalidad recordar contraseña.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado en el sistema.	
<b>Entrada/ Pasos de ejecución:</b> El usuario inserta los datos necesarios y se ejecuta la acción <b>Enviar</b> .	
<b>Resultado esperado:</b> El usuario recibe la contraseña correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.50 Prueba para la funcionalidad recordar contraseña.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU1_P4	<b>Historia de Usuario:</b> Autenticar Usuario
<b>Descripción:</b> Prueba para la funcionalidad recordar usuario.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado en el sistema.	
<b>Entrada/ Pasos de ejecución:</b> El usuario inserta los datos necesarios y se ejecuta la acción <b>Enviar</b> .	
<b>Resultado esperado:</b> El usuario recibe el usuario correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.51 Prueba para la funcionalidad recordar usuario.*

## Anexos

---

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> Gestionar Usuario
<b>Descripción:</b> Prueba para la funcionalidad insertar usuario.	
<b>Condiciones de ejecución:</b> El administrador debe acceder al admin del sistema.	
<b>Entrada/ Pasos de ejecución:</b> El administrador inserta los datos necesarios y se ejecuta la acción <b>Aplicar/Guardar</b> .	
<b>Resultado esperado:</b> El usuario es insertado correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.52 Prueba para la funcionalidad insertar usuario.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P2	<b>Historia de Usuario:</b> Gestionar Usuario
<b>Descripción:</b> Prueba para la funcionalidad eliminar usuario.	
<b>Condiciones de ejecución:</b> El administrador debe acceder al admin del sistema.	
<b>Entrada/ Pasos de ejecución:</b> El administrador modifica los datos necesarios y se ejecuta la acción <b>Aplicar/Guardar</b> .	
<b>Resultado esperado:</b> Se modifican los datos del usuario correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.53 Prueba para la funcionalidad eliminar usuario.*

## Anexos

---

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P3	<b>Historia de Usuario:</b> Gestionar Usuario
<b>Descripción:</b> Prueba para la funcionalidad modificar datos del usuario.	
<b>Condiciones de ejecución:</b> El administrador debe acceder al admin del sistema.	
<b>Entrada/ Pasos de ejecución:</b> El administrador selecciona el usuario y se ejecuta la acción <b>Eliminar</b> .	
<b>Resultado esperado:</b> Se elimina usuario correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.54 Prueba para la funcionalidad modificar datos del usuario.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P1	<b>Historia de Usuario:</b> Gestionar Desempeño
<b>Descripción:</b> Prueba para la funcionalidad registrar profesor.	
<b>Condiciones de ejecución:</b> El administrador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El administrador inserta los datos necesarios y se ejecuta la acción <b>Guardar</b> .	
<b>Resultado esperado:</b> Se registra el profesor correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.55 Prueba para la funcionalidad registrar profesor.*

## Anexos

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P2	<b>Historia de Usuario:</b> Gestionar Desempeño
<b>Descripción:</b> Prueba para la funcionalidad insertar área.	
<b>Condiciones de ejecución:</b> El administrador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El administrador inserta los datos necesarios y se ejecuta la acción <b>Guardar Área</b> .	
<b>Resultado esperado:</b> Se inserta el área correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.56 Prueba para la funcionalidad insertar área.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P3	<b>Historia de Usuario:</b> Gestionar Desempeño
<b>Descripción:</b> Prueba para la funcionalidad evaluar actividad.	
<b>Condiciones de ejecución:</b> El evaluador debe estar previamente registrado en el sistema. El evaluador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El evaluador inserta los datos necesarios y se ejecuta la acción <b>Evaluar Profesor</b> .	
<b>Resultado esperado:</b> Se evalúa el profesor correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.57 Prueba para la funcionalidad evaluar actividad.*

## Anexos

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P4	<b>Historia de Usuario:</b> Gestionar Desempeño
<b>Descripción:</b> Prueba para la funcionalidad actualizar evaluación de desempeño.	
<b>Condiciones de ejecución:</b> El jefe de área debe estar previamente registrado en el sistema. El jefe de área debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El jefe de área inserta los datos necesarios y se ejecuta la acción <b>Evaluar Profesor</b> .	
<b>Resultado esperado:</b> Se evalúa el profesor correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.58 Prueba para la funcionalidad actualizar evaluación de desempeño.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU4_P1	<b>Historia de Usuario:</b> Gestionar Período Evaluativo.
<b>Descripción:</b> Prueba para la funcionalidad insertar período evaluativo.	
<b>Condiciones de ejecución:</b> El administrador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El administrador inserta los datos necesarios y se ejecuta la acción <b>Guardar</b> .	
<b>Resultado esperado:</b> Se inserta el período evaluativo correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.59 Prueba para la funcionalidad insertar período evaluativo.*

## Anexos

Caso de Prueba de Aceptación	
<b>Código:</b> HU4_P2	<b>Historia de Usuario:</b> Gestionar Período Evaluativo.
<b>Descripción:</b> Prueba para la funcionalidad modificar período evaluativo.	
<b>Condiciones de ejecución:</b> El administrador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El administrador inserta los datos necesarios y se ejecuta la acción <b>Modificar</b> .	
<b>Resultado esperado:</b> Se modifica el período evaluativo correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.60 Prueba para la funcionalidad modificar período evaluativo.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P1	<b>Historia de Usuario:</b> Gestionar Aspectos e Indicadores.
<b>Descripción:</b> Prueba para la funcionalidad insertar aspectos.	
<b>Condiciones de ejecución:</b> El administrador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El administrador inserta los datos necesarios y se ejecuta la acción <b>Guardar Aspecto</b> .	
<b>Resultado esperado:</b> Se inserta el aspecto correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.61 Prueba para la funcionalidad insertar aspectos.*



## Anexos

---

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P2	<b>Historia de Usuario:</b> Gestionar Aspectos e Indicadores.
<b>Descripción:</b> Prueba para la funcionalidad insertar indicadores.	
<b>Condiciones de ejecución:</b> El administrador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El administrador inserta los datos necesarios y se ejecuta la acción <b>Guardar Indicador</b> .	
<b>Resultado esperado:</b> Se inserta el indicador correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.62 Prueba para la funcionalidad insertar indicadores.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P3	<b>Historia de Usuario:</b> Gestionar Aspectos e Indicadores.
<b>Descripción:</b> Prueba para la funcionalidad eliminar aspectos.	
<b>Condiciones de ejecución:</b> El administrador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El administrador selecciona los datos necesarios y se ejecuta la acción <b>Eliminar</b> .	
<b>Resultado esperado:</b> Se elimina el aspecto correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.63 Prueba para la funcionalidad eliminar aspectos.*

## Anexos

---

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P4	<b>Historia de Usuario:</b> Gestionar Aspectos e Indicadores.
<b>Descripción:</b> Prueba para la funcionalidad eliminar indicadores.	
<b>Condiciones de ejecución:</b> El administrador debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El administrador selecciona los datos necesarios y se ejecuta la acción <b>Eliminar</b> .	
<b>Resultado esperado:</b> Se elimina el indicador correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.64 Prueba para la funcionalidad eliminar indicadores.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU6_P1	<b>Historia de Usuario:</b> Mostrar Reportes.
<b>Descripción:</b> Prueba para la funcionalidad mostrar reportes.	
<b>Condiciones de ejecución:</b> El usuario debe estar previamente registrado en el sistema. El usuario debe tener los permisos para realizar la operación descrita.	
<b>Entrada/ Pasos de ejecución:</b> El usuario selecciona los datos necesarios.	
<b>Resultado esperado:</b> Se muestra el reporte correctamente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla2.65 Prueba para la funcionalidad mostrar reportes.*

## Anexos

### Anexo C: Instrumento aplicado para la validación mediante Criterio de Expertos.

Marque con una X, según su criterio, sus conocimientos acerca del tema:

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>

Antes de proceder a la valoración de los criterios nos gustaría que autoevalúe su conocimiento acerca del tema en cuestión:

FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted			
Su experiencia obtenida			
Trabajos de autores nacionales			
Trabajos de autores extranjeros			
Su propio conocimiento del estado del problema en el extranjero			
Su intuición			

Debe valorar los aspectos que se relacionan a continuación utilizando una escala de 1 a 5, donde 5 indica el máximo, ajustado a los siguientes términos:

Puntos	Escala					
5	Excelente	Totalmente de acuerdo				
4	Bueno	Muy de acuerdo				
3	Aceptable	De acuerdo				
2	Cuestionable	Parcialmente de acuerdo				
1	Malo	En desacuerdo				
Criterio		Calificación				
Criterio de mérito científico		E	B	A	C	M
1. Calidad de la investigación.						
2. Novedad Científica.						
3. Aporte Científico.						
Criterio de implantación		TA	MA	A	PA	D
4. Necesidad del empleo de la nueva aplicación.						

## Anexos

5. Satisfacción de las necesidades para gestionar el desempeño de los profesores y trabajadores.					
6. Contribución al proceso de gestión de la evaluación de desempeño de los profesores y trabajadores de la facultad.					
<b>Criterios de Generalización</b>	<b>E</b>	<b>B</b>	<b>A</b>	<b>C</b>	<b>M</b>
7. Facilidades de utilización como medio de apoyo a los evaluadores y jefes de área.					
8. Posibilidad de uso en las demás facultades de la universidad.					
9. Posibilidades de extensión de la aplicación mediante la inclusión de otras funcionalidades.					
<b>Criterios de impacto</b>	<b>E</b>	<b>B</b>	<b>A</b>	<b>C</b>	<b>M</b>
10. Posibilidades de utilización de la aplicación informática.					

### Valoración Final:

Elementos a Suprimir:

Elementos a Mejorar:

Elementos a Añadir:

### Categoría Final de la Investigación:

\_\_\_ **Excelente:** *Alta novedad científica, con aplicabilidad y resultados relevantes.*

\_\_\_ **Bueno:** *Novedad Científica. Resultados Destacados.*

\_\_\_ **Aceptable:** *Suficientemente bueno con reservas.*

\_\_\_ **Cuestionable:** *No tiene relevancia científica.*

\_\_\_ **Malo:** *No aplicable.*

**Nota:** Sus criterios y opiniones se manejarán de forma anónima. Le agradecemos de antemano su valiosa colaboración y estamos seguros que sus valiosas experiencias y señalamientos críticos contribuirán a perfeccionar la guía tanto en su concepción teórica como en su futura aplicación.

## Anexos

### Anexo D: Resultados del método de evaluación por criterios de expertos.

*Tabla 2.67: Competencia de los expertos en el tema.*

Experto	Conocimiento	P1	P2	P3	P4	P5	P6	Ka	Kc	K	Competencia
1	8	0.2	0.5	0.05	0.05	0.05	0.05	0.9	0.8	0.85	ALTO
2	9	0.2	0.5	0.05	0.05	0.05	0.05	0.9	0.9	0.9	ALTO
3	6	0.1	0.4	0.05	0.05	0.05	0.05	0.7	0.6	0.65	MEDIO
4	8	0.2	0.4	0.05	0.05	0.05	0.05	0.8	0.8	0.8	ALTO
5	9	0.1	0.4	0.05	0.05	0.05	0.05	0.7	0.9	0.8	ALTO
6	8	0.2	0.5	0.05	0.05	0.05	0.05	0.9	0.8	0.85	ALTO
7	7	0.1	0.4	0.05	0.05	0.05	0.05	0.7	0.7	0.7	MEDIO
<b>Total</b>	7.85714	0.17	0.44	0.05	0.05	0.05	0.05	0.8	0.79	0.80	ALTO

Ka: coeficiente de argumentación del conocimiento.

Kc: coeficiente de conocimiento.

K: coeficiente de competencia.

*Tabla 2.68: Frecuencias absolutas.*

No	Elementos	C1	C2	C3	C4	C5	Total
1	Calidad de la investigación.	3	2	2	0	0	7
2	Novedad Científica.	2	3	2	0	0	7
3	Aporte Científico.	1	4	2	0	0	7
4	Necesidad del empleo de la nueva aplicación.	2	4	1	0	0	7
5	Satisfacción de las necesidades para gestionar el desempeño de los profesores y trabajadores.	4	2	1	0	0	7
6	Contribución al proceso de gestión de la evaluación de desempeño de los profesores y trabajadores de la facultad.	5	2	0	0	0	7
7	Facilidades de utilización como medio de apoyo a los evaluadores y jefes de área.	4	3	0	0	0	7
8	Posibilidad de uso en las demás facultades de la universidad.	6	1	0	0	0	7
9	Posibilidades de extensión de la aplicación mediante la inclusión de otras funcionalidades.	7	0	0	0	0	7

## Anexos

10	Posibilidades de utilización de la aplicación informática.	4	3	0	0	0	7
----	--	---	---	---	---	---	---

**Tabla 2.69: Frecuencias acumuladas**

No	Elementos	C1	C2	C3	C4	C5
1	Calidad de la investigación.	3	2	2	0	0
2	Novedad Científica.	2	3	2	0	0
3	Aporte Científico.	1	4	2	0	0
4	Necesidad del empleo de la nueva aplicación.	2	4	1	0	0
5	Satisfacción de las necesidades para gestionar el desempeño de los profesores y trabajadores.	4	2	1	0	0
6	Contribución al proceso de gestión de la evaluación de desempeño de los profesores y trabajadores de la facultad.	5	2	0	0	0
7	Facilidades de utilización como medio de apoyo a los evaluadores y jefes de área.	4	3	0	0	0
8	Posibilidad de uso en las demás facultades de la universidad.	6	1	0	0	0
9	Posibilidades de extensión de la aplicación mediante la inclusión de otras funcionalidades.	7	0	0	0	0
10	Posibilidades de utilización de la aplicación informática.	4	3	0	0	0

**Tabla2.70: Frecuencias relativas.**

No	Elementos	C1	C2	C3	C4	C5
1	Calidad de la investigación.	0.4286	0.7143	0.9999	0.9999	0.9999
2	Novedad Científica.	0.2857	0.7143	0.9999	0.9999	0.9999
3	Aporte Científico.	0.1429	0.7143	0.9999	0.9999	0.9999
4	Necesidad del empleo de la nueva aplicación.	0.2857	0.8571	0.9999	0.9999	0.9999
5	Satisfacción de las necesidades para gestionar el desempeño de los profesores y trabajadores.	0.5714	0.8571	0.9999	0.9999	0.9999
6	Contribución al proceso de gestión de la evaluación de desempeño de los profesores y trabajadores de la facultad.	0.7143	0.9999	0.9999	0.9999	0.9999
7	Facilidades de utilización como medio de apoyo a los evaluadores y jefes de área.	0.5714	0.9999	0.9999	0.9999	0.9999
8	Posibilidad de uso en las demás facultades de la universidad.	0.8571	0.9999	0.9999	0.9999	0.9999
9	Posibilidades de extensión de la aplicación mediante la inclusión de otras funcionalidades.	0.9999	0.9999	0.9999	0.9999	0.9999

## Anexos

10	Posibilidades de utilización de la aplicación informática.	0.5714	0.9999	0.9999	0.9999	0.9999
----	--	--------	--------	--------	--------	--------

N =	1,82
-----	------

Puntos de Corte									
No	Aspectos	C1	C2	C3	C4	Suma	P	N-P	Categorías
1	I1	-0,18	0,57	3,72	3,72	7,82	1,96	-0,13	Muy adecuado
2	I2	-0,57	0,57	3,72	3,72	7,44	1,86	-0,04	Muy adecuado
3	I3	-1,07	0,57	3,72	3,72	6,94	1,73	0,09	Bastante Adecuado
4	I4	-0,57	1,07	3,72	3,72	7,94	1,98	-0,16	Muy adecuado
5	I5	0,18	1,07	3,72	3,72	8,69	2,17	-0,35	Muy adecuado
6	I6	0,57	3,72	3,72	3,72	11,72	2,93	-1,11	Muy adecuado
7	I7	0,18	3,72	3,72	3,72	11,34	2,83	-1,01	Muy adecuado
8	I8	1,07	3,72	3,72	3,72	12,22	3,06	-1,23	Muy adecuado
9	I9	3,72	3,72	3,72	3,72	14,88	3,72	-1,89	Muy adecuado
10	I10	0,18	3,72	3,72	3,72	11,34	2,83	-2,83	Muy adecuado
Suma		3,51	22,43	37,19	37,19	100,32			
P.de corte		0,35	2,24	3,72	3,72				

## GLOSARIO DE TÉRMINOS

**CASE:** Ingeniería asistida por ordenadores.

**Cliente:** Persona, organización o grupo de personas que encargan la construcción de un producto software.

**CMS:** Sistema de gestión de contenidos (Content Management System). Es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes.

**Framework:** es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**Herramientas:** Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

**Iteraciones:** En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio. Múltiples iteraciones contribuyen a crear un producto completamente integrado.

**Metodología:** del griego (metà "más allá" odòs "camino" logos "estudio"). Se refiere a los métodos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia. Aun cuando el término puede ser aplicado a las artes cuando es necesario efectuar una observación o análisis más riguroso o explicar una forma de interpretar la obra de arte. En resumen son el conjunto de métodos que se rigen en una investigación científica o en una exposición doctrinal.

**SGBD (Sistemas de gestión de base de datos):** Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

**Sprint:** Equivale a una iteración, en la metodología Scrum.

**Usuario:** Persona encargada de utilizar el sistema, obteniendo algún beneficio.