



INSTITUTO SUPERIOR MINERO METALÚRGICO  
“Dr. Antonio Núñez Jiménez”.  
Facultad de Metalurgia-Electromecánica.  
Departamento de Informática

**“SISTEMA DE GESTIÓN DE LA CALIDAD PARA LA EVALUACIÓN  
DE LA CALIDAD DE SITIOS WEB.”**

**Trabajo de Diploma**

*Trabajo de diploma para optar por el título de Ingeniero informático*

Autor: Humberto Torres Cabrera

Tutor: Ing. Eloy Rafael Jiménez Iglesias.

**Moa, Cuba**

**Junio, 2011**

**Año 53 de la Revolución**

## DECLARACIÓN DE AUTORÍA.

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez” para que hagan el uso que estimen pertinente del mismo.

Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del 2011.

Humberto Torres Cabrera

---

Firma del autor

Ing. Eloy R. Jiménez Iglesias.

---

Firma del tutor

## OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título:

Autor:

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad>

<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de <nota 2-Desaprobado, 3-Aprobado, 4-Bien, 5-Excelente>. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

---

Nombre completo del primer tutor

<Grado científico, Categoría docente  
y/o investigativa>

(Si procede)

---

Nombre completo del segundo tutor

<Grado científico, Categoría docente y/o investigativa>

Fecha:\_\_\_\_\_

## ***Pensamiento:***

*“Sólo los necios se encuentran satisfechos y confiados con la calidad de su trabajo”.*

*Mercedes Milá.*

## ***Dedicatoria:***

*Dedico este proyecto a mis padres Humberto Torres Socía y Belquis Cabrera Delfino porque gracias a ellos existo.*

*A mi novia por su ayuda y apoyo en todo momento.*

*A toda mi familia, que me brindó el apoyo que he necesitado para dar un feliz término a este trabajo.*

## *Agradecimientos:*

*Quisiera agradecerles en primer lugar a mis padres, porque sin ustedes nada de esto fuese posible, por su amor y comprensión. Por su esfuerzo y empeño en mi educación. Gracias por su ayuda y cariño.*

*A mi hermano Yoniel Torres por estar siempre ahí, ayudándome y por haber sido siempre un ejemplo a seguir.*

*A mi novia Mirna Dayanis, mi amiga incondicional, quien ha sabido apoyarme y comprenderme aún en los momentos más difíciles.*

*A mis suegros Danilo y Mirna, por haberme dado su amor y apoyo, por considerarme como un hijo más.*

*A mi familia por todo su afecto y ayuda que me brindaron, en especial a mi abuela y mi prima Yurania.*

*A mi tutor gracias por la ayuda y orientación que me diste.*

*A mis compañeros de aula por ayuda desinteresada que me brindaron en el transcurso de estos 5 años, en especial a Ernesto Leyva.*

*Agradezco a esta Revolución que me ha dado la oportunidad de formarme como profesional.*

*Y a todos aquellos que de una forma u otra me tendieron desinteresadamente la mano.*

*A todos, gracias.*

## **Resumen:**

Actualmente las entidades en Cuba dedicadas a la producción de software tienen entre sus principales objetivos desarrollar productos y servicios Informáticos de alta calidad. Para poder obtener estos resultados y lograr una buena posición y un reconocimiento en el mercado, es necesaria la implantación de Modelos o Estándares de calidad que garanticen la comprobación objetiva de la evaluación de la calidad de los productos de software, desarrollados en cada una de las entidades.

En el siguiente trabajo se pretende crear un sistema automatizado que permita evaluar los sitios Web y los productos no conformes de las diferentes áreas del ISMMM. Con el desarrollo del mismo se desea obtener mayor calidad, rapidez y organización a la hora de corregir los problemas encontrados en la entidad. Esta aplicación Web puede llevarse a cabo por los especialistas en la temática o cualquier otra persona interesada en esta actividad.

Para la realización de la investigación se hizo una revisión bibliográfica sobre las aplicaciones Web y las herramientas para la construcción de las mismas, además se presenta la metodología de Ingeniería de Software que se siguió para la construcción del software.

**Palabras Claves:** Calidad, Evaluación.

## **Summary**

Nowadays, enterprises in Cuba are dedicated to the software's production that have as its main objectives to develop some products and informatic services of high quality. For getting these results and to achieve an outstanding place in marketing, it's necessary to implant standards and models of quality that guarantee the evaluation of the quality of the softwares' products developed in every enterprise.

This work is intended to create an automatic system that allows to evaluate the web's siege and the products that are not in agreement with different areas of the ISMM .Whit the development of this work is wanted to get a higher quality fastness and the organization of. The mistakes found in the enterprises. This web application could carry on by some informatic specialist or any other person interested in this activity.

For doing this research work a biographical checking is made about web applications and the tools for its constructions, besides it's presented the methodology of softwares engineering that was followed to the software construction.

# Índice

INTRODUCCIÓN .....	1
CAPÍTULO I – FUNDAMENTACIÓN TEÓRICA .....	6
1.1 Introducción.....	6
1.2 Estado del Arte .....	6
1.2.1 Sistemas Automatizados existentes vinculados al campo de acción .....	6
1.3 Conceptos Fundamentales .....	8
1.3.1 Evaluación de Calidad de recursos web. ....	9
1.3.2 Métodos existentes de evaluación de la Usabilidad .....	9
1.3.3 Tipos de técnicas.....	10
1.4 Propuesta de Solución.....	11
1.5 Tendencias y tecnologías actuales para el desarrollo de aplicaciones.....	12
1.5.1 Lenguajes de programación.....	12
1.5.2 Sistemas Gestores de Base de Datos (SGBD) .....	14
1.5.3 Herramientas CASE.....	15
1.5.4 Metodologías para el desarrollo de sistemas informáticos .....	16
1.6 Herramientas y tecnologías a emplear en la propuesta de solución. ....	18
1.6.1 Fundamentación de la selección del lenguaje a utilizar.....	19
1.6.2 Fundamentación del SGBD a utilizar. ....	20
1.6.3 ¿Por qué elegir XP? .....	21
1.7 Patrones Arquitectónicos .....	21
1.8 Conclusiones del Capítulo. ....	25
CAPÍTULO II – PLANEACIÓN Y DISEÑO .....	27
2.1 Introducción.....	27
2.2.1 Lista de reserva. ....	28
2.2.2 Historias de usuario. ....	28
2.3 Planificación de entregas. ....	30
2.3.1-Estimación de esfuerzos por historias de usuarios .....	31
2.3.2- Planificación de Iteraciones.....	31
2.3.2-Plan de duración de las iteraciones.....	<b>¡Error! Marcador no definido.</b>
2.4-Tarjetas CRC .....	33
2.5 - Modelo de datos .....	34
2.6- Conclusiones .....	35
CAPÍTULO III – DESARROLLO Y PRUEBAS .....	36
3.1 Introducción.....	36
3.2-Desarrollo de las iteraciones.....	36
3.2.1-Tareas por historias de usuario .....	36
3.2.2-Iteración No.1 .....	37
3.2.3-Iteración No.2 .....	38
3.2.4-Iteración No.3 .....	38
3.4-Pruebas .....	39
3.4.1-Desarrollo dirigido por pruebas.....	39
3.4.2- Pruebas de aceptación .....	40
3.5-Conclusiones .....	42
CAPÍTULO IV – ESTUDIO DE FACTIBILIDAD .....	43
4.1 Introducción.....	43

4.2 Efectos Económicos.....	43
4.3 Beneficios y costos Intangibles en el proyecto .....	45
4.4 Ficha de costo .....	46
4.4.1 Costos en Moneda Libremente Convertible (C.U.C) .....	46
4.4.2 Costos en moneda nacional (M.N): .....	47
4.5 Conclusiones del Capítulo .....	49
CONCLUSIONES GENERALES: .....	50
RECOMENDACIONES: .....	51
REFERENCIA BIBLIOGRÁFICA:.....	52
ANEXOS:.....	55
Anexo 1: Historias de Usuarios .....	55
Anexo 2: Tarjeta CRC .....	60
Anexo 3: Tarjeta de Ingeniería .....	62
Iteración 1 .....	62
Iteración 2 .....	67
Iteración 3 .....	69
Anexo 4: Pruebas de Aceptación.....	71
Anexo 5: Principales interfaces del sistema. ....	78



## INTRODUCCIÓN

Desarrollar un producto de software no solo es escribir instrucciones de programación juntas y ejecutarlas en un ordenador, es preciso obtener un producto en tiempo con buena calidad y pocos errores.

La calidad es un activo estratégico clave, del que dependen la mayor parte de las organizaciones, no sólo para mantener su posición en el mercado sino incluso para asegurar su supervivencia. El desarrollo de un producto que satisfaga, en la mayor medida posible, los requerimientos del cliente, es la medida de calidad buscada en la producción.

Una especificación y evaluación integral y detallada de la calidad de los productos de software es un factor clave para asegurar que la calidad sea la adecuada. Esto se puede lograr definiendo de manera apropiada las características de calidad, teniendo en cuenta el propósito del uso del producto de software en la institución.

Es importante especificar y evaluar cada característica relevante de la calidad de los productos de software, cuando esto sea posible, utilizando mediciones validadas o de amplia aceptación, que hagan técnicamente transparente esta actividad.

A nivel mundial en la industria del software hay tendencia al crecimiento de volumen y complejidad de los productos, los proyectos están excesivamente tardes, se exige mayor calidad y productividad en menos tiempo y hay suficiente personal calificado. Esto trae consigo que sea un reto para la industria de software desarrollar las estrategias que le permitan alcanzar un nivel de calidad realmente alto, por lo que se hace necesario todo un estudio para la elección e Implantación del Modelo o Estándar de calidad indicado.

Nuestro país se ha visto envuelto en la necesidad de crear, desarrollar y consolidar los comités de calidad, como vía adecuada para estimular la iniciativa creadora de los trabajadores en el mejoramiento de la calidad, de ahí que tenga gran importancia la constante elevación de la calidad y la eficiencia económica en la generación de productos y la prestación de servicios, al menor costo.



Con el desarrollo de las tecnologías en las universidades se ha notado un notable auge en el desarrollo de sitios Web para la realización de los mismos se debe de seguir una serie de aspectos o normas que sin estos el mismo no tendría la calidad requerida para su comercialización o futura implantación en una empresa o institución determinada.

En el Instituto Superior Minero Metalúrgico de Moa (ISMMM) los sitios Web que se realizan no tienen la máxima calidad. Uno de los factores que influye en una mala aplicación de la Usabilidad en el ISMMM es que los usuarios finales mayormente son excluidos del desarrollo del producto de software y en ocasiones su participación es nula. Los usuarios finales son los que determinan si un software es usable o no, y esto trae consigo que de una forma u otra determine si el software tiene o no calidad. Debe existir una interacción usuario-grupo de desarrollo-cliente desde el negocio para así lograr una mejor calidad en los sitios Web producidos.

Actualmente en el ISMMM el proceso de evaluación de los software que se lleva a cabo es el resultado de una investigación realizada, dicho proceso se realiza de forma manual, lo que hace el proceso más lento, no solo a la hora de su redacción, sino también en el momento de darles seguimiento a las acciones tomadas, además esto aumenta los gastos de papel y otros materiales de oficina e incrementa la morosidad del seguimiento de las mismas, puesto que no permite que la actividad sea interactiva. También se torna un poco engorroso en el momento de generar diversos reportes correspondientes a las no conformidades.

Partiendo de esta situación problemática se decidió llevar a cabo la presente investigación definiendo la siguiente pregunta:

¿Cómo agilizar y a su vez facilitar el proceso de evaluación para la calidad de sitios Web en el ISMMM?

Como **objeto de estudio** se tiene: El Sistema de gestión de la calidad en el proceso de evaluación de sitios Web. De ahí se desprende nuestro **campo de acción** el cual se enmarca en la automatización de los procesos de evaluación de la calidad de sitios Web en el ISMMM.



Para darle solución al problema se propuso el siguiente **objetivo general** de la investigación es desarrollar un Sistema que permita agilizar y facilitar el proceso de evaluación de la calidad de los sitios web en el ISMMM.

Para dar respuesta al problema se plantea como **idea a defender** que: Si se desarrolla una aplicación para la evaluación de los sitios Web se agilizará esta actividad y además permitirá comprobar la eficacia de las mismas en el ISMMM, mejorando así la calidad del producto.

Para darle cumplimiento al objetivo general, se propusieron las siguientes **tareas**:

- ❖ Revisar de la literatura relacionada con el tema y entrevistas realizadas a especialistas.
- ❖ Buscar referencias bibliográficas
- ❖ Lograr un mayor conocimiento de las tecnologías y herramientas actuales necesarias para la posterior implementación del sistema.
- ❖ Analizar la metodología para el desarrollo del software.
- ❖ Desarrollar el análisis y diseño del software
- ❖ Efectuar el desarrollo de la aplicación.
- ❖ Realizar las pruebas al software.

Para complementar estas tareas se han empleado **métodos empíricos** y **teóricos** de la investigación científica.

Los métodos empíricos ayudan en el descubrimiento de los hechos, información, procesamiento de datos y en el conocimiento de las características fundamentales del problema, los que posibilitan su estudio y explicación.

Entre los **métodos empíricos** usados podemos citar:

**Entrevista:** Posibilitó conocer en detalles cuáles eran las necesidades de los especialistas para la evaluación de la calidad del software y además para determinar los principales requerimientos del sistema.

**Análisis de documentos:** Permitió conocer cómo funcionan actualmente el procedimiento de evaluación del software.



Los métodos teóricos se tuvieron en cuenta durante el transcurso de la investigación; pues crearon las condiciones para la interpretación y desarrollo de las teorías de interpretación de los datos obtenidos.

Los **métodos teóricos** utilizados fueron:

**Análisis y Síntesis:** Se utilizó para la recopilación y el procesamiento de la información obtenida en los métodos empíricos y arribar a las conclusiones de la investigación.

**Hipotético \_ deductivo:** Para la elaboración de la idea a defender.

El presente trabajo consta de introducción, 4 capítulos, conclusiones, recomendaciones, bibliografía, glosario de términos y anexos.

**Capítulo 1 “Fundamentación Teórica”:** se ofrece una breve descripción del objeto de estudio y algunos conceptos fundamentales. Se realiza un estudio acerca de los diferentes sistemas existentes vinculados al campo de acción, además se presenta la metodología y las herramientas a utilizar en el desarrollo del sistema propuesto.

**Capítulo 2 “Planeación y Diseño”:** se hace uso de la metodología expuesta en el capítulo inicial para el desarrollo del proyecto, abordando sus dos primeras fases. Se obtiene el modelo de datos del sistema y se desarrollaron las tarjetas CRC.

**Capítulo 3 “Desarrollo y Pruebas”:** se presentan las principales interfaces de la aplicación. La implementación de las tarjetas de ingeniería así como las pruebas realizadas con sus resultados.



**Capítulo 4 “Estudio de Factibilidad”:** se realiza un estudio para ver la factibilidad del producto por la metodología Coste-Beneficio. Además se ilustra el gráfico con el tiempo que demora el proceso de evaluación de forma manual y automatizada.



# CAPÍTULO I – FUNDAMENTACIÓN TEÓRICA

## 1.1 Introducción

En este capítulo se presenta una descripción general del espacio donde radica nuestro objeto de estudio así como del proceso a automatizar además de abordar los conceptos principales asociados al dominio del problema, se realiza un análisis de las herramientas más comunes para el desarrollo de aplicaciones y se fundamenta la elección del lenguaje seleccionado, así como la metodología a utilizar y sus principales características.

## 1.2 Estado del Arte

### 1.2.1 Sistemas Automatizados existentes vinculados al campo de acción

Después de una profunda búsqueda en numerosas fuentes de información y entre ellas Internet (nacional e internacional), se pudo observar que existen varios sistemas automatizados vinculados al campo de acción que le ocupa a este trabajo. Consideremos los siguientes:

**WAMMI** (Web site Analysis and MeasureMent Inventory) es una herramienta creada para la evaluación de la calidad de uso de sitios Web. Es un cuestionario que permite obtener una medida de la facilidad de uso de un sitio Web desde la percepción del usuario. Dependiendo de donde se encuentra el proceso en el ciclo de vida de producción del sitio Web, puede ser usada de tres modos básicos:

- **Predicción.** Permite investigar la reacción de los visitantes antes del lanzamiento definitivo del sitio Web, para investigar la reacción de los visitantes ante el mismo.
- **Monitorización.** Permite conocer las razones de la conducta del usuario, es decir porque no desean regresar al sitio, por que se quejan de aspectos aparentemente triviales, de las costumbres de diseño, etc.



- Referencia. Permite conocer lo que piensan los usuarios del sitio respecto a otros sitios, considerando una serie de aspectos de diseño diferente. (ALVA, 2005)

**SUMI** (Software Usability Measuring Inventory) es un software utilizado para medir la satisfacción y valorar la percepción del usuario de la calidad del software, fue desarrollado por la Universidad College Cork como parte del proyecto MUSiC como una solución a los problemas de medición de la percepción de usabilidad del software por parte del usuario. Proporciona un método válido para la comparación tanto de productos como de diferentes versiones del mismo producto.

SUMI debe ser aplicado a una muestra de usuarios con alguna experiencia con el tipo de software que va a evaluarse para poder obtener resultados confiables. Establece el uso de 10 usuarios representativos para conseguir resultados satisfactorios. Puede ser utilizado para evaluar un producto o serie de productos con el fin de realizar una comparación producto-producto o comparar el producto contra la base de datos estandarizada que permita ver como el producto que esta siendo evaluado compara contra el perfil promedio establecido en el mercado. (ALVA, 2005)

**PROKUS** (PROgramm system zur Kommunikations ergonomischen UnterSuchung rechnerunterstützter Verfahren) es una herramienta enfocada al mercado, desarrollada en el Laboratorio del Instituto de Ingeniería Humana e Industrial de la Universidad de Karlsruhe, Alemania. Es un sistema para el diseño de procedimientos de evaluación para llevar a cabo evaluaciones de usabilidad de acuerdo a diferentes situaciones de evaluación. Está herramienta está basada en una evaluación con líneas guía para la evaluación sistemática de mercados, pruebas de conformidad y pruebas de comparación. Está fundamentada en que la calidad de un sistema de software altamente interactivo con el humano depende del diseño ergonómico de la interfaz de usuario. PROKUS esta basada en ISO 9241-10 como criterios de evaluación obligados para el usuario cuando evalúa, selecciona o compra un producto software en el mercado]. Esta evaluación de



productos software puede realizarse en distintas etapas en el ciclo de vida del software:

- Durante el proceso de desarrollo de los productos, cuya finalidad es evaluar si la interfaz es la adecuada, empleando para ello pruebas de calidad (evalúa el funcionamiento del producto) y pruebas de conformidad (compara el producto con los requisitos del estándar).
- Durante la selección de un producto, realizado por clientes o futuros usuarios, mediante pruebas de comparación (examinando y comparando los productos existentes en el mercado) y pruebas de conformidad (evaluando la adherencia del producto a los estándares de amigabilidad al usuario).
- Durante la instalación del producto, con la finalidad de dar soporte a la fase de rediseño, mediante pruebas de usabilidad (para medir el logro de las metas de usabilidad y las necesidades adicionales). (ALVA, 2005)

### 1.3 Conceptos Fundamentales

**La gestión de software(GS):** Es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos; garantizando la calidad del producto. La GS comprende la gestión de configuración y la gestión de proyecto. (ZALDIVAR, 2010)

**Calidad de Software:** Se define como la ausencia de errores de funcionamiento, la adecuación a las necesidades del usuario, y el alcance de un desempeño apropiado (tiempo, volumen, espacio), además del cumplimiento de los estándares. Los objetivos que la calidad persigue son: La aceptación (utilización real por parte del usuario) y la Mantenibilidad (posibilidad y facilidad de corrección, ajuste y modificación durante largo tiempo. La calidad del software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. (ZALDIVAR, 2010)



### 1.3.1 Evaluación de Calidad de recursos web.

Se distinguen dos formas de llevar a cabo la evaluación de calidad de recursos Web los cuales se mencionan a continuación:

a) Mediante la aplicación de un método formal, científico, riguroso, que permita medir de manera objetiva a través de varios parámetros si estos recursos cumplen o alcanzan unos niveles mínimos exigidos de calidad, especialmente la relacionada con la propia información.

b) Mediante un proceso inductivo-deductivo resultante del cumplimiento de un conjunto de indicadores, y a través del cual el usuario se forma una opinión o juicio totalmente subjetivo sobre la calidad del recurso basándose sobre todo en la utilidad y uso que pretenda hacer de dicho recurso. (COLUMBIÉ, 2010)

La recuperación de información de sitios que ofrezcan información de calidad evaluada de manera objetiva no implica que el usuario no deba de evaluar subjetivamente dicha información. Considerando a Internet como un Sistema de Información, al igual que lo son los diferentes directorios temáticos y motores de búsqueda, se puede aplicar el mismo principio que rige la evaluación del éxito de los sistemas de información en general y que se basa en dos principales dimensiones: la calidad de la información y la satisfacción del usuario. Por lo tanto, siempre es el usuario quien tiene la última palabra y decide la relevancia de un recurso para satisfacer sus necesidades de información, y por ende, su calidad, aunque se le pueda facilitar el proceso de excluir y filtrar aquellos recursos que con toda seguridad no cumplan unos criterios mínimos de calidad. (COLUMBIÉ, 2010)

### 1.3.2 Métodos existentes de evaluación de la Usabilidad

Existen algunos métodos para la evaluación de la Usabilidad teniendo en cuenta las diferentes formas de captar la información, y de ésta lo más importante que nos puedan transmitir los usuarios. Algunas de estas técnicas son las siguientes:



El **Intercambio** es uno de los métodos más utilizados ya que los resultados dependen directamente de que los usuarios encuentren las dificultades que tiene dicho sistema para su interrelación diaria, para su aprendizaje o quizás para su entretenimiento.

La **Observación** es un método sencillo de analizar el nivel de Usabilidad que tiene algún software determinado. Mediante una ojeada superficial una persona que tenga suficiente conocimiento del tema puede decir qué tan usable o no es el software.

Las **encuestas** son de los métodos más directos con que cuentan los desarrolladores para saber el criterio, la opinión que existe sobre el sistema de todos los que hacen algún tipo de uso de él.

Los **expertos** son un grupo de personas que tienen los conocimientos para hacerle varios tipos de evaluación para saber si un software es usable o no. (COLUMBIÉ, 2010)

### 1.3.3 Tipos de técnicas.

- **Inspección:** Es una técnica que emplea expertos; que son personas que conocen del tema en cuestión, tienen estudios sobre este y una clasificación especial para poder evaluar diferentes productos.

a) **Heurística:** Proceso de aplicar reglas o principios de diseño de interfaces a un sistema o prototipo, identificando problemas de Usabilidad. Evalúa la consistencia, mensajes de error, el lenguaje, la existencia de ayuda en línea y elementos como íconos, entre otros.

b) **Recorrido cognitivo:** se centra en evaluar en un diseño su facilidad de aprendizaje, básicamente por exploración y está inspirado en la observación que muchos usuarios prefieren aprender software a base de explorar las posibilidades que éste ofrece. Dicha técnica tiene algunas deficiencias que no le permiten estar un escalón más arriba en comparación con las demás técnicas y la principal de estas deficiencias es la ausencia total de los usuarios finales en el proceso de evaluación. (DE MIGUEL, 2008)



- **Indagación:** Se realiza mediante la recogida de información; ya sea observando, preguntando o interactuando con los usuarios. Se utiliza a menudo en las primeras etapas del proceso de desarrollo de software que es donde se necesita una mayor información con respecto al trabajo que se va a realizar.

a) Entrevista: Se realiza con el objetivo de obtener datos que servirán para realizar la evaluación con la(s) técnica(s) determinadas a utilizar.

b) Cuestionario: Su función es muy parecida a la entrevista, a diferencia que este es un poco más específico a la hora de realizar las preguntas pertinentes.

- **Test:** Se realiza con los usuarios cuando se les da tareas concretas y así poder obtener los resultados, como hasta qué punto se cumplió o no el objetivo de dichas tareas.

a) Pensando en voz alta (thinking-aloud): Técnica formal cuyo objetivo es recoger comentarios u observaciones del usuario. Para ellos se le hace preguntas relacionadas con la lista de tareas de la prueba de Usabilidad.

b) Card Sorting: Es una técnica de ayuda en la toma de decisiones para realizar una organización de categorías centrada en el usuario. Esta técnica la lleva a cabo el arquitecto de Información.

El tipo de técnica de evaluación a utilizar será un test a usuarios en forma de encuesta. (DE MIGUEL, 2008)

### 1.4 Propuesta de Solución

Desarrollar una aplicación web que permita agilizar y facilitar el proceso de evaluación de la calidad de los sitios Web, que permita la gestión de los software y los aspectos que se deseen evaluar, además que de la posibilidad a los usuarios finales a interactuar con dicho sistema para así logra una mejor calidad.



### 1.5 Tendencias y tecnologías actuales para el desarrollo de aplicaciones.

Ante el incesante avance de las tecnologías, la sociedad, ávida de nuevas herramientas y funcionalidades, exige a los desarrolladores de software nuevos retos y nuevas concepciones para satisfacer sus exigencias, cada vez más ambiciosas por lo que los desarrolladores deben buscar nuevas ideas surgiendo así nuevas metodologías y formas de desarrollo que permitan confeccionar productos cada vez más complejos.

#### 1.5.1 Lenguajes de programación

**Personal Home Page** - (PHP) es el acrónimo de procesador hipertexto (Hipertexto Preprocessor). Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML. Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. (HERNÁN, 2006)

PHP es un lenguaje de programación de estilo clásico, esto significa que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. No es un lenguaje de marcas como podría ser HTML, XML o WML. A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor, como por ejemplo podría ser, una base de datos. (HERNÁN, 2006)

PHP es la gran tendencia en el mundo de Internet, últimamente puede observar un ascenso imparable, puesto que cada día son muchas las páginas Web que lo utilizan para su funcionamiento, según las estadísticas, PHP se utiliza en más de 10 millones de páginas, y cada mes realiza un aumento del 15%. Como síntesis, PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de



aplicaciones Web muy robustas, y contiene unas 40 extensiones estables sin contar las que se están experimentando, igualmente tiene soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros, además de que:

- Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas.
- Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si está familiarizado con esta sintaxis, resultará un poco mejor aprender PHP.
- Su librería estándar es realmente amplia, lo que permite reducir los llamados “costes ocultos”, uno de los principales defectos de ASP.
- PHP tiene una de las comunidades más grandes en internet, esto permite encontrar fácilmente ayuda, documentación, artículos, noticias y otros recursos.
- Permite las técnicas de Programación Orientada a Objetos (POO).
- Posibilita crear los formularios para la Web. (HERNÁN, 2006)

**HTML** – El lenguaje HTML es el idioma de la Web. Se basa en el uso de “Etiquetas” para la definición del formato del texto, los distintos elementos que conforman la página, sus propiedades y disposición. Este lenguaje es interpretado por los navegadores, procesado y convertido en una Web tal como la vemos en la pantalla, con imágenes, tablas, texto, videos y toda clase de elementos. El lenguaje está compuesto por etiquetas o marcas, gracias a ellos es posible darles forma a todos los componentes de una página o un documento HTML.

Las etiquetas de HTML están divididas en etiquetas de apertura y de cierre, aunque no siempre existen estas últimas. (HERNÁN, 2006)



**JavaScript** – Es un lenguaje pensado para agregar interactividad con el usuario a las páginas HTML. Permite ejecutar secuencias de comandos en el mismo navegador del usuario. Con JavaScript se puede realizar cálculos rápidos y complejos, verificar formularios antes de enviarlos, crear calendarios, convertir divisas. Es un lenguaje que distingue entre minúscula y mayúscula, no exige la declaración explícita de las variables, es posible crear las variables. Es importante saber que JavaScript no lo soportan todos los navegadores por lo que nos vemos en la situación de probar el código resultante en más de un navegador. La sintaxis es muy parecida a C o C++, por lo que se convierte en un lenguaje fácil para el que lo domine. (HERNÁN, 2006)

### 1.5.2 Sistemas Gestores de Base de Datos (SGBD)

Actualmente existen muchos sistemas gestores de bases de datos, entre ellos, analizaremos las características y facilidades que brinda cada uno de los que se han tenido en consideración, los que siguen son: PostgreSQL y MySQL.

**PostgreSQL** - es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, no es un sistema de gestión de bases de datos puramente orientado a objetos. PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo y proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. Entre las que se encuentran:

- Implementación del Standard SQL92/SQL99.
- Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. (HERNÁN, 2006)



**MySQL** - es un sistema de gestión de base de datos relacional, multi-hilo y multiusuario, con más de seis millones de instalaciones. Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. (HERNÁN, 2006)

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius. (HERNÁN, 2006)

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. (HERNÁN, 2006)

### 1.5.3 Herramientas CASE

- **CaseStudio:** Es una herramienta que nos posibilita relacionar las tablas de una base de datos y normalizar las mismas, con el objetivo de eliminar campos repetidos en esta. Desarrolla todo el proceso de una manera más rápida, está hecho para relacionar una gran cantidad de tablas en un pequeño período de tiempo, aunque se puede relacionar pocas tablas también. (Herramientas CASE, 2004)
  
- **MYSQL Workbench 5.1 OSS:** Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. Se diseña para hacer más fácil de entender el estado actual de los datos. Simple y fácil al usuario, ayuda a



organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia. (MySQL,2008)

- **Embarcadero ER/Studio:** Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. Se diseña para hacer más fácil de entender el estado actual de los datos de las empresas. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. (ER/Studio,2008)
  
- **Rational Rose:** proporciona un lenguaje de modelado común para permitir la creación más rápida de la calidad de software. Incluye Unified Modeling Language (UML) de apoyo y es una de la más completa de productos de la familia Rational Rose. Proporciona la base de datos para el modelado UML diseños, con la capacidad de representar a la integración de los datos y los requisitos de las aplicaciones a través de diseños físicos y lógicos. (Rational Rose,2008)

### 1.5.4 Metodologías para el desarrollo de sistemas informáticos

El desarrollo de software no es una tarea fácil, prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros. (MONMANY, 2007)



Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. (MONMANY, 2007)

Esta es la filosofía de las metodologías ágiles, las cuáles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software. Algunas de las metodologías para el desarrollo informático son:

**RUP.** (Proceso Unificado del Racional)-Es una metodología para proyectos más largos, debido a su gran cantidad de diagramas que lleva consigo, además se documenta poco sobre el Sistema que se está llevando a cabo.

Las principales características de este proceso unificado son:

- Dirigido por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

Y además utiliza un solo lenguaje de modelación (UML).

**XP.** (Programación Extrema)-Metodología que adopta 12 prácticas que se pueden utilizar todas o no, eso lo deciden el programador y el cliente según las necesidades de este último o si la aplicación no requiere de todas. Se centra especialmente en documentar en forma de plantillas, tiene cuatro fases:

Planeación, Diseño, Desarrollo o Implementación y Pruebas. En la primera fase se generan como artefactos los usuarios del negocio, las historias de usuarios, la lista de reserva del producto, el plan de iteraciones, entre otros. En la segunda se tiene el modelo de datos, tarjetas CRC. En tercera fase se desarrollaron las tareas de ingeniería y la cuarta fase son efectuadas las pruebas al software para verificar que el mismo cumpla con todos las funcionalidades acordadas, estas pruebas pueden ser aceptadas por el cliente o denegadas por el mismo.



**SXP.** (Scrum + Programación. Extrema). Esta Metodología fue elaborada en la Universidad de las Ciencias Informáticas, escogieron una parte de Scrum y otra parte de XP, para una mayor organización dividen en carpetas todas sus plantillas. Pero no existe una documentación amplia de esta metodología como en XP.

### **1.6 Herramientas y tecnologías a emplear en la propuesta de solución.**

**Macromedia Dreamweaver** – Es un editor de texto o un entorno de desarrollo donde el Webmaster puede olvidarse de las partes más tediosas del diseño, como tablas, formularios, y demás elementos.

Es una de las herramientas más utilizadas para el trabajo de aplicaciones visuales, el programa se adapta increíblemente a las necesidades de todo tipo de profesional del diseño Web tanto como para los que deseen programar el código como para los que gustan de una metodología totalmente visual.

Soporta varios lenguajes tales como: PHP, ASP. HTML, JavaScript o CSS. Otra característica interesante del programa es su integración con Flash y Fireworks también productos de Macromedia.

Permite insertar algunos elementos básicos en Flash sin necesidad de tener este programa instalado, como botones, viñetas y textos. Finalmente si queremos potenciar el programa podemos instalarle gran cantidad de plug-ins, o extensiones, los cuales pueden ser descargados del sitio de Macromedia o bien podemos programarlos nosotros mismos. (HERNÁN, 2006)

**XAMPP** - Es un paquete independiente de plataforma, software libre, incluye el servidor Web Apache que fue el que se utilizó en la aplicación, además trae el servidor de datos MySQL, sus respectivos gestores phpMyAdmin y phpSQLiteAdmin, el intérprete del lenguaje homónimo PHP con los extras incluidos en PEAR, el intérprete del lenguaje Perl. El programa esta liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de



interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux. (BASULTO, 2010)

- Incluye chequeo de seguridad.
- Contiene un panel de control.

**MYSQL Workbench 5.1 OSS** - Después de un estudio de las herramientas que se abordaron anteriormente se escogió esta ya que nos proporciona muchas ventajas.

- Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia.
- La creación de diagramas es clara y rápida.
- Tiene la posibilidad de realizar diagramas con desempeño rápido.
- Te da la posibilidad de darle un nombre a la Base de Datos ante de exportarla.

### **1.6.1 Fundamentación de la selección del lenguaje a utilizar.**

**PHP v5.0** - Las iniciales PHP significan "PHP Hipertexto Pre-processor y se trata de un lenguaje de programación que es usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Es un lenguaje de programación usado generalmente para la creación de contenido para sitios o aplicaciones Web. La versión 5 de PHP presenta un magnífico trabajo con el paradigma orientado a objeto que permite la reutilización de código entre otras facilidades.

Ventajas de trabajar con PHP comparado con otros lenguajes similares:

- Es un lenguaje multi plataforma.
- Rapidez de ejecución.
- Mantiene un bajo consumo de recursos de máquina.
- Gran seguridad, muy poca probabilidad de corromper los datos.



- Capacidad de conexión con la mayoría de los manejadores de bases de datos que se emplean en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación en internet, incluyendo una gran variedad de ejemplos y de ayudas.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de programación orientada a objetos.
- Permite crear formularios para la Web.
- No requiere definición de tipos de variables ni manejo detallado de bajo nivel.

### **1.6.2 Fundamentación del SGBD a utilizar.**

**MySQL v5.0** - MySQL es muy rápido, fiable y fácil de usar, surge para manipular bases de datos muy grandes. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, cuenta con un sistema de privilegios contraseñas muy seguro que permite la autenticación básica para el acceso al servidor. MySQL es un sistema de administración de base de datos. Opera en una arquitectura cliente/servidor.

Es el sistema gestor de bases de datos más popular, además que cualquiera puede estudiar su código y comprenderlo fácilmente.

Luego de analizadas las características y facilidades del SGBD presentado, y la de la herramienta a desarrollar se decide usar el MySQL como SGBD, por las siguientes razones:

- No se necesitará de un manejo complejo de la información.
- El PHP maneja más fácil al MySQL, debido a la gran cantidad de funciones que tiene explícita.
- El MySQL es multiplataforma.



### 1.6.3 ¿Por qué elegir XP?

Actualmente XP es la metodología ágil más documentada (hay una colección de libros “XP Series” de Addison Wesley) y extendido. Existe una gran comunidad de desarrolladores XP. Otra de las ventajas de XP es que no es necesario adoptarlo en forma completa, sino que pueden utilizarse varias de sus prácticas en forma independiente. Esto hace que el costo de su implementación sea mucho más accesible que el de otras metodologías. Algunas de las ventajas que tiene XP:

- Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas)
- Puede ser implementado en forma gradual.
- Puede adaptarse a las necesidades de cualquier equipo de desarrollo. De hecho, Kent Beck recomienda a los equipos que lo adapten a sus necesidades (BECK, 1999).
- Exige que se establezca una comunicación más fluida con el cliente y que este tenga mayor participación en el proceso de desarrollo. La consecuencia de esto es que el cliente se involucre más en el desarrollo del producto.
- Actualmente es la metodología ágil más extendida y documentada.
- Se realizan pruebas constantemente del sistema.

## 1.7 Patrones Arquitectónicos

**Patrón de arquitectura MCV (Modelo vista controlador)** - Para el diseño de aplicaciones con sofisticadas interfaces se emplea el patrón de diseño MCV. (BASULTO, 2010)

Si se realiza un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces, la consecuencia será que, cuando se necesite cambiar la interfaz, tendrá que modificarse trabajosamente los componentes de negocio, por lo que propiciará mayor trabajo y más riesgo de error. (BASULTO, 2010)

Se trata de realizar un diseño que desacople la vista del modelo, con el fin de perfeccionar la reusabilidad.

**Modelo-Vista- Controlador (MVC)**

Elementos del patrón:

- Modelo: datos y reglas de negocio (Base de Datos).
- Vista: muestra la información del modelo a los usuarios.
- Controlador: gestiona las entradas de los usuarios.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información en una base de datos, que puede presentarse de diversas formas: diagrama de pastel, de barras, tabular, etc.

El modelo es responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Definir las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: “si Ud. es un profesor y no entra a la aplicación, consultar el Administrador de la misma”.
- Llevar un registro de las vistas y controladores del sistema.
- Si se está en presencia de un modelo activo, el mismo notificará a las vistas los cambios que en los datos pueda producir un agente externo.

El controlador es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).



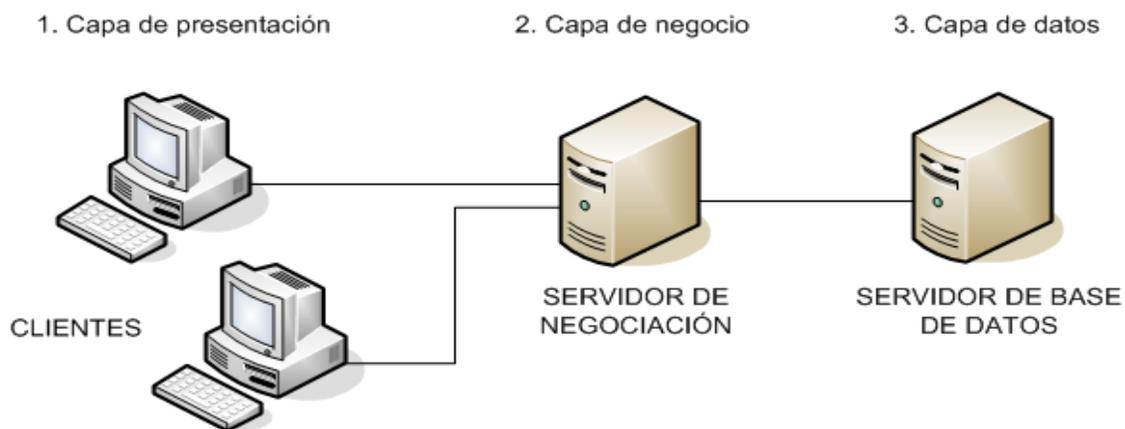
- Contiene reglas de gestión de eventos, del tipo “si el evento z, entonces acción w”. estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar.

Las vistas son responsables de:

- Recibir datos del modelo y mostrarlo al usuario.
- Pueden suministrar el servicio de actualizar, para que sea invocado por el controlador o por el modelo, (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

### Arquitectura en tres capas

El diseño de sistemas informáticos suele usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables, (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). (BASULTO, 2010)



### Capas o niveles

- Capa de presentación o interface: es la capa que le permite al usuario interactuar con el sistema, captura y le comunica la información al mismo, dando un mínimo de proceso, (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la del negocio. (BASULTO, 2010)



- Capa de lógica o de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio e incluso lógica del negocio, pues es aquí donde se establecen las reglas que deben cumplirse. Esta capa se comunica con la de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos para solicitar al gestor de bases de datos para almacenar o recuperar datos de él. (BASULTO, 2010)
- Capa de datos: es donde se ubican los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de los mismos, reciben solicitudes de almacenamiento o de recuperación de información desde la lógica del negocio. (BASULTO, 2010)

Todas estas capas pueden residir en un único ordenador, esto no sería lo normal, lo más usual es que haya una multitud de ordenadores donde reside la capa de interface (son los clientes de la arquitectura cliente/servidor).

Las capas de negocio y de datos pueden residir en un mismo ordenador, y si el crecimiento de las necesidades lo aconseja, pueden dividirse en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, pueden separarse en varios ordenadores los cuáles recibirán las peticiones del ordenador en que resida la capa de negocio. Si por el contrario, la complejidad fuese en la capa de negocio lo que obligase a la separación, esta lógica del negocio podría residir en uno o más ordenadores que realizarían las solicitudes a una única base de datos. (BASULTO, 2010)

En una arquitectura de tres niveles, los términos “capas” y “niveles” no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: interface/lógica del negocio/datos.

En cambio, el término nivel, corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo: Una solución de tres capas (interface, lógica, datos), que residen en un solo ordenador (interface+lógica+datos), se dice que la arquitectura de la solución es de tres capas y un nivel.



## CAPÍTULO I – FUNDAMENTACIÓN TEÓRICA

---

Una solución de tres capas que residen en dos ordenadores (interface+lógica, lógica+datos), se dice que la arquitectura de la solución es de tres capas y dos niveles. Una solución de tres capas que residen en tres ordenadores, la arquitectura que la define es: una solución de tres capas y tres niveles. (BASULTO, 2010)

Para la implementación de la aplicación se escogió la arquitectura en tres capas, porque aumenta la escalabilidad del sistema antes futuros cambios y adición de funcionalidades, y nos proporciona la reutilización de soluciones.



### **1.8 Conclusiones del Capítulo.**

En este capítulo quedaron reflejados todos los conceptos relacionados con el problema. Se ha podido apreciar todo lo referente a la base teórica que fundamenta esta investigación, dando a conocer al lector los principales conceptos que se manejan, las diferentes tecnologías para el desarrollo de aplicaciones, los sistemas automatizados que existen en el mundo vinculados al campo donde se proyecta el objeto de estudio así como la propuesta de solución con las herramientas a utilizar, fundamentación del lenguaje de programación a utilizar, el Sistema Gestor de Base de Datos y la metodología que se eligió para resolver el problema planteado, así como los patrones definidos para el desarrollo de este trabajo.



### CAPÍTULO II – PLANEACIÓN Y DISEÑO

#### 2.1 Introducción

En este capítulo, se introduce la fase de planeación y diseño, donde se detallan las necesidades del cliente, se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema para su terminación.

Símbolo	Intención	Extensión
especialistas	Se le denomina especialistas a todos aquellos usuarios que se le asignan ciertos privilegios para operar sobre el sistema (puede ser profesor o administrador del sistema).	Personas
Usuarios	Se le denomina Usuarios todos aquellos usuarios que pueden ejecutar las opciones no privilegiadas que brinda el sistema (invitados).	Personas
Aspectos a chequear	Se refiere a los parámetros por los cuales se realiza la evaluación del sitio web	Cuestionarios
Evaluaciones	Se refiere a los diagnósticos realizados a través de los aspectos a chequear	-

Tabla 2.1 Definición de entidades y conceptos fundamentales



### 2.2.1 Lista de reserva.

De acuerdo a lo antes expuesto, el sistema debe ser capaz de:

- Autenticar usuarios
- Insertar usuarios.
- Eliminar usuarios.
- Cambiar contraseña.
- Mostrar listado de usuarios.
- Insertar proyecto
- Eliminar proyecto.
- Listar proyecto.
- Insertar los aspectos a chequear.
- Eliminar los aspectos a chequear.
- Modificar los aspectos a chequear.
- Mostrar aspectos a chequear de estudiantes.
- Mostrar aspectos a chequear de especialista.
- Evaluar proyecto.
- Eliminar listado de evaluaciones.
- Mostrar listado de evaluaciones.
- Comparar proyecto.
- Graficar proyecto.
- Mostrar resumen valorativo.
- Exportar resumen valorativo.

### 2.2.2 Historias de usuario.

Las HU, son las técnicas utilizadas en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas *tormenta de ideas (brain storm)* arrojan



no solo los requerimientos, sino también las posibles soluciones. Las HU permiten responder ágilmente a los requerimientos cambiantes y aunque se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas. Para definir las se emplea la siguiente plantilla:

Historia de usuario	
<b>Número:</b> No. Historia de usuario	<b>Usuario:</b> Usuario entrevistado para obtener la función requerida a automatizar.
<b>Nombre:</b> nombre de la historia de usuario que sirve para identificarla mejor entre los desarrolladores y el cliente.	
<b>Prioridad en el negocio:</b> Importancia: Alta / Media / Baja	<b>Riesgo en desarrollo:</b> Dificultad: Alta / Media / Baja
<b>Puntos estimados:</b> Estimación: de 1 a 3 puntos	<b>Iteración asignada:</b> Iteración a la que corresponde
<b>Programador responsable:</b> Nombre de encargado de programación.	
<b>Descripción:</b> Se especifican las operaciones por parte del usuario y las respuestas del sistema.	
<b>Observaciones:</b> Algunas observaciones de interés, como glosario, información sobre usuario etc.	

Tabla 2.2. Planilla de historia de usuario.

### Las historias de usuario descritas por el cliente:

Historia de usuario	
<b>Número:</b> 1	<b>Usuario:</b> Cliente.
<b>Nombre :</b> Autenticación de usuarios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta



<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Humberto Torres Cabrera	
<b>Descripción:</b> Los usuarios del sistema ingresan sus datos para necesitar entrar al sistema (nombre de usuario y contraseña). El sistema verifica que los datos estén correctos, en caso de que no sean correctos los mismos la aplicación muestra un mensaje de error.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.3 HU No. Autenticar usuarios

Para consultar el resto de las HU ir: [Anexo1](#)

### 2.3 Planificación de entregas.

En esta fase se establece la prioridad de cada HU, y a continuación, se realiza una estimación del esfuerzo necesario de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de dos o tres meses.

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción, este punto de estimación que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor de 1 a 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.



### 2.3.1-Estimación de esfuerzos por historias de usuarios

Para el buen desarrollo del sistema propuesto, se realizó una estimación para cada una de las HU identificadas, y se obtienen los resultados que se muestran a continuación:

<b>Historias de usuarios</b>	<b>Puntos de estimación</b>
<b>Autenticar usuarios</b>	<b>1 semana</b>
<b>Gestionar usuarios.</b>	<b>1 semana</b>
<b>Gestionar proyectos.</b>	<b>1 semana</b>
<b>Gestionar aspecto a chequear.</b>	<b>1 semana</b>
<b>Mostrar aspectos a chequear</b>	<b>1 semana</b>
<b>Evaluar proyectos.</b>	<b>2 semanas</b>
<b>Gestionar evaluaciones.</b>	<b>1 semana</b>
<b>Comparar proyecto.</b>	<b>2 semanas</b>
<b>Graficar proyecto.</b>	<b>2 semanas</b>
<b>Mostrar resumen valorativo.</b>	<b>1 semana</b>
<b>Exportar resumen.</b>	<b>2 semanas</b>

Tabla 2.13- Estimación de esfuerzo por historia de usuario

### 2.3.2- Planificación de Iteraciones

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este plan se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU se incluirán en un reléase, lo cual debería ser consistente con el contenido de una o dos iteraciones. En relación con lo antes tratado se decide realizar el sistema en 3 iteraciones, las cuales se explican de forma detalla a continuación:



### Primera iteración:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU 1 ,2 ,3 4 y 5 las cuales hacen alusión a la autenticación de usuarios, gestión de usuarios, gestión de proyectos, gestión de los aspectos a chequear y mostrar los aspectos a chequear.

### Segunda iteración:

Esta iteración tiene como finalidad desarrollar las HU 6, 7, y 8. Las mismas son las que brindan las funcionalidades de evaluar el proyecto, gestionar evaluaciones y comparar proyecto.

### Tercera iteración:

Esta última iteración de la aplicación tiene como propósito llevar a cabo el desarrollo de la HU 9, 10 y 11 las cuáles proporcionan las funcionalidades de graficar proyecto, generar y exportar resumen valorativo. Estas historias de usuario serán integradas con el resultado de las iteraciones anteriores, y como fruto de esta integración se obtendrá la versión 2.0 del producto final. A partir de este momento la aplicación será puesta a un proceso de prueba para evaluar el desempeño de la misma.

### 2.3.2-Plan de duración de las iteraciones

Iteración	Historias de usuario	Duración total
Iteración 1	Autenticar usuarios.	5 semanas
	Gestionar usuarios.	
	Gestionar proyectos.	
	Gestionar aspectos a chequear.	
	Mostrar los aspectos a chequear	
Iteración 2	Comparar proyecto.	5 semanas
	Evaluar el proyecto.	



	Mostrar listado de evaluaciones	
Iteración 3	Graficar proyectos.	5 semanas
	Mostrar resumen valorativo.	
	Exportar resumen.	

Tabla 2.3.2 Plan de duración de las iteraciones

### 2.4-Tarjetas CRC

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como *tarjetas CRC*, las cuáles se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño. Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU.

Clases	
Responsabilidades	Colaboraciones
Clase A	Clase B

Tabla 2.15 Plantilla de tarjeta CRC.

Clases: Autenticar _ usuarios	
Responsabilidades	Colaboraciones
Autenticar usuarios	Conexion.

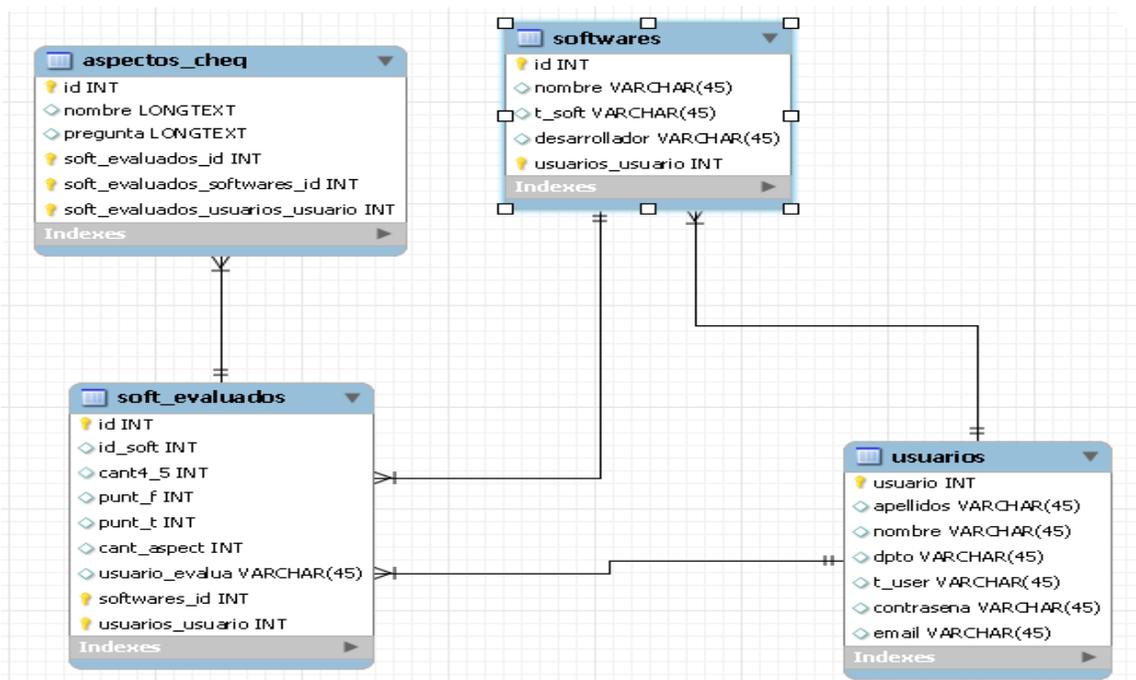
Tabla 2.16 Tarjeta CRC No.1 Autenticar usuarios



Para consultar el resto de las tarjetas CRC ir al: [Anexo 2](#)

### 2.5 - Modelo de datos

Es el modelo físico de la Base de Datos con la cual trabaja esta Aplicación Web, nos sirve para entender con más facilidad como está diseñada nuestra base de datos y las relaciones de las mismas, así como el tipo de relación que tienen estas.





### **2.6- Conclusiones**

En este capítulo se abordó la fase de planeación y diseño donde se delinearon las HU con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas. Además presentando las principales clases mediante el empleo de las tarjetas CRC y se obtuvo el modelo de datos.



### CAPÍTULO III – DESARROLLO Y PRUEBAS

#### 3.1 Introducción

En este capítulo se inicia la fase de desarrollo y pruebas conforme a la metodología XP. Se realiza el desarrollo de las iteraciones a partir del desglose de las historias de usuario en tareas. Así mismo aparecen las interfaces gráficas de usuario diseñadas para la aplicación final.

#### 3.2-Desarrollo de las iteraciones

Durante la fase planificación y diseño fueron detalladas las historias de usuario correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las prioridades y restricciones de tiempo, previstas por el cliente.

Para darle cumplimiento a cada HU, primeramente se debe realizar una revisión del plan de iteraciones, y si es necesario, se le hacen modificaciones a este.

##### 3.2.1-Tareas por historias de usuario

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o ingeniería, y a su vez, estas son asignadas al equipo de desarrollo para su implementación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Las mismas se representan mediante las tarjetas de tareas. (BASULTO, 2010)

Historia de usuario	Tareas
Autenticar usuarios	Insertar los datos para entrar al sistema.
Gestionar usuarios	Insertar usuarios. Eliminar usuarios. Cambiar contraseña. Mostrar listado de usuarios.



Gestionar proyecto	Insertar proyecto. Eliminar proyecto. Mostrar listado de proyecto.
Gestionar aspectos a chequear	Insertar aspectos a chequear Eliminar aspectos a chequear Modificar aspectos a chequear.
Mostrar aspectos a chequear	Mostrar aspectos de estudiantes. Mostrar aspectos de profesores.
Evaluar el proyecto	Evaluar el proyecto.
Gestionar evaluaciones	Mostrar las evaluaciones. Eliminar las evaluaciones
Comparar proyecto	Comparar proyecto
Graficar proyecto	Graficar proyecto
Mostrar resumen valorativo	Mostrar resumen valorativo
Exportar resumen valorativo	Exportar resumen valorativo

Tabla 3.1 Tareas por Historias de usuarios

### 3.2.2-Iteración No.1

En esta iteración se le dio cumplimiento a la implementación a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación, con el fin de obtener una versión del software.

#### Historias de usuario abordadas en la primera iteración.

Historias de usuario	Tiempo de estimación (semanas)
Autenticar usuarios.	1 semana
Gestionar usuarios.	1 semana
Gestionar proyecto.	1 semana
Gestionar aspectos a chequear.	1 semana
Mostrar aspectos a chequear.	1 semana

Tabla 3.2 Iteración 1



Tarea ingeniería	
Número tarea: 1	Número historia: 2
Nombre tarea: Insertar usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 7/2/2011	Fecha fin: 8/2/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea es realizada por el Administrador y la misma facilita insertar a los usuarios que van a trabajar en la misma,	

Tabla 3.3 Tarjeta de tarea No. 1 Insertar usuario.

Para ver las otras Tarjetas de tareas de la primera iteración ir al: [Anexo 3](#)

### 3.2.3-Iteración No.2

Historias de usuario abordadas en la segunda iteración.

Historias de usuario	Tiempo de estimación (semanas)
Evaluar proyecto	2
Gestionar evaluaciones	1
Comparar proyecto	2

Tabla 3.15 Iteración 2

Para ver las Tarjetas de tareas de la segunda iteración ir al: [Anexo 3](#)

### 3.2.4-Iteración No.3

Historias de usuario abordadas en la tercera iteración.

Historias de usuario	Tiempo de estimación (semanas)
Graficar proyecto	2
Mostrar resumen valorativo.	1
Exportar resumen valorativo	2

Tabla 3.20 Iteración 2

Para ver las Tarjetas de tareas de la tercera iteración ir al: [Anexo3](#)



### 3.3- Principales Interfaces de la aplicación

Aquí se verán algunas de las principales interfaces de la aplicación llevadas a cabo en la fase de implementación de la metodología expuesta en el capítulo inicial.

- [Interfaz principal de los profesores](#)
- [Interfaz de evaluación.](#)
- [Interfaz del resultado de la evaluación](#)
- [Interfaz de la comparación de los software.](#)

### 3.4-Pruebas

En la metodología XP las pruebas juegan un papel fundamental, pues esta permite la comprobación continua del código. El desarrollo constante de las pruebas da lugar a que se desarrolle un software con mayor calidad dando una mayor seguridad de lo que se está haciendo. En esta metodología hay dos tipos de testing; las unitarias o desarrollo dirigido por pruebas, desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación, las cuáles son evaluadas luego de culminar una iteración verificando así que se cumplió la funcionalidad requerida por el cliente.

#### 3.4.1-Desarrollo dirigido por pruebas

El desarrollo dirigido por pruebas, se enfoca en la implementación orientada a pruebas. El código debe ser probado paso a paso para lograr un resultado, aunque no con lógica para el negocio, pero si funcional. Algunas personas confunden este término con las llamadas “pruebas de caja blanca” las cuáles se les practican a los métodos u operaciones para medir la funcionalidad del mismo, desde el punto de vista de validez del cliente. Sin embargo, el TDD se aplica antes de comenzar a implementar cada paso de la tarea en desarrollo, asumiendo que la prueba es insatisfactoria desde un inicio. Sólo una vez que se haya cumplido de la forma más sencilla posible la lógica del código a probar se asume como cumplida.



Luego se realiza un proceso conocido como refactorización de código perteneciente a una de las doce prácticas planteadas por la metodología XP, el cual consiste en mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez. (BASULTO, 2010)

### 3.4.2- Pruebas de aceptación

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en la metodología RUP, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas satisfactoriamente. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación.

Luego de ver lo plasmado anteriormente y el autor reunirse con el cliente para su análisis, se decidió llevar a cabo el proceso mediante las pruebas de aceptación. La planilla utilizada para plasmar el contenido de las pruebas de aceptación se muestra a continuación. (COLUMBIÉ, 2010)

<b>Prueba de aceptación</b>
<b>HU:</b> Nombre de la historia de usuario que va a comprobar su funcionamiento.
<b>Nombre:</b> Nombre del caso de prueba.
<b>Descripción:</b> Descripción del propósito de la prueba.
<b>Condiciones de ejecución:</b> Precondiciones para que la prueba se realice.
<b>Entrada/Pasos ejecución:</b> Pasos para probar la funcionalidad.
<b>Resultado:</b> Resultado que se desea de la prueba.
<b>Evaluación de la prueba:</b> ( Aceptada o denegada)

Tabla 3.24 Planilla de prueba de aceptación.



<b>Prueba de aceptación</b>
<b>HU:</b> Autenticar usuarios
<b>Nombre:</b> Prueba para comprobar la autenticación de usuarios. (Nombre de usuario y contraseña).
<b>Descripción:</b> Validación de entrada de los datos de los usuarios.
<b>Condiciones de ejecución:</b> El usuario debe introducir su nombre de usuario y contraseña.
<b>Entrada/Pasos ejecución:</b> El usuario escribe su nombre de usuario y contraseña y luego da clic en el botón Entrar.
<b>Resultado:</b> <ul style="list-style-type: none"><li>• Si el usuario tiene acceso para entrar a la aplicación e inserta sus datos correctamente entrará sin problemas al Sistema.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• Se inserte los datos de un usuario no válido para el Sistema. (Ya sea su nombre de usuario o su contraseña).</li><li>• Se dé clic en el botón Entrar sin llenar todos los campos de texto.</li></ul>
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.25 Prueba para comprobar la autenticación de usuarios.

Para ver las otras Prueba de aceptación ir al: [Anexo 4.](#)



### 3.5-Conclusiones

En este capítulo se llevó a cabo la fase de desarrollo y pruebas, Se realizó el desarrollo de las iteraciones a partir de la distribución de tareas por historias de usuarios y se le practicó las pruebas de aceptación a las mismas para verificar que las funcionalidades de la aplicación estén correctamente implementadas, siendo todas estas aceptadas por el cliente.



## CAPÍTULO IV – ESTUDIO DE FACTIBILIDAD

### **4.1 Introducción**

En la actualidad para un satisfactorio desarrollo de cualquier proyecto se hace imprescindible el estudio de factibilidad para tener en cuenta una estimación de los costos a incurrir logrando así definir si será factible o no desarrollar el mismo. Para el estudio de factibilidad de este proyecto se utilizará la metodología Costo Efectividad (Beneficio), la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación de dos factores en conjunto:

- El costo que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados.
- La efectividad que se entiende como capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo por el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad del cumplimiento del objetivo). (PÉREZ, 2008)

### **4.2 Efectos Económicos.**

- Efectos directos.
- Efectos indirectos.
- Efectos externos.
- Intangibles.

#### **Efectos directos:**

##### **Positivos:**

- Los usuarios con acceso al sistema tendrán la posibilidad de evaluar de forma personalizada cualquier software.



- Se mejora la eficiencia y calidad del proceso de gestión de evaluación de los sitios web.
- Se agilizará el proceso de evaluación en cada departamento.
- Se obtendrá un resumen con todas las evaluaciones realizadas (usuario que evalúa, resultado, aspectos evaluados y clasificación).
- Se facilitará el proceso de llevar todas las evaluaciones a un formato para imprimir.

### **Negativos:**

- Para el uso de esta aplicación implementada en plataforma Web se necesitará que la misma sea ejecutada con el navegador Mozilla Firefox, porque es con el que se trabajó en la elaboración del producto por lo que el diseño está adaptado a este tipo de navegador.

### **Efectos indirectos:**

- Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de venta.

### **Efectos externos:**

- Se obtendrá un producto disponible que le facilitará gran parte del trabajo a los profesores encargados de la gestión para la evaluación de la calidad de sitios web.

### **Intangibles:**

- En la valoración económica siempre hay elementos como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible.

A fin de medir con precisión los efectos, deberán considerarse dos situaciones:



- **Situación sin el producto.**

Para llevar a cabo la gestión de evaluación de la calidad de sitios web se deben tener en cuenta los siguientes pasos:

1. Los profesores deben de realizar la evaluación al software a través de varios aspectos para calcular un resultado y su evaluación correspondiente en el área administrativa donde se desarrolla el proceso.
2. Hacerle llegar esta información a los encargados del proceso.

- **Situación con el producto.**

Para la entrada de los datos al sistema propuesto debemos seguir los siguientes pasos:

1. Cargar la aplicación en el navegador de la computadora.
2. Entrar los datos correspondientes del usuario que hará uso de la aplicación.
3. Realizar la evaluación al software deseado.
4. Realizar una comparación de los software deseados.
5. Mostrar un resumen de las evaluaciones.
6. Exportar el resumen con las evaluaciones.

### ***4.3 Beneficios y costos Intangibles en el proyecto***

#### **Costos**

- Resistencia al cambio.

#### **Beneficios**

- Mejor comodidad para los usuarios.
- Mejora la calidad de evaluación.
- Menor tiempo empleado en el proceso de evaluación.
- Conectividad desde cualquier PC que esté conectada a la red.



#### 4.4 Ficha de costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar una ficha de costo de un producto. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional. (ZALDIVAR, 2010)

##### 4.4.1 Costos en Moneda Libremente Convertible (C.U.C)

Ficha de Costo.	
	Precio(s)
<b>Costos Moneda Libremente Convertible</b>	
<b>Costos Directos</b>	
Compra de equipos de cómputo	0,00
Alquiler de equipos de cómputo	0,00
Compra de licencia de Software	0,00
Depreciación de equipos	25,00
Materiales directos	0,00
<b>Subtotal</b>	25,00
<b>Costos Indirectos</b>	
Formación del personal que elabora el proyecto	0,00
Gastos en llamadas telefónicas	0,00
Gastos para el mantenimiento del centro	0,00
Know How	0,00
Gastos en representación	0,00
<b>Subtotal</b>	0,00
<b>Gastos de Distribución y Venta</b>	
Participación en ferias o exposiciones	0,00
Gastos en transportación	0,00
Compra de materiales de propagandas	0,00
<b>Subtotal</b>	0,00
<b>Total</b>	25,00

Tabla 4.1: Costo en Moneda Libremente Convertible



4.4.2 Costos en moneda nacional (M.N):

Ficha de Costo.	
	Precio(s)
<b>Costos Moneda Nacional</b>	
<b>Costos Directos</b>	
Salario del personal que laborará en el proyecto	100,00
12,5% del total de gastos por salarios se dedica a la seguridad social	0,00
9.09% de salario total, por concepto de vacaciones a acumular	0,00
Gasto por consumo de energía eléctrica	235,5
Gastos en llamadas telefónicas	0,00
Gastos administrativos	0,00
<b>Subtotal</b>	335,5
<b>Costos Indirectos</b>	
Gasto por consumo de energía eléctrica	0,00
Know How	0,00
<b>Subtotal</b>	0,00
<b>Total</b>	335,5

Tabla 4.2: Costo en Moneda Nacional.

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la metodología Costo- Efectividad. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado para resolver la gestión de evaluación de la calidad de sitios web y la variable sería la complejidad de las pruebas que se realizan durante este proceso.

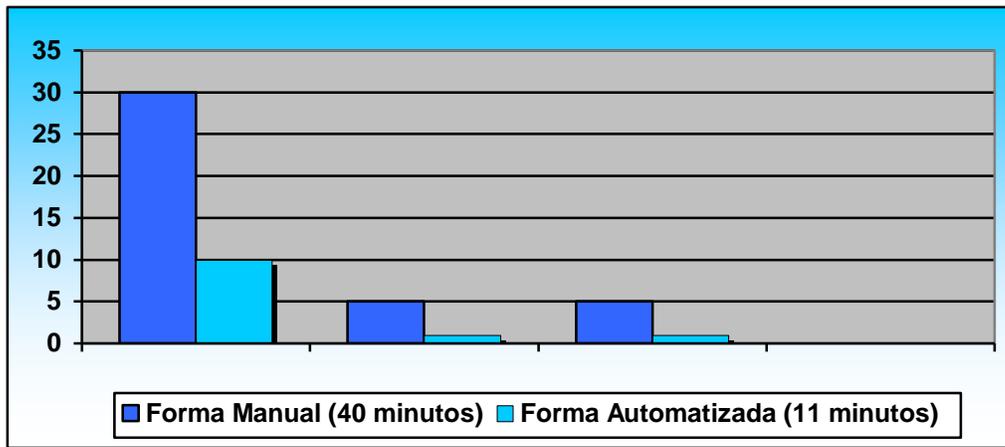


### Valores de la variable (Solución manual):

- Realizar la evaluación a los sitios web a través de aspectos. 3 variables (30 min.).
- Elaborar en resumen de las evaluaciones. 1 variable (5 min.).
- Imprimir resumen de evaluaciones. 1 variable (5 min.).

### Valores de la variable (Solución con el software):

- Cargar cada aspecto con los datos necesarios para la realización de la evaluación. 3 variables (10 min.).
- Cargar formulario con los datos necesarios para mostrar un resumen de todas las evaluaciones. 1 variable (0.033 min.).
- Exportar resumen con las evaluaciones realizadas. 1 variable (0.16 min.).



Teniendo en cuenta los resultados reflejados en la gráfica en cuanto al punto de equilibrio queda demostrada la factibilidad del sistema evidenciado por la relación entre la complejidad del problema (cantidad de variables) y el tiempo que demora la solución del mismo de forma manual y automatizada.



### ***4.5 Conclusiones del Capítulo***

En este capítulo se hace un estudio profundo del costo real en que se incurrió durante el diseño e implementación de la aplicación, software mediante la metodología Costo Efectividad (Beneficios), se analizaron todos los factores directos, indirectos, externos e intangibles, además se calculó el costo de ejecución del producto software mediante la ficha de costo arrojando como resultado \$25,00 C.U.C y \$335,5 M.N, así como el tiempo que demora realizar una evaluación de forma manual y de forma automatizada, demostrándose la factibilidad del proyecto.



### CONCLUSIONES GENERALES:

Hemos llegado a la meta de este trabajo el cual consta de cuatro capítulos donde recogen ampliamente de una forma u otra todo lo realizado en la investigación y el Sistema que está arrojaron las siguientes conclusiones:

- Se desarrolló el producto final que consistía en la automatización de una aplicación informática para la para la evaluación de la calidad de los sitios web.
- Se realizó la planeación y diseño de la aplicación, en los cuales se identificaron y especificaron los requisitos funcionales, así como se llevaron a cabo las posteriores fases de codificación y pruebas, que se definen en la metodología de desarrollo utilizada.
- Se llevó a cabo un estudio de las principales metodologías, lenguajes y herramientas que se consideraron factibles para el desarrollo del sistema.
- Se realizaron las pruebas de aceptación definidas por el usuario, lo que arrojó como resultado su aceptación, con lo cual se demostró el cumplimiento satisfactorio de las historias de usuarios.

Se efectuó un estudio detallado de la factibilidad del producto final, el cual arrojó los resultados esperados.

Como resultado de los objetivos propuestos en la investigación se logró la implementación del producto software que se esperaba. Con el propósito fundamental de simplificar las demoras que se producen en el tratamiento manual de la información, disminuir el grado de errores y para contribuir a elevar la calidad del desarrollo del trabajo.



### RECOMENDACIONES:

Se recomienda lo siguiente:

- Primeramente la utilización del sistema.
- Agregar nuevas funcionalidades acorde a nuevos requisitos que pudiesen surgir por alguna causa o para aumentar el rendimiento del Sistema.
- Realizar un estudio más profundo de este sistema en vista a perfeccionarlo en versiones futuras.
- Extender el software a cualquier entidad de nuestro país.
- Trabajar en la aplicación con navegadores Open-Source ya que traducen los estilos de diseño con mayor calidad.



**REFERENCIA BIBLIOGRÁFICA:**

ALVA OBESO, María Elena. *“Metodología de Medición y Evaluación de la Usabilidad de Sitios Web Educativos”* (Tesis Doctoral). Universidad de Oviedo. España. 2005.

(BASULTO Jorge Mario). *Sistema de Gestión integral de la empresa Empleadora del NIQUEL “EMPLENI”*. MODULO GESTION DE CONTRATOS DE COMPRAS. ISMM. Moa, 2010

(COLUMBIÉ Rafael Wilian). *Evaluación de Calidad de Sitios Web*. ISMM. Moa, 2010.

De Miguel, Enrique. *“Usabilidad el Nuevo Paradigma del Software de Gestión”*.2008.[Consultado2011-02-25]. Disponible en: <http://mejornegocios.com/>

**ER/Studio** [Consultado 2011-03-30]; Disponible en:

<http://www.monografias.com/trabajos14/modelobase/modelobase.shtml>

(HERNÁN RUIZ Marcelo, 2006), *Programación Web Avanzada*, La Habana, Editorial Félix Varela, 2006

Introducción a herramientas CASE y System Architect. Universidad Politécnica de Valencia. . [Consultado: 2011-03-28], 2004. Disponible en: [http://www.dsic.upv.es/assignaturas/eui/mtp/doc-practicas/intro\\_case\\_SA.pdf](http://www.dsic.upv.es/assignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf)

(MONMANY, J). *Aplicaciones web*. [Consultado: 2011-03-20]. Disponible en: <http://www.webvillage.info>

MySQL [Consultado: 2011-03-15].Disponible en <http://es.wikipedia.org/wiki>.



PÉREZ GARCÍA, Dra. Ana María. *Procedimientos para la elaboración de la ficha de costo de un producto informático*. Facultad MFC UCLV. Villa Clara.

Rational Rose Enterprise. [Consultado 2011-03-30]; Disponible en:

<http://www-306.ibm.com/software/awdtools/developer/rose/enenterprise>

(ZALDIVAR Yusimi). *Sistema automatizado de control de acciones correctivas y preventivas de la Empresa Empleadora del Níquel*. ISMM. Moa, 2010



### GLOSARIO DE TERMINOS:

**HTML:** Hyper Text Markup Language, o simplemente HTML, es un lenguaje de etiquetas de programación muy sencillo que se utiliza para crear los textos y las páginas web. Si se hace la traducción de su nombre del inglés al castellano, sería “Lenguaje de Marca de Hipertextos”, ya que es justamente un lenguaje que se basa en las marcas para crear los hipertextos.

**Testing:** Las pruebas de software, en inglés *testing* son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

**Open source:** Código abierto o código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como software libre.

**UML:** (*Unified Model Language*): Es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de una gran parte de la comunidad informática. Constituye una técnica utilizada para el modelado de diferentes tipos de sistemas, para describir una información bien detallada sobre el funcionamiento de los mismos mediante símbolos estándares.



### ANEXOS:

#### Anexo 1: Historias de Usuarios

Historia de usuario	
<b>Número:</b> 2	<b>Usuario:</b> cliente.
<b>Nombre:</b> Gestión de usuarios.	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El Administrador debe de insertar los usuarios (profesores y estudiantes) que van a trabajar con el Sistema. Además eliminarlos, cambiarles la contraseña y mostrar un listado de los mismos. El sistema guarda los datos.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.4 HU No.2: Gestionar usuarios.

Historia de usuario	
<b>Número:</b> 3	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Gestión de proyectos	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El especialista o los estudiantes deben entrar el nombre de los proyectos con los que van a trabajar. Además eliminarlos y mostrar un listado de los mismos. El sistema guarda los datos mismos.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.5 HU No.3: Gestionar Proyectos



<b>Historia de usuario</b>	
<b>Número:</b> 4	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Gestión de aspectos a chequear.	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El especialista debe de insertar los aspectos que vaya a chequear, así como eliminar y modificar. El sistema guarda los datos.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.6 HU No.4: Gestionar aspectos a chequear

<b>Historia de usuario</b>	
<b>Número:</b> 5	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Mostrar aspectos a chequear.	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El especialista y el estudiante podrán mostrar un listado de los aspectos que vayan a chequear. El sistema guarda los datos.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.7 HU No.5: Mostrar aspectos a chequear.



Historia de usuario	
<b>Número:</b> 6	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Evaluación de los proyectos.	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El especialista o estudiante deben de evaluar los proyectos a través de los aspectos a chequear.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.8 HU No.6: Evaluar proyecto

Historia de usuario	
<b>Número:</b> 7	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Gestionar evaluaciones.	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El profesor o estudiante debe de mostrar el listado de las evaluaciones, además el profesor podrá eliminar evaluaciones.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.9 HU No.7 Mostrar listado de evaluaciones.



Historia de usuario	
<b>Número: 8</b>	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Comparación de proyectos	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El especialista y el estudiante podrán comparar los proyectos.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.10 HU No.8: Comparar proyectos.

Historia de usuario	
<b>Número: 9</b>	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Graficar proyectos	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El especialista podrá graficar los proyectos.	
<b>Observaciones:</b> Confirmado.	

Tabla 2.11 HU No.9: Graficar proyectos



<b>Historia de usuario</b>	
<b>Número:</b> 9	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Mostrar resumen valorativo.	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> media
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El profesor o estudiantes podrán mostrar un resumen del software que desee.	
Observaciones: Confirmado.	

Tabla 2.11 H.U No.10: Mostrar resumen valorativo

<b>Historia de usuario</b>	
<b>Número:</b> 9	<b>Usuario:</b> Cliente.
<b>Nombre:</b> Exportar resumen.	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> media
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> El profesor debe de exportar el resumen.	
Observaciones: Confirmado.	

Tabla 2.12 H.U No.11: Exportar resumen



## Anexo 2: Tarjeta CRC

<b>Clase: Gestionar_ usuarios</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Insertar usuarios. Eliminar usuarios. Listar usuarios. Cambiar contraseña.	Conexion.

Tabla 2.17 Tarjeta CRC No.2 Gestionar usuarios.

<b>Clase: Gestionar_ usuarios</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Insertar proyecto Eliminar proyecto. Listar proyecto.	Conexion.

Tabla 2.18 Tarjeta CRC No.3 Gestionar proyecto

<b>Clase: Gestionar_ Aspectos a chequear.</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Insertar aspectos Eliminar aspectos Modificar aspectos	Conexion.

Tabla 2.19 Tarjeta CRC No.4 Gestionar aspectos a chequear.

<b>Clase: Mostrar_ Aspectos a chequear.</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Mostrar aspectos de estudiantes. Mostrar _ aspectos de profesores.	Conexion.

Tabla 2.20 Tarjeta CRC No.5 Mostrar aspectos a chequear.



<b>Clase: Evaluar Proyecto.</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Evaluación de proyecto	Conexion.

Tabla 2.21 Tarjeta CRC No.6 Evaluar proyecto.

<b>Clase: Gestionar Evaluaciones.</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Listar evaluaciones Eliminar evaluaciones	Conexion.

Tabla 2.22 Tarjeta CRC No.7 Gestionar evaluaciones.

<b>Clase: Comparar Proyecto.</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Comparar proyecto	Conexion.

Tabla 2.23 Tarjeta CRC No.8 Comparar proyectos

<b>Clase: Graficar _ proyecto.</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Graficar proyecto	Conexion.

Tabla 2.24 Tarjeta CRC No.9 Graficar proyectos

<b>Clase: Mostrar resumen valorativo.</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Mostrar resumen valorativo	Conexion.

Tabla 2.25 Tarjeta CRC No.10 Mostrar resumen valorativo.

<b>Clase: Exportar Resumen.</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Exportar_resumen	Conexion.

Tabla 2.26 Tarjeta CRC No.11 Exportar resumen.



### Anexo 3: Tarjeta de Ingeniería

#### Iteración 1

Tarea ingeniería	
<b>Número tarea:</b> 2	<b>Número historia:</b> 2
<b>Nombre tarea:</b> Eliminar Usuario.	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 8/2/2011	<b>Fecha fin:</b> 9/2/2011
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> Esta tarea facilita eliminar los usuarios que ya no van a interactuar con el Software.	

Tabla 3.4 Tarjeta de tarea No.2: Eliminar usuario.

Tarea ingeniería	
<b>Número tarea:</b> 3	<b>Número historia:</b> 2
<b>Nombre tarea:</b> Cambiar Contraseña.	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 9/2/2011	<b>Fecha fin:</b> 10/2/2011
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> Esta tarea facilita cambiarles la contraseña a los usuarios sin necesidad de pedirle su contraseña actual, previniendo en caso de olvido u otros casos.	



Tabla 3.5 Tarjeta de tarea No.3: Cambiar contraseña.

<b>Tarea ingeniería</b>	
<b>Número tarea:</b> 4	<b>Número historia:</b> 2
<b>Nombre tarea:</b> Mostrar listado de usuarios.	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 10/2/2011	<b>Fecha fin:</b> 11/2/2011
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> Esta tarea muestra un listado con todos los usuarios que tienen acceso al Sistema son los mismos que los administradores registran.	

Tabla 3.6 Tarjeta de tarea No.4: Mostrar listado de usuarios

<b>Tarea ingeniería</b>	
<b>Número tarea:</b> 5	<b>Número historia:</b> 3
<b>Nombre tarea:</b> Insertar proyecto.	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 14/2/2011	<b>Fecha fin:</b> 15/2/2011
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> Esta tarea inserta los nombres de los proyecto que vayan a ser evaluados.	

Tabla 3.7 Tarjeta de tarea No.5: Insertar proyecto.



Tarea ingeniería	
<b>Número tarea:</b> 6	<b>Número historia:</b> 3
<b>Nombre tarea:</b> Eliminar proyecto.	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 15/2/2011	<b>Fecha fin:</b> 16/2/2011
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> Esta tarea elimina los proyecto con los que no se a trabajar.	

Tabla 3.8 Tarjeta de tarea No.6: Eliminar proyecto.

Tarea ingeniería	
<b>Número tarea:</b> 7	<b>Número historia:</b> 3
<b>Nombre tarea:</b> Mostrar listado de los proyectos.	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 17/2/2011	<b>Fecha fin:</b> 18/2/2011
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> Esta tarea muestra un listado de los proyectos.	



Tabla 3.9 Tarjeta de tarea No.7: Mostrar listado de los proyectos.

Tarea ingeniería	
Número tarea: 8	Número historia: 4
Nombre tarea: Insertar aspectos a chequear.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 21/2/2011	Fecha fin: 21/2/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea inserta los aspectos a chequear.	

Tabla 3.10 Tarjeta de tarea No.8: Insertar aspectos a chequear.

Tarea ingeniería	
Número tarea: 9	Número historia: 4
Nombre tarea: Eliminar los aspectos a chequear.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 22/2/2011	Fecha fin: 22/2/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea elimina los aspectos a chequear.	

Tabla 3.11 Tarjeta de tarea No.9: Eliminar los aspectos a chequear.



Tarea ingeniería	
Número tarea: 10	Número historia: 4
Nombre tarea: Modificar los aspectos a chequear.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 23/2/2011	Fecha fin: 23/2/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea modifica los aspectos a chequear.	

Tabla 3.12 Tarjeta de tarea No.10: Modificar los aspectos a chequear.

Tarea ingeniería	
Número tarea: 11	Número historia: 5
Nombre tarea: Mostrar aspectos a chequear de estudiantes	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 28/2/2011	Fecha fin: 28/2/2011
Programador responsable: Humberto Torres Cabrera	
Descripción: Esta tarea muestra los aspectos a chequear de los estudiantes.	

Tabla 3.13 Tarjeta de tarea No.11: Mostrar aspectos a chequear de estudiantes.



Tarea ingeniería	
Número tarea: 12	Número historia: 5
Nombre tarea: Mostrar aspectos a chequear de los profesores.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 1/3/2011	Fecha fin: 1/3/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea muestra los aspectos a chequear de los profesores.	

Tabla 3.14 Tarjeta de tarea No.12: Mostrar aspectos a chequear de los profesores.

## Iteración 2

Tarea ingeniería	
Número tarea: 13	Número historia: 6
Nombre tarea: Evaluar proyecto.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 7/3/2011	Fecha fin: 18/3/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea evalúa los proyectos.	

Tabla 3.16 Tarjeta de tarea No.14: Evaluar proyecto



Tarea ingeniería	
Número tarea: 14	Número historia: 7
Nombre tarea: Eliminar evaluaciones	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 21/3/2011	Fecha fin: 21/3/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea elimina las evaluaciones.	

Tabla 3.17 Tarjeta de tarea No.15: Eliminar evaluaciones.

Tarea ingeniería	
Número tarea: 15	Número historia: 7
Nombre tarea: Mostrar evaluaciones.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 22/3/2011	Fecha fin: 23/3/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea muestra el listado de evaluaciones.	

Tabla 3.18 Tarjeta de tarea No.16: Mostrar evaluaciones.



Tarea ingeniería	
Número tarea: 16	Número historia: 8
Nombre tarea: Comparar proyecto.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 28/3/2011	Fecha fin: 8/4/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea compara el proyecto.	

Tabla 3.19 Tarjeta de tarea No.17: Comparar proyecto.

### Iteración 3

Tarea ingeniería	
Número tarea: 17	Número historia: 9
Nombre tarea: Graficar proyecto.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 25/4/2011	Fecha fin: 6/5/2011
Programador responsable: Humberto Torres Cabrera.	
Descripción: Esta tarea grafica el proyecto.	

Tabla 3.21 Tarjeta de tarea No.19: Graficar proyecto.



Tarea ingeniería	
<b>Número tarea:</b> 18	<b>Número historia:</b> 10
<b>Nombre tarea:</b> Mostrar resumen valorativo.	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 16/5/2011	<b>Fecha fin:</b> 18/5/2011
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> Esta tarea muestra un resumen valorativo	

Tabla 3.22 Tarjeta de tarea No.20: Mostrar resumen valorativo.

Tarea ingeniería	
<b>Número tarea:</b> 19	<b>Número historia:</b> 11
<b>Nombre tarea:</b> Exportar resumen.	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 30/5/2011	<b>Fecha fin:</b> 10/6/2011
<b>Programador responsable:</b> Humberto Torres Cabrera.	
<b>Descripción:</b> Esta tarea exporta un resumen valorativo	

Tabla 3.23 Tarjeta de tarea No.21: Mostrar resumen valorativo.



## Anexo 4: Pruebas de Aceptación

<b>Prueba de aceptación</b>
<b>HU:</b> Gestión de usuarios
<b>Nombre:</b> Prueba para verificar la gestión de usuarios.
<b>Descripción:</b> Validación de la gestión de usuarios.
<b>Condiciones de ejecución:</b> El usuario debe de entrar a la aplicación y el mismo tiene que ser administrador de esta para poder insertar usuarios, eliminarlos, cambiarles su contraseña sin necesidad de pedir su contraseña actual, así como listar todos los usuarios con acceso al Sitio.
<b>Entrada/Pasos ejecución:</b> El administrador escribe todos los datos que se le piden para poder insertar un usuario determinado asignándole el tipo de usuario, luego presiona el botón Insertar. Después que el usuario este insertado es que se le puede cambiar su contraseña, eliminarlo y ver sus datos en el listado de usuarios.
<b>Resultado:</b> <ul style="list-style-type: none"><li>• Si se insertó el usuario correctamente muestra un mensaje de confirmación.</li><li>• Cuando se cambia la contraseña se emite otro mensaje de confirmación.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• Falten datos del usuario a la hora de insertarlo.</li><li>• Se inserte un usuario que ya exista, es decir que tenga ese nombre.</li><li>• A la hora de confirmar la contraseña no coincida con la que se puso anteriormente.</li><li>• Cuando se va a eliminar el usuario administrador, ya que este está definido por el Sistema.</li><li>• Se dé clic en botón eliminar sin antes seleccionar ningún usuario.</li><li>• Cuando se le va a cambiar la contraseña a un usuario que no exista o se escriba incorrectamente.</li></ul>



**Evaluación de la prueba:** Aceptada.

Tabla 3.26 Prueba para comprobar la Gestión de Usuarios.

<b>Prueba de aceptación</b>
<b>HU:</b> Gestionar Proyectos
<b>Nombre:</b> Prueba para verificar la gestión de proyectos
<b>Descripción:</b> Validación de gestión de proyectos.
<b>Condiciones de ejecución:</b> El usuario debe de entrar a la aplicación y el mismo tiene que ser profesor de esta para poder insertar proyectos, eliminarlos, y tanto el cómo los estudiantes podrán listar todos los proyectos que han sido insertados
<b>Entrada/Pasos ejecución:</b> El profesor escribe todos los datos que se le piden para poder insertar un nuevo software determinado, luego presiona el botón Insertar. Después que el software este insertado es que se le puede eliminar y ver sus datos en el listado de software.
<b>Resultado:</b> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>● Falten datos del software a la hora de insertarlo.</li><li>● Se dé clic en el botón insertar sin llenar los campos de texto.</li><li>● Si se entra el nombre de un software que ya este insertado.</li><li>● Se dé clic en botón eliminar sin antes seleccionar ningún software.</li></ul>
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.27 Prueba para comprobar la gestión de proyectos.

<b>Prueba de aceptación</b>
<b>HU:</b> Gestionar aspectos a chequear
<b>Nombre:</b> Prueba para verificar la gestión de aspectos a chequear
<b>Descripción:</b> Validación de gestión de aspectos a chequear.
<b>Condiciones de ejecución:</b> El usuario debe de entrar a la aplicación y el



<p>mismo tiene que ser profesor de esta para poder insertar un nuevo aspecto, eliminarlo, y modificar los aspectos a chequear que se requiera.</p>
<p><b>Entrada/Pasos ejecución:</b> El profesor escribe todos los datos que se le piden para poder insertar un nuevo aspectos, luego presiona el botón Insertar. Después que el aspecto este insertado es que se le puede eliminar y modificar.</p>
<p><b>Resultado:</b></p> <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none"><li>• Falten datos del nuevo aspecto a la hora de insertarlo.</li><li>• Se dé clic en el botón insertar sin llenar los campos de texto.</li><li>• Se dé clic en botón eliminar sin antes seleccionar ningún aspecto.</li><li>• Se dé clic en botón modificar sin antes seleccionar ningún aspecto.</li></ul>
<p><b>Evaluación de la prueba:</b> Aceptada.</p>

Tabla 3.28 Prueba para comprobar Aspectos a Chequear.

<b>Prueba de aceptación</b>
<b>HU:</b> Mostrar aspecto a chequear
<b>Nombre:</b> Prueba para verificar si se mostró los aspecto a chequear
<b>Descripción:</b> Validación de mostración de los aspectos a chequear.
<b>Condiciones de ejecución:</b> El usuario debe de entrar a la aplicación sin importar el tipo de usuario que sea, podrá ver el listado de los aspectos a chequear.
<b>Entrada/Pasos ejecución:</b> El usuario muestra el listado aspectos con sus preguntas incluidas, dependiendo el tipo de usuario que sea.
<b>Resultado:</b> Se muestran el listado de los aspectos según el tipo de usuario.
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.29 Prueba para mostrar aspectos a chequear.



<b>Prueba de aceptación</b>
<b>HU:</b> Evaluar proyectos.
<b>Nombre:</b> Prueba para verificar la evaluación de los proyectos.
<b>Descripción:</b> Validación de la evaluación de los proyectos.
<b>Condiciones de ejecución:</b> El usuario tiene que ser profesor o estudiante para poder proceder a la evaluación del software.
<b>Entrada/Pasos ejecución:</b> Los profesores o estudiantes una vez dentro de la aplicación se dirigen al menú de navegación, a la pestaña Evaluar, donde dice evaluar proyecto, ahí se muestran los aspectos, seleccionas lo que vayas a evaluar y una vez seleccionado se muestra los aspectos en forma de preguntas, le damos la evaluación que le corresponda y apretamos el botón evaluar.
<b>Resultado:</b> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• No selecciones menos de 5 aspectos a evaluar.</li><li>• .falten evaluar todos los aspectos que fueron seleccionados.</li><li>• No seleccioné el nombre del software a evaluar.</li></ul>
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.30 Prueba para comprobar la evaluación del proyecto.

<b>Prueba de aceptación</b>
<b>HU:</b> Gestionar Evaluaciones
<b>Nombre:</b> Prueba para verificar la gestión de las evaluaciones
<b>Descripción:</b> Validación de gestión de proyectos Evaluaciones.
<b>Condiciones de ejecución:</b> El usuario debe de entrar a la aplicación y el mismo tiene que ser profesor de esta para poder eliminar evaluaciones, y tanto el cómo los estudiantes pueden mostrar un listado de evaluaciones eliminarlos, y listar todos los proyectos que han sido insertados
<b>Entrada/Pasos ejecución:</b> El profesor escribe todos los datos que se le piden



para poder eliminar una evaluación determinada, luego presiona el botón eliminar. También podrá ver los datos en el listado de evaluaciones.
<b>Resultado:</b> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• Se dé clic en botón eliminar sin antes seleccionar ninguna evaluación.</li></ul>
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.31 Prueba para comprobar la gestión de evaluaciones.

<b>Prueba de aceptación</b>
<b>HU:</b> Comparar proyecto
<b>Nombre:</b> Prueba para verificar la comprobación de los proyectos
<b>Descripción:</b> Validación de comparar proyecto.
<b>Condiciones de ejecución:</b> El usuario debe de entrar a la aplicación y el mismo tiene que ser profesor o estudiante de esta para poder compara proyectos, tendrán que elegir los software para la realización de la evaluación.
<b>Entrada/Pasos ejecución:</b> El profesor o estudiantes tienen que elegir los dos software que quieran comparar y presionar el botón comparar.
<b>Resultado:</b> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• Falten seleccionar los dos software a la hora de compararlo.</li><li>• Falten seleccionar uno de los software a la hora de compararlo.</li><li>• Se seleccionen dos software iguales para la comparación.</li></ul>
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.32 Prueba para comprobar la comparación de proyectos.



<b>Prueba de aceptación</b>
<b>HU:</b> Graficar proyectos.
<b>Nombre:</b> Prueba para verificar la gráfica de los proyectos.
<b>Descripción:</b> Validación de la gráfica de los proyectos.
<b>Condiciones de ejecución:</b> El usuario podrá ver la gráfica del proyecto, una vez que realice la comprobación.
<b>Entrada/Pasos ejecución:</b> Los profesores o estudiantes una vez dentro de la aplicación se dirigen al menú de navegación, a la comparar, después de elegido los software que va a comparar se le mostrara la grafica de estos.
<b>Resultado:</b> Se muestran la grafica de los software elegidos para la comprobación.
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.33 Prueba para comprobar la graficación del proyecto.

<b>Prueba de aceptación</b>
<b>HU:</b> Mostrar resumen valorativo.
<b>Nombre:</b> Prueba para verificar si se mostró el resumen.
<b>Descripción:</b> Validación de la mostración del resumen valorativo.
<b>Condiciones de ejecución:</b> El usuario podrá mostrar un resumen valorativo del software que desee.
<b>Entrada/Pasos ejecución:</b> Los usuarios una vez dentro de la aplicación se dirigen al menú de navegación, a la pestaña resumen, y damos clic, ahí para mostrar el resumen del software deseado.
<b>Resultado:</b> <ul style="list-style-type: none"><li>• Se muestra un resumen detallado de las evaluaciones de los software que se han evaluados.</li></ul>
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.34 Prueba para comprobar la mostración del resumen de evaluaciones



<b>Prueba de aceptación</b>
<b>HU:</b> Exportar Resumen
<b>Nombre:</b> Prueba para verificar si se exportó el resumen
<b>Descripción:</b> Validación de la exportación del resumen
<b>Condiciones de ejecución:</b> El usuario tiene que tener privilegios de profesor, para poder exportar el resumen de evaluaciones.
<b>Entrada/Pasos ejecución:</b> Los Profesores una vez dentro de la aplicación se dirigen al menú de navegación, a la pestaña exportar resumen, y damos clic. Ahí para exportar el resumen de todas las evaluaciones realizadas.
<b>Resultado:</b> <ul style="list-style-type: none"><li>• Se exportó un resumen detallado con todas las evaluaciones que se han hecho y aún están guardadas en la Base de Datos.</li></ul>
<b>Evaluación de la prueba:</b> Aceptada.

Tabla 3.35 Prueba para comprobar la exportación del resumen de evaluaciones



### Anexo 5: Principales interfaces del sistema.

#### Interfaz principal

**SECSWEB**  
Sistema de Evaluación para la Calidad de Sitios Web

Gestionar Proyectos | Gestionar Aspectos a Chequear | Gestionar Evaluaciones | Comparar Proyecto | Ayuda | Finalizar sesión

#### Bienvenidos

La calidad es un activo estratégico clave, del que dependen la mayor parte de las organizaciones, no sólo para mantener su posición en el mercado sino incluso para asegurar su supervivencia. El desarrollo de un producto que satisfaga, en la mayor medida posible, los requerimientos del cliente, es la medida de calidad buscada en la producción. Una especificación y evaluación integral y detallada de la calidad de los productos de software es un factor clave para asegurar que la calidad sea la adecuada. Esto se puede lograr definiendo de manera apropiada las características de calidad, teniendo en cuenta el propósito del uso del producto de software en la institución. Esta Aplicación Web puede llevarse a cabo por los especialistas en la temática o cualquier otra persona interesada en esta actividad. Es importante especificar y evaluar cada característica relevante de la calidad de los productos de software.

**Calidad del Software**

Se define como la ausencia de errores de funcionamiento, la adecuación a las necesidades del usuario, y el alcance de un desempeño apropiado (tiempo, volumen, espacio), además del cumplimiento de los estándares. Los objetivos que la calidad persigue son: La aceptación (utilización real por parte del usuario) y la Mantenibilidad (posibilidad y facilidad de corrección, ajuste y modificación durante largo tiempo).

**La gestión de software**

Es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos; garantizando la calidad del producto.

#### Interfaz de evaluación.

**SECSWEB**  
Sistema de Evaluación para la Calidad de Sitios Web

Gestionar Proyectos | Gestionar Aspectos | Gestionar Evaluaciones | Comparar Proyectos | Resumen | Ayuda | Finalizar sesión

Nombre del software: SIGESE

#### Evaluación de los aspectos

¿El software es capaz de mostrar de forma clara y sencilla su objetivo?	4
¿El texto es comprensible?	5
¿Se resaltan los títulos y sub-títulos de los textos?	2
¿Los textos han sido redactados de forma afirmativa?	4
¿Se hace uso de las animaciones para ayudar al enriquecimiento del contenido?	3
¿Se utilizan varias formas de representación de la información?	1
¿Se utilizan colores estándares para los vínculos visitados y no visitados?	

Evaluar



## ANEXO 5: PRINCIPALES INTERFACES DEL SISTEMA

### Interfaz del resultado de la evaluación

SECSWEB
Sistema de Evaluación para la Calidad de Sitios Web

Gestionar Proyectos
Gestionar Aspectos
Gestionar Evaluaciones
Comparar Proyectos
Resumen
Ayuda
Finalizar sesión

Evaluación del Software: SIGESE

**Puntos de la Evaluación**

Puntuaciones	Aspectos chequeados	Total de puntos
2	Claridez del software	
5	Comprensión del texto	
4	Fuentes de los textos	
3	Redacción de los textos	
4	Uso de animaciones	
4	Formas de representación de la información	
5	Uso de los colores	
Total	7	27

**Criterio de la Evaluación**

Aspectos entre 4-5	Clasificación
5	Buena Usabilidad

Inicio  
Instituto Superior Minero Metalurgico

### Interfaz de la comparación de los software.

SECSWEB
Sistema de Evaluación para la Calidad de Sitios Web

Gestionar Proyectos
Gestionar Aspectos
Gestionar Evaluaciones
Comparar Proyectos
Resumen
Ayuda
Finalizar sesión

Software 1:

Software 2:

**Software: BPA**

Evaluadores:

- 1- Red de la Calidad + Suarez
- 2- Red de la Calidad + Suarez
- 3- Encaso uyva Gonzalez
- 4- Red de la Calidad + Suarez

**Software: seccswb**

Evaluadores:

- 1- Red de la Calidad + Suarez
- 2- Red de la Calidad + Suarez

**Características de la evaluación: Software BPA**

Nombre del evaluador	Cantidad de aspectos evaluados	Cantidad de 4-5	Puntuación total	Clasificación
Red de la Calidad + Suarez	5	2	10	Uso de
Red de la Calidad + Suarez	5	3	22	Buena Usabilidad
Encaso uyva Gonzalez	9	9	27	Buena Usabilidad
Red de la Calidad + Suarez	14	9	42	Problema Craso

**Características de la evaluación: Software seccswb**

Nombre del evaluador	Cantidad de aspectos evaluados	Cantidad de 4-5	Puntuación total	Clasificación
Red de la Calidad + Suarez	5	2	10	Problema de Usabilidad
Red de la Calidad + Suarez	7	4	28	Uso de

**Evaluación final:**

Según la evaluación realizada y la cantidad de evaluadores que realizaron cada software, BPA es más exitosa que seccswb en la evaluación de la calidad.