

Universidad de las Ciencias Informáticas



Facultad 2

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS**

**Sistema Evaluador Basado en el Procedimiento de Evaluación de la
Usabilidad en los Software de Gestión sobre Plataforma Web en la
Facultad 2.**

AUTORES:

Mislaidy Lastre Lago

Ángel Caboverde Pelier

TUTORES:

Ing. Halena Yayset Leyva

Ing. Yunier Alexander Pimienta

Ciudad de La Habana, Junio del 2010

Agradecimientos

Declaramos que Mislaidy Lastre Iago y Angel Caboverde Pelier somos los únicos autores de este trabajo y autorizamos a la Facultad 2 de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Mislaidy Lastre Lago

Firma del Autor

Angel Caboverde Pelier

Firma del Autor

Ing. Halena Yayset Leyva

Firma de la Autora

Ing. Yunier Alexander Pimienta

Firma del Co-Tutor

INDICE

INTRODUCCIÓN.....	I
CAPÍTULO 1. Fundamentación Teórica	5
Introducción.....	5
1.1 Conceptos Fundamentales	5
1.1.1 Usabilidad	5
1.1.2 Diseño Centrado en el Usuario	6
1.1.3 Factores Relevantes de la Usabilidad.....	7
1.1.4 Interacción Persona – Ordenador	8
1.1.5 Experiencias del Usuario.....	8
1.1.6 Diseño de Interacción	8
1.1.7 Accesibilidad	9
1.1.8 Calidad de Software.....	9
1.1.9 Ingeniería de la Usabilidad.....	10
1.1.10 Diseño de Interfaz de Usuario.....	11
1.2 Evaluación de la Usabilidad.....	11
1.2.1 Principios de evaluación de la usabilidad.....	12
1.2.2 Técnicas de evaluación de Usabilidad	13
1.3 Tendencias actuales de la Usabilidad	16
1.4 Sistemas de Evaluación de Usabilidad Existentes	17
1.5 Lenguajes de programación Web	19
1.5.1 Java	19

1.5.2	PERL.....	19
1.5.3	PHP.....	19
1.6	Framework.....	21
1.6.1	Seagull	21
1.6.2	Prado	21
1.6.3	Zoop.....	21
1.6.4	WACT.....	21
1.6.5	Symfony	22
1.7	Gestor de Base de Datos	23
1.7.1	Firebird.....	23
1.7.2	PostgreSQL.....	24
1.7.3	MySQL	25
1.8	Servidor Web.....	26
1.8.1	Lighttpd	26
1.8.2	Apache.....	26
1.9	Metodologías de desarrollo de Software	27
1.10	Herramienta CASE	30
1.10.1	Rational Rose.....	30
1.10.2	Visual Paradigm	31
	Conclusiones.....	32
	CAPÍTULO 2. Características del Sistema (SEUSGOW)	33
	Introducción.....	33

2.1	Proceso a Automatizar	33
2.1.1	Lista de Chequeo por Especialista.....	37
2.2	Modelo de Negocio.....	38
2.2.1	Entidades del Negocio	38
2.2.2	Actores del Negocio	38
2.2.3	Trabajadores de Negocio	38
2.2.4	Casos de uso del Negocio	40
2.2.5	Descripción de los Principales Casos de Uso del Negocio	40
2.2.6	Diagrama de Casos de Uso del Negocio	43
2.2.7	Modelo Objeto.....	43
2.2.8	Reglas del Negocio	43
2.3	Modelo de Sistema.....	44
2.3.1	Especificación de Requisitos del Software.....	44
2.3.2	Actores del Sistema	50
2.3.3	Diagrama de caso de Uso del Sistema	51
2.3.4	Descripción de Casos de uso del Sistema.....	51
2.3.4.1	Descripción del CU Gestionar Usuario.....	52
2.3.4.2	Descripción del CU Gestionar Evaluación Objetivos y Lenguaje	54
	Conclusiones.....	57
	CAPÍTULO 3. Análisis y Diseño del Sistema (SEUSGOW).....	58
3.1	Modelo de análisis.....	58
3.2	Diagramas de clases del análisis.....	58

3.3	Diagramas de Interacción	60
3.3.1	Diagrama de Colaboración.....	60
3.4	Modelo Diseño.....	61
3.5	Diagrama de clases del diseño.....	62
3.6	Descripción de la Arquitectura	63
3.7	Modelo Lógico de datos.....	64
3.8	Modelo Físico de datos.....	65
	Conclusiones	65
CAPÍTULO 4. Implementación.....		66
	Introducción	66
4.1	Modelo de Despliegue.	66
4.2	Diagrama de componentes.....	66
	Conclusiones	68
CAPÍTULO 5. Estudio de Factibilidad		69
	Introducción	69
	Método de Estimación Puntos por Casos de Uso	69
	Conclusiones	79
Conclusiones Generales.....		80
Recomendaciones		81
Bibliografía Referenciada.....		82

RESUMEN

La usabilidad es un factor determinante en la Calidad de un Software, sobre todo cuando es el elemento que más influye en la percepción que tienen los usuarios del mismo.

Actualmente la Infraestructura Productiva de la Universidad de las Ciencias Informáticas no cuenta con un sistema que les permita realizar una evaluación de usabilidad en los software de gestión sobre tecnología web. Por lo antes expuesto este trabajo tiene como objetivo principal realizar el análisis, diseño e implementación de dicho sistema específicamente en la Facultad 2.

Su realización estuvo motivada por la necesidad de conocer en qué medida los Software de Gestión sobre Plataforma Web que se desarrollan pueden ser usados, aprendidos y por tanto aceptados de forma fácil por los usuarios.

Este trabajo tiene como base un procedimiento realizado con anterioridad en la universidad que tuvo como recomendación fundamental el desarrollo de una herramienta que hiciera la aplicación del mismo más simple. Para el desarrollo de la misma se hace un estudio de las principales metodologías, técnicas, lenguajes, y herramientas existentes utilizadas para la creación de sistemas Web, para poder realizar una adecuada selección de las que van a ser usadas en este trabajo. Posteriormente se realiza el análisis y diseño de la aplicación, concluyéndose con la implementación de la misma y sus pruebas correspondientes garantizándose la calidad del producto.

Palabras Claves: usabilidad, sistema, evaluación, calidad, heurística.

INTRODUCCIÓN

La expansión del uso de ordenadores a una gran parte de la sociedad, el desarrollo de las TIC (Tecnología de la Información y la Comunicación) y la gran demanda de aplicaciones web seguras, novedosas, fiables y cada vez más sencillas desde el punto de vista interactivo ha hecho que la usabilidad del software adquiera una gran importancia en el desarrollo de sistemas informáticos.

La usabilidad surge desde épocas tempranas en la historia de las computadoras. La misma estudia la interacción usuario – software. Busca que los usuarios se sientan cómodos al usar un software determinado, que sus opciones sean fáciles de recordar, que sea entendible, que su manipulación sea muy intuitiva y que el trabajo en este no se vuelva difícil. Si el software es capaz de atraer al usuario entonces tiene calidad. Su reto es entender cómo los usuarios ven el software.

La usabilidad está recibiendo un creciente interés en el mundo del desarrollo de software, como factor de calidad determinante. Las organizaciones contratantes de proyectos de desarrollo de software están empezando a incluir requisitos de usabilidad en las especificaciones de requisitos de software, ya que se han dado cuenta de la importancia que se le está dando en la actualidad a desarrollar productos “usables” y de esta forma atraer mayor cantidad de usuarios. Por otra parte, en los desarrollos dirigidos al mercado, la competitividad actual obliga a mejorar el nivel de usabilidad, porque existe un riesgo real de perder clientes si la aplicación no es fácil de usar y/o fácil de aprender.

Evaluar la usabilidad de un software constituye sólo una parte de la Ingeniería de la Usabilidad que además incluye aplicar de manera sistemática y práctica métodos apropiados de recogida de requisitos, desarrollo y evaluación de prototipos, evaluación de alternativas, definición y evaluación de objetivos de usabilidad, analizar los usuarios finales e implicarlos tanto en la evaluación de los avances producidos como con todas las fases en las que el proyecto lo requiera y constituir además equipos multidisciplinares en los que el peso de las responsabilidades parciales recae en la disciplina más adecuada.

La Universidad de Ciencias Informáticas (UCI) está asociada directamente a la producción de software y esto implica que todo lo que se produzca debe tener la máxima calidad para así lograr cumplir uno de los objetivos por los cuales se construyó la UCI; aportar a la economía del país.

En la UCI se desarrolló un procedimiento para evaluar la usabilidad de un software sobre plataforma web. En este se utilizan métodos de recopilación de información como la entrevista, la encuesta y la revisión de documentos, donde se profundiza en la utilización en proyectos productivos de la usabilidad, así como del conocimiento teórico y práctico en torno a ella por parte de las personas relacionadas a la producción de software en la universidad. Este procedimiento a grandes rasgos propone la evaluación a través de un panel de expertos y listas de chequeo.

La ejecución de este procedimiento de forma manual, trae consigo disímiles consecuencias entre las que se puede encontrar: la pérdida de información por estar en formato duro; la seguridad también se puede comprometer ya que personas no autorizadas pueden acceder; se hace más difícil la obtención de reportes estadísticos que ayuden a la toma de decisiones importantes; la disponibilidad de la información no es la más óptima; la organización y la búsqueda de la información en formato duro lleva consigo un trabajo adicional, difícil y extenuante; además de existir un desgaste físico por parte de las personas que realizan esta evaluación. Por todas estas razones y las muchas facilidades y potencialidades que brindan la digitalización de los procesos es que se decide automatizar el mismo.

Teniendo en cuenta lo antes expuesto, el **problema científico** queda formulado de la siguiente manera: ¿Cómo mejorar el procedimiento de evaluación de usabilidad en los software de gestión de la facultad 2?

Para dar respuesta a lo planteado anteriormente se plantea como **objeto de estudio**: La Usabilidad en los Software de Gestión y se enmarca en el **campo de acción**: La Usabilidad en los Software de Gestión sobre Plataforma Web de la Facultad 2.

Como **objetivo general** se propone: Desarrollar un Sistema Evaluador basado en el Procedimiento de Evaluación de la Usabilidad en los Software de Gestión sobre Plataforma Web de la Facultad 2.

Se derivan los **objetivos específicos**:

1. Estudio del estado del arte de sistemas evaluadores de usabilidad.
2. Estudio de los escenarios a automatizar del procedimiento de evaluación de usabilidad propuesto.
3. Desarrollo de la aplicación para evaluar la usabilidad en los software de gestión.

Para dar cumplimiento a los objetivos que se han planteado se proponen las siguientes **tareas**:

1. Investigar sobre el desarrollo de procedimientos para evaluar la usabilidad de un software sobre plataforma Web.
2. Estudiar y analizar el Procedimiento basado en la Usabilidad de los Software de Gestión sobre Plataforma Web propuesto por la facultad 2.
3. Estudiar y valorar las herramientas existentes acerca de la evaluación de la usabilidad sobre plataforma Web.
4. Definir análisis del sistema para evaluar la usabilidad de un software sobre plataforma Web.
5. Definir el diseño del sistema para evaluar la usabilidad de un software sobre plataforma Web.
6. Implementar el sistema para evaluar la usabilidad de un software sobre plataforma Web.

Para lograr un mejor desarrollo de esta investigación se emplearon los siguientes **métodos científicos**:

Métodos teóricos:

- Método analítico – sintético: Se utilizó este método ya que se va a analizar como tal el procedimiento basado en la evaluación de la Usabilidad de los Software de Gestión sobre Plataforma Web y sintetizar dicha información.
- Método histórico - lógico: Este método se utilizó cuando se hizo el estudio del comportamiento del tema de la usabilidad a lo largo de los años.

El documento está organizado en cuatro capítulos, a continuación se brinda una breve descripción:

Capítulo 1: “**Fundamentación Teórica**”. Se ilustran conceptos relacionados con la usabilidad y la evaluación de la misma así como las técnicas que existen para dicha evaluación. Brinda información sobre sistemas de evaluación de la usabilidad en el ámbito internacional. Se hace énfasis en el estudio de las tecnologías, lenguajes de programación, herramientas y metodologías seleccionadas para el desarrollo del sistema.

Capítulo 2: “**Características del Sistema**”. Se presentan los requisitos funcionales y no funcionales con los que debe cumplir el sistema propuesto, las reglas del negocio así como sus actores y diagrama de casos de uso con la descripción expandida de cada uno de ellos.

Capítulo 3: “**Análisis y Diseño del Sistema**”. Se muestran los diagramas de clases del análisis y del diseño para cada caso de uso del sistema, junto con los correspondientes diagramas de interacción, además del diseño de la base de datos y otras restricciones del diseño.

Capítulo 4: “**Implementación del Sistema**”. Se muestra el modelo de implementación además del diagrama de despliegue. Se especifican las pautas usadas para la implementación del sistema así como las principales funcionalidades de la aplicación.

Capítulo 5: “**Estudio de Factibilidad**”. Se presenta una estimación en términos de esfuerzo que permite predecir el tamaño de un sistema a partir de las características de los requerimientos expresados en casos de uso, utilizando el método de estimación “Análisis de puntos de casos de uso”.

CAPÍTULO 1. Fundamentación Teórica

Introducción

En el presente capítulo se abordan conceptos que están relacionados con la evaluación de la usabilidad, se realiza una investigación sobre sistemas de evaluación de la usabilidad en el ámbito nacional e internacional. Se hace énfasis en el estudio de las tecnologías, lenguajes de programación, herramientas y metodologías seleccionadas para el desarrollo del sistema, en función de las tendencias actuales.

1.1 Conceptos Fundamentales

1.1.1 Usabilidad

La usabilidad se define como la medida en que un producto puede ser usado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado. [1]

- **Efectividad:** Es la precisión y la plenitud con las que los usuarios alcanzan los objetivos especificados.
- **Eficiencia:** Los recursos empleados en relación con la precisión y plenitud con que los usuarios alcanzan los objetivos especificados.
- **Satisfacción:** La ausencia de incomodidad y la actitud positiva en el uso del producto.

La usabilidad, hace referencia, a la rapidez y facilidad con que las personas llevan cabo sus tareas propias a través del uso del producto objeto de interés, idea que se refleja en estos aspectos:

- **Aproximación al usuario:** usabilidad significa enfocarse en los usuarios. Para desarrollar un producto usable, se tienen que conocer, entender y trabajar con las personas que representan a los usuarios actuales o potenciales del producto.
- **Amplio conocimiento del contexto de uso:** Las personas utilizan los productos para incrementar su propia productividad. Un producto se considera fácil de aprender y usar en términos del tiempo

que toma el usuario para llevar a cabo su objetivo, el número de pasos que tiene que realizar para ello, y el éxito que tiene en predecir la acción apropiada para llevar a cabo. Para desarrollar productos usables hay que entender los objetivos del usuario, hay que conocer los trabajos y tareas del usuario que el producto automatiza, modifica o embellece.

- **El producto ha de satisfacer las necesidades del usuario:** Los usuarios son gente ocupada intentando llevar a cabo una tarea. Se va a relacionar usabilidad con productividad y calidad. El hardware y el software son las herramientas que ayudan a la gente ocupada a realizar su trabajo y a disfrutar de este.

En la actualidad la usabilidad a nivel mundial es una exigencia en todo software, pues los estudios han demostrado que el aumento de los visitantes y de la aprobación del software por los usuarios aumenta cuando tienen aplicados estudios sobre la temática, principalmente en los software sobre plataforma web.

En la UCI la usabilidad es un tema poco usado y la mayoría de los software que se crean no se les aplica correctamente este proceso para lograr mejorarla, incluso es un tema que algunos de los participantes en los proyectos productivos desconocen.

Los proyectos productivos que aplican algunos de estos conocimientos son los que cuentan en su equipo de desarrollo con arquitectos de información, rol que se ocupa de la organización del entorno de información dentro del software y para lo cual aplica técnicas y herramientas del diseño centrado en el usuario.

La usabilidad es inminentemente necesaria en el desarrollo del software sobre plataforma web ya que es quien hace que éste se use en mayor o menor proporción por los usuarios finales.

1.1.2 Diseño Centrado en el Usuario

El diseño centrado en el usuario se caracteriza por asumir que todo el proceso de diseño y desarrollo del software debe estar conducido por el usuario, sus necesidades, características y objetivos.

Los usuarios son los que van a trabajar e intercambiar a tiempo completo con los software y son ellos los que verán de una forma más cercana las deficiencias y las mejoras del producto, por eso es fundamental

su participación, pues son los que van a dar las ideas más acabadas para facilitar su trabajo y su aprendizaje.

La importancia del diseño de la aplicación se basa en que éste será el que modele la interacción entre usuario y aplicación, y por tanto posibilitará o no la consecución de los objetivos perseguidos por el usuario ya sean encontrar información, comprar, comunicarse y aprender. [2]

Es fácil inferir que un buen diseño deberá ser comprensible, fácil de usar, amigable, claro, intuitivo y de fácil aprendizaje para el usuario. Para poder asegurar que un diseño cumple con estos requisitos no basta simplemente con una actitud empática del diseñador durante el desarrollo de la aplicación; es imprescindible la adopción por parte de éste de técnicas, procedimientos y métodos que aseguren empíricamente la adecuación del diseño a las necesidades, habilidades y objetivos del usuario. [2]

1.1.3 Factores Relevantes de la Usabilidad

La usabilidad consta de varios factores que son importantes para el desarrollo de su evaluación, como son: [3]

- **Facilidad de Aprendizaje:** Necesidad de minimizar el tiempo que el usuario emplea en aprender a utilizar correctamente el software.
- **Tiempo de Respuesta:** Capacidad del software para dar respuesta a las peticiones que le hace el usuario. Este factor es muy variable ya que depende de las características que tenga la PC donde se encuentre el usuario.
- **Flexibilidad:** Diversidad de formas de intercambiar el usuario con el sistema la información. Aportar flexibilidad al sistema implica brindar control al usuario, capacidad de sustitución y capacidad de adaptación.
- **Robustez:** Caracteriza la necesidad de que el usuario cumpla con sus objetivos.
- **Recuperabilidad:** Facilidad que brinda el software o la aplicación al usuario para corregir alguna operación que previamente se haya hecho pero que se ha reconocido el error.

- **Sintetizabilidad:** Este factor se caracteriza porque el usuario sea capaz de captar cuando ocurra algún cambio de operación en el sistema.

1.1.4 Interacción Persona – Ordenador

La Interacción persona – ordenador es la disciplina que estudia el intercambio entre las personas y los ordenadores. Ésta se encarga del diseño, evaluación e implementación de las interacciones. La comunicación entre el usuario y la máquina debe ser especial para poder comprender cada información y con qué está relacionada.

Los sistemas o software que se creen deben ser cada vez más fáciles de manejar y deben tratar de motivar a los usuarios a los cuales van dirigidos. La idea es que los usuarios sean capaces de aprender de una forma fácil y sencilla las principales cuestiones para manejar el software que está brindándole la información. El intercambio debe ser agradable para que el usuario se sienta identificado con la herramienta y de una forma u otra se motive a seguir visitando, trabajando e incluso recomendando dicho software. [3]

1.1.5 Experiencias del Usuario

Los usuarios rechazan todo aquel software donde pasan mucho trabajo buscando la información, pues ésta no es del todo clara, aquellos en los cuales tienen que navegar con mucha profundidad para encontrar lo que buscan, o simplemente otras en que el diseño les causa impresión y los confunde al volver a utilizarlos. Sería una pérdida de tiempo y de dinero tener que trabajar con software donde nunca se haya analizado y aplicado alguna técnica para mejorar el uso de los mismos pues estarían trabajando sin eficiencia y con pérdidas.

1.1.6 Diseño de Interacción

El diseño de interacción está conformado por los temas antes mencionados (Usabilidad, Interacción Persona-Ordenador y la Experiencia del Usuario). Se crea para relacionar las partes mencionadas y lograr que el resultado cumpla con los requisitos previstos y con una alta calidad. Para ello se tiene que conocer cómo es el producto que se va a crear, los servicios que va a brindar, el aspecto que va a tener en cuanto a color, textos, menús, imágenes.

Se recomienda realizar primero el diseño de interacción de un software para entonces comenzar a programar, debido a que los programadores, a pesar de ser personas inteligentes se centran más en la programación, que en los detalles extras que debe llevar un software.

Diseñar antes de programar genera un software de mayor calidad y usuarios más satisfechos, además, reduce la cantidad de cambios a realizar en el proceso de desarrollo. Los programadores podrían concentrarse exclusivamente en su tarea: *programar*. El producto final se desarrollaría por programadores más centrados en su función; en fin, se obtendrían mejores resultados.

1.1.7 Accesibilidad

Uno de los componentes esenciales y que va muy ligado a la usabilidad es la accesibilidad, ya que ésta define la manera en que se establecen las prioridades cuando se brinda la información. Todo software debe tratar de optimizar el tiempo que se utiliza para mostrar la información, por eso se hace énfasis en que la información esté organizada, pues así si existiera algún caso que no se pueda eliminar, por lo menos se acercaría bastante al caso más favorable.

En el caso de los software en la Web este factor es muy importante porque mantiene alto el número de visitantes que use el software y por ende un nivel de aceptación favorable. Usabilidad y Accesibilidad son dos términos relacionados, pero no son sinónimos. Usabilidad es la facilidad de uso y Accesibilidad es la capacidad de acceso e interacción con un sistema Web por todo tipo de usuarios, independientemente de sus capacidades o su contexto de Navegación.

1.1.8 Calidad de Software

La calidad de un software está dada de manera general en si cumple con todos los requisitos de la manera más óptima, más eficaz y además más asequible para las personas a quien va destinado. Busca satisfacer todas las necesidades y cumplir los objetivos que se proponga el usuario cubriendo todas las alternativas posibles. Debe garantizar una mejora continua, por lo que debe trazarse metas para que lo faciliten. Está basada generalmente en que pueda motivar a los usuarios cada vez más, logrando hacerlos usuarios fijos del software al cual se le haya aplicado. [3]

De acuerdo a la norma ISO/IEC 9126, la usabilidad está relativamente independizada de la calidad del software, pero esto no quiere decir que estén en bandos completamente separados, ya que un software que tiene aplicada una técnica de usabilidad adecuada influye en la calidad del mismo, es decir, con buena usabilidad existe un mayor grado de calidad.

La usabilidad es uno de los atributos de calidad claves en el desarrollo de software ya que sus atributos: eficacia, eficiencia y satisfacción son directamente proporcionales a la calidad. Los dos primeros están relacionados a la calidad objetiva del software. La eficacia, cumple con la funcionalidad y la eficiencia cumple con la rapidez y el bajo consumo. El tercer atributo que es la satisfacción, forma parte de la calidad subjetiva, que no es más que la calidad percibida por el usuario. [5]

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan equilibrar la filosofía de trabajo, en aras de lograr una mayor confiabilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

1.1.9 Ingeniería de la Usabilidad

La ingeniería de la usabilidad estudia y desarrolla software para que alcancen características cuantitativas que servirán para mejorar la usabilidad de los mismos. La aplicación de esta materia ayuda a que el software tenga una mayor calidad, que tenga mejor aceptación por los usuarios y que el producto final tenga los mejores resultados posibles. También se puede destacar que tiene gran importancia dentro del proceso los test de usabilidad que se realizan, las evaluaciones heurísticas y que se crea un prototipo de interfaz por el cual mediante la interacción de los usuarios se puede medir el nivel de motivación que los mismos tienen.

El objetivo de la ingeniería de la usabilidad es minimizar la sobrecarga cognitiva y perceptiva del usuario de esa aplicación. Utiliza un método de diseño iterativo con un prototipo rápido, cuyo esqueleto es el ciclo "análisis-diseño-implementación-evaluación", que se repite varias veces con vistas a ir enriqueciendo progresivamente el sistema. La etapa de evaluación del prototipo, confrontándolo con usuarios reales a

cada repetición del ciclo, se revela como trascendental para obtener resultados dignos de una ingeniería. [3]

1.1.10 Diseño de Interfaz de Usuario

Para que el producto tenga un cierto nivel de usabilidad no puede faltar un buen diseño que se base en el estudio que se haya hecho a diferentes usuarios mediante algún test o mediante la interacción con algún prototipo inicial que se haya creado. La información a la que los usuarios deben acceder debe ser de fácil alcance para este, pues uno de los objetivos principales es hacer que esta interfaz sea una ayuda para el usuario y no un trabajo más a la hora de interactuar con ella.

En el desarrollo de software se identifica a menudo la usabilidad con las características de la interfaz gráfica de usuario, sin embargo, no sólo tiene que ver con la interfaz. Es cierto que la interfaz gráfica es una parte importante del sistema, y un buen diseño de la misma puede hacer que un sistema aumente su nivel de aceptación, pero un sistema con un diseño de la interacción pobre no puede mejorar su nivel de usabilidad tan sólo cambiando la interfaz gráfica.

Por ejemplo, se ocupa del sistema de ayuda, de la documentación de usuario, y del procedimiento de instalación haciendo todas estas cosas fáciles para cualquier tipo de usuario. La forma en que las personas se relacionan con un producto o software mediante la interfaz la cual posibilita la unión (hardware - software), suele ser más profunda en la interacción persona - ordenador. Se conoce que la mayoría de los errores demandados por los usuarios tienen que ver con la usabilidad en las interfaces de las páginas y no con errores que éstas puedan traer en su funcionalidad. [3]

1.2 Evaluación de la Usabilidad.

La evaluación de la usabilidad abarca una serie de metodologías y técnicas que ayudan a medir la forma en que los usuarios son capaces de utilizar un sitio web, al mismo tiempo que determinan la manera en que lo hacen. El llevarla a cabo derivará en la creación de mejores productos, por lo que conseguirá que los usuarios realicen sus actividades más fácilmente. De hecho, sin evaluación será imposible saber si un producto cumple las expectativas de sus creadores, o si se adapta a su contexto social, físico y organizativo. [6]

La evaluación se presenta como la etapa más importante en el proceso de Diseño Centrado en el Usuario y supondrá un largo proceso de recogida sistemática de datos sobre cómo un usuario o grupo de usuarios usan un producto para una tarea concreta en un determinado entorno. Los resultados de la evaluación nos dirigirán hacia la correcta adaptación del sitio a sus necesidades. [7]

El principal objetivo de la evaluación de la usabilidad consiste en establecer si un sistema cumple las necesidades y expectativas del usuario y si, por lo tanto, éste se encuentra satisfecho. Además se encargará de recoger información útil para futuros desarrollos del producto, sentando de esta forma las bases de actuación en el diseño del sitio. [8]

1.2.1 Principios de evaluación de la usabilidad

Estos principios determinan algunos de los aspectos a tener en cuenta al revisar una página web.

- **Visibilidad del estado del sistema:** La web siempre debe tener informado al usuario acerca de lo que sucede y de lo que está haciendo. Así pues, se le debe ofrecer una constante retroalimentación respecto a las acciones que realice. Es vital asegurar que sabe exactamente dónde se encuentra en todo momento. Uno de los mayores problemas con los que se suele tropezar es la desorientación.
- **Lenguaje común entre sistema y usuario:** El sistema debe hablar el lenguaje del usuario por medio de palabras y conceptos familiares, usando convenciones del mundo real y organizando la información en orden lógico y natural. Se procurará huir de tecnicismos incomprensibles o mensajes crípticos.
- **Libertad y control por parte del usuario:** El usuario debe poseer el control absoluto del sistema. Siempre han de ofrecerse múltiples posibilidades y salidas de emergencia ya que a menudo, por error, se eligen opciones no deseadas. Habrá de marcarse claramente la posibilidad de deshacer y rehacer.
- **Consistencia y estándares:** La consistencia se refiere, por ejemplo, a no utilizar dos rótulos distintos para referirse a un mismo contenido o dos estilos diferentes en un mismo sitio. Habrá de

guardarse mayor cuidado si el desarrollo del sitio se encuentra a cargo por grupos de profesionales web. La solución que desde hace unos años se está llevando a cabo es la utilización de hojas de estilo, que facilitan mantener la coherencia deseada.

- **Prevención de errores:** Por encima de un correcto mensaje de error debe situarse un diseño usable que evite los errores. Los pasos e instrucciones se escribirán de manera clara, inequívoca, destacando las partes principales. Se recomienda el uso de validadores, antes de la definitiva publicación, para comprobar si existen fallos que se nos hubieran pasado por alto.
- **Es mejor reconocer que recordar:** El usuario no se encuentra obligado a recordar dónde se encontraba cierta información o cómo se llegaba a determinada página. Todas las instrucciones de uso del sistema, opciones y enlaces deben de ser visibles en todo momento.
- **Flexibilidad y eficiencia de uso:** El sitio debe ser fácil de usar para novatos y, al mismo tiempo, proporcionar atajos o aceleradores para usuarios avanzados. Una opción recomendable es la de permitir personalizar ciertas acciones frecuentes.
- **Diseño minimalista:** La información más importante debe situarse en la parte superior de la página, zona siempre visible en el navegador. Cualquier tipo de información que no sea relevante para el usuario que sobrecargue la interfaz debe ser eliminada. Por ello se recomienda reducir el número de imágenes al mínimo ya que cada una implica una descarga que, con conexiones lentas, se convierte en un problema serio.
- **Permitir al usuario solucionar el error:** Es conveniente que en el caso de la interacción del usuario con la página, si se detectase un error, se habrá de generar un mensaje que le informe de porqué ha fallado el sistema y cómo puede solucionarlo. Estos mensajes de error deberán ser claros y sin ambivalencias.

1.2.2 Técnicas de evaluación de Usabilidad

Existen diferentes técnicas de evaluación de sitios web que varían en su rigurosidad, costes y conocimientos necesarios para llevarlas a cabo.

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.[3]

- **Entrevista:** Se realiza con el objetivo de obtener datos que servirán para realizar la evaluación con la(s) técnica(s) determinadas a utilizar.
- **Cuestionario:** Su función es muy parecida a la entrevista, a diferencia que éste es un poco más específico a la hora de realizar las preguntas pertinentes.

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

- **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.
- **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.

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

- **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.
- **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.

La evaluación heurística es una de las variantes de la inspección de usabilidad donde los expertos o especialistas juzgan si cada elemento de la interfaz de usuario cumple con los principios de la usabilidad. El grupo de expertos investiga y analiza todos los aspectos relacionados con la interfaz de usuario para realizar una evaluación al respecto, utilizando una lista de principios que usualmente son aceptados.

Esta evaluación detecta aproximadamente el 42% de los problemas graves de diseño y el 32% de los problemas menores, dependiendo del número de evaluadores que revisen el sitio. Posteriormente se recomienda realizar un test de usuarios para completar la evaluación. [9]

En este trabajo se va a utilizar esta técnica ya que tiene grandes ventajas:

La principal ventaja de la evaluación por criterios es su bajo coste, en realidad este tipo de evaluación puede tener el coste que se desee. Un número mínimo de tres evaluadores permite realizar una evaluación por criterios. Los costes son por tanto mucho menores que cualquier otro método de evaluación.

En comparación con otras técnicas de evaluación donde el observador debe interpretar las acciones del usuario, en la evaluación heurística no es necesaria la interpretación externa, porque las ideas, comentarios e información elaborada por los evaluadores están contenidos en sus informes.

Otra ventaja es que en la evaluación por criterios es posible interrogar a los evaluadores, profundizar en determinadas cuestiones de interés y ayudarles cuando tienen problemas. En los test de usuario por el contrario, los usuarios no deben disponer de más información que la necesaria para permitir su comportamiento espontáneo.

En el caso de que se disponga de un sitio totalmente elaborado es aconsejable realizar la evaluación heurística antes que las pruebas de usuario. Un test de usuario previo sólo serviría para detectar problemas de usabilidad que en una evaluación heurística hubieran sido fácilmente detectadas por los expertos a un coste mucho menor.

Debido a estos problemas muchos usuarios pueden quedar bloqueados en las fases iniciales del proceso a evaluar y muchos elementos de interés no se podrán evaluar (a no ser que se instruya a los usuarios

sobre cómo superar las dificultades, lo que anularía la validez del test). Detectar previamente los problemas más graves de usabilidad en la evaluación por criterios permite realizar posteriormente pruebas de usuarios con mejores resultados. [9]

1.3 Tendencias actuales de la Usabilidad

Los avances tecnológicos hacen, en muchas ocasiones, la vida más fácil. Se fabrican actualmente una gran cantidad de dispositivos que poco a poco los usuarios van usando con mayor frecuencia. Muchos de esos dispositivos y aparatos electrónicos disponen de un ordenador y de una interfaz digital que requiere cada vez más interacción. Esto significa más trabajo, o más oportunidad, para la optimización de la usabilidad. [12]

Así como las tecnologías están cambiando a cada segundo, la usabilidad tiene que seguir su paso. De este modo hoy día ya no es posible estar conforme con una página que se pueda visualizar bien en una PC, sin saber si realmente fue útil al usuario final.

Actualmente es necesario pensar que el usuario exige cada vez más de un sitio y espera antes que nada cubrir sus objetivos y después poder interactuar con él, así como consultarlo a través del dispositivo que posea desde cualquier parte del mundo.

Desde sus inicios, la usabilidad ha hecho énfasis en el usuario, justo es él quien le dio origen. De esta forma, al analizar las tendencias de la usabilidad en la web, lo primero que debe contemplarse es quiénes la están empleando.

Los usuarios de la web actual son aquellos que buscan una alta interacción y participación, ser lectores o escritores que encuentren fácilmente lo que buscan y de forma ágil puedan aportar riqueza a los sitios.

Un aspecto crítico que ha preocupado durante mucho tiempo a los expertos de usabilidad es cómo eliminar la subjetividad en la evaluación de la experiencia del usuario que emplea el sitio que se ha diseñado, la tendencia actual es automatizar esta experiencia y obtener datos duros de la misma.

Esto sin duda es algo muy interesante para todos aquellos que han venido trabajando en usabilidad, ya

que permite dar el seguimiento a cada clic del usuario, sin importar dónde se encuentre, ni la cantidad de que se trate.

A continuación se presentan algunos ejemplos de sistemas que existen para evaluar de una forma u otra la usabilidad en sitios web.

1.4 Sistemas de Evaluación de Usabilidad Existentes

1.4.1 ClickHeat

Es una herramienta visual abierta que permite ver en qué puntos hacen clic más frecuentemente los usuarios, y cuáles permanecen ignorados. De esta forma aporta información sobre qué elementos pueden ser eliminados para evitar una sobre carga de la página o bien, darse cuenta de zonas en las que no se tienen los elementos necesarios para llamar la atención del visitante y que son importantes para la finalidad de las páginas web. [14]

Esta herramienta no fue posible utilizarla para el desarrollo de la aplicación ya que no cumple con los objetivos que plantea el procedimiento propuesto por la facultad.

1.4.2 Plugin YSlow para Firebug

Es una herramienta gratuita para Mozilla Firefox que brinda información sobre los distintos elementos que componen las páginas que se van a evaluar, desde el punto de vista del rendimiento y la optimización.

El principal resultado es un análisis del rendimiento de la página y luego hace un reporte detallado, además de recomendaciones necesarias para optimizar adecuadamente los Java Script, CSS y HTML, Colocar los Java script al final del HTML.

Además de ello tiene una opción para ver las estadísticas de rendimiento y una opción para visualizar todos los componentes cargados mediante la página. En definitiva con este complemento Firebug se convierte en una potente herramienta de análisis para el desarrollo de páginas web. [19]

Esta herramienta no cumple con los objetivos que persigue el procedimiento plantado en la facultad por lo que no fue posible utilizarla.

1.4.3 Clickdensity

Clickdensity es una herramienta que permite generar mapas de calor, o comúnmente llamados heatmaps de un sitio web, esto sirve para saber dónde hacen clic los usuarios que visitan el sitio, mientras más clic se haga en una zona esta se pone más caliente.

Este tipo de mapas sirve para saber dónde orientar los anuncios en los sitios, o ciertos contenidos importantes que se quiere que los visitantes vean, de este modo se puede organizar todo dependiendo de la densidad de clic de la pantalla. El monitoreo de esta actividad se hace mediante un script que hay que insertar en el código del sitio, así a medida que los usuarios entran el sistema genera los mapas. [20]

Esta herramienta no fue útil ya que no cumple con los objetivos que persigue el procedimiento plantado en la facultad.

1.4.4 ClickTale

ClickTale estudia el comportamiento de los usuarios de forma colectiva, creando mapas de calor de la web, en las que se pueden observar las zonas más vistas, e individualmente mediante la grabación de la navegación de los visitantes por la web para que se pueda visualizar y analizar posteriormente las acciones de los visitantes y entender el qué y el porqué de sus movimientos.

Infiltrando un código de seguimiento en la web se podrá tener un registro de las acciones de los visitantes, ya sean movimientos del mouse, los clic, lo que han escrito en un formulario, los campos que se han saltado, la cantidad de visitas, el tiempo total y el tiempo promedio que se ha destinado a una acción específica para así entender cómo los visitantes recorren el sitio, qué les resulta más interesante y qué menos. [21]

En el caso de esta herramienta no fue de mucha ayuda ya que mide aspectos diferentes a los que se tienen en cuenta en el procedimiento.

1.5 Lenguajes de programación Web

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, y es utilizado para controlar el comportamiento físico y lógico de una máquina.

La programación web, parte de las siglas www, que significan World Wide Web o telaraña mundial. Para realizar una página con la programación Web, se deben tener claros tres conceptos fundamentales, los cuales son, el URL (Uniform Resource Locator), es un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita.

A continuación se presentan varios lenguajes de programación que se utilizan para la construcción de sitios Web:

1.5.1 Java

Es un lenguaje orientado a objeto, de una plataforma independiente. Este lenguaje tiene muchas similitudes con el lenguaje C y C++. Java es independiente de la arquitectura: podría ejecutarse en un entorno UNIX, Mac o Windows. Esto es posible ya que el código generado por el compilador se ejecuta mediante una máquina virtual y por el procesador del ordenador directamente.

1.5.2 PERL

Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesador de texto y no tener ninguna de las limitaciones de los otros lenguajes de script. Sus principales características son, la estructura completa de Perl deriva ampliamente del lenguaje C, es fácil de usar, soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles.

1.5.3 PHP

PHP (Hypertext Preprocessor) es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgre, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas. [9]

Ventajas.

- Es un lenguaje multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server).
- Similar en sintaxis a C y a PERL.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Evitando que el usuario tenga que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son válidos se hace en el servidor y no en el cliente (como se hace con java script) de forma que se puede chequear que no se reciban solicitudes adulteradas.
- Es software libre.
- Se puede obtener en la web y su código está disponible bajo la licencia GPL.

Desventajas:

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.

Con el estudio de los lenguajes de programación web realizado con anterioridad, se llegó a la conclusión de que el lenguaje a utilizar en la implementación de la aplicación propuesta será PHP; ya que entre las principales características de dicho lenguaje es que es multiplataforma, es software libre, tiene un excelente soporte de acceso a bases de datos.

1.6 Framework

Un framework es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. La utilización del mismo en el desarrollo de una aplicación implica un cierto coste inicial de aprendizaje, aunque a largo plazo es probable que facilite tanto el desarrollo como el mantenimiento. [11]

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener y facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. [15]

1.6.1 Seagull

Permite realizar una programación modular, posee un sistema de gestión de Contenidos (CMS). Entre sus características de encuentran su compatibilidad con PHP 4 y PHP 5, ORM integrado, incorpora el patrón MVC, uso de templates, soporte para múltiples bases de datos, validación de datos, alto nivel de configuración, autenticación, integración de librerías PEAR.

1.6.2 Prado

Está basado en componentes de eventos con el objetivo de acelerar el desarrollo de aplicaciones web usando PHP 5. El concepto del desarrollo de aplicaciones en Prado es diferente, se utilizan componentes, eventos y propiedades en vez de procedimientos, URL y parámetros.

1.6.3 Zoop

Es un Framework PHP Orientado a Objeto basado en el modelo MVC, sus desarrolladores lo caracterizan por ser rápido, eficiente y fácil destinado para programadores.

1.6.4 WACT

Facilita un aprovechamiento modular, sus componentes permiten la integración con otras aplicaciones web. Implementa el patrón MVC. La versión actual de WACT requiere PHP 5, por ahora sólo está disponible desde Subversión.

1.6.5 Symfony

Symfony es un completo framework diseñado para optimizar, separa la lógica de negocio, la del servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas (Unix, Linux, etc.) como en plataformas Windows.

Características de Symfony

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.

- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Según el estudio realizado sobre los framework para PHP. Se pudo ver las posibilidades que estos brindan para el desarrollo de cualquier sistema web. En la investigación realizada, se seleccionó el Symfony por las ventajas que este nos proporcionaba, como son:

- Probado: Ha sido probado con éxito durante varios años en aplicaciones muy diferentes.
- Licencia: Utiliza una licencia MIT, con la que puedes hacer aplicaciones web gratuitas y de software libre.
- Seguro: Se puede controlar hasta el último acceso a la información e incluye por defecto protección contra ataques XSS y CSRF.
- Documentado: Se trata del framework PHP mejor documentado: miles de páginas en el wiki oficial, tutoriales de hasta 250 páginas y un libro gratuito de casi 500 páginas. Además, el libro está completamente traducido al español.
- Calidad: Su código fuente incluye más de 8000 pruebas unitarias y funcionales.

1.7 Gestor de Base de Datos

Los sistemas de gestión de base de datos (SGBD); son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para un buen manejo de datos.

1.7.1 Firebird

Es un sistema de administración de base de datos relacional de código abierto, fue liberado por Borland en 2000. Su código fue reescrito de C a C++. Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows. Presenta un ejecutable pequeño, con requerimientos de hardware bajos. Utiliza la arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros, además de que soporta transacciones y claves foráneas.

1.7.2 PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional (ORDBMS), es una derivación libre (OpenSource) del proyecto POSTGRES, esta versión incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, pero a pesar de esto PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Unas de sus principales características es que soporta distintos tipos de datos, además de los del tipo base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. y permite la creación de tipos propios.

Ventajas:

- Posee una gran escalabilidad. Es capaz de ajustarse al número de Unidades Centrales de Procesamiento (CPUs) y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
- Implementa el uso de rollback, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- Requiere pocos recursos de hardware y la simplificación del proceso de administración de licencias de software, que no es necesario cuando se usa software libre.
- Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Puede lidiar con gran volumen de datos.

Desventajas:

- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.

1.7.3 MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Funciona en diferentes plataformas. Posee un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. Soporta a grandes bases de datos. Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT. MySQL Server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas.

Ventajas:

- La velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
- El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro y de buscadores de aplicaciones.

Desventajas:

- Carece de soporte para transacciones, rollback y subconsultas.
- El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
- No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad. [17]
- No es de código abierto.

Se decide utilizar como servidor de base de datos PostgreSQL: ya que este está ampliamente considerado como uno de los sistemas de bases de datos de código abierto más avanzados del mundo y sus características y potencia se ajustan muy bien al volumen de información que debe manejar la aplicación.

1.8 Servidor Web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. [18]

1.8.1 Lighttpd

Es un servidor web diseñado para ser rápido, seguro, flexible, y fiel a los estándares. Está optimizado para entornos donde la velocidad es muy importante, y por eso consume menos CPU y memoria RAM, incluso en servidores con 512MB funcionará de maravilla. Este servidor es software libre y se distribuye bajo la licencia BSD. Funciona en GNU/Linux y UNIX de forma oficial. Para Microsoft Windows. También permite comunicarse con programas externos mediante FastCGI o SCGI, que son mejoras al CGI original (también soportado). De esta forma, se pueden usar programas en prácticamente cualquier lenguaje de programación.

1.8.2 Apache

Apache es un servidor de páginas web de código abierto multiplataforma y modular, se desarrolla dentro del proyecto HTTP Server (http) de la Apache Software Foundation. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

Características principales:

- Trabaja sobre múltiples plataformas (Unix, Linux, MacOSX, Vms, Win32, OS2, etc.).
- Incluye módulos que se cargan de forma dinámica.
- Soporta CGI, Perl, PHP.
- Soporte para Bases de datos.
- Soporte SSL para transacciones seguras.
- Incluye soporte para host virtuales.
- Soporta HTTP 1.1.
- Código Abierto.
- Rápido.
- Eficiente. [16]

Luego de hacerse un estudio profundo de los servidores web, se decidió que para el entorno a desarrollar el mejor servidor web es el Apache. Este servidor es multiplataforma lo que lo convierte en un servidor prácticamente universal, es una tecnología gratuita de código fuente abierto, esto le da una transparencia a este software de manera que si se desea ver qué es lo que se está instalando como servidor, se puede conocer, sin ningún secreto, sin ninguna puerta trasera; y por último es altamente configurable de diseño modular por lo que es muy sencillo ampliar sus capacidades.

1.9 Metodologías de desarrollo de Software

Con el objetivo de crear y mantener las aplicaciones de software, aplicando las tecnologías y prácticas computacionales, surge la Ingeniería de Software. El desarrollo y evolución constante experimentada por los procesos de Ingeniería de Software, ha traído consigo la realización de varias tareas en este campo, como son: análisis de requisitos, especificación, diseño y arquitectura, programación, prueba, documentación y mantenimiento.

Se hace necesario definir metodologías para guiar el proceso de desarrollo de un producto de software. Las metodologías se definen por pasos a seguir para el cumplimiento de un objetivo. El objetivo dentro del desarrollo del software es producir un producto de alta calidad que cumpla con los requerimientos del cliente.

1.9.1 Programación Extrema (XP)

Programación Extrema es una metodología de desarrollo que se encuentra dentro de las llamadas metodologías ágiles en las que se da máxima prioridad a la obtención de resultados. XP no se basa en principios nuevos, sino que todas o casi todas sus características ya se conocen dentro de la ingeniería del Software, las cuales se complementan para minimizar los típicos problemas que pueden surgir en todo desarrollo de proyectos software.

Ésta metodología promueve la idea de que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Las características fundamentales de esta metodología son:

- Desarrollo iterativo e incremental.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad pero sin modificar su comportamiento.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto.

- **Simplicidad en el código:** Es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo en caso necesario, que realizar algo complicado y quizás nunca utilizarlo.

1.9.2 El Proceso Unificado de Modelado (RUP)

El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos. Dicho proceso divide en 4 fases el desarrollo del software:

- **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- **Transición:** El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Características del Proceso Unificado

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Representan requerimientos funcionales. Los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso.
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema

que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. La arquitectura se refleja en los casos de uso pues cada producto tiene tanto una función como una forma, ninguna es suficiente por sí sola.

- **Iterativo e Incremental:** RUP propone dividir el trabajo en partes más pequeñas o mini proyectos, donde cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son mini proyectos.

Para obtener una mejor organización, eficiencia y eficacia en el desarrollo de dicha aplicación se realizó un estudio detallado de dos de las metodologías más usadas a nivel mundial: XP y RUP; después de haber realizado la investigación se llegó a la conclusión de que se utilizará la metodología RUP, debido a que se adapta en gran medida, tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo. RUP permite la captura de varias de las mejores prácticas en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos. Permite un entendimiento incremental del problema a través de refinamientos sucesivos. Anticipa y ubica cambios. Se inclina a la automatización de muchas de las tediosas tareas asociadas con el desarrollo de software.

1.10 Herramienta CASE

1.10.1 Rational Rose

Rational Rose Enterprise es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++ y Visual Basic. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente. [23]

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo, cubre todo el ciclo de vida de un proyecto: [24]

Presenta algunas características adicionales como son:

- Capacidad de análisis de calidad de código
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos
- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación
- Integración con otras herramientas de desarrollo de Rational
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.

1.10.2 Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. [25]

Independientemente a todas las facilidades que ofrece Visual Paradigm se escogió Rational Rose como herramienta CASE para realizar el modelado visual de los procesos en el sistema debido a la mayor experiencia que se tenía en el uso de esta herramienta y a que se requieren menos requisitos de hardware. Además Rational Rose posee todas las características que los desarrolladores, analistas, y arquitectos exigen: soporte UML incomparable, completo soporte al equipo, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria, facilidad de uso, integración optimizada, entre otras características que conllevan a que Rational sea reconocido como el líder

tecnológico por su rol en el desarrollo del UML, lenguaje de modelado seleccionado para la realización de este trabajo.

Conclusiones

Se puede afirmar que los software que brindan entretenimiento, diversión y ocio son los más visitados, los que más motivan pues los temas son bien seleccionados de acuerdo al gusto de muchas personas. Cuando se comience con la creación de un software se debe valorar y aplicar la usabilidad para que así el producto final sea capaz de atraer a quienes lo usarán y además no sientan la necesidad de buscar otro producto que le sea más fácil de trabajar o más fácil de encontrar lo que necesitan.

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio, a través de los conceptos y definiciones planteadas. Se analizaron las características de diferentes herramientas, para la creación de un software o aplicación así como algunas metodologías.

Después de este análisis y la fundamentación realizada, el lenguaje de programación que se utilizará es PHP para el control de las diferentes funciones a realizar, como gestor de base de datos se estableció el PostgreSQL. Se empleará la metodología RUP y la herramienta que se usará como entorno de creación de diagramas para UML el Rational Rose.

CAPÍTULO 2. Características del Sistema (SEUSGOW)

Introducción

En el presente capítulo se ofrece una descripción del Negocio en el que se va a trabajar, se especifican los actores y trabajadores del Negocio, los Casos de Uso, los diagramas de actividades y el modelo de objetos del Negocio. Para el buen desarrollo del sistema a desarrollar es necesario basarse en los requisitos funcionales y no funcionales, a partir de los cuales se identifican las opciones del sistema, que se representan mediante los Casos de Uso del sistema y la descripción de los mismos.

2.1 Proceso a Automatizar

El procedimiento a automatizar (Procedimiento de Evaluación de Usabilidad en los Software de Gestión sobre Plataforma Web) describe varios procesos que se enuncian a continuación.

- El Jefe de Proyecto hace la solicitud para la evaluación de la Usabilidad al Líder de la Comisión Evaluadora (CE) mediante una la Planilla de Solicitud propuesta para ello, utilizando las vías digitales o tangibles. Las vías digitales serían Correo Electrónico o la subida online de la planilla a un sistema Web y la vía tangible sería, haciendo la solicitud en la oficina determinada. Esta planilla debe ser de fácil acceso y es por esto que debe estar publicada en el Portal de la Facultad para su adquisición al igual que el Correo electrónico del encargado de recibirla (Líder de la CE).
- El Líder de la CE analiza la solicitud de evaluación. El objetivo de este análisis es verificar la completitud de los datos necesarios para llevar a cabo dicha evaluación. En caso de no estar todos los datos se le informa al Jefe de Proyecto y se le explican las causas por las cuales no fue aprobada la evaluación, en caso de tener todos los datos se le informa al Jefe de Proyecto y se le explican los detalles de la revisión.
- En el FT Requisitos se genera el primer prototipo de Interfaz de Usuario (IU) y es precisamente en este FT donde se chequean los primeros aspectos de la Evaluación. Participan en este paso primeramente tres de los evaluadores (Especialista en Comunicación y Lenguaje, Especialista en Diseño Gráfico y Especialista en Diseño de Interfaz II). El primero de éstos chequea los aspectos

Capítulo 2. Características del sistema(SEUSGOW)

- 1, 2 y 5. El Especialista en Diseño Gráfico y Especialista en Diseño de Interfaz II chequean el primero de los aspectos definidos para cada uno de ellos en el mismo Anexo.
- Documentar los resultados obtenidos en el Documento de Evaluación para su posterior uso y que sirva de guía para la corrección de errores. Este paso lo realiza el Especialista en Comunicación y Lenguaje.
 - Se repiten los pasos 1, 2 y 3 (Solicitud para nueva etapa de evaluación). Luego se continúa con el Paso 8.
 - En el **FT Análisis y Diseño** se realiza el Diseño de la IU y un nuevo Prototipo de IU. Se debe hacer la revisión de los errores encontrados en la fase anterior de evaluación, en caso de que aún existan deben ser señalados una vez más e influyen en la evaluación de esta etapa. Aquí se realizan un mayor número de chequeos participando 3 evaluadores, el Especialista en Diseño Gráfico, Especialista en Diseño de Interfaz I y el Especialista en Diseño de Interfaz II. El Especialista en Diseño Gráfico chequea los aspectos 2, 4 y 5. El Especialista en Diseño de Interfaz I chequea los aspectos 3, 4, 7 y 8. El Especialista en Diseño de Interfaz II chequea los aspectos 3, 4 y 7.
 - Documentar los resultados obtenidos para su posterior uso y que sirva de guía para la corrección de errores actualizando el Documento de Evaluación..
 - Se repiten los pasos 1, 2 y 3 (Solicitud para nueva etapa de evaluación). Luego se continúa con el Paso 12.
 - La última etapa de evaluación del procedimiento planteado es en el **FT Pruebas** donde se debe hacer la revisión de los errores encontrados en la fase anterior de evaluación, en caso de que aún existan deben ser señalados una vez más e influyen en la evaluación de esta etapa. Aquí participan todos los Expertos y chequean los siguientes aspectos: Especialista en Comunicación y Lenguaje (aspectos 3, 4 y 6), Especialista en Diseño Gráfico (aspecto 3), Especialista en Diseño de Interfaz I (aspectos 2, 5 y 6), Especialista en Diseño de Interfaz II (aspectos 2, 5, 6 y 8) y el Especialista en Arquitectura de Información que chequea todos los aspectos definidos para él.
 - Documentar los resultados obtenidos para su posterior uso y que sirva de guía para la corrección de errores actualizando el Documento de Evaluación.

Capítulo 2. Características del sistema(SEUSGOW)

- Se realiza una reunión final con el Jefe de Proyecto, Líder de la CE, Expertos de la CE y Planificador del Proyecto. En esta reunión el Líder de la CE da una explicación de los resultados obtenidos dando pie a un debate de las revisiones y a la evaluación final. Además se exponen sugerencias o recomendaciones si es posible.

En el desarrollo de este procedimiento se plantea la necesidad de que se haga la solicitud de ésta desde el mismo comienzo del desarrollo de software, ya que uno de sus objetivos es ir corrigiendo errores desde el inicio y durante el desarrollo de software.

El Líder de la CE es el responsable de elaborar el informe final de la evaluación a partir de las deficiencias encontradas por los expertos tomando en cuenta lo antes expuesto. El mismo dará una puntuación a cada uno de los aspectos chequeados por cada uno de los especialistas y luego emitirá una puntuación final que definirá el nivel de Usabilidad del software en cuestión. El resultado final estará basado en 5 criterios:

1. **No se aplica:** El software en cuestión no presenta ningún indicio de Usabilidad.
2. **Problema Crítico:** El software en cuestión presenta problemas críticos de Usabilidad. *Solución Inmediata.*
3. **Problema Grave:** El software en cuestión presenta problemas graves de Usabilidad. *Solución de Alta Prioridad.*
4. **Problema mínimo:** El software en cuestión presenta un alto nivel de Usabilidad pero presenta algunos problemas no fundamentales. *Solución de Baja Prioridad.*
5. **No es problema:** El software en cuestión no presenta problemas de Usabilidad por lo que se debe obtener un producto con nivel de Usabilidad alta.

Para el caso que el software a evaluar **presente ayuda y buscador** se utiliza el rango de puntos siguientes:

Total de aspectos a evaluar: 35

Suficientes aspectos para ser Usable: 28

Rangos:

- 01 - 15 → Problema Crítico.
- 16 - 27 → Problemas de Usabilidad.
- 28 - 31 → Usable.
- 32 - 33 → Buena Usabilidad.
- 34 - 35 → Óptima Usabilidad.

Para el caso que el software a evaluar **presente ayuda pero no presente buscador** se utiliza el rango de puntos siguientes:

Total de aspectos a evaluar: 31

Suficientes aspectos para ser Usable: 25

Rangos:

- 01 - 12 → Problema Crítico.
- 13 - 24 → Problemas de Usabilidad.
- 25 - 27 → Usable.
- 28 - 29 → Buena Usabilidad.
- 30 - 31 → Óptima Usabilidad.

Para el caso que el software a evaluar **presente buscador pero no presente ayuda** se utiliza el rango de puntos siguientes:

Total de aspectos a evaluar: 32

Suficientes aspectos para ser Usable: 26

Rangos:

- 01 - 12 → Problema Crítico.
- 13 - 25 → Problemas de Usabilidad.
- 26 - 28 → Usable.
- 29 - 30 → Buena Usabilidad

31 - 32 → Óptima Usabilidad.

Para el caso que el software a evaluar **no presente buscador ni ayuda** se utiliza el rango de puntos siguientes:

Total de aspectos a evaluar: 28

Suficientes aspectos para ser Usable:

Rangos:

01 - 10 → Problema Crítico.

11 - 20 → Problemas de Usabilidad.

22 - 24 → Usable.

25 - 26 → Buena Usabilidad.

27 - 28 → Óptima Usabilidad.

Las actividades que se toman en cuenta para incluir en los rangos son las que están evaluadas entre 4 y 5 puntos, es decir, cuando se evalúa el software se cuentan las actividades entre 4 y 5 puntos, el número obtenido de dicha suma se revisa con que intervalo coincide y entonces se da el criterio correspondiente, El aspecto que evalúa la existencia de más de un objetivo, no se le da puntuación en caso de que no exista en el software.

2.1.1 Lista de Chequeo por Especialista

Cada lista de chequeo realiza la evaluación de acuerdo a una especialidad, las especialidades son: Especialista en Comunicación, Especialista en Diseño Gráfico, Especialista en Diseño de Interfaz I, Especialista en Diseño de Interfaz II y Especialista en Arquitectura de Información. Cada uno de ellos evalúa los aspectos de acuerdo a su especialidad y en la etapa que se solicite. Se determina que los principales procesos a automatizar se dividen en los siguientes grupos: Gestión de Solicitud, Gestión de Evaluación y Gestión de Especialistas.

2.2 Modelo de Negocio

El Modelo de Negocio es un artefacto del flujo de trabajo Modelamiento del Negocio en la Fase de Inicio de la metodología RUP. Sus objetivos son: comprender la estructura y la dinámica del centro en el cual se va a implantar el sistema, comprender los problemas existentes en la actualidad dentro de la unidad e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común del centro y derivar los requisitos del sistema que va a soportar la organización. [26]

2.2.1 Entidades del Negocio

Representa un contenedor de información, algo físico que se utilice en el proceso del Negocio y que sirva para obtener información o para actualizar información.

- ✓ Planilla de Solicitud
- ✓ Documento de Evaluación

2.2.2 Actores del Negocio

Un Actor del Negocio es cualquier individuo, grupo, organización o máquina que interactúa con el Negocio.

Actores del Negocio	Descripción
Jefe de Proyecto	Jefes de proyectos de la facultad 2 encargados de realizar las solicitudes de evaluación.

2.2.3 Trabajadores de Negocio

Capítulo 2. Características del sistema(SEUSGOW)

Define el comportamiento y responsabilidad (rol) de un individuo, grupo de individuos, Sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Trabajadores del Negocio	Descripción
Líder de la Comisión	Líder de la Comisión Evaluadora encargado de realizar distintas tareas durante el proceso de evaluación con la finalidad de organizar y controlar el trabajo a realizar.
Especialista en Comunicación	Especialista en Comunicación encargado de la revisión de los objetivos y lenguajes
Especialista en Diseño Gráfico	Especialista en Diseño Gráfico encargado de la revisión de las opciones de diseño y las posibilidades del usuario.
Especialista en Diseño de Interfaz I	Especialista en Diseño de Interfaz I encargado la revisión de los estándares y consistencia.
Especialista en Diseño de Interfaz II	Especialista en Diseño de Interfaz II encargado de la revisión los diálogos, ayudas y búsqueda.
Especialista en Arquitectura de Información	Especialista en Arquitectura de Información encargado de la revisión de la arquitectura, las fechas y noticias.

2.2.4 Casos de uso del Negocio

Un Caso de uso del Negocio representa a un proceso de Negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del Negocio.

- ✓ Realizar Solicitud de evaluación de usabilidad
- ✓ Realizar Evaluación

2.2.5 Descripción de los Principales Casos de Uso del Negocio

2.2.5.1 Descripción del caso de uso Realizar Solicitud de evaluación de usabilidad.

Caso de Uso:	Realizar Solicitud de evaluación de usabilidad.
Actores:	Jefe de proyecto
Resumen:	El Caso de uso se inicia cuando el Jefe de proyecto envía la solicitud al Líder de la Comisión Evaluadora.
Flujo Normal de Eventos	
Acción de Actor	Respuesta del Negocio
1. El Jefe de proyecto realiza la solicitud	2. El Líder de la Comisión Evaluadora analiza la solicitud mediante una planilla de evaluación verificando la completitud de los datos necesarios para llevar a cabo dicha evaluación y se le informa al Jefe de Proyecto.
1. El Jefe de proyecto recibe la información.	

Prioridad	Alta
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	2.1 El Líder de la comisión Evaluadora le informa al Jefe de Proyecto que no están todos los datos y le explica las causas por las cuales no fue aprobada la evaluación.

2.2.5.2 Descripción del caso de uso Realizar Evaluación.

Caso de Uso:	Realizar Evaluación
Actores:	Jefe de Proyecto
Resumen:	El caso de uso se inicia cuando el líder de la comisión le informa a los especialistas el comienzo de la revisión.
Casos de uso Asociados	Realizar Solicitud
Flujo Normal de Eventos	
Acción de Actor	Respuesta del Negocio
	1. En el FT de Requisitos los Especialista en Comunicación y Lenguaje, en Diseño Gráfico y en Diseño de Interfaz II

Capítulo 2. Características del sistema(SEUSGOW)

	<p>chequean los aspectos pertenecientes a este flujo de trabajo y se registran los resultados en un documento de evaluación.</p> <p>2. En FT de Análisis y Diseño se revisan los errores encontrados en la fase anterior y se comienza una nueva evaluación chequeando los especialistas en Diseño Gráfico, en Diseño de Interfaz I y el de Diseño de Interfaz II los aspectos pertenecientes a este flujo y se registran los resultados en un documento de evaluación.</p> <p>3. En el FT de Pruebas se revisan los errores encontrados en la fase anterior y se comienza una nueva evaluación chequeando todos los expertos los aspectos pertenecientes a este flujo y se registran los resultados en un documento de evaluación.</p>
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Negocio</p>
	<p>2.1 y 3.1 En caso de que aún existan errores deben ser señalados una vez más e influyen en la evaluación de la etapa actual.</p>

2.2.6 Diagrama de Casos de Uso del Negocio

Un diagrama de Casos de Uso del Negocio representa gráficamente a los procesos del Negocio y su interacción con los actores del Negocio.

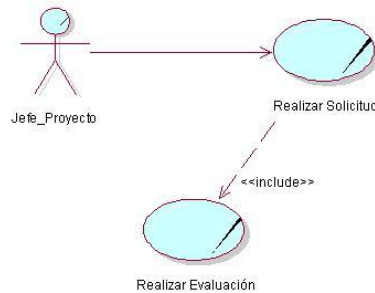


Figura 1. Diagrama Casos de Uso del Negocio

2.2.7 Modelo Objeto

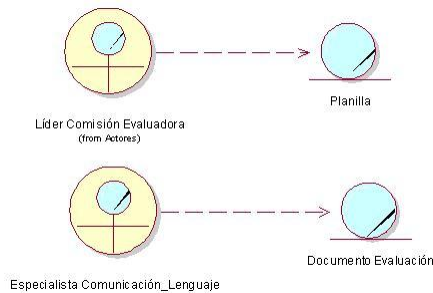


Figura 2. Modelo Objeto

2.2.8 Reglas del Negocio

Las Reglas de Negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del Negocio:

- Sólo el Líder de la Comisión puede tener acceso a la información de todos los especialistas.
- El Líder de la Comisión mantiene un control estricto sobre los resultados de dicha evaluación.

- El Líder de la Comisión es el responsable de elaborar el informe final de la evaluación a partir de las deficiencias encontradas por los especialistas.
- Cada especialista tiene asignada una actividad, sólo ellos son los responsables de la revisión según el rol asignado.
- El Jefe de proyecto es el único autorizado en realizar la solicitud.
- El Líder de la comisión es el responsable de aceptar o no la solicitud.
- La Usabilidad del software se clasifica en crítico, problemas de usabilidad, usable, buena usabilidad y óptima usabilidad, según la puntuación obtenida en la evaluación.
- El Líder de la Comisión es el único autorizado a la gestión de la lista de chequeos.

2.3 Modelo de Sistema

2.3.1 Especificación de Requisitos del Software.

El proceso de definir y especificar los requisitos del software consiste en establecer los servicios que un producto de software debe cumplir. Proceso que lleva a la construcción de un documento conocido como Especificación de requisitos del software, esencial para lograr un producto de alta calidad.

Obtener los Requisitos del Software es paso esencial para la construcción de un producto de software para el posterior desarrollo de las etapas. Un error en este proceso, traería graves consecuencias en la obtención de un producto, pues podría no cumplir las expectativas del cliente.

2.3.1.1 Requisitos Funcionales.

Los Requisitos Funcionales son capacidades o condiciones con las que debe cumplir el sistema a elaborar, no alteran la funcionalidad del software, se mantienen invariables sin importar con que cualidades o propiedades se relacionen.

A partir del estudio y las investigaciones que se realizaron de los procesos del negocio, se obtuvieron una serie de requisitos funcionales que ha cumplir el software.

RF1 Autenticar Usuario

RF2 Gestionar Usuario

- Crear Usuario
- Modificar Usuario
- Eliminar Usuario

RF3 Gestionar Evaluador

- Crear Evaluador
- Modificar Evaluador
- Eliminar Evaluador

RF4 Gestionar Lista de Chequeo

- Insertar Lista de Chequeo
- Modificar Lista de Chequeo
- Eliminar Lista de chequeo

RF5 Gestionar Evaluación Objetivos y Lenguaje

- Mostrar lista de chequeo Objetivos y Lenguaje.
- Asignar evaluación por punto de chequeo.
- Modificar Evaluación Objetivos y Lenguaje.
- Eliminar Evaluación Objetivos y Lenguaje.

RF6 Gestionar Evaluación Diseño y Posibilidades del usuario

- Mostrar lista de chequeo Diseño y Posibilidades del usuario.
- Asignar Evaluación a cada aspecto de la lista de chequeo.
- Modificar Evaluación Diseño y Posibilidades del usuario

- Eliminar Evaluación Diseño y Posibilidades del usuario

RF7 Gestionar Evaluación Estándares y Consistencia

- Mostrar lista de chequeo Estándares y Consistencia
- Asignar Evaluación a cada aspecto de la lista de chequeo..
- Modificar Evaluación Estándares y Consistencia.
- Eliminar Evaluación Estándares y Consistencia.

RF8 Gestionar Evaluación Ayudas y Búsquedas

- Mostrar lista de chequeo Ayudas y Búsquedas.
- Asignar Evaluación a cada aspecto de la lista de chequeo.
- Modificar Evaluación Ayudas y Búsquedas.
- Eliminar Evaluación Ayudas y Búsquedas.

RF9 Gestionar Evaluación Arquitectura, fechas y Noticias

- Mostrar lista de chequeo Arquitectura, fechas y Noticias.
- Asignar Evaluación a cada aspecto de la lista de chequeo.
- Modificar Evaluación Arquitectura, fechas y Noticias.
- Eliminar Evaluación Arquitectura, fechas y Noticias.

RF10 Gestionar Solicitud.

- Insertar Solicitud
- Modificar Solicitud
- Eliminar Solicitud

RF11 Generar Documento de Evaluación.

RF12 Gestionar Etapa de Evaluación.

- Insertar Etapa de Evaluación
- Modificar Etapa de Evaluación

RF13 Gestionar Errores por Etapa

- Insertar Errores por Etapa
- Modificar Errores por Etapa
- Eliminar Errores por Etapa

RF14 Gestionar Roles

- Insertar Roles
- Modificar Roles
- Eliminar Roles

RF15 Gestionar Recomendación.

- Insertar Recomendación
- Modificar Recomendación
- Eliminar Recomendación

RF16 Gestionar Criterio de Evaluación

- Insertar Criterio de Evaluación
- Modificar Criterio de Evaluación
- Eliminar Criterio de Evaluación

RF17 Gestionar Rango Evaluación

- Insertar Rango Evaluación
- Modificar Rango Evaluación
- Eliminar Rango Evaluación

RF18 Gestionar Pautas Diseño

- Insertar Pautas de Diseño
- Modificar Pautas de Diseño
- Eliminar Pautas de Diseño

2.3.1.2 Requisitos No Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. A continuación se muestran los no funcionales:

Requisitos de software

Estaciones de Trabajo (PC cliente)

- Navegador web: Internet Explorer y Mozilla Firefox.
- Sistema operativo Windows XP o superior.

Servidor de Aplicaciones

- Servidor Web Apache versión 2.0 o superior.
- PHP versión 5.0 o superior.
- Marco de trabajo Symfony 1.2

Servidor de Base de Datos

- Sistema gestor de base de dato PostgreSQL 8.2.

Requisitos de hardware

- Memoria RAM de 256 MB o superior
- Los servidores Web y de Bases de Datos deben poseer 512 MB de memoria RAM como mínimo.
- Tarjeta de Red.

Requisitos de diseño e implementación

- Se utilizará Rational Rose Enterprise Edition 2003 como herramienta CASE para el modelado y obtención de los diferentes artefactos que requiere el software.
- El software utilizará PHP 5 como lenguaje de programación.
- Se utilizará como framework Symfony.

Requisitos de apariencia o interfaz externa

- Diseño sencillo, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.
- Navegación simple.

Requisitos de seguridad

- Autenticación de usuario necesaria para acceder al Sistema.
- Garantizar diferentes niveles de acceso al Sistema, brindado por la asignación de roles.

Requisitos de usabilidad

- El sistema presentará sus mensajes y textos, en idioma español.
- El sistema facilitará la entrada de datos a los usuarios, presentando listas que permitan escoger valores descriptivos.
- La aplicación Web debe facilitar la interacción usuario-aplicación, con el objetivo de que el usuario se sienta atraído por la tecnología evitando cualquier tipo de rechazo.

Soporte

- Realizar pruebas al software para comprobar su funcionalidad.
- El gestor de base de datos usado debe soportar grandes volúmenes de datos y velocidad de procesamiento.

2.3.2 Actores del Sistema

Son todas aquellas personas u otros Sistemas que interactúan con el mismo.

Actores	Descripción
Líder de la Comisión	Líder de la Comisión Evaluadora es quien decide si se acepta o no la solicitud de evaluación en el sistema y confecciona el informe final de dicha evaluación.
Especialista en Comunicación	Especialista en Comunicación interactúa con el sistema revisando los objetivos y lenguajes
Especialista en Diseño Gráfico	Especialista en Diseño Gráfico interactúa con el sistema revisando las opciones de diseño y las posibilidades del usuario.
Especialista en Diseño de Interfaz I	Especialista en Diseño de Interfaz I interactúa con el sistema revisando los estándares y consistencia del software.
Especialista en Diseño de Interfaz II	Especialista en Diseño de Interfaz II interactúa con el sistema revisando los diálogos, ayudas y búsqueda.
Especialista en Arquitectura de Información	Especialista en Arquitectura de Información interactúa con el sistema

	revisando la arquitectura, las fechas y noticias.
Administrador	Administrador del sistema encargado de la gestión de usuarios y roles.

2.3.3 Diagrama de caso de Uso del Sistema

EL diagrama de Casos de Uso del Sistema es un artefacto de Ingeniería de Software que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario, permitiendo de esta forma el establecimiento de un acuerdo entre clientes y desarrolladores sobre las condiciones y requerimientos que debe cumplir el sistema.

Este modelo está formado por actores, casos de uso y las relaciones que se establecen entre estos, es decir representa gráficamente a los procesos y su interacción con los actores.

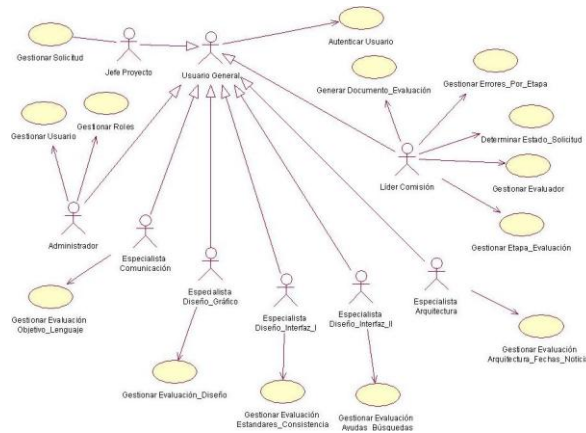


Figura 3. Diagrama de Casos de uso del Sistema

2.3.4 Descripción de Casos de uso del Sistema

La descripción de Casos de uso del Sistema se realiza con el objetivo de describir brevemente como funcionarán los Casos de uso en el Sistema, y así conseguir un mejor entendimiento de cada uno.

2.3.4.1 Descripción del CU Gestionar Usuario

Nombre del CU	Gestionar Usuario
Actores	Administrador
Propósito	Permitir crear, modificar y eliminar usuarios.
Resumen	El caso de uso inicia cuando el administrador del sistema necesita crear, modificar o eliminar un usuario.
Referencias	RF2
Prioridad	Critico
Sección " Principal "	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1- El administrador necesita crear, eliminar o modificar los datos del usuario.	El sistema ejecuta alguna de las siguientes acciones: <ul style="list-style-type: none"> a. Si decide crear un usuario, ir a la sección "Crear Usuario" b. Si decide Eliminar usuario, ir a la sección "Eliminar Usuario". c. Si decide Modificar un Usuario, ir a la sección "Modificar usuario".

Sección " Crear usuario "	
Acción del actor	Respuesta del sistema
1- El administrador selecciona crear usuarios. 2- Introduce los datos del usuario	1.1 El sistema muestra una interfaz con los campos para agregar usuarios. 1.2 El sistema almacena los datos del usuario en la base de datos.
Flujo Alternativo	
Sección " Modificar Usuario "	
Flujo Normal de Eventos	
Acción del actor	Respuesta del Sistema
1- El administrador selecciona el usuario a modificar.	1.1 El sistema muestra una interfaz con los datos a modificar.
2- El administrador introduce los datos del usuario.	2.1 El sistema guarda los datos del usuario en la base de datos.
Flujo Alternativo	
Sección " Eliminar Usuario "	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema

Capítulo 2. Características del sistema(SEUSGOW)

1- El administrador selecciona el usuario que desea eliminar.	1.1 Elimina el usuario de la base de datos.
Flujo Alternativo	
3. El administrador cancela la acción de eliminar el usuario.	3.1 El sistema regresa a la acción 2 de esta sección.

2.3.4.2 Descripción del CU Gestionar Evaluación Objetivos y Lenguaje

Nombre del CU	Gestionar Evaluación Objetivos y Lenguaje
Actores	Especialista en Objetivos y Lenguajes
Propósito	Permitir asignar modificar o eliminar una evaluación.
Resumen	Este caso de uso se inicia cuando el Especialista en Comunicación y Lenguajes va a asignar una evaluación a cada uno de los renglones de evaluación (renglones relacionados con los objetivos y lenguajes del software), también podría modificar la evaluación en caso de haber una equivocación y eliminarla además.
Referencias	RF5
Prioridad	Crítico
Sección Principal	

Flujo Normal de Eventos	
Acción del actor	Respuesta del Sistema
<p>1- El Especialista en Comunicación y Lenguaje necesita asignar, modificar o eliminar una determinada evaluación.</p>	<p>El sistema ejecuta alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> a. Si decide Asignar Evaluación, ir a la sección "Asignar Evaluación" b. Si decide Modificar Evaluación, ir a la sección "Modificar Evaluación". c. Si decide Eliminar Evaluación, ir a la sección "Eliminar Evaluación".
Sección " Asignar Evaluación "	
Acción del actor	Respuesta del sistema
<p>1. El Especialista en Comunicación y Lenguaje selecciona asignar evaluación.</p> <p>2. Asigna la evaluación.</p>	<p>1. El sistema muestra una interfaz con una lista de chequeos para asignarles una evaluación al software.</p> <p>2. El sistema almacena los datos de la evaluación en el documento de evaluación para su posterior uso y de guía para la corrección de errores.</p>

Capítulo 2. Características del sistema(SEUSGOW)

Flujo Alternativo	
Sección " Modificar Evaluación"	
Flujo Normal de Eventos	
Acción del actor	Respuesta del Sistema
1- El Especialista en Comunicación y Lenguaje selecciona la evaluación a modificar.	1.1 El sistema muestra una interfaz con la evaluación a modificar.
2-El Especialista en Comunicación y Lenguaje modifica la evaluación,	2.1 El sistema guarda los datos de la evaluación en el documento de evaluación.
Flujo Alternativo	
Sección " Eliminar Evaluación"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1- El Especialista en Comunicación y Lenguaje selecciona la evaluación que desea eliminar.	1.1 Elimina la evaluación del documento de evaluación.
Flujo Alternativo	
Sección " Modificar Evaluación"	
Flujo Normal de Eventos	

Capítulo 2. Características del sistema(SEUSGOW)

Acción del actor	Respuesta del Sistema
1- El Especialista en Comunicación selecciona la evaluación a modificar.	1.1 El sistema muestra una interfaz con la evaluación a modificar.
2-El Especialista en Comunicación modifica la evaluación,	2.1 El sistema guarda los datos de la evaluación en el documento de evaluación.
Flujo Alternativo	

Conclusiones

En el presente capítulo se realizó toda la descripción del proceso de Negocio y Sistema, obteniendo sus respectivos diagramas de casos de uso, la descripción de cada caso, sus relaciones y el modelo de objetos del Negocio. También se determinaron las actividades a automatizar por el sistema, a través de los requerimientos funcionales y con los que se obtuvieron los casos de uso del sistema. Además se describieron los requisitos no funcionales que definen las cualidades que el sistema debe cumplir.

CAPÍTULO 3. Análisis y Diseño del Sistema (SEUSGOW)

Introducción


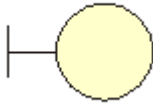
En este capítulo se realiza el análisis y diseño del sistema que se desarrollará. Se utiliza como metodología RUP que incluye principios y diagramas que muestran la forma en que los componentes de software interactúan y se comportan dentro del sistema. Además se tratan temas específicos sobre los modelos de análisis, diseño y sobre su construcción como una propuesta RUP para el desarrollo del software; todo esto para lograr al final un eficiente diseño del producto.

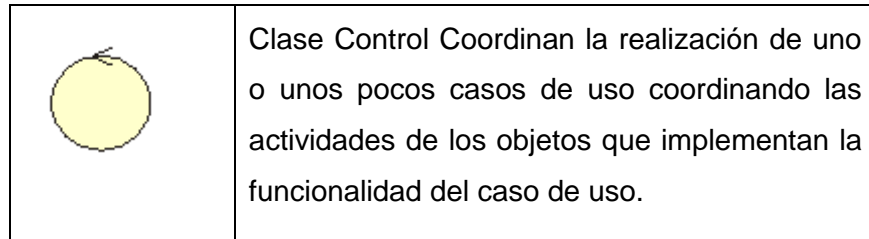
3.1 Modelo de análisis.

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución.

3.2 Diagramas de clases del análisis.

En el diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de las mismas. Además se representa una abstracción de una o varias clases:

	Clase Entidad: Modelan información que posee larga vida y que es a menudo persistente.
	Clase Interfaz: Modelan la interacción entre el sistema y sus actores.



3.2.1 Diagrama de clases de análisis de Gestionar Usuario

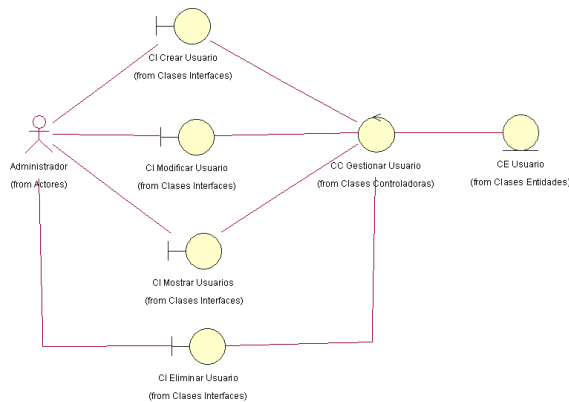


Figura 4. Diagrama de clases de análisis de Gestionar Usuario

3.2.2 Diagrama de clases de análisis de Gestionar Evaluación Objetivos y Lenguaje.

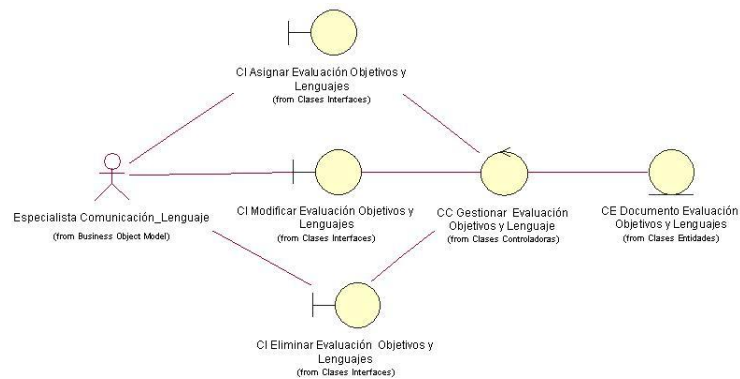


Figura 5. Diagrama de clases de análisis de Gestionar Evaluación Objetivos y Lenguaje

3.3 Diagramas de Interacción

El Diagrama de interacción se utiliza para modelar los aspectos dinámicos del sistema, consta de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Los diagramas de interacción contienen: objetos, enlaces y mensajes. Tienen como función principal describir cómo interactúan las clases con sus respectivos objetos del análisis, mediante diagramas de colaboración o de secuencia.

3.3.1 Diagrama de Colaboración

Se utilizan para ilustrar la realización de un Caso de Uso. Muestra como los objetos interactúan para lograr el comportamiento de un Caso de Uso o parte de este y de esta forma definen los roles de los mismos.

3.3.1.1 Caso de Uso Gestionar Evaluación Objetivos y Lenguajes: Escenario Asignar Evaluación Objetivos y Lenguajes.

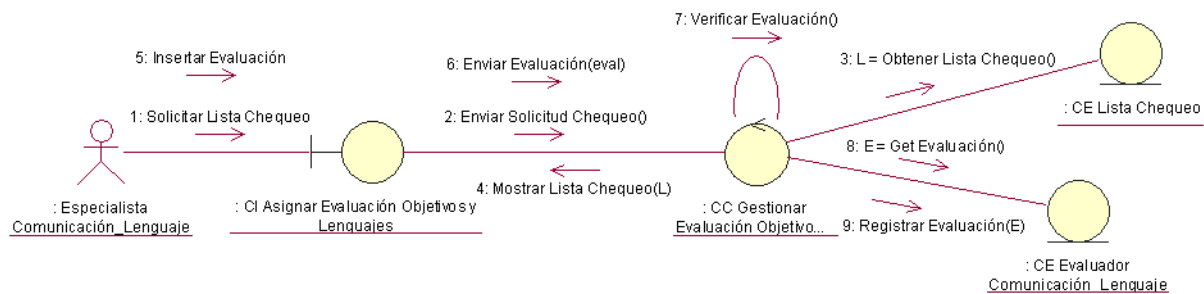


Figura 6. Caso de Uso Gestionar Evaluación Objetivos y Lenguajes: Escenario Asignar Evaluación Objetivos y Lenguajes.

3.3.1.2 Caso de Uso Gestionar Evaluación Objetivos y Lenguajes: Escenario Modificar Evaluación Objetivos y Lenguajes.

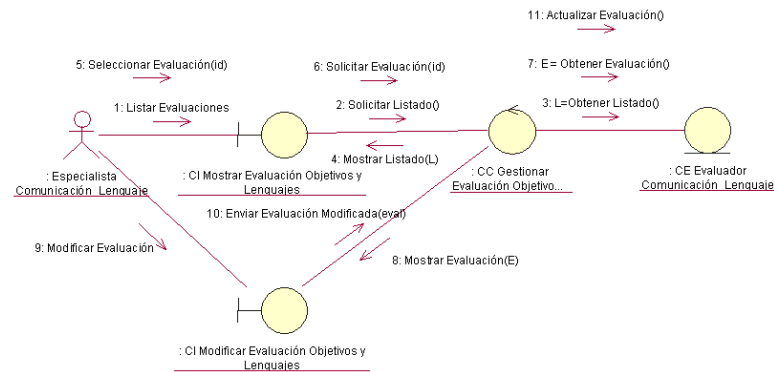


Figura 7. Caso de Uso Gestionar Evaluación Objetivos y Lenguajes: Escenario Modificar Evaluación Objetivos y Lenguajes.

3.3.1.3 Caso de Uso Gestionar Evaluación Objetivos y Lenguajes: Escenario Eliminar Evaluación Objetivos y Lenguajes.

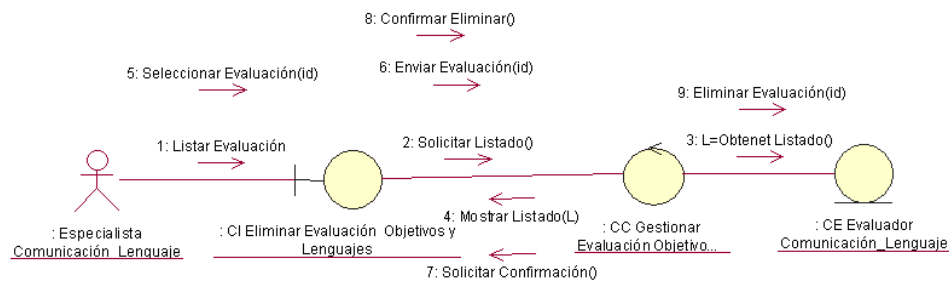


Figura 8. Caso de Uso Gestionar Evaluación Objetivos y Lenguajes: Escenario Eliminar Evaluación Objetivos y Lenguajes

3.4 Modelo Diseño

Es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a implementación. Representa a los casos de uso en el dominio de la solución.

El Modelo de Diseño puede contener: los diagramas, las clases, paquetes, subsistemas, capsulas, protocolos, interfaces, relaciones, colaboraciones, atributos, las realizaciones de los casos de uso, entre otros que se puedan considerar para el sistema en desarrollo.

3.5 Diagrama de clases del diseño

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases del software y contienen las clases, atributos, métodos, navegabilidad y dependencias existentes entre ellas. En las aplicaciones Web, éste representa la colaboración entre las páginas donde cada página lógica puede ser representada como una clase.

3.5.1 Gestionar Evaluación Objetivos y Lenguajes

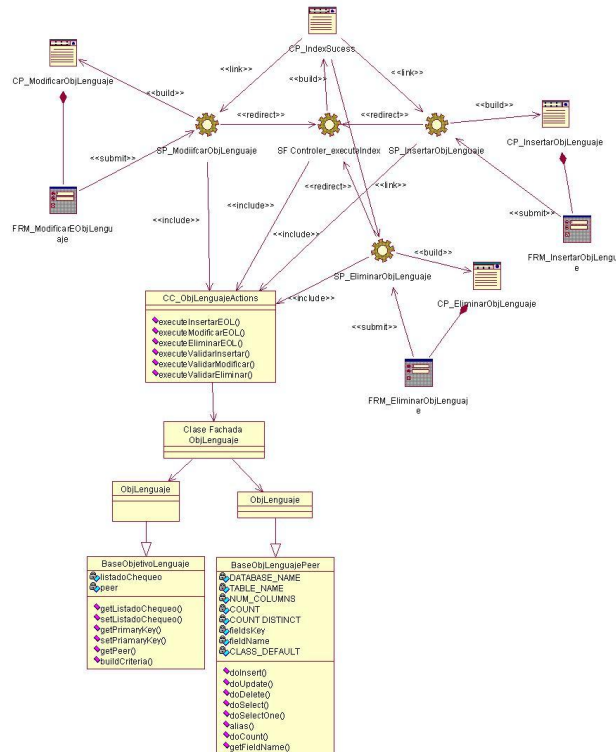


Figura 9. Diagrama de clases del diseño Gestionar Evaluación Objetivos y Lenguajes

3.6 Descripción de la Arquitectura

La Arquitectura es una vista estructural de alto nivel, la cual se define muy tempranamente en el ciclo de vida y define los estilos o grupos de estilos adecuados para cumplir con los requerimientos no funcionales. El surgimiento de los patrones ha sido uno de los tres grandes temas de la ingeniería que han dominado la arquitectura de software.

La aplicación fue diseñada siguiendo el patrón de arquitectura Modelo Vista Controlador (MVC), Symfony está basado en este patrón clásico del diseño web, que está formado por tres niveles:

- **El Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **La Vista:** Transforma el modelo en una página web que permite al usuario interactuar con ella.
- **El Controlador:** Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

Symfony divide el proyecto creado en tres carpetas principales: App, Web y Lib. En App almacena todos los módulos creados, con sus success(vistas) y sus actions(controladoras), en Lib todo lo referente a la capa del modelo, la cual se agrupa específicamente en la carpeta Model, donde en la carpeta om, de almacena la abstracción de los datos y en map la abstracción generada para el mapeo de la base de datos. Por último en la carpeta Web se almacenas varios componentes referentes a la Vista, las imágenes a utilizar, los CCS y los archivos Java Script, entre otros.

Importante destacar el tratamiento que recibe la Modelo, de parte del framework de desarrollo, el cual crea

una capa de abstracción de los datos, usando la aplicación Propel, por lo cual no es necesario modelar la Modelo y nos concentramos en modelar las Vistas y las Controladoras.

3.7 Modelo Lógico de datos

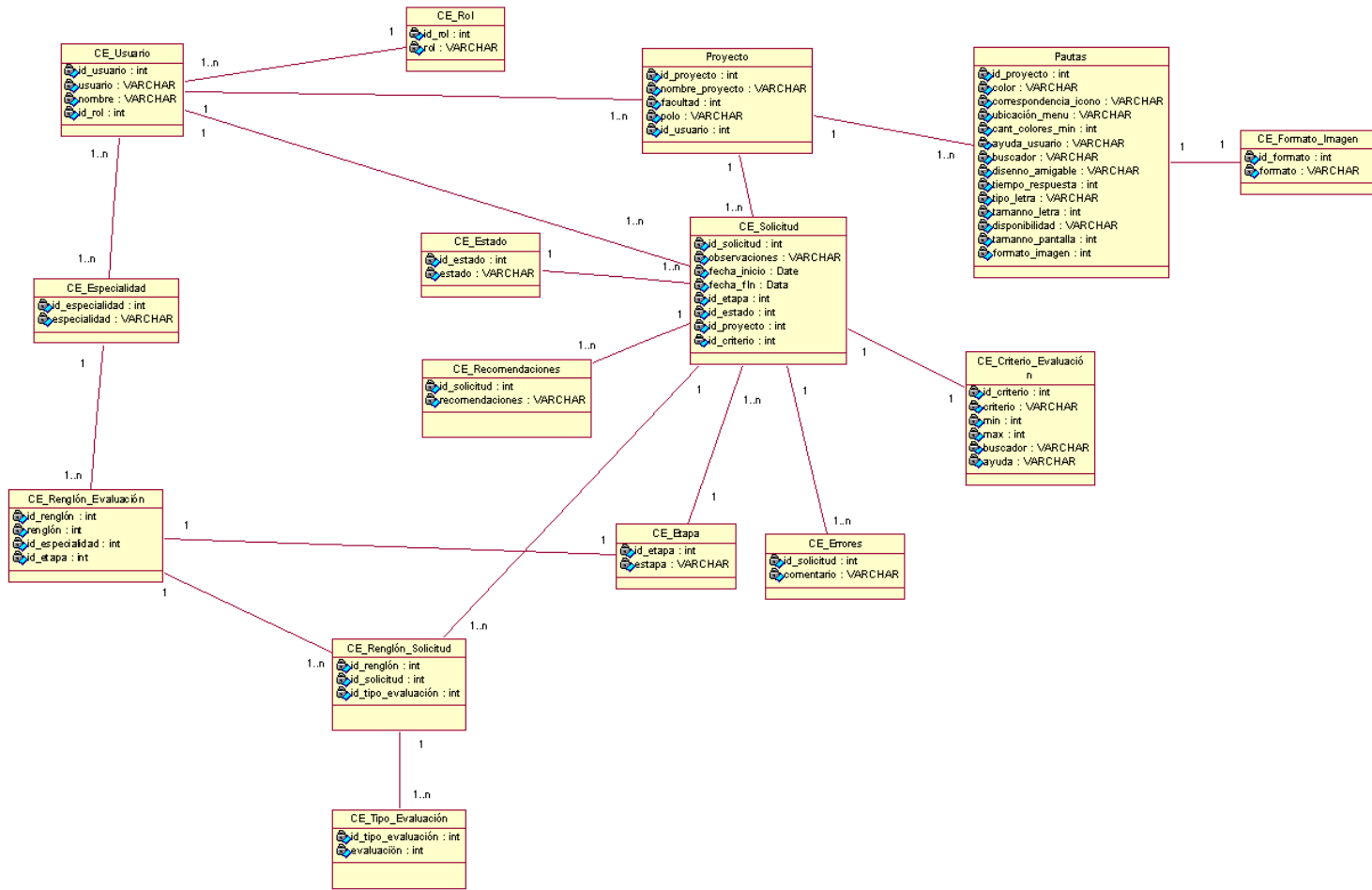


Figura 10. Modelo Lógico de Datos

3.8 Modelo Físico de datos

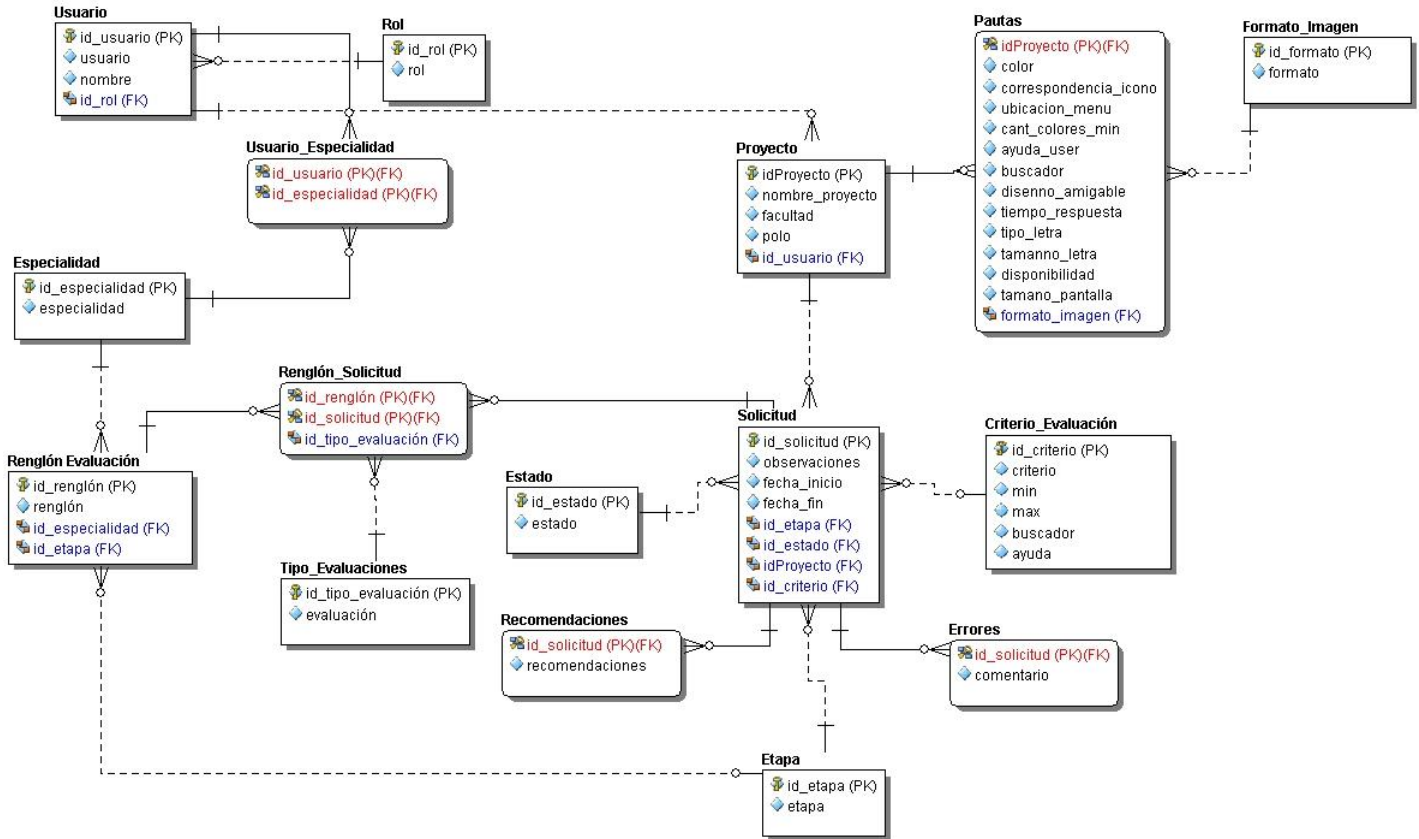


Figura 11. Modelo Físico de Datos

Conclusiones

Como resultado de la elaboración de este capítulo se logró la terminación de la etapa de análisis y diseño del sistema, se realizaron los diagramas de clases del análisis y diagramas de clases del diseño para cada caso de uso identificado. Se realizaron además los diagramas de interacción (diagramas de secuencia y diagrama de colaboración) para cada escenario de los casos de uso. Como parte de este capítulo se describe la arquitectura y los patrones de diseño utilizados, y por último se plasma el diagrama de clases persistentes.

CAPÍTULO 4. Implementación

Introducción

Partiendo de los resultados del diseño, en el presente capítulo se analiza la disciplina de implementación, se presentan el Diagrama de despliegue y el Diagrama de componentes como objetivo primordial de esta fase. Además de que se lleva a cabo el proceso de pruebas siguiendo el método de caja negra.

4.1 Modelo de Despliegue.

El Modelo de Despliegue agrupa los objetos que describen la distribución física del Sistema en términos de funcionalidad entre los nodos. Cada nodo representa un recurso que interviene en el despliegue del Sistema. Describe la arquitectura física del Sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software. Además, describen la topología del Sistema, es decir, la estructura de los elementos de hardware y software que ejecuta cada uno de ellos.

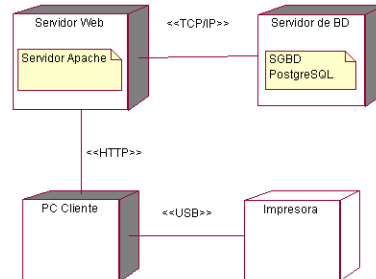


Figura 12. Diagrama de Despliegue

4.2 Diagrama de componentes.

Un diagrama de componentes muestra un conjunto de componentes y sus relaciones. Los diagramas de componentes se utilizan para describir la vista de implementación estática de un sistema y están estrechamente relacionados con los diagramas de clases, ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones. Al utilizar Symfony, los diagramas de componentes, muestran como están distribuidos según el patrón arquitectónico Modelo-vista-Controlador.

El Controlador Frontal (sfFrontWebController) maneja las peticiones web, siendo el punto de entrada de toda la aplicación en un entorno determinado. El controlador frontal es el encargado de determinar qué combinación de módulo-acción se ejecutará.

El paquete Vista es el encargado de producir las páginas que se muestran como resultado de las acciones que se soliciten, las cuales se integran con el layout.

El paquete controlador realiza todas las acciones, estas actions.php están relacionadas con todos los archivos Success.php de la vista que contiene el código que liga la lógica de negocio con la presentación.

En Symfony los formularios están compuestos por campos (Widgets) y validadores (Validators), cada campo debe tener asignado un validador para comprobar los datos que vienen de las vistas y proteger de ataques al sistema. En el paquete Validators están recogidos las clases que permiten las validaciones de los formularios para cada tipo de datos.

El paquete modelo es la capa que contiene las clases: php, las Peer, las Bases y las Base Peer. Estas clases son construidas por el subsistema Propel de Symfony para el acceso a datos, permitiendo el acceso mediante el mapeo de objetos a la base de datos SIGIPE.

4.2.1 Diagrama de Componentes Autenticar Usuario

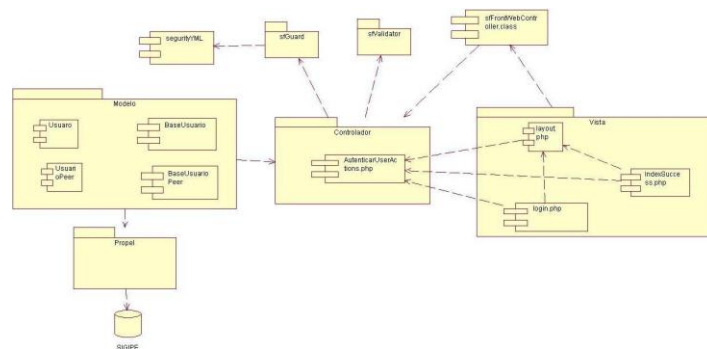


Figura 13. Diagrama de Componentes Autenticar Usuario

4.2.2 Diagrama de Componente Gestionar Usuario

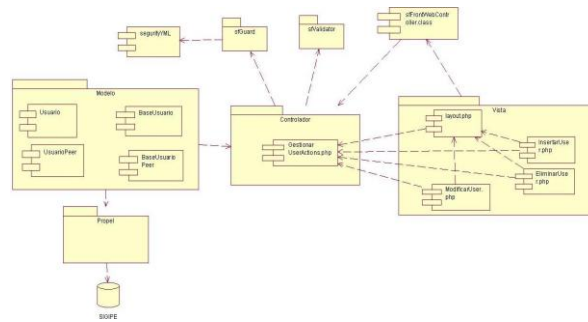


Figura 14. Diagrama de Componentes Gestionar Usuario

4.2.3 Diagrama de Componente Gestionar Evaluación Objetivos y Lenguajes

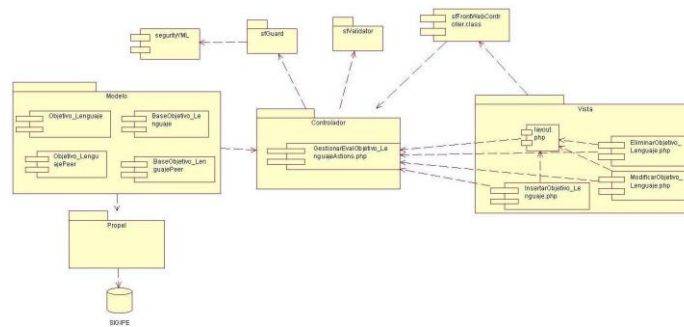


Figura 15. Diagrama de Componentes Gestionar Evaluación Objetivos y Lenguajes

Conclusiones

En este capítulo se ha mostrado mediante el diagrama de despliegue los nodos necesarios para la utilización del sistema, garantizando que este funcione correctamente. Lo que se traduce en la utilización de una PC cliente en donde el usuario accede a la aplicación, una PC servidor web que es donde estará situada la aplicación. Además de una PC servidor de base de datos en donde se encontrará la base de datos del sistema y una impresora para que el usuario tenga la posibilidad de imprimir los reportes efectuados por la aplicación. También se mostró la relación existente entre todos los archivos que conforman el sistema, y su relación con la base de datos mediante el diagrama de componentes del sistema.

CAPÍTULO 5. Estudio de Factibilidad

Introducción

La planificación de proyectos tiene objetivo primordial proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, costos y planificación temporal, así como, establecer planes razonables para realizar las actividades de ingeniería de software y manejar los cambios de los proyectos de software. En el presente capítulo se realizará una estimación en términos de esfuerzo que permitirá considerar el tiempo necesario para la entrega del sistema, fechas de terminación para cada etapa de desarrollo, entre otras. El método que se va a utilizar para calcular la estimación es el Análisis de puntos de casos de uso. El mismo permite predecir el tamaño de un sistema a partir de las características de los requerimientos expresados en casos de uso.

Método de Estimación Puntos por Casos de Uso

La estimación mediante el análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. A continuación, se detallan los pasos a seguir para la aplicación de éste método.

Cálculo de Puntos de Casos de Uso sin ajustar

Se calcula a partir de la siguiente ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Factor de Peso de los Actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Actor	Tipo de actor	Factor de Peso
Líder de la Comisión	Complejo	3
Especialista en Comunicación	Complejo	3
Especialista en Diseño Gráfico	Complejo	3
Especialista en Diseño de Interfaz I	Complejo	3
Especialista en Diseño de Interfaz II	Complejo	3
Especialista en Arquitectura de Información	Complejo	3
Administrador	Complejo	3

Tabla. Factor de Peso de Actores sin Ajustar.

Cantidad de actores de Tipo complejo: 7

UAW=Σ (actores*Peso)

UAW=7*3

UAW= 21.

Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una **transacción** se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia y está representada por uno o más pasos del flujo de eventos principal del Caso de Uso, pudiendo existir más de una transacción dentro del mismo Caso de Uso. Los criterios se muestran en la siguiente tabla:

Nombre del CU	Tipo	Descripción	Peso
Gestionar usuario	Medio	7 Transacciones	10
Gestionar Roles	Medio	7 Transacciones	10
Gestionar errores por etapa	Medio	7 Transacciones	10
Generar Documento Evaluación	Simple	2 Transacciones	5
Gestionar Cronograma de Trabajo	Medio	7 Transacciones	10

Determinar Estado Solicitud	Simple	3 Transacciones	5
Gestionar Evaluador	Medio	7 Transacciones	10
Gestionar Etapa de Evaluación	Medio	7 Transacciones	10
Gestionar Evaluación Arquitectura, Fechas y Noticias	Medio	7 Transacciones	10
Gestionar Evaluación Ayudas y Búsquedas	Medio	7 Transacciones	10
Gestionar Evaluación Estándares y Consistencia	Medio	7 Transacciones	10
Gestionar Evaluación de diseño	Medio	7 Transacciones	10
Gestionar Evaluación Objetivos y Lenguajes	Medio	7 Transacciones	10
Gestionar Pautas de Diseño	Complejo	8 Transacciones	15
Gestionar Lista de chequeo	Complejo	8 Transacciones	15
Autenticar Usuario	Simple	1 Transacciones	5

Tabla. Factor de Peso de los Casos de Uso sin ajustar

Cantidad de casos de uso medios: 14

Cantidad de casos de uso simples: 2.

$$\text{UUCW} = \Sigma \text{ CU Medios} * \text{Peso} + \Sigma \text{ CU Simples} * \text{Peso} + \Sigma \text{ CU Complejo} * \text{Peso}$$

$$\text{UUCW} = 11 * 10 + 3 * 5 + 2 * 15$$

$$\text{UUCW} = 155.$$

Cálculo de Puntos de Casos de Uso ajustados

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de estos factores:

Factor	Descripción	Valor	Peso	Valor * Peso

T1	Sistema distribuido	0	2	0
T2	Objetivos de performance o tiempo de respuesta	3	1	3
T3	Eficiencia del usuario final	3	1	3
T4	Procesamiento interno complejo	3	1	3
T5	El código debe ser reutilizable	5	1	5
T6	Facilidad de instalación	5	0.5	2.5
T7	Facilidad de uso	5	0.5	2.5
T8	Portabilidad	3	2	6
T9	Facilidad de cambio	2	1	2
T10	Concurrencia	4	1	4
T11	Incluye objetivos especiales de seguridad	5	1	5
T12	Provee acceso directo a terceras partes	0	1	0
T13	Se requieren facilidades especiales de entrenamiento a	2	1	2

	usuarios			
--	----------	--	--	--

Tabla. Factor de complejidad técnica.

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$\text{TCF} = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$\text{TCF} = 0.6 + 0.01 \times (0 + 3 + 3 + 3 + 5 + 2.5 + 2.5 + 6 + 2 + 4 + 5 + 0 + 2)$$

$$\text{TCF} = 0.6 + (0.01 \times 38)$$

$$\text{TCF} = 0.6 + 0.38$$

$$\text{TCF} = 0.98$$

Factor de ambiente

Para calcular este coeficiente se analizan una serie de factores que pueden determinar el tiempo requerido para el desarrollo del sistema, teniendo en cuenta aspectos como habilidades, conocimientos, entre otros, de los involucrados en la realización del sistema. A estos factores se les asocia un valor de 0 a 5 que determina el vínculo del mismo con las características deseadas para del sistema.

Factor	Descripción	Peso	Valor Asignado	Valor * Peso
E1	Familiaridad con el modelo de proyecto utilizado	1.5	0	0
E2	Experiencia en la aplicación	0.5	2	1
E3	Experiencia en orientación	1	3	3

	a objetos			
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	2	2
E6	Estabilidad de los requerimientos	2	5	10
E7	Personal part-time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	3	-3

Tabla. Factor ambiente.

El Factor de ambiente resulta:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso} \times \text{Valor asignado})$$

$$EF = 1.4 - 0.03 \times \Sigma (0+1+3+2+.2+10+0-3)$$

$$EF = 1.4 - (0.03 \times 15)$$

$$EF = 1.4 - 0.45$$

$$EF = 0.95$$

Finalmente, los Puntos de Casos de Uso ajustados resultan:

$$UCP = UUCP \times TCF \times EF$$

$$\text{UCP} = 176 \times 0.98 \times 0.95$$

$$\text{UCP} = 163.856$$

Calculo del Esfuerzo

$$\text{E} = \text{UCP} \times \text{CF}$$

Donde:

E: Esfuerzo

UCP: Puntos de Casos de Uso ajustados

CF: Factor de conversión

Para calcular CF

Total EF = Cantidad EF < 3 (entre E1 – E6) + Cantidad EF > 3 (entre E7, E8)

$$\text{Total EF} = 3 + 0$$

$$\text{Total EF} = 3$$

CF = 20 horas-hombre (si Total EF \leq 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

Para este tipo de proyecto 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.

$$\text{E} = \text{UCP} \times \text{CF}$$

$$\text{E} = 163.856 \times 28$$

E = 4587.968 horas hombres.

Se considera que este esfuerzo representa un porcentaje del esfuerzo total del proyecto. Para una estimación más completa de la duración total del proyecto se debe agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Teniendo en cuenta los siguientes valores porcentuales para la distribución del esfuerzo entre las diferentes actividades de un proyecto, que estadísticamente se considera aceptable, se obtiene:

Distribución del Esfuerzo entre las diferentes actividades de un proyecto:

Actividad	Porcentaje	Horas _ Hombre
Análisis	10	1146.992
Diseño	10	1146.992
Implementación	60	6881.952
Pruebas	15	1720.488
Sobrecarga (otras actividades)	5	573.496
Total	100	11469.92

Tabla. Distribución del esfuerzo.

Esfuerzo Total (horas-hombres) = 11469.92 h/h.

Para la etapa de análisis y diseño se requiere de un esfuerzo de **2293.984** horas-hombres, si se considera que trabajan dos hombres, se debe dividir este valor entre dos.

$$2293.984 / 2 = 1146.992 \text{ h/h}$$

Este valor resultante se divide entre 8 horas que son las horas que se trabaja diariamente y se obtiene la cantidad de días trabajados entre 2 hombres:

$$1146.992 / 8 = 143.374 \text{ días/ 2 hombres}$$

A este valor obtenido se divide entre los 6 días laborables que tiene la semana, y se obtiene las semanas trabajadas entre dos hombres:

$$143.374 / 6 = 23.8956 \text{ semanas/ 2 hombres}$$

Este valor se divide entre la cantidad de semanas que tiene un mes para saber cuántos meses se demorarían dos hombres en realizar dicho trabajo.

$$23.8956 / 4 = 5.973916 \text{ meses / 2 hombres.}$$

Si se hace el trabajo entre dos hombres y los dos realizan el mismo esfuerzo entonces se puede decir que el problema analizado puede concluir en 5.9 meses.

Conclusiones

La realización del análisis de costos ha servido para establecer cronogramas y estimaciones razonables que permiten una mejor organización y control de las tareas establecidas durante el desarrollo del sistema. Esta estimación sólo tendría un valor económico si se pudiera multiplicar por el costo de un estudiante en la UCI, por lo que la misma, sólo tiene un valor referencial en cuanto al factor humano de tiempo-trabajo.

Conclusiones Generales

Con la realización del presente trabajo se llegaron a cumplir los siguientes aspectos dándole respuesta a los objetivos trazados:

- Se realizó un análisis exhaustivo de las diferentes metodologías, técnicas, lenguajes y herramientas actuales existentes en el mundo y en nuestro país, el cual nos permitió obtener un conocimiento profundo de las tendencias actuales para desarrollar el sistema concebido de la mejor forma, realizando un software lo más estable y seguro posible.
- La realización del Modelo de Negocio permitió entrar seguidamente al Levantamiento de Requisitos donde se obtuvo todos sus artefactos correspondientes, los cuales permitieron tener una idea más clara del sistema a desarrollar.
- Se realizó el modelo de análisis y diseño del sistema de acuerdo a las necesidades y requerimientos que se capturaron, pasando luego al flujo de trabajo de implementación.
- Como resultado se obtuvo el desarrollo del sistema.

Recomendaciones

Una vez terminada la aplicación se puede constatar que los objetivos trazados al comenzar el trabajo fueron resueltos de manera satisfactoria, aunque se debe tener en cuenta que esta no es más que una primera versión del sistema que puede alcanzar un elevado grado de optimización de ser tratados con profundidad una serie de puntos que se mencionan a continuación, por ello se recomienda:

- Insertar en el sistema funcionalidades que permitan a usuarios determinados subir al sistema manuales de ayuda en cuanto a usabilidad así como documentación relacionada con el tema.
- Extender el uso del sistema en un entorno más amplio dentro de la universidad para contribuir al aumento de la usabilidad en los software desarrollados.
- Incluir en el sistema funcionalidades que permitan realizar al mismo, evaluaciones de manera automática a través de técnicas de inteligencia artificial.

Bibliografía Referenciada

1. Cortés, Alejandro Floría. SID@R. [En línea] [Citado el: 10 de Febrero de 2010.] <http://www.sidar.org/recur/desdi/traduc/es/visitable/quees/usab.htm>.
2. Iazza, Yusef Hassan & Francisco J. Martín Fernández & Ghzala. hipertext.net. [En línea] 2 de mayo de 2004 . [Citado el: 28 de enero de 2010.] <http://www.hipertext.net/web/pag206.htm> .
3. Enrique., Daysel Labañino Griñan. Yusniel Sánchez. Procedimiento para Evaluar la usabilidad de un software sobre Plataforma Web. Ciudad de La Habana : s.n., 2009.
4. Krug, Jakob Nielsen y Steve. Guía para el desarrollo de sitios Web. [En línea] [Citado el: 10 de enero de 2010.] www.guiaweb.gob.cl/.../05/images/fig.001.jpg.
5. Casanovas, Josep. alzado.org. [En línea] 31 de julio de 2004. [Citado el: 10 de enero de 2010.] http://www.alzado.org/comentarios.php?id_art=355.
6. Acosta, María Benítez. Evaluación de la usabilidad. [En línea] 26 de abril de 2007. [Citado el: 10 de febrero de 2010.] <http://evaluausabilidad.sprinterweb.net/index.html>.
7. Acosta, María Benítez. Evaluación de la usabilidad. [En línea] 26 de abril de 2007. [Citado el: 10 de febrero de 2010.] http://evaluausabilidad.sprinterweb.net/evaluacion_objetivos.html.
8. Acosta, María Benítez. Evaluación de la usabilidad. [En línea] 26 de abril de 2007. [Citado el: 10 de febrero de 2010.] http://evaluausabilidad.sprinterweb.net/evaluacion_porque.html.
9. Murugarren, Joaquín Gracia. WebEstilo. [En línea] 15 de febrero de 2003. [Citado el: 10 de febrero de 2010.] <http://www.webestilo.com/guia/articulo.phtml?art=40>.
10. forosdelweb.com. [En línea] 8 de julio de 2009. [Citado el: 20 de enero de 2010.] http://www.forosdelweb.com/wiki/Manual_de_PHP:_Qu%C3%A9_es_PHP.

11. RamSys. jordisan.net. [En línea] 29 de septiembre de 2006. [Citado el: 8 de febrero de 2010.] <http://jordisan.net/blog/2006/que-es-un-framework/>.
12. olgacarreras.blogspot.com. [En línea] febrero de 2007. [Citado el: 12 de enero de 2010.] <http://olgacarreras.blogspot.com/2007/02/la-Usabilidad-como-metodologa-para-el.html>.
13. *revista unam.mx*. Valenzuela, Rebeca. Ciudad Universitaria, México D.F. : s.n., 208.
14. jlopezto. blog.almacenplantillasweb.es. [En línea] [Citado el: 15 de enero de 2010.] <http://blog.almacenplantillasweb.es/2010/01/herramientas-para-evaluar-un-diseno-we/>.
15. bgta.net. [En línea] [Citado el: 10 de febrero de 2010.] <http://www.bgta.net/2008/04/29/10-razones-por-las-que-usar-symfony-en-la-empresa-framework-php/>
16. Programacionphp.net. mailxmail.com. [En línea] 10 de enero de 2006. [Citado el: 11 de febrero de 2010.] <http://www.mailxmail.com/curso-configuracion-apache/que-es-apache-que-alternativas-hay-mercado>.
17. superuser. soporte.tiendalinux.com. [En línea] 31 de Mayo de 2003. [Citado el: 16 de febrero de 2010.] http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
18. misrespuestas.com. [En línea] [Citado el: 16 de febrero de 2010.] <http://www.misrespuestas.com/que-es-un-servidor-web.html>.
19. blog.unijimpe.net/. [En línea] 6 de Mayo de 2008. [Citado el: 16 de Febrero de 2010.] <http://blog.unijimpe.net/yslow-plugin-para-firebug/>.
20. Juanguis. incubaweb.com. [En línea] 4 de Junio de 2007. [Citado el: 16 de Febrero de 2010.] <http://www.incubaweb.com/clickdensity-genera-mapas-de-calor-en-tu-web/>.
21. desarrolloweb.com. [En línea] 23 de octubre de 2008. [Citado el: 16 de febrero de 2010.] http://www.desarrolloweb.com/de_interes/click-tale-analiza-comportamiento-visitantes-web-868.html.

22. Duarte, Aingeru. analiticaweb.es. [En línea] 19 de enero de 2010. [Citado el: 16 de febrero de 2010.] <http://www.analiticaweb.es/clicktale-la-importancia-del-tiempo/>.
23. rational.com. [En línea] [Citado el: 12 de febrero de 2010.] <http://www.rational.com.ar/herramientas/roseenterprise.html> .
24. slideshare.net. [En línea] [Citado el: 12 de febrero de 2010.] http://www.slideshare.net/vivi_jocadi/rational-rose .
25. freedownloadmanager.org. [En línea] 5 de marzo de 2007. [Citado el: 12 de febrero de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).