

**Universidad de las Ciencias Informáticas
Facultad 1**



Automatización del Proceso de
Evaluación de la Calidad de un Producto Software

*Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas*

Autor(es): Anabeisy Pérez González
Lisbet Suárez Lissabet

Tutora: Ing. Yayneris Zambrana Hernández

Ciudad de La Habana, Junio 2011

Año 53 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo titulado: **Automatización del Proceso de Evaluación de la Calidad de un Producto Software** y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Lisbet Suárez Lissabet
(Firma de la autora)

Anabeisy Pérez González
(Firma de la autora)

Ing. Yayneris Zambrana Hernández
(Firma de la tutora)

DATOS DE CONTACTO

Yayneris Zambrana Hernández: Ingeniero en Ciencias Informáticas.

Correo electrónico: yzambrana@uci.cu

AGRADECIMIENTOS

A nuestra tutora Ing. Yanyneris Zambrana Hernández por habernos guiado en la investigación del trabajo, por el tiempo que le dedicó a la tesis y por ayudarnos cuando más la necesitábamos.

Anabeisy Pérez González

Quiero agradecer ante todo a mis padres, por su amor, confianza y apoyo incondicional en la realización de mis sueños, principalmente en estos cinco años de carrera. Se me hace difícil hablar de tanto en tan pocas palabras pero les estaré agradecida toda mi vida. A mi hermano, por creer en mí y seguir mis pasos.

A mi novio Frank, por haberme dado todo su cariño desde el primer día y de quien he aprendido el verdadero significado de la palabra “amor”. Por su apoyo en toda mi carrera, sin él no hubiera sido posible cumplir mis sueños. Te quiero con todas las fuerzas de mi corazón.

A mis abuelos, por enseñarme el significado de la vida y por guiarme por el camino correcto.

A Zory y Arnaldo por ser como mis segundos padres, quienes sin reparos me acogieron y me dieron todo su cariño. A Hildeliza y Segismundo por ser tan especiales en mi vida. A Yani, Martha, Roberto y Daniel que aunque estén lejos siempre me han apoyado y querido como un miembro más de la familia. A mi tía Aurora por estar siempre pendiente de mí como una madre. A José Ramón, Juana y Yanet (la niña) por ayudarme en el transcurso de mi carrera y siempre estar cuando los necesité. De forma general a toda mi familia que no ha tenido reparos en ayudarme a lograr alcanzar mis metas.

A mis amistades y todas las personas que he tenido el placer de conocer en esta maravillosa etapa de mi vida, especialmente a Yenisey, Alejandro, Reinier, Marisdelis, Yicenia, José Miguel, Yumi, Addiel, Lázaro, Zuzana, Javier Alejandro, Alain, Yoan Carlos, Giraldo David, Zenia, Ramón Alberto, Mayte, Antonio, Ludmary, Indira, Hazzel, Daimara, Guillermo, Yadira, Luis Orlando, Mairelys, Enmanuel, Ediagnis, Dreifys, Yoana.

A mi compañera de tesis, por apoyarme en todo y ayudarme en la realización de este trabajo.

AGRADECIMIENTOS

Lisbet Suárez Lissabet

Quiero agradecerles a las personas más lindas del mundo, que durante todos estos años me han dado todo su amor y me han convertido en la persona que soy, mis abuelos Hiran y Zenaida. A mis padres queridos Elizabeth y Armando por estar siempre cuando los necesité y por apoyarme en los momentos más difíciles.

A mis hermanos Lianet, Amanda y Pupito por permitirme ser una guía para ellos y por ser los mejores hermanos del mundo.

A mi tía Walkirita, Juan, y Piti por quererme tanto y darme su apoyo incondicional. A mi tía Ana, mi tío Oscar y mis primos Yúnior y Yaima por acogerme estos 5 años en su hogar, por su cariño, su apoyo y hacerme sentir uno más de su familia.

A mi novio Yadier por saberme comprender, por su apoyo, sobre todo su amor y que a pesar de la distancia siempre me está apoyando y dándome ánimo para salir adelante. A mis suegros que son los suegros más maravillosos del mundo, por permitirme formar parte de su familia y por tratarme como a una hija. A mi cuñada Yami y su esposo Alban por su cariño y sobre todo sus buenos consejos.

A Rosa, Edistio y Yoel por sus buenos consejos, su preocupación y todas las cosas buenas que he aprendido de ellos. A mis amistades Ediagnis, Dairo, Eider, Harry, Ramon, Nayla, que siempre los recordaré con cariño.

DEDICATORIA

Dedico este Trabajo de Diploma a las personas que siempre han estado sin fallo alguno en los buenos y malos momentos, quienes siempre sintieron un gran orgullo de que estudiara en la UCI y con muchas ansias de verme graduada: mis padres.

A mi hermano, por ser lo mejor que tengo en mi vida y por darme su apoyo incondicional.

A mi familia, por ser mi ejemplo a seguir.

A mi novio, por ser mi apoyo, mi guía y mi verdadero amor.

A la familia de mi novio, por ser las personas más maravillosas que he conocido en mi vida.

Anabeisy Pérez González

Le dedico mi tesis a mis abuelos, por estar siempre a mi lado cuando más lo necesitaba y darme fuerzas de seguir adelante.

A mis padres porque sin ellos no estuviera hoy aquí y por darme todo el amor que un padre y una madre pueden darle a una hija.

A mis hermanos que espero ser un ejemplo para ellos.

A Yadier por ser mi ángel de la guarda.

Lisbet Suárez Lisabet

RESUMEN

El trabajo que se presenta es una propuesta de un sistema para la automatización del Proceso de evaluación de la calidad de un producto software durante el proceso de pruebas. Se brinda la información necesaria sobre el proceso que se quiere automatizar, así como de los temas relacionados con la Calidad de Software y las Normas de Calidad. Se realiza un profundo análisis sobre las metodologías de desarrollo, las herramientas y técnicas para el diseño de la aplicación, lenguajes de programación, así como los sistemas de gestión de contenido con el objetivo de determinar el más conveniente para el desarrollo del sistema. Se detallan cada una de las características del sistema, y se muestran cada uno de los elementos y artefactos a medida que va pasando cada una de las fases definidas por la metodología de desarrollo. La propuesta de solución puede ser aplicada a los proyectos productivos que se desarrollan en la Universidad de las Ciencias Informáticas (UCI) con el objetivo de hacer más fácil la evaluación de la calidad del producto software.

PALABRAS CLAVE

Calidad de software, Proceso de evaluación, Métricas.

ÍNDICE

INTRODUCCIÓN	1
1.1 Conceptos Fundamentales	5
1.1.1 Calidad de software	5
1.1.2 Normas de Calidad	6
1.2 Proceso de Evaluación de un Producto Software	6
1.2.1 Proceso de Pre-evaluación.....	7
1.2.2 Proceso de Medición	7
1.2.3 Proceso de análisis de los resultados de la medición	8
1.3 Soluciones existentes	9
1.4 Metodología de Desarrollo de Software	10
1.4.1 Feature Driven Development (FDD).....	10
1.4.2 Extreme Programing (XP).....	11
1.4.3 Rational Unified Process (RUP).....	12
1.5 Herramientas de diseño para la aplicación	14
1.5.1 Rational Rose	14
1.5.2 Visual Paradigm.....	15
1.5.3 Zend Studio	15
1.5.4 Servidor Web Apache.....	16
1.6 Aplicaciones Web para el Desarrollo del Software	16
1.7 Sistemas de Gestión de Contenidos	17
1.7.1 Joomla.....	17
1.7.2 Drupal.....	18
1.8 Lenguajes de programación. Definición y Características	19
1.8.1 PHP	20
1.8.2 HTML.....	21
1.8.3 JavaScript.....	21
1.9 Gestor de Base de Datos. Definición y Características	22
1.9.1 MySQL	22

1.9.2	PostgresSQL	23
1.10	Conclusiones del capítulo.....	23
CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA		25
2.1	Descripción de los procesos del negocio	25
2.2	Descripción del sistema a desarrollar	26
2.3	Requisitos no funcionales del sistema.....	27
2.4	Fase de Exploración.....	29
2.5	Historias de Usuario. Definición y Descripción.....	29
2.6	Fase de Planificación	32
2.6.1	Estimación del esfuerzo por Historia de Usuario	33
2.6.2	Planificación de las Iteraciones	34
2.6.3	Duración de las Iteraciones.....	35
2.6.3.1	Plan de Entrega	36
2.7	Diseño del sistema	36
2.7.1	Estructura de Drupal. Módulos. Patrones de diseño y Flujo de información	37
2.8	Diagrama de despliegue	42
2.9	Conclusiones del capítulo	43
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA		44
3.1	Fase de Implementación	44
3.1.1	Tareas de las HU implementadas en la Iteración 1	44
3.1.2	Tareas de las HU implementadas en la Iteración 2	50
3.1.3	Tareas de las HU implementadas en la Iteración 3	53
3.2	Fase de Pruebas.....	58
3.2.1	Tipos de Pruebas	59
3.2.1.1	Pruebas unitarias	59
3.2.1.2	Pruebas aceptación.....	59
3.3	Conclusiones del capítulo	61
CONCLUSIONES		62

RECOMENDACIONES.....	63
REFERENCIAS BIBLIOGRÁFICA.....	64
BIBLIOGRAFÍA.....	65
ANEXOS.....	68
GLOSARIO DE TÉRMINOS Y DEFINICIONES	92

ÍNDICE DE TABLAS

Tabla 1: Diferencia entre metodología ágil y no ágil. (8)	12
Tabla 2: Historia de usuario "Autenticar usuario".	30
Tabla 3: Historia de usuario "Gestionar usuario".	30
Tabla 4: Historia de usuario "Gestionar proyecto".	31
Tabla 5: Historia de usuario "Realizar encuesta".	31
Tabla 6: Historia de usuario "Procesar encuesta".	32
Tabla 7: Plan de estimación de esfuerzo por historia de usuario.	33
Tabla 8: Plan de iteraciones.	35
Tabla 9: Plan de entrega.	36
Tabla 10: HU Implementada en la Iteración 1.	44
Tabla 11: Tarea 1 "Autenticar Usuario".	45
Tabla 12: Tarea 2 "Autenticar Usuario".	45
Tabla 13: Tarea 3 "Autenticar Usuario".	45
Tabla 14: Tarea 4 "Autenticar Usuario".	46
Tabla 15: Tarea 5 "Autenticar Usuario".	46
Tabla 16: Tarea 1 "Gestionar Usuario".	47
Tabla 17: Tarea 2 "Gestionar Usuario".	47
Tabla 18: Tarea 3 "Gestionar Usuario".	48
Tabla 19: Tarea 1 "Gestionar Proyecto".	48
Tabla 20: Tarea 2 "Gestionar Proyecto".	49
Tabla 21: Tarea 1 "Realizar encuesta".	49
Tabla 22: Tarea 2 "Realizar encuesta".	49
Tabla 23: Tarea 1 "Procesar encuesta".	50
Tabla 24: HU Implementada en la Iteración 2.	50
Tabla 25: Tarea 1 "Mostrar resultado de la encuesta".	51
Tabla 26: Tarea 1 "Mostrar métricas".	51
Tabla 27: Tarea 1 "Calcular métricas".	52
Tabla 28: Tarea 1 "Generar resultado de las métricas".	52
Tabla 29: Tarea 1 "Mostrar resultado de las métricas".	53
Tabla 30: HU Implementada en la Iteración 3.	53
Tabla 31: Tarea 1 "Mostrar puntuación asignada a las sub-características".	54
Tabla 32: Tarea 2 "Mostrar puntuación asignada a las sub-características".	54
Tabla 33: Tarea 1 "Mostrar el cálculo de la puntuación de las características".	55
Tabla 34: Tarea 2 "Mostrar el cálculo de la puntuación de las características".	55
Tabla 35: Tarea 1 "Mostrar grado de conformidad".	55
Tabla 36: Tarea 2 "Mostrar grado de conformidad".	56
Tabla 37: Tarea 1 "Mostrar criterio de evaluación".	56
Tabla 38: Tarea 2 "Mostrar criterio de evaluación".	57

Tabla 39: Tarea 1 “Gestionar informe de las no conformidades”	57
Tabla 40: Tarea 1 “Exportar a pdf”	58
Tabla 41: No conformidades por casos de pruebas.....	60

ÍNDICE DE FIGURAS

Figura 1: Proceso de Evaluación.....	6
Figura 2: Proceso de Pre-Evaluación de un producto software	7
Figura 3: Proceso de Medición de un producto software	8
Figura 4: Proceso de Análisis de los resultados de la medición	9
Figura 5: Diagrama de Clases del CMS Drupal	38
Figura 6: Diagrama de Despliegue.....	43

INTRODUCCIÓN

Actualmente el desarrollo de software es una actividad comercial con un alto valor en todas las esferas de la sociedad. Internacionalmente son muchas las empresas que se dedican a desarrollar y exportar software cada vez con una mayor y mejor calidad, lo que hace que el mercado sea más competitivo y exigente con los productos que se comercializan, de ahí la importancia de la Calidad en los productos software.

Cuba en su posición de país subdesarrollado y bloqueado económicamente, no se ha limitado a adentrarse en el mundo del desarrollo de software para la informatización de la sociedad cubana y para exportar a otros países, entre los que se encuentra Venezuela. El país cuenta con la Universidad de las Ciencias Informáticas (UCI), una Institución que ha logrado mucho en pocos años de creada.

La UCI es una institución educacional que además tiene varios centros de desarrollo de software especializados en distintas líneas temáticas, como por ejemplo el Centro de Identificación y Seguridad Digital (CISED) que se dedica a brindar productos, servicios y soluciones integrales en el campo de la identificación y la seguridad digital. Cada uno de estos centros se dedica a desarrollar proyectos a la medida y para un cliente específico. En estos proyectos productivos se tiene en cuenta la Gestión de la Calidad de los mismos, siempre teniendo como objetivo final cumplir con las necesidades y expectativas del cliente.

La Gestión de la Calidad es una de las áreas de conocimientos que incluye la Gestión de Proyectos Informáticos, esta área describe los procesos necesarios para asegurarse de que el proyecto cumpla con los objetivos por los cuales ha sido emprendido. La misma requiere de especial atención en el desarrollo de software, ya que un producto sin calidad no se vende. Durante el desarrollo de un producto software es necesario tener en cuenta una serie de elementos claves relacionados con la calidad del mismo como son, las pruebas, las revisiones, el control de cambios, entre otros elementos que son esenciales para obtener resultados satisfactorios.

Es por eso que en el desarrollo de un producto software es necesario aplicarle las pruebas de calidad necesarias al mismo para comprobar que cumpla con las expectativas del cliente. Con este fin muchas empresas se han esforzado en definir estándares, métricas y herramientas para la evaluación del software.

Actualmente la UCI cuenta con un proceso para la evaluación de un producto software durante el proceso de pruebas basado en la Norma ISO/IEC 9126 que define las características de calidad a ser evaluadas en un software, además de proponer las métricas para cada una de estas, y la Norma ISO/IEC 14598 que define el proceso para evaluar un producto software. El proceso de evaluación durante las pruebas del software es de gran importancia para los proyectos productivos del CISED, pero no se lleva a cabo debido a que el proceso se tiene que realizar de forma manual, lo que hace más complicado y engorroso su aplicación. Esto conlleva a un aumento del tiempo planificado para la construcción del producto, provocando un atraso en el cronograma de trabajo e incumplimiento con lo pactado con el cliente. De ahí la necesidad de automatizar los procesos que incluye la evaluación de la calidad del producto software.

A partir de la situación problemática anteriormente planteada, surge la siguiente interrogante: ¿Cómo mejorar la ejecución del proceso de evaluación de la calidad de un producto software en los proyectos productivos?

El **Objeto de estudio** se enmarca en el proceso de evaluación de la calidad de un producto software y se propone como **Campo de Acción** el proceso de medición de la calidad de un producto software.

Se define como **Objetivo general** desarrollar una aplicación informática que permita evaluar la calidad de un producto software.

Entre los **Objetivos Específicos** se encuentran:

- Analizar los referentes teóricos del tema.
- Diseñar los procesos para la evaluación de la calidad de un producto software.
- Implementar los procesos que permitan la evaluación de la calidad.

- Probar el correcto funcionamiento de la aplicación informática a desarrollar.

Para dar cumplimiento a dichos objetivos específicos se proponen las siguientes **Tareas de la investigación:**

1. Realizar un estudio de las herramientas utilizadas para la evaluación de la calidad de un producto software.
2. Investigar las soluciones similares al sistema a desarrollar.
3. Analizar el proceso de evaluación de la calidad de un producto software.
4. Realizar el diseño de la aplicación informática.
5. Realizar la implementación de la aplicación informática.
6. Realizar pruebas funcionales a la aplicación informática desarrollada.

Para guiar la investigación se plantea la siguiente **Idea a Defender:**

Con el desarrollo de un sistema informático para la evaluación de la calidad de un producto software se logrará mejorar la aplicación de dicho proceso durante la ejecución de la pruebas en los proyectos productivos.

Para el desarrollo de la investigación se utilizaron los siguientes métodos:

Teóricos:

- **Histórico-lógico:**

Se utiliza este método cuando se analiza la evolución de las tecnologías y herramientas utilizadas en la evaluación de la calidad de productos software y de esta manera dar solución a las necesidades de la gestión de los procesos de evaluación de la calidad de los mismos en el CISED.

- **Analítico-sintético:**

Al analizar todo lo relacionado al tema específico a abordar se pudo hacer una profundización de todas sus características, dígame objetivos, logros, o deficiencias que se derivan, así como también las mejoras que se pueden proponer para obtener resultados eficientes.

Empíricos:

➤ **Entrevista:**

Se utiliza este método cuando se realizan entrevistas al cliente con el objetivo de obtener la mayor información posible sobre las necesidades del CISED.

Posibles resultados: Se espera aportar al CISED una herramienta informática que automatice los procesos de evaluación de la calidad de un producto software durante el proceso de pruebas, con el objetivo de que se puedan aplicar de manera más rápida, eficiente y sistemática.

El presente trabajo de diploma consta de tres capítulos:

Capítulo 1. Fundamentación teórica

En este capítulo se realiza un estudio referente al estado del arte del tema de investigación mencionado anteriormente, se describen conceptos relacionados con la Calidad de Software, así como una descripción del Proceso de Evaluación de la Calidad. También se realiza un profundo estudio de las metodologías, técnicas, herramientas de diseño y lenguajes de programación posibles a usar para el desarrollo de la propuesta de solución.

Capítulo 2. Características y Diseño del sistema

En este capítulo se desarrollan las dos primeras Fases de la metodología las cuales son Exploración y Planificación. Serán descritas las Historias de Usuario con los Requisitos funcionales y no funcionales del sistema. Además se realizará un análisis del diseño de la aplicación, y en forma general de la estructura de los Sistemas de Gestión de Contenidos (CMS) a usar.

Capítulo 3. Implementación y Prueba

En el capítulo se desarrollan las dos últimas Fases definidas por la metodología usada, Implementación y Prueba. Se describen las tareas implementadas, así como las pruebas necesarias para validar la solución propuesta.

Capítulo 1: Fundamentación Teórica

En el presente capítulo se describirán los aspectos relacionados con el objeto de estudio y el campo de acción, además de definirse las herramientas, tecnologías, lenguaje de programación, y metodología a usar para darle solución a la situación problemática.

1.1 Conceptos Fundamentales

Actualmente la ISO/IEC ha definido una serie de normas para lograr la calidad máxima en los proyectos de software, definiendo un conjunto de normas para alcanzar este objetivo. A continuación, para un mejor entendimiento de estos términos, se definen los conceptos fundamentales asociados a este tema.

1.1.1 Calidad de software

La IEEE en su estándar 610-1900 define la Calidad del Software como:

“Grado con el que un sistema, componente o proceso cumple los requerimientos especificados según las necesidades o expectativas del cliente o usuario”.

Roger Pressman definió la Calidad de Software como:

“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”. (1)

Otra definición de Pressman sobre la Calidad de Software es:

“Concordancia del software producido con los requerimientos explícitamente establecidos con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario” (2)

Según las ideas expuestas anteriormente, se puede concluir que la calidad de un producto software se determina midiendo la exactitud con la que fueron cumplidos los requerimientos, logrando la satisfacción del cliente.

1.1.2 Normas de Calidad

Una norma de calidad es un documento que proporciona para un uso común una serie de reglas, características y directrices con el objetivo de conseguir un alto grado de orden en el contenido de la calidad.

Una norma de calidad es un documento, establecido por consenso y aprobado por un organismo reconocido (nacional o internacional), que proporciona para un uso común y repetido, una serie de reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo de orden en el contexto de la calidad. (3)

1.2 Proceso de Evaluación de un Producto Software

Con la unión de la norma ISO/IEC 9126 y la ISO/IEC 14598 se conformó el proceso de evaluación de un producto software el cual está dividido en tres subprocesos. [Ver Figura 1] (4)

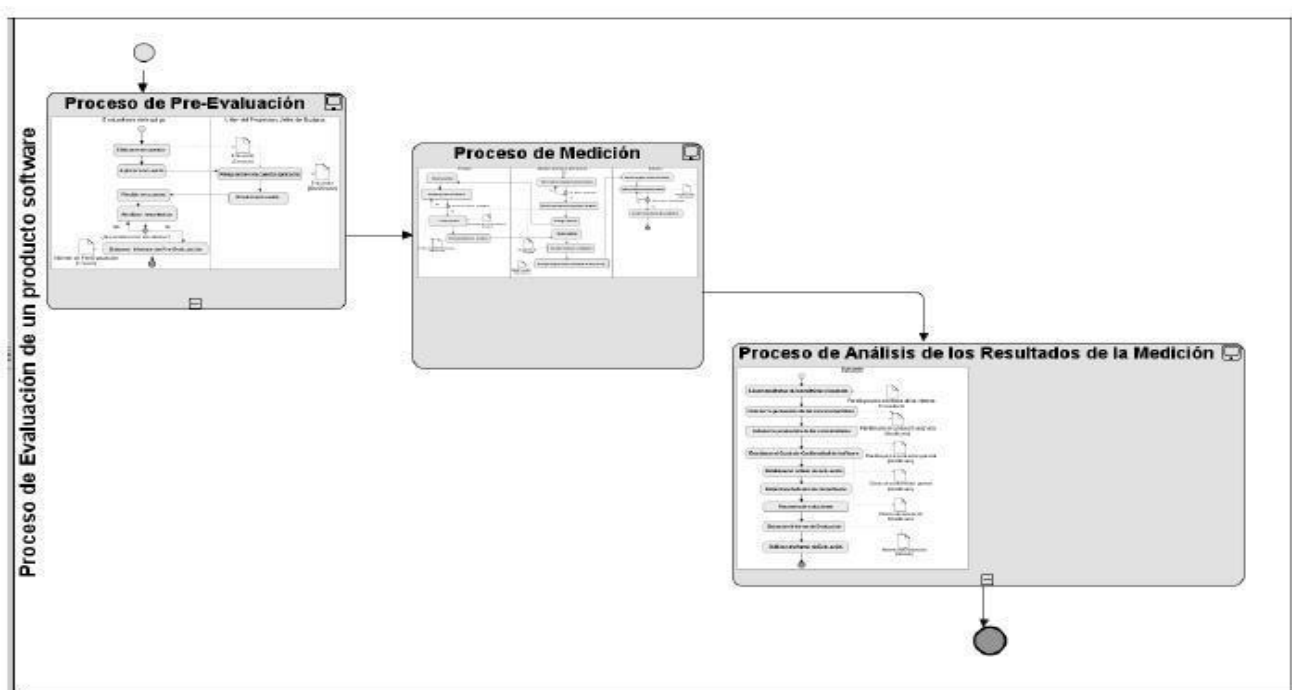


Figura 1: Proceso de Evaluación

1.2.1 Proceso de Pre-evaluación

En este proceso se definen las características y sub-características que se medirán en el software. El mismo consta de la aplicación de una encuesta a los directivos del proyecto y de acuerdo a los resultados, se enfocará todos los esfuerzos en el cumplimiento de las características de mayor peso. [Ver Figura 2]

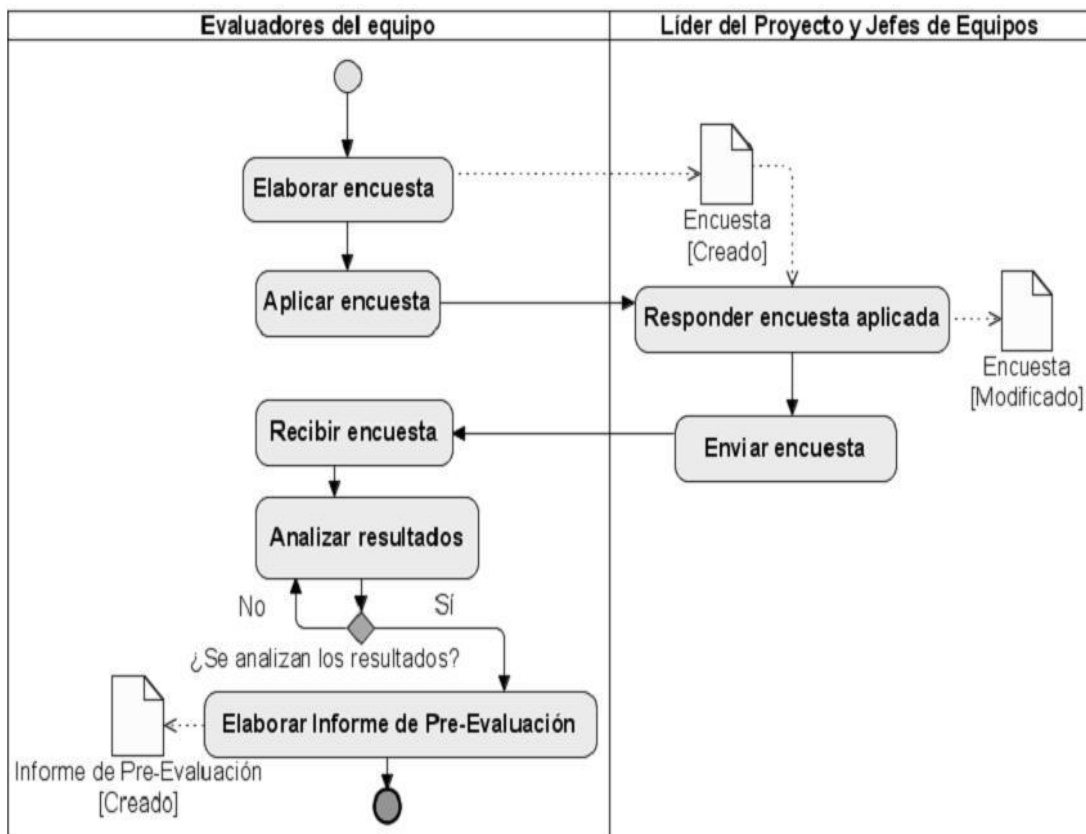


Figura 2: Proceso de Pre-Evaluación de un producto software

1.2.2 Proceso de Medición

En este proceso se evalúa la calidad de un producto software, se obtiene una evaluación cualitativa y cuantitativa de la calidad del producto, permitiendo la mejora del proceso de desarrollo. Este consta de

tres partes fundamentales, la recopilación de datos, cálculo de las métricas seleccionadas, y la evaluación de la calidad. [Ver Figura 3]

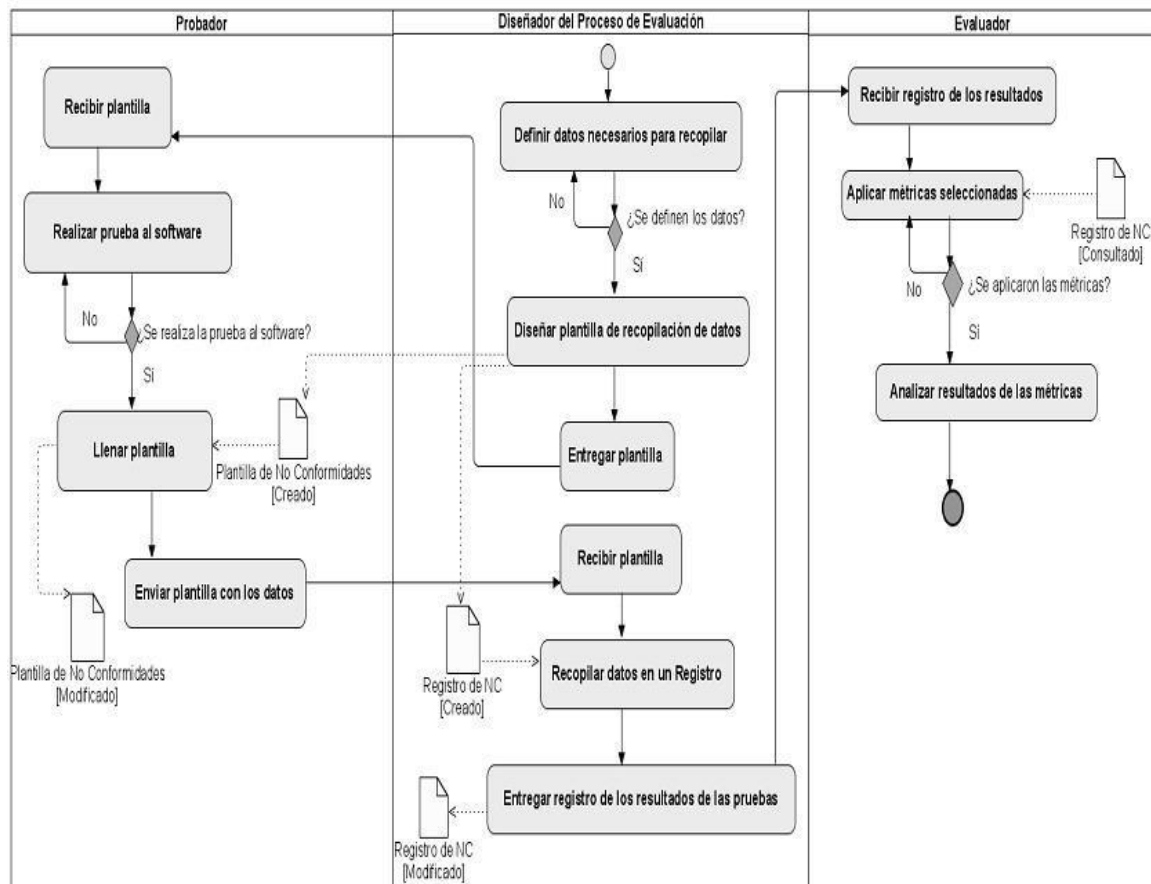


Figura 3: Proceso de Medición de un producto software

1.2.3 Proceso de análisis de los resultados de la medición

El proceso consiste en analizar los resultados obtenidos en el **Proceso de Medición**, se detectan las deficiencias encontradas en el software y además se recomiendan soluciones. [Ver Figura 4]

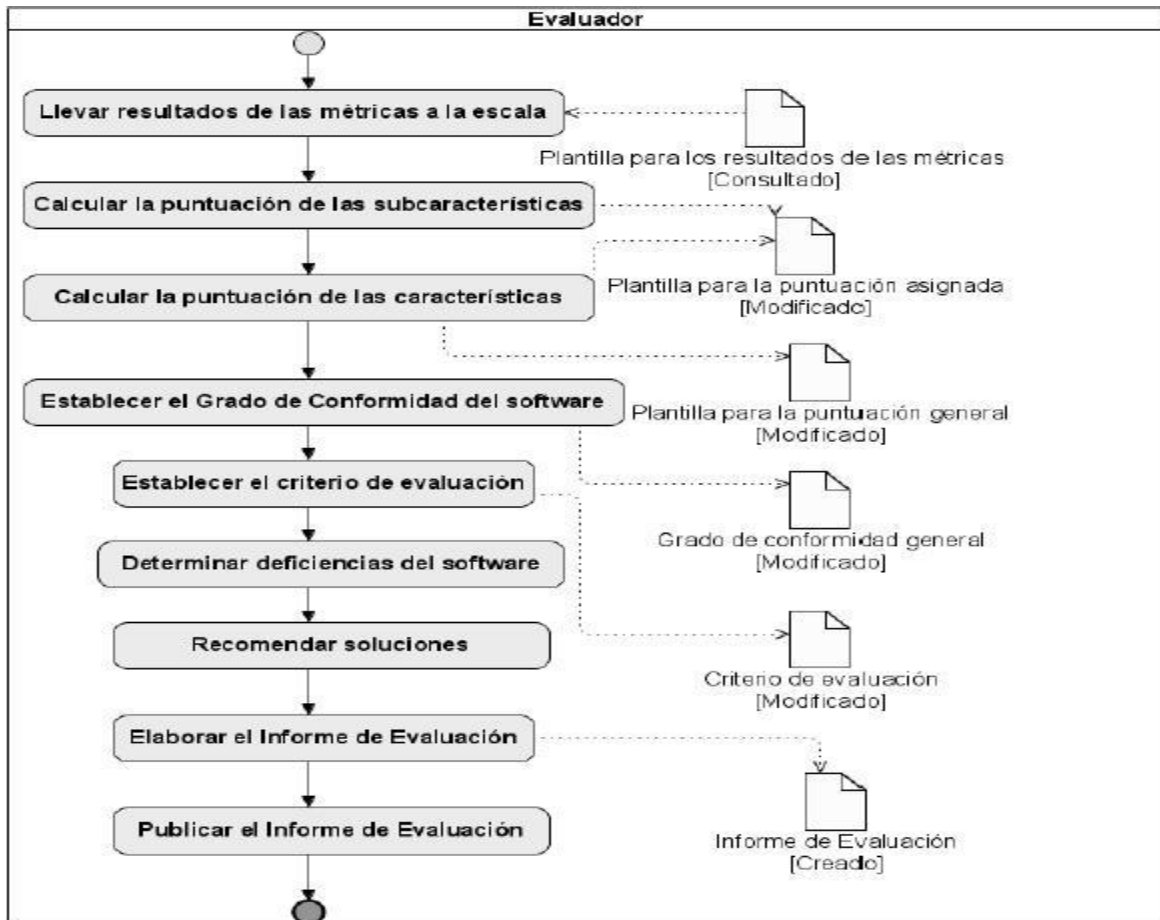


Figura 4: Proceso de Análisis de los resultados de la medición

1.3 Soluciones existentes

El análisis y estudio de sistemas similares resulta de gran importancia y utilidad, ya que pueden mostrar una guía o referencia acerca de los procesos que cumplen con los objetivos trazados. En el presente trabajo se llevó a cabo una extensa investigación sobre sistemas capaces de automatizar el proceso de evaluación de la calidad de un producto software. A pesar de ser encontrados sistemas donde se brinda información sobre la calidad de los productos software, ninguno de estos satisfacen las necesidades fundamentales por la cual surgió esta investigación.

1.4 Metodología de Desarrollo de Software

«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.» (5)

La Ingeniería de software es una disciplina fundamental para lograr la calidad del producto software, ya que aquí es donde se valora cómo construir el software, además de estructurar, planificar y controlar el proceso de desarrollo. Según Pressman en la Ingeniería de Software se pueden identificar: los métodos, procesos y las herramientas a utilizar en el software.

En un proyecto de desarrollo de software la metodología define Quién debe hacer, Qué, Cuándo y Cómo debe hacerlo. La metodología constituye la columna vertebral del proceso de desarrollo de software. No existe una metodología universal. Las características de cada proyecto (equipo de desarrollo, recursos, entre otros.) exigen que el proceso sea configurable.

Para dirigir el proceso de desarrollo se realizará un estudio de las principales metodologías utilizadas en la Universidad: como metodología ágil FDD y XP, y como metodología tradicional RUP.

1.4.1 Feature Driven Development (FDD)

La metodología FDD es un proceso ágil para el desarrollo de sistemas, basado en la calidad del software, que incluye un monitoreo constante del proyecto. Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado. Define claramente entregas tangibles y formas de evaluación del progreso del proyecto. (6)

El equipo de trabajo está estructurado en jerarquías, siempre debe haber un jefe de proyecto, y aunque es un proceso considerado ligero también incluye documentación (la mínima necesaria para que algún nuevo integrante pueda entender el desarrollo de inmediato).

Esta metodología está pensada para proyectos relativamente cortos, con etapas de cierre (Iteraciones) cada dos semanas que producen un software funcional que puede ser visto, probado y monitorizado por el cliente. Estas iteraciones son decididas en base a las funcionalidades que el software debe tener, funcionalidades definidas por el cliente.

La metodología FDD cuenta con cinco fases durante las cuales se diseña y se construye el sistema:

- Desarrollo de un modelo global.
- Construcción de una lista de funcionalidades.
- Planeación por funcionalidad.
- Diseño por funcionalidad.
- Construcción por funcionalidad.

1.4.2 Extreme Programing (XP)

La metodología XP es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo, equipo pequeño y cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. El objetivo fundamental es la satisfacción del cliente. XP define cuatro fases fundamentales: Exploración, Planificación, Implementación y Pruebas.

Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

El cliente como miembro fundamental del equipo es el encargado de decidir qué se va a implementar, es el único autorizado en añadir, cambiar o quitar requerimientos en cualquier momento, por lo que debe estar al tanto del estado real y el progreso del proyecto. (7)

1.4.3 Rational Unified Process (RUP)

La metodología RUP es una metodología tradicional que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. El proceso de desarrollo de software se organiza a través de nueve flujos de trabajo (los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo) organizados en cuatro fases de desarrollo: inicio, elaboración, construcción y transición.

¿Qué metodología usar?

Tabla 1: Diferencia entre metodología ágil y no ágil. (8)

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

Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.
--	---

La tabla recoge las principales diferencias entre las metodologías ágiles con respecto a las metodologías tradicionales (robustas). Tomando en cuenta los resultados de la tabla anterior, se puede decir que las metodologías ágiles son las que más se adaptan al entorno del proyecto por tanto RUP queda descartada de nuestra propuesta.

Las metodologías ágiles presentan diversas ventajas, entre las que se destacan:

- Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.
- Entrega continua y en plazos breves de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo.
- Importancia de la simplicidad, eliminado el trabajo innecesario.
- Atención continua a la excelencia técnica y al buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo.

De acuerdo a las características que presenta el proyecto a desarrollar, las cuales se mencionan a continuación:

- El proyecto es pequeño.
- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- Se definen pocos roles.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El cliente o el usuario se convierte en miembro del equipo.
- Propiedad colectiva del código.
- Comunicación de los programadores a través del código.

Se puede concluir que la metodología que se adapta completamente a las características de la solución que se desea implementar es XP.

1.5 Herramientas de diseño para la aplicación

Las herramientas CASE (*Computer Aided Software Engineering*, en español Ingeniería de Software Asistida por Ordenador) brindan componentes que incluyen todos o la mayoría de los requisitos necesarios para el desarrollo de sistemas, han sido creadas con gran exactitud entorno a las necesidades del desarrollo del software para la automatización del proceso incluyendo análisis, diseño e implementación. Las herramientas CASE cuentan con una credibilidad y exactitud que tienen un reconocimiento universal, siendo usadas por cualquier desarrollador y/o programador que busca un resultado óptimo y eficiente. (9)

Entre los beneficios más significativos de las herramientas CASE se encuentran los siguientes:

- Facilidad para la revisión de aplicaciones.
- Soporte para el desarrollo de prototipos de sistemas.
- Generación de código.
- Mejora en la habilidad para satisfacer los requerimientos del usuario.
- Soporte interactivo para el proceso de desarrollo.

Las herramientas CASE se han venido ampliando y desarrollando, algunos de sus ejemplos son: *Microsoft Project, Rational Rose, JDeveloper, MagicDraw, Visual Paradigm, Microsoft Visio, Enterprise Architect*, entre otros.

1.5.1 Rational Rose

Es una herramienta que permite el modelado visual mediante el Lenguaje Unificado de Modelado (UML). Facilita los procesos arquitectónicos y de diseño de cualquier producto software y soporta la generación de código para lenguajes como Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++ y Visual Basic. Dentro de sus características más importantes se destacan el control por separado de los componentes para el modelado, lo que permite una administración mejor estructurada. También es capaz de analizar la calidad del código y crear definiciones de tipo de documento XML para el uso en la aplicación, además de integrarse con varios sistemas de control de versiones. (9)

1.5.2 Visual Paradigm

Es una herramienta CASE que utiliza el Lenguaje Unificado de Modelado (UML) y permite la ingeniería directa o inversa. Es capaz de trabajar de forma colaborativa dando la posibilidad de que varios usuarios trabajen sobre el mismo proyecto y generar documentación en formatos como web o pdf. Contiene un generador de mapeo objeto-relacional para varios lenguajes como Java, PHP y .Net. Otra característica importante es su capacidad de integración con algunas herramientas Java como Eclipse/IBM WebSphere, JBuilder, NetBeans IDE, Oracle JDeveloper y BEA Weblogic. Además tiene funcionalidades para la generación y soporte de código en varios lenguajes de los cuales se puede citar Java, C + +, CORBA IDL, PHP, XML Schema, Ada, Python, C #, VB. NET, Object Definition Language (ODL), Flash Action Script, Delphi, Perl, C-Objetivo, y Ruby. Las últimas versiones de Visual Paradigm pueden brindar respuestas rápidas con el uso de poca memoria utilizando de manera moderada los tiempos del procesador, lo que permite el procesamiento de grandes y complicadas estructuras de un proyecto de manera eficiente. (10)

1.5.3 Zend Studio

Es el único entorno de desarrollo integrado IDE (*Integrated Development Environment*) disponible para desarrolladores profesionales que ofrecen las capacidades necesarias para desarrollar aplicaciones de negocio. Características como la refactorización, la generación de código, el código de ayudar y análisis semántico se combinan para permitir el desarrollo rápido de aplicaciones tanto del lado del servidor (en PHP) como del lado del navegador (en Java Script). Incluye todos los componentes necesarios durante el ciclo de vida de una aplicación en PHP y es un IDE propietario, compatible con las plataformas Linux, MAC y Windows.

Entre sus principales características se encuentra un excelente completamiento de código, múltiples lenguajes, incorpora el Framework de Zend, PHP Documento y manual de PHP. Contiene soporte para Web Services, PHP4, PHP5 y SQL. Agiliza el trabajo, cuenta con un buen depurador, además de infinitas opciones que permiten un desarrollo profesional de nuestras aplicaciones. (11)

1.5.4 Servidor Web Apache

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Es una tecnología gratuita de código fuente abierto y altamente configurable de diseño modular. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. Permite la creación de ficheros de log lo que ayuda a tener un mayor control en el servidor.

(12)

1.6 Aplicaciones Web para el Desarrollo del Software

Las aplicaciones web son sistemas informáticos que utilizan los usuarios para acceder a un servidor web mediante intranet o internet. En la actualidad se han convertido en sistemas muy usados, debido a la práctica del navegador web como cliente ligero, y la posibilidad de brindar mantenimiento a las aplicaciones, sin tener que distribuir ni instalar software en las computadoras de los usuarios.

Existen dos grupos de lenguaje de programación para la web, un grupo es llamado lenguajes del lado del Cliente y el otro grupo lenguajes del lado del Servidor.

La modalidad o arquitectura Cliente/Servidor es aquella en la que coinciden una serie de aplicaciones basadas en dos categorías que cumplen diferentes funciones (una requiere servicios y la otra los brinda) las cuales pueden realizar tanto actividades conjunta como independientes. Esas dos categorías son justamente Cliente y Servidor.

Ventajas de las aplicaciones web

Además de posibilitar el acceso de varios usuarios al sistema simultáneamente, estas poseen un gran número de ventajas que se muestran a continuación:

- Las aplicaciones web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo.
- No necesitan ser descargadas, instaladas ni configuradas. Solamente el usuario debe acceder a su cuenta online y las mismas estarán listas para realizar el trabajo sin importar cuál es su configuración o su hardware.
- Con las aplicaciones web el usuario podrá acceder a la información desde cualquier parte del mundo.

1.7 Sistemas de Gestión de Contenidos

El sistema de gestión de contenido (*Content Management Systems* o CMS) es un software que se utiliza principalmente para facilitar la gestión de webs, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido web (*Web Content Management* o WCM).

Entre las características más importantes de los sistemas de gestión de contenidos y que constituyen una evolución de los sistemas de información tradicionales, está el hecho de que sean manejables por todos los usuarios, todos pueden subir información, manipularla, representarla, entre otros. Permiten distribuir funciones a sus usuarios, así siempre existirán individuos encargados de comprobar qué se incluye y quién lo incluye, con la opción de rechazar o solicitar la modificación del material.

Existen diferentes tipos de CMS como son: Zope, MODx, OpenCMS, TYPO3, Apache lenya, Joomla, Drupal.

1.7.1 Joomla

Es un sistema automatizado de noticias basado en la web y sistema de gestión de contenido basado en tecnologías PHP y MySQL. Es *Open Source* con licencia GPL. Existe una comunidad activa de desarrolladores que constantemente están trabajando para mejorar sus características, su escalabilidad y facilidad de uso del software. Destaca la calidad de su código, el respeto por los estándares web, un

énfasis especial en la usabilidad y consistencia de todo el sistema. Este administrador de contenidos puede trabajar en Internet o Intranet y requiere de una base de datos MySQL, preferiblemente de un servidor HTTP Apache. (13)

1.7.2 Drupal

Drupal es un paquete de software gratuito que permite que cualquiera pueda publicar, administrar y organizar una amplia variedad de contenido en un sitio web. El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. Su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar diferentes tipos de sitios web.

Drupal cuenta con un conjunto de características que hacen de este CMS uno de los más sencillos y más fácil de usar, dichas características se mencionan a continuación:

Ayuda on-line: Contiene un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.

Búsqueda: Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.

Código abierto: El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.

Módulos: La comunidad de Drupal ha contribuido con infinidad de módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados y bookmarks, entre otros.

Personalización: Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo con las preferencias definidas por el usuario.

URLs amigables: Drupal usa el *mod_rewrite* de Apache para crear *URLs* que son manejables por los usuarios y los motores de búsqueda.

Plataforma Independiente de la base de datos: Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

Multiplataforma: Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.

Por ser un CMS distribuido bajo licencias libres (Drupal) y por las características antes expuestas, se decide escoger para el desarrollo de la propuesta de solución. (14)

1.8 Lenguajes de programación. Definición y Características

Como consecuencia de la selección del CMS Drupal para lograr el desarrollo de la aplicación surge la necesidad de utilizar PHP como lenguaje de programación ya que es un lenguaje implícito en el funcionamiento del CMS.

Dentro de los lenguajes del lado del Cliente se encuentran: JavaScript, HTML, Java, Visual Basic Script y dentro de los lenguajes del lado del Servidor se destacan: PERL, ASP, JSP, PHP.

1.8.1 PHP

PHP (PHP: *Hypertext Preprocessor*) es un lenguaje de script interpretado del lado del servidor utilizado para la generación de páginas Web dinámicas. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.

Al ser un lenguaje libre, dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, SybasemSQL, Informix, etc.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- PHP es soportado por una gran comunidad de desarrolladores, como producto de código abierto, goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
- Su rapidez, su facilidad de aprendizaje, su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos. (15)

1.8.2 HTML

El (*Hyper Text MarkupLanguage*) Lenguaje de Marcas de Hipertexto es el lenguaje con el que se desarrollan las páginas web. Permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento, es decir, es un lenguaje de hipertexto. Un documento hipertexto no sólo se compone de texto, sino que también contiene imágenes, sonido, vídeos y otros, por lo que el resultado puede considerarse como un documento multimedia. Los documentos HTML deben tener la extensión html o htm, para que puedan ser visualizados en los navegadores. Los navegadores se encargan de interpretar el código HTML de los documentos, y de mostrar a los usuarios las páginas web resultantes del código interpretado. (16)

1.8.3 JavaScript

JavaScript no es un lenguaje orientado a objetos ya que no dispone de Herencia y Polimorfismo, es más bien un lenguaje basado en prototipos. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con Java script se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Java script y ejecutarlas de modo que el mayor recurso con que cuenta este lenguaje es el propio navegador. Permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, entre otros. Entre algunas de las acciones que se pueden realizar con java script se tienen dos vertientes. Por un lado permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que facilita crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo y por el otro los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. (17)

Ventajas

- Lenguaje de scripting seguro y fiable.

- Los script tienen capacidades limitadas, por razones de seguridad.
- El código Java script se ejecuta en el cliente.

Desventajas

- Código visible por cualquier usuario.
- El código debe descargarse completamente.

1.9 Gestor de Base de Datos. Definición y Características

Es un conjunto de programas que permiten el almacenamiento de información, velando porque se pueda disponer de ella en cualquier momento, que la información sea correcta desde un punto de vista lógico y que sólo pueda ser accedida por las personas adecuadas. (18)

Los Sistemas Gestor de Bases de Datos (SGBD) ayudan a realizar las siguientes acciones:

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos.

Los SGBD que soporta el CMS Drupal son MySQL y PostgreSQL.

1.9.1 MySQL

El servidor de Base de Datos MySQL es de código abierto. Se distribuye en dos versiones, una comercial, de pago y que incluye soporte, y otra gratuita, basada en el soporte de la comunidad. Uno de los puntos fuertes de MySQL es su facilidad de uso y la documentación existente. Además de una potente interfaz en modo consola, cuenta con diversas herramientas de administración, siendo tres los más populares: MySQLAdministrator, phpMyAdmin y el módulo de administración de MySQL para Webmin. Es multiplataforma, se puede instalar igualmente sobre Windows que sobre GNU/Linux. Utiliza

SQL (del inglés Structured Query Language, Lenguaje de consulta estructurado), el lenguaje estándar para la consulta de bases de datos utilizado en todo el mundo. (19)

1.9.2 PostgreSQL

Es un sistema de gestión de Base de Datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution*) y con su código fuente disponible libremente. La licencia de BSD es una licencia de software otorgada principalmente para los sistemas BSD2. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Es el sistema de gestión de base de datos de código abierto más potente del mercado. Las características que se tuvieron en cuenta durante su desarrollo fueron estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows. Contiene una documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios y es altamente adaptable a las necesidades del cliente. (20)

Por estas razones se decide escoger este gestor de base de datos para el desarrollo de la propuesta de solución.

1.10 Conclusiones del capítulo

Después de haber concluido el análisis bibliográfico de la investigación se ha llegado a la conclusión de que el proceso de evaluación representa un gran beneficio para los proyectos productivos del CISED ya que de cierto modo agiliza el proceso de pruebas del producto software y permite emitir un criterio de evaluación y un grado de conformidad, una vez que se termine el proceso de pruebas del software.

Se abordó sobre los diferentes subprocesos que componen el Proceso de Evaluación que se desea automatizar, lo que permitió conocer a profundidad cada una de las actividades definidas en el proceso para una posterior implementación.



Se realizó un profundo análisis de las diferentes metodologías de desarrollo de software, las herramientas posibles a usar para el desarrollo de la propuesta de solución y el lenguaje de programación que facilitaría la implementación del sistema.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

En el presente capítulo se definirán las características del sistema que se desea desarrollar, así como los Requisitos con los que debe cumplir para que cuente con las exigencias del cliente. Además, el capítulo se desarrollará teniendo en cuenta las fases de desarrollo definidas por la metodología que se decidió utilizar y según la fase en que se encuentre se realizarán las actividades correspondientes, y se generarán los artefactos necesarios para lograr los objetivos trazados. También se realizará un análisis de los patrones de diseño que utiliza Drupal y el funcionamiento del mismo.

2.1 Descripción de los procesos del negocio

La existencia de un Proceso de Evaluación de un producto software en la UCI, es poco conocido ya que para implementar el mismo en un proyecto productivo se hace bastante engorroso, debido a que se deben manejar y procesar una serie de datos que para realizarlo manualmente es complejo.

De automatizarse este proceso le permitiría a muchos proyectos productivos de la UCI conocer el estado en que se encuentra, el grado de conformidad que se tiene con el proyecto según los datos arrojados de las pruebas funcionales realizadas al software, y por consiguiente dar un criterio de evaluación del proyecto en correspondencia con la conformidad que se tiene.

De ahí la necesidad de desarrollar el sistema para aplicarlo a los proyectos productivos del CISED y con esto se pudiera extender a los demás proyectos de la universidad facilitando así el proceso de pruebas.

El proceso de evaluación de un producto software está compuesto por tres subprocesos:

- Proceso de Pre-evaluación.
- Proceso de Medición.
- Proceso de análisis de los resultados de la evaluación.

De manera general se tiene que el proceso comienza con la aplicación de una encuesta para definir cuáles son las características y sub-características de calidad que se consideran que son importantes para evaluar en el software. Esta encuesta se le aplica a la alta gerencia del proyecto, a los jefes de

módulos o sub-proyectos y a otras personas con experiencia de las que se considere importante su criterio.

Después de aplicada la encuesta, se procesan los datos de la misma, y se decide cuáles serán las características y sub-características que se evaluarán en el software. Después se realizan una serie de pasos a seguir para la medición y evaluación de la calidad del software para emitir un criterio de evaluación. Durante el desarrollo de estos subprocesos se utilizan una serie de artefactos y plantillas que son importantes para la recopilación de los datos arrojados de las pruebas.

2.2 Descripción del sistema a desarrollar

El sistema propuesto es una aplicación web vinculada al Centro de Identificación y Seguridad Digital. La misma le brindará al usuario una interfaz amigable y sencilla con el objetivo de hacer más fácil la navegación con el sistema.

La aplicación a desarrollar estará constituida por una serie de vínculos de interés para el usuario (Calisoft, Intranet, Google, Biblioteca) y contará además con toda la documentación relacionada con el proceso de evaluación de un producto como lo es:

- Norma ISO/IEC 9126 (Para la evaluación del software).
- Norma ISO/IEC 14598 (Para la medición del proceso de evaluación).
- Norma ISO/IEC 25000 (Unión de la Norma ISO/IEC 9126 y la Norma ISO/IEC 14598).
- Tesis de Grado (Medición de la calidad de Software durante el Proceso de Pruebas en el Proyecto Modernización del CICPC Trabajo).

Los niveles de usuario definidos para el sistema de evaluación se especifican a continuación junto a sus permisos:

- *Usuario Invitado:* Es la persona que podrá consultar la información que se brinda en el sistema, referente a las noticias, los artículos relacionados con el tema y vínculos a otros sitios de interés.
- *Usuario Registrado:* Es la persona que tendrá los privilegios del usuario invitado y además, tendrá los permisos para acceder al sistema y realizar la encuesta para la pre-evaluación del software.

- *Administrador de Calidad:* Es la persona que controlará funcionalidades como la gestión de los proyectos a los que se les realizará la evaluación, el control de los usuarios a los que se les realizará las encuestas y además será el encargado de realizar todo el proceso de evaluación de un producto software.

El sistema estará compuesto por el módulo Gestión de la Calidad en el cual se mostrarán encuestas a ser realizadas por el usuario, las métricas correspondientes a cada una de las características y sub-características a ser evaluadas en el proyecto y el resultado del proceso de evaluación.

2.3 Requisitos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el sistema debe cumplir. Estas propiedades son las características que hacen al producto atractivo, usable, rápido y confiable. Están vinculados con los requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se podrá entonces determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Para el sistema se definieron los siguientes requisitos no funcionales:

Apariencia o interfaz externa:

- Interfaz sencilla y atractiva.
- Usar colores claros.
- Fácil navegación.
- Diseñado para 1024 x 768.
- Los nombres de las secciones deben ser lo más claro posible.

Usabilidad:

- La aplicación será usada por personas autorizadas según sus roles y su nivel de privilegio.

Soporte:

- Sistema multiplataforma.
- Drupal como Sistema de Gestión de Contenido.

- PostgreSQL como Gestor de Base de Datos.

Software:

- Sistema Operativo Multiplataforma.
- Servidor Web Apache.
- Navegador: Internet Explorer y Mozilla Firefox.

Hardware:

- La estación de trabajo (PC) tiene que tener como mínimo 512 MB de memoria RAM.
- Impresora para imprimir los Informes de Evaluación.
- Procesador Pentium IV o superior.
- Los servidores Proxy, Web y Bases de Datos deben poseer 512 MB de memoria RAM como mínimo.

Disponibilidad:

- La aplicación estará disponible las 24 horas del día y los 7 días de la semana para todos los usuarios que deseen acceder.

Seguridad:

- Comprobar que el usuario esté autenticado y que sea correcto el nombre de usuario y la contraseña antes de realizar las encuestas.
- Mostrar la funcionalidad correcta según los privilegios del usuario autenticado.
- La información estará protegida de accesos no autorizados.
- La información debe ser mostrada en el momento en que el usuario desee obtener la misma.

Restricciones del diseño:

- La aplicación será desarrollada en PHP, usando el CMS Drupal, para lograr un diseño sencillo.

2.4 Fase de Exploración

En la fase de exploración los clientes plantean a grandes rasgos las HU que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. (21)

En esta fase se define el alcance general del proyecto. Los programadores estiman los tiempos de desarrollo en base a las HU. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen detalladamente en cada iteración. Esta fase dura normalmente un par de semanas, el resultado es una visión general del sistema, y un plazo total estimado.

2.5 Historias de Usuario. Definición y Descripción

Las HU son artefactos de la metodología XP que permite describir las características que el sistema debe cumplir. Las mismas deben ser lo suficientemente comprensibles, sencillas y delimitadas para que los programadores puedan implementar la funcionalidad en un corto tiempo.

El tratamiento de las HU es muy dinámico y flexible, en cualquier momento pueden cambiar, reemplazarse por otras más específicas o generales, añadirse nuevas características o ser modificadas.

Las HU deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores se entrevistarán directamente con el cliente para obtener todos los detalles necesarios.

A continuación se describen algunas HU que se definieron para que el sistema que se desea implementar cumpla con las expectativas y necesidades del cliente. Las demás HU descritas se pueden consultar en el Anexo 3.

Tabla 2: Historia de usuario "Autenticar usuario".

<i>Historia de Usuario</i>	
Número: 1	Nombre: Autenticar usuario.
Usuario: Registrado, Administrador de calidad.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Descripción: <ul style="list-style-type: none"> ➤ El usuario registrado podrá autenticarse para acceder a las opciones según sus privilegios. ➤ El administrador del sistema podrá autenticarse para acceder a todas las opciones del sistema. 	
Observaciones:	

Tabla 3: Historia de usuario "Gestionar usuario".

<i>Historia de Usuario</i>	
Número: 2	Nombre: Gestionar usuario.
Usuario: Administrador de calidad.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Descripción: El administrador de calidad podrá adicionar, eliminar y modificar los siguientes datos del usuario: <ul style="list-style-type: none"> ➤ Usuario. ➤ Dirección de correo electrónico. ➤ Contraseña. ➤ Estado (Bloqueado o Activo). 	
Observaciones:	

Tabla 4: Historia de usuario “Gestionar proyecto”.

<i>Historia de Usuario</i>	
Número: 3	Nombre: Gestionar proyecto.
Usuario: Administrador de calidad.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Descripción: El administrador de calidad podrá adicionar, eliminar y modificarlos siguientes datos de un proyecto: <ul style="list-style-type: none"> ➤ Nombre del proyecto. ➤ Estado (Activo o Inactivo). 	
Observaciones:	

Tabla 5: Historia de usuario “Realizar encuesta”.

<i>Historia de Usuario</i>	
Número: 4	Nombre: Realizar encuestas.
Usuario: Registrado.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Descripción: <ul style="list-style-type: none"> ➤ El sistema muestra la opción de realizar encuesta. ➤ El sistema muestra la encuesta a realizar. ➤ El sistema brinda la opción de enviar la encuesta realizada. ➤ El sistema almacena los resultados obtenidos. 	

Observaciones:

Tabla 6: Historia de usuario “Procesar encuesta”.

<i>Historia de Usuario</i>	
Número: 5	Nombre: Procesar encuesta.
Usuario: Administrador de calidad.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Descripción:	
<ul style="list-style-type: none"> ➤ El sistema promedia los resultados de las encuestas realizadas. 	
Observaciones:	

2.6 Fase de Planificación

La fase de planificación es una fase corta. El cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las HU, y asociadas a éstas, las entregas. Esta fase consiste en una o varias reuniones grupales de planificación y el resultado de la misma es un Plan de Entrega.

La fase de planificación es fundamental para cualquier proyecto de software a desarrollar. En ella queda plasmado cómo se organizará el trabajo, cronograma de trabajo, qué roles intervienen y en qué momento. En esta fase el cliente establece la prioridad de cada HU, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. (21)

2.6.1 Estimación del esfuerzo por Historia de Usuario

La estimación del esfuerzo asociado a la implementación de cada una de las HU la establecen los programadores, utilizando como medida el punto que equivale a una semana ideal de programación. Las HU generalmente se les asignan de uno a tres puntos.

La estimación del esfuerzo de cada HU definido en la fase de exploración se recoge en la siguiente tabla:

Tabla 7: Plan de estimación de esfuerzo por historia de usuario.

<i>Historia de Usuario</i>	<i>Puntos Estimados</i>
Autenticar Usuario.	5
Gestionar Usuario	1
Gestionar Proyecto	1
Realizar encuesta.	1
Procesar encuesta.	1
Mostrar los resultados de la encuesta.	1
Mostrar métricas.	1
Calcular métricas.	1
Generar resultado de métricas.	1
Mostrar resultado de métricas.	1
Mostrar puntuación asignada a sub-características.	1
Mostrar el cálculo de la puntuación de las características.	1
Mostrar grado de conformidad.	1/2
Mostrar criterio de evaluación.	1/2
Gestionar informe de las no conformidades.	1
Exportar a pdf.	1/2
Total	15/2

2.6.2 Planificación de las Iteraciones

El Plan de Entrega está compuesto por iteraciones las cuales no exceden las cinco semanas. En la primera iteración se escogen las HU que fueren la creación de la arquitectura del sistema, sin embargo, esto no siempre es posible ya que es el cliente quien decide cuantas HU se implementarán en cada iteración. Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: HU no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

Las 16 HU definidas por el cliente estarán divididas en tres iteraciones, la duración de las iteraciones no debe ser mayor de un mes. A continuación se especifica el orden de prioridad de cada HU que se tuvo en cuenta para cada iteración.

Iteración 1

En esta iteración serán implementadas las HU de prioridad alta. Al culminar esta iteración, se tendrá una primera versión del sistema

Iteración 2

En la segunda iteración se implementarán las HU de prioridad alta que no se realizaron en la primera iteración, además de las de prioridad media, obteniendo como resultado una segunda versión del producto.

Iteración 3

Por último en la tercera iteración se deberán implementar las HU de prioridad media que no se realizaron en la iteración anterior y todas las HU con prioridad baja, obteniéndose al final de la misma la versión final del producto.

2.6.3 Duración de las Iteraciones

El Plan de Iteraciones permite una mayor organización y control de cada una de las Iteraciones. A continuación se muestra el plan definido para el desarrollo del sistema, el cual refleja por iteraciones el orden de las HU a implementar y la duración de cada una de ellas.

Tabla 8: Plan de iteraciones.

<i>Iteración</i>	<i>Orden de la HU a Implementar</i>	<i>Duración</i>
<i>Iteración # 1</i>	<ol style="list-style-type: none"> 1. Autenticar Usuario. 2. Gestionar Usuario. 3. Gestionar Proyecto. 4. Realizar encuesta. 5. Procesar encuesta. 	5 semanas
<i>Iteración # 2</i>	<ol style="list-style-type: none"> 1. Mostrar resultado de la encuesta. 2. Mostrar métricas. 3. Calcular métricas 4. Generar resultado de métricas. 5. Mostrar resultado de métricas. 	5 semanas
<i>Iteración # 3</i>	<ol style="list-style-type: none"> 1. Mostrar puntuación asignada a sub-características. 2. Mostrar el cálculo de la puntuación de las características. 3. Mostrar grado de conformidad. 4. Mostrar criterio de evaluación. 5. Gestionar informe de las no conformidades. 6. Exportar a pdf. 	4/2 semanas

2.6.3.1 Plan de Entrega

El plan estimado de entregas se realiza con las HU, y la duración de cada iteración. Al concluir cada iteración se obtiene una versión del producto y así sucesivamente dependiendo de la cantidad de iteraciones definidas en el Plan de Iteraciones, hasta llegar a una última iteración que una vez concluida se debe obtener un producto final.

El plan de entregas definido para la construcción del sistema se muestra a continuación, así como la fecha de terminación del mismo:

Tabla 9: Plan de entrega.

<i>Producto a Entregar</i>	<i>Aplicación para la calidad de los productos software</i>		
Iteración	1	2	3
Fecha inicio	10 de enero	21 de febrero	3 de abril
Fecha fin	12 de febrero	25 de marzo	29 de abril
Versión	0.1	0.2	1.0

2.7 Diseño del sistema

El uso de las tarjetas C.R.C (Clases, Responsabilidades y Colaboración) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

Las tarjetas C.R.C (Ver Anexo 4) representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad. (22)

2.7.1 Estructura de Drupal. Módulos. Patrones de diseño y Flujo de información

Todo el contenido creado en Drupal es funcionalmente un objeto (nodo), lo cual permite que se le brinde un tratamiento uniforme a la información. Unido a este contenido creado existen enlaces permanentes (permalinks), con el objetivo de que pueda ser enlazado externamente sin temor a que este enlace falle en un futuro. Drupal posee además un sistema de control de versiones que permite auditar las sucesivas actualizaciones del contenido, o sea, qué se ha cambiado, hora y fecha y quién lo cambió. (23)

Una aplicación construida en el CMS Drupal se caracteriza por contar con cuatro ficheros principales, los cuales a su vez incluyen gran cantidad de archivos en su interior. Por esta causa los mismos serán denominados como paquetes.

Dentro del paquete *Themes* se puede encontrar el mecanismo de soporte de diferentes plantillas, el paquete *Includes* contiene una serie de ficheros de configuración del sistema, los scripts para generar la Base de Datos, y clases utilitarias. Dentro del paquete *Scripts* se encuentran un conjunto de ficheros que se encargan de manejar el aspecto visual de la aplicación, fundamentalmente archivos CSS y javascript. Por último dentro del paquete *Modules* se encuentran los diferentes módulos que brindan las diversas funcionalidades de Drupal.

Básicamente Drupal contiene una única página servidor, la cual genera el contenido a mostrar en la página final, de acuerdo a los argumentos con que se realicen la petición. Estas páginas generadas por Drupal pueden o no contener formularios, esto depende del módulo sobre el que se esté trabajando y el propósito del mismo. En el presente trabajo solo se modelará el paquete *Modules* de Drupal, ya que este es el que contiene las principales funcionalidades del CMS. (23)

En la Figura 5 mostrada a continuación se puede observar detalladamente el diagrama de clases que propone Drupal para su diseño.

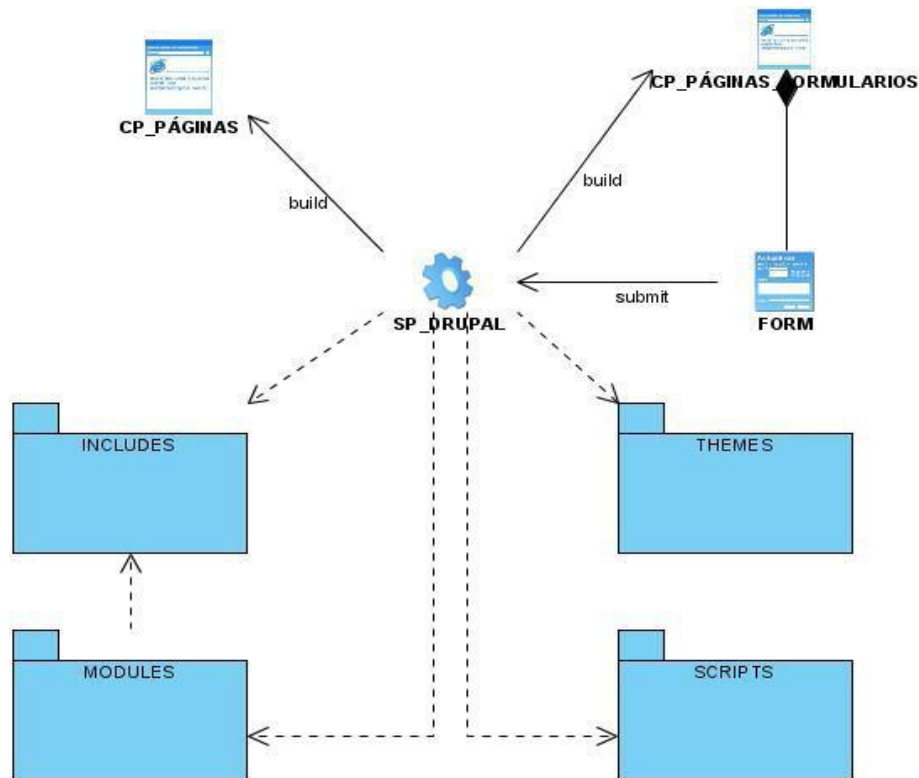


Figura 5: Diagrama de Clases del CMS Drupal

Los módulos en Drupal son ficheros con extensión .module que contienen funciones escritas en PHP. Estas funciones actúan como funciones gancho (hooks), que son llamadas por Drupal durante sus procesos habituales de gestión de contenido y construcción de páginas web. Por ejemplo, cada vez que un nodo es creado, visualizado, modificado o borrado, Drupal llama a una de estos hooks pasándoles el contenido del nodo. De esta forma los módulos tienen la posibilidad de modificar y adaptar la información a visualizar en las páginas web antes de que se mande definitivamente al navegador.

Actualmente existe un número realmente grande de módulos disponibles, temática diversa y de gran calidad. Antes de decidirse por un módulo u otro se deben revisar los requerimientos de los mismos. La mayoría de los módulos sólo piden que se instalen en una versión determinada de Drupal, pero otros pueden exigir la presencia de algún componente externo a Drupal, como la posibilidad de ejecutar

código Perl, o quizás una librería de terceros. Incluso algunos requieren la modificación del esquema de la Base de Datos, mediante la adición de columnas a las tablas ya existentes o la creación de tablas nuevas.

Los módulos específicamente en Drupal permiten extender la funcionalidad de la web, añadiendo por ejemplo, la posibilidad de realizar búsquedas por palabras dentro del contenido o de agregar comentarios a los nodos.

En el desarrollo de la aplicación se usaron los siguientes módulos:

Menu: Permite a los administradores personalizar el menú de navegación del sitio web.

FCKeditor: Habilita un editor de texto para campos de texto.

LDAP_Integration: Para la autenticación de un dominio externo.

Gestión de la calidad: Es el encargado de realizar todos los procesos relacionados con la propuesta solución.

Upload: Permite que los usuarios puedan adjuntar archivos a los contenidos.

Views: Proporciona un método flexible a los diseñadores para controlar la forma en que se lista el contenido de un sitio Drupal. Este módulo es esencialmente un constructor de consultas inteligentes, que proporcionándole la suficiente información, es capaz de construir la consulta adecuada, ejecutarla y mostrar los resultados.

Un patrón de diseño es básicamente una solución a un problema común en el desarrollo de software, aplicable a otros contextos referentes al diseño.

El funcionamiento de Drupal hace uso de algunos patrones de diseño que son propiamente de sistemas orientados a objetos, particularmente los patrones GOF (*Gang-of-Four*). Dentro de los patrones GOF existen varias categorías como son: creacionales, estructurales y de comportamiento, todos ellos contribuyen a la construcción de un diseño más elegante y robusto. Los patrones que se manifiestan en el funcionamiento de Drupal son: *Singleton* (instancia única), *Decorator* (decorador), *Observer* (observador), *Bridge* (puente) y *Chain of Responsibility* (cadena de responsabilidades). Estos patrones le

proporcionan al diseño de Drupal una gran flexibilidad y extensibilidad, posibilitando además que su funcionamiento tenga características similares al funcionamiento de los sistemas orientados a objetos, lo cual lo convierte en una poderosa plataforma de construcción de aplicaciones web.

Creacionales: Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo qué clases instanciar o sobre qué objetos delegará responsabilidades otro objeto.

➤ **Singleton**

El patrón de diseño *Singleton* (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Las situaciones más habituales de aplicación de este patrón son aquellas en las que dicha clase controla el acceso a un recurso físico único (como puede ser el ratón o un archivo abierto en modo exclusivo) o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación.

Estructurales: Describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.

➤ **Decorator:**

El patrón *Decorator* (decorador) responde a la necesidad de añadir dinámicamente funcionalidades a un objeto. Esto permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera.

➤ **Bridge**

El patrón *Bridge* (puente), es una técnica usada en programación para desacoplar una abstracción de su implementación, de manera que ambas puedan ser modificadas independientemente sin necesidad de alterar por ello la otra.

Comportamiento: Se utilizan para organizar, manejar y combinar comportamientos.

➤ **Observer**

El patrón *Observer* (Observador) también conocido como "spider" define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes.

➤ **Chain of Responsibility**

El patrón de diseño *Chain of Responsibility* (cadena de responsabilidades) permite establecer una cadena de objetos receptores a través de los cuales se pasa una petición formulada por un objeto emisor. Cualquiera de los objetos receptores puede responder a la petición en función de un criterio establecido.

ReflectionPattern

Este patrón de diseño especifica cómo se comportan las aplicaciones creadas en Drupal. Ayuda a modificar toda la información del sistema a través de los módulos, sin tener que interactuar con el núcleo, logrando que el sitio sea realmente adaptable a diferentes entornos.

Flujo de la información en Drupal.

Para comprender como fluye la información en el CMS Drupal, es necesario tomar en cuenta que este gestor de contenidos cuenta con cinco capas fundamentales a considerar. En la ilustración que se encuentra en los anexos se representa la distribución de estas capas:

Capa de Datos y Nodos: En la base del sistema está la colección de nodos, el banco de datos. Antes que cualquier elemento sea mostrado en el sitio, debe introducirse como datos.

Capa de Módulos: En esta capa se encuentran los módulos que son complementos funcionales que son parte del núcleo de Drupal (que vienen con Drupal) o que son aportados por los miembros de la comunidad de Drupal. Los módulos se basan en la funcionalidad básica de Drupal, lo que le permite personalizar los elementos de datos (campos) de los tipos de nodo.

Capa de Bloques y Menús: En esta capa se encuentran bloques y menús. Los bloques proporcionan a menudo la salida de un módulo o pueden ser creados para mostrar lo que el usuario desee, luego se pueden colocar en varios puntos de la plantilla (tema) de diseño. Los bloques se pueden configurar para la producción de diversas formas, así como simplemente mostrarse en determinadas páginas definidas, o para determinados usuarios definidos.

Capa de Permisos de Usuario: En esta capa están configurados los permisos para determinar a qué tipos de usuarios se les permiten hacer y ver. Los permisos se definen para funciones diversas, y a su vez, los usuarios son asignados a estas funciones con el fin de concederles los permisos definidos.

Capa de Tema: Se compone fundamentalmente de XHTML y CSS, con algunas variables PHP mezclados, por lo que el contenido generado por Drupal puede ir en los lugares adecuados. También se incluye con cada tema un conjunto de funciones que pueden utilizarse para reemplazar las funciones estándar de los módulos con el fin de proporcionar un control completo sobre cómo los módulos generan sus marcas a la hora de salida. Las plantillas también se pueden asignar en la marcha sobre la base de permisos de usuario.

2.8 Diagrama de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. (24)

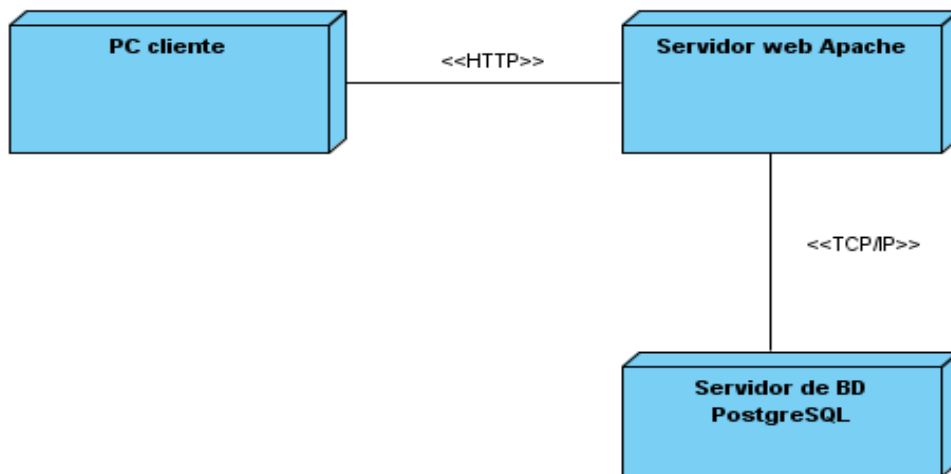


Figura 6: Diagrama de Despliegue

2.9 Conclusiones del capítulo

En este capítulo se describió de manera general los procesos de negocio, además se expusieron las características del sistema, así como las HU y los Requisitos No Funcionales del mismo, lo que permitió conocer la complejidad del sistema que se quiere desarrollar. Se elaboró un plan de entrega que define cada una de las iteraciones que conformarán el ciclo de vida del software. Por cada iteración se definen las HU correspondientes estableciéndose una duración para la iteración.

Se realizó un análisis del funcionamiento, la estructura y los módulos que posee el CMS Drupal, lo que permitió hacer un uso adecuado de sus funcionalidades. De modo general se puede decir que para que el software cuente con una buena calidad, debe haberse realizado un análisis cabal de cada uno de los requisitos que este debe cumplir.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Este capítulo abarcará las otras dos fases que define la metodología XP, que son Implementación y Prueba, cumpliendo cuidadosamente con todo lo definido en la etapa de Planificación. Se implementará cada una de las HU en correspondencia con la iteración para la cual se planificó. Además, se le realizará al software las pruebas que se definirán, para que cuente con la calidad máxima requerida, y finalmente cumpla con las expectativas del cliente.

3.1 Fase de Implementación

Durante el transcurso de las iteraciones se realizará la implementación de las HU seleccionadas para cada una de estas. Al inicio de las mismas, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Ajustándose a la planificación realizada, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose al finalizar, un producto listo para su puesta en función. A continuación se detalla cada una de las iteraciones. (25)

3.1.1 Tareas de las HU implementadas en la Iteración 1

Durante esta iteración se abordaron las HU de mayor prioridad en el sistema, siguiendo como objetivo el de obtener una primera versión del producto con las principales funcionalidades para ser mostrado al cliente.

Tabla 10: HU Implementada en la Iteración 1.

No	Historia de Usuario	Estimación	Real
1	Autenticar Usuario.	1	1
2	Gestionar Usuario.	1	1
3	Gestionar Proyecto.	1	1
4	Realizar Encuesta.	1	1
5	Procesar Encuesta.	1	1
Total		5	5

HU Autenticar Usuario

Tabla 11: Tarea 1 “Autenticar Usuario”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 1
Nombre de la tarea: Configuración de la interfaz para la autenticación de usuarios.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-01-10	Fecha de fin: 2010-01-10
Programador responsable: Anabeisy Pérez González.	
Descripción: Se configura la interfaz para la autenticación de los usuarios que harán uso de la aplicación.	

Tabla 12: Tarea 2 “Autenticar Usuario”.

<i>Tarea de Implementación</i>	
Número de tarea: 2	Número de historia: 1
Nombre de la tarea: Crear el usuario Administrador de calidad.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-01-11	Fecha de fin: 2010-01-11
Programador responsable: Anabeisy Pérez González.	
Descripción: Se creará el usuario Administrador de calidad el cual será el encargado de la gestión de los proyectos, de gestionar los permisos de los usuarios que interactuarán con el sistema y realizar todo el proceso de evaluación del producto software.	

Tabla 13: Tarea 3 “Autenticar Usuario”.

<i>Tarea de Implementación</i>	
--------------------------------	--

Número de tarea: 3	Número de historia: 1
Nombre de la tarea: Agregar el módulo <i>LDAP_Integration</i> y activar el sub_módulo <i>ldapauth</i> .	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-01-12	Fecha de fin: 2010-01-13
Programador responsable: Anabeisy Pérez González.	
Descripción: Se agregará el módulo <i>LDAP_Integration</i> en la carpeta de módulos del CMS y se activará el sub_módulo <i>ldapauth</i> . De esta manera los usuarios se registrarán por el dominio UCI.	

Tabla 14: Tarea 4 “Autenticar Usuario”.

<i>Tarea de Implementación</i>	
Número de tarea: 4	Número de historia: 1
Nombre de la tarea: Comprobar si el usuario es válido.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-01-14	Fecha de fin: 2010-01-14
Programador responsable: Anabeisy Pérez González.	
Descripción: Se comprueban los datos del usuario autenticado verificando si es válido o no.	

Tabla 15: Tarea 5 “Autenticar Usuario”.

<i>Tarea de Implementación</i>	
Número de tarea: 5	Número de historia: 1
Nombre de la tarea: Acceso con los correspondientes permisos.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: : 2010-01-15	Fecha de fin: : 2010-01-15

Programador responsable: Anabeisy Pérez González.
Descripción: Se le concede permiso de acceder a la aplicación si el usuario es válido con los permisos correspondientes al mismo.

HU Gestionar Usuario

Tabla 16: Tarea 1 “Gestionar Usuario”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 2
Nombre de la tarea: Crear Usuario.	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha de inicio: : 2010-01-17	Fecha de fin: : 2010-01-17
Programador responsable: Anabeisy Pérez González.	
Descripción: Se crean los usuarios que interactuarán con la aplicación, se insertan en la base de datos y se le asignan los permisos correspondientes.	

Tabla 17: Tarea 2 “Gestionar Usuario”.

<i>Tarea de Implementación</i>	
Número de tarea: 2	Número de historia: 2
Nombre de la tarea: Eliminar Usuario.	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha de inicio: : 2010-01-18	Fecha de fin: : 2010-01-19
Programador responsable: Anabeisy Pérez González.	
Descripción: Se debe acceder a editar perfil para eliminarlo, donde los datos y privilegios del usuario	

seleccionado son eliminados de la base de datos.

Tabla 18: Tarea 3 “Gestionar Usuario”.

<i>Tarea de Implementación</i>	
Número de tarea: 3	Número de historia: 2
Nombre de la tarea: Modificar Usuario.	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha de inicio: : 2010-01-20	Fecha de fin: : 2010-01-21
Programador responsable: Anabeisy Pérez González.	
Descripción: Se debe acceder a editar perfil para modificarlo, donde los datos insertados serán guardados en la base de datos.	

HU Gestionar Proyecto

Tabla 19: Tarea 1 “Gestionar Proyecto”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 3
Nombre de la tarea: Crear Proyecto.	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha de inicio: 2010-01-24	Fecha de fin: 2010-01-26
Programador responsable: Anabeisy Pérez González.	
Descripción: Se implementan las funcionalidades que permiten la captura de los datos (nombre del proyecto, el estado (activo o inactivo)). Luego se realiza la validación de los datos, de forma tal que si existe un proyecto al que se le esté realizando el proceso de evaluación, el proyecto que se inserte nuevo se le	

mostrará la información de que solamente puede estar un proyecto activo.

Tabla 20: Tarea 2 “Gestionar Proyecto”.

<i>Tarea de Implementación</i>	
Número de tarea: 2	Número de historia: 3
Nombre de la tarea: Eliminar Proyecto.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-01-27	Fecha de fin: 2010-01-28
Programador responsable: Anabeisy Pérez González.	
Descripción: Se implementan las funcionalidades que permiten eliminar el proyecto seleccionado.	

HU Realizar encuesta

Tabla 21: Tarea 1 “Realizar encuesta”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 4
Nombre de la tarea: Creación de la Encuesta.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-01-31	Fecha de fin: 2010-02-2
Programador responsable: Anabeisy Pérez González.	
Descripción: Se crean los formularios para mostrar la encuesta.	

Tabla 22: Tarea 2 “Realizar encuesta”.

<i>Tarea de Implementación</i>	
--------------------------------	--

Número de tarea: 2	Número de historia: 4
Nombre de la tarea: Validación de la Encuesta.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-02-3	Fecha de fin: 2010-02-4
Programador responsable: Anabeisy Pérez González.	
Descripción: Se implementan las funcionalidades para la validación de los datos entrados en la encuesta.	

HU Procesar encuesta

Tabla 23: Tarea 1 “Procesar encuesta”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 5
Nombre de la tarea: Procesar Encuesta.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-02-7	Fecha de fin: 2010-02-12
Programador responsable: Anabeisy Pérez González.	
Descripción: Se implementan todas las funcionalidades para procesar el resultado de la encuesta realizada.	

3.1.2 Tareas de las HU implementadas en la Iteración 2

Durante esta iteración se implementan las funcionalidades de prioridad en el sistema alta y media, con el objetivo de percibir una idea más amplia de cómo quedará la funcionalidad de la aplicación al final de su desarrollo.

Tabla 24: HU Implementada en la Iteración 2.

No	Historia de Usuario	Estimación	Real
----	---------------------	------------	------

6	Mostrar resultado de la encuesta.	1	1
7	Mostrar métricas.	1	1
8	Calcular métricas.	1	1
9	Generar resultado de las métricas.	1	1
10	Mostrar resultado de las métricas.	1	1
Total		5	5

HU Mostrar resultado de la encuesta

Tabla 25: Tarea 1 “Mostrar resultado de la encuesta”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 6
Nombre de la tarea: Mostrar resultado de la encuesta.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010- 02- 21	Fecha de fin: 2010- 02- 25
Programador responsable: Anabeisy Pérez González.	
Descripción: Se implementan todas las funcionalidades para mostrar el resultado de la encuesta.	

HU Mostrar métricas

Tabla 26: Tarea 1 “Mostrar métricas”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 7
Nombre de la tarea: Creación de las métricas.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1

Fecha de inicio: 2010- 02- 28	Fecha de fin: 2010- 03-4
Programador responsable: Anabeisy Pérez González.	
Descripción: Se crean los formularios para mostrar las métricas y se implementan las funcionalidades que permiten la captura de los datos (peso).	

HU Calcular métricas

Tabla 27: Tarea 1 “Calcular métricas”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 8
Nombre de la tarea: Calcular métricas.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010-03 -7	Fecha de fin: 2010-03-11
Programador responsable: Anabeisy Pérez González.	
Descripción: Se implementan todas las funcionalidades para calcular las métricas.	

HU Generar resultado de las métricas

Tabla 28: Tarea 1 “Generar resultado de las métricas”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 9
Nombre de la tarea: Generar resultados de las métricas.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010- 03- 14	Fecha de fin: 2010- 03- 18

Programador responsable: Anabeisy Pérez González.
Descripción: Se implementan las funcionalidades para generar el resultado de las métricas.

HU Mostrar resultado de las métricas

Tabla 29: Tarea 1 “Mostrar resultado de las métricas”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 10
Nombre de la tarea: Creación de los resultados de las métricas.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010- 03- 21	Fecha de fin: 2010- 03- 25
Programador responsable: Anabeisy Pérez González.	
Descripción: Se crea el formulario para mostrar el resultado de las métricas.	

3.1.3 Tareas de las HU implementadas en la Iteración 3

Al culminar ésta iteración se tendrá una versión final del producto, ya que se habrán implementado todas las funcionalidades del sistema, quedando listo para su puesta en funcionamiento.

Tabla 30: HU Implementada en la Iteración 3.

No	Historia de Usuario	Estimación	Real
1	Mostrar puntuación asignada a las sub-características.	1	1
2	Mostrar el cálculo de la puntuación de las características.	1	1
3	Mostrar grado de conformidad.	1/2	1/2
4	Mostrar criterio de evaluación.	1/2	1/2

5	Gestionar informe de las no conformidades.	1	1
6	Exportar a PDF.	1/2	1/2
Total		4/2	4/2

HU Mostrar puntuación asignada a las sub-características

Tabla 31: Tarea 1 “Mostrar puntuación asignada a las sub-características”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 11
Nombre de la tarea: Calcular puntuación de las sub-características.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010- 04- 4	Fecha de fin: 2010- 04- 6
Programador responsable: Anabeisy Pérez González.	
Descripción: Se implementa la función “ <i>CalcularPuntuaciónAsignada</i> ” para calcular la puntuación asignada a las sub-características.	

Tabla 32: Tarea 2 “Mostrar puntuación asignada a las sub-características”.

<i>Tarea de Implementación</i>	
Número de tarea: 2	Número de historia: 11
Nombre de la tarea: Crear formulario para las sub-características.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010- 04- 7	Fecha de fin: 2010- 04- 8
Programador responsable: Anabeisy Pérez González.	
Descripción: Se crea el formulario para mostrar la puntuación asignada a las sub-características.	

HU Mostrar el cálculo de la puntuación de las características

Tabla 33: Tarea 1 “Mostrar el cálculo de la puntuación de las características”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 12
Nombre de la tarea: Calcular puntuación de las características.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010- 04-11	Fecha de fin: 2010- 04- 13
Programador responsable: Anabeisy Pérez González.	
Descripción: Se implementa la función “ <i>CalcularPuntuaciónFinal</i> ” para calcular la puntuación asignada a las características.	

Tabla 34: Tarea 2 “Mostrar el cálculo de la puntuación de las características”.

<i>Tarea de Implementación</i>	
Número de tarea: 2	Número de historia: 12
Nombre de la tarea: Crear formulario de las características.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1
Fecha de inicio: 2010- 04- 14	Fecha de fin: 2010- 04- 15
Programador responsable: Anabeisy Pérez González.	
Descripción: Se crea el formulario para mostrar el cálculo de la puntuación de las características.	

HU Mostrar grado de conformidad

Tabla 35: Tarea 1 “Mostrar grado de conformidad”.

<i>Tarea de Implementación</i>	
--------------------------------	--

Número de tarea: 1	Número de historia: 13
Nombre de la tarea: Calcular grado de conformidad.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1/2
Fecha de inicio: 2010- 04- 18	Fecha de fin: 2010- 04- 19
Programador responsable: Lisbet Suárez Lissabet.	
Descripción: Se implementa la función “ <i>CalcularGradoConformidad</i> ” para calcular el grado de las conformidades del producto.	

Tabla 36: Tarea 2 “Mostrar grado de conformidad”.

<i>Tarea de Implementación</i>	
Número de tarea: 2	Número de historia: 13
Nombre de la tarea: Crear formulario para el grado de conformidad.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1/2
Fecha de inicio: 2010- 04- 20	Fecha de fin: 2010- 04- 20
Programador responsable: Lisbet Suárez Lissabet.	
Descripción: Se crea el formulario para mostrar el resultado del cálculo de las conformidades del producto.	

HU Mostrar criterio de evaluación

Tabla 37: Tarea 1 “Mostrar criterio de evaluación”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 14
Nombre de la tarea: Calcular criterio de evaluación.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1/2

Fecha de inicio: 2010- 04- 21	Fecha de fin: 2010- 04- 22
Programador responsable: Lisbet Suárez Lissabet.	
Descripción: Se implementa la función “ <i>CalcularCriterioEvaluación</i> ” para calcular el criterio de evaluación del producto.	

Tabla 38: Tarea 2 “Mostrar criterio de evaluación”.

<i>Tarea de Implementación</i>	
Número de tarea: 2	Número de historia: 14
Nombre de la tarea: Crear formulario para el criterio de evaluación.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1/2
Fecha de inicio: 2010- 04- 23	Fecha de fin: 2010- 04- 23
Programador responsable: Lisbet Suárez Lissabet.	
Descripción: Se crea el formulario para mostrar el resultado del cálculo del criterio de evaluación del producto.	

HU Gestionar informes de las no conformidades

Tabla 39: Tarea 1 “Gestionar informe de las no conformidades”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 15
Nombre de la tarea: Crear formularios de las no conformidades.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1/2
Fecha de inicio: 2010- 04- 25	Fecha de fin: 2010- 04- 26
Programador responsable: Lisbet Suárez Lissabet.	

Descripción: Se crean los formularios para la entrada de las no conformidades encontradas, teniendo en cuenta los principales problemas que fueron detectados durante el proceso de evaluación y la propuesta de acciones correctivas.

HU Exportar a pdf

Tabla 40: Tarea 1 “Exportar a pdf”.

<i>Tarea de Implementación</i>	
Número de tarea: 1	Número de historia: 16
Nombre de la tarea: Exportar a pdf.	
Tipo de tarea: Desarrollo.	Puntos Estimados: 1/2
Fecha de inicio: 2010- 04- 27	Fecha de fin: 2010- 04- 29
Programador responsable: Lisbet Suárez Lissabet.	
Descripción: Se implementa la función “ <i>GenerarPdfReporte</i> ” para exportar a pdf el informe de evaluación.	

3.2 Fase de Pruebas

El proceso de pruebas a un software, permite verificar y relevar la calidad del producto. Estas pruebas se integran dentro de las diferentes fases del ciclo de vida del software, demostrando que los requisitos implementados funcionen correctamente.

El Plan de pruebas describe la estrategia, recursos y planificación de las pruebas. La estrategia de prueba incluye la definición del tipo de pruebas a realizar para cada iteración y sus objetivos, el nivel de cobertura de prueba y el porcentaje de prueba que deberían ejecutarse con un resultado específico.

(26)

3.2.1 Tipos de Pruebas

Se pueden determinar muchos casos de pruebas para determinar que un requisito es completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga acciones secundarias. En ese caso, cada requisito secundario deberá tener por lo menos un caso de prueba.

3.2.1.1 Pruebas unitarias

Una forma de probar el correcto funcionamiento del software es la realización de las pruebas unitarias. Esta prueba conocida también como Caja Blanca o Pruebas Modulares, asegura que cada módulo implementado funcione adecuadamente. La base de este método es el hacer pruebas en pequeños fragmentos al código, encargados de una tarea específica.

El objetivo de las pruebas unitaria es inspeccionar el código y la demostración de que no contengan errores y en caso de contener proceder a su pronta eliminación. (27)

3.2.1.2 Pruebas aceptación

En las pruebas de aceptación, se evalúa el grado de calidad del software con relación a todos los aspectos relevantes. Estas pruebas son conocidas también como pruebas de Caja negra. El objetivo de estas pruebas es validar que el sistema cumpla con lo pactado con el cliente.

La validación al sistema demuestra la conformidad del cliente antes la implementación de las HU definidos por él. Además se tienen en cuenta los requisitos no funcionales relacionados con el rendimiento y la seguridad, así como los distintos recursos del sistema. Estas pruebas son muy importantes ya que especifican etapas nuevas del proyecto como el despliegue y mantenimiento del mismo. (28)

Las pruebas de aceptación serán desarrolladas por un grupo de usuarios finales y estos a su vez se asegurarán de que el sistema cumpla con todos los requisitos. Las mismas serán evaluadas de Excelente, Bien, Regular y Mal. Los casos de pruebas descritos podrán ser consultados en el Anexo 5.

A continuación se muestran los casos de pruebas por iteración, enumerándose varias no conformidades detectadas en la realización de las pruebas.

Tabla 41: No conformidades por casos de pruebas

<i>Iteraciones</i>	<i>Casos de Prueba</i>	<i>Cantidad de NC por CP</i>
1	Autenticar usuario con permiso.	1
	Autenticar usuario sin permisos.	0
	Adicionar datos de usuario.	0
	Eliminar datos de usuario.	0
	Modificar datos de usuario.	0
	Adicionar proyecto.	1
	Eliminar proyecto.	0
	Modificar proyecto.	1
	Comprobar la correcta entrada de datos.	1
2	Mostrar resultados de la encuesta.	1
	Mostrar las métricas.	0
	Calcular las métricas.	1
	Procesar resultados de las métricas.	1
	Mostrar resultados de las métricas.	0
3	Comprobar el correcto cálculo de la puntuación asignada.	1
	Comprobar el correcto cálculo de puntuación de las características.	1
	Grado de conformidad.	0
	Criterio de evaluación.	0
	Informe de no conformidad.	1
	Exportar a pdf.	0

Durante la fase de prueba realizada al Sistema de Automatización de la Calidad de un Producto Software, se aplicaron las pruebas de aceptación o caja negra, con el objetivo de detectar deficiencias y comprobar así que la aplicación funcione correctamente. En una primera iteración se detectaron diez no conformidades, las cuales fueron corregidas posteriormente en una segunda iteración. De esta forma se logró la implementación de todas las HU satisfactoriamente y se obtuvo como resultado final una aplicación que cumple con las expectativas del cliente.

3.3 Conclusiones del capítulo

En el capítulo se desarrollaron las fases Implementación y Prueba donde se realizaron un total de 27 tareas y 21 casos de pruebas de aceptación, equivalente a las 16 HU descritas en la Fase de Exploración. Las tareas fueron desarrolladas en el tiempo establecido, cumpliendo con la entrega de las versiones del producto en la fecha señalada, llegándose a la conclusión de que el software funciona correctamente y que las pruebas fueron superadas satisfactoriamente.

CONCLUSIONES

El proceso de evaluación de la calidad de un producto software representa gran importancia para todos los proyectos productivos del CISED, ya que permite emitir un criterio de evaluación y un grado de conformidad con respecto a la calidad del producto software que se está evaluando.

Para darle cumplimiento a los objetivos propuestos se realizó un profundo análisis de las metodologías de desarrollo de software, herramientas, técnicas para el diseño de la aplicación, los lenguajes de programación, así como los Sistemas de Gestión de Contenidos, lo que facilitó realizar una correcta selección de los mismos para el desarrollo de la aplicación.

La metodología de desarrollo seleccionada permitió elaborar todos los artefactos necesarios para desarrollar el sistema, además de optimizar el tiempo de construcción y de permitir un plan de iteraciones para una mejor planificación.

Con la realización de las pruebas al sistema se pudo confirmar que el mismo cuenta con la calidad requerida para llevar a cabo el proceso de evaluación de la calidad de un producto software.

Se aporta a la UCI una herramienta que permite evaluar la calidad del producto software durante el proceso de las pruebas funcionales al software.

RECOMENDACIONES

El sistema para la Evaluación de la Calidad del Producto Software se implementó satisfactoriamente, no obstante, con el propósito de lograr un sistema que cuente con más funcionalidades y que abarque más en elementos de la Calidad de un producto software se recomienda que:

- Se incluyan las características de la calidad correspondientes a la Mantenibilidad y la Portabilidad.
- Que se incluyan otras métricas para la evaluación de las sub-características.
- Que el sistema para la evaluación de la calidad de un producto software sea extendido a los demás proyectos de la UCI.

REFERENCIAS BIBLIOGRÁFICA

1. **Pressman, Roger S.** *Libro Ingeniería de Software. Un enfoque práctico.* s.l. : MC GRAW HILL, 1998.
2. —. *Calidad de Software.* [En línea] 1998. [Citado el: 4 de Diciembre de 2010.] <http://www.slideshare.net/lcahuich/calidad-del-software-presentation>.
3. *Deltell & Asociados Contadores Públicos.* [En línea] 2000. [Citado el: 4 de Diciembre de 2010.] <http://deltellyasociados.com/certificacion.php>.
4. **Zambrana, Yayneris Hernandez.** *Medición de la calidad de Software durante el proceso de Pruebas en el Proyecto Modernización del CICPC.* Ciudad de la Habana : s.n., 2008.
5. **Beck, Kent.** *Xp Programing.* [En línea] [Citado el: 6 de Diciembre de 2010.] <http://www.buenastareas.com/ensayos/Xp-Programing/25570.html>.
22. *Fases de la Programación Extrema.* [En línea] [Citado el: 5 de Mayo de 2011.] <http://programacionextrema.tripod.com/fases.htm>.
23. **Veigas Chkout, Zenia.** *Portal para la gestión de la información del Centro de Identificación y Seguridad Digital.* Ciudad de la Habana : s.n., 2009.
25. **Fernández Escribano, Gerardo.** *Introducción a Extreme Programming.* [En línea] 9 de Diciembre de 2002. [Citado el: 15 de Abril de 2011.] <<http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion>.
26. *Fases de la Programación Extrema.* [En línea] [Citado el: 15 de Abril de 2011.] <http://programacionextrema.tripod.com/fases.htm>.
29. **Marqués.** *METODOLOGÍA PARA LA ELABORACIÓN DE SOFTWARE EDUCATIVO.* 1995.

BIBLIOGRAFÍA

1. **Pressman, Roger S.** *Libro Ingeniería de Software. Un enfoque práctico.* s.l. : MC GRAW HILL, 1998.
2. —. *Calidad de Software.* [Online] 1998. [Cited: Diciembre 4, 2010.] <http://www.slideshare.net/lcahuich/calidad-del-software-presentation>.
3. *Deltell & Asociados Contadores Públicos.* [Online] 2000. [Cited: Diciembre 4, 2010.] <http://deltellyasociados.com/certificacion.php..>
4. **Zambrana, Yayneris Hernandez.** *Medición de la calidad de Software durante el proceso de Pruebas en el Proyecto Modernización del CICPC.* Ciudad de la Habana : s.n., 2008.
5. **Beck, Kent.** *Xp Programing.* [Online] [Cited: Diciembre 6, 2010.] <http://www.buenastareas.com/ensayos/Xp-Programing/25570.html..>
6. **Rafael, Jorge Collorana.** *FDD : Feature Driven Development.* [Online] UNIVERSIDAD UNION BOLIVARIANA, Agosto 1, 2009. [Cited: Diciembre 11, 2010.] <http://www.adolfo.mex.tl/images/18149/JORGE.ppt..>
7. **Flower, Martin.** *La Nueva Metodología .* [Online] Abril 2003. [Cited: Diciembre 5, 2010.] <http://www.programacionextrema.org/articulos/newMethodology.es.html>.
8. **Patricio Letelier y M^a Carmen Penadés.** *Métodologías ágiles para el desarrollo de software:.* [Online] Universidad Politécnica de Valencia. [Cited: Diciembre 5, 2010.] <http://www.willydev.net/descargas/masyxp.pdf>.
9. **Menéndez, Evelin Alonso.** *Monografías.* [Online] [Cited: Enero 10, 2011.] <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
10. *Visual Paradigm For Uml.* [Online] [Cited: Enero 12, 2011.] <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
11. *Zend Studio - the leading PHP IDE.* [Online] [Cited: Enero 13, 2011.] <http://www.zend.com/en/products/studio/>.

12. *El servidor de web Apache: Introducción práctica: Apache 1.x y 2.0 alpha*. [Online] [Cited: Enero 16, 2011.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html>.
13. **Ayllapan, Walter Ulises** . [Online] <http://www.joomlaos.net/caracteristicas-de-joomla>.
14. *CMS Drupal, el futuro es ahora...!!!* [Online] [Cited: Enero 24, 2011.] <http://www.pilos.com.co/drupal/27-caracteristicas-de-drupal/>.
15. **Gillermo**. *El lenguaje de programación PHP*. [Online] Media Textual, Enero 29, 2010. [Cited: Diciembre 15, 2010.] <http://programandoideas.com/el-lenguaje-de-programacion-php/>.
16. *AulaClic*. [Online] Enero 2004. [Cited: Enero 16, 2010.] http://www.aulaclic.es/html/t_1_1.htm.
17. **Alvares, Miguel Angel**. *Qué es Javascript*. [Online] Enero 1, 2001. [Cited: Diciembre 17, 2010.] <http://www.desarrolloweb.com/articulos/25.php>.
18. **Alvares, Sara**. *Sistemas gestores de bases de datos*. [Online] Julio 31, 2007. [Cited: Enero 10, 2011.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
19. *PostgreSQL vs. MySQL*. [Online] [Cited: Enero 11, 2011.] http://danielpecos.com/docs/mysql_postgres/x57.html.
20. **Martinez, Rafael**. *Sobre PostgreSQL*. [Online] Octubre 2, 2010. [Cited: Enero 11, 2011.] http://www.postgresql-es.org/sobre_postgresql.
21. *Ciclo de vida de un proyecto XP*. [Online] [Cited: Enero 20, 2011.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
22. *Fases de la Programación Extrema*. [Online] [Cited: Mayo 5, 2011.] <http://programacionextrema.tripod.com/fases.htm>.
23. **Veigas Chkout, Zenia**. *Portal para la gestión de la información del Centro de Identificación y Seguridad Digital*. Ciudad de la Habana : s.n., 2009.
24. *Diagrama de Despliegue UML 2*. [Online] [Cited: Marzo 20, 2011.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
25. **Fernández Escribano, Gerardo**. *Introducción a Extreme Programming*. [Online] Diciembre 9, 2002. [Cited: Abril 15, 2011.] <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion>.

26. *Fases de la Programación Extrema*. [Online] [Cited: Abril 15, 2011.] <http://programacionextrema.tripod.com/fases.htm>.
27. **Oré B, Alexander**. *UNIT TESTING - PRUEBAS UNITARIAS - CAP 1*. [Online] [Cited: Abril 15, 2011.] http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php.
28. **Rojas, Johanna and Barrios, Emilio**. *Pruebas de Aceptación*. [Online] 2007. [Cited: Abril 15, 2011.] <http://gemini.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node55.html>.
29. **Marqués**. *METODOLOGÍA PARA LA ELABORACIÓN DE SOFTWARE EDUCATIVO*. 1995.

ANEXOS

Anexo 1 Tablas de las métricas

Tabla de Métricas de Funcionalidad

MÉTRICAS DE USABILIDAD				
Nombre de la métrica	Propósito de la Métrica	Método de Aplicación	Fórmula	Interpretación del valor obtenido
Subcaracterística: Comprensibilidad				
Entendimiento de funciones	¿Qué proporción de las funciones del producto será el usuario capaz de comprender correctamente?	Contar el número de interfaces de usuario donde su propósito es el entendimiento del usuario y comparar con el número de funciones de interfaces de usuarios.	$X = A/B$ A: Número de funciones de interfaz de usuario cuyo propósito es entendido por el usuario. B: Número total de funciones de interfaz de usuario.	$0 \leq X \leq 1$ Mientras más cercano a 1, mejor.
Subcaracterística: Operabilidad				
Chequeo de la Validez de la entrada	¿Qué proporción de número de elementos de entrada chequean la validez de los datos y compara con el número de	Contar el número de elementos de entrada, que chequean la validez de los datos y compara con el número de	$X = A/B$ A: Número de elementos de entrada que chequean validez de los	$0 \leq X \leq 1$ Mientras más cercano a 1, mejor.

	<i>validez de los datos?</i>	<i>elementos de entrada, que podría verificar para datos válidos.</i>	<i>datos. B: Número de elementos de entrada que pueden chequear la validez de los datos.</i>	
--	------------------------------	---	--	--

Tabla de Métricas de Confiabilidad

MÉTRICAS DE FUNCIONALIDAD				
Nombre de la métrica	Propósito de la Métrica	Método de Aplicación	Fórmula	Interpretación del valor obtenido
Subcaracterística: Idoneidad				
Adecuación funcional	<i>¿Cuán adecuadas son las funciones verificadas?</i>	<i>Contar la cantidad de funciones implementadas que son fáciles de entender para completar la tarea especificada, y luego se divide entre la cantidad total de funciones implementadas. Pueden ser medidas en: Todas o parte de las especificaciones</i>	$X = 1 - A/B$ <i>A: Número de funciones en las cuales se detectaron problemas en la evaluación. B: Número de funciones evaluadas.</i>	$0 \leq X \leq 1$ <i>A mayor cercanía al 1 resultará más adecuada.</i>

		de diseño. Módulos completados / componentes del producto de SW.		
Subcaracterística: Exactitud				
Exactitud	¿Cuán completa estuvo la implementación de los niveles de precisión específicos para los elementos de datos?	Contar la cantidad de elementos de datos que tienen niveles específicos de precisión y compararlo con el total de elementos de datos que lo requerían según la especificación.	$X = A/B$ A: Número de datos implementados con los niveles de precisión especificados, confirmado en la evaluación. B: Número de datos que requirieron los niveles de precisión.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más adecuada.
Subcaracterística: Interoperabilidad				
Esta Sub-característica no se medirá porque aún no se ha probado el intercambio de datos del SIIPOL con otros sistemas.				
Subcaracterística: Seguridad				
Controlabilidad de acceso	¿Cuán controlable es el acceso al sistema?	Contar la cantidad de requerimientos de control de acceso que fueron implementados	$X = 1 - A/B$ A: Número de CU que presentan problemas con	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más controlable.

		correctamente y compararlo con la cantidad de requerimientos de control de acceso descritos en la especificación.	el control de acceso. B: Número total de CU probados.	
--	--	---	--	--

Tabla de Métricas de Usabilidad

MÉTRICAS DE USABILIDAD				
Nombre de la métrica	Propósito de la Métrica	Método de Aplicación	Fórmula	Interpretación del valor obtenido
Subcaracterística: Comprensibilidad				
Entendimiento de funciones	¿Qué proporción de las funciones del producto será el usuario capaz de comprender correctamente?	Contar el número de interfaces de usuario donde su propósito es el entendimiento del usuario y comparar con el número de funciones de interfaces de usuarios.	$X = A/B$ A: Número de funciones de interfaz de usuario cuyo propósito es entendido por el usuario. B: Número total de funciones de interfaz de usuario.	$0 \leq X \leq 1$ Mientras más cercano a 1, mejor.
Subcaracterística: Operabilidad				
Chequeo de la Validez de la entrada	¿Qué proporción de	Contar el número de elementos de	$X = A/B$ A: Número de	$0 \leq X \leq 1$ Mientras más

	número de elementos de entrada chequean validez de los datos?	entrada, que chequean la validez de los datos y compara con el número de elementos de entrada, que podría verificar para datos válidos.	elementos de entrada que chequean validez de los datos. B: Número de elementos de entrada que pueden chequear la validez de los datos.	cercano a 1, mejor.
--	---	---	---	---------------------

Tabla de Métricas de Eficiencia

MÉTRICAS DE EFICIENCIA				
Nombre de la métrica	Propósito de la Métrica	Método de Aplicación	Fórmula	Interpretación del valor obtenido
Subcaracterística: Rendimiento				
Rendimiento	¿Cuál es el número estimado de tareas que pueden ser realizadas sobre una unidad de tiempo?	Evaluar la eficiencia de manejar recursos en el sistema. Haga un factor basado sobre las llamadas del uso al sistema en la manipulación de los recursos.	$X = \text{Número de tareas por unidad de tiempo}$	Mientras mayor mejor.

Subcaracterística: Utilización de recursos				
Uso de memoria	<i>¿Cuál es el tamaño de memoria estimado que el producto ocupará para completar una tarea específica?</i>	<i>Estimar los requerimientos de memoria.</i>	<i>X = Tamaño en bytes (calculado o simulado).</i>	<i>Mientras menor sea, mejor.</i>

Anexo 2 Escala para cuando se acerca a 0 y 1

Escala para cuando se acerca a 1

0 - 0,4	Mal	0
0,4 - 0,7	Regular	1
0,7 - 0,9	Bien	2
0,9 - 1	Muy Bien	3

Escala para cuando se acerca a 0

0 - 0,1	Muy bien	3
0,1 - 0,3	Bien	2
0,3 - 0,6	Regular	1
0,6 - 1	Mal	0

Anexo 3 Historias de Usuario

Historia de usuario “Mostrar los resultados de las encuestas”.

<i>Historia de Usuario</i>	
Número: 6	Nombre: Mostrar los resultados de las encuestas.
Usuario: Administrador de calidad.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Descripción: <ul style="list-style-type: none"> ➤ El sistema mostrará un informe de pre-evaluación. ➤ El usuario podrá guardar el informe de pre-evaluación. 	
Observaciones:	

Historia de usuario “Mostrar métricas”.

<i>Historia de Usuario</i>	
Número: 7	Nombre: Mostrar métricas.
Usuario: Administrador de calidad.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Descripción: <ul style="list-style-type: none"> ➤ El sistema muestra las métricas y brinda la opción para insertar los datos de las mismas. 	
Observaciones:	

Historia de usuario “Calcular métricas”.

<i>Historia de Usuario</i>	
----------------------------	--

Número: 8	Nombre: Calcular métricas.
Usuario: Administrador de calidad.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Descripción: ➤ El sistema calcula las métricas en correspondencia con la fórmula asociada a cada una de ellas. (Ver Anexo 1).	
Observaciones:	

Historia de usuario “Generar resultado de métricas”.

<i>Historia de Usuario</i>	
Número: 9	Nombre: Generar resultado de métricas.
Usuario: Administrador de Calidad.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Descripción: ➤ El sistema procesará los resultados de las métricas evaluadas.	
Observaciones:	

Historia de usuario “Mostrar resultados de métricas”.

<i>Historia de Usuario</i>	
Número: 10	Nombre: Mostrar resultados de métricas
Usuario: Administrador de calidad.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Bajo.

Puntos estimados: 1	Iteración asignada: 2
Descripción: ➤ El sistema muestra los resultados de las métricas.	
Observaciones:	

Historia de usuario “Mostrar la puntuación asignada a las sub-características”.

<i>Historia de Usuario</i>	
Número: 11	Nombre: Mostrar la puntuación asignada a las sub-características.
Usuario: Administrador del sistema.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Descripción: ➤ El sistema muestra la puntuación asignada a las sub-características.	
Observaciones: ➤ Los resultados recopilados de las métricas se ajustan a una escala con valores entre 0 y 3. (Ver Anexo 2)	

Historia de usuario “Mostrar el cálculo de la puntuación de las características.”

<i>Historia de Usuario</i>	
Numero: 12	Nombre: Mostrar el cálculo de la puntuación de las características.
Usuario: Administrador de calidad.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Bajo.
Puntos estimados: 1	Iteración asignada: 3
Descripción:	

- El sistema muestra el cálculo de la puntuación de las características.

Observaciones:

Historia de usuario “Mostrar Grado de Conformidad.”

<i>Historia de Usuario</i>	
Número: 13	Nombre: Mostrar Grado de Conformidad.
Usuario: Administrador de calidad.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 3
Descripción: <ul style="list-style-type: none"> ➤ El sistema asigna el grado de conformidad y luego lo muestra. 	
Observaciones:	

Historia de usuario “Mostrar criterio de evaluación.”

<i>Historia de Usuario</i>	
Número: 14	Nombre: Mostrar criterio de evaluación.
Usuario: Administrador de calidad.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Bajo.
Puntos estimados: 1	Iteración asignada: 3
Descripción: <ul style="list-style-type: none"> ➤ El sistema asigna el criterio de evaluación y luego lo muestra. 	
Observaciones:	

Historia de usuario “Gestionar informe de las no conformidades.”

<i>Historia de Usuario</i>	
----------------------------	--

Número: 15	Nombre: Gestionar informe de las no conformidades.		
Usuario: Administrador de calidad.			
Prioridad en negocio: Medio.		Riesgo en desarrollo: Bajo.	
Puntos estimados: 1		Iteración asignada: 3	
Descripción: <ul style="list-style-type: none"> ➤ El sistema muestra un formulario a llenar, teniendo en cuenta los principales problemas que fueron detectados durante el proceso de evaluación y la propuesta de acciones correctivas. ➤ El sistema brinda la opción de almacenar la información. ➤ El sistema muestra el informe de evaluación del producto software. 			
Observaciones:			

Historia de usuario "Exportar a pdf"

<i>Historia de Usuario</i>			
Número: 16	Nombre: Exportar a pdf.		
Usuario: Administrador de calidad.			
Prioridad en negocio: Bajo.		Riesgo en desarrollo: Medio.	
Puntos estimados: 1/2		Iteración asignada: 3	
Descripción: <ul style="list-style-type: none"> ➤ El sistema brinda la opción de exportar a pdf. 			
Observaciones:			

Anexo 4 Tarjetas CRC

Tarjeta CRC. Autenticar Usuario

<i>Tarjeta CRC. Autenticar Usuario</i>	
FUNCIONALIDADES	COLABORACIONES

Autenticar usuario	Autenticar usuario
--------------------	--------------------

Tarjeta CRC. Gestionar Usuario

<i>Tarjeta CRC. Gestionar Usuario</i>	
FUNCIONALIDADES	COLABORACIONES
Crear usuario Modificar usuario Eliminar usuario	Autenticar usuario

Tarjeta CRC. Gestionar Proyecto

<i>Tarjeta CRC. Gestionar Proyecto</i>	
FUNCIONALIDADES	COLABORACIONES
Crear proyecto Modificar proyecto Eliminar proyecto	Autenticar usuario

Tarjeta CRC. Realizar encuesta

<i>Tarjeta CRC. Realizar Encuesta</i>	
FUNCIONALIDADES	COLABORACIONES
Realizar encuesta	Autenticar usuario Gestionar proyecto

Tarjeta CRC. Procesar encuesta

<i>Tarjeta CRC. Procesar Encuesta</i>	
FUNCIONALIDADES	COLABORACIONES
Procesar encuesta	Autenticar usuario Gestionar proyecto Realizar encuesta

Tarjeta CRC. Mostrar los resultados de la encuesta

<i>Tarjeta CRC. Mostrar los Resultados de la Encuesta</i>	
FUNCIONALIDADES	COLABORACIONES

Mostrar resultado de la encuesta	Autenticar usuario Gestionar proyecto Realizar encuesta Procesar encuesta
----------------------------------	--

Tarjeta CRC. Mostrar métricas

<i>Tarjeta CRC. Mostrar Métricas</i>	
FUNCIONALIDADES	COLABORACIONES
Mostrar métricas	Autenticar usuario Gestionar proyecto Realizar encuesta Procesar encuesta Mostrar resultado de la encuesta

Tarjeta CRC. Calcular métricas

<i>Tarjeta CRC. Calcular Métricas</i>	
FUNCIONALIDADES	COLABORACIONES
Calcular métricas	Autenticar usuario Gestionar proyecto Mostrar métricas

Tarjeta CRC. Generar resultado de las métricas

<i>Tarjeta CRC. Generar Resultado de las Métricas</i>	
FUNCIONALIDADES	COLABORACIONES
Generar resultado de las métricas	Autenticar usuario Gestionar proyecto Realizar encuesta Procesar encuesta Mostrar resultado de la encuesta Mostrar métricas

Tarjeta CRC. Mostrar resultado de métricas

<i>Tarjeta CRC. Mostrar Resultado de Métricas</i>

FUNCIONALIDADES	COLABORACIONES
Mostrar resultado de las métricas	Autenticar usuario Generar resultado de las métrica

Tarjeta CRC. Mostrar la puntuación asignada a sub-características

<i>Tarjeta CRC. Mostrar la Puntuación Asignada a Sub-características</i>	
FUNCIONALIDADES	COLABORACIONES
Mostrar puntuación asignada a sub-características	Autenticar usuario Generar resultado de las métricas

Tarjeta CRC. Mostrar el cálculo de la puntuación de la característica

<i>Tarjeta CRC. Mostrar el Cálculo de la Puntuación de la Característica</i>	
FUNCIONALIDADES	COLABORACIONES
Mostrar el cálculo de la puntuación asignada de las características	Autenticar usuario Generar resultado de las métricas Mostrar la puntuación asignada a sub-características

Tarjeta CRC. Mostrar grado de conformidad

<i>Tarjeta CRC. Mostrar Grado de Conformidad</i>	
FUNCIONALIDADES	COLABORACIONES
Mostrar grado de conformidad	Autenticar usuario Generar resultado de las métricas Mostrar el cálculo de la puntuación asignada de las características

Tarjeta CRC. Mostrar criterio de evaluación

<i>Tarjeta CRC. Mostrar Criterio de Evaluación</i>	
FUNCIONALIDADES	COLABORACIONES
Mostrar criterio de evaluación	Autenticar usuario Generar resultado de las métricas Mostrar el cálculo de la puntuación asignada

	de las características Mostrar grado de conformidad
--	--

Tarjeta CRC. Gestionar informe de las no conformidades

<i>Tarjeta CRC. Gestionar Informe de las no Conformidades</i>	
FUNCIONALIDADES	COLABORACIONES
Crear informe Eliminar informe	Autenticar usuario Mostrar el cálculo de la puntuación asignada de las características Mostrar grado de conformidad Mostrar criterio de evaluación

Anexo 5 Casos de pruebas

Caso de Prueba HU1_P1 “Autenticar usuario con permiso”.

<i>Casos de Pruebas</i>	
Código: HU1_P1	Historia de Usuario: Autenticar Usuario.
Nombre: Autenticar usuario con permiso.	
Descripción: Prueba de funcionalidad autenticar usuarios sin permisos.	
Condiciones de ejecución: Se debe activar el módulo <i>LDAP_Integration</i> y configurado el sub-módulo <i>Idapauth</i> . El usuario debe tener permiso para acceder a la aplicación.	
Entrada/Pasos de ejecución: Se insertan nombre de usuario y contraseña y se ejecuta la acción autenticar usuario.	
Resultado esperado: El usuario es autenticado correctamente.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU1_P2 “Autenticar usuario sin permiso”.

<i>Casos de Pruebas</i>	
Código: HU1_P2	Historia de Usuario: Autenticar Usuario.

Nombre: Autenticar usuarios sin permisos.
Descripción: Prueba de funcionalidad autenticar usuarios sin permisos.
Condiciones de ejecución: Se debe activar el módulo <i>LDAP_Integration</i> y configurado el sub-módulo <i>ldapauth</i> . El usuario no debe tener permiso para acceder a la aplicación.
Entrada/Pasos de ejecución: Se insertan nombre de usuario y contraseña y se ejecuta la acción autenticar usuario.
Resultado esperado: El sistema deniega la petición del usuario a autenticarse.
Evaluación de la prueba: Excelente.

Caso de Prueba HU2_P1 “Adicionar datos de usuario”.

<i>Casos de Pruebas</i>	
Código: HU2_P1	Historia de Usuario: Gestionar usuario.
Nombre: Adicionar datos de usuario.	
Descripción: Prueba de funcionalidad adicionar datos usuario.	
Condiciones de ejecución: El usuario debe de estar registrado en el módulo <i>LDAP_Integration</i> .	
Entrada/Pasos de ejecución: Se procede a la inserción de los datos necesarios para adicionar un usuario, los cuales son: nombre de usuario, dirección de correo electrónico, contraseña, se especifica el estado (activado o bloqueado).	
Resultado esperado: El usuario se adiciona correctamente.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU2_P2 “Eliminar datos de usuario”.

<i>Casos de Pruebas</i>	
Código: HU2_P2	Historia de Usuario: Gestionar usuario.
Nombre: Eliminar datos de usuario.	
Descripción: Prueba de funcionalidad eliminar datos de usuario.	
Condiciones de ejecución: Los datos del usuario a eliminar deben de estar registrados en la base de datos de la aplicación.	
Entrada/Pasos de ejecución: Se procede la eliminación de los datos los cuales son: nombre de usuario, dirección de correo electrónico, contraseña y el estado.	
Resultado esperado: Se elimina correctamente los datos del usuario.	

Evaluación de la prueba: Excelente.
--

Caso de Prueba HU2_P3 “Modificar datos de

<i>Casos de Pruebas</i>	
Código: HU2_P3	Historia de Usuario: Gestionar usuario.
Nombre: Modificar datos de usuario.	
Descripción: Prueba de funcionalidad modificar datos de usuario.	
Condiciones de ejecución: El usuario al que se le quiere modificar los datos debe de estar registrado y con los permisos pertinentes en la base de datos de la aplicación.	
Entrada/Pasos de ejecución: Se modifican los datos necesarios y se ejecuta la acción de modificar datos.	
Resultado esperado: Se modifican los datos correctamente.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU3_P1 “Adicionar proyecto”.

<i>Casos de Pruebas</i>	
Código: HU3_P1	Historia de Usuario: Gestionar proyecto.
Nombre: Adicionar proyecto.	
Descripción: Prueba de funcionalidad adicionar proyecto.	
Condiciones de ejecución: El proyecto a adicionar no puede estar registrado en la base de datos.	
Entrada/Pasos de ejecución: Se insertan los datos del proyecto a adicionar, los cuales son: nombre del proyecto, y estado que puede ser activo o inactivo.	
Resultado esperado: Se inserta correctamente el proyecto a evaluar.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU3_P2 “Eliminar proyecto”.

<i>Casos de Pruebas</i>	
Código: HU3_P2	Historia de Usuario: Gestionar proyecto.
Nombre: Eliminar proyecto.	
Descripción: Prueba de funcionalidad eliminar proyecto.	
Condiciones de ejecución: El proyecto a eliminar debe de estar registrado en la base de datos.	

Entrada/Pasos de ejecución: Seleccionar el nombre del proyecto a eliminar y ejecutar la acción eliminar proyecto.
Resultado esperado: Se elimina el proyecto correctamente.
Evaluación de la prueba: Excelente.

Caso de Prueba HU3_P3 “Modificar proyecto”.

<i>Casos de Pruebas</i>	
Código: HU3_P3	Historia de Usuario: Gestionar proyecto.
Nombre: Modificar proyecto.	
Descripción: Prueba de funcionalidad modificar proyecto.	
Condiciones de ejecución: El proyecto a modificar debe estar registrado en la base de datos de la aplicación.	
Entrada/Pasos de ejecución: Seleccionar el proyecto a modificar y ejecutar la acción modificar proyecto.	
Resultado esperado: Se modifican los datos del proyecto satisfactoriamente.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU4_P1 “Comprobar la correcta entrada de datos”.

<i>Casos de Pruebas</i>	
Código: HU4_P1	Historia de Usuario: Realizar encuesta.
Nombre: Comprobar la correcta entrada de datos.	
Descripción: Prueba de funcionalidad realizar encuesta.	
Condiciones de ejecución: El usuario a realizar la encuesta debe de estar autenticado y con los permisos necesarios para realizar la encuesta. Se debe seleccionar la opción realizar encuesta.	
Entrada/Pasos de ejecución: Entrar los valores de la encuesta y ejecutar la acción enviar encuesta y se almacenan los resultados obtenidos.	
Resultado esperado: Se realiza la encuesta correctamente.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU6_P1 “Mostrar resultados”.

<i>Casos de Pruebas</i>

Código: HU6_P1	Historia de Usuario: Mostrar los resultados de las encuestas.
Nombre: Mostrar resultados de la encuesta.	
Descripción: Prueba de funcionalidad mostrar resultados.	
Condiciones de ejecución: Se debe haber realizado la encuesta.	
Entrada/Pasos de ejecución: Se ejecuta la acción mostrar resultado de las encuestas.	
Resultado esperado: Se muestra los resultados de la encuesta realizada.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU7_P1 “Mostrar las métricas”.

<i>Casos de Pruebas</i>	
Código: HU7_P1	Historia de Usuario: Mostrar métricas.
Nombre: Mostrar las métricas.	
Descripción: Prueba de funcionalidad mostrar las métricas.	
Condiciones de ejecución: Se debe haber realizado la encuesta.	
Entrada/Pasos de ejecución: Se ejecuta la acción mostrar métricas y se insertan los datos de las mismas.	
Resultado esperado: Se muestra las métricas y se insertan los valores correctamente.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU8_P1 “Calcular las métricas”.

<i>Casos de Pruebas</i>	
Código: HU8_P1	Historia de Usuario: Calcular métricas.
Nombre: Calcular las métricas.	
Descripción: Prueba de funcionalidad calcular las métricas.	
Condiciones de ejecución: Se debe haber realizado la encuesta y haber introducido las métricas.	
Entrada/Pasos de ejecución: Se ejecuta la acción calcular métricas.	
Resultado esperado: El sistema calcula las métricas en correspondencia con la fórmula asociada a cada una de ellas.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU9_P1 “Procesar resultados”.

<i>Casos de Pruebas</i>	
Código: HU9_P1	Historia de Usuario: Generar resultado de métricas.
Nombre: Procesar resultados de las métricas.	
Descripción: Prueba de funcionalidad procesar resultados.	
Condiciones de ejecución: Se debe haber realizado la encuesta y la evaluación de las métricas.	
Entrada/Pasos de ejecución: Se ejecuta la acción resultado de métricas.	
Resultado esperado: El sistema procesará los resultados de las métricas evaluadas.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU10_P1 “Mostrar métricas”.

<i>Casos de Pruebas</i>	
Código: HU10_P1	Historia de Usuario: Mostrar resultado de métricas.
Nombre: Mostrar resultado de las métricas.	
Descripción: Prueba de funcionalidad mostrar métricas.	
Condiciones de ejecución: Se debe haber realizado la encuesta, introducido los valores de las métricas y procesado los mismos.	
Entrada/Pasos de ejecución: Se ejecuta la acción mostrar resultados de métricas.	
Resultado esperado: El sistema muestra los resultados de las métricas.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU11_P1 “Comprobar el correcto cálculo de la puntuación asignada”.

<i>Casos de Pruebas</i>	
Código: HU11_P1	Historia de Usuario: Mostrar la puntuación asignada a las sub-características.
Nombre: Comprobar el correcto cálculo de la puntuación asignada.	
Descripción: Prueba de funcionalidad puntuación asignada.	
Condiciones de ejecución: Se ajustan las métricas a una escala con valores entre 0 y 3.	
Entrada/Pasos de ejecución: Se ejecuta la acción mostrar puntuación asignada a las sub-características.	
Resultado esperado: Se muestra correctamente la puntuación de las sub-características.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU12_P1 “Cálculo de puntuación a características”.

<i>Casos de Pruebas</i>	
Código: HU12_P1	Historia de Usuario: Mostrar el cálculo de la puntuación de las características.
Nombre: Cálculo de puntuación de las características.	
Descripción: Prueba de funcionalidad cálculo de puntuación a características.	
Condiciones de ejecución: Se debe haber realizado la encuesta y haber seleccionado las métricas a evaluar.	
Entrada/Pasos de ejecución: Se ejecuta la acción mostrar cálculo de puntuación de las características.	
Resultado esperado: Se muestra correctamente el cálculo de la puntuación de las características.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU13_P1 “Grado de conformidad”.

<i>Casos de Pruebas</i>	
Código: HU13_P1	Historia de Usuario: Mostrar Grado de Conformidad.
Nombre: Grado de conformidad.	
Descripción: Prueba de funcionamiento grado de conformidad.	
Condiciones de ejecución: Se debe haber asignado la puntuación de las características.	
Entrada/Pasos de ejecución: Se ejecuta la acción de mostrar grado de conformidad.	
Resultado esperado: Se muestra el grado de conformidad.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU14_P1 “Criterio de evaluación”.

<i>Casos de Pruebas</i>	
Código: HU14_P1	Historia de Usuario: Mostrar criterio de evaluación.
Nombre: Criterio de evaluación.	
Descripción: Prueba de funcionamiento criterio de evaluación.	
Condiciones de ejecución: Se debe haber asignado el grado de conformidad.	
Entrada/Pasos de ejecución: Se ejecuta la acción de mostrar criterio de evaluación.	

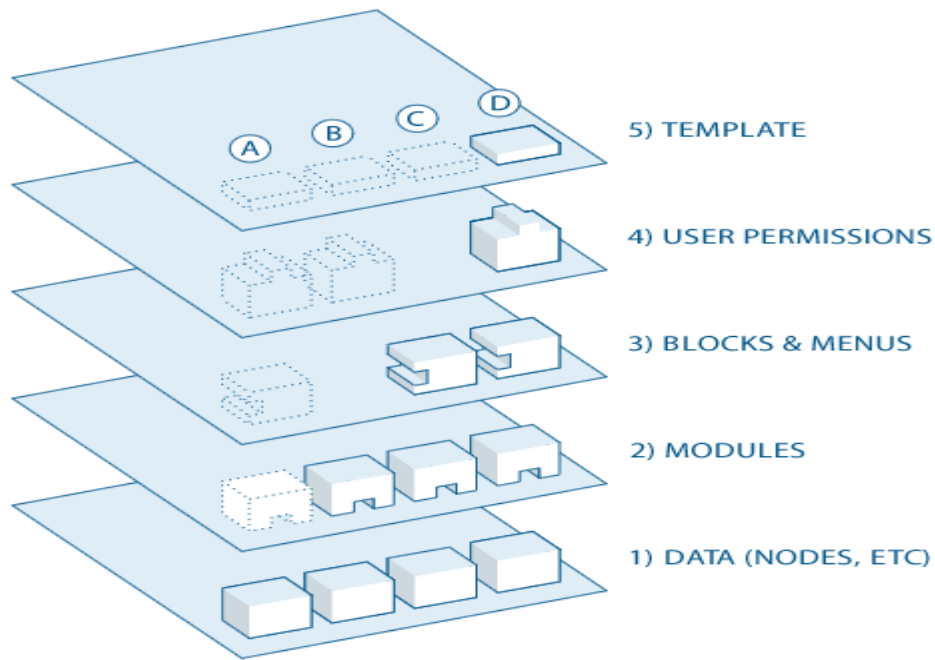
Resultado esperado: Se muestra el criterio de evaluación.
Evaluación de la prueba: Excelente.

Caso de Prueba HU15_P1 “Informe de no conformidad”.

<i>Casos de Pruebas</i>	
Código: HU15_P1	Historia de Usuario: Gestionar informe de las no conformidades.
Nombre: Informe de no conformidad.	
Descripción: Prueba de funcionalidad informe de no conformidad.	
Condiciones de ejecución: Tener en cuenta los principales inconvenientes que fueron detectados durante el proceso de evaluación. El sistema muestra un formulario a llenar.	
Entrada/Pasos de ejecución: Llenar el formulario de las no conformidades con los inconvenientes detectados. Ejecutar la acción enviar	
Resultado esperado: Se envía correctamente la no conformidad.	
Evaluación de la prueba: Excelente.	

Caso de Prueba HU16_P1 “Exportar a pdf”.

<i>Casos de Pruebas</i>	
Código: HU16_P1	Historia de Usuario: Exportar a pdf.
Nombre: Exportar a pdf.	
Descripción: Pruebas de funcionalidad exportar a pdf.	
Condiciones de ejecución: Debe mostrarse la información.	
Entrada/Pasos de ejecución: Se ejecuta la acción exportar a pdf.	
Resultado esperado: Se muestra un documento en formato pdf con la información deseada.	
Evaluación de la prueba: Excelente.	



Flujo de la información en Drupal

Inicio de sesión

Usuario: *

Contraseña: *

Iniciar sesión

ECSoft

Mar, 05/24/2011 - 11:28 — Anónimo



Bienvenidos al Sistema para la Evaluación de la Calidad de un Producto Software, perteneciente al Centro de Identificación y Seguridad Digital (CISED), centro de excelencia que se dedica al desarrollo de productos, servicios y soluciones integrales de alta confiabilidad, precisión y eficiencia económica.

Enlaces

[Calisoft](#)

[Intranet](#)

[Google](#)

[Biblioteca](#)

El sistema tiene como principal meta elevar la calidad de los productos software que se desarrollan en el CISED, para una mayor satisfacción de los clientes, así como para lograr una mayor eficiencia en el desarrollo de los proyectos productivos y mantener el prestigio con que cuenta el Centro.

GLOSARIO DE TÉRMINOS Y DEFINICIONES

APLICACIÓN: Cualquier programa que corra en un sistema operativo y que haga una función específica para un usuario. Por ejemplo, procesadores de palabras, bases de datos, agendas electrónicas, entre otros.

BASE DE DATOS: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

CICED: (Centro de Identificación y Seguridad Digital) está encaminado a desarrollar productos, servicios y soluciones integrales en el campo de la identificación y la seguridad digital.

CMS: Acrónimo de *Content Management System*, sistema de gestión o administración de contenidos.

DRUPAL: Es un sistema de administración de contenido para sitios Web.

GOF: (*Gang Of Four*) es el nombre con el que se conoce comúnmente a los autores del libro *DesignPatterns*. Se conoce de esta manera a los patrones de diseño.

Hardware: Conjuntos de componentes que integran la parte material de una computadora. Componentes físicos de una tecnología.

HTML: *HyperText MarkupLanguage* (lenguaje de marcas hipertextuales), diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

HTTP: (*Hypertext Transfer Protocol*). Protocolo de nivel de aplicación usado extensivamente en Internet para el acceso a documentos.

HU: (Historias de Usuarios) artefactos que se definen en la Fase de Exploración de la metodología XP.

INFORME DE EVALUACIÓN: Es el informe que presenta los resultados de la evaluación y cualquier otra información relevante para la evaluación.

ITERACIÓN: Se refiere a la acción de repetir una serie de pasos un cierto número de veces.

LDAP: (*Lightweight Directory Access Protocol*) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

MÓDULO: Bloques de código que proveen extra funcionalidad o mejoras.

Mozilla Firefox: Navegador de Internet libre y de código abierto.

PDF: *Portable Document Format*, (formato de documento portátil) es un formato de almacenamiento de documentos.

PHP: Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

XP: (*Extreme Programming*) Metodología de desarrollo de software basada en valores como simplicidad, comunicación, retroalimentación y coraje.