

**Universidad de las Ciencias Informáticas**

**Facultad 1**



Título: Sistema para la evaluación de la seguridad de un  
software

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

Autores:

Lieny Rodríguez Rodríguez  
Yaciel Leyva Góngora

Tutores

Ing. Yayneris Zambrana Hernández  
Ing. Geidis Sánchez Michell

**La Habana, Junio 2013  
"Año 55 de la Revolución"**



*“Podrán morir las personas, pero jamás sus ideas”*

*Ernesto Che Guevara*

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: "Sistema para la evaluación de la seguridad de un software", y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

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

\_\_\_\_\_  
Lieny Rodríguez Rodríguez

\_\_\_\_\_  
Yaciel Leyva Góngora

\_\_\_\_\_  
Ing. Yayneris Zambrana Hernández

\_\_\_\_\_  
Ing. Geidis Sánchez Michel

*Dedico este trabajo a mis padres, en especial a mi papá, mi luz y ejemplo, por cada uno de sus consejos, por cada pedacito de felicidad que viví a su lado y por cada momento que no podremos compartir físicamente.*

*A mi mami, que tanto sacrificio ha hecho para que yo cumpliera esta meta, este logro es tuyo.*

*A toda mi familia, en el sentido más general de esta palabra, a ustedes que siempre me apoyaron y pensaron en la niña de Félix y María Flena.*

*A todos los amigos que sin decir una palabra me ayudaron a crecer, hoy tengo el placer de dedicarle este trabajo.*

*Fieny Rodríguez Rodríguez*

*Este trabajo se lo dedico a dos personas que han sido muy especiales en mi vida, primeramente a mi mamita por ser madre y padre durante todo este tiempo, por ser mi mejor amiga en todos los momentos difíciles,*

*por darme fuerzas y confiar en mí desde el momento en que le dije que quería ser Ingeniero.*

*A mi abuelo por ser mi primer maestro, por enseñarme las tablas de multiplicación, por llenar mis venas de amor por la patria, por sus bellas anécdotas, por su entusiasmo, por su dedicación y apoyo en todo mi proceso educativo.*

*Yaciel Leyva Góngora*

*Agradezco a mis padres, por toda la felicidad que he vivido, por el orgullo de ser su hija, por ser mi meta para ser mejor cada día, por cada consejo y cada regaño, gracias, mil gracias.*

*A mi padre, gracias por ser tan grande, por vivir para mí, por siempre apoyarme, llegue a ti, donde quiera que estés, el beso con más amor y dolor que una hija le pueda dar a su padre.*

*Mi más especial agradecimiento es para mi madre, tú eres mi Mariana Grajales, tuvo que llover mucho para comprender cada regaño, cada prohibición, cada consejo y hoy te lo agradezco infinitamente. Sin cada una de tus sonrisas, de tus lágrimas, hoy tu hija no fuera ingeniera. Gracias por simplemente ser mi amiga, porque para el mundo eres mi madre pero para mí eres el mundo.*

*A toda mi familia, que siempre me ha sacado la sonrisa, que me ha apoyado en los momentos más difíciles*

*A mis hermanas de corazón, Danmy y Yuri, por sembrar en mí este cariño especial.*

*A todos mis amigos del carapacho, del politécnico y la universidad, a todos mis más sinceros agradecimientos, siempre tendrán un pedacito en mi corazón.*

*A mis compañeros, a cada persona que compartió conmigo y se preocupó por mí.*

*A todos los que de un modo u otro aportaron a la realización de esta aspiración.*

*Lieny Rodríguez Rodríguez*

*Agradecer infinitamente a mi mamá por ser más que una madre una amiga; por haberme formado bajo principios de honestidad, gracias mamita por estar siempre ahí cuando más te necesito.*

*A mis hermanas por ser el motor impulsor que me ha permitido pasar todo este tiempo tan lejos de casa y presionado por momentos tan difíciles.*

*A mis queridos abuelos Fermín y Juana mis segundos padres a los que jamás olvidaré, a toda mi familia por apoyarme cuando todo parecía no tener salida.*

*A mi linda novia Alisnay por ser tan paciente conmigo y darme todo lo que estuvo en sus manos durante todo este tiempo.*

*A mis amigos Alfre, Jorge, Yasmani, Omar, Rede, Atiel, Royli, Fliza, Billy y todos los demás que forman parte de mi vida.*

*A todos mis colegas que desde primer año compartimos el mismo local y grandes vivencias.*

*Yaciel Leyva Góngora*

*Agradecer a nuestras tutoras, sin ustedes este trabajo no hubiese sido posible, gracias por cada momento tenso en el que siempre nos mostraron la claridad cuando todo parecía oscuro. Gracias por cada consejo, cada sugerencia, cada corrección, gracias por defendernos en todo momento.*

*A los profes que en algún momento nos hicieron sentir orgullosos por formar parte de esta obra de la Revolución, por motivarnos como estudiantes y contagiarnos con su conocimiento y pedagogía.*

*A todas las personas que nos preguntaron por la tesis y las que nos brindaron su ayuda, sin esperar nada a cambio.*

*A todos, nuestros agradecimientos.*

*Los autores*

En este trabajo se presenta un análisis del proceso de evaluación de la calidad de un software establecido por el estándar ISO/IEC 14598. A través de este estudio se identifican las etapas y actividades para realizar la evaluación, así como las métricas para medir la seguridad del mismo, factores necesarios para realizar dicha evaluación y conocer el nivel de seguridad del software evaluado.

Se tiene como objetivo general potenciar y flexibilizar el sistema de evaluación de la seguridad de un software mediante el uso de métricas, contribuyendo a la mejora del proceso según los estándares internacionales establecidos. Como parte de la propuesta de solución se define la herramienta necesaria para realizar las pruebas de seguridad a un software. También se describe en qué consiste el proceso ETL (Extracción, Transformación y Carga), utilizado para la integración de los datos arrojados por las pruebas de seguridad realizadas. Además, se establecen las tecnologías, herramientas y lenguajes para el desarrollo de la solución siendo guiada por la metodología *Microsoft Solution Framework for Agile Software Development*.

En el trabajo se muestran los resultados de las pruebas al Sistema para la evaluación de la seguridad desarrollado, corroborando la validez del sistema y garantizando un producto confiable, seguro y eficiente. Con la obtención de un producto de calidad se podrá analizar y controlar la seguridad de los proyectos productivos del Centro de Identificación y Seguridad Digital (CISED).

**Palabras clave:** Calidad, Estándar, Métrica, Proceso de evaluación, Seguridad.

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....</b>	<b>5</b>
1.1 Calidad .....	5
1.2 Análisis de los estándares internacionales .....	5
1.3 Métricas de seguridad .....	8
1.4 Proceso de evaluación .....	11
1.5 Análisis de las herramientas de pruebas de seguridad .....	13
1.6 Descripción de las características del Sistema de evaluación de la seguridad v 1.0.....	14
1.7 Entorno de desarrollo .....	15
1.7.1 Metodología de desarrollo Microsoft Solution Framework for Agile Software Development.....	15
1.7.2 Lenguaje de modelado Unified Modeling Language (UML).....	17
1.7.3 Herramientas CASE Visual Paradigm .....	17
1.7.4 Servidor web Apache .....	18
1.7.5 Sistema de Gestión de Contenidos Drupal.....	19
1.7.6 Lenguajes para el desarrollo web.....	20
1.7.7 Entorno Integrado de Desarrollo NetBeans.....	22
1.7.8 Sistema Gestor de Base de Datos MySQL .....	22
1.7.9 Herramienta de desarrollo para el proceso ETL. Pentaho Data Integration.....	23
1.7.10 Proceso de integración de datos ETL.....	24
Conclusiones del capítulo.....	25
<b>CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA .....</b>	<b>26</b>
2.1 Fase visión y alcance .....	26
2.1.1 Propuesta de solución.....	26
2.2 Fase de planificación.....	28
2.2.1 Escenarios del sistema .....	29
2.2.2 Priorización de los escenarios.....	30
2.2.3 Requisitos de calidad del servicio .....	30
2.2.4 Plan de iteraciones.....	31
2.3 Descripción de los escenarios .....	32
2.3.1 Especificaciones de tareas por escenario .....	33
2.4 Elementos del diseño del sistema .....	35
2.4.1 Especificación de la arquitectura a utilizar.....	35
2.4.2 Patrones de diseño .....	37
2.4.3 Modelo de datos .....	38
2.4.4 Descripción de las tablas .....	39



Conclusiones del capítulo.....	41
<b>CAPÍTULO 3: DESARROLLO Y ESTABILIZACIÓN DEL SISTEMA.....</b>	<b>42</b>
3.1 Fase de desarrollo.....	42
3.1.1 Estándares de codificación .....	42
3.1.2 Diagrama de despliegue .....	43
3.1.3 Interfaz gráfica .....	44
3.2 Fase de estabilización .....	46
3.2.1 Pruebas unitarias. Aplicación de pruebas de caja blanca.....	47
3.2.2 Aplicación de pruebas de caja negra .....	49
3.2.3 Resultado de las pruebas.....	50
3.3 Análisis del cumplimiento del sistema.....	51
Conclusiones del capítulo.....	52
<b>CONCLUSIONES GENERALES .....</b>	<b>53</b>
<b>RECOMENDACIONES .....</b>	<b>54</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>55</b>
<b>BIBLIOGRAFÍA.....</b>	<b>57</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>59</b>

Tabla 1: Descripción de las métricas aplicables al sistema .....	8
Tabla 2: Escala para la evaluación .....	11
Tabla 3: Herramientas de pruebas de seguridad .....	13
Tabla 4: Definición de personas .....	28
Tabla 5: Priorización de los escenarios.....	30
Tabla 6: Planificación de los escenarios .....	32
Tabla 7: Descripción del escenario “Gestionar reporte de evaluación” .....	32
Tabla 8: Descripción de la tarea “Mostrar reporte de evaluación” .....	33
Tabla 9: Descripción de la tarea “Exportar reporte a formato pdf” .....	34
Tabla 10: Descripción de la tabla “proyectos” .....	39
Tabla 11: Descripción de la tabla “historial” .....	40
Tabla 12: Descripción de la tabla “proyectos_historial” .....	40
Tabla 13: Descripción de la tabla “datos_fichero” .....	40
Tabla 14: Descripción de la tabla “metricas” .....	41
Tabla 15: Descripción de la tabla “detalles_metricas” .....	41
Tabla 16: Descripción de la prueba unitaria a la función “modulo_evaluacion_get_proyecto” .....	48
Tabla 17: Descripción de la prueba unitaria a la función “cargar_datos_bd” .....	48
Tabla 18: Descripción del caso de prueba “Cargar fichero” .....	49

Figura 1: Pasos para realizar el proceso de evaluación según ISO/IEC 14598-5.....	12
Figura 2: Proceso de evaluación del SES v1.0 .....	15
Figura 3: Vista conceptual del sistema.....	28
Figura 4: Arquitectura de Drupal .....	36
Figura 5: Arquitectura del sistema.....	37
Figura 6: Modelo de datos del sistema.....	39
Figura 7: Diagrama de despliegue .....	44
Figura 8: Interfaz gráfica “Autenticar usuario” .....	45
Figura 9: Interfaz principal del sistema.....	45
Figura 10: Interfaz gráfica “Gestionar proyecto” .....	46
Figura 11: Prueba unitaria a la función “test_Realizar_escala” y “test_Modulo_evaluacion_nivel_general”	47
Figura 12: Iteraciones de pruebas.....	51

# INTRODUCCIÓN

La informática en todas las esferas de la vida es una realidad conocida; en particular, en la empresa moderna, ya sea productiva, comercial o de servicio, convirtiéndose poco a poco en una fuente científica fundamental para asegurar sus crecientes y complejas funciones. Hoy en día la sociedad está siendo rebasada por el rápido avance de las Tecnologías de la Información y las Comunicaciones (TIC), lo que trae como consecuencia que el desarrollo de software tenga una gran importancia, y que en el mundo informático exista una competencia en cuanto a la calidad y seguridad del software.

Y es que la calidad y la seguridad del software son campos que han tenido una fructífera actividad investigativa en los últimos años, estando principalmente centrada en la calidad de los procesos que se siguen para desarrollar el software. Prueba de ello es la gran cantidad de modelos y estándares de referencia, evaluación y mejora de procesos de software que han surgido durante las últimas décadas (Fernández, y otros, 2010). De ahí la aparición de estándares y métricas que ayuden al control y perfeccionamiento del software.

La Organización Internacional de Estándares, ISO por sus siglas en inglés, en conjunto con la IEC (Comisión Electrotécnica Internacional) en su estándar ISO/IEC 9126-1 define la calidad como un conjunto estructurado de características y sub-características, que además pueden ser medidas mediante el uso de métricas. Entre las características que se definen, se encuentra la funcionalidad: “Capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas” (Oficina Nacional de Normalización, 2005). A su vez dentro de esta característica, se define como sub-característica la seguridad: “Capacidad del producto de software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o modificar los mismos, y las personas o sistemas autorizados tengan el acceso a ellos” (Oficina Nacional de Normalización, 2005).

Para un mayor avance, en este sentido mundialmente existen disímiles centros destinados al desarrollo informático. En Cuba existen varias entidades que persiguen como objetivo la producción de software, entre ellos la Universidad de las Ciencias Informáticas (UCI). Dicha universidad cuenta con diferentes centros de desarrollo de software que tributan al avance científico-técnico del país, investigando y desarrollando software para diferentes áreas temáticas, entre los que se encuentra la seguridad digital.

Precisamente, esta línea de investigación se lleva a cabo en algunos centros de la universidad, uno de ellos es el Centro de Identificación y Seguridad Digital (CISED), el cual cuenta con un Departamento de Seguridad Digital que investiga y desarrolla proyectos o productos que tributan al aumento de su seguridad. Una de las líneas temáticas que investiga el departamento es la evaluación de la seguridad, en la cual se tiene como aporte un sistema que mediante la herramienta de prueba W3AF, obtiene resultados

de las pruebas de seguridad realizadas a un determinado software, y a estos le aplica métricas que contribuyen a obtener como resultado final el nivel de seguridad del software evaluado.

Para realizar el procesamiento de los datos y el cálculo de las métricas se utilizó el proceso de Extracción, Transformación y Carga (*Extract, Transform and Load*, por sus siglas en inglés ETL), con el objetivo de ahorrar tiempo y agilizar el trabajo con la utilización y el procesamiento de las métricas.

Dada la necesidad de potenciar el proceso de evaluación de la seguridad del software, con el fin de obtener mejores productos y con un elevado nivel de seguridad de los mismos, se hace necesario determinar cuáles son las deficiencias del sistema desarrollado anteriormente, y así poder definir e introducir las nuevas mejoras al mismo. El sistema anterior:

- No brinda la posibilidad de consultar un historial de evaluaciones; lo que implica que se desconozca el comportamiento de la seguridad a lo largo de todas las evaluaciones realizadas.
- Sólo tiene definido para la evaluación tres métricas, por lo que no se tienen indicadores suficientes para determinar cuan seguro es el software evaluado.
- No brinda la posibilidad de introducir nuevas métricas para evaluar la seguridad, limitando al evaluador a utilizar las que están predefinidas.
- Los reportes no presentan los datos suficientes que contribuyen a la toma de decisiones, además de que el diseño de los mismos no es el más adecuado.
- No cuenta con un módulo de notificación, por lo que el envío de los resultados se realiza de manera manual, lo que hace que el proceso de entrega sea más engorroso.
- No permite ver de manera gráfica el comportamiento de la seguridad de un proyecto, lo que hace que se invierta mayor tiempo en la interpretación de los resultados de cada reporte perteneciente a la evaluación del mismo.

Dada la problemática anteriormente planteada, se define como **problema de investigación**:

La evaluación de la seguridad que se realiza a un software con el sistema de evaluación existente, no abarca de manera automatizada todas las etapas del proceso de evaluación definido por los estándares internacionales establecidos, lo que provoca insatisfacción con el proceso.

El **objeto de estudio** responde al proceso de evaluación de la seguridad de un software.

Para darle solución al problema planteado, en la presente investigación se trazó como **objetivo general**:

Potenciar y flexibilizar el sistema de evaluación de la seguridad de un software mediante el uso de métricas, contribuyendo a la mejora del proceso según los estándares internacionales establecidos.

El cumplimiento del objetivo será posible con la realización de las **tareas de investigación** que a continuación se detallan:

1. Análisis de las herramientas de pruebas de seguridad a un software para seleccionar la que se utilizará.

2. Análisis de los referentes teórico-prácticos que preceden la realización del presente trabajo en relación con el proceso de evaluación de un software.
3. Análisis de las métricas para evaluar la seguridad.
4. Definición de la metodología, herramientas, tecnologías y lenguajes a utilizar para el desarrollo del sistema.
5. Confección de la vista conceptual del sistema para identificar los principales elementos que conforman la solución del problema planteado.
6. Definición de la arquitectura del sistema para establecer el diseño conceptual y la estructura operacional del mismo.
7. Definición de los patrones de diseño del sistema para lograr un diseño general uniforme.
8. Confección del modelo de datos para describir la estructura lógica de la información persistente manejada por el sistema.
9. Definición de los estándares de codificación del sistema para mantener uniformidad en el código del mismo.
10. Desarrollo del sistema para la evaluación de la seguridad en los proyectos productivos.
11. Diseño del Modelo de despliegue del sistema, para definir la arquitectura física de este durante la ejecución.
12. Diseño de los Casos de pruebas del sistema para verificar los requisitos funcionales del software.
13. Realización de pruebas de software al “Sistema para la evaluación de la seguridad” para garantizar que funcione óptimamente.

Los **posibles resultados** del presente trabajo son los siguientes:

- Un sistema que permitirá realizar una evaluación de la seguridad en un software mediante el uso de métricas.
- Documento con la especificación de los escenarios del sistema.
- Documento con la descripción de los requisitos de calidad del servicio del sistema.
- Un manual de usuario del sistema.

## **Justificación de la investigación**

Con el desarrollo de la presente investigación para el Departamento de Seguridad Digital, se podrá obtener un sistema que realice la evaluación de la seguridad de un software, basado en estándares internacionales que definen el proceso de evaluación en sí, además de seleccionar las métricas para medir los indicadores. Esta investigación surge con el objetivo de potenciar y flexibilizar el proceso de evaluación existente, dada la importancia que se le atribuye a la seguridad en el software y a la satisfacción de un cliente que espera de un producto la calidad máxima requerida.

Para el desarrollo de la investigación se decidió utilizar los **métodos científicos** que a continuación se detallan:

## **Métodos teóricos:**

- **Análítico-Sintético:** para analizar las diferentes maneras de evaluar la seguridad en las aplicaciones, realizando pruebas de seguridad y usando métricas definidas por estándares internacionales.
- **Modelación:** para modelar un conjunto de diagramas que representan el proceso de desarrollo, facilitando el diseño de la propuesta de solución.

## **Métodos empíricos:**

- **Entrevista:** Para abundar más sobre la utilización de métricas de seguridad, así como de herramientas de pruebas y su aplicación en el CISED.

El presente documento se encuentra estructurado en tres capítulos, los cuales se describen a continuación:

## **Capítulo 1: Fundamentación teórica**

En este capítulo se aborda el problema planteado según las bases teóricas. Se reflejan los conceptos esenciales relacionados con la calidad del software, seguridad informática, estándares, métricas y el proceso de evaluación. Además de estudiar las diferentes herramientas de pruebas de seguridad con el fin de seleccionar la idónea para ser utilizada. Así como un estudio del ambiente de desarrollo que permitirá desarrollar la propuesta de solución a la problemática existente.

## **Capítulo 2: Caracterización y diseño del sistema**

En este capítulo se describe y caracteriza la solución propuesta teniendo como punto de partida el análisis de la problemática planteada. Se especifica la arquitectura y los patrones de diseño a utilizar. Se realiza además una descripción de las funcionalidades que el sistema debe cumplir a través del levantamiento de requisitos.

## **Capítulo 3: Desarrollo y estabilización del sistema**

En este capítulo los estándares de codificación son establecidos a fin de lograr la implementación del sistema, se crean los diagramas necesarios para guiar la implementación, y finalmente se realizan las pruebas necesarias que validan la calidad de la solución.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se aborda el problema planteado según las bases teóricas. Se reflejan los conceptos esenciales relacionados con la calidad del software, seguridad informática, estándares, métricas y el proceso de evaluación. Además de estudiar las diferentes herramientas de pruebas de seguridad con el fin de seleccionar la idónea para ser utilizada. Así como un estudio del ambiente de desarrollo que permitirá desarrollar la propuesta de solución a la problemática existente.

## 1.1 Calidad

La calidad es un término que a través de los años ha estado en constante evolución, por su importancia en el desarrollo de productos competitivos en el mundo informático. En este sentido la ISO/IEC establece sus criterios respecto a la calidad en todos sus estándares, ejemplo de ellos: el estándar ISO/IEC 8042:1994, que define calidad como: “Conjunto de propiedades y características de un producto o servicio que le confieren capacidad para satisfacer necesidades expresadas o implícitas” (López, Ana María, 2008).

Por otra parte el estándar ISO/IEC 9000:2000 plantea la calidad como: “Grado en el que un conjunto de características inherentes cumple con los requisitos” (López, Ana María, 2008).

También en 1998, Roger S. Pressman aporta otra definición, esta dice: “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” ( Pressman, 1998).

De esta forma la calidad es uno de los elementos más importante para que un producto sea competitivo, esta requiere de tiempo y esfuerzo para satisfacer las necesidades del cliente, por lo que se puede concluir que calidad es sinónimo de eficiencia, confiabilidad y seguridad.

## 1.2 Análisis de los estándares internacionales

Según la ISO/IEC, un estándar (ISO, 2010) se define como un conjunto de acuerdos documentados que establece los requisitos, especificaciones, directrices o características que se pueden usar de manera constante para garantizar que los materiales, productos, procesos y servicios sean adecuados para su propósito. Para el apoyo de la investigación se tuvo en cuenta algunos de los estándares de calidad que permiten obtener la información necesaria para el desarrollo de la solución. A continuación se describen los usados con este propósito:

- Estándar ISO/IEC 9126.
- Estándar ISO/IEC 14598.
- Estándar ISO/IEC 25000.



## **Estándar ISO/IEC 9126**

El estándar ISO/IEC 9126 (ISO/IEC, 1999) fue lanzado en el año 1991. Creado con el fin de evaluar el software estableciendo un modelo de calidad, es supervisado por el proyecto Requisitos y Evaluación de Calidad de Productos de Software (por sus siglas en inglés SQuaRE) y pensado para los desarrolladores. Con una estructura que se encuentra dividida en cuatro partes, distribuidas de la siguiente forma:

- ISO/IEC 9126-1 Modelo de calidad.
- ISO/IEC 9126-2 Métricas externas.
- ISO/IEC 9126-3 Métricas internas.
- ISO/IEC 9126-4 Métricas de calidad en el uso.

Dentro de las características que se proponen en la ISO/IEC 9126-1 para la calidad y evaluación de un producto, se definen:

- Funcionalidad.
- Fiabilidad.
- Usabilidad.
- Eficiencia.
- Mantenibilidad.
- Portabilidad.

Cada una de estas características define paralelamente una serie de sub-características. El tema principal del presente trabajo es la seguridad, y precisamente es una sub-característica que pertenece a la característica funcionalidad dentro de la cual se evalúa el grado en que el software satisface las necesidades. Y es que seguridad informática (DECRETO-LEY No.199, 1999) es el conjunto de medidas administrativas, organizativas, físicas, técnicas, legales y educativas dirigidas a prevenir, detectar y responder a las acciones que puedan poner en riesgo la confidencialidad, integridad y disponibilidad de la información que se procesa, intercambia, reproduce o conserva por medio de las tecnologías de información. En la práctica la seguridad se representa por un conjunto de tareas, procesos y actividades, implementados conjuntamente con elementos de computación y telecomunicaciones para controlar y proteger contra amenazas que pongan en riesgo los recursos informáticos.

## **Estándar ISO/IEC 14598**

Este estándar es el responsable de indicar los requisitos a tener en cuenta para la aplicación de los métodos de medición y para el proceso de evaluación, logrando un entorno de trabajo que evalúa la calidad de diferentes tipos de productos de software. La ISO/IEC 14598 (ISO/IEC, 1999) está conformada por seis partes especificando el proceso a seguir para la evaluación:

- ISO/IEC 14598-1 Visión general.
- ISO/IEC 14598-2 Planificación y Gestión.

- ISO/IEC 14598-3 Procedimiento para desarrolladores.
- ISO/IEC 14598-4 Procedimiento para compradores.
- ISO/IEC 14598-5 Procedimiento para evaluadores.
- ISO/IEC 14598-6 Documentación de los módulos de evaluación.

La parte que especifica y explica el procedimiento por el cual los evaluadores del software se pueden guiar para desarrollar este proceso de evaluación está dentro de la quinta parte del estándar (ISO/IEC 14598-5), proporcionando requisitos y recomendaciones para la implantación práctica de la evaluación del producto de software. Puede ser usada para aplicar los conceptos descritos en el estándar ISO/IEC 9126. Las actividades de evaluación son una característica fundamental para el desarrollo efectivo del software y se necesita tener cronogramas de evaluación que brinden información óptima durante el ciclo de vida del producto.

## **Estándar ISO/IEC 25000**

Surge en el año 2005 y es actualizado en mayo de 2010. Creado con el fin de integrar en una sola familia el estándar ISO/IEC 9126 y el ISO/IEC 14598, es decir, la idea fue solapar el modelo de calidad y el proceso de evaluación.

Esta nueva familia, la ISO/IEC 25000 (Ramírez , y otros, 2011) tiene como principales objetivos:

- Guiar el desarrollo de los productos de software además de establecer criterios para la especificación de requisitos de calidad y métricas.
- Establecer un modelo de calidad para el producto de software, definiendo las características y sub-características que se deben tener en cuenta.
- Definir el proceso de evaluación de la calidad del producto de software, con el fin de poder establecer la calidad en función del modelo.

Este estándar proporciona una guía para el uso de las nuevas series de estándares internacionales llamados SQuaRE con el fin de lograr organizar, enriquecer y unificar las series que cubren dos procesos principales: especificación de requerimientos de calidad del software y evaluación de la calidad del mismo, soportada por el proceso de medición de calidad del software.

Después de haber analizado los diferentes estándares de calidad y seguridad, resulta factible utilizarlos para el desarrollo del actual trabajo, haciendo énfasis en las métricas de seguridad de la ISO/IEC 9126 y en el proceso de evaluación de la ISO/IEC 14598.

## 1.3 Métricas de seguridad

Según la IEEE “*Standard Glossary of Software Engineering Terms*” métrica se define como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” (IEEE, 1993).

Otros conceptos de métrica son:

“La continua aplicación de técnicas basadas en la medición al proceso de desarrollo de software y a sus productos para proveer información administrativa, significativa y oportuna, junto con el uso de esas técnicas para mejorar el proceso y sus productos” (Westfall, 1995).

“Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento” (Briand, y otros, 1996).

La clasificación de las métricas (Corletti, y otros, 2008) está en dependencia del tipo de prueba para las que estén creadas. Las métricas analizadas y aplicadas en el presente trabajo son las métricas de seguridad, las cuales se definen como el conjunto de preceptos y reglas, necesarias para poder medir de forma real el nivel de seguridad de un sistema. Con el fin de lograr una correcta selección de las métricas a utilizar no se debe dejar pasar por alto alguno de los siguientes aspectos:

**¿Para qué?** (se quiere medir):

- Medir la evolución de la seguridad de un sistema o producto.
- Tener datos elocuentes sobre el estado de seguridad de un producto.
- Saber dónde hay que hacer hincapié en la mejora de la seguridad.

**¿Qué?** (se quiere medir):

- Cantidad y tipo de amenazas.
- Vulnerabilidades y puntos débiles.

**¿Cómo?** (se puede medir):

- ¿Cómo convertir la información en datos válidos?
- ¿Qué fuentes y herramientas se necesitan?

A continuación se muestra la Tabla 1, donde se relacionan un conjunto de métricas de seguridad tomadas del estándar ISO/IEC 9126, las que permitirán que el sistema cuente con métricas ya definidas para la evaluación de la seguridad de las aplicaciones.

**Tabla 1: Descripción de las métricas aplicables al sistema**

Métrica	Propósito	Método de aplicación	Fórmula	Interpretación del valor obtenido
Detección de vulnerabilidades	Determinar cantidad de vulnerabilidades de seguridad del sistema	Contar el número de vulnerabilidades del sistema y comparar con el valor mínimo	$X = B/A$ A: Número de vulnerabilidades detectadas en el	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

		requerido para tener una adecuada seguridad en el sistema	sistema. <b>B:</b> Valor mínimo requerido de vulnerabilidades	seguro
Identificación de riesgos	Identificar posibles riesgos de seguridad que pudieran afectar el funcionamiento del sistema en determinado momento	Contar la cantidad de riesgos detectados	<b>X = 1 - A/B</b> <b>A:</b> Número de riesgos que pudieran convertirse en vulnerabilidades. <b>B:</b> Número de riesgos detectados en el sistema	<b>0 &lt;= X &lt;= 1</b> A mayor cercanía al 1 menor probabilidad de que los riesgos se conviertan en vulnerabilidad
Controlabilidad de acceso	¿Cuán controlable es el acceso al sistema?	Determinar posibles puntos que permitan realizar inyecciones SQL con el objetivo de violar el acceso al sistema y obtener información valiosa	<b>X =</b> Cantidad de puntos que permiten realizar inyecciones SQL	Mientras menor más seguro
Adecuación de pruebas	¿Qué parte de los casos de prueba necesarios están cubiertos por el plan de pruebas?	Contar el número de casos de prueba previstos y compararlo con el número de casos de prueba necesarios para obtener la cobertura de pruebas adecuadas	<b>X = A/B</b> <b>A:</b> Número de casos de prueba diseñados en el plan de pruebas y confirmados en la revisión <b>B:</b> Número de casos de prueba requeridos	<b>0 &lt;= X</b> Donde <b>X</b> sea más grande será más adecuado
Evitación de fallos	¿Cuántos patrones de falla fueron puestos bajo control para evitar los fallos?	Contar el número de patrones de fallas evitadas y compararlo con el número de patrones de falla a considerar	<b>X = A/B</b> <b>A:</b> Número de fallos. <b>B:</b> Número de fallos totales	<b>0 &lt;= X</b> Donde <b>X</b> sea más grande es mayor la evitación de fallos

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Prevencción de corrupción de datos	Se encarga de cuidar los datos del sistema ante la corrupción	Se basa en la operación de división entre las posibles instancias de prevención de corrupción de los datos y las instancias que se detecten capaces de corromper los datos	$X = A/B$ <b>A:</b> Número de instancias de prevención de corrupción de datos implementadas según especificaciones. <b>B:</b> Número de instancias de operaciones o accesos identificadas en los requerimientos como capaces de destruir o corromper datos	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más completo
Encriptamiento de datos	Se encarga de reflejar los datos que se deben encriptar	Su resultado se basará en el resultado de la operación matemática que se realice entre el número de instancias que se declaren como encriptables y las que requieren ser encriptadas	$X = A/B$ <b>A:</b> Número de instancias implementadas de datos encriptables según las especificaciones. <b>B:</b> Número de datos que requieren encriptamiento según las especificaciones	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más completo

Luego de realizar los cálculos correspondientes a cada métrica se obtienen los valores que definirán un nivel bajo, medio, alto y muy alto para el software. En la Tabla 2 (Rubalcaba Betancourt, y otros, 2008) se muestra la escala por la cual se evaluarán las métricas propuestas en la tabla anterior, en el caso de las métricas que el evaluador inserte, este será el encargado de especificar la escala de evaluación.

Tabla 2: Escala para la evaluación

Intervalo	Nivel
0-0.4	Bajo
0.4-0.7	Medio
0.7-0.9	Alto
0.9-1	Muy alto

## 1.4 Proceso de evaluación

En el estándar ISO/IEC 14598-1 se plantea como propósito principal del proceso de evaluación del software el apoyo directo tanto para el desarrollo como para la adquisición de este, que cumpla las necesidades del usuario y del cliente, teniendo como objetivo final asegurar que el producto aporte la calidad requerida y satisfaga las necesidades declaradas e implícitas de los usuarios. (ISO/IEC 14598-1:1999, 1999)

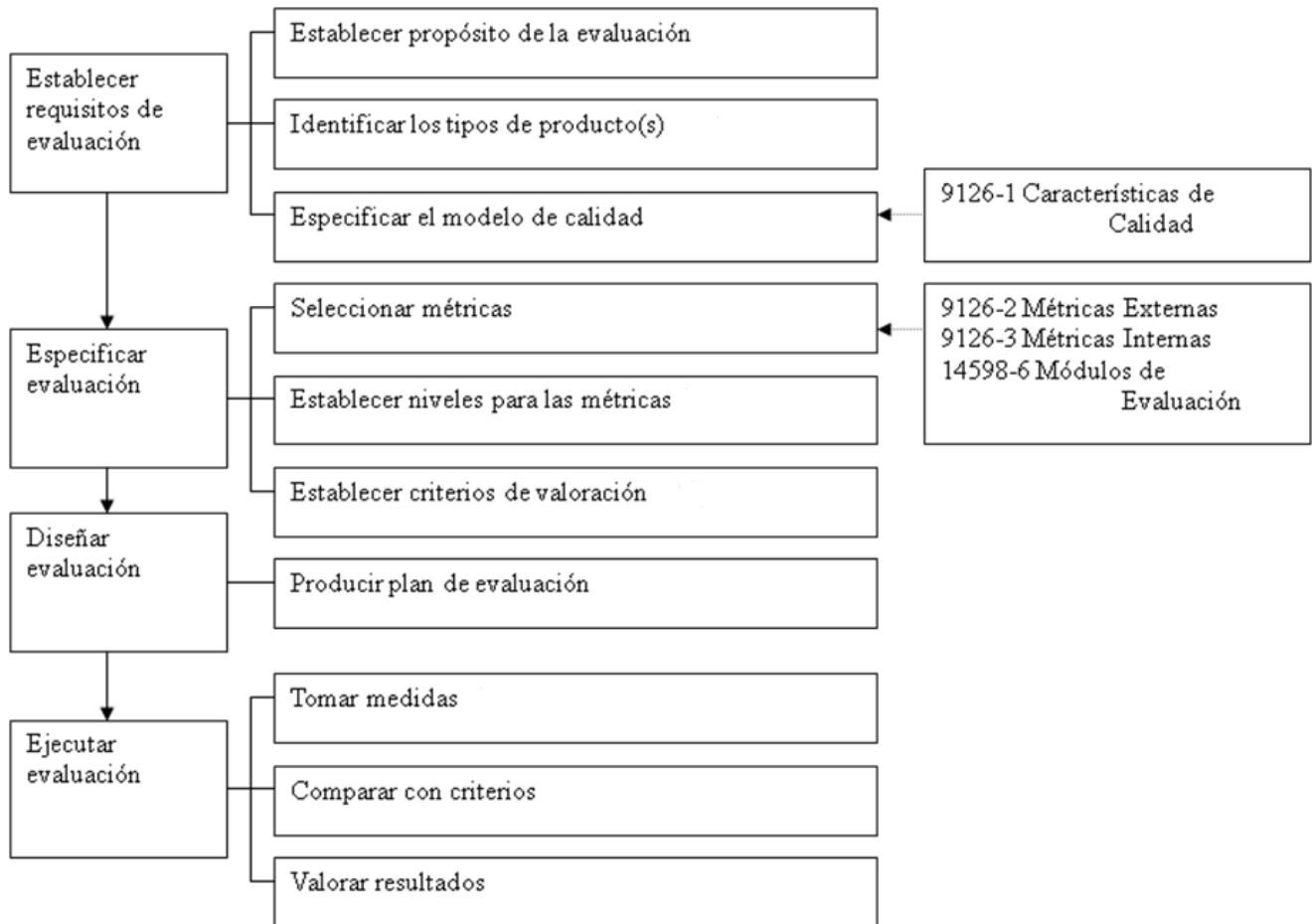
Para definir el proceso de evaluación del presente trabajo se realizó un estudio previo, identificando la guía de evaluación del proyecto MEDUSAS (Mejora y Evaluación del Diseño, Usabilidad, Seguridad y Mantenibilidad del Software). Este proyecto define sus componentes de trabajo en un entorno metodológico, tecnológico y un tercero de gestión y divulgación. Asignándole la responsabilidad al entorno metodológico de guiar el trabajo a través de la metodología MEDUSAS y esta a su vez define las actividades, roles, entradas y salidas necesarias para llevar a cabo este proceso, teniendo en cuenta los principales estándares de evaluación de software (ISO/IEC 25000, ISO/IEC 9126, ISO/IEC 14598, ISO/IEC 27000). Este proceso es variable, pues se ajusta a la necesidad de cada proyecto o empresa, pasando así por un ciclo de planificación, especificación, ejecución y conclusión en dependencia del cliente.

Por otra parte, el estándar ISO/IEC 14598-5 persigue también como uno de sus principales objetivos guiar el proceso de evaluación promoviendo las siguientes características del mismo (ISO/IEC 14598-1:1999, 1999):

- **Repetibilidad:** la evaluación repetida del mismo producto con respecto a la misma especificación de evaluación por el mismo evaluador debería producir resultados que pueden ser aceptados por ser idénticos.
- **Reproducibilidad:** la evaluación repetida del mismo producto con respecto a la misma especificación de evaluación por un evaluador diferente debería producir resultados que pueden ser aceptados por ser idénticos.
- **Imparcialidad:** la evaluación no debería estar orientada hacia un resultado particular.

- **Objetividad:** los resultados de la evaluación deberían ser ciertos, por ejemplo, no influidos por los sentimientos u opiniones del evaluador.

En la Figura 1 se muestran las etapas que componen el proceso de evaluación antes mencionado, junto con sus actividades correspondientes. Para evaluar el software se sigue un orden lógico, primero se establecen los requisitos de evaluación, después se especifica, se diseña y se ejecuta la evaluación. Cada una de estas etapas está compuesta por actividades que completan su realización.



**Figura 1: Pasos para realizar el proceso de evaluación según ISO/IEC 14598-5**

Luego del estudio realizado se decide hacer uso del proceso de evaluación definido por el estándar ISO/IEC 14598-5 que satisface las necesidades actuales del equipo de desarrollo, pues especifica y describe los pasos que los evaluadores del software deben seguir para desarrollar este proceso. Debe aclararse que se ha decidido omitir el paso relacionado con la toma de medidas establecida en el estándar, por lo que solo se realizará la comparación de los criterios y la valoración de los resultados. Además, este estándar permite no solo evaluar el proyecto durante las fases de desarrollo, sino también medir el producto software final, lo cual resulta imposible con el proyecto MEDUSAS.

## 1.5 Análisis de las herramientas de pruebas de seguridad

El uso de herramientas para realizar pruebas de seguridad a las aplicaciones web, trae consigo resultados que brindan información referente a las vulnerabilidades que presentan estas aplicaciones. Hoy en día existen varias herramientas con este propósito, por lo que resulta necesario realizar una comparación entre estas teniendo en cuenta diferentes criterios de selección.

Partiendo de que estas herramientas tienen en común que son libres, se debe observar en la Tabla 3 los otros criterios comparativos que se establecen, como son la plataforma que soporta, los tipos de pruebas de seguridad que realizan (basándose en las tres pruebas de mayor prioridad que presenta OWASP<sup>1</sup>), y las limitantes que pueden presentar.

**Tabla 3: Herramientas de pruebas de seguridad**

Nombre	Descripción	Plataforma	Vulnerabilidades más comunes en aplicaciones web			Limitantes
			Inyección SQL	Fallos de <i>Cross-Site Scripting</i> (XSS)	Fallos de gestión de la sesión y la autenticación rota	
Acunetix (versión libre)	Permite explorar un sitio web completo para luego ejecutar pruebas de seguridad	Windows	X	X		No permite guardar y generar informes de escaneo
Wapiti	Escáner de vulnerabilidades escrito en Python	Windows, Linux	X	X	X	No posee interfaz gráfica. Necesita tener instalado python para ejecutarla
Scrawlr	Revisa los sitios web en busca de posibles debilidades que puedan facilitar la	Windows	X	X		Explora un máximo de 1500 páginas. No comprueba

<sup>1</sup> Open Web Application Security Project, en español Proyecto Abierto de Seguridad de Aplicaciones Web.



# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

	inyección de SQL					formularios y no vale para sitios que requieran autenticación
W3AF	Permite realizar diferentes tipos de pruebas de seguridad a aplicaciones web para determinar las vulnerabilidades más frecuentes, entre ellas las identificadas por OWASP	Windows, FreeBSD, Linux	X	X	X	Depende de python para ser ejecutada

Finalmente se decide utilizar W3AF, valorando que la misma es un *framework* libre, es multiplataforma, de fácil uso e integra todas las pruebas que realizan las herramientas similares a esta. Además brinda la posibilidad de realizar otros tipos de pruebas de seguridad.

## 1.6 Descripción de las características del Sistema de evaluación de la seguridad v 1.0.

El “Sistema de evaluación de la seguridad” es un sistema, que tiene como objetivo facilitar el control y la evaluación de la seguridad en las aplicaciones que se desarrollan en el Departamento de Seguridad Digital. Dicho sistema se encarga de procesar los datos arrojados por las pruebas de seguridad realizadas a un software con el uso de la herramienta W3AF. Dentro de las características esenciales del sistema se tiene el almacenamiento de los datos del proyecto que se desea evaluar, luego estos datos son procesados a través del proceso ETL, lo que permite obtener de ellos solo la información necesaria para lograr la evaluación del software mediante el uso de métricas de seguridad.

En la Figura 2 (Peña Romero, y otros, 2012) se muestra el proceso de evaluación:



Figura 2: Proceso de evaluación del SES v1.0

Como se puede observar, el proceso es bastante sencillo y logra obtener un resultado de evaluación que permite conocer el nivel de seguridad del software.

## 1.7 Entorno de desarrollo

En este epígrafe se define el entorno de desarrollo para la propuesta de solución del presente trabajo.

### 1.7.1 Metodología de desarrollo *Microsoft Solution Framework for Agile Software Development*

Durante el desarrollo de un software la metodología a utilizar es un factor fundamental para lograr el éxito, y esto se debe a que en ocasiones el desarrollo de software puede ser riesgoso o difícil de controlar. En este sentido el uso de una correcta metodología facilita que al final se obtenga el producto esperado, evitando la insatisfacción de los clientes. Las metodologías se clasifican en dos tipos, las robustas y las ágiles, siendo estas últimas las que se establece que sean aplicadas en el Departamento de Seguridad Digital, con el objetivo de optimizar las prácticas de desarrollo de software.

*MSF Agile* (Mendoza Sánchez, 2004) es la nueva propuesta de *Microsoft* en el mundo de procesos y prácticas ágiles de desarrollo de software, es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. *MSF Agile* se centra en los modelos de proceso y de equipo, dejando en un segundo plano las elecciones tecnológicas.

Esta metodología cuenta con las siguientes características (Microsoft, 2005):

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

Cómo metodología cuenta con una serie de principios para su aplicación que a continuación se enuncian:

- Formar equipo con el cliente.
- Promocionar las comunicaciones abiertas.
- Trabajar con una visión común.
- La calidad es el negocio de todos, todos los días.
- Mantenerse ágil, adaptarse a los cambios.
- Hacer del despliegue un hábito.
- Crear un flujo de valor.

Esta metodología incluye cinco fases para el desarrollo y seguimiento del producto, estas son: Visión y Alcance, Planificación, Desarrollo, Estabilización y Despliegue.

*MSF Agile* dispone de los elementos de trabajo siguientes:

- **Escenario:** descripción de la necesidad o solicitud del usuario.
- **Error:** defecto o desviación entre el comportamiento esperado y el comportamiento observado en el producto.
- **Requisitos de calidad del servicio:** material resultante esperado del producto final. El mismo puede ser un resultado, un problema resuelto o una característica, entre otros.
- **Tarea:** acción independiente que debe realizar una persona o un grupo de personas.
- **Riesgo:** evento o condición probable que puede dar resultados potencialmente negativos para el proyecto en el futuro.

Esta metodología tiene un diseño óptimo para proyectos pequeños con un calendario de entrega rápido, por lo que puede ser conveniente elegir cuando:

- No se necesita muchos procesos documentados.
- Los equipos de desarrollo de software son reducidos.
- Existen ciclos de desarrollo de software cortos (medidos en semanas o meses).

Finalmente se define *MSF Agile* como metodología de desarrollo, ya que se tiene como política del proyecto la aplicación de la misma. Esta permite guiar el proceso de desarrollo del software a construir,

definir los roles y responsabilidades que se desempeñan en el proyecto, y generar los elementos de trabajo necesarios en cada una de las fases.

## 1.7.2 Lenguaje de modelado *Unified Modeling Language (UML)*

Para guiar el diseño de la aplicación fue seleccionado como lenguaje de modelado el Lenguaje Unificado de Modelado (por sus siglas en inglés UML, *Unified Modeling Language*). UML (Hernández Orallo, 2011), en cuanto al diseño de software es un lenguaje de modelado conocido y utilizado en la actualidad. Se caracteriza por su importancia a la hora de especificar o describir métodos o procesos. Se utiliza además en la definición de un sistema, para detallar los artefactos, para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se compone por distintos diagramas reflejando las diferentes etapas del desarrollo de un proyecto de software. Incluye aspectos conceptuales como lo son los procesos de negocio, esquemas de base de datos (BD), funciones del sistema y componentes reutilizables. Entre sus funciones se encuentran:

- **Visualizar:** permite expresar gráficamente un sistema de forma que otra persona lo pueda entender.
- **Especificar:** permite especificar las características de un sistema antes de su construcción.
- **Construir:** a partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Abogando para que se contribuya a comprender como va a funcionar el sistema y de qué manera debe ser implementado se utilizará como guía en el diseño de los diagramas el lenguaje UML en su versión 2.0.

## 1.7.3 Herramientas CASE *Visual Paradigm*

Las herramientas CASE (Ingeniería de Software Asistida por Ordenador, por sus siglas en inglés *Computer Aided Software Engineering*) son aplicaciones informáticas que tienen como objetivo el aumento de la productividad en el desarrollo de software, lo que disminuye el coste en términos de dinero y tiempo. Una de las herramientas CASE que cumple con estos objetivos es el *Visual Paradigm*, precisamente esta es la seleccionada por el equipo de desarrollo. *Visual Paradigm for UML* (Cabrera, y otros, 2012) soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML.

Algunas de las características de la herramienta son:

- **Integración con entornos de desarrollo:** apoyo al ciclo de vida completo de desarrollo de software en IDE como: Eclipse, NetBeans, Oracle, JDeveloper, JBuilder y otros.
- **Detalles de casos de uso:** entorno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- **Multiplataforma:** soportado en la plataforma Java para varios sistemas operativos (*Windows / Linux / Mac OS X*).

Su versión 8.0 incluye la funcionalidad de crear y especificar perfiles UML, la cual resulta de vital importancia para la implementación y ejecución de extensiones para la herramienta. Debido a todas las características mencionadas, al dominio de los autores del trabajo con esta herramienta y los beneficios que brinda para el desarrollo de software, especialmente referentes al modelado, se decidió utilizar *Visual Paradigm for UML* para modelar los diagramas que apoyarán el diseño del presente trabajo.

#### 1.7.4 Servidor web Apache

Apache (Naramone, y otros, 2005) actúa como servidor web. Su función principal es analizar cualquier archivo solicitado por el navegador y mostrar los resultados correctos, de acuerdo con el código dentro de ese archivo. Este servidor es muy potente y puede lograr prácticamente cualquier tarea que se le asigne. Lo que hace a este servidor web universal, es que se puede ejecutar en varios sistemas operativos. Es una tecnología 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. Brinda soporte para varios lenguajes: PHP, JAVA, Perl y librerías ASP.

Algunas de las características y funciones de este servidor en su versión 2.2.21 son:

- Páginas protegidas por contraseña para una multitud de usuarios.
- Páginas de error personalizadas.
- Pantalla de código en numerosos niveles de HTML, y la capacidad para determinar en qué nivel el navegador puede aceptar el contenido.
- Uso de los registros de errores en múltiples formatos.
- Hospedaje virtual para diferentes direcciones IP asignada al mismo servidor.
- Es un servidor web conforme al protocolo HTTP.
- El diseño modular de Apache permite a los administradores de sitios web elegir qué características se van a incluir en el servidor al seleccionar los módulos que se van a cargar, ya sea al compilar o al ejecutar el servidor.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.

Apache puede ser utilizado para alojar un sitio web, o simplemente para probar sus páginas antes de que se carguen. El servidor a utilizar será precisamente Apache en su versión 2.2.21, este se encargará de albergar el sistema y tendrá la función de estar a la espera de que un usuario realice alguna petición, el protocolo por el que se publicará la solicitud será HTTPS (*Hypertext Transfer Protocol Secure*).

## 1.7.5 Sistema de Gestión de Contenidos Drupal

Un Sistema de Gestión de Contenidos (por sus siglas en inglés *Content Management System* o CMS) es aquel sistema que contribuye a formar una estructura de soporte (*framework*) para la creación y administración de contenidos.

Algunas de sus principales características son:

- Se adapta a las necesidades o preferencias de los usuarios.
- Es de fácil uso, pues cualquier usuario puede añadir contenido en el sitio web sin tener conocimientos de programación.
- Es compatible con disímiles navegadores disponibles en todas las plataformas, además de adaptarse al idioma del usuario.

Drupal (Tesis-OyS, 2010) es un sistema de gestión de contenido modular y configurable, escrito en PHP, desarrollado y mantenido por una comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema. 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 es un CMS que tiene varias características que lo hacen sencillo y fácil de entender, dichas características se exponen a continuación:

- **Código abierto:** al contrario de otros sistemas de blogs o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades reales de cada usuario.
- **Permisos basados en roles:** los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un rol y agrupar los usuarios por roles.
- **Ayuda on-line:** posee un sistema de ayuda *on-line* eficaz y rápido.
- **Plantillas:** el sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar el aspecto del sitio web. Se pueden crear plantillas con HTML y PHP.
- **Módulos:** la comunidad de Drupal ha contribuido con muchos módulos que proporcionan funcionalidades como “página de categorías”, autenticación mediante LDAP (*Lightweight Directory Access Protocol*, en español Protocolo Ligero de Acceso a Directorios), mensajes privados, foros, entre otros.

Para dar solución al problema planteado se ha seleccionado el CMS Drupal en su versión 7.14, por las características antes mencionadas, además de ser de código abierto y estar en constante actualización. Drupal se presta para el desarrollo de la presente propuesta de solución, ya que el equipo de desarrollo se siente más cómodo con el uso de esta herramienta, pues ya existía un conocimiento previo de este CMS, que destaca por su eficiencia y flexibilidad.

## 1.7.6 Lenguajes para el desarrollo web

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, órdenes, comandos, operadores y reglas de sintaxis; que se ponen a disposición del programador para que este pueda darle solución al proceso a través del ordenador, comunicándose además con los dispositivos de hardware y software. En el desarrollo de aplicaciones web existe una gran diversidad, diferenciándose entre sí, teniendo en cuenta las características específicas del producto final que se quiera alcanzar.

### Lenguajes del lado del servidor

PHP (Santana Mancilla, 2001) es un lenguaje de programación de estilo clásico, o sea, utiliza variables, sentencias condicionales, ciclos o bucles, funciones, entre otros. No es un lenguaje de marcado como podría ser HTML, XML o WML sino que está más cercano a JavaScript o a C. Pero a diferencia de JavaScript que se ejecuta en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador, este es normalmente una página HTML pero igualmente podría ser una página WML. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente de este, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportarlo.

Algunas de las ventajas de PHP son:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a BD.
- Gran capacidad de conectividad con los SGBD, tales como MySQL y PostgreSQL.
- Es un lenguaje multiplataforma.
- Su programación en PHP es segura y confiable.
- Cuenta con una amplia documentación en su página oficial.

La selección de PHP en su versión 5.3 como lenguaje del lado del servidor se respalda por sus características, entre las que destaca que es un lenguaje implícito en el funcionamiento del CMS Drupal. Constituye un lenguaje *script* de alto nivel interpretado del lado del servidor. Funciona principalmente para proporcionar características dinámicas a una página web y al ser ejecutado del lado del servidor, permite acceder tanto a los recursos internos del mismo como a otros externos, por ejemplo a una BD.

## **Lenguajes del lado del cliente**

Los lenguajes del lado del cliente son reconocidos por no necesitar un tratamiento previo, ya que el navegador los asimila directamente.

### **HTML (*Hypertext Markup Language*)**

HTML (Vaquero, 2010) es un lenguaje de marcas orientado a la estructuración y publicación de documentos en forma de hipertexto. La mayoría de las marcas son semánticas. HTML es un lenguaje extensible, al que se le pueden añadir nuevas características, marcas y funciones. Los documentos HTML están formados por una serie de bloques de texto con una entidad lógica (titulares, párrafos, listas, entre otros). La interpretación de estas entidades se deja al navegador, lo cual da una gran flexibilidad a la presentación del documento, que puede ser mostrado, por ejemplo, en terminales gráficos o de texto. Este lenguaje permite diseñar hipertextos y hoy en día, la mayoría de los procesadores de textos disponen de opciones para guardar los documentos en este formato, por lo que no presenta dificultad.

HTML permite:

- Publicar documentos en línea con títulos, textos, tablas, fotos, y otros.
- Recuperar información en línea mediante enlaces de hipertexto.
- Insertar videoclips, audio, hojas de cálculo, entre otras aplicaciones.

Este lenguaje organiza el documento en elementos lógicos, además de definir las funciones a ejecutar por el programa visualizador y las operaciones tipográficas.

### **CSS (*Cascading Style Sheets*)**

CSS (TIC, 2008) es una hoja de estilo en cascada que permite cambiar el formato de presentación de cualquier etiqueta de HTML. CSS también es un lenguaje que permite a un usuario asociar un estilo (por ejemplo, tipo de letra del texto, color del texto, fondo de la hoja, espacio entre líneas, tamaño y color de los títulos, entre otros) a los documentos estructurados, ya sean documentos HTML o aplicaciones XML. La gran importancia de las hojas CSS reside en que:

- Permiten independizar el estilo del contenido del documento.
- Si se aplica una misma hoja de estilo a varias páginas HTML, al modificarla, automáticamente se actualiza en todas las páginas.
- Separando el estilo de presentación del contenido de los documentos, CSS simplifica la creación y mantenimiento de los sitios web.
- Al extraer de la página HTML todo el estilo, queda solamente la estructura del texto, con lo cual se devuelve a la hoja HTML su función original.
- Al independizar el estilo de la estructura, el acceso a ambas es mucho más rápido y flexible, se pueden definir varios estilos para un mismo documento, de forma que el usuario elija el que más le conviene.



- Mejora el rendimiento del ordenador, ya que la hoja de estilo se carga una única vez para todas las páginas HTML a las que se aplica.

## 1.7.7 Entorno Integrado de Desarrollo NetBeans

Un Entorno Integrado de Desarrollo (IDE por sus siglas en inglés *Integrated Development Environment*), es un programa compuesto por un conjunto de herramientas utilizadas para desarrollar código. Las herramientas que comúnmente lo componen son: un editor de texto, un compilador, un intérprete, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones.

Un IDE puede pensarse para un único lenguaje de programación o para varios de estos, además provee un marco de trabajo poco complejo para la mayoría de los lenguajes de programación y en algunos casos puede funcionar como un sistema en tiempo de ejecución donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad del trabajo orientado a archivos de texto.

NetBeans (Funes, 2011) es un proyecto de código abierto para desarrolladores de software, que se ejecuta en varias plataformas incluyendo Windows, Linux, Mac OS X y Solaris. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software que se denominan módulos.

Un módulo es un archivo Java que contiene clases escritas para interactuar con las APIs de NetBeans y un archivo especial (*manifest file*) que lo identifica. Las aplicaciones construidas a partir de estos pueden ser extendidas agregándole nuevos módulos, que pueden ser desarrollados de manera independiente, lo que hace posible que las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas por otros desarrolladores de software. Precisamente NetBeans en su versión 6.9 es el IDE que se utilizará para el desarrollo del sistema.

## 1.7.8 Sistema Gestor de Base de Datos MySQL

Según (Sánchez Asenjo, 2009) un Sistema Gestor de Bases de Datos o SGBD es aquel software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos.

MySQL (Mateu, 2004) es un SGBD desarrollado por la empresa MySQL AB, una empresa de origen sueco que lo desarrolla bajo licencia de código libre (concretamente bajo GPL), aunque también, si se desea, puede ser adquirido con licencia comercial para ser incluido en proyectos propietarios. Aunque no ofrece las mismas capacidades y funcionalidades que otras BD, compensa esto con un rendimiento

adecuado que hace de ella la base de datos de elección en aquellas situaciones en las que se necesitan sólo capacidades básicas.

Las funcionalidades más destacadas de MySQL son:

- Soporte de transacciones.
- Soporte de replicación (con un master actualizando múltiples claves).
- Librerías.
- Búsqueda por texto.
- Cache de búsquedas (para aumentar el rendimiento).

Precisamente MySQL en su versión 5.5.16 es el SGBD más conveniente para el sistema que se desea desarrollar, ya que el equipo de trabajo lo reconoce por ser rápido, robusto y de fácil uso, además de tener un mejor dominio del conocimiento necesario para trabajar con él.

### 1.7.9 Herramienta de desarrollo para el proceso ETL. *Pentaho Data Integration*

En el mundo actual existen diversas herramientas que permiten desarrollar el proceso ETL. A continuación se muestran algunas características (Rodríguez Lorenzo, 2010) que se deben tener en cuenta para su selección:

- **Multiplataforma:** la herramienta debe funcionar en cualquier plataforma.
- **Independencia del tipo de fuente o destino:** la herramienta debe ser capaz de leer y escribir directamente desde y hacia las fuentes o destinos, independientemente de su tipo.
- **Soporte para metadatos:** la herramienta debe tener disponible la información sobre todos los datos durante el desarrollo y ejecución de los procesos.
- **Soporte funcional:** debe ser posible la realización eficiente de operaciones para la limpieza de los datos, transformaciones, agregaciones, reorganización y carga.
- **Facilidad de uso:** la herramienta debe ser de propósito general y amigable, de forma tal que el usuario pueda identificarse con la misma.

*Pentaho Data Integration* (Pentaho, 2009) es una potente herramienta de extracción, transformación y carga que permite la integración de ambientes y datos para soportar las áreas de negocio. Esta herramienta cuenta con una interfaz gráfica e intuitiva, probada y escalable, con muchas facilidades para los usuarios. Una de las principales características es que permite realizar transformaciones complejas sin tener que generar algún código personalizado.

Es una aplicación implementada en Java, se caracteriza por ser multiplataforma. Se considera un motor de transformación, y cuenta con una disponibilidad de componentes limitada, pero suficiente para la realización del proceso ETL. Además tiene un repositorio de BD que brinda muchas posibilidades para el trabajo en equipo y soporta SGBD como Oracle, MySQL y Postgres. A continuación se muestran las ventajas de esta herramienta:

- Funciona en Windows, Unix y Linux.
- Tiene una interfaz gráfica con indicadores de las transformaciones.
- Es una aplicación implementada en Java con algunas características avanzadas en JavaScript.
- Ofrece una licencia pública GPL.
- Basada en metadatos.
- Como soporte se encuentran los foros y la comunidad *Pentaho*.
- Soporta Oracle, MySQL y Postgres.
- Con respecto a la escalabilidad, soporta la arquitectura de procesamiento en paralelo para distribuir las tareas de ETL a través de múltiples servidores.

Luego del estudio de esta herramienta, el equipo de trabajo considera que *Pentaho Data Integration 4.2.0* es la que se ajusta para realizar el proceso ETL, ya que esta es estable, permite leer y escribir de cualquier BD y realiza el proceso de manera eficiente.

## 1.7.10 Proceso de integración de datos ETL

ETL (*Extract, Transform and Load*) o lo que es lo mismo: Extraer, Transformar y Cargar; es un proceso que permite mover los datos desde disímiles fuentes, reformatearlos, limpiarlos y finalmente cargarlos en otro sistema, que pudiera ser una BD operacional donde se analizarían los datos con el fin de obtener solo los necesarios para trabajar. Su función permite ahorrar tiempo y es óptimo a la hora de agilizar el trabajo. La ejecución de las acciones que responden al proceso ETL se realiza de forma continua, garantizando el éxito del proceso. A continuación se explica brevemente en que consiste cada una de las acciones (Peña Romero, y otros, 2012) que conforman este proceso:

- **Extracción:** es la primera parte del proceso ETL, esta se encarga de extraer los datos desde los sistemas de origen, por lo general las fuentes de almacenamiento de datos fusionan datos que provienen de disímiles sistemas de origen. Los formatos de las fuentes por lo común se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes. Con la extracción se convierten los datos a un formato que ya está listo para darle paso al proceso de transformación.
- **Transformación:** este es el elemento clave del diseño del proceso ETL. Una transformación está compuesta por pasos, que están relacionados entre sí a través de los saltos. Los pasos son el elemento más pequeño dentro de las transformaciones y los saltos constituyen el elemento a través del cual fluye la información entre los diferentes pasos (siempre es la salida de un paso y la entrada de otro). En esta fase se aplica una serie de funciones o reglas de negocio sobre los datos extraídos para que de este modo estén listos para ser cargados.
- **Carga:** en este momento los datos de la fase anterior son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una extensa

variedad de acciones distintas. En esta fase se interactúa directamente con la BD de destino, en algunas se sobrescribe la información antigua con nuevos datos. Al realizar esta operación se aplicarán todas las restricciones y disparadores (*triggers*) que se hayan definido (por ejemplo, valores únicos, campos obligatorios, rangos de valores). Estas restricciones y disparadores (si están bien definidos) contribuyen a que se garantice la calidad de los datos en el proceso ETL, los cuales deben ser tenidos en cuenta.

Con el objetivo de minimizar la complejidad de la interpretación del fichero que devuelve la herramienta de pruebas de seguridad se decide hacer uso de este proceso de integración de datos mediante la herramienta *Pentaho Data Integration 4.2.0*.

## **Conclusiones del capítulo**

El estudio de los referentes teóricos permitió el entendimiento de los conceptos esenciales que se relacionan con la problemática existente dentro del objeto de estudio definido. Con el análisis de los estándares definidos se logró establecer y abarcar todas las etapas del proceso de evaluación para guiar el flujo de actividades del sistema a desarrollar; también permitió realizar la selección de métricas de seguridad a ser aplicadas. La metodología seleccionada permitirá guiar el proceso de desarrollo de la aplicación, y la definición del entorno de desarrollo posibilitó establecer las herramientas y tecnologías para desarrollar la solución que se propone.

## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

En este capítulo se describe y caracteriza la solución propuesta teniendo como punto de partida el análisis de la problemática planteada. Se especifica la arquitectura y los patrones de diseño a utilizar. Se realiza además una descripción de las funcionalidades que el sistema debe cumplir a través del levantamiento de requisitos.

### 2.1 Fase visión y alcance

La visión y el alcance son dos aspectos que no se deben perder de vista, con el fin de mantener el equipo de trabajo con una visión de cuál sería solución a la situación problemática. En este sentido, para garantizar el cumplimiento de los aspectos antes mencionados se parte de que en el Departamento de Seguridad Digital se trabaja actualmente en el proyecto “Sistema para la evaluación de la seguridad de un software”. Dicho proyecto tiene como propósito determinar el nivel de seguridad de una aplicación, mediante el uso de una herramienta automatizada de pruebas de seguridad, de las cuales se obtienen una serie de indicadores con los que se pueden aplicar métricas de seguridad y obtener una evaluación cuantitativa de la seguridad de un software.

#### 2.1.1 Propuesta de solución

Partiendo del problema a resolver la propuesta de solución queda conformada por el desarrollo de un nuevo sistema para la evaluación de la seguridad, guiada por el proceso definido en el estándar ISO/IEC 14598-5 y haciendo uso de la herramienta de prueba W3AF, con el fin de procesar los resultados que arroja mediante la utilización de métricas definidas por el estándar ISO/IEC 9126. Esta versión del sistema incluye nuevas métricas que se encuentran predefinidas, no obstante la solución brinda la posibilidad al evaluador de introducir métricas que conozca, o que desee aplicar en el momento de la evaluación. Además cuenta con un historial de proyectos, que permite buscar evaluaciones realizadas en un período de tiempo dado. También brinda la posibilidad de generar un reporte con el objetivo de mostrar los resultados de la evaluación, los cuáles se podrán graficar y notificar vía correo electrónico a las personas interesadas.

#### 2.1.2 Vista conceptual del sistema

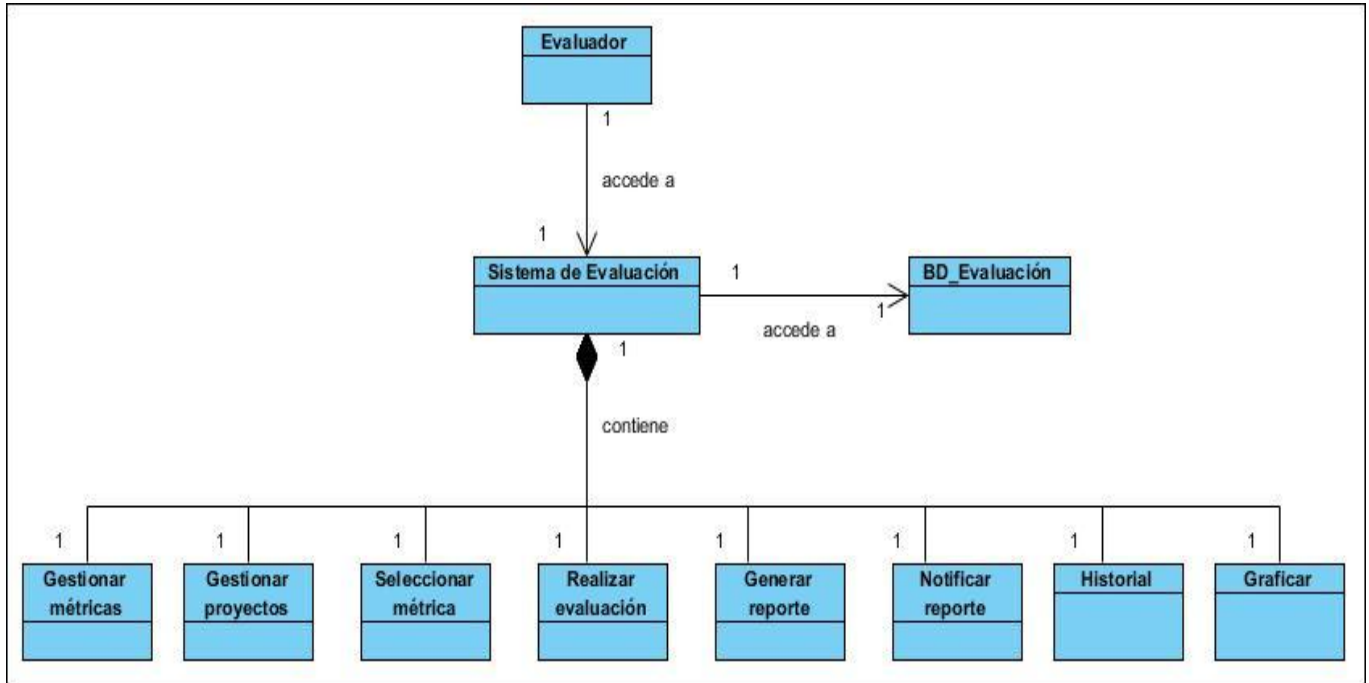
En la Figura 3 se refleja la vista conceptual del sistema, que responde a una representación de toda la información del mismo, funcionando como guía para el desarrollo de los elementos que conforman la propuesta de solución planteada anteriormente, en este caso particular su fundamento está basado en la investigación realizada por los autores, ya que el sistema surge por la necesidad del Departamento de Seguridad Digital. De este modo, la estructura de los procesos de negocio no es la mejor, y en busca del

## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

balance del mismo se presenta una vista conceptual general del sistema en función de la captura de los eventos que ocurren en su entorno, los objetos más significativos y la relación que pueda existir entre ellos. A continuación se ofrece una descripción de los elementos que componen la vista conceptual del sistema.

### **Descripción:**

- **Evaluador:** es la representación de la persona que interactúa con el sistema.
- **Sistema de Evaluación:** es la representación de la aplicación web con la que interactúa el evaluador con el fin de realizar el proceso de evaluación.
- **BD\_Evaluación:** es la representación de la BD, en la cual se registra el proceso ETL realizado.
- **Gestionar métricas:** es la funcionalidad que gestiona las métricas con las que se puede evaluar el proyecto.
- **Gestionar proyectos:** es la funcionalidad del sistema encargada de gestionar los datos de los proyectos a evaluar.
- **Seleccionar métricas:** es la funcionalidad que se encarga de mostrar las métricas que el evaluador puede seleccionar, además de permitir añadir nuevas métricas.
- **Realizar evaluación:** es la funcionalidad responsable de permitir al evaluador la realización de la evaluación al proyecto.
- **Generar reporte:** es la funcionalidad encargada de mostrar al evaluador un reporte con el nivel de seguridad de su aplicación una vez finalizado el proceso de evaluación.
- **Notificar reporte:** es la funcionalidad encargada de notificar el resultado de la evaluación del proyecto a los interesados del mismo.
- **Historial:** es la funcionalidad responsable de mostrar un historial con todas las evaluaciones realizadas.
- **Graficar:** es la funcionalidad encargada de graficar el resultado de la evaluación, con el fin de brindar una fácil y rápida interpretación.



**Figura 3: Vista conceptual del sistema**

### 2.1.3 Definición de personas

*MSF Agile* define una persona como aquel individuo que representa las necesidades específicas de sus grupos. En conjunto, estas vistas proporcionan los controles y equilibrios necesarios para asegurar que el equipo produce la solución correcta.

Para el sistema se define una persona que es el evaluador, encargado de realizar todo el proceso de evaluación de la seguridad. En la Tabla 4 se muestran sus responsabilidades.

**Tabla 4: Definición de personas**

Personas	Descripción
Evaluador	<p><u>Es responsable de:</u></p> <ol style="list-style-type: none"> <li>1. Gestionar los datos del proyecto a evaluar.</li> <li>2. Subir al sistema los datos necesarios para realizar el proceso de evaluación.</li> <li>3. Seleccionar y gestionar las métricas.</li> <li>4. Gestionar el reporte de la evaluación.</li> <li>5. Notificar los resultados de la evaluación.</li> <li>6. Graficar los resultados de las evaluaciones.</li> </ol>

### 2.2 Fase de planificación

Durante esta fase se define la planificación del proyecto y el equipo prepara las especificaciones funcionales. Además se confeccionan los planes de trabajo y se realiza el proceso de diseño para la

solución. Se describen los elementos necesarios para el diseño del sistema y se especifican los escenarios y los requisitos de calidad del servicio, siendo estos la guía para todo el proceso de desarrollo.

### 2.2.1 Escenarios del sistema

Un escenario (EC) es un tipo de elemento de trabajo, la grabación de un solo camino de interacción con el usuario a través del sistema. A medida que la persona trata de llegar a una meta, el escenario registra las medidas concretas que tomarán en el intento de alcanzar ese objetivo. Para el desarrollo del sistema propuesto, se identificaron los siguientes escenarios, así como las tareas que lo componen:

- **EC 1: Autenticar usuario**
- **EC 2: Gestionar métrica**
  - T 2.1: Crear métrica
  - T 2.2: Modificar métrica
  - T 2.3: Eliminar métrica
- **EC 3: Gestionar datos del proyecto**
  - T 3.1: Crear datos del proyecto
  - T 3.2: Modificar datos del proyecto
  - T 3.3: Eliminar datos del proyecto
- **EC 4: Seleccionar métrica**
- **EC 5: Cargar fichero**
  - T 5.1: Seleccionar fichero
  - T 5.2: Subir fichero
- **EC 6: Procesar datos del fichero**
  - T 6.1: Extraer datos del fichero
  - T 6.2: Transformar datos del fichero
  - T 6.3: Cargar datos del fichero
- **EC 7: Calcular métrica**
- **EC 8: Gestionar reporte de evaluación**
  - T 8.1: Mostrar reporte de evaluación
  - T 8.2: Exportar a formato pdf
- **EC 9: Consultar historial de evaluaciones**
- **EC 10: Notificar resultados de la evaluación**
  - T 10.1: Introducir datos de contacto
- **EC 11: Graficar evaluaciones de un proyecto**



### 2.2.2 Priorización de los escenarios

La prioridad de los escenarios viene dada por la importancia que tenga para los usuarios y para el progreso de la aplicación. El proceso de priorizar la lista de escenarios se basa en la identificación de los que tienen mayor importancia para que sean implementados con un orden de prioridad teniendo en cuenta su nivel de complejidad. La calificación de la prioridad viene dada por tres niveles: bajo, medio y alto; que cuantitativamente serían 3, 4 y 5 respectivamente. Estos niveles pueden aplicarse también para la complejidad de los escenarios.

A continuación, en la Tabla 5 se muestra la relación entre los escenarios definidos y su prioridad.

Tabla 5: Priorización de los escenarios

EC	Escenario	Prioridad
1	Autenticar usuario	5
2	Gestionar métrica	5
3	Gestionar datos del proyecto	5
4	Seleccionar métrica	5
5	Cargar fichero	5
6	Procesar datos del fichero	5
7	Calcular métrica	5
8	Gestionar reporte de evaluación	4
9	Consultar historial de evaluaciones	3
10	Notificar resultados de la evaluación	4
11	Graficar evaluaciones de un proyecto	4

Como se puede observar algunos escenarios cuentan con una prioridad media o baja, y es que a raíz de la propia realización de la priorización de los escenarios el equipo identifica que los EC 8, 9 ,10 y 11 dependen de la realización de los EC del 1 al 7, lo que no implica que su importancia sea menor, sino que su desarrollo depende de la implementación de los demás escenarios. En el caso de la prioridad del EC 9 se determina como baja debido a que su implementación no es de alta complejidad.

### 2.2.3 Requisitos de calidad del servicio

Los requisitos de calidad del servicio forman una parte significativa de las características del sistema. Son importantes para que los clientes puedan valorar las especificaciones no funcionales del producto, representando las restricciones o limitaciones con las cuales debe operar el sistema, dentro de sus características se encuentran: requerimientos de hardware y de software, rendimiento, accesibilidad,

## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

disponibilidad, utilidad y facilidad de mantenimiento. A continuación se definen los requisitos de calidad del servicio que se identificaron para el desarrollo del sistema propuesto.

### Software

- Se debe utilizar la máquina virtual de Java 6.0 o superior.
- Sistema operativo multiplataforma.
- Navegador: *Internet Explorer* (superior a la versión 6.0), *Mozilla Firefox* y *Chrome*.

### Restricciones en el diseño y la implementación

- El SGDB deberá ser MySQL 5.5.16.
- Servidor web Apache 2.2.21.
- El sistema debe utilizar como herramienta de desarrollo *Pentaho Data Integration* para llevar a cabo el proceso de integración de datos ETL.

### Usabilidad

- Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo.
- El sistema contará con una estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación, contando con un menú lateral que permitirá el acceso rápido a la información.

### Seguridad

- La aplicación será usada por personas autorizadas según el rol con el que se autentique y el nivel de privilegio.

### Disponibilidad

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

#### **2.2.4 Plan de iteraciones**

La estimación del tiempo que tomará al programador ejecutar la codificación de cada uno de los escenarios contribuye a determinar una correcta y adecuada planificación del desarrollo del sistema. En dependencia del orden de prioridad de cada escenario se determina cuáles se implementarán en las primeras iteraciones.

- **Iteración #1:** En esta primera iteración se propone codificar los escenarios que tienen una alta prioridad, ya que son esenciales para el funcionamiento del sistema.
- **Iteración #2:** En esta iteración se codificarían los escenarios que tienen una media o baja prioridad.

En la Tabla 6 se puede observar el nombre de los escenarios, su prioridad, el esfuerzo en días y su distribución por iteraciones para su implementación.

## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

Tabla 6: Planificación de los escenarios

Nº	Escenario	Prioridad	Esfuerzo (días)	Iteración
1	Autenticar usuario	5	10	1
2	Gestionar métrica	5	10	1
3	Gestionar datos del proyecto	5	10	1
4	Seleccionar métrica	5	10	1
5	Cargar fichero	5	12	1
6	Procesar datos del fichero	5	12	1
7	Calcular métrica	5	12	1
8	Gestionar reporte de evaluación	4	10	2
9	Consultar historial de evaluaciones	3	7	2
10	Notificar resultados de la evaluación	4	10	2
11	Graficar evaluaciones de un proyecto	4	10	2

### 2.3 Descripción de los escenarios

En este epígrafe se realiza la descripción de los escenarios según las necesidades, contribuyendo al desarrollo de la aplicación. A continuación, en la Tabla 7 se muestra el escenario “Gestionar reportes de las evaluaciones”, los demás pueden ser consultados en el **Anexo 1**.

Tabla 7: Descripción del escenario “Gestionar reporte de evaluación”

<b>Nombre del escenario:</b> Gestionar reporte de evaluación		<b>Identificador:</b> EC 8
<b>Objetivo del escenario:</b> Permitir realizar operaciones sobre el reporte obtenido		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 2da	<b>Prioridad:</b> 4	<b>Complejidad:</b> 4
<b>Precondiciones:</b> Debe existir al menos un proyecto evaluado		
<b>Descripción:</b> Una vez que el sistema muestra el reporte el evaluador podrá exportarlo y el sistema le permitirá escoger la dirección física donde desea guardarlo o abrirlo directamente		
<b>Validaciones:</b> N/P		
<b>Prototipo de interfaz de usuario:</b>		

**Requisitos de calidad del servicio**

- ✓ SGBD MySQL 5.5.16 o superior
- ✓ Máquina virtual de Java 6.0
- ✓ *Pentaho Data Integration* 4.2.0
- ✓ Servidor web Apache 2.2.21

### 2.3.1 Especificaciones de tareas por escenario

Algunos escenarios se desglosan en tareas y precisamente en este epígrafe se describen las tareas por escenarios. A continuación se muestra la especificación de las tareas del escenario del epígrafe anterior.

➤ **Especificación de tareas del escenario: “Gestionar reporte de las evaluaciones”.**

El escenario se divide en dos tareas fundamentales: “Mostrar reporte de evaluación” y “Exportar reporte a formato pdf”. En la Tabla 8 se muestra la tarea “Mostrar reporte de evaluación”


**Tabla 8: Descripción de la tarea “Mostrar reporte de evaluación”**

<b>Nombre del escenario:</b> Mostrar reporte de evaluación		<b>Identificador:</b> T 8.1
<b>Objetivo del escenario:</b> Mostrar los indicadores de las métricas		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 2da	<b>Prioridad:</b> 4	<b>Complejidad:</b> 4
<b>Precondiciones:</b> Debe haberse evaluado un proyecto		
<b>Descripción:</b> El sistema muestra un reporte con los valores de las métricas aplicadas, así como el nivel general de seguridad del proyecto evaluado		
<b>Validaciones:</b> N/P		

## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

**Prototipo de interfaz de usuario:**

**Resultado de la evaluación**

 Generar PDF


<b>Nombre de la Aplicación</b>	<b>Nombre del Evaluador</b>
SegMat	Antonio
<b>Métricas</b>	<b>Resultado</b>
Controlabilidad de acceso	120
Detección de vulnerabilidades	1
Identificación de Riesgos	0.84
<b>Nivel de la Métrica</b>	
Bajo	
Muy Alto	
Alto	
<b>Nivel General de Seguridad</b>	
Medio	

<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>
-------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

A continuación la Tabla 9 muestra las especificaciones de la tarea de exportar en formato pdf el reporte de la evaluación:

**Tabla 9: Descripción de la tarea “Exportar reporte a formato pdf”**

<b>Nombre del escenario:</b> Exportar a formato pdf		<b>Identificador:</b> T 8.2
<b>Objetivo del escenario:</b> Obtener un informe en formato pdf de la evaluación realizada		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 2da	<b>Prioridad:</b> 4	<b>Complejidad:</b> 4
<b>Precondiciones:</b> Debe haberse evaluado la aplicación		
<b>Descripción:</b> El escenario comienza cuando el usuario selecciona la opción “Exportar a pdf”, entonces el sistema guarda los resultados de la evaluación y elabora el informe en el formato seleccionado		
<b>Validaciones:</b> Verificar que no quede ningún campo del informe vacío		
<b>Prototipo de interfaz de usuario:</b>		

Resultado de la evaluación	
 Generar PDF	
Requisitos de calidad del servicio	<ul style="list-style-type: none"><li>✓ SGBD MySQL 5.5.16 o superior</li><li>✓ Máquina virtual de Java 6.0</li><li>✓ <i>Pentaho Data Integration</i> 4.2.0</li><li>✓ Servidor web Apache 2.2.21</li></ul>

Para conocer las especificaciones de las tareas correspondientes a cada escenario se recomienda consultar la sección del **Anexo 2**.

### 2.4 Elementos del diseño del sistema

Con el diseño del sistema se logra la definición de las funcionalidades que contribuyen a la construcción de un software apto para satisfacer las expectativas de los clientes. Esto se logra mediante la modelación del sistema de manera que cumpla con lo establecido por los escenarios y los requisitos de calidad del servicio. Permite la interpretación de las funcionalidades definidas, en busca de obtener durante la construcción del sistema un bajo nivel de cambios.

En el capítulo anterior se definió el uso del CMS Drupal para dar solución al problema planteado, a continuación, teniendo en cuenta sus particularidades, se describen algunos elementos como la arquitectura y los patrones de diseño.

#### 2.4.1 Especificación de la arquitectura a utilizar

Según (Gil Rodríguez, 2011) explica en su curso de creación y gestión de portales web en Drupal 7 que el CMS Drupal cuenta con una lógica programada en PHP, siguiendo un modelo de programación estructurada, haciendo uso de un sistema de BD relacional. El código que constituye el núcleo de Drupal está formado por un conjunto de librerías que permiten gestionar los procesos de arranque del sistema, ofreciendo además un conjunto de servicios que permiten integrar las funcionalidades adicionales de los módulos, servicios como conexión y administración de la BD, tratamiento de imágenes, internacionalización, soporte para la codificación y un potente entorno de integración de utilidades.

Drupal es, por tanto, un sistema con una arquitectura modular que permite ampliar sus funcionalidades a través de unos métodos uniformes de desarrollo e integración de nuevos módulos. En última instancia un módulo consiste en un conjunto de archivos con código PHP, que utiliza la arquitectura y las APIs de Drupal para incorporar nuevas características funcionales al sitio web.

En la Figura 4 se puede observar de forma esquemática los elementos que conforman la arquitectura de un sistema Drupal.

## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

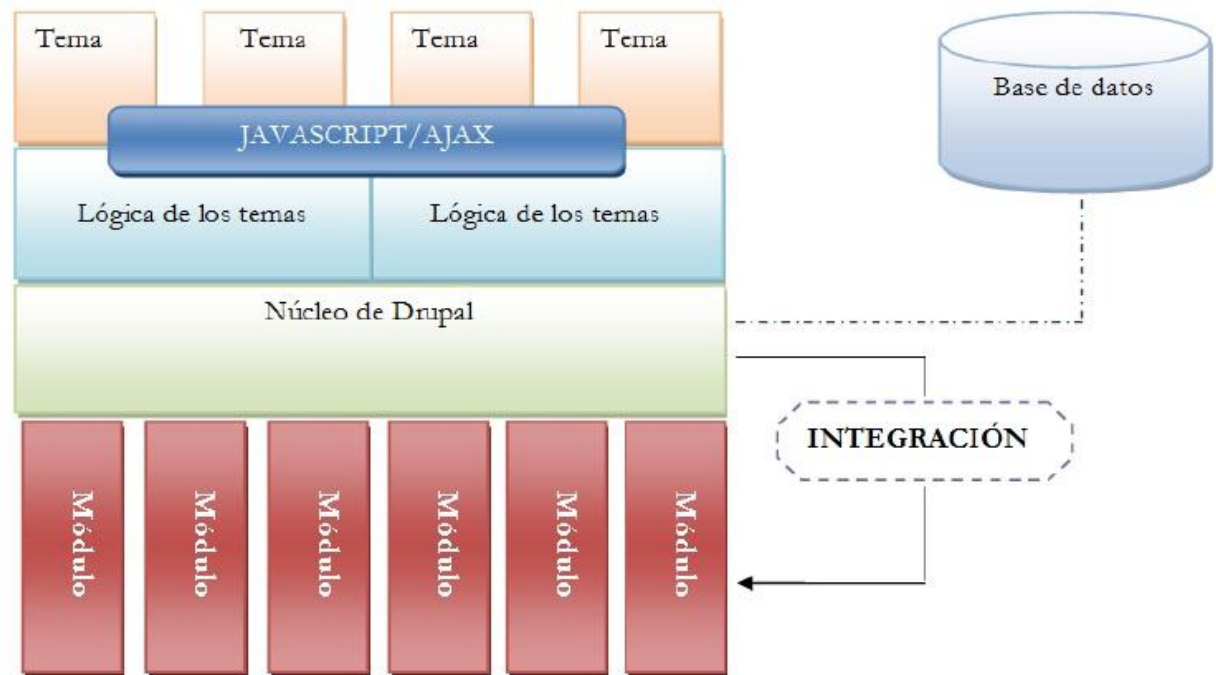


Figura 4: Arquitectura de Drupal (Gil Rodríguez, 2011)

El patrón de arquitectura a utilizar en el presente sistema es el Modelo-Vista-Controlador (MVC), ya que permite realizar la programación multicapa, separando en tres componentes distintos los datos de una aplicación, la interfaz del usuario y la lógica de control. Este patrón se ve usualmente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el SGBD y el controlador representa la lógica del negocio. Por lo que los niveles quedan formados de la siguiente manera (Bahit, 2010):

- **Modelo:** representa la información con la que trabaja la aplicación, o sea, su lógica de negocio.
- **Vista:** convierte el modelo en una página web que facilita al usuario interactuar con ella.
- **Controlador:** es el encargado de procesar las interacciones del usuario y ejecuta los cambios adecuados en el modelo o en la vista.

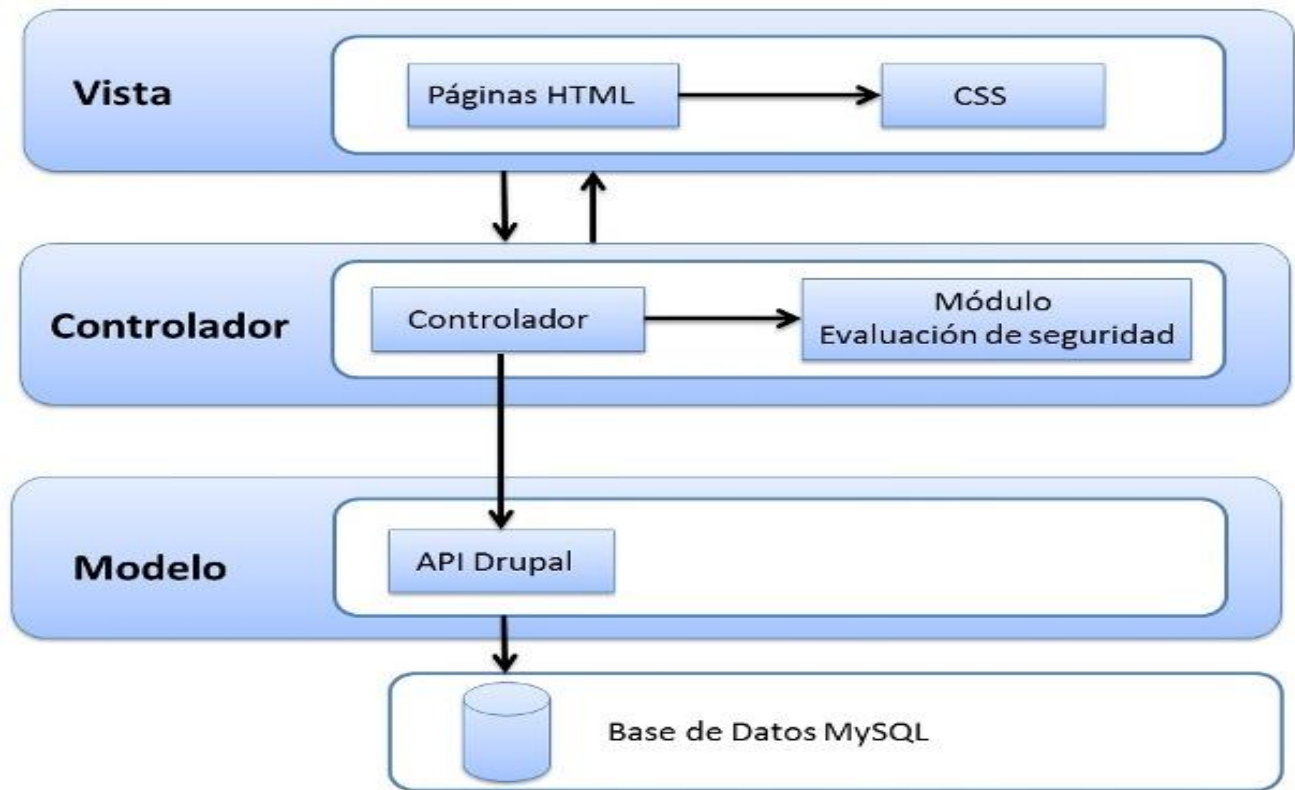


Figura 5: Arquitectura del sistema

### 2.4.2 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Para el funcionamiento del “Sistema para la evaluación de la seguridad” se utilizan algunos patrones de diseño propiamente de sistemas orientados a objetos, tal es el caso de los patrones GoF (*Gang of Four*). Estos se dividen en tres tipos de clasificaciones (Gamma, 1995):

- **Creacionales:** los patrones creacionales abstraen el proceso de creación de instancias y ocultan los detalles de cómo los objetos son creados o inicializados.
- **Estructurales:** los patrones estructurales se ocupan de cómo las clases y objetos se combinan para formar grandes estructuras y proporcionan nuevas funcionalidades.
- **Comportamiento:** los patrones de comportamiento están relacionados con los algoritmos y la asignación de responsabilidades entre los objetos. Son utilizados para organizar, manejar y combinar comportamientos.

Algunos de los patrones que se manifiestan en el funcionamiento de la solución son:

- **Decorador (*decorator*):** pertenece a la categoría de los patrones estructurales. Patrón que responde a la necesidad de añadir dinámicamente funcionalidades a un objeto. Es usado para utilizar funciones de otros módulos o del Core de Drupal en el módulo evaluación, permitiendo no



## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

tener que crear estas funciones nuevamente sino decorarlas de acuerdo con la necesidad que se tenga.

- **Comando (*command*):** tributa a la categoría de los patrones de comportamiento. Este permite solicitar una operación a un objeto sin conocer realmente el contenido de la misma, ni el receptor real de esta. Para ello se encapsula la petición como un objeto, con lo que además se facilita la parametrización de los métodos. Este patrón es utilizado para permitir llevar a cabo la ejecución de ciertas tareas pasando como parámetro el operador, esta es la base fundamental del funcionamiento de los *hooks* (ganchos) usados en el actual sistema.
- **Observador (*observer*):** está dentro de la categoría de los patrones de comportamiento. 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. Este patrón se usa para actualizar de manera automática cuando se modifican los nodos a los que se referencian en el menú lateral del módulo evaluación.
- **Instancia única (*singleton*):** pertenece a la categoría de los patrones creacionales. Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Es usado para crear un solo objeto de un módulo del “Sistema para la evaluación de la seguridad” usando este donde se desee conocer sus atributos o funcionalidades.
- **Puente (*bridge*):** este patrón permite comunicar la capa de datos con los módulos contenidos por el sistema, de manera tal que cada módulo es independiente de la BD pero que puede interactuar con esta mediante un puente, además este patrón brinda soporte para más sistemas de BD sin la necesidad de modificar el código.
- **Cadena de responsabilidades (*chain of responsibility*):** el sistema de menú del “Sistema para la evaluación de la seguridad” sigue el patrón cadena de responsabilidades. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados, y qué función se llama para hacer el trabajo. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud. Si el elemento de menú no puede manejar la petición, se pasa de la cadena. Esto continúa hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario, o la cadena se haya agotado.

Con estos patrones el diseño del sistema se beneficia de flexibilidad y extensibilidad, además facilita su funcionamiento con las características propias de los sistemas orientados a objetos.

### 2.4.3 Modelo de datos

Un modelo de datos es aquel lenguaje utilizado para describir la estructura física y lógica de la información persistente manejada por el sistema. Determina la estructura de la información, con el objetivo de mejorar

# CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

la comunicación y la precisión en aplicaciones que usan e intercambian datos. A continuación, se muestran las entidades del modelo de datos con las que interactúa la aplicación.

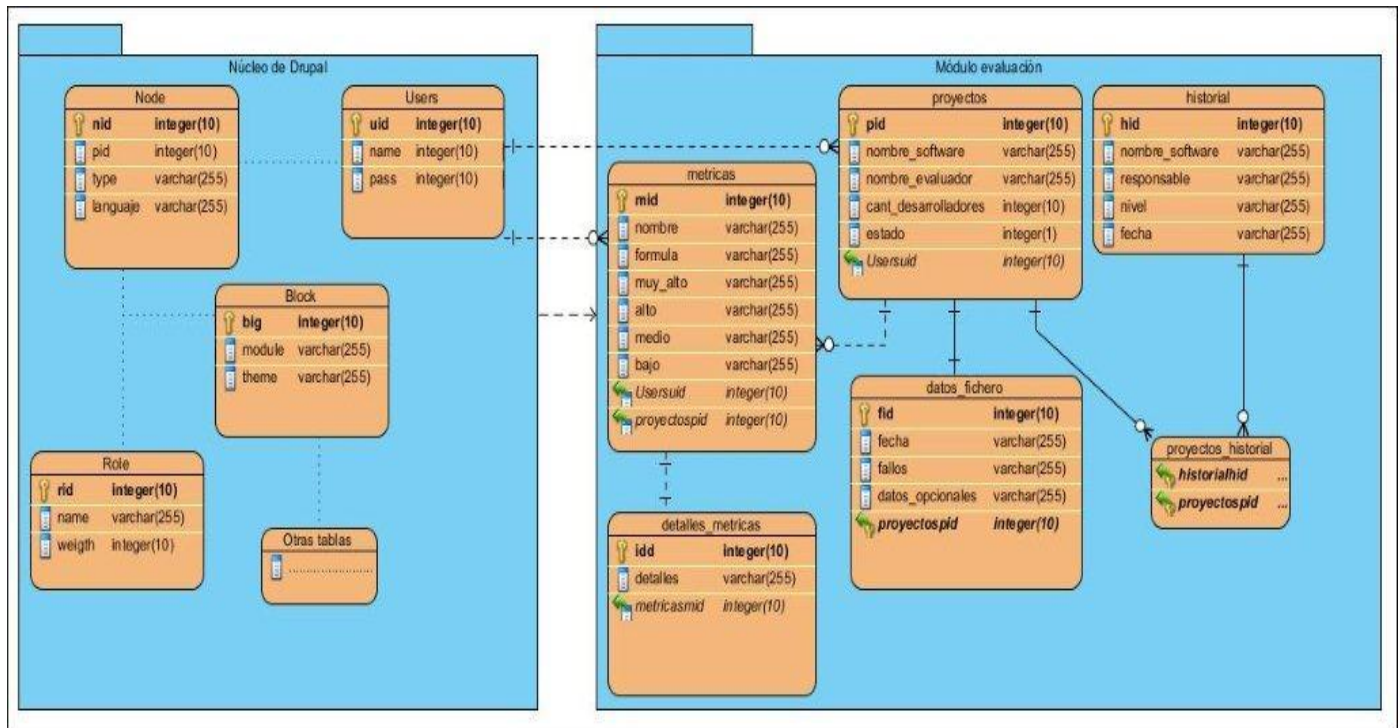


Figura 6: Modelo de datos del sistema

La imagen anterior muestra la distribución de las tablas en la BD del “Sistema para la evaluación de la seguridad”. Nótese que en la parte izquierda se muestran algunas tablas que son vitales para el funcionamiento básico del CMS y a la derecha se muestran las tablas del módulo de evaluación de seguridad con sus atributos.

## 2.4.4 Descripción de las tablas

En este epígrafe se muestra la descripción de las tablas que componen el modelo de datos del sistema, específicamente el módulo de evaluación.

Tabla 10: Descripción de la tabla “proyectos”

<b>Nombre</b>	proyectos	
<b>Descripción</b>	Tabla para guardar los datos de los proyecto	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
pid	Integer	Llave primaria de la tabla
nombre_software	Varchar	Nombre del software
nombre_evaluador	Varchar	Nombre del evaluador
cant_desarrolladores	Integer	Cantidad de desarrolladores del proyecto

## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

estado	Integer	Estado del proyecto
Usersuid	Integer	Usuario de la persona que creó la métrica

**Tabla 11: Descripción de la tabla “historial”**

<b>Nombre</b>	historial	
<b>Descripción</b>	Tabla para almacenar los historiales de los proyectos	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
hid	Integer	Llave primaria de la tabla
nombre_software	Varchar	Nombre del software
responsable	Varchar	Nombre del líder del proyecto
nivel	Varchar	Nivel general de la evaluación
fecha	Varchar	Fecha en la que se evaluó el software

**Tabla 12: Descripción de la tabla “proyectos\_historial”**

<b>Nombre</b>	proyectos_historial	
<b>Descripción</b>	Tabla que refleja la relación de mucho a mucho que existe entre las tablas proyectos e historial	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
historialhid	Integer	Llave foránea que representa la primaria de la tabla “historial”
proyectospid	Integer	Llave foránea que representa la primaria de la tabla “proyectos”

**Tabla 13: Descripción de la tabla “datos\_fichero”**

<b>Nombre</b>	datos_fichero	
<b>Descripción</b>	Tabla que almacenas los datos arrojados por las pruebas de seguridad, los cuales serán procesados para dar la evaluación del software	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
fid	Integer	Llave primaria de la tabla
fecha	Varchar	Fecha en que se realizan las pruebas de seguridad
fallos	Varchar	Vulnerabilidades encontradas mediante las pruebas de seguridad
datos_opcionales	Varchar	Otros datos arrojados por las pruebas de seguridad
proyectospid	Integer	Llave foránea que representa la llave primaria de la tabla “proyectos”

## CAPÍTULO 2: CARACTERIZACIÓN Y DISEÑO DEL SISTEMA

Tabla 14: Descripción de la tabla “metricas”

<b>Nombre</b>	metricas	
<b>Descripción</b>	Tabla almacena las métricas que pueden ser utilizadas para la evaluación de un proyecto	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
mid	Integer	Llave primaria de la tabla
nombre	Varchar	Nombre que le corresponde a la métrica
formula	Varchar	Fórmula para calcular la métrica
muy_alto	Varchar	Nivel de evaluación de la métrica
alto	Varchar	Nivel de evaluación de la métrica
medio	Varchar	Nivel de evaluación de la métrica
bajo	Varchar	Nivel de evaluación de la métrica
Usersuid	Integer	Usuario de la persona que creó la métrica
proyectospid	Integer	Llave foránea que representa la llave primaria de la tabla “proyectos”

Tabla 15: Descripción de la tabla “detalles\_metricas”

<b>Nombre</b>	detalles_metricas	
<b>Descripción</b>	Esta tabla almacenará los detalles de las métricas	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
idd	Integer	Llave de la tabla
detalles	Varchar	Detalles de la métrica
metricasmid	Integer	Llave foránea que representa la llave primaria de la tabla “metricas”

### Conclusiones del capítulo

La metodología de desarrollo permitió establecer por fases cada una de las actividades que se realizan durante el proceso de desarrollo. Describir la visión general del sistema, permitió la realización de una planificación del desarrollo, determinando el tiempo estimado de duración. La especificación de los escenarios hizo posible conocer al detalle cada una de las funcionalidades del sistema. Con la realización del modelo de datos se hizo posible materializar la visión de la BD a utilizar. En este capítulo se crean las bases necesarias para la posterior implementación del sistema.

## CAPÍTULO 3: DESARROLLO Y ESTABILIZACIÓN DEL SISTEMA

En este capítulo los estándares de codificación son establecidos a fin de lograr la implementación del sistema, se crean los diagramas necesarios para guiar la implementación, y finalmente se realizan las pruebas necesarias que validan la calidad de la solución.

### 3.1 Fase de desarrollo

El objetivo principal de esta fase es desarrollar la arquitectura del sistema a desarrollar, además se define la organización que tendrá el código y por último se materializa la implementación del sistema en términos de componentes, dígame *scripts*, ejecutables, ficheros de código fuente, entre otros.

#### 3.1.1 Estándares de codificación

Un estándar de codificación es aquel estilo de codificación que se establece y aplica para escribir el código. Para su definición se valoran elementos como las características propias del lenguaje de programación, los recursos que se utilizarán, el tipo de programa que se va a implementar e incluso el estilo personal del programador. Al iniciarse un proyecto se debe establecer un estándar de codificación asegurándose así de que todos los programadores del proyecto trabajen de manera uniforme.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento. Además, si un estándar de codificación bien definido se aplica de forma continua, se utilizan técnicas de programación apropiadas, y posteriormente, se efectúan revisiones del código de rutinas, existe la posibilidad de que un proyecto de software se convierta en un sistema fácil de comprender y de mantener. La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantenibilidad del código es la facilidad con que el software pueda modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento.

En el desarrollo del actual sistema, se utilizan algunos de los estándares de codificación (Drupal.org, 2013) propuestos por el CMS Drupal, estos son:

#### ➤ Funciones y variables

Las funciones y variables deben ser nombradas usando minúsculas, y las palabras deben ser separadas utilizando un guión bajo. Las funciones deben tener además el nombre del grupo/módulo como prefijo, para evitar el conflicto de nombres entre los módulos. Este estándar se evidencia en el sistema con los siguientes ejemplos:

- ✓ variables: \$nombre\_metrica, \$formula
- ✓ funciones: modulo\_evaluacion\_metricas()

### ➤ Variables persistentes

Las variables persistentes (variables/configuraciones definidas usando las funciones de Drupal *variable\_get()/variable\_set()*) deben ser nombradas usando minúsculas y las palabras deben ser separadas utilizando un guión bajo. Deben usar el nombre del grupo/módulo como prefijo, para evitar el conflicto de nombre entre los módulos. Ejemplo:

- ✓ `variable_set('modulo_evaluacion_reporte', $resultados)`

### ➤ Operadores

Todos los operadores binarios (operadores que están entre dos valores), como `+`, `-`, `+=`, `!=`, `==`, `>`, entre otros, deben tener un espacio antes y después del operador para facilitar la lectura. Por ejemplo, una asignación debe tener el formato `$var = $foo`; en vez de `$var=$foo`. Los operadores unarios (operadores que operan sobre un solo valor), como `++`, no deben tener espacios entre el operador y la variable o número que se utiliza.

### ➤ Punto y coma

El lenguaje PHP requiere puntos y comas al final de la mayoría de las líneas, pero permite ser omitidos al final de bloques de código. En particular para una línea de bloques PHP.

```
<?php
print();
$tax;?>
```

### ➤ Inclusión de código

En cualquier parte que se esté incluyendo incondicionalmente un archivo de una clase, se usa *required\_once*. En cualquier parte donde se esté condicionalmente incluyendo un archivo de clase (por ejemplo, métodos de fábrica) se usa *include\_once*. Cualquiera de estas asegurará que el archivo sea incluido únicamente una vez.

### 3.1.2 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Este diagrama describe la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software.

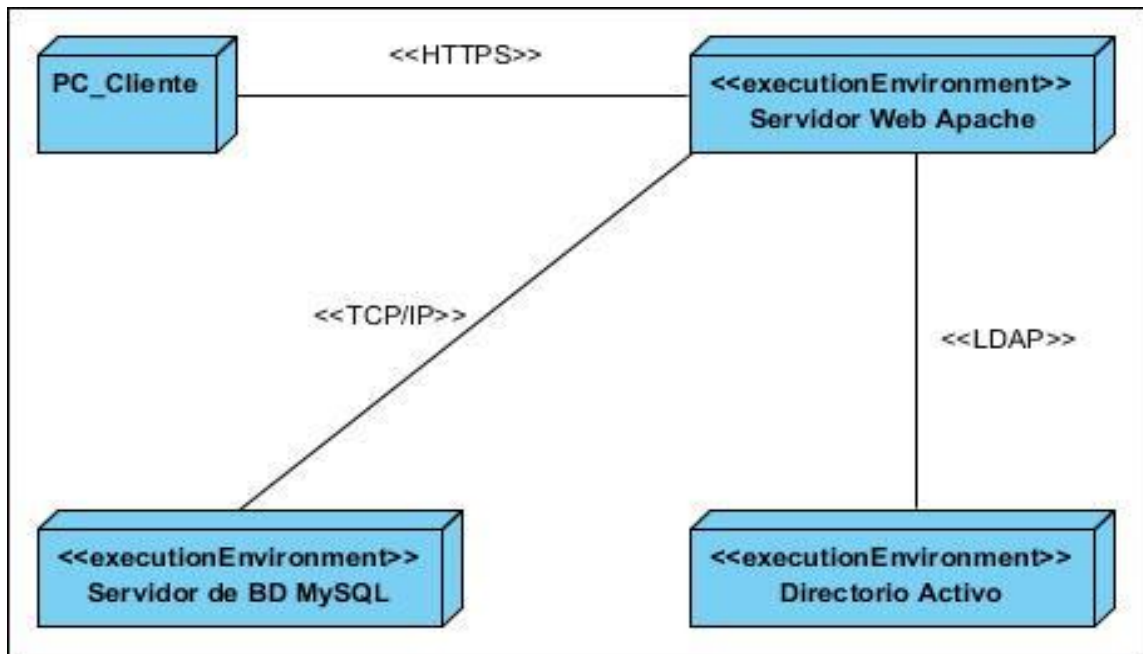


Figura 7: Diagrama de despliegue

La distribución física de las relaciones entre los diferentes nodos queda reflejada en la figura anterior; esta distribución la compone tres servidores y una PC cliente. De manera más explícita, el Servidor Web Apache se comunica con el Servidor MySQL a través del protocolo TCP/IP con el fin de realizar las operaciones sobre los datos del sistema. El mismo Servidor Web se comunica mediante el protocolo LDAP con el Directorio Activo de la Universidad para realizar la autenticación de los usuarios y finalmente el evaluador se puede conectar al Servidor web para realizar sus peticiones mediante el protocolo HTTPS.

### 3.1.3 Interfaz gráfica

Las interfaces gráficas permiten al cliente tener una visión del sistema antes de interactuar directamente con este. En el presente epígrafe se muestran algunas interfaces que representan el inicio del uso del sistema. A continuación se brinda una vista inicial del mismo, donde se señala la zona de autenticación.

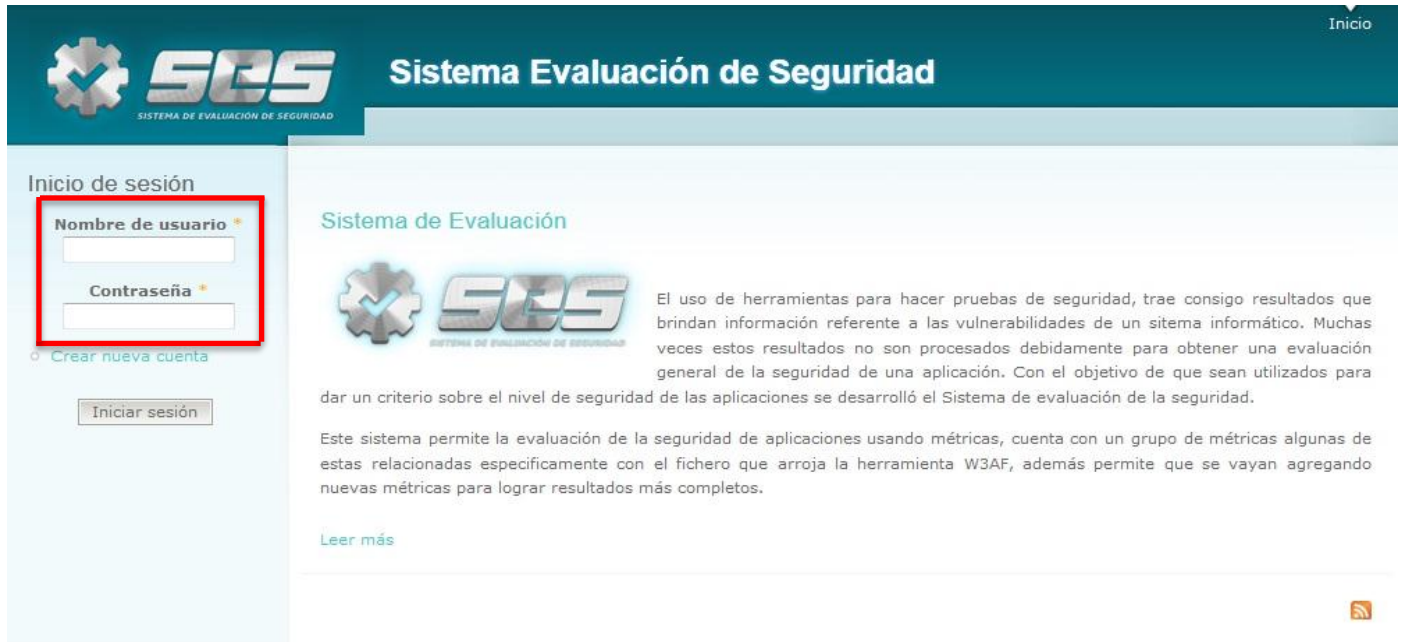


Figura 8: Interfaz gráfica “Autenticar usuario”

Luego de autenticarse el usuario ya puede tener acceso a las funcionalidades que brinda el sistema.

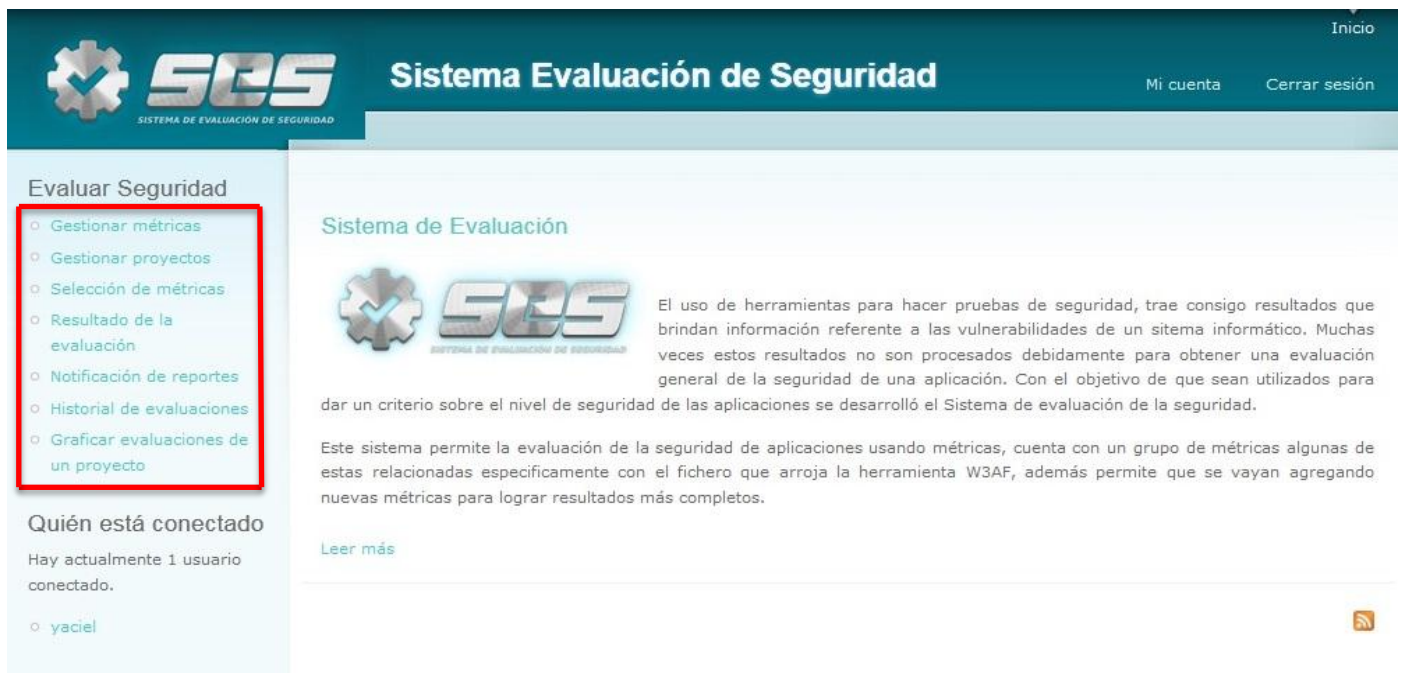
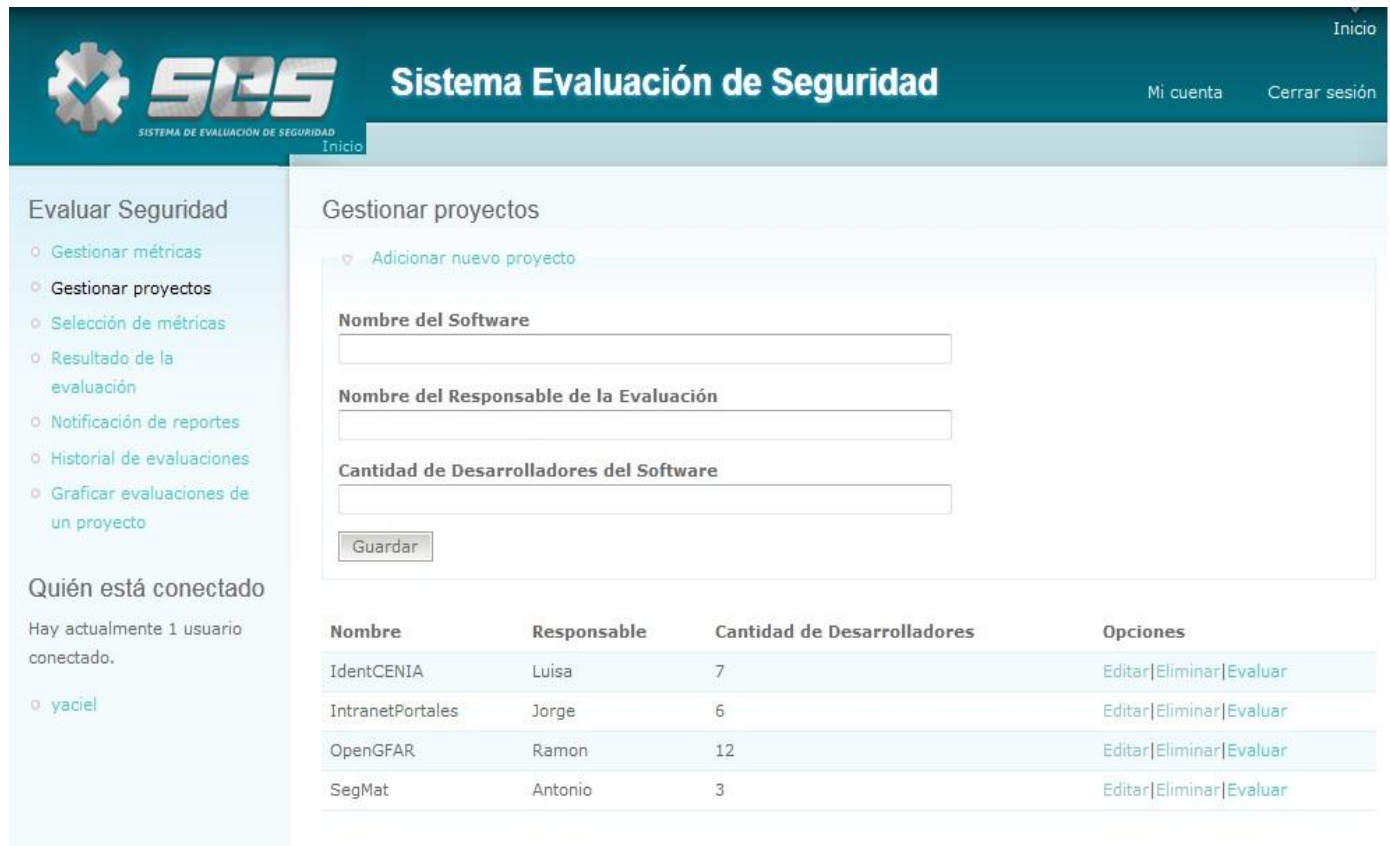


Figura 9: Interfaz principal del sistema

En el caso de seleccionar la gestión de un nuevo proyecto la interfaz que se muestra es la siguiente:





**Figura 10: Interfaz gráfica “Gestionar proyecto”**

En la figura anterior el evaluador puede adicionar un nuevo proyecto, posteriormente ya puede hacer uso de las opciones de editar, evaluar o eliminar y en caso de que el proyecto ya exista puede realizar cualquier opción directamente. En el **Anexo 3** se explican los demás flujos de interfaces que acompañan al presente sistema.

### 3.2 Fase de estabilización

Con la realización de pruebas es posible controlar los requisitos mínimos de operatividad y garantizar así la calidad del producto, por lo que es común que sean fundamentales dentro de cada una de las etapas de desarrollo de un software. Las mismas constituyen una actividad en la cual un sistema es ejecutado bajo condiciones específicas, los resultados son observados y registrados a fin de determinar si los errores ocurren cuando no tendrían que ocurrir, realizando a partir de aquí una evaluación del estado del sistema. El primer paso que la metodología identifica para realizar las pruebas al sistema, es definir el tipo de prueba que se va a desarrollar. Para esta solución se determina realizar las pruebas unitarias mediante la técnica de caja blanca y pruebas funcionales mediante la técnica de caja negra.

### 3.2.1 Pruebas unitarias. Aplicación de pruebas de caja blanca

Al desarrollar un nuevo software una de las pruebas que no se deben pasar por alto son las pruebas unitarias, específicamente la técnica de pruebas de caja blanca. Para realizar este tipo de pruebas el personal debe estar familiarizado en el uso de herramientas con este fin, además de conocer el lenguaje de programación en el que se está desarrollando el sistema. Las pruebas unitarias permiten conocer si el funcionamiento de un módulo de código es correcto, con el fin de garantizar que cada módulo funcione por separado de la manera correcta. El objetivo de estas pruebas es alejar determinadas partes del código y demostrar que no tienen errores, probándole de este modo al programador que la solución está libre de errores lógicos de programación, esto se evidencia si el probador introduce determinados datos al sistema y los valores que se obtienen son los esperados.

En la actualidad existen una gran cantidad de herramientas que tienen como objetivo la realización de estas pruebas. PHPUnit es un ejemplo de estas, y precisamente es la herramienta seleccionada por el equipo de trabajo para realizar las pruebas unitarias a la aplicación. PHPUnit 3.6.10 es uno de los *framework* más importantes para realizar pruebas unitarias ya que permite crear y ejecutar juegos de *tests* unitarios de una forma sencilla.

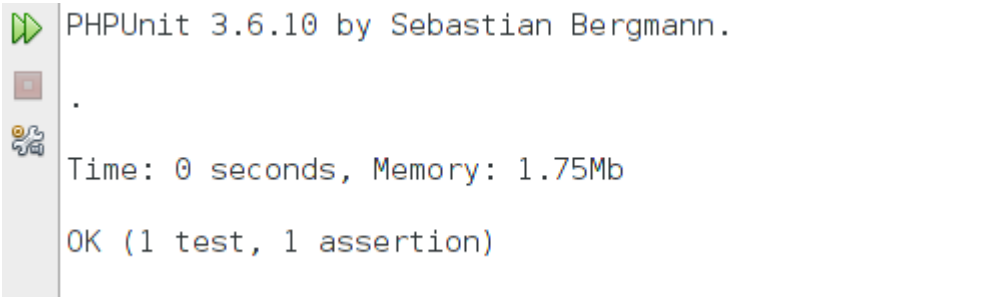
A continuación en la Figura 11 se evidencia un ejemplo del resultado de las pruebas, en este caso a las funciones “test\_Realizar\_escala” y “test\_Modulo\_evaluacion\_nivel\_general”.

```
class evaluacion_modulo_Test extends PHPUnit_Framework_TestCase {  
    public function test_Realizar_escala() {  
        $modulo_evaluacion = new Modulo_Evaluacion();  
        $valor = 0.7;  
        $resultado = "Medio";  
        $this->assertEquals($resultado, $modulo_evaluacion->realizar_escala($valor));  
    }  
    public function test_Modulo_evaluacion_nivel_general() {  
        $modulo_evaluacion = new Modulo_Evaluacion();  
        $niveles = array("Bajo", "Medio", "Alto");  
        $resultado = "Medio";  
        $this->assertEquals($resultado, $modulo_evaluacion->modulo_evaluacion_nivel_general($niveles));  
    }  
}
```

Figura 11: Prueba unitaria a la función “test\_Realizar\_escala” y “test\_Modulo\_evaluacion\_nivel\_general”

## CAPÍTULO 3: DESARROLLO Y ESTABILIZACIÓN DEL SISTEMA

**Tabla 16: Descripción de la prueba unitaria a la función “modulo\_evaluacion\_get\_proyecto”**

Prueba de unidad		
<b>Nombre de la prueba:</b> test_Realizar_escala		
<b>Estado:</b> Satisfactoria	<b>Tipo:</b> Caja blanca	<b>Última ejecución:</b> 15-05-2013
<b>Ejecutado por:</b> Yaciel Leyva Góngora		<b>Verificado por:</b> Yayneris Zambrana
<b>Descripción:</b> Esta prueba permite saber si la función para un valor dado devuelve correctamente el nivel que le corresponde según una escala.		
<b>Entrada:</b> \$valor (valor de la métrica)		
<b>Criterio de Aceptación:</b> Se muestra en los resultados de la evaluación el nivel según la escala para ese valor.		
<b>Resultado:</b>		
 <pre> PHPUnit 3.6.10 by Sebastian Bergmann. . Time: 0 seconds, Memory: 1.75Mb  OK (1 test, 1 assertion) </pre>		

**Tabla 17: Descripción de la prueba unitaria a la función “cargar\_datos\_bd”**

Prueba de unidad		
<b>Nombre de la prueba:</b> test_Modulo_evaluacion_nivel_general		
<b>Estado:</b> Satisfactoria	<b>Tipo:</b> Caja blanca	<b>Última ejecución:</b> 15-05-2013
<b>Ejecutado por:</b> Yaciel Leyva Góngora		<b>Verificado por:</b> Yayneris Zambrana
<b>Descripción:</b> Con esta prueba se verifica que el sistema a través de esta funcionalidad brinde el nivel general de seguridad para la aplicación que se está evaluando.		
<b>Entrada:</b> \$niveles (arreglo con los niveles de cada métrica)		
<b>Criterio de Aceptación:</b> Que se escoja correctamente el nivel general de seguridad para el conjunto de niveles a tener en cuenta.		
<b>Resultado:</b>		

```

PHPUnit 3.6.10 by Sebastian Bergmann.
.
Time: 0 seconds, Memory: 1.75Mb

OK (1 test, 1 assertion)
    
```

### 3.2.2 Aplicación de pruebas de caja negra

Las pruebas de caja negra (Ruiz Tenorio, 2010) permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales o escenarios de un programa. Estas pruebas no son una alternativa a las técnicas de pruebas de caja blanca sino más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que estos últimos no pueden. La técnica de pruebas de caja negra es utilizada para validar las funcionalidades del sistema sin fijarse en el funcionamiento interno de este.

En la Tabla 18 se muestra la descripción del caso de prueba para el escenario “Cargar fichero”

**Tabla 18: Descripción del caso de prueba “Cargar fichero”**

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC5. Cargar fichero	El evaluador selecciona la opción enviar fichero de pruebas, el sistema muestra el formulario donde se encuentran los elementos necesarios para que sea cargado el fichero en la BD	Ruta del archivo (Válido)	El sistema carga el fichero	Opción Selección de métrica/ Botón examinar
		Ruta del archivo (Inválido)	El sistema muestra un mensaje de error informando que el campo está vacío	
EC5.1 Seleccionar fichero	Se procede a seleccionar el fichero que posee el resultado de las pruebas de	Ruta del archivo (Válido)	En esta sección el sistema no muestra nada, solo carga la dirección del	Opción Selección de métrica/ Botón Examinar

## CAPÍTULO 3: DESARROLLO Y ESTABILIZACIÓN DEL SISTEMA

	seguridad realizadas al software que será evaluado		archivo seleccionado	
		Ruta del archivo (Inválido)	En este caso no sucede nada hasta que se pase a la otra sección de Subir el fichero	
EC 5.2 Subir fichero	El fichero es cargado en la BD	Ruta del archivo (Válido)	El sistema inserta los datos del fichero en la BD	Opción Selección de métrica/ Botón Enviar
		Ruta del archivo (Inválido)	El sistema muestra un mensaje de error informando que los datos del fichero no fueron insertados en la BD, debido a que el archivo cargado no es el correcto	

Los restantes casos de pruebas diseñados pueden ser consultados en el **Anexo 4**.

### 3.2.3 Resultado de las pruebas

Una vez que se realizan las pruebas se analizan los elementos más relevantes que se obtuvieron, representando así el mayor peso en el análisis de los resultados de las pruebas ejecutadas. Luego de vencer las dos iteraciones de pruebas al sistema, se evidencia en sentido general que las No Conformidades (NC) surgidas fueron totalmente solucionadas, por lo que el desarrollo del sistema no se ve afectado y de este modo la aplicación se encuentra lista para ser usada.

En la Figura 12 se muestra la relación entre la cantidad de NC generadas y las NC corregidas en cada iteración.

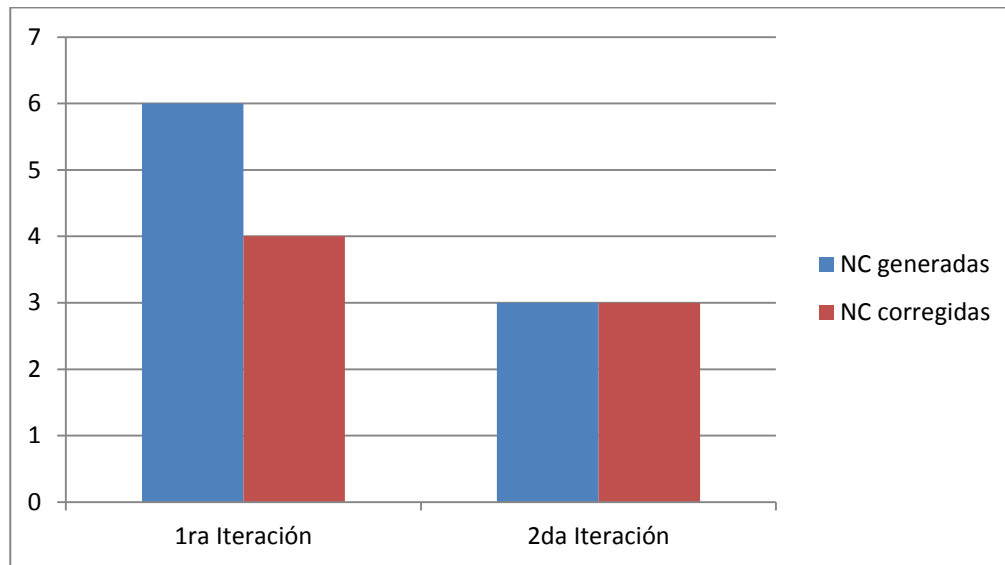


Figura 12: Iteraciones de pruebas

### 3.3 Análisis del cumplimiento del sistema

Durante el transcurso de esta investigación se realizó un análisis del proceso de evaluación de software, conociendo de este modo algunas vías para su aplicación, entre ellas la que se propone en el estándar ISO/IEC 14598, el cual permitió la realización del actual sistema. A continuación se definen por cada etapa los pasos realizados por el “Sistema de evaluación de la seguridad”:

#### ➤ Establecer requisitos de evaluación

- ✓ Se define como propósito contribuir a mejorar los niveles de seguridad para un software.
- ✓ A través de la funcionalidad "Gestionar proyectos" se establece el conjunto de aplicaciones a evaluar.
- ✓ Se basa en el modelo de calidad establecido por la ISO/IEC 9126.

#### ➤ Especificar evaluación

- ✓ Se realizó un estudio de las diferentes métricas seleccionando las que contienen indicadores de mayor impacto para las aplicaciones desarrolladas en el Departamento de Seguridad Digital.
- ✓ Con la implementación de la funcionalidad "Gestionar métricas" se establecen los criterios y la escala en la cual estarán contemplados los niveles de seguridad.
- ✓ A través de la funcionalidad "Seleccionar métricas" se escogen las métricas a usar en la evaluación.

### ➤ Diseñar evaluación

- ✓ Se define un plan de evaluación con los datos de las métricas usadas, sus niveles de seguridad así como el nivel general de seguridad para un software.

### ➤ Ejecutar evaluación

- ✓ Con la ejecución de la funcionalidad "Procesar datos de las pruebas" se obtienen los resultados de las métricas usadas.
- ✓ Se obtiene un nivel de seguridad para el valor de cada una de las métricas de acuerdo a los criterios antes definidos y un nivel general para la evaluación.
- ✓ Se compara a través de la funcionalidad "Graficar evaluaciones de un proyecto" para conocer el estado en el que se encuentra el software.

Una vez especificadas las acciones realizadas en cada una de las etapas del proceso de evaluación por el "Sistema para la evaluación de la seguridad", este tiene especial importancia para las soluciones informáticas del Departamento de Seguridad Digital, pues se podrá analizar y controlar la seguridad de los proyectos productivos del CISED. Además de contribuir en gran medida a conocer el nivel de seguridad de un software.

### Conclusiones del capítulo

Los estándares de codificación hicieron posible mantener de forma estandarizada los elementos para la implementación, lo que permitió contar con una programación legible y organizada. El diagrama de despliegue hizo posible tener una visión del funcionamiento del sistema. El diseño de las interfaces de usuario, permitió tener una vista previa de la aplicación. Con la realización de las pruebas se logró mitigar los errores que presentaba la aplicación, contribuyendo así a la calidad de la misma.

---

## CONCLUSIONES GENERALES

A partir de la necesidad de conocer el nivel de seguridad con que cuentan las aplicaciones desarrolladas en el departamento de Seguridad Digital se desarrolló el presente trabajo, del que se puede concluir que:

- Con la aplicación del proceso de evaluación establecido en el estándar ISO/IEC 14598 se potencia la evaluación de la seguridad que realiza el sistema, logrando cumplir con todas las etapas definidas para este proceso.
- La gestión de métricas garantiza flexibilidad al sistema obteniendo un mayor número de indicadores a tener en cuenta en la evaluación de la seguridad de un software.
- El desarrollo de este sistema brinda al Departamento de Seguridad Digital un mecanismo para la evaluación de la seguridad de sus aplicaciones, lo que impacta directamente en la producción de software de mayor calidad y seguridad.



---

## RECOMENDACIONES

Aunque se cumplieron los objetivos propuestos para el trabajo y el sistema cumplió las expectativas se recomienda:

- Extender el uso de la herramienta en el CISED, para posteriormente generalizarla en la universidad.

## REFERENCIAS BIBLIOGRÁFICAS

- Pressman, Roger S. 1998.** *Ingeniería del software. Un enfoque práctico.* 1998.
- Bahit, Eugenia. 2010.** eByte. *Introducción al patrón arquitectónico MVC.* [En línea] 2010.  
<http://www.eugeniabahit.com/mvc/>.
- Briand, L., Daly, J. y Differding, C. 1996.** "An experimental comparison of the maintainability of object-oriented and structured design documents". 1996.
- Cabrera, Lianet y Pompa, Enrique R. 2012.** *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información.* 2012.
- Corletti, Alejandro y Muñoz, Carmen de Alba. 2008.** *Métricas de Seguridad, Indicadores y Cuadro de Mando.* 2008.
- DECRETO-LEY No.199. 1999.** "DECRETO-LEY No.199/ SOBRE LA SEGURIDAD Y PROTECCION DE LA INFORMACION OFICIAL". 1999.
- Drupal.org. 2013.** Coding standars. [En línea] 2013. <http://drupal.org/coding-standards>.
- Fernández, Mellado y Rodríguez, Daniel. 2010.** *TECNIMAP, EVALUACION DE LA CALIDAD Y SEGURIDAD EN PRODUCTOS SOFTWARE.* 2010.
- Funes, Luis Enrique. 2011.** *Conociendo Netbeans Platform Introduccion.* 2011.
- Gamma, Eric. 1995.** *Design Patterns: Elements of Reusable Object Oriented Software.* 1995.
- Gil Rodríguez, Fran. 2011.** *Experto en Drupal 7 Nivel Avanzado.* 2011.
- Hernández Orallo, Enrique. 2011.** El Lenguaje Unificado de Modelado (UML). [En línea] 2011.  
<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
- IEEE. 1993.** "Software Engineering Standards Std. 6 10.12-1990." . 1993.
- ISO. 2010.** [En línea] 2010. <http://www.iso.org/iso/home/standards.htm>.
- ISO/IEC 14598-1:1999. 1999.** *Information technology - Software product evaluation -Part 1: General overview.* 1999.
- ISO/IEC. 1999.** Information technology - Software product evaluation -Part 1. 1999.
- López, Ana María. 2008.** INTRODUCCIÓN A LA CALIDAD DE SOFTWARE. [En línea] 2008.  
<http://revistas.utp.edu.co/index.php/revistaciencia/article/view/3241>.
- Mateu, Carles. 2004.** *Desarrollo de aplicaciones web.*  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/Desarrollo\\_de\\_Aplicaciones\\_Web.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Desarrollo_de_Aplicaciones_Web.pdf) : s.n., 2004.

- Mendoza Sánchez, María A. 2004.** Informatizate.net. [En línea] 2004.  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).
- Microsoft. 2005.** Tutorial: Seguimiento de elementos de trabajo. [En línea] 2005. <http://msdn.microsoft.com/es-es/library/ms181269%28v=vs.80%29.aspx>.
- Naramone, Elizabeth y Gerner, Jason. 2005.** *Beginning PHP5, Apache, and MySQL. Web Development.* 2005.
- Oficina Nacional de Normalización. 2005.** *NC-ISO-IEC 9126-1. Cuba : s.n. .* 2005.
- Pentaho. 2009.** Kettle Pentaho Data Integration. [En línea] 2009. <http://kettle.pentaho.org/>.
- Peña Romero, Yanisleydis y Fonseca Martínez, Jorge. 2012.** *Sistema de evaluación de seguridad utilizando el proceso de integración de datos ETL.* 2012.
- Ramírez , Lama y Juan, José. 2011.** *"Requisitos y Evaluación de Calidad de Productos de Software"*. 2011.
- Rodríguez Lorenzo, Orestes. 2010.** *Almacén de datos para los subsistemas de Reclutamiento y Potencial Humano.* 2010.
- Rubalcaba Betancourt, Maria de los Angeles y Zambrana Hernández, Yayneris. 2008.** *Medición de la calidad de Software durante el Proceso de Pruebas en el Proyecto Modernización del CICPC.* 2008.
- Ruiz Tenorio, Roberto. 2010.** *Las pruebas de software y su importancia en las organizaciones.* 2010.
- Sánchez Asenjo, Jorge. 2009.** Apuntes Completos sobre Sistemas gestores de Bases de Datos. [En línea] 2009.  
<http://ubuntuone.com/p/sqt>.
- Santana Mancilla, Pedro César . 2001.** *Taller de PHP.* 2001.
- Tesis-OyS. 2010.** [En línea] 2010. <http://www.tesis-oys.com.ar/content/drupal>.
- TIC. 2008.** Web Técnica de TIC. [En línea] 2008.  
<http://www.tic2.org/WebTecnica/Programacion/CSS/CSSDocEstructura/CSSDocEstructura.htm>.
- Vaquero, Miguel. 2010.** *deciencias.net. Los lenguajes de marca.* [En línea] 2010.  
[http://www.deciencias.net/disenoweb/elaborardw/paginas/intro\\_html.htm](http://www.deciencias.net/disenoweb/elaborardw/paginas/intro_html.htm).
- Westfall, L.L. 1995.** *"Software metrics that meet your information needs"* . 1995.
- Zend. 2013.** The PHP Company. [En línea] 2013. <http://www.zend.com/en/products/studio/>.

## BIBLIOGRAFÍA

- Pressman, Roger S. 1998.** *Ingeniería del software. Un enfoque práctico.* 1998.
- Bahit, Eugenia. 2010.** eByte. *Introducción al patrón arquitectónico MVC.* [En línea] 2010. <http://www.eugeniabahit.com/mvc/>.
- Barriocal, Luis. 2009.** [En línea] 2009. <http://www.edujoomla.es/que-es-joomla..>
- Briand, L., Daly, J. y Differding , C. 1996.** "An experimental comparison of the maintainability of object-oriented and structured design documents". 1996.
- Cabrera, Lianet y Pompa, Enrique R. 2012.** *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información.* 2012.
- Canós, José H., Letelier , Patricio y Penadé, M<sup>a</sup> Carmen. 2008.** *Métodologías Ágiles en el Desarrollo de Software.* [En línea] 2008. <http://sunshine.prod.uci.cu/gridfs/sunshine/books/TodoAgil.pdf>.
- Corletti, Alejandro y Muñoz, Carmen de Alba. 2008.** *Métricas de Seguridad, Indicadores y Cuadro de Mando.* 2008.
- Ecured. 2013.** [En línea] 2013. [http://www.ecured.cu/index.php/Herramienta\\_CASE](http://www.ecured.cu/index.php/Herramienta_CASE) .
- . **2013.** [En línea] 2013. [http://www.ecured.cu/index.php/Altova\\_Umodel](http://www.ecured.cu/index.php/Altova_Umodel).
- . **2013.** [En línea] 2013. [http://www.ecured.cu/index.php/PHP#Caracter.C3.ADsticas\\_de\\_PHP](http://www.ecured.cu/index.php/PHP#Caracter.C3.ADsticas_de_PHP).
- . **2012.** [En línea] 2012. [http://www.ecured.cu/index.php/Lenguaje\\_de\\_Programaci%C3%B3n\\_Web](http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n_Web).
- . **2013.** [En línea] 2013. [http://www.ecured.cu/index.php/Sistema\\_Gestor\\_de\\_Base\\_de\\_Datos](http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos).
- . **2010.** [En línea] 2010. [http://www.ecured.cu/index.php/Flujo\\_de\\_pruebas\\_de\\_un\\_software..](http://www.ecured.cu/index.php/Flujo_de_pruebas_de_un_software..)
- . **2013.** [En línea] 2013. <http://www.ecured.cu/index.php/CMS>.
- Espinoza, Humberto . 2008.** *PostgreSQL. Una Alternativa de DBMS Open Source.* 2008.
- Fernández, Mellado y Rodríguez, Daniel. 2010.** *TECNIMAP, EVALUACION DE LA CALIDAD Y SEGURIDAD EN PRODUCTOS SOFTWARE.* 2010.
- Gamma, Eric. 1995.** *Design Patterns: Elements of Reusable Object Oriented Software.* 1995.
- Gil Rodríguez, Fran. 2011.** *Experto en Drupal 7 Nivel Avanzado.* 2011.
- López, Ana María. 2008.** *INTRODUCCIÓN A LA CALIDAD DE SOFTWARE.* [En línea] 2008. <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/3241>.

**Martín, Enrique y Sáenz, Fernando . 2000.** *Sistema ANACONDA para el análisis automático de la calidad del software.* 2000.

**Mateu, Carles. 2004.** *Desarrollo de aplicaciones web.*

[http://sunshine.prod.uci.cu/gridfs/sunshine/books/Desarrollo\\_de\\_Aplicaciones\\_Web.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Desarrollo_de_Aplicaciones_Web.pdf) : s.n., 2004.

**Mendoza Sánchez, María A. 2004.** Informatizate.net. [En línea] 2004.

[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).

**Naramone, Elizabeth y Gerner, Jason. 2005.** *Beginning PHP5, Apache, and MySQL. Web Development.* 2005.

**Pentaho. 2009.** Kettle Pentaho Data Integration. [En línea] 2009. <http://kettle.pentaho.org/>.

**Peña Romero, Yanisleydis y Fonseca Martínez, Jorge. 2012.** *Sistema de evaluación de seguridad utilizando el proceso de integración de datos ETL.* 2012.

**Ramírez , Lama y Juan, José. 2011.** *"Requisitos y Evaluación de Calidad de Productos de Software"*. 2011.

**Rubalcaba Betancourt, Maria de los Angeles y Zambrana Hernández, Yayneris. 2008.** *Medición de la calidad de Software durante el Proceso de Pruebas en el Proyecto Modernización del CICPC.* 2008.

**Ruiz Tenorio, Roberto. 2010.** *Las pruebas de software y su importancia en las organizaciones.* 2010.

**Sánchez Asenjo, Jorge. 2009.** Apuntes Completos sobre Sistemas gestores de Bases de Datos. [En línea] 2009. <http://ubuntuone.com/p/sqt>.

**Santana Mancilla, Pedro César . 2001.** *Taller de PHP.* 2001.

**Talend. 2012.** [En línea] 2012. <http://es.talend.com/index.php>.

**Zend. 2013.** The PHP Company. [En línea] 2013. <http://www.zend.com/en/products/studio/>.

## GLOSARIO DE TÉRMINOS

### A

**Apache:** es un servidor web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales.

### B

**Base de Datos (BD):** conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente. En una BD, la información se organiza en campos y registros.

### C

**CASE (*Computer Aided Software Engineering*):** sus siglas en español significan Ingeniería de Software Asistida por Ordenador y responde al conjunto de aplicaciones informáticas destinadas a ayudar en todos los aspectos del ciclo de vida de desarrollo del software.

**CISED (*Centro de Identificación y Seguridad Digital*):** centro de desarrollo de software, perteneciente a la Facultad 1, adjunto a la Universidad de las Ciencias Informáticas.

### E

**ETL (*Extract, Transform and Load*):** es el proceso de Extracción, Transformación y Carga, este es un proceso que permite mover datos desde múltiples fuentes, reformatearlos, limpiarlos, y cargarlos en una BD o en otro sistema operacional para analizarlos y obtener de ellos solo lo necesario para trabajar.

### F

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

### G

**GPL (*General Public License*):** es la Licencia Pública General de GNU, es una licencia creada por la *Free Software Foundation* en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta

licencia es libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

### H

**HTML (*Hypertext Transfer Protocol*):** es un lenguaje de marcado que tiene como objetivo estructurar documentos y mostrarlos en forma de hipertexto

**HTTP (*Hypertext Transfer Protocol*):** es el protocolo usado en cada transacción de la *World Wide Web*. Este sigue el esquema petición-respuesta entre un cliente y un servidor.

### I

**Indicador:** es un valor que se obtiene comparando datos lógicamente relacionados, referentes al comportamiento de una actividad, proceso o control, dentro de un tiempo específico.

### L

**LDAP (*Lightweight Directory Access Protocol*):** el Protocolo Ligero de Acceso a Directorios 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.

### M

**MSF (*Microsoft Solution Framework*):** *framework* que brinda *Microsoft* para guiar el desarrollo de sistemas. Es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos.

**Multiplataforma:** es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas. Una plataforma es una combinación de hardware y software usada para ejecutar aplicaciones.

### U

**UML (*Unified Modeling Language*):** Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software que se utiliza para especificar, visualizar, modificar, construir y documentar los artefactos que se obtienen durante el desarrollo.

**Usuario:** persona que tiene una cuenta en una determinada computadora por medio de la cual puede acceder a los recursos y servicios que ofrece una red.

### W

**W3AF (*Web Application Attack and Audit Framework*):** es un *framework* libre, que permite realizar diferentes tipos de pruebas de seguridad a aplicaciones web para determinar sus vulnerabilidades.

### X

**XML (*eXtensible Markup Language*):** Lenguaje de Marcado Extensible, es un metalenguaje extensible de etiquetas.



## Anexo 1: Descripción de los escenarios

Tabla 19: Descripción del escenario "Autenticar usuario"

<b>Nombre del escenario:</b> Autenticar usuario		<b>Identificador:</b> EC 1
<b>Objetivo del escenario:</b> Verificar que el usuario tenga acceso al sistema		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b>		
<b>Descripción:</b> El evaluador introduce su usuario y contraseña, el sistema verifica la autenticidad de los datos, si los datos son correctos el evaluador accede al sistema		
<b>Validaciones:</b> Verificar que no existan campos vacío		
<b>Prototipo de interfaz de usuario:</b>		
		
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>	

Tabla 20: Descripción del escenario "Gestionar métrica"

<b>Nombre del escenario:</b> Gestionar métrica		<b>Identificador:</b> EC 2
<b>Objetivo del escenario:</b> Gestionar los datos correspondientes a una métrica		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> En caso de modificar o eliminar debe existir al menos una métrica		

<b>Descripción:</b> El evaluador puede insertar, modificar o eliminar los datos de las métricas	
<b>Validaciones:</b>	
<b>Prototipo de interfaz de usuario:</b>	
	
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>

Tabla 21: Descripción del escenario “Gestionar datos del proyecto”

<b>Nombre del escenario:</b> Gestionar datos del proyecto		<b>Identificador:</b> EC 3
<b>Objetivo del escenario:</b> Brindar al usuario las opciones de insertar, modificar o eliminar algún proyecto		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Para eliminar o modificar debe existir al menos un proyecto evaluado		
<b>Descripción:</b> El escenario comienza cuando el usuario selecciona alguna de las siguientes opciones : Insertar proyecto: el usuario inserta los datos del nuevo proyecto y el sistema adiciona el proyecto a la lista. Modificar proyecto: el usuario modifica los datos del proyecto y el sistema actualiza la lista de proyectos con los nuevos valores. Eliminar proyecto: el usuario elimina el proyecto, automáticamente el sistema lo elimina de la lista		
<b>Validación:</b>		
<ul style="list-style-type: none"> <li>✓ No pueden existir campos vacíos.</li> <li>✓ El campo Nombre del Responsable solo permite letras.</li> <li>✓ El campo Cantidad de Desarrolladores solo permite números.</li> </ul>		

✓ No deben repetirse proyectos.	
<b>Prototipo de interfaz de usuario:</b>	
	
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>

Tabla 22: Descripción del escenario “Seleccionar métrica”

<b>Nombre del escenario:</b> Seleccionar métrica		<b>Identificador:</b> EC 4
<b>Objetivo del escenario:</b> Seleccionar las métricas con las que desea evaluar el software		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Debe haber escogido un proyecto para evaluar		
<b>Descripción:</b> El evaluador selecciona las métricas que desea para la evaluación del software		
<b>Validaciones:</b>		
<b>Prototipo de interfaz de usuario:</b>		

Selección de métricas	
<p>▼ Listado de métricas a seleccionar</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Controlabilidad de acceso</li> <li><input type="checkbox"/> Detección de vulnerabilidades</li> <li><input type="checkbox"/> Identificación de riesgos</li> <li><input checked="" type="checkbox"/> Evitación de fallos</li> <li><input type="checkbox"/> Adecuación de pruebas</li> <li><input type="checkbox"/> Prevención de corrupción de datos</li> </ul>	
<p>▼ Detalles de la métrica <b>Evitación de fallos</b></p> <p>Función de evaluación de la métrica: A/B</p> <p><b>Valor A</b></p> <input type="text"/> <p><b>Valor B</b></p> <input type="text"/> <p>Aceptar</p>	
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>

Tabla 23: Descripción del escenario “Cargar fichero”

<b>Nombre del escenario:</b> Cargar fichero		<b>Identificador:</b> EC 5
<b>Objetivo del escenario:</b> Cargar en la BD el fichero que almacena los resultados que arrojaron las pruebas de seguridad		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Se debe haber seleccionado un proyecto para evaluar		
<b>Descripción:</b> El evaluador selecciona la opción enviar fichero de pruebas, el sistema muestra el formulario donde se encuentran los elementos necesarios para que sea cargado el fichero en la BD		
<b>Validaciones:</b> Verificar que el campo no esté vacío y que el adjunto sea el fichero txt		
<b>Prototipo de interfaz de usuario:</b>		

<p><b>Cargar fichero de pruebas</b></p> <p>♥ Enviar el archivo con los resultados de las pruebas de seguridad</p> <p>Aquí debe cargar un fichero generado por la herramienta de pruebas W3AF, de no ser así el sistema le mostrará un error.</p> <p><b>Ruta del Archivo</b></p> <input type="text"/> <input type="button" value="Examinar..."/> <input type="button" value="Enviar"/>	
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>

**Tabla 24: Descripción del escenario “Procesar datos del fichero”**

<b>Nombre del escenario:</b> Procesar datos del fichero		<b>Identificador:</b> EC 6
<b>Objetivo del escenario:</b> Realizar el proceso de ETL a los datos del fichero cargado		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Se debe haber cargado el fichero con los datos de las pruebas de seguridad		
<b>Descripción:</b> El evaluador selecciona la opción procesar datos de las pruebas y a partir de ahí comienza el proceso ETL		
<b>Validaciones:</b> Verificar que exista el fichero en el directorio del servidor		
<b>Prototipo de interfaz de usuario:</b>		
<p><b>Procesar datos de las pruebas</b></p> <p>Si seleccionó algún fichero de pruebas de la herramienta W3AF este debe ser válido en caso contrario no se mostrarán los resultados para las métricas relacionadas con él.</p> <input type="button" value="Procesar datos"/>		
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>	

Tabla 25: Descripción del escenario “Calcular métrica”

<b>Nombre del escenario:</b> Calcular métrica		<b>Identificador:</b> EC 7
<b>Objetivo del escenario:</b> Calcular las fórmulas de las métricas		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Debe haber seleccionado al menos una métrica		
<b>Descripción:</b> El sistema automáticamente calcula los valores de las métricas en correspondencia con su fórmula		
<b>Validaciones:</b>		
<b>Prototipo de interfaz de usuario:</b> N/P		
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>	

Tabla 26: Descripción del escenario “Consultar historial de evaluaciones”

<b>Nombre del escenario:</b> Consultar el historial de evaluaciones		<b>Identificador:</b> EC 9
<b>Objetivo del escenario:</b> Mostrar un historial de todos los proyectos evaluados		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 2da	<b>Prioridad:</b> 3	<b>Complejidad:</b> 3
<b>Precondiciones:</b> Debe existir al menos un proyecto evaluado		
<b>Descripción:</b> El escenario comienza cuando el usuario selecciona la opción “Historial de evaluaciones” y el sistema muestra un listado con todos los proyectos, la fecha de su evaluación y el nivel general de seguridad así como los resultados de cada una de las métricas aplicadas.		
<b>Validaciones:</b> Debe existir al menos un proyecto evaluado		
<b>Prototipo de interfaz de usuario:</b>		

Historial de evaluaciones				
<input type="text"/> Operaciones para el historial				
Nombre	Responsable	Fecha	Nivel de Seguridad	Detalles
SegMat	Antonio	19-05-2013 09:44:42	Alto	>>>
SegMat	Antonio	19-05-2013 09:43:36	Alto	>>>
SegMat	Antonio	19-05-2013 09:42:12	Alto	>>>
SegMat	Antonio	19-05-2013 04:50:55	Alto	>>>
OpenGFAR	Ramon	17-05-2013 15:45:44	Alto	>>>
OpenGFAR	Ramon	17-05-2013 15:31:41	Alto	>>>
IntranetPortales	Jorge	17-05-2013 15:29:24	Medio	>>>
IntranetPortales	Jorge	17-05-2013 15:26:51	Medio	>>>
IdentCENIA	Luisa	16-05-2013 09:44:34	Alto	>>>
IdentCENIA	Luisa	15-05-2013 11:42:49	Bajo	>>>
OpenGFAR	Ramon	03-05-2013 23:41:58	Muy Alto	>>>

<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>
-------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Tabla 27: Descripción del escenario “Notificar resultado de la evaluación”**

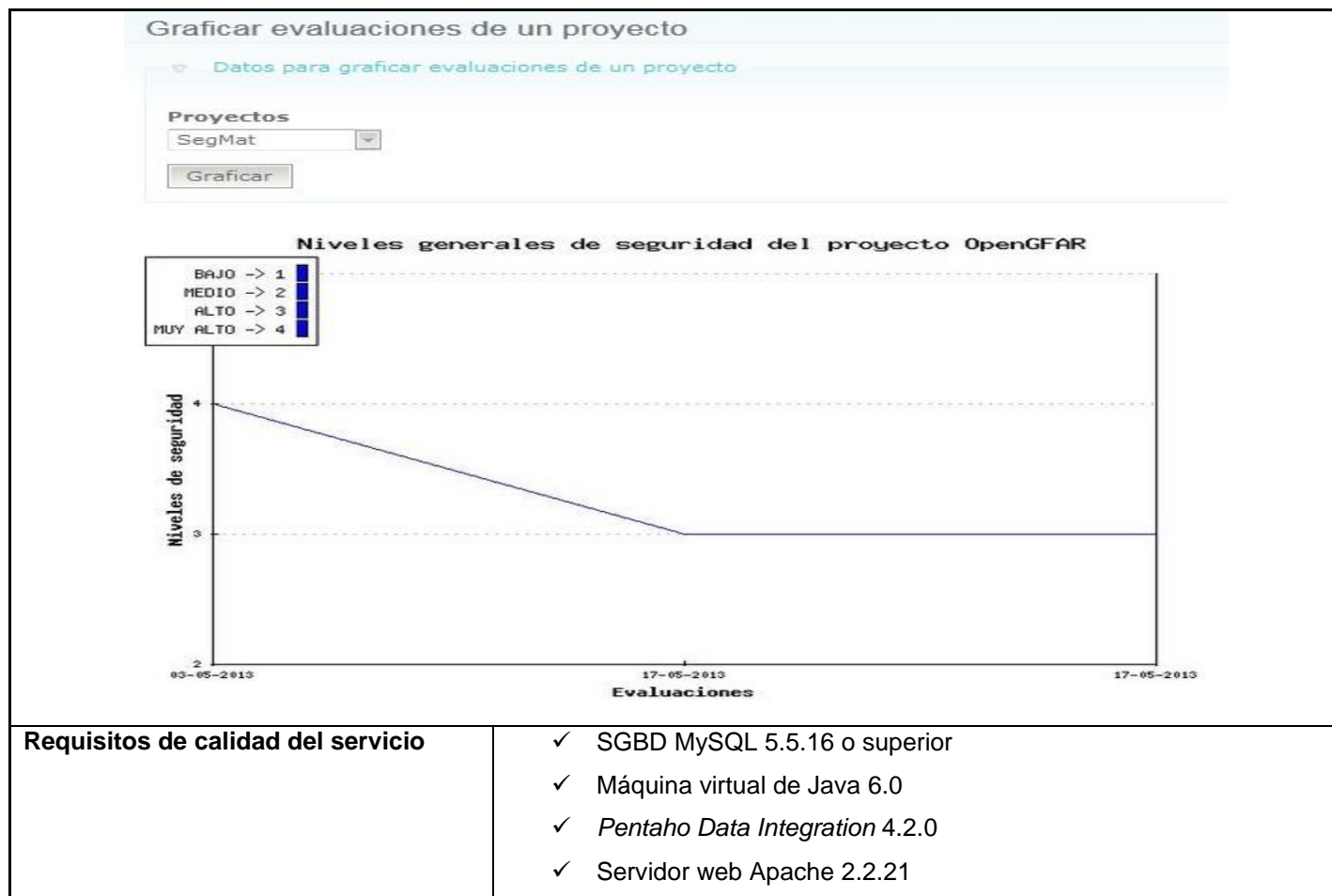
<b>Nombre del escenario:</b> Notificar resultado de la evaluación		<b>Identificador:</b> EC 10
<b>Objetivo del escenario:</b> Notificar a la alta gerencia la evaluación o nivel de seguridad de la aplicación evaluada		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 2da	<b>Prioridad:</b> 4	<b>Complejidad:</b> 4
<b>Precondiciones:</b> Debe existir el informe de evaluación		
<b>Descripción:</b> El escenario comienza cuando el usuario selecciona la opción “Notificación de reportes”, entonces el sistema permite entrar la dirección de correo del interesado en conocer el informe, el usuario oprime el botón “Enviar reporte” y el sistema envía la notificación vía correo electrónico.		
<b>Validaciones:</b> La dirección del correo y la contraseña del que envía el reporte deben ser válidas, al igual que el correo destino		
<b>Prototipo de interfaz de usuario:</b>		

Notificación de reportes	
<p>☐ Datos para notificar el resultado</p> <p><b>Correo UCI</b></p> <input type="text"/>	
<p><b>Contraseña UCI</b></p> <input type="text"/>	
<p><b>Correo del destinatario</b></p> <input type="text"/>	
<p>Ej. ramonperezg@uci.cu joseht@estudiantes.uci.cu</p>	
<p><b>Ruta del Adjunto</b></p> <input type="text"/> <input type="button" value="Examinar..."/>	
<input type="button" value="Enviar reporte"/>	
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>

Tabla 28: Descripción del escenario “Graficar evaluaciones de un proyecto”

<b>Nombre del escenario:</b> Graficar evaluaciones de un proyecto		<b>Identificador:</b> EC 11
<b>Objetivo del escenario:</b> Mostrar una gráfica con los resultados obtenidos		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 2da	<b>Prioridad:</b> 4	<b>Complejidad:</b> 4
<b>Precondiciones:</b> Debe existir al menos un proyecto evaluado.		
<b>Descripción:</b> El escenario comienza cuando el usuario se dirige a la opción “Graficar evaluaciones de un proyecto”, luego el sistema muestra una gráfica con los resultados		
<b>Validaciones:</b>		
<b>Prototipo de interfaz de usuario:</b>		





## Anexo 2: Descripción de tareas por escenarios

### ➤ Especificación de tareas del escenario: “Gestionar métrica”.

El escenario se dividió en 3 tareas fundamentales: Crear métrica, Modificar métrica y Eliminar métrica.

Tabla 29: Descripción de la tarea “Crear métrica”

<b>Nombre del escenario:</b> Crear métrica		<b>Identificador:</b> T 2.1
<b>Objetivo del escenario:</b> Introducir los datos para crear una nueva métrica		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b>		
<b>Descripción:</b> El evaluador inserta los datos para crear la nueva métrica		
<b>Validaciones:</b>		
<ul style="list-style-type: none"> <li>✓ No deben existir campos vacíos</li> </ul>		

✓ La métrica no debe estar creada	
<b>Prototipo de interfaz de usuario:</b>	
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>

Tabla 30: Descripción de la tarea “Modificar métrica”

<b>Nombre del escenario:</b> Modificar métrica		<b>Identificador:</b> T 2.2
<b>Objetivo del escenario:</b> Modificar los datos de una métrica		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Debe existir al menos una métrica		
<b>Descripción:</b> El evaluador selecciona una métrica para modificar sus datos		
<b>Validaciones:</b> No deben existir campos vacíos		
<b>Prototipo de interfaz de usuario:</b>		

▽ Editar datos de métrica

**Nombre de la métrica**

**Fórmula de la métrica**  
  
Las variables deben escribirse en mayúsculas  
Ej. (A-B)/1-C

▽ Niveles para medir el valor de la métrica  
Cada nivel comprende un rango.Ej 0.0-0.5

<b>Bajo</b>	<b>Medio</b>	<b>Alto</b>	<b>Muy Alto</b>
<input type="text" value="0.0-0.2"/>	<input type="text" value="0.2-0.5"/>	<input type="text" value="0.5-0.8"/>	<input type="text" value="0.8-1.0"/>

<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>
-------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 31: Descripción de la tarea “Eliminar métrica”

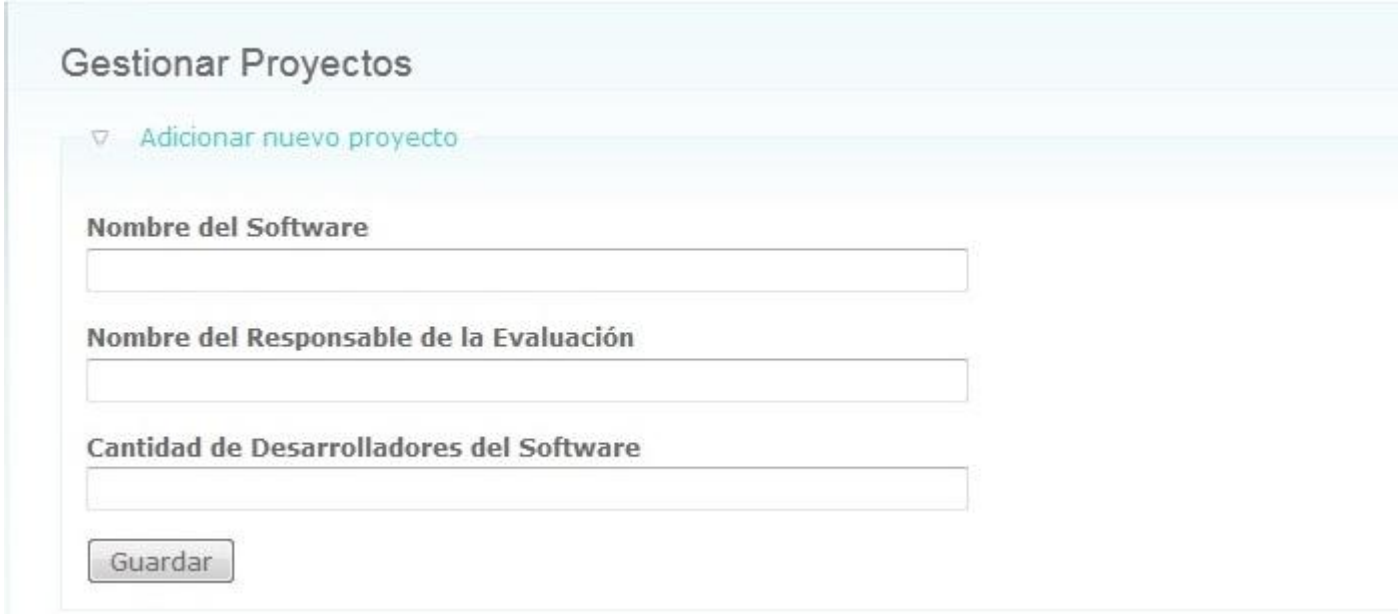
<b>Nombre del escenario:</b> Eliminar métrica			<b>Identificador:</b> T 2.3			
<b>Objetivo del escenario:</b> Eliminar una métrica						
<b>Persona:</b> Evaluador						
<b>Iteración:</b> 1ra		<b>Prioridad:</b> 5		<b>Complejidad:</b> 5		
<b>Precondiciones:</b> Debe existir al menos una métrica para ser eliminada						
<b>Descripción:</b> El evaluador selecciona la métrica que desea eliminar						
<b>Validaciones:</b>						
<b>Prototipo de interfaz de usuario:</b>						
Nombre	Fórmula	Bajo	Medio	Alto	Muy Alto	Opciones
metrica1	A-(B/C)	0.0-0.2	0.2-0.5	0.5-0.8	0.8-1.0	Editar Eliminar
metrica2	A-(B/C)-1	0.0-0.3	0.3-0.4	0.4-0.7	0.7-1.0	Editar Eliminar
metrica3	A-(B/C)	0.0-0.3	0.3-0.4	0.4-0.5	0.5-1.0	Editar Eliminar
<b>Requisitos de calidad del servicio</b>		✓ SGBD MySQL 5.5.16 o superior				

	<ul style="list-style-type: none"> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

➤ **Especificación de tareas del escenario: “Gestionar los datos del proyecto”.**

El escenario se dividió en 3 tareas fundamentales: Insertar, Modificar y Eliminar el proyecto.

**Tabla 32: Descripción de la tarea “Crear datos del proyecto”**

<b>Nombre del escenario:</b> Crear datos del proyecto		<b>Identificador:</b> T 3.1
<b>Objetivo del escenario:</b> Insertar los datos de un nuevo proyecto		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b>		
<b>Descripción:</b> El usuario inserta los datos del nuevo proyecto y el sistema adiciona el proyecto a la lista		
<b>Validaciones:</b>		
<ul style="list-style-type: none"> <li>✓ No pueden existir campos vacíos.</li> <li>✓ El campo Nombre del Responsable solo permite letras.</li> <li>✓ El campo Cantidad de Desarrolladores solo permite números.</li> <li>✓ No deben repetirse proyectos.</li> </ul>		
<b>Prototipo de interfaz de usuario:</b>		
		
<b>Requisitos de calidad del servicio</b>	✓ SGBD MySQL 5.5.16 o superior	

	<ul style="list-style-type: none"> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration 4.2.0</i></li> <li>✓ Servidor web Apache 2.2.21</li> </ul>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 33: Descripción de la tarea “Modificar proyecto”

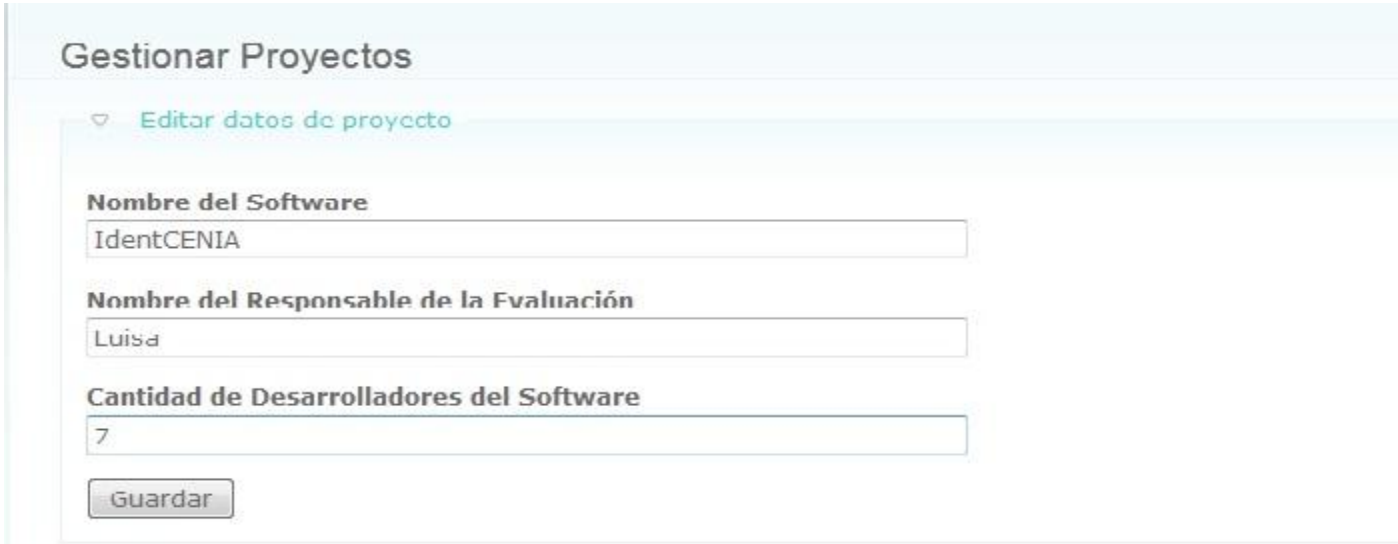
<b>Nombre del escenario:</b> Modificar proyecto		<b>Identificador:</b> T 3.2
<b>Objetivo del escenario:</b> Realizarle cambios a la información de un proyecto que fue insertado en el sistema con anterioridad		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Debe existir al menos un proyecto insertado en el sistema		
<b>Descripción:</b> El escenario comienza cuando el usuario selecciona la opción Editar, el sistema procede a mostrar los campos que pueden ser modificados y una vez que el usuario oprima el botón Guardar, el sistema guarda los cambios realizados en la base de datos		
<b>Validaciones:</b> Verificar que no existan campos vacíos		
<b>Prototipo de interfaz de usuario:</b>		
		
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration 4.2.0</i></li> <li>✓ Servidor web Apache 2.2.21</li> </ul>	

Tabla 34: Descripción de la tarea “Eliminar proyecto”

<b>Nombre del escenario:</b> Eliminar proyecto		<b>Identificador:</b> T 3.3	
<b>Objetivo del escenario:</b> Eliminar todos los datos de un proyecto			
<b>Persona:</b> Evaluador			
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5	
<b>Precondiciones:</b> Debe existir al menos un proyecto insertado en el sistema			
<b>Descripción:</b> El escenario comienza cuando el usuario selecciona la opción Eliminar y el sistema elimina de la base de datos toda la información referente al proyecto			
<b>Validaciones:</b>			
<b>Prototipo de interfaz de usuario:</b>			
<b>Nombre</b>	<b>Responsable</b>	<b>Cantidad de Desarrolladores</b>	<b>Opciones</b>
IdentCENIA	Luisa	7	Editar <b>Eliminar</b> Evaluar
Intranet/Portales	Jorge	6	Editar Eliminar Evaluar
OpenG-FAR	Ramon	12	Editar Eliminar Evaluar
SegMat	Antonio	3	Editar Eliminar Evaluar
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>		

➤ **Especificación de tareas del escenario: “Cargar fichero”.**

El escenario se dividió en 2 tareas fundamentales, que proveerán las funcionalidades de Seleccionar fichero y Subir fichero.

Tabla 35: Descripción de la tarea “Seleccionar fichero”

<b>Nombre del escenario:</b> Seleccionar fichero		<b>Identificador:</b> T 5.1	
<b>Objetivo del escenario:</b> Seleccionar el fichero de pruebas			
<b>Persona:</b> Evaluador			
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5	
<b>Precondiciones:</b> Se debe haber escogido un proyecto para evaluar			
<b>Descripción:</b> El evaluador selecciona el fichero que contiene las pruebas arrojadas por la herramienta W3AF			
<b>Validaciones:</b>			
<b>Prototipo de interfaz de usuario:</b>			

Cargar fichero de pruebas	
<p>♥ Enviar el archivo con los resultados de las pruebas de seguridad</p> <p>Aquí debe cargar un fichero generado por la herramienta de pruebas W3AF, de no ser así el sistema le mostrará un error.</p> <p><b>Ruta del Archivo</b></p> <input type="text"/> <input type="button" value="Examinar..."/> <input type="button" value="Enviar"/>	
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>

Tabla 36: Descripción de la tarea “Subir fichero”

<b>Nombre del escenario:</b> Subir fichero		<b>Identificador:</b> T 5.2
<b>Objetivo del escenario:</b> Subir el fichero para el servidor		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Se debe haber escogido la dirección del fichero		
<b>Descripción:</b> Una vez escogida la dirección del fichero el evaluador oprime el botón Enviar y el fichero es guardado en una carpeta del servidor		
<b>Validaciones:</b>		
<b>Prototipo de interfaz de usuario:</b>		
<p>Cargar fichero de pruebas</p> <p>♥ Enviar el archivo con los resultados de las pruebas de seguridad</p> <p>Aquí debe cargar un fichero generado por la herramienta de pruebas W3AF, de no ser así el sistema le mostrará un error.</p> <p><b>Ruta del Archivo</b></p> <input type="text" value="D:\DOCENCIA\Tesis\w3af.txt"/> <input type="button" value="Examinar..."/> <input type="button" value="Enviar"/>		
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> </ul>	

	<ul style="list-style-type: none"> <li>✓ <i>Pentaho Data Integration 4.2.0</i></li> <li>✓ Servidor web Apache 2.2.21</li> </ul>
--	---------------------------------------------------------------------------------------------------------------------------------

➤ **Especificación de tareas del escenario: “Procesar datos del fichero”.**

El escenario se dividió en 3 tareas fundamentales: Extraer datos del fichero, Transformar datos del fichero y Cargar datos del fichero.

**Tabla 37: Descripción de la tarea “Extraer datos del fichero”**

<b>Nombre del escenario:</b> Extraer datos del fichero		<b>Identificador:</b> T 6.1
<b>Objetivo del escenario:</b> Obtener los datos que contiene el fichero		
<b>Persona:</b>		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Se debe haber subido el fichero de pruebas		
<b>Descripción:</b> Luego de cargar el fichero con los datos en la BD, se procede a la extracción de los mismos. El sistema extrae los datos necesarios para realizar posteriormente las transformaciones		
<b>Validaciones:</b>		
<b>Prototipo de interfaz de usuario:</b> N/P		
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration 4.2.0</i></li> <li>✓ Servidor web Apache 2.2.21</li> </ul>	

**Tabla 38: Descripción de la tarea “Transformar datos del fichero”**

<b>Nombre del escenario:</b> Transformar datos del fichero		<b>Identificador:</b> T 6.2
<b>Objetivo del escenario:</b> Transformar los datos de la extracción		
<b>Persona:</b>		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Se deben haber extraído los datos necesarios		
<b>Descripción:</b> Después de haber extraído los datos, se procede a la transformación de los mismos. El sistema inicia el proceso de transformación, el cual selecciona únicamente los datos necesarios para realizar el proceso de evaluación		
<b>Validaciones:</b>		
<b>Prototipo de interfaz de usuario:</b> N/P		
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> </ul>	



	<ul style="list-style-type: none"> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 39: Descripción de la tarea “Cargar datos del fichero”

<b>Nombre del escenario:</b> Cargar datos del fichero		<b>Identificador:</b> T 6.3
<b>Objetivo del escenario:</b> Cargar los datos de la transformación		
<b>Persona:</b>		
<b>Iteración:</b> 1ra	<b>Prioridad:</b> 5	<b>Complejidad:</b> 5
<b>Precondiciones:</b> Se deben haber transformado los datos necesarios		
<b>Descripción:</b> Una vez transformados los datos, se procede a cargar la información nuevamente a la BD. El sistema envía los datos a la tabla correspondiente		
<b>Validaciones:</b>		
<b>Prototipo de interfaz de usuario:</b> N/P		
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>	

➤ **Especificación de tareas del escenario: “Notificar resultados de la evaluación”.**

Este escenario solo cuenta con una tarea: Introducir datos de contacto.

Tabla 40: Descripción de la tarea “Introducir datos de contacto”

<b>Nombre del escenario:</b> Introducir datos de contacto		<b>Identificador:</b> T 10.1
<b>Objetivo del escenario:</b> Permitir introducir datos del contacto a notificar		
<b>Persona:</b> Evaluador		
<b>Iteración:</b> 2da	<b>Prioridad:</b> 4	<b>Complejidad:</b> 4
<b>Precondiciones:</b> Debe haberse evaluado un proyecto		
<b>Descripción:</b> El evaluador introduce los datos del contacto al que se debe notificar los resultados de la evaluación		
<b>Validaciones:</b> Verificar que no quede ningún campo del informe vacío		
<b>Prototipo de interfaz de usuario:</b>		

Notificación de reportes	
<p>○ Datos para notificar el resultado</p> <p><b>Correo UCI</b></p> <input type="text"/> <p><b>Contraseña UCI</b></p> <input type="text"/> <p><b>Correo del destinatario</b></p> <input type="text"/> Ej. ramonperezg@uci.cu joseht@estudiantes.uci.cu <p><b>Ruta del Adjunto</b></p> <input type="text"/> <input type="button" value="Examinar..."/> <p><input type="button" value="Enviar reporte"/></p>	
<b>Requisitos de calidad del servicio</b>	<ul style="list-style-type: none"> <li>✓ SGBD MySQL 5.5.16 o superior</li> <li>✓ Máquina virtual de Java 6.0</li> <li>✓ <i>Pentaho Data Integration</i> 4.2.0</li> <li>✓ Servidor web Apache 2.2.21</li> </ul>

### Anexo 3: Descripción de las interfaces

- **Interfaz “Enviar fichero de prueba”:** permite enviar el fichero que contiene el resultado de la pruebas de seguridad.

**Sistema Evaluación de Seguridad**

Inicio

Mi cuenta Cerrar sesión

**Evaluar Seguridad**

- Gestionar métricas
- Gestionar proyectos
- Selección de métricas
- Resultado de la evaluación
- Notificación de reportes
- Historial de evaluaciones
- Graficar evaluaciones de un proyecto

**Quién está conectado**

Hay actualmente 1 usuario conectado.

- yaciel

**Cargar fichero de pruebas**

Enviar el archivo con los resultados de las pruebas de seguridad

Aquí debe cargar un fichero generado por la herramienta de pruebas W3AF, de no ser así el sistema le mostrará un error.

**Ruta del Archivo**

Examinar...

Enviar

Figura 13: Interfaz “Enviar fichero de prueba”

- **Interfaz “Resultados de la evaluación”:** muestra los resultados luego de ser evaluado el software.

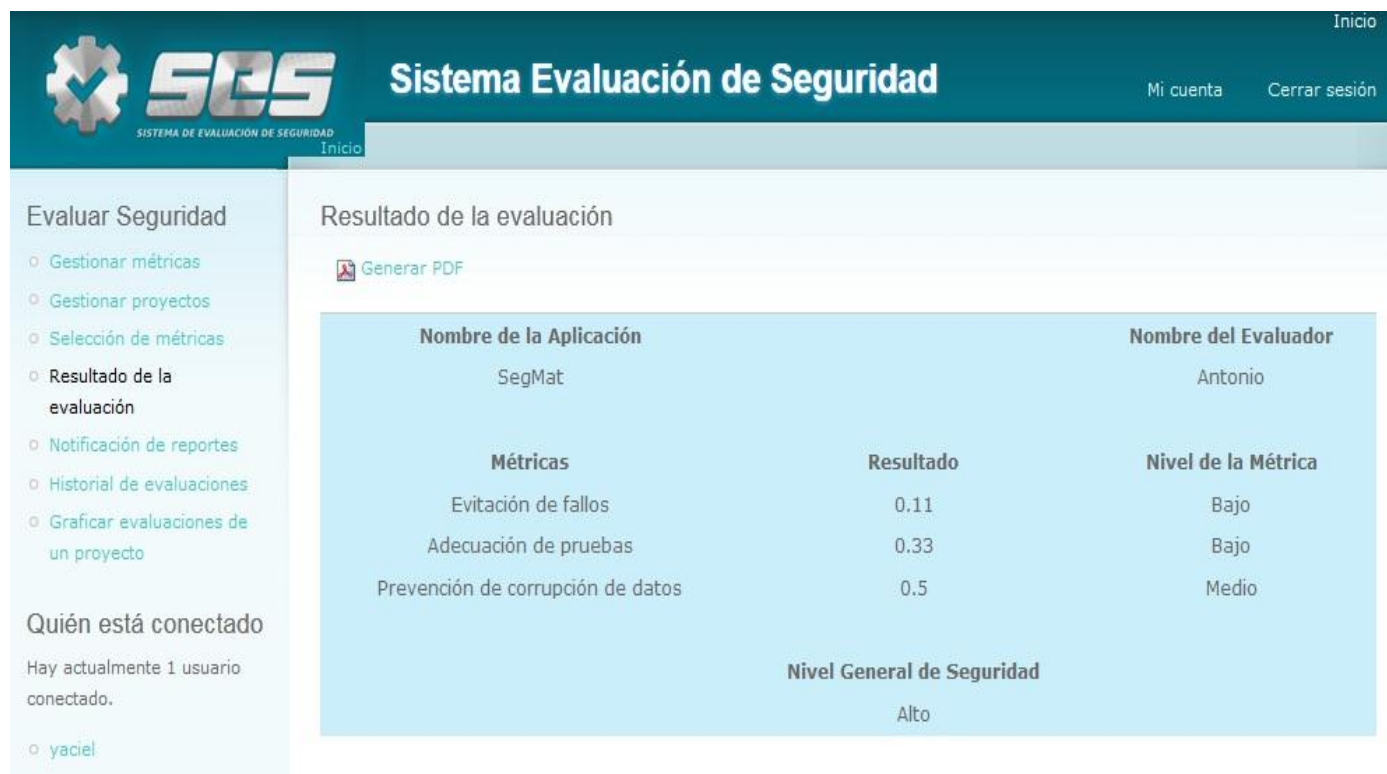


Figura 14: Interfaz “Resultados de la evaluación”

## Anexo 4: Descripción de las pruebas de caja negra

### ➤ Escenario “Autenticar usuario”

Tabla 41: Descripción del caso de prueba “Autenticar usuario”

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC1: Autenticar usuario	El evaluador debe autenticarse para poder acceder al sistema y realizar la evaluación	Nombre de Usuario y Contraseña (Válidos)	El sistema le permite al usuario acceder a las opciones del sistema	Inicio de sesión

		Nombre de Usuario y Contraseña (Inválidos)	El sistema muestra un mensaje de error informando que existen datos incorrectos	
--	--	--------------------------------------------	---------------------------------------------------------------------------------	--

➤ Escenario “Gestionar métrica”

Tabla 42: Descripción del caso de prueba “Gestionar métrica”

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC2: Gestionar métricas	El escenario inicia cuando el usuario selecciona la opción de crear, modificar o eliminar alguna métrica	Gestionar métricas (Válidos)	El sistema inicia la gestión de métricas en dependencia de la opción seleccionada	Opción Gestionar Métricas
		Gestionar métricas (Inválidos)	El sistema muestra un mensaje de error en caso de que exista algún campo vacío o con caracteres incorrectos	
EC2.1: Crear métrica	Se crea una nueva métrica	Nombre métrica Variable 1 Variable 2 Otro valor Niveles de evaluación	El sistema crea la nueva métrica	Opción Gestionar Métricas / Opción Crear

		(Válidos)		
		Nombre métrica Variable 1 Variable 2 Otro valor Niveles de evaluación (Inválidos)	El sistema muestra un mensaje de error informando que se entraron datos incorrectos, o campos vacíos que imposibilitaron la inserción.	
EC2.2: Modificar métrica	Se modifican los datos de determinada métrica	Nombre métrica Variable 1 Variable 2 Otro valor Niveles de evaluación (Válidos)	El sistema modifica los datos de la métrica en la BD	Opción Gestionar Métricas / Opción Editar
		Nombre métrica Variable 1 Variable 2 Otro valor Niveles de evaluación (Inválidos)	El sistema muestra un mensaje de error informando que se entraron datos incorrectos, o campos vacíos que imposibilitaron la inserción.	

EC2.3: Eliminar métrica	Se elimina determinada métrica existente.	Identificador de la métrica en la BD (N/A)	El sistema elimina de la BD la métrica seleccionada	Opción Gestionar Métricas / Opción Eliminar
-------------------------	-------------------------------------------	--------------------------------------------	-----------------------------------------------------	---------------------------------------------

➤ **Escenario “Gestionar los datos del proyecto”**

**Tabla 43: Descripción del caso de prueba “Gestionar los datos del proyecto”**

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC3: Gestionar los datos del proyecto	El escenario comienza cuando el usuario selecciona alguna de las siguientes opciones: insertar, modificar o eliminar.	Gestionar datos (Válidos)	El sistema inicia la gestión del proyecto en dependencia de la opción seleccionada.	Opción Gestionar Proyectos.
		Gestionar datos (Inválidos)	El sistema muestra un mensaje de error en caso de que exista algún campo vacío o con caracteres incorrectos.	
EC3.1: Insertar proyecto	Se procede a insertar un proyecto una vez seleccionada la opción insertar proyecto.	Nombre Software Cantidad de desarrolladores Nombre Evaluador Estado	El sistema inserta el proyecto en la BD.	Opción Gestionar Proyectos/ Opción Crear

		(Válidos)		
		Nombre Software Cantidad de desarrolladores Nombre Evaluador Estado (Inválidos)	El sistema muestra un mensaje de error informando que se entraron datos incorrectos, o campos vacíos que imposibilitaron la inserción.	
EC3.2: Modificar proyecto	Se modifican los datos del proyecto.	Nombre Software Cantidad de desarrolladores Nombre Evaluador Estado (Válidos)	El sistema modifica los datos del proyecto en la BD.	Opción Gestionar Proyecto / Opción Editar
		Nombre Software Cantidad de desarrolladores Nombre Evaluador Estado (Inválido)	El sistema muestra un mensaje de error informando que se entraron datos incorrectos, o campos vacíos que imposibilitaron la inserción.	



EC3.3: Eliminar proyecto	Se elimina un proyecto existente.	Identificador del proyecto en la BD (N/A)	El sistema elimina de la BD el proyecto.	Opción Gestionar Proyecto / Opción Eliminar
--------------------------	-----------------------------------	-------------------------------------------	------------------------------------------	---------------------------------------------

➤ Escenario “Seleccionar métrica”

Tabla 44: Descripción del caso de prueba “Seleccionar métrica”

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC4: Seleccionar métrica	El escenario inicia con la selección de las métricas que se desean utilizar para la evaluación del proyecto	Seleccionar métrica (Válido)	El sistema muestra las métricas seleccionadas por el evaluador	Opción Selección de métricas
		Seleccionar métrica (Inválido)	El sistema muestra un mensaje de error indicándole que debe seleccionar alguna métrica	

➤ Escenario “Procesar datos del fichero”

Tabla 45: Descripción del caso de prueba “Procesar datos del fichero”

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC6: Procesar datos del fichero	El evaluador selecciona la opción procesar datos de las pruebas y a partir	Procesar datos (Válido: equivale a que se cargó un fichero a la BD)	El sistema inicia el proceso ETL	Sección Selección de métricas/ Botón

	de ahí comienza el proceso ETL	Procesar datos (Inválido: equivale a que no se cargó un fichero a la BD)	El sistema informa que no existe fichero para procesar	Procesar datos
EC6.1: Extraer datos del fichero	Se procede extraer los datos del fichero que posee el resultado de las pruebas de seguridad	Procesar datos (Válido: equivale a que se cargó un fichero a la BD)	El sistema inicia el proceso ETL	Sección Selección de métricas/ Botón Procesar datos
		Procesar datos (Inválido: equivale a que no se cargó un fichero a la BD)	El sistema informa que no existe fichero para procesar	
EC6.2: Transformar datos del fichero	Se procede a seleccionar los datos necesarios para la evaluación	N/P	El sistema realiza esta sección una vez finalizada la anterior.	Se realiza de forma automática.
EC6.3: Cargar datos del fichero	Son almacenados en la BD solo los resultados de la transformación	N/P	El sistema realizar esta sección una vez finalizada la anterior	Se realiza de forma automática.

➤ **Escenario “Calcular métrica”**

**Tabla 46: Descripción del caso de prueba “Calcular métrica”**

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
-----------	-------------	-------	-----------------------	---------------

EC7: Calcular métrica	El escenario inicia cuando el usuario selecciona la opción de calcular las métricas para ser evaluado su proyecto	(N/P)	Devuelve los resultados de los cálculos aplicados a las métricas de seguridad utilizadas para la evaluación	Se realiza de forma automática
-----------------------	-------------------------------------------------------------------------------------------------------------------	-------	-------------------------------------------------------------------------------------------------------------	--------------------------------

➤ **Escenario “Gestionar reporte de evaluación”**

**Tabla 47: Descripción del caso de prueba “Gestionar reporte de evaluación”**

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC8: Gestionar reportes de las evaluaciones	Una vez que el sistema muestra el reporte el evaluador podrá exportarlo y el sistema le permitirá escoger la dirección física donde desea guardarlo o abrirlo directamente.	Gestionar datos (Válidos)	El sistema inicia la gestión del reporte de la evaluación	Opción Resultado de la evaluación
		Gestionar datos (Inválidos)	El sistema muestra un mensaje de error en caso de que exista algún campo vacío o con caracteres incorrectos	
EC8.1: Mostrar reporte de evaluación	Se muestra al evaluador el reporte final donde se especifica el nivel de seguridad que posee su sistema	El resultado de la evaluación es generado por el sistema automáticamente	El sistema muestra los resultados arrojados después de realizar ETL y aplicar las métricas de seguridad a estos	Opción Reporte de evaluación

			resultados. En caso de existir problema con los datos necesarios para la evaluación, el sistema lo informará mediante un mensaje de error	
EC8.2: Exportar reporte a formato pdf	Esta tarea permite al evaluador exportar el resultado de la evaluación a pdf para una mejor visualización.	N/P	El sistema devuelve la evaluación en el formato especificado	Opción Reporte de evaluación / Opción Generar PDF

➤ **Escenario “Consultar historial de la evaluación de los proyectos”**

**Tabla 48: Descripción del caso de prueba “Consultar historial de la evaluación de los proyectos”**

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC9: Consultar el historial de la evaluación de los proyectos	El escenario comienza cuando el usuario selecciona la opción “Historial de evaluaciones” y el sistema muestra un listado con todos los proyectos, la fecha de su evaluación y el nivel general de seguridad así como los resultados de	(N/P)	El sistema muestra el historial de las evaluaciones realizadas a los proyectos	Opción Historial de evaluaciones

	cada una de las métricas aplicadas			
--	------------------------------------	--	--	--

➤ **Escenario “Notificar los resultados de la evaluación de un proyecto”**

**Tabla 49: Descripción del caso de prueba “Notificar los resultados de las evaluaciones de los proyectos”**

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
EC10: Notificar los resultados de las evaluaciones del proyecto	Permite notificar a la alta gerencia la evaluación o nivel de seguridad de la aplicación evaluada.	Notificar (Válido)	a. El sistema envía la notificación vía correo electrónico	Opción Notificación de evaluaciones
		Notificar (Inválido)	a. El sistema muestra un mensaje de error indicando que faltan campos o existe algún error en estos	
EC10.1: Introducir datos del contacto	Su función es permitir que el usuario inserte sus datos para posteriormente notificarle los resultados	(N/P)	El sistema guarda los datos del contacto	Opción Notificación de evaluaciones

➤ **Escenario “Graficar los resultados de las evaluaciones”**

**Tabla 50: Descripción del caso de prueba “Graficar evaluaciones de un proyecto”**

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
-----------	-------------	-------	-----------------------	---------------

EC 11. Graficar los resultados de las evaluaciones del proyecto	El escenario comienza cuando el usuario accede a la opción "Graficar evaluaciones de un proyecto" y el sistema muestra una gráfica con los resultados	El sistema gráfica los resultados de las evaluaciones automáticamente	El sistema muestra la gráfica que refleja los resultados de las evaluaciones, en caso de no existir ningún proyecto evaluado el sistema mostrará un mensaje informando al usuario	Opción Graficar evaluaciones de un proyecto
-----------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------