Universidad de las Ciencias Informáticas
Facultad 4

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

Herramienta para detectar cambios en el código fuente del Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES)

Autor: Norlys Suárez Valderrama

Tutores: Ing. Yordanis Rodríguez Rodríguez

Ing. Roberto Alejandro García Rodríguez

Ing. Daniel Díaz León

La Habana, junio de 2017

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 21 días del mes de junio del año 2017.

Autor

_____

Norlys Suárez Valderrama

Tutores:

_____ _____

Ing. Yordanis Rodríguez Rodríguez Ing. Roberto Alejandro García Rodríguez

_____

Ing. Daniel Díaz León

Agradecimientos

A mi mamá y a mi papá, a ellos les debo todo lo que soy, no hay palabras para expresar lo agradecido que estoy por el apoyo brindado desde que vine a este mundo.

A mi novia, por siempre estar presente en los buenos y malos momentos, y a su familia por brindarme su apoyo incondicionalmente.

A mis tías Irene y Julia, gracias por apoyarme siempre a lo largo de estos años.

A mi prima Yosleida y a mis primos Leosvany, Alexey, Nosley, Norisbel y Hoydel junto a sus respectivas familias que nunca dejaron de confiar en mí y me hicieron saber que en todo momento puedo contar con ellos.

A mi tío Mario Pablo y a mi abuela Paula, por ayudarme y desearme lo mejor durante estos años de estudios.

A esas personas que hoy no están a mi lado, pero que en su momento nunca dejaron de preocuparse por su querido nieto, ellos son: mis abuelos Mario y Daniel y mi abuela Margarita, a los tres donde quiera que estén les estoy eternamente agradecido por todo el amor que me dieron.

A mis otros tíos Miguel y Leovel, que siempre han estado pendiente a mis estudios.

A mis tutores Roberto, Yordanis y Daniel por toda la ayuda que me dieron durante la confección de este trabajo.

Al profesor Jorge Luis Piña, por su valiosa y muy necesaria ayuda con la herramienta desarrollada.

Agradezco infinitamente la ayuda de todos mis compañeros de estudios a lo largo de estos años, tanto a los de mi grupo como a los del 4501, a los que vienen en camino y a los que no pudieron terminar agradecerles también por poder contar con su amistad, a la gente del apartamento un agradecimiento especial por apoyarme y por convivir con ellos en estos tres últimos años.

A mis amigos de la infancia Abrandi y Reinier a quienes siempre les estaré agradecido por tener la dicha de haberlos conocidos.

Al equipo de desarrollo de SIGIES que de una forma u otra me ayudaron con aspectos claves de la aplicación.

A todas las personas que tomaron parte de su tiempo para preguntarme por la tesis y que me brindaron su ayuda desinteresadamente.

¡A todos, muchas, muchísimas gracias!

Dedicatoria

A mi mamá y a mi papá.

A mi novia.

A toda mi familia.

Resumen

La informatización de la sociedad cubana es un proceso que se lleva a cabo paulatinamente y en el que juega un papel importante la Universidad de las Ciencias Informáticas (UCI). Muchos son los productos que se desarrollan en la universidad para su uso tanto en el territorio nacional como internacional. Uno de estos es el Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES), encargado de asignarle a los estudiantes que aspiran ingresar a la Educación Superior la carrera que van a cursar. El mismo no cuenta con un mecanismo para detectar cambios en su sistema de ficheros, por lo que se hace necesario la confección de una herramienta para monitorear y notificar estos cambios. El presente trabajo tiene como objetivo fundamental el desarrollo de una herramienta que permita detectar cambios en el código fuente de SIGIES. Para el desarrollo de la investigación se realizó el estudio perteneciente al estado del arte, además del análisis de las soluciones similares. Se utilizó como metodología de desarrollo de software el Proceso Unificado Ágil (AUP) en su versión UCI y varias técnicas y herramientas que facilitaron el desarrollo de la solución.

Palabras clave: cambios, notificación, monitoreo.

Índice

Introducción

En los últimos años se han venido desarrollando con gran auge las Tecnologías de la Informática y las Comunicaciones (TIC), provocando una gran competencia en el mundo del software con herramientas cada vez más avanzadas. Por sus características las TIC han tenido un papel importante en la sociedad al transformar las formas de comunicación, llegando a influir en los distintos problemas que se presentan en prácticamente todos los ámbitos. A través de ellas se abren nuevas posibilidades de investigar e innovar, permitiendo nuevas creaciones y potenciando el desarrollo de las distintas habilidades comunicativas que existen en la actualidad.

Las TIC como ninguna otra tecnología, originó cambios en la sociedad, en la cultura y en la economía. La humanidad viene alterando significativamente los modos de entretener, de trabajar, de negociar, de gobernar y de socializar, sobre la base de la difusión y uso de la tecnología a escala global. Es reconocido que el uso de las TIC ha propiciado un aumento en la productividad en diversos sectores empresariales, hecho que algunos años atrás no era imaginable (1).

En la actualidad las TIC están estrechamente relacionadas con la enseñanza universitaria, sobre todo en el proceso enseñanza – aprendizaje y haciendo énfasis en la docencia, en los cambios de estrategias didácticas de los profesores y en los sistemas de comunicación y distribución de los materiales de aprendizaje. Así en las universidades podemos encontrar multitud de experiencias, incluyendo proyectos institucionales que responden, en muchos casos, a iniciativas particulares (2).

Existen diversas instituciones que han incorporado el uso de las tecnologías como parte fundamental de su proceso de formación, destacándose la Universidad de las Ciencias Informáticas (UCI) en este aspecto. En la UCI, se llevan a cabo varios proyectos con gran impacto en el territorio nacional. Uno de estos es el Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES) que tiene como objetivo principal asignar a los estudiantes que aspiran ingresar a la Educación Superior la carrera que van a cursar dependiendo, entre otros factores, del promedio alcanzado en sus respectivos exámenes de ingreso.

Este proceso de ingreso a la Educación Superior es controlado por el Ministerio de Educación Superior (MES), que fue creado en 1976 con el objetivo de aplicar la política educacional en el nivel de la enseñanza superior y dirigirla metodológicamente. En él se ha visto el legado de los más ilustres educadores cubanos, con las actuales exigencias científico – técnicas de la formación de profesionales, imprimiéndole a dicho proceso educativo una personalidad propia

que, sin desconocer las principales tendencias actuales vigentes en otros países, tiene un gran reconocimiento internacional (3).

Los Centros de Educación Superior (CES) se encuentran vinculados al MES que a su vez se encarga de preparar los exámenes de ingreso y de confeccionar el plan de plazas de acuerdo a la necesidad de cada localidad del país. El mismo convoca a todas las universidades del país a trabajar de un modo coherente en una estrategia de desarrollo común estableciendo una alianza entre las propias universidades, los centros de producción y servicios que potencia el proceso de formación convirtiéndolos en una tarea de primera prioridad a escala social.

El SIGIES gestionará los subprocesos de Organización, Exámenes, Asignación y Otorgamiento comprendidos en el proceso de ingreso a la Educación Superior. Estos subprocesos incluyen elementos como son:

- Plan de plazas.
- Discapacitados que optan por carreras.
- Gestión de la realización de las convocatorias de exámenes.
- El anonimato.
- Introducción de las calificaciones obtenidas por los estudiantes.
- Proceso de asignación de carreras (4).

Una vez implantado en las Comisiones de Ingreso Provincial (CIP), el SIGIES manejará información sensible con respecto a los datos personales y evaluaciones de los estudiantes en las pruebas de ingreso a la Educación Superior. En los últimos años el MES ha tenido problemas con los exámenes de ingreso, pues en varias ocasiones se han filtrado las pruebas provocando la reanudación de los exámenes en fechas posteriores. Esto ha provocado cierto descontento en la población y en los propios estudiantes que muchas veces muestran su insatisfacción con la carrera que se les asigna, pues el proceso de asignación de carreras también tiene deficiencias.

Al realizarse parte de este proceso de asignación de carreras de forma manual está expuesto a diferentes vulnerabilidades entre las que se puede mencionar, el mal manejo de toda la información que conlleva este proceso, dígase, datos personales de los estudiantes y las notas obtenidas en los exámenes de ingreso. Esto puede provocar una mala asignación de carreras, existiendo la posibilidad de que un determinado estudiante no obtenga la carrera deseada con

relación a las calificaciones obtenidas. Por otra parte, existe la posibilidad de que haya estudiantes que se excluyan del listado de cantera para el Servicio Militar Activo (SMA), del cual también se encargará el SIGIES.

En relación a los fraudes cometidos, se pudo comprobar a través del sitio www.cubadebate.cu los delitos en el proceso de ingreso a la enseñanza superior del curso escolar 2014 – 2015, en la provincia de La Habana. Estos delitos están relacionados directamente con las pruebas de ingreso a la Educación Superior, pues estas tributan al buen promedio que cada estudiante obtendrá para el proceso de asignación de carreras.

En la actualidad, no se cuenta con un mecanismo que le ofrezca seguridad a todos los ficheros con que cuenta el SIGIES. Desde el lugar donde esté implantado el sistema pudieran ocurrir situaciones extraordinarias que provoquen un mal funcionamiento del mismo, haciendo necesario tener un control y monitoreo de los cambios en todos sus archivos. Además, no se cuenta con un Sistema de Detección de Intrusos (IDS) capaz de detectar accesos no autorizados, tanto a sus archivos de configuración como a los ficheros generados en el proceso de instalación. Siendo lo anterior descrito la principal problemática, se hace necesario realizarle al proyecto un monitoreo de todos sus ficheros planteándose el siguiente problema a resolver:

Problema científico: ¿Cómo detectar posibles cambios en el código fuente de SIGIES?

Objetivo general: Desarrollar una herramienta que permita identificar cambios en el código fuente de SIGIES.

El **objeto de estudio** recae en las herramientas que detecten cambios en los archivos de un directorio y el **campo de acción** en las herramientas que permitan identificar cambios en el código fuente de SIGIES.

Hipótesis: si se desarrolla una herramienta para identificar cambios en el código fuente de SIGIES se podrá realizar un monitoreo de los ficheros del mismo para detectar cualquier cambio que pudieran sufrir.

Los **objetivos específicos** que se propone la investigación son:

- Elaborar el marco teórico de la investigación para obtener los fundamentos necesarios en la propuesta de solución.
- Desarrollar la propuesta de solución.

- Realizar pruebas de software a la solución desarrollada.

Para dar cumplimiento a los objetivos propuestos se planificaron las siguientes **tareas**:

- Estudio de las principales herramientas tecnológicas que se utilizan en el desarrollo de sistemas para detectar cambios en el código fuente de un sistema.
- Fundamento de las tendencias actuales y conceptos relacionados con el desarrollo de sistemas para notificar cambios en ficheros.
- Análisis comparativo de las características y funcionalidades de soluciones similares.
- Diseño de la propuesta de solución.
- Implementación de una herramienta para detectar cambios en el código fuente de SIGIES.
- Realización de pruebas a la solución propuesta para comprobar el cumplimiento de los requerimientos de la misma.
- Corrección de las no conformidades detectadas en la realización de las pruebas.

Como **resultado** de lo anteriormente planteado se espera obtener:

- Una herramienta que permita detectar cambios en el código fuente de SIGIES.
- Documentación del proceso y del producto, entendiéndose como producto la herramienta en sí.

Métodos de investigación utilizados:

Para la investigación del presente trabajo se han utilizado diferentes métodos científicos entre ellos los métodos teóricos y los empíricos.

Métodos teóricos:

Analítico – sintético: en la presente investigación se utilizó este método para el análisis de la teoría y extracción de los principales conceptos a incluir en el marco teórico.

Histórico – lógico: se utiliza este método para realizar el estudio del arte, o sea, para investigar acerca de otras soluciones similares y de los lenguajes y metodologías de software existentes.

Métodos empíricos:

Entrevista: se utiliza para precisar cuáles son las principales funcionalidades y subprocesos con que cuenta el SIGIES.

La presente investigación está compuesta por tres capítulos, quedando estructurados de la siguiente manera:

Capítulo 1: Fundamentación teórica

En este capítulo se describen los principales elementos teóricos de la investigación, se definen los diferentes conceptos, herramientas y metodologías de desarrollo de software a utilizar. Además, se realiza un análisis de las soluciones similares existentes mediante las cuales se pudo establecer un punto de partida para posteriormente desarrollar la herramienta que permita detectar cambios en el código fuente de SIGIES.

Capítulo 2: Análisis y diseño

En este capítulo se exponen los elementos que permiten describir la propuesta de solución, tales como: modelo de dominio, requerimientos funcionales y no funcionales, especificación de los requisitos, descripción de los requisitos y se presenta el plan de entrega por etapas. Se realiza el análisis y diseño del sistema, presentando los diagramas de clases de análisis y de diseño. Se muestra además el modelo de bases de datos, presentando la descripción de las tablas que lo componen.

Capítulo 3: Implementación y pruebas

Este capítulo hace énfasis en las dos etapas presentada en el análisis y diseño describiendo lo realizado en cada una de las etapas. Se realizan las pruebas pertinentes mostrando los resultados arrojados en cada una de las iteraciones para así darle solución a las no conformidades detectadas que no hacen posible un buen funcionamiento de la aplicación.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En este capítulo se describen los conceptos fundamentales relacionados con los Sistemas de Detección de Intruso (IDS) y se realiza un estudio del estado del arte de acuerdo a la investigación realizada. Se hace un análisis de las soluciones similares existentes donde se tiene una visión global del uso de varias herramientas para monitorear un sistema informático y se desarrolla también el análisis y descripción de las distintas herramientas y tecnologías, con el objetivo de seleccionar aquella que brinde mejores resultados en el desarrollo de la investigación.

1.2 Conceptos asociados al dominio del problema

Para mejorar el desarrollo de la aplicación es necesario el dominio de varios conceptos fundamentales, para ello se realizó el estudio de tres conceptos fundamentales: Sistema de detección de intrusos (IDS, por sus siglas en inglés), Notificación y Sistema de gestión de ficheros.

1.2.1 Sistema de detección de intrusos (IDS)

Según Óscar Andrés López y Misael Leonardo Prieto: ***"****El uso de los Sistemas de Detección de Intrusos extienden las capacidades en la administración de la seguridad, incluyendo auditoría, monitoreo, reconocimiento de ataques y respuestas, además provee capas adicionales de seguridad al sistema a proteger: puede reconocer ataques, y potencialmente responder a ellos, mitigando los daños. Adicionalmente, cuando los dispositivos fallan debido a ataques que cambian la configuración, ataques conocidos, o errores del usuario, los IDS pueden reconocer el problema iniciando un mecanismo de respuesta, como, por ejemplo, notificar a la persona indicada por un medio seguro****"*** (5).

Otros autores han propuesto varias definiciones de la auditoría informática entre las que vale destacar la de Rafael Castillo Santos: *"Un Sistema de Detección de Intrusos es todo aquel que resulta de la combinación tanto de software como hardware, que mediante alguna acción, alarma o indicador permite establecer con algún grado de precisión, cuándo se lleva o llevó un ataque a un sistema informático"* (6).

Por otra parte, Emilio José Alfaro propone que: *"Un Sistema de Detección de Intrusos es una herramienta de seguridad encargada de monitorizar los eventos que ocurren en un sistema informático en busca de intentos de intrusión".* Además, define un intento de intrusión *"como*

cualquier intento de comprometer la confidencialidad, integridad, disponibilidad o evitar los mecanismos de seguridad de una computadora o red" (7).

A partir del análisis realizado a las definiciones anteriores acerca de los IDS, se considera que la de Emilio José Alfaro es la que más se asemeja al tema que aborda la presente investigación.

Funciones de un IDS

Los IDS cuentan con varias funciones entre las que se pueden mencionar las siguientes:

- Monitoreo y análisis de la actividad de los usuarios y del sistema.
- Auditoría de configuraciones del sistema y vulnerabilidades (*firewall*, *routers*, servidores, archivos críticos, entre otros).
- Evaluación de la integridad de archivos de datos y sistemas críticos.
- Reconocimiento de patrones reflejando ataques conocidos (5).

Dentro de estas funciones el autor de la presente investigación destaca la evaluación de la integridad de datos haciendo énfasis en el objetivo principal del trabajo, pues a través de ella se podrá detectar el cambio en los ficheros. La nueva herramienta se enmarca en esta función haciendo uso de un IDS.

1.2.2 Notificación

El estudio de varias definiciones permitió identificar que una notificación es un proceso mediante el cual se le informa a alguien acerca de una determinada circunstancia que la incumbe. Dicho proceso puede darse en una pluralidad de contextos, público, privado, a nivel de una persona física, jurídica, entre otros (8).

En el área de la informática las notificaciones no son más que alertas que emiten ciertos programas o servicios para advertir algo al usuario, dígase algún fallo, operaciones o cambios realizados en el sistema (9).

1.2.3 Sistema de gestión de ficheros

Los sistemas gestores de archivos han evolucionado hasta convertirse en herramientas de gestión integral, que permiten cubrir el ciclo vital de los documentos, desde su fase de tramitación administrativa hasta su definitiva transferencia al archivo histórico (10).

Remei Morera en su criterio establece los siguiente: *"Los sistemas de gestión de archivos son sistemas de gestión de bases de datos específicos para el tratamiento documental de los archivos e incorporan todas las prestaciones para la informatización integral de las funciones propias de un archivo: descripción y recuperación de fondos, gestión documental, control de depósitos y gestión de la consulta y el préstamo"* (13). Además, añade lo siguiente: *"Los sistemas gestores de archivos tienen la ventaja de que al ser programas gestados desde una óptica archivística incorporan la automatización de todos (o casi todos) los procesos de gestión documental de un archivo. En consecuencia, se reducen los costes de diseño y elaboración de la aplicación puesto que la estructura de los campos está ya fijada, el sistema de consulta está ya definido y los informes de salida están ya diseñados"* (11).

Se realizó el estudio de otras definiciones propuestas por varios autores, sin embrago, se considera que la de Remei Morera es la que más se asemeja al tema que aborda la presente investigación.

1.3 Estudio de herramientas similares

Como parte de la investigación realizada se estudiaron varias herramientas similares relacionadas con la notificación de cambios en un sistema de ficheros de donde se toma la que se utilizará como base para el desarrollo de la nueva herramienta.

1.3.1 Ossec

Ossec es un sistema de detección de intrusos basado en host usado en diferentes sistemas para realizar funciones de monitoreo, pues posee una configuración que permite chequear todo el Sistema Operativo, así como archivos y directorios especificados por el usuario. Permite tres tipos de instalaciones: servidor, local y agente, en la primea podemos instalar el servidor y después configurarlo para darle seguimiento a otras computadoras que se instalarán en el modo agente.

Esta herramienta en comparación con otras de su tipo posee parámetros de configuración que son fáciles de entender por parte de los usuarios. En su archivo de configuración principal cuenta con un grupo de opciones entre las que se encuentra la notificación por correo electrónico, brindando la posibilidad de especificar el tiempo mediante el cual se desea recibir dicha notificación. Además, admite el trabajo con un número predeterminado de reglas donde el usuario puede escoger las que desee para que Ossec las tome en cuenta. Tiene opciones para introducir directorios a chequear explícitamente y una opción para alertar en caso de que aparezcan nuevos archivos en esos directorios.

Ossec trabaja con varios módulos que vienen integrados en su instalación, tal es el caso de Syscheck, Rootcheck y Active response, este última posee la opción para bloquear un número determinados de IP y deshabilitar a ciertos usuarios a los que no se les quiera dar acceso al sistema. Una de sus funcionalidades fundamentales es que brinda la posibilidad de almacenar esta información que genera en bases de datos, pues en su configuración es admisible tanto para Postgresql como Mysql.

1.3.2 WinAudit

WinAudit es un software de auditoría que crea un informe completo sobre la configuración, el hardware y el software de una máquina. Es libre, de código abierto, puede ser utilizado o distribuido por cualquier persona y está publicado bajo la Licencia Pública de la Unión Europea (EUPL) (12).

Al ejecutar WinAudit en una computadora se muestra una interfaz con todas las configuraciones realizadas en la misma y junto a ello tenemos toda la información referente al hardware y software con que cuenta el sistema. Además, tenemos información de todos los programas instalados y de aquellos que se están ejecutando en el momento, en la interfaz se puede observar la carpeta de instalación de los programas, la fecha de instalación, la empresa productora del software y el estado de la instalación. Es mucha la información mostrada por esta herramienta relacionada con el sistema donde se ejecuta, sin embargo, no hace referencia a cambios realizados en archivos específicos del sistema.

1.3.3 Systraq

Systraq es una herramienta disponible para los Sistemas Operativos Linux que monitorea y controla los archivos del sistema avisando cuando hay un cambio en uno de ellos. Envía un correo electrónico diario indicando el estado del sistema y en caso de que se hayan modificado algunos de los archivos más críticos se recibe este correo con una notificación mucho más breve (13).

Esta herramienta puede ayudar a implementar una política de seguridad según desee el usuario, pues primeramente se debe observar la configuración que trae por defecto, pues puede que no cumpla con la política de seguridad de algún sitio. Ejecuta varios comandos del sistema para inspeccionar el estado del sistema, estos comandos se encuentran en pequeños *scripts* a donde el usuario puede acceder a modificarlos o a agregar sus propios *scripts*.

1.3.4 Diffmon

Esta herramienta también desarrollada para Linux toma un *diff* de archivos de configuración del sistema especificados y los envía por correo electrónico a una dirección de correo especificada. Las opciones de *diff* se pueden especificar, esto es útil en entornos amigables donde hay varios administradores de sistemas que trabajan en las configuraciones de los archivos y se reportan los cambios entre ellos (14).

Dentro de Diffmon se destacan algunos comandos entre los que se destacan –C utilizado para comprimir las imágenes de archivos guardadas, -c para configurar un archivo y usarlo en lugar del predeterminado y –e que es el encargado de mantener informado al usuario a través del correo electrónico. A pesar del fácil uso que se le puede dar a esta herramienta tiene como desventaja el no poder escoger los archivos de configuración a los que se le quiere dar seguimiento (14).

1.3.5 Entorno Avanzado de Detección de Intrusos (AIDE)

AIDE crea una base de datos a partir de las reglas de expresión regular que encuentra en los archivos de configuración. Una vez que se inicializa esta base de datos se puede utilizar para verificar la integridad de los archivos. Cuenta con varios algoritmos de resumen de mensajes que se utilizan para comprobar la integridad del archivo dentro de los que se puede mencionar el MD5 (*Message Digest Algorithm* 5, por sus siglas en inglés) y el SHA1 (*Secure Hash Algorithm* 1, por sus siglas en inglés). Todos los atributos archivos habituales también se pueden comprobar para detectar inconsistencias y puede leer base de datos de versiones más antiguas o más nuevas (15).

La base de datos de AIDE almacena varios atributos de archivo incluyendo: tipo de archivo, permisos, usuario, grupo, tamaño de archivo y número de enlaces. También, crea una suma de comprobación o hash criptográfica de cada archivo utilizando una combinación de los algoritmos de resumen de mensajes. Cuando el administrador crea la base de datos en el nuevo sistema esta es una instantánea del sistema en su estado normal y el criterio por el cual se medirán todas las actualizaciones y cambios posteriores (15).

1.3.6 Integrit

Integrit es otras de las alternativas usadas en los programas de verificación de la integridad de los archivos. Ayuda a determinar si un intruso ha modificado algunos de los archivos en un sistema informático. Funciona creando una base de datos con una instantánea de las partes más

esenciales del sistema, esta base de datos puede ser utilizada para verificar que nadie ha hecho modificaciones ilícitas en el sistema (16).

Para su funcionamiento Integrit utiliza la librería Boehmy y en su código fuente hace uso de tabla hash por lo que se puede utilizar en cualquiera aplicación que utilice esta estructura de datos. El algoritmo utilizado para comprobar la integridad de los datos es el rmd160 ya que este cuenta con muy pocas debilidades. Aunque no es el único algoritmo que se puede utilizar, pero la base de datos sería mucho más grande, el tiempo de ejecución sería más lento y esto implicaría más complejidad para el usuario (17).

1.3.7 Conclusión del análisis de las soluciones similares

Después del análisis de las herramientas existentes para controlar y monitorear un sistema informático, se decide usar como base a la herramienta Ossec puesto que las demás presentan varias debilidades. En el caso de Integrit y AIDE necesitan la creación de una base de datos para su buen funcionamiento, Systraq y Diffmon poseen una configuración muy compleja para el usuario y en el caso de WinAudit se encuentra disponible únicamente para el Sistema Operativo Linux.

Ossec en comparación con las herramientas antes mencionadas presenta como ventaja que permite escoger los archivos a los cuales se les quiere dar seguimiento, sin necesidad de monitorear todos los archivos del sistema. El trabajo con los principales módulos que vienen integrados en su instalación hace posible su buen funcionamiento, destacándose Syscheck el cual se encarga de chequear los directorios para verificar la integridad de todos sus archivos.

1.4 Metodología de desarrollo de software

Según lo define Iván Jacobson un proceso de software detallado y completo suele denominarse Metodología. Varias son las metodologías que se han desarrollado a lo largo de los años, cada una de ella con sus respectivas características. Estas metodologías se clasifican en dos grandes grupos, las metodologías tradicionales y las metodologías ágiles, las cuales se explican a continuación:

1.4.1 Metodologías tradicionales

Las metodologías tradicionales se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Este enfoque tradicional ha demostrado ser efectivo y

necesario en proyectos de gran tamaño, respecto a tiempo y recursos, donde por lo general se exige un alto grado de ceremonia en el proceso. Algunas de estas metodologías más conocidas se encuentran: RUP y Microsoft Solution Framework (MSF) (18).

1.4.2 Metodologías ágiles

El término ágil fue aplicado en una reunión celebrada en febrero del 2001 en Utah, Estados Unidos. En esta reunión participaron varios expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Aquí se creó The Agile Alliance (La alianza ágil), una organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto ágil, un documento que resume la filosofía ágil (18).

Las metodologías ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental de software con iteraciones muy cortas. Están especialmente orientadas a proyectos pequeños, las cuales constituyen una solución a medida para ese entorno, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. Dentro de estas metodologías se pueden mencionar las siguientes: SCRUM, Crystal Methodologies, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD) y Extreme Programming (XP) (18).

1.4.3 AUP – UCI (Proceso Unificado Ágil, versión UCI)

AUP (Proceso Unificado Ágil): Es una versión simplificada de Rational Unified Process (RUP). Describe un enfoque simple y fácil de entender para desarrollar software de aplicaciones empresariales que utiliza técnicas y conceptos ágiles, pero que sigue siendo fiel al RUP. La disciplina de modelación incluye la modelación del negocio, requisitos, análisis y diseño. Por otra parte, se integran además la Gestión de Cambios y Gestión de Configuración en una sola disciplina (19).

AUP - UCI surge al no existir *“una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable”*. Por lo que *“se decide hacer una variación de*

la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI" (20).

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre. Además, AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP (20).

Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto) (20).

Por otra parte, a partir de que el modelado de negocio propone varias variantes a utilizar en los proyectos y existen tres formas de encapsular los requisitos surgen cuatro escenarios para modelar el sistema en los proyectos:

- Proyectos que modelen el negocio con CUN (Caso de Uso del Negocio) solo pueden modelar el sistema con CUS (Casos de Uso del Sistema).
- Proyectos que modelen el negocio con MC (Modelo Conceptual) solo pueden modelar el sistema con CUS.
- Proyectos que modelen el negocio con DPN (Descripción de Proceso de Negocio) solo pueden el sistema con DRP (Descripción de Requisitos por Proceso).
- Proyectos que no modelen negocio solo pueden modelar el sistema con HU (Historia de Usuario) (20).

Como metodología de desarrollo de software se utilizará AUP – UCI, pues brinda múltiples variantes a los proyectos para modelar el sistema a través de los cuatro escenarios, guiando el

desarrollo de la herramienta propuesta mediante el escenario tres (DRP). Es importante destacar que es la metodología utilizada tanto en el proyecto SIGIES como en otros proyectos de la UCI.

1.5 Herramienta para el modelado

Para modelar la nueva herramienta que se va a desarrollar se hace necesario la investigación de varias herramientas de este tipo, para esto se estudiaron dos herramientas en específico mostrando características de ellas y escogiendo una como herramienta de modelado.

1.5.1 Lenguaje de Modelado Unificado (UML)

UML es un lenguaje de modelado orientado a objetos que permite representar gráficamente los elementos estáticos y dinámicos de una aplicación de software. UML permite una modelación de los componentes estáticos de una aplicación de software (diagramas de casos de uso, diagramas de clases), así como del comportamiento dinámico de sus principales elementos durante su funcionamiento (entre ellos diagramas de estados y diagramas de secuencias). Los diagramas de estados permiten la modelación de los principales estados y los eventos que ocasionan sus cambios para una instancia de una clase, o para un sistema como un todo, mientras que los diagramas de secuencias permiten modelar instancias de interacción entre actores u objetos de clases de un sistema a través de mensajes. Mediante estos últimos diagramas es posible conocer lo que ocurre internamente entre los actores e instancias de clases que participan en un diagrama de estados de un sistema de software (21).

Los principales objetivos en el diseño de UML son obtener un lenguaje simple pero suficientemente expresivo, que permita modelar aplicaciones en cualquier dominio y obtener un lenguaje legible, puesto que sería un lenguaje utilizado por las personas. Para combinar la simplicidad con la aplicabilidad a cualquier dominio, UML incorpora un conjunto de mecanismos de extensibilidad que permiten definir perfiles que lo adaptan a un domino concreto (22).

La aparición de UML ha supuesto el reconocimiento de la actividad del modelado como una actividad clave para producir software de calidad. Esta es un área que cada vez cobra más importancia y que no se reduce a el ámbito académico, sino que está recibiendo atención por parte de las principales empresas de desarrollo de software.

1.5.2 Herramientas de modelado que utilizan UML

Entre las herramientas de modelado que utilizan UML se encuentran las siguientes:

- Umbrello.

- ArgoUML.
- Rational Rose.
- BoUML.
- MagicDraw UML.
- Visual Paradigm.

1.5.3 Umbrello

Umbrello es la herramienta de software libre más avanzada de su clase y como característica está empezando a competir con las distintas herramientas comerciales existentes. Consta de más de sesenta y cinco mil líneas de código incluyendo comentarios y se compone de casi trescientos archivos. Utiliza una serie de tecnologías y recursos y está escrito en C++ para los sistemas operativos UNIX (23).

Permite instalarse en diferentes plataformas y posee más de 30 idiomas diferentes, gracias a su licencia original GPL. Brinda la posibilidad de:

- Copiar, cortar y pegar los objetos de los diferentes diagramas, puede copiar los objetos como imágenes PNG de forma que pueda insertarlos en cualquier tipo de documento.
- Exportar como imagen un diagrama completo.
- Imprimir diagramas individuales.
- Organizar mejor la maqueta, especialmente en los proyectos grandes (23).

1.5.4 Visual Paradigm

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computación) que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal (24).

Entre sus principales características se encuentran las siguientes:

- Disponibilidad en múltiples plataformas.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- Disponibilidad de múltiples versiones, con diferentes especificaciones.
- Licencia: gratuita y comercial.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Soporte de UML versión 2.1 (24).

1.5.5 Selección de la herramienta para el modelado

Se selecciona Visual Paradigm, en su versión 8.0 la cual cumple con el Standard UML 2.0 y 2.1. Esta herramienta es compatible con varias plataformas y permite modelar todo el ciclo de desarrollo del software. Dos aspectos relevantes en la selección de la misma, son su uso por parte del proyecto SIGIES en la elaboración de los artefactos ingenieriles generados a partir de la metodología de desarrollo y la experiencia previa en su utilización que tiene el equipo de trabajo.

1.6 Marco de trabajo Xalix

En el Centro FORTES se ha comenzado la formalización de un marco de trabajo en cada Línea de Productos de Software (LPS), con el propósito de disminuir la diversidad tecnológica de las soluciones y facilitarle el trabajo al cliente, permitiéndole utilizar distintas funcionalidades tanto de RHODA, ZERA como de CRODA, en un mismo marco de trabajo denominado Xalix.

Xalix cuenta con un conjunto de herramientas y tecnologías, las cuales se mencionan a continuación:

- Framework de desarrollo: Symfony, versión 2.7.

- Lenguaje de programación para el servidor: PHP versión 7.0.
- Lenguaje para el cliente: HTML versión 5.
- Gestor de base de datos: PostgreSQL versión 9.4.
- Librería de CSS: Bootstrap versión 3.3.6.
- Librería de JavaScript: jQuery versión 2.1.4.

Para un mejor desarrollo de la nueva herramienta se necesita que la misma se elabore sobre las bases de Xalix, es decir, que se implementen con tecnologías compatibles a las desarrolladas en el marco de trabajo.

1.7 Lenguajes de desarrollo

"Los lenguajes de programación, se pueden definir como lenguajes artificiales diseñados por el ser humano para la comunicación con sistemas computacionales. Se utilizan para desarrollar programas que controlan el comportamiento lógico o físico de una máquina, implementar algoritmos e incluso para la comunicación entre personas". Estos están conformados por un conjunto de reglas sintácticas y semánticas que dan orden a su estructura y el significado de sus expresiones. La definición de un lenguaje de programación consiste básicamente de tres elementos:

- Sintaxis.
- Semántica.
- Pragmática (25).

1.7.1 PHP

PHP (Hypetext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de otros lenguajes es que el código es ejecutado en el servidor, generando HTML, y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era (26).

Lo mejor de utilizar PHP es su extrema simplicidad para quien comienza a usarlo, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. Está enfocado a la programación de scripts del lado del servidor, por lo que se puede hacer lo que se desee,

como recopilar datos de un formulario, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Existen principalmente tres campos donde se usan los scripts en PHP: scripts del lado del servidor, scripts desde la línea de comandos y escribir aplicaciones de escritorio (27).

Entre las ventajas que presenta PHP se encuentran las siguientes:

- PHP puede emplearse en todos los sistemas operativos por lo que es multiplataforma y admite la mayoría de servidores web de la actualidad.
- Se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.
- Soporte para varias bases de datos.
- Permite comunicarse con otros servicios usando protocolos tales como IMAP, SNMP, HTTP y POP3.
- Soporte para el intercambio de datos complejos entre virtualmente todos los lenguajes de programación web y para la instalación de objetos de Java y emplearlos de forma transparente como objetos de PHP.
- Tiene características de procesamiento de texto, las cuales incluyen las expresiones regulares y muchas extensiones y herramientas para el acceso y análisis de documentos XML (27).

1.7.2 HTML y CSS

HTML (HyperText Mark-up Language) es el lenguaje de marcado predominante en el desarrollo de páginas web, se utiliza en la traducción y descripción de la estructura y la información en forma de texto, así como para complementar este con otros elementos. De manera general, permite definir la estructura y el contenido de las páginas, permitiendo combinar textos, imágenes, sonidos, vídeos y enlaces a otras páginas y entre sus características resalta la reducción de la necesidad de *plugins* externos y un mejor manejo de errores (28).

En cuanto a HTML, su última versión es la 5 y permite explorar nuevas etiquetas semánticas como <*header*>, <*footer*>, <*nav*>, <*audio*>, <*video*>, <*article*> y <*aside*>. Todas esas etiquetas semánticas nos indican qué es el contenido que engloban y cuál es su relación con el conjunto de elementos del documento HTML con el fin de definir el esquema principal del documento (29).

CSS (Cascade Style Sheet) fue creado como un lenguaje formal usado para definir el aspecto de un documento estructurado escrito en HTML. Permite separar el contenido de una página de su presentación, a la vez que permite a los diseñadores mantener un control preciso sobre la apariencia de las páginas web (30). Las CSS presentan un conjunto de características destacándose las siguientes:

- Mantenibilidad
- Simplicidad
- Flexibilidad
- Accesibilidad
- Combinación con lenguajes alternativos (31).

1.7.3 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos tales como textos que aparecen y desaparecen, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlo (32).

Como todo lenguaje de programación JavaScript tiene ciertas ventajas que le facilitan el trabajo al usuario, algunas de estas ventajas se describen a continuación:

- Velocidad: JavaScript es muy rápido y cualquier función puede ser ejecutada inmediatamente en lugar de tener que contactar con el servidor y esperar una respuesta.
- Simplicidad: es un lenguaje de programación relativamente simple de aprender e implementar.
- Versatilidad: encaja perfectamente en otros lenguajes y puede ser usado en una gran variedad de aplicaciones y scripts escritos en otros lenguajes como Perl o el propio PHP.
- Carga del servidor: al ejecutarse del lado del cliente reduce la carga en el servidor de la página web (33).

1.7.4 Fundamentación de los lenguajes a utilizar

Para el desarrollo de la solución será utilizado PHP en su versión 7.0, HTML versión 5, CSS versión 3 y JavaScript, los mismos se encuentran dentro del marco de trabajo Xalix y son los utilizados en el proyecto SIGIES.

1.8 Frameworks de desarrollo

Desde el punto de vista del desarrollo de software, un *framework* es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Los *frameworks* suelen incluir soporte de programas, bibliotecas, lenguajes de *scripting* y software para desarrollar y unir diferentes componentes de un proyecto (34).

1.8.1 Symfony

Symfony2, es la versión más reciente del *framework* Symfony para el desarrollo de aplicaciones utilizando el lenguaje PHP. Es un *framework* completo, diseñado para optimizar el desarrollo de las aplicaciones web, basado en el patrón Modelo Vista Controlador. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse a tiempo completo a las funcionalidades específicas de la aplicación web. Posee facilidad de instalación y configuración en plataformas Unix y Windows. Es independiente del Sistema Gestor de Base de Datos (SGBD) que se utilice. Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización (35).

Se versión 2.7, contiene diversas características y ventajas con respecto a otras versiones, entre las que se encuentran:

- Mejora en el sistema de traducciones.
- Mejoras para *debuggear*.
- Mejoras en el componente de seguridad.
- Permite incorporar *bundles* de terceros (36).

1.8.2 Bootstrap

Bootstrap es un *framework* HTML, CSS y JavaScript que permite crear interfaces web, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. El mismo está diseñado pensando en ofrecer la mejor experiencia de usuario tanto a

usuarios de computadora, como a *smartphones* y *tablet* (37). A continuación, se presentan sus principales ventajas:

- Utiliza componentes y servicios creados por la comunidad web.
- Utiliza HTML y CSS3.
- Tiene una gran comunidad que soporta este desarrollo y cuenta con implementaciones externas como WordPress, Drupal y jQuery.
- Maneja un conjunto de buenas prácticas que perdurarán en el tiempo (37).

1.8.3 jQuery

jQuery es un *framework* JavaScript que implementa una serie de clases de programación orientadas a objetos permitiendo programar sin preocuparse del navegador que utilice el usuario. Ofrece una infraestructura en la que se tiene mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Su licencia permite su uso en cualquier tipo de plataforma, personal o comercial (38). Entre sus principales ventajas se encuentran las siguientes:

- Es fácil de usar
- Está bien documentado y cuenta con un gran equipo de desarrolladores a cargo de su mejora y actualización.
- Ahorro de tiempo y esfuerzo
- Integración con AJAX.
- Permite agregar *plugins* fácilmente (38).

1.8.4 Fundamentación de los frameworks a utilizar

En el desarrollo de la solución se utilizará Symfony en su versión 2.7, la librería de CSS, Boostrap en su versión 3.3.6 y la librería para JavaScript, jQuery en su versión 2.1.4. Estos *frameworks* se encuentran dentro del marco de trabajo Xalix y son los utilizados por el proyecto SIGIES.

1.9 Servidor Web

En informática los servidores web sirven para almacenar contenidos de internet y facilitar su disponibilidad de forma constante y segura. Cuando se visita una página web desde un navegador, es en realidad un servidor web el que envía los componentes individuales de dicha

página directamente a tu ordenador. Esto quiere decir que para que una página web sea accesible en cualquier momento, el servidor web debe estar permanentemente online**.** Los servidores web son un tipo de servidores que se utilizan para la distribución de contenido web en intranets o en internet. Como parte de una red de ordenadores, un servidor web transfiere documentos a los llamados clientes, por ejemplo, una página web a un explorador (39)**.**

Detrás de un servidor web se encuentra el software, que se utiliza principalmente para ofrecer archivos que faciliten la visualización de contenido web. Para ello, el programa se comunica con un cliente web, que es por lo general, un navegador web. Aunque su principal función es la transferencia de contenido web, muchos programas de servidor web ofrecen características adicionales:

- Seguridad.
- Autenticación del usuario.
- Redirección.
- Caching.
- Asignación de *cookies* (39).

1.9.1 Nginx

Nginx es un servidor de código abierto, de alto rendimiento y es conocido por su calidad, por su sencilla configuración y bajo consumo de recursos. A diferencia de los servidores tradicionales no se basa en hilos para manejar las solicitudes. En su lugar, utiliza una arquitectura basada en eventos mucho más escalable. Esta arquitectura si no espera manejar miles de solicitudes simultáneas, puede beneficiarse del alto rendimiento y de la pequeña huella de memoria de Nginx (40).

Fue escrito por Igor Sysoev para www.rambler.ru, el sitio más visitado de Rusia, donde ha estado funcionando en la producción durante más de dos años y medio. Lo que hace que este servidor web sea diferente de los demás es la eficiencia general del *daemon* y el número de opciones de configuración (41).

Nginx genera disimiles de procesos donde cada uno puede manejar varias conexiones. Cada una de estas conexiones se colocan dentro de un bucle de eventos donde existen otras conexiones. Dentro del bucle, los eventos se procesan asincrónicamente, permitiendo que el trabajo se

maneje de manera que no se bloquee. Cuando se cierra la conexión, se quita el bucle. Este estilo de procesamiento de conexión permite a Nginx escalar increíblemente lejos con recursos muy limitados (42).

Se decide utilizar Nginx en su versión 2.7, pues el mismo es un servidor de código abierto habilitado para soportar un gran número de conexiones simultáneas. De igual manera, forma parte de las herramientas que utiliza el marco de trabajo Xalix.

1.10 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. El éxito de un SGBD consiste en mantener la seguridad e integridad de los datos. Lógicamente tiene que proporcionar herramientas a los distintos usuarios y entre estas herramientas se pueden mencionar las siguientes:

- Herramientas para la creación y especificación de los datos. Así como la estructura de la base de datos.
- Herramientas para administrar y crear la estructura física requerida en las unidades de almacenamiento.
- Herramientas para la manipulación de los datos de las bases de datos, para añadir, modificar, suprimir o consultar datos.
- Herramientas de recuperación en caso de desastre.
- Herramientas para la creación de copias de seguridad.
- Herramientas para la gestión de la comunicación de la base de datos (43).

1.10.1 PostgreSQL

Es el sistema de gestión de base de datos relacional, orientada a objetos de código abierto más avanzado del mundo. Resulta un potente sistema de base de datos objeto/relacional. Cuenta con más de dieciséis años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, integridad de los datos y corrección. Funciona en todos los principales sistemas operativos, incluyendo Linux, UNIX y Windows. Es altamente escalable,

tanto en la enorme cantidad de datos que puede administrar, como en el número de usuarios concurrentes que puede soportar (44).

Tiene varias ventajas, entre las que se pueden mencionar:

- Instalación ilimitada.
- Mejor soporte que los proveedores comerciales.
- Estabilidad y confiabilidad legendarias.
- Extensible pues el código fuente está disponible para todos sin costo.
- Multiplataforma.
- Diseñado para ambientes de alto volumen.
- Herramientas gráficas de diseño y administración de bases de datos. Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect).
- Altamente extensible, pues soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Múltiples métodos de autenticación.
- Completa documentación.
- Acceso encriptado vía SSL (44).

Se decide utilizar PostgreSQL en su versión 9.4 por su estabilidad, potencia, robustez y por poseer una gran cantidad de documentación en español en su sitio web oficial. Es la base de datos de código abierto más potente del mundo y se puede ejecutar en la gran mayoría de los sistemas operativos. Este SGBD es usado dentro del marco de trabajo Xalix y por el proyecto SIGIES.

1.11 Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Está compuesto por un conjunto de herramientas para programar en un lenguaje de programación

o varios, donde podemos encontrar como mínimo un editor, compilador, interprete y depurador de uno o varios lenguajes de programación (45).

1.11.1 PHPStorm

PHPStorm es perfecto para trabajar con algunos *framework* como Symphony, Drupal y Joomla. Apoya todas las características del lenguaje PHP y se pueden realizar muchas tareas de rutina desde este IDE gracias a la integración de los sistemas de control de versiones. Soporta base de datos SQL, es un IDE bastante rápido y cientos de inspecciones se encargan de verificar el código mientras se escribe (46).

Es uno de los entornos de programación más completos de la actualidad y es compatible con varios sistemas operativos como Windows, Linux y MAC OS X. Entre sus principales características se pueden mencionar las siguientes:

- Permite la gestión de proyectos fácilmente.
- Proporciona un fácil autocompletado de código.
- Soporta el trabajo con las más recientes versiones de PHP.
- Sintaxis abreviada (47).

1.11.2 NetBeans

NetBeans es un IDE que permite programar en diversos lenguajes, es ideal para trabajar en Java, así como también ofrece un excelente entorno para programar en PHP. Esta plataforma es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. NetBeans ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación (48).

Entre las características de la plataforma están:

- Suele dar soporte a casi todas las novedades en el lenguaje Java.
- Asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos *frameworks*.

- Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario.
- Administración de las interfaces de usuario (ejemplo: menús y barras de herramientas).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
- Administración de ventanas (48).

1.11.3 Selección del IDE de desarrollo

Se selecciona NetBeans en su versión 8.0 ya que el mismo es de código abierto, y presenta buenas funciones para el desarrollo web. Ofrece un entorno para programar en PHP, es multiplataforma y fácilmente extensible a través de *plugins.* Al mismo tiempo, cuenta con una gran comunidad de desarrolladores y es de amplio dominio el trabajo con la aplicación por parte del autor.

1.12 Propuesta de solución

Se tiene como propuesta de solución el desarrollo de una herramienta que permita mostrar los resultados arrojados al ejecutar la herramienta Ossec. Se utilizará varios módulos fundamentales dentro de Ossec entre los que se destaca Syscheck, el cual se encarga de detectar el momento en que los archivos incluidos en el monitoreo sufren alguna modificación, y Analysis encargado de generar las alertas que son usadas para la notificación de los últimos archivos modificados. La nueva herramienta será desarrollada en el lenguaje de programación PHP versión 7.0 utilizando el entorno de desarrollo integrado NetBeans versión 8.0 integrado con el servidor web Nginx versión 2.7. Como sistema de gestor de base de datos se utilizará PostgreSQL versión 9.4 por las características y soporte que brinda al lenguaje seleccionado.

1.13 Conclusiones del capítulo

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

El estudio del estado del arte permitió concluir que resulta ventajoso el uso de la herramienta Ossec por la información que brinda al ser ejecutada y por las opciones de configuración que posee.

Al realizar un análisis de las herramientas y tecnologías se consideró la selección del lenguaje de programación PHP versión 7.0 y de los *frameworks* Bootstrap en su versión 3.3.6 y jQuery en su versión 2.1.4 por ser los más aceptados en la comunidad para el trabajo con CSS 3 y JavaScript.

Como entorno de desarrollo integrado se selecciona NetBeans versión 8.0, a su vez se elige como servidor web Nginx versión 2.7 y como sistema gestor de base de datos PostgreSQL versión 9.4 por ser los que usa el proyecto SIGIES.

Capítulo 2: Análisis y diseño del sistema

2.1 Introducción

En este capítulo se define la propuesta de solución y se describen los procesos fundamentales que intervienen en el objeto de estudio dando como válido el problema a resolver que se definió anteriormente. Se muestran cada uno de los requisitos, así como las características y cualidades que el sistema debe tener. Además, se modelan los diagramas de clases del análisis y el diseño para cada requisito de la propuesta del sistema a desarrollar teniendo en cuenta el tercer escenario de trabajo que propone la metodología AUP, Análisis y Diseño.

2.2 Modelo de dominio

Un modelo de dominio o modelo conceptual, como también se le conoce, es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. El mismo se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. En el modelo se pueden mostrar objetos del dominio o clases conceptuales, asociaciones entre las clases conceptuales y atributos de las clases conceptuales. Para su confección no son adecuados artefactos del software, como una ventana o una base de datos, a menos que el dominio que se esté modelando sea de conceptos de software, como un modelo de interfaces de usuario gráficas (49).

2.2.1 Conceptos del dominio

Para un mayor entendimiento del problema se definen los términos más importantes asociados al dominio para la posterior confección del modelo:

- SIGIES: sistema que permite gestionar todo lo referente al proceso de ingreso a la educación superior.
- Ossec: IDS basado en host para monitorear el sistema.
- Base de datos: conjunto de datos que se almacenarán de la herramienta Ossec una vez que se ejecuten los comandos correspondientes.
- Administrador DIUL: persona registrada en el sistema que cuenta con todos los privilegios, es el encargado de ejecutar los distintos comandos con que cuenta el sistema.
- Comandos*:* comandos que son ejecutados para consultar los archivos que han sido modificados y las alertas generadas.

- Configuración*:* archivo de configuración de Ossec, en él se encuentra principalmente la dirección de los directorios que se deben monitorear, así como las principales reglas a incluir en el monitoreo.
- Módulos*:* principales módulos con que cuenta Ossec para su funcionamiento.
- Syscheck: módulo de Ossec encargado de analizar los directorios especificados en la configuración y de detectar aquellos archivos que han sufrido alguna modificación.
- Archivos Modificados*:* lista de archivos que han sufrido alguna modificación.
- Rootcheck: módulo de Ossec encargado de la detección de *rootkit*.
- Analysis: módulo de Ossec encargado de generar las alertas.
- Alertas: alertas generadas que contienen principalmente información de los últimos archivos modificados o a algún error del servidor web.

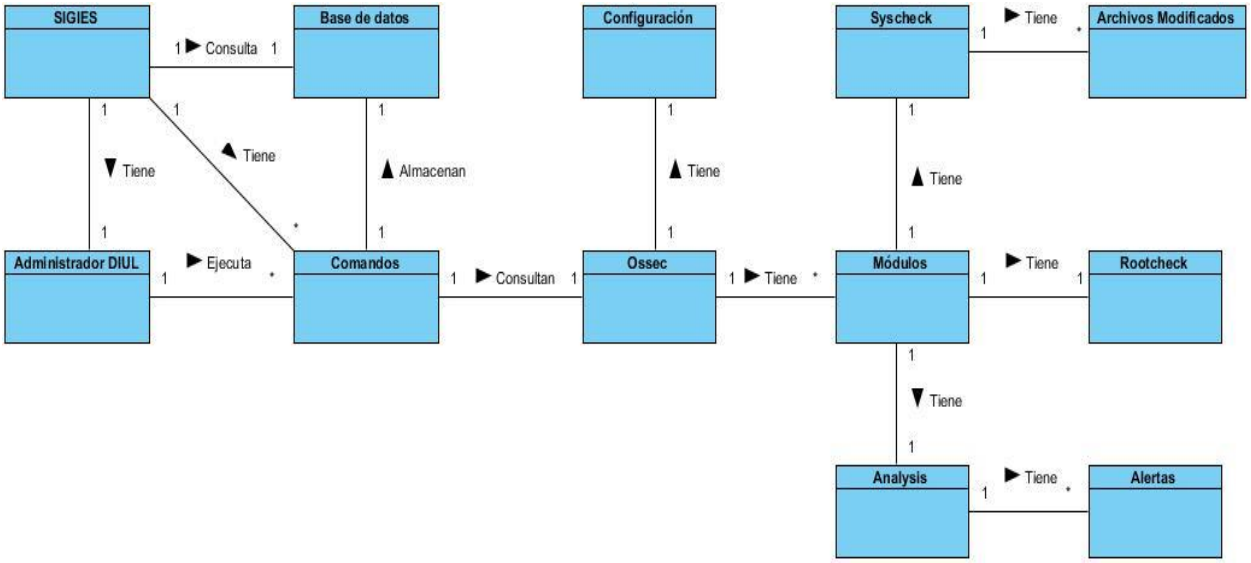

Ilustración 1. Modelo de dominio.

2.3 Descripción del sistema

Inicialmente la descripción inicial de un sistema informático les permite a los clientes imaginar y avizorar como quedará el producto final y desde ese entonces, serán ellos los encargados de cometer nuevas peticiones de acuerdo a sus necesidades. Es habitual que en los primeros encuentros con el cliente estos hagas peticiones irregulares de requisitos e ideas de interfaz de usuario.

La herramienta a desarrollar debe permitir visualizar los resultados que arroja Ossec al ser ejecutado, la misma realizará un monitoreo al sistema dependiendo de la configuración establecida una vez que se instale el SIGIES. Para su funcionamiento el usuario deberá ejecutar dos comandos en específico, uno de ellos consultará a Ossec para buscar aquellos archivos que han sido modificados y el otro realizará la misma operación, pero deberá buscar las alertas que hayan sido generadas.

A medida que se ejecuten los comandos se irá almacenando en la base de datos toda la información referente a la modificación de los archivos y de las alertas para que esta pueda ser consultada por la herramienta y así mostrársela al usuario. Esta información será guardada en varias tablas permitiendo consultar el historial de un archivo para verificar cuantas modificaciones ha sufrido y mostrar varias características de las alertas para un mejor entendimiento por parte del usuario.

De los archivos modificados y de las alertas se mostrarán una serie de características, las cuales se explicarán a continuación:

- **Para los archivos modificados**
  - Archivo: hace referencia a la dirección donde se encuentra ubicado ese archivo.
  - Fecha: fecha en la que se modificó el archivo.
  - Tamaño: tamaño actual que tiene archivo representado en kilobyte (Kb).
  - Suma de verificación MD5: se refiere a uno de los algoritmos criptográficos que usa Ossec para comprobar la integridad de un archivo, mostrando solamente su codificación MD5.
  - Suma de verificación SHA1: familia de funciones hash utilizada en la criptografía que Ossec también emplea para la integridad de los archivos mostrando solamente su respectiva codificación SHA1.
- **Para las alertas**
  - Fecha: fecha en la que se generó la alerta.
  - ID de la regla: identificador de la regla que corresponde a la alerta generada.
  - Nivel: nivel perteneciente a la regla.

- Ubicación: hace referencia a la ubicación del error generado en la alerta o al módulo de Ossec al cual pertenece la alerta.
- Descripción: breve descripción de la alerta haciendo énfasis en el archivo al cual pertenece la alerta.
- Información: especifica el tipo de alerta que se generó.

2.4 Requisitos funcionales y no funcionales

La ingeniería de requerimientos agrupa un conjunto de tareas que permiten al analista de desarrollo plasmar con claridad los requerimientos realizados por el cliente de una manera clara, la cual permite realizar un análisis de factibilidad de lo que se pretende desarrollar. Además, debe comprobar los requerimientos y administrarlo durante el desarrollo.

2.4.1 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. Para la propuesta de solución se definen los siguientes requisitos funcionales:

- **RF1** Listar integridad de los directorios. Permite al administrador listar aquellos archivos que han sufrido alguna modificación.
- **RF2** Filtrar integridad de los directorios. Permite al administrador filtrar los archivos por su dirección, tamaño y por un rango de fecha.
- **RF3** Exportar integridad de los directorios. Permite al administrador exportar los archivos a formatos PDF, XLS y DOC.
- **RF4** Eliminar integridad de los directorios. Permite al administrador eliminar los archivos modificados.
- **RF5** Mostrar integridad de los directorios. Permite al administrador mostrar los archivos modificados especificando, además de los elementos expuestos en el listado, su codificación MD5 y SHA1.
- **RF6** Listar alertas. Permite al administrador listar las alertas generadas por Ossec.

- **RF7** Filtrar alertas. Permite al administrador filtrar las alertas por la descripción y por un rango de fecha.
- **RF8** Exportar alertas. Permite al administrador exportar las alertas a formatos PDF, XLS y DOC.
- **RF9** Eliminar alertas. Permite al administrador eliminar las alertas generadas por Ossec.
- **RF10** Mostrar alertas. Permite al administrador mostrar las alertas especificando, además de los elementos expuestos en el listado, el ID de la regla y su nivel correspondiente.

2.4.2 Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema, incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Estos no se refieren a las funcionalidades que proporciona el sistema, sino a las propiedades como restricciones del entorno o de la implementación.

- Portabilidad

RNF 1 Ambiente

El sistema debe ser instalado en un entorno con:

- Sistema operativo: Distribución de CentOS última versión estable.
- Servidor de bases de datos relacional PostgreSQL 9.4.x con memoria RAM: 16 GB, disco Duro: 500 GB y microprocesador: 6 x 800 Ghz.
- Servidor de aplicaciones Ngnix: 2.7.x con memoria RAM: 16 GB, disco Duro: 500 GB y microprocesador: 6 x 800 Ghz.
- Servidor de reportes: Según especificaciones del Generador de Reportes.
- Servidor de réplicas entre nodos: CentOS, PostgreSQL 9.4.x con memoria RAM: 8 GB, disco Duro: 500 GB y microprocesador: 3 x 800 Ghz.
- Red de cable para todos los servidores: 10Mbytes/100Mbytes/1000Mbytes.

El sistema será accesible desde estaciones de trabajo de escritorio, laptop y tablets. Estos deberán contar con un navegador web moderno (Navegadores web: Firefox (v10.x en adelante), Chrome (v20.x en adelante) y Opera (v10.x en adelante).

- Seguridad

RNF 2 Acceso restringido

- Garantizar la protección de información de accesos no autorizados.
- Mantener el sistema disponible evitando que los mecanismos de seguridad impidan el acceso a la información requerida por los usuarios autorizados.

- Confiabilidad

RNF 3 Tolerancia a fallas

Si se interrumpe la comunicación con el servidor se activa la comunicación con algunos de los restantes nodos disponibles, los cuales tendrán la misma información que el nodo con fallas debido al sistema de réplicas.

2.5 Organización del proceso de entrega

Para una mejor organización durante el ciclo de vida del software se decide definir un plan de entregas por etapas y se realizó una estimación en semanas de cuánto tiempo durará este plan de entregas.

2.5.1 Entregas por etapas

Las entregas por etapas engloban la planificación detallada de un período corto de tiempo dentro del proyecto, lo que hace referencia a una iteración. Entre otras funciones, debe identificar las actividades, los riesgos involucrados y los artefactos a actualizar o crear (50). Estas etapas son de suma importancia no solo porque representan la implementación de una parte de los requisitos funcionales sino por las dificultades que se pueden presentar una vez se realicen las pruebas pertinentes.

Cada requisito funcional contiene tareas específicas de aceptación. De igual modo, para cada requisito se instauran las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en que se desarrollan y al final de cada uno de los ciclos siguientes, para comprobar que las siguientes iteraciones no han afectado a las anteriores. Una vez se termine una iteración y no se encuentren errores es un modo de prueba del avance obtenido.

Etapa 1:

En esta etapa serán implementados una parte de los requisitos de mayor prioridad pertenecientes a los primeros cinco que hacen énfasis en la integridad de los directorios. Una vez terminado se podrán listar, mostrar, eliminar, filtrar y exportar aquellos archivos que hayan sufrido alguna

modificación o aquellos que hayan sido añadidos.

Etapa 2:

Esta etapa tiene como objetivo implementar los requisitos pertenecientes a las alertas generadas por Ossec, las cuales también son de prioridad alta y describen detalladamente los últimos archivos modificados mostrando información específica de su modificación o algún error del servidor web. Cuando se termine la implementación se podrá listar, mostrar, eliminar, filtrar y exportar todas las alertas generadas por Ossec.

2.5.2 Estimación de entregas por etapa (Semanas)

Tabla 1. Estimación de entregas por etapas.

Etapas	Orden de los requisitos a implementar	Tiempo de duración (semanas)
1	**RF1**, **RF2**, **RF3**, **RF4**, **RF5**	4
2	**RF6**, **RF7**, **RF8**, **RF9**, **RF10**	4

2.6 Especificación de requisitos

La especificación de requisitos muestra la respuesta que debe tener el sistema, la prioridad y complejidad las cuales son clasificadas como Alta y Media dependiendo del orden de prioridad y su complejidad. En la siguiente tabla se muestran las especificaciones de requisitos pertenecientes a los requisitos funcionales listar integridad de los directorios y mostrar integridad de los directorios.

- **RF1. Listar integridad de los directorios.**

Leyenda utilizada en la tabla: Prioridad para el cliente (PRC), Complejidad (C), Referencia cruzada (RC).

Tabla2. Especificación de requisitos. Requisito funcional listar integridad de los directorios.

No.	Nombre	Descripción	PRC	C	RC
RF 1	Listar integridad de los directorios	El sistema debe permitir listar la integridad de los directorios. En el listado se deben mostrar los siguientes datos: - Archivo: contiene la dirección donde se encuentra ubicado el archivo que se modificó. Ejemplo: /var/ossec/etc/ossec.conf - Tamaño (Kb): especifica el tamaño que tiene el archivo una vez que este ha sido modificado. Ejemplo: 453. - Fecha de modificación: contiene la fecha en que se modificó el archivo. Ejemplo: lunes, 3 de abril de 2017.	Alta	Alta	N/A

- **RF5. Mostrar integridad de los directorios.**

Tabla 3. Especificación de requisitos. Requisito funcional mostrar integridad de los directorios.

No.	Nombre	Descripción	PRC	C	RC
RF 5	Mostrar integridad de los directorios	El sistema debe mostrar la integridad del archivo especificando: - Suma de verificación MD5: especifica la codificación MD5 que tiene el archivo. Ejemplo: dd607d75bb2fa0366495fb496b28e3d36. - Suma de verificación SHA1: especifica la codificación MD5 que tiene el archivo. Ejemplo: c75bf26c7e8a4368dd0daa9de51c1d5b1be29113.	Alta	Alta	N/A

		- Tamaño (Kb): especifica el tamaño tiene el archivo una vez que este ha sido modificado. Ejemplo: 453. - Fecha de modificación: contiene la fecha en que se modificó el archivo. Ejemplo: 3-Abr-2017.			

2.7 Descripción de requisitos

A continuación, se presentan la descripción de requisitos correspondiente a los requisitos funcionales listar integridad de los directorios y mostrar integridad de los directorios además de los prototipos desarrollados para cada requisito. Para tener información sobre las demás descripciones de requisitos consultar el Anexo I.

- **RF1. Listar integridad de los directorios.**

Tabla 4. Descripción del requisito funcional listar integridad de los directorios.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Listar integridad de los directorios	
1.	El usuario selecciona la opción Listar integridad de los directorios.
2.	El sistema debe permitir listar la integridad de los directorios existentes de forma ascendente o descendente según la dirección donde se encuentra ubicado el archivo, su tamaño y la fecha de modificación mostrando los siguientes datos: • Dirección del archivo. • Tamaño (Kb). • Fecha de modificación.
3.	Concluye así el requisito.
Pos-condiciones	
1.	Se listó la integridad de los directorios satisfactoriamente.

Flujos alternativos			
1.	N/A		
Pos-condiciones			
1.	NA		
Validaciones			
1.	NA		
Conceptos	NA	NA	
Requisitos especiales	NA		
Asuntos pendientes	NA		

- **Prototipo de interfaz gráfica de usuario**

Ilustración 2. Prototipo de interfaz gráfica de usuario. Requisito funcional listar integridad de los directorios.

- **RF5. Mostrar integridad de los directorios.**

Tabla 5. Descripción del requisito funcional mostrar integridad de los directorios.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Mostrar datos de los archivos modificados	
1.	El usuario selecciona la opción Mostrar del listado de archivos modificados.

2.	El sistema debe mostrar los siguientes datos de los archivos modificados: • Suma de verificación MD5. • Suma de verificación SHA1. • Tamaño. • Fecha de modificación. Y permite, además, realizar las siguientes opciones: • Regresar al listado		
3.	Concluye así el requisito.		
Pos-condiciones			
1	Se listó la integridad de los directorios satisfactoriamente.		
Flujos alternativos			
1	N/A		
Pos-condiciones			
1.	NA		
Validaciones			
1.	NA		
Conceptos		NA	NA
Requisitos especiales		NA	
Asuntos pendientes		NA	

- **Prototipo de interfaz gráfica de usuario**

Ilustración 3. Prototipo de interfaz gráfica de usuario. Requisito funcional mostrar integridad de los directorios.

2.8 Modelo de análisis

El propósito fundamental del análisis es analizar los requisitos con mayor profundidad que los que se tienen como resultado de la captura de requisitos. Este modelo se describe utilizando el lenguaje de los desarrolladores y por tanto puede ser utilizado para razonar el funcionamiento interno del sistema. Es utilizado fundamentalmente por los desarrolladores para comprender cómo debería ser diseñado e implementado el sistema. Esboza como se debería llevar a cabo la funcionalidad dentro del sistema, sirve como una primera aproximación al diseño (51).

2.8.1 Diagramas de clases del análisis

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Las clases del análisis siempre encajan en uno de los tres estereotipos básicos: clase de interfaz, de control y de entidad (51).

Clases de Interfaz (CI): se utilizan para modelar las interacciones entre el sistema y sus actores. Representan abstracciones de ventanas, formularios, paneles, interfaces de comunicaciones, de impresoras, sensores y terminales.

Ilustración 4. Prototipo que representa la CI.

Clases de Entidad (CE): se utilizan para modelar la información que posee larga vida y que es a menudo persistente. Suelen mostrar una estructura de datos lógica y contribuyen a comprender de que información depende el sistema

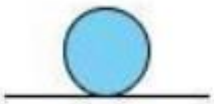

Ilustración 5. Prototipo que representa la CE.

Clases de Control (CC): representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto, además, el manejo y coordinación de las acciones a otros objetos, es decir, objetos de interfaz y de entidad.

Ilustración 6. Prototipo que representa la CC.

A continuación, se muestra el diagrama de clase del análisis correspondiente a los requisitos funcionales listar integridad de los directorios y mostrar integridad de los directorios. Para el estudio de los demás diagramas de clases del análisis, remitirse al Anexo II.

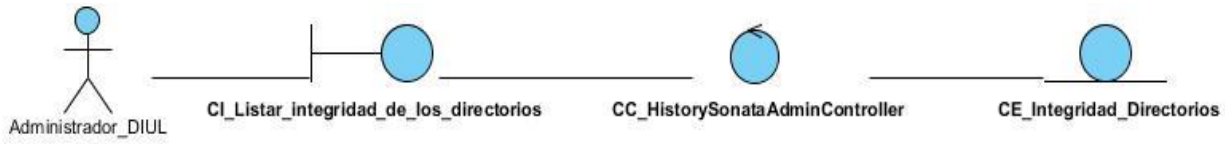

Ilustración 7. Diagrama de clases del análisis del requisito funcional listar integridad de los directorios.

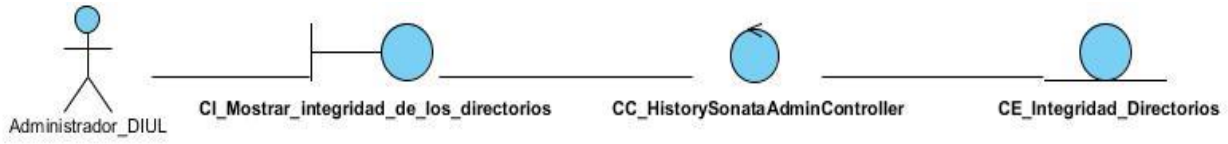

Ilustración 8. Diagrama de clases del análisis del requisito funcional mostrar integridad de los directorios.

2.8.2 Diagramas de colaboración del análisis

Los diagramas de colaboración muestran las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces. El nombre de un mensaje debería denotar el propósito del objeto que invoca en la interacción con el objeto invocado (51).

A continuación, se muestra el diagrama de colaboración del análisis correspondiente a los requisitos funcionales listar integridad de los directorios y mostrar integridad de los directorios. Para el estudio de los demás diagramas de colaboración del análisis, remitirse al Anexo III.

Ilustración 9. Diagrama de colaboración del análisis del requisito funcional listar integridad de los directorios.

Ilustración 10. Diagrama de colaboración del análisis del requisito funcional mostrar integridad de los directorios.

2.9 Patrón arquitectónico Modelo Vista Controlador

El patrón Modelo Vista Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. EL patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en reducido espacio de tiempo. A partir del uso *de frameworks* basados en este patrón se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores (52).

Este patrón de arquitectura presenta varias ventajas:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.
- Interfaz de Programación de Aplicaciones (API) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- Conexión entre el Modelo y sus Vistas dinámica; se produce en tiempo de ejecución, no en tiempo de compilación (52).

Definiendo este patrón por partes el Modelo es el objeto que representa los datos del programa, maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo.

Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo (52).

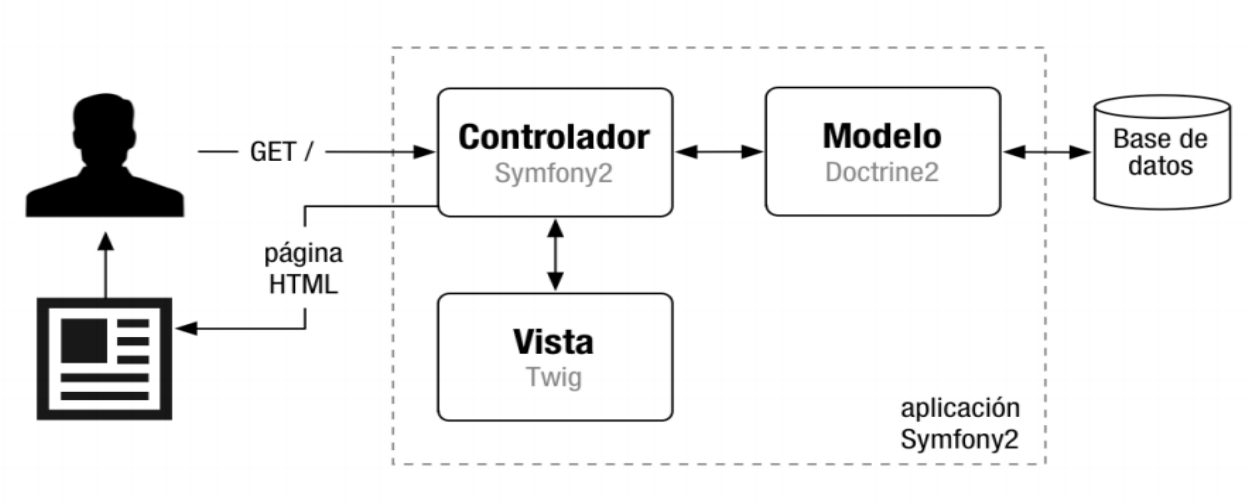

Ilustración 11. Interrelación entre los elementos del patrón MVC en Symfony2.

2.10 Modelo de diseño

En el diseño se modela el sistema y se encuentra la forma de que soporte todos los requisitos incluyendo los no funcionales y otras restricciones. Permite descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. Da una forma al sistema mientras que intenta preservar la estructura definida por el modelo de análisis (53).

2.10.1 Aplicaciones de los 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. Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que puede utilizarse un millón de veces sin tener que hacer dos veces lo mismo (54).

Patrones GRASP

Los Patrones Generales de Software para Asignar Responsabilidades (GRASP, por sus siglas en inglés), son parejas de problema y solución, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Los patrones GRASP utilizados en la investigación son:

- **Experto:** el patrón experto posibilita una adecuada asignación de responsabilidades facilitando la comprensión del sistema, su mantenimiento y adaptación a los cambios con

reutilización de componentes (55). Symfony2 incluye la librería ORM Doctrine como interfaz de comunicación con las clases del modelo permitiendo agrupar toda la lógica de los datos y generar clases para manipular la información de las entidades de la base de datos. Estas clases que se generan cuentan con toda la información de la entidad que representan y sus funcionalidades, además se encuentran bajo el nombre *entity_nameRepository*. En la herramienta desarrollada se evidencia con la entidad *OssecHistory* y su respectivo *OssecHistoryRepository.*

- **Bajo acoplamiento:** el patrón bajo acoplamiento es una medida de la fuerza con que una clase se relaciona con otras, porque las conoce y recurre a ellas; una clase con bajo acoplamiento no depende de muchas otras, mientras que otra con alto acoplamiento presenta varios inconvenientes: es difícil entender cuando está aislada, es ardua de reutilizar porque requiere la presencia de otras clases con las que esté conectada y es cambiante a nivel local cuando se modifican las clases afines (55). Se evidencia a través las clases que se encuentran en el modelo, las cuales no representan asociaciones con las de las vistas o las controladoras.

- **Alta cohesión:** El patrón alta cohesión es una medida que determina cuán relacionadas y adecuadas están las responsabilidades de una clase, de manera que no realice un trabajo colosal; una clase con baja cohesión realiza un trabajo excesivo, haciéndola difícil de comprender, reutilizar y conservar (55). Se evidencia mediante la implementación de la clase controladora *AlertsSonataAdminController.php*, la cual está formada por diferentes funcionalidades en la que se encuentra el método *showAction()*, en donde se realiza la búsqueda de datos sobre la entidad *OssecAlerts***.**

- **Creador:** el patrón Creador aporta un principio general para la creación de objetos, una de las actividades más frecuentes en programación. El propósito fundamental de este patrón consiste en guiar la asignación de responsabilidades relacionada con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos (55). Su utilización se evidencia mediante las clases controladoras las cuales crean instancias de las entidades del modelo.

Patrones GOF

Los patrones de diseño GOF describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objeto. Eric Braude define los patrones de diseño como *"combinaciones de componentes, casi siempre clases y objetos que por experiencia se sabe que*

resuelven ciertos problemas de diseño comunes". En términos generales es posible decir que un patrón de diseño es una solución a un problema recurrente en el diseño de software (56).

- **Observer (Observador):** define de una a muchas dependencias entre objetos de forma que cuando un objeto cambia de estado, todos sus dependientes son notificados y actualizadas de forma automática (56). Su utilización se evidencia a través de la entidad *OssecDirectoryIntegrity* que se encuentra relacionada con la entidad *OssecHistory* por lo que si se modifica un archivo en la primera cambiará todo su historial.

2.10.2 Diagrama de clases del diseño

Los diagramas de clases del diseño (DCD), corresponden a los requisitos del sistema y muestran el diseño estático a través de las clases, subsistemas y relaciones que participan. A continuación, se presenta el diagrama de clases del diseño correspondiente a los requisitos funcionales listar integridad de los directorios y mostrar integridad de los directorios. Para el estudio del diagrama de clases del diseño restante remitirse al Anexo IV.

Para una mejor compresión de los diagramas, se describe DCD asociado al requisito funcional: listar integridad de los directorios. En el mismo las clases son agrupadas por paquetes basándose en la estructura del patrón MVC. El paquete Vista agrupa las interfaces asociadas al usuario, mientras el paquete Controlador contiene una Página Servidora (SP) la cual mediante los métodos implementados accede a las clases del paquete del Modelo para dar respuestas a las peticiones del usuario. El paquete Componentes Symfony contiene las clases donde se implementan los métodos que son llamados en la clase que se encuentra en el paquete Controlador, puesta esta clase a través de la herencia hace uso de estos métodos. Por último, se tiene el ORM Doctrine que es utilizado para la comunicación de las clases del modelo con las entidades de la base de datos.

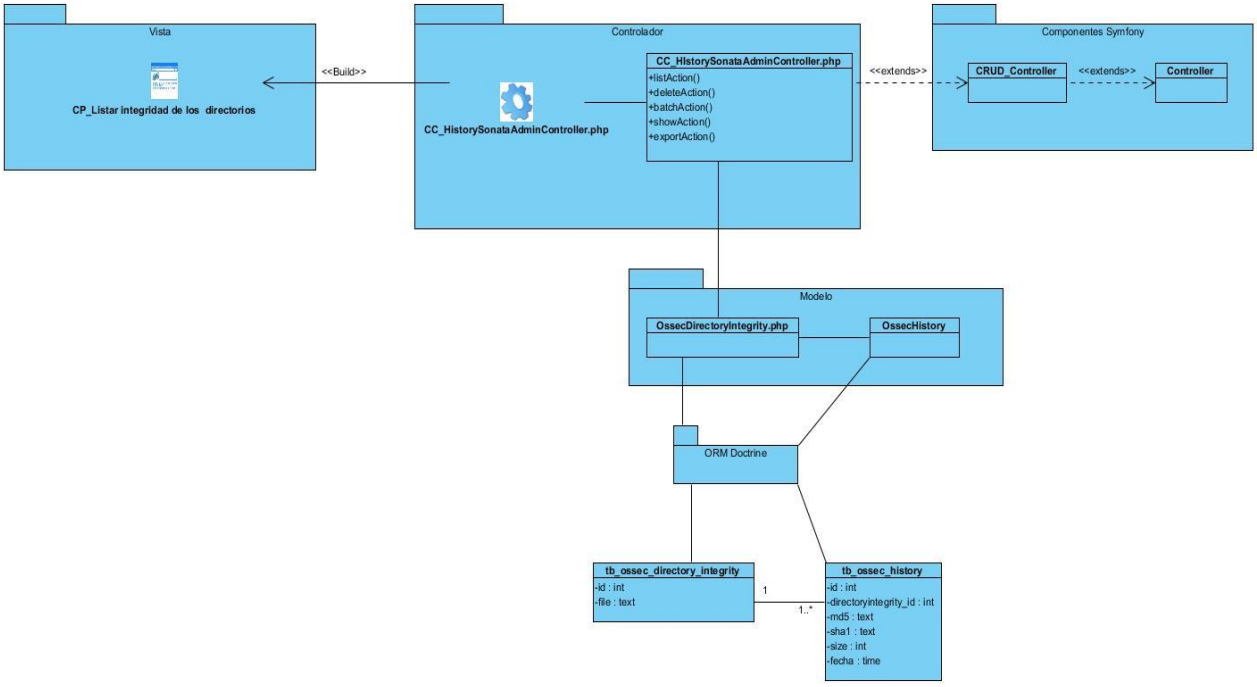

Ilustración 12. Diagrama de clases del diseño del requisito funcional listar integridad de los directorios.

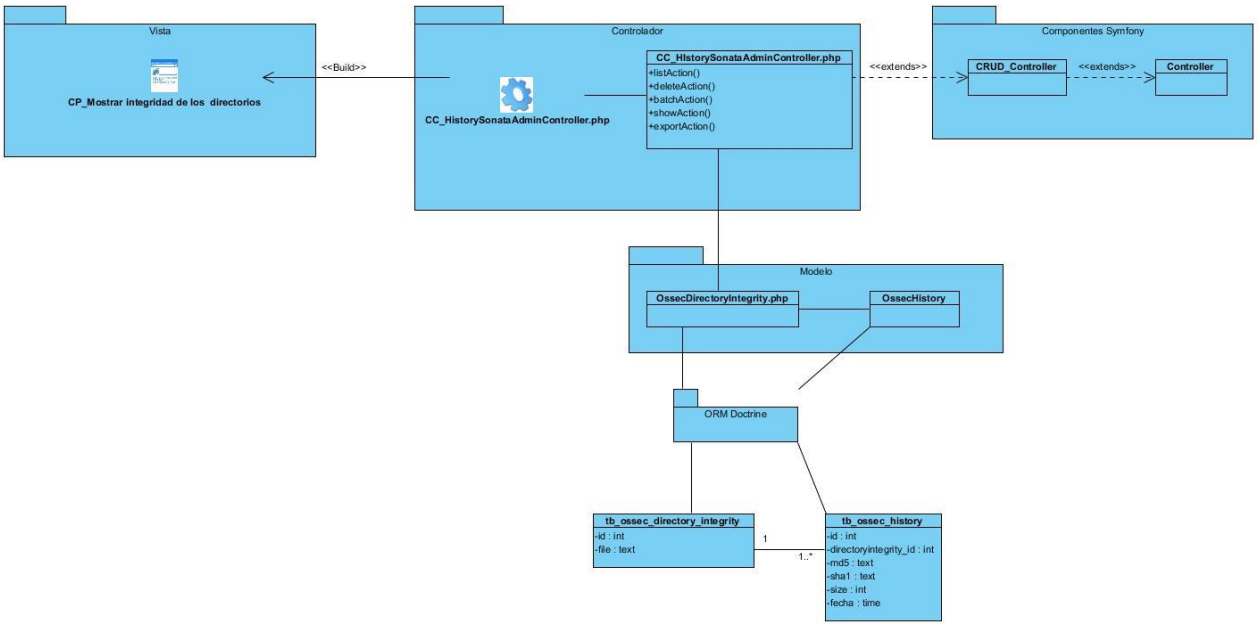

Ilustración 13. Diagrama de clases del diseño del requisito funcional mostrar integridad de los directorios.

2.10.3 Diagrama de secuencia del diseño

Los diagramas de secuencia muestran las interacciones entre objetos mediante transferencia de mensajes entre objetos o subsistemas. El nombre del mensaje debería indicar una operación del objeto que recibe la invocación o de una interfaz que el objeto proporciona (53).

A continuación, se presenta los diagramas de secuencia de los requisitos funcionales listar integridad de los directorios y mostrar integridad de los directorios, el resto de los diagramas se encuentran en el Anexo V.

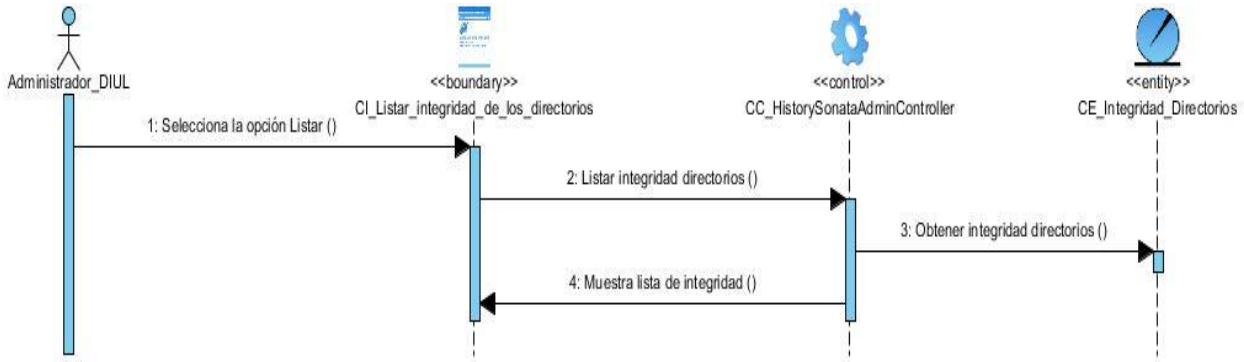

Ilustración 14. Diagrama de secuencia del diseño del requisito funcional listar integridad de los directorios.

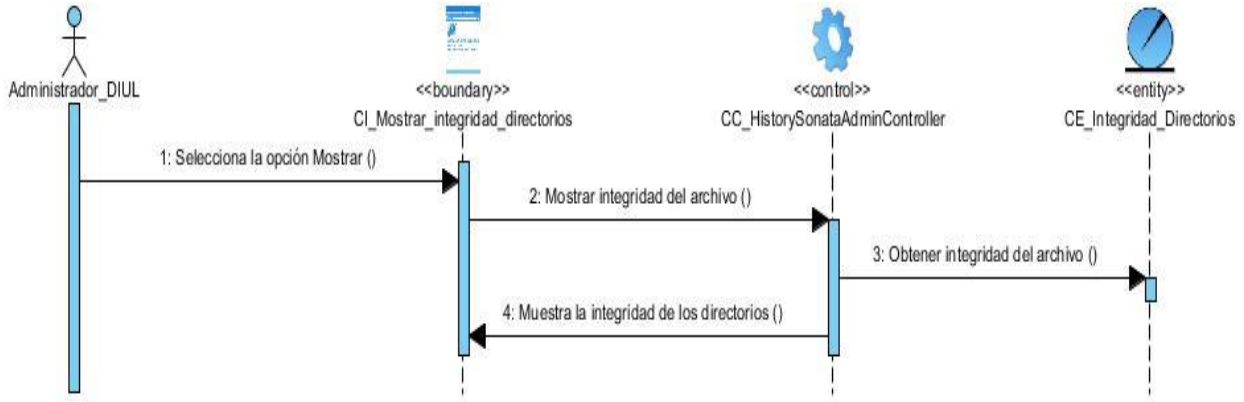

Ilustración 15. Diagrama de secuencia del diseño del requisito funcional mostrar integridad de los directorios.

2.11 Diseño de la base de datos

Uno de los modelos más utilizados para diseñar base de datos es el modelo entidad- relación, ya que permite una definición clara y concisa de los esquemas conceptuales y de su visión. Este modelo se encuentra basado en dos conceptos: las entidades, que son objetos sobre los cuales se desea guardar información y las relaciones, que constituyen asociaciones entre entidades (57).

A continuación, se presenta el modelo entidad - relación referente a la base de datos y la descripción de todas las tablas que lo componen.

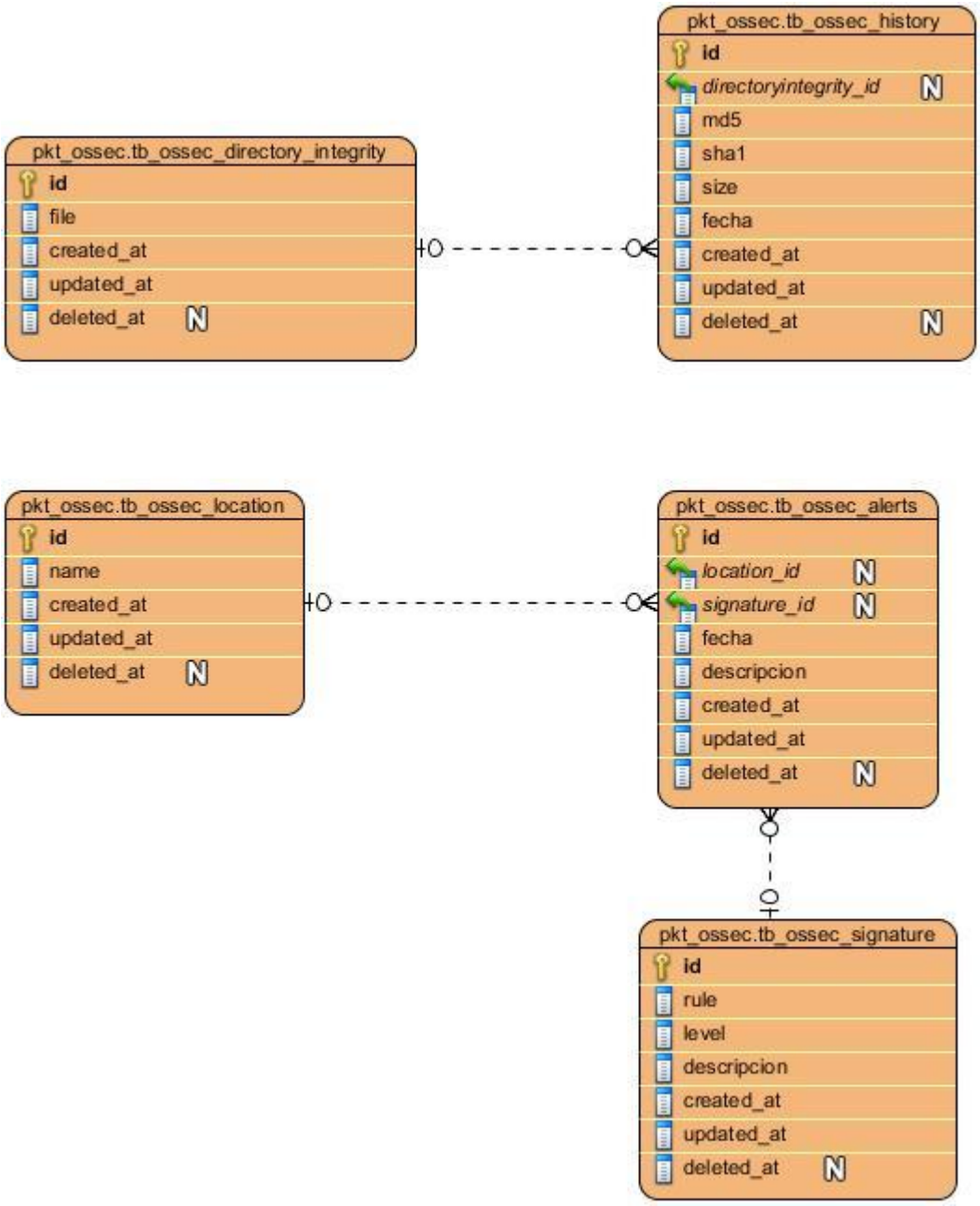

Ilustración 16. Diseño de la base de datos.

- **pkt_ossec.tb_ossec_directory_integrity**: en esta tabla se almacena la dirección donde se encuentra ubicado el archivo.
- **pkt_ossec.tb_ossec_history:** esta tabla almacena el historial de todos los archivos que se han modificado.

- **pkt_ossec.tb_ossec_alerts:** en esta tabla se almacena información referente a las alertas generadas por Ossec, de ellas se guardará su descripción y la fecha en que se generaron.

- **pkt_ossec.tb_ossec_location:** esta tabla almacena información referente a la ubicación del archivo por el cual se generó la alerta.

- **pkt_ossec.tb_ossec_signature:** en esta tabla se almacena la regla y el nivel correspondiente por el cual fue generado la alerta.

2.12 Conclusiones parciales

En este capítulo se reflejaron los principales conceptos asociados al modelo de dominio, así como su representación para facilitar su comprensión. Se definieron los requisitos funcionales y no funcionales que sustentan la propuesta de solución planteada. Se realizó la especificación de requisitos para varios requisitos con sus correspondientes descripciones. Se confeccionaron los respectivos diagramas pertenecientes al Análisis y al Diseño. Además, se definió como patrón arquitectónico el MVC y se diseñó el diagrama entidad-relación que define la estructura de la base de datos.

Capítulo 3: Implementación y prueba

3.1 Introducción

En este capítulo se muestra el diagrama de componentes, los estándares de codificación usados durante la implementación de la solución y también se incluyen los resultados de las pruebas realizadas para verificar la correcta implementación de todos los requisitos. Se describen las tres etapas llevadas a cabo durante el desarrollo del diseño, las cuales corresponden a todos los requisitos expuestos en el capítulo anterior.

3.2 Modelo de implementación

El modelo de implementación está compuesto por un conjunto de componentes y subsistemas que forman la parte física de la implementación de un sistema. Este describe como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado. El mismo se representa con un sistema de implementación que denota el subsistema de nivel superior del modelo (58).

3.2.1 Diagrama de componentes

Los diagramas de componentes son utilizados para mostrar los componentes de software y la relación existente entre ellos en un sistema. Estos tienen relaciones de traza con los elementos de modelo que implementan y también implementan varios elementos, por ejemplo, varias clases (58).

A continuación, se describen los estereotipos que fueron utilizados para representar dichos componentes:

- <<file>>: ficheros que contienen código fuente de la aplicación.
- <<entity>>: entidades de la aplicación.
- <<command>>: comandos utilizados para configurar Ossec.
- <<framework>>: framework de desarrollo utilizado por el proyecto SIGIES.

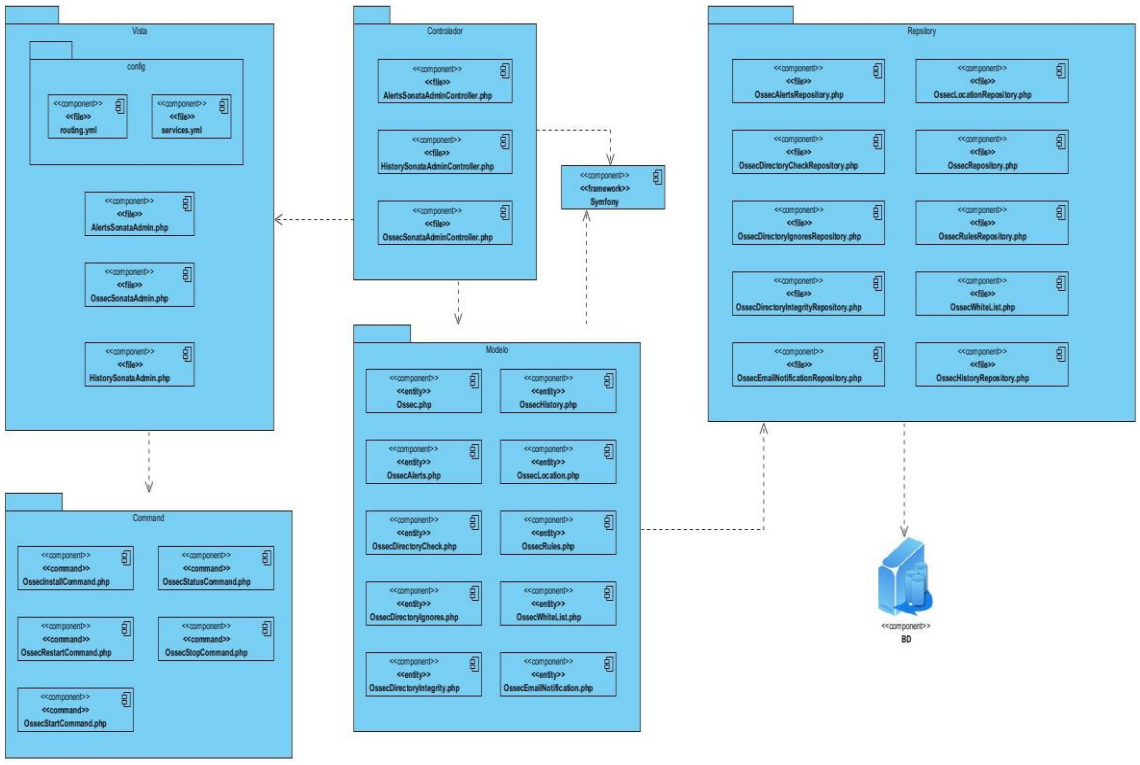

Ilustración 17. Diagrama de componentes.

3.3 Estándares de codificación

Los estándares de codificación abarcan todos los aspectos generales del código fuente de una aplicación. Su importancia radica en la legibilidad del código para que sea entendible por todos los programadores de un proyecto como si este hubiese sido escrito por una sola persona. Cuando se añade código nuevo o se realice mantenimiento del sistema el estándar de codificación debe sugerir como trabajar con el código ya existente.

En la implementación de la herramienta se utilizaron estándares de codificación tanto para Symfony (59) como para PHP (60), los cuales son los utilizados por el proyecto SIGIES. A continuación, se definen los mismos:

- **Symfony**

Estructura

- Se añade un solo espacio después del delimitador coma.
- Se añade una línea en blanco antes de las declaraciones *return*, a menos que el valor devuelto solo sea dentro de un grupo de declaraciones (tal como una declaración *if*).

- Se usan llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga.
- Se define una clase por archivo.
- Se declara las propiedades de las clases antes de los métodos.

Convenciones de nomenclatura

- Se utiliza mayúsculas, sin guiones bajos, en nombre de variable, función, método o argumentos.
- Se usa guiones bajos para nombre de opción y nombres de parámetro.
- Se utiliza caracteres alfanuméricos y guiones bajos para los nombres de archivos.

- **PHP**

Tamaño máximo de línea

- La longitud recomendable para una línea de código es de 80 caracteres. Esto significa que los desarrolladores deberían intentar mantener cada línea de su código por debajo de los 80 caracteres, siempre que sea posible. No obstante, líneas más largas pueden ser aceptables en algunas situaciones. El tamaño máximo de cualquier línea de código PHP es de 120 caracteres.

Convenciones de nombres

- Los nombres de clases pueden contener sólo caracteres alfanuméricos. Los números están permitidos en los nombres de clase, pero desaconsejados en la mayoría de casos. Las barras bajas (_) están permitidas solo como separador de ruta (el archivo " Zend/Db/Table.php " debe apuntar al nombre de clase " Zend_Db_Table ").

Nombres de archivos

- Para los archivos, sólo caracteres alfanuméricos, barras bajas (_) y guiones (-) están permitidos. Los espacios en blanco están estrictamente prohibidos.
- Cualquier archivo que contenga código PHP debe terminar con la extensión ".php ", con la excepción de los scripts de la vista.

Funciones y métodos

- Los nombres de funciones pueden contener únicamente caracteres alfanuméricos. Los guiones bajos (_) no están permitidos. Los números están permitidos en los nombres de función, pero no se aconseja en la mayoría de los casos.

- Los nombres de funciones deben empezar siempre con una letra minúscula. Cuando un nombre de función consiste en más de una palabra, la primera letra de cada nueva palabra debe estar en mayúsculas.

Variables

- Los nombres de variables pueden contener caracteres alfanuméricos. Las barras bajas (_) no están permitidas. Los números están permitidos en los nombres de variables, pero no se aconseja en la mayoría de los casos.

Constantes

- Las constantes pueden contener tanto caracteres alfanuméricos como barras bajas (_). Los números están permitidos. Todas las letras pertenecientes al nombre de una constante deben aparecer en mayúsculas.

Declaración de clases

- Las Clases deben ser nombradas de acuerdo a las convenciones de nombres. La llave "{" deberá escribirse siempre en la línea debajo del nombre de la clase.

Declaración de funciones y métodos

- Las funciones deben ser nombradas de acuerdo a las convenciones de nombrado establecido.

- Los métodos dentro de clases deben declarar siempre su visibilidad usando un modificador *private*, *protected* o *public*.

3.4 Propuesta de solución por etapas

Etapa 1

Para dar solución a esta etapa perteneciente a una parte de los requisitos de mayor complejidad, relacionados con la integridad de los directorios, se hizo necesario le revisión de la interfaz gráfica Ossec-wui la cual fue desarrollada para visualizar los resultados que arroja el monitoreo realizado por Ossec. El mismo lista en su vista principal las alertas que se generan describiendo cada una de ellas, mientras en otra vista muestra los directorios que han sido modificados. De este módulo se crearon varias clases y entidades principales, en algunas de ellas utilizando funciones y métodos de Ossec-wui.

- Se creó la clase *OsLibSyscheck.php* encargada de monitorizar la integridad de los archivos, para ello definen varias funciones para analizar los ficheros de los directorios

especificados por el usuario. Para conocer cuando se modifican los archivos se utiliza la suma de verificación MD5 y SHA1 de cada fichero que son mostrados en la lista que se genera notificando el cambio en los archivos.

- Se creó la clase *OsLibHandle.php* utilizada como referencia para indicar la dirección donde se encuentra instalado Ossec. A partir de esta dirección se especifican los subdirectorios donde se encuentran almacenados las alertas y los ficheros modificados.
- Se crearon las entidades *OssecDirectoryIntegrity* y *OssecHistory* encargadas de manejar la información de todos los archivos. En la primera se maneja la dirección del archivo en específico y en la segunda información respecto a su código *hash* MD5 y SHA1, la fecha en que se realiza la modificación y el tamaño del archivo. De la primera entidad hacia la segunda *existe una relación de OneToMany* (uno a muchos) mientras que al revés la relación es de *ManyToOne* (muchos a uno), así podremos tener un historial de un archivo en cuanto a modificaciones realizadas.

Etapa 2

En esta etapa que hace referencia a los requisitos restantes de complejidad alta, los cuales están relacionados con las alertas y se utilizó al igual que en la etapa uno la interfaz gráfica Ossec-wui. De ella se tomaron varios ficheros necesarios para realizar el análisis del archivo donde se almacenan las alertas y se definieron varias clases esenciales para realizar este proceso.

- Se creó la clase *OsLibAlerts.php* que cuenta con varias funciones principales encargadas de analizar sintácticamente el archivo *alerts.log* donde se van almacenando todas las alertas y de crear la lista de alertas que serán mostradas al usuario con la nueva herramienta.
- Se creó la entidad *OssecAlerts.php* que será la encargada de llevar toda la información relacionada con las alertas. La fecha, la ubicación y la descripción son algunos de los elementos que manejará esta clase por cada alerta generada.

3.5 Pruebas de software

La fase de pruebas es una de las más costosas del ciclo de vida software. En sentido estricto, deben realizarse pruebas de todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación. Obviamente, se aplican diferentes técnicas de prueba a cada tipo de producto software (61). Para

aplicar pruebas a un software existen métodos, estos se definen en pruebas de caja blanca y pruebas de caja negra.

3.5.1 Pruebas de caja blanca

Las pruebas de caja blanca o estructurales, como también se les denomina, realizan un seguimiento del código fuente según va ejecutando los casos de prueba, de manera que se determinen las instrucciones o bloques de código donde existan errores (61). Estas pruebas excluyen detalles referidos a datos de entrada o salida del programa, pues inspeccionan sobre la estructura interna del código.

A través de ella se pueden generar diseños de casos de prueba que (62):

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de los datos para asegurar su validez.

3.5.2 Pruebas de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, pretenden examinar el programa para que cuente con todas las funcionalidades analizando los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos. Al ejecutar las pruebas se desarrollan casos de pruebas reales cada condición y se analizan los resultados que arroja el sistema para cada uno de los casos. En esta estrategia se verifica el programa considerándolo una caja negra y no se hacen en base al código, sino a la interfaz (62).

Según Pressman existen varios métodos para realizar las pruebas de caja negra, entre los que se encuentran (51):

- **Método de partición equivalente:** divide el dominio de entradas de un programa en clases de datos, a partir de las cuales pueden derivarse casos de prueba.
- **Método del análisis de valores límites:** es un método que complementa a la partición equivalente, en lugar de seleccionar cualquier elemento de una clase de equivalencia, lleva a la elección de casos de prueba en los extremos de la clase.

- **Método de pruebas de comparación:** se emplean en aplicaciones críticas en que se desarrollen versiones de software independiente, empleando como entrada casos de prueba diseñados mediante algún otro método de caja negra.
- **Método de prueba de la tabla ortogonal:** se aplica en problemas donde el dominio de entrada es relativamente pequeño, el método resulta útil para encontrar fallas de región.

Para verificar el correcto funcionamiento de la nueva herramienta se realizaron pruebas de caja negra, aplicando el método de partición equivalente.

Tomando como base todos los requisitos funcionales se generaron los Diseños de Casos de Prueba (DCP) correspondientes, se elaboró cada caso de prueba identificando los principales escenarios en dependencia de cada acción del actor.

A continuación, se muestra los DCP asociados a los requisitos funcionales listar integridad de los directorios y mostrar integridad de los directorios.

DCP del requisito funcional: listar integridad de los directorios.

Descripción general: El requisito comienza cuando el usuario accede a la interfaz Ossec y selecciona la opción History. El sistema muestra un listado con todos aquellos archivos que han sido modificados de los directorios especificados en la configuración.

Tabla 6. DCP del requisito funcional listar integridad de los directorios.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Listar archivos modificados.	El usuario selecciona la opción History.	El sistema permite listar los archivos que han sufrido alguna modificación de los directorios especificados en la configuración de Ossec. En el listado se muestran los siguientes datos: - Archivo. - Tamaño (Kb). - Fecha. Y permite además, realizar las siguientes opciones: - Mostrar - Eliminar	General/Seguridad/Ossec/History

		- Filtros - Exportar	
EC 1.2 Opción Mostrar.	El usuario selecciona la opción Mostrar.	El sistema permite visualizar los datos de los archivos modificados. Ver: DCP_5_Mostrar_integridad_de_los_directorios.ods	General/Seguridad/Ossec/History/Mostrar
EC 1.3 Opción Eliminar.	El usuario selecciona la opción Eliminar.	El sistema permite eliminar los datos de los archivos modificados. Ver: DCP_4_Eliminar_integridad_de_los_directorios.ods	General/Seguridad/Ossec/History/Eliminar
EC 1.4 Opción Filtros.	El usuario selecciona la opción Filtros.	El sistema permite filtrar los datos de los archivos modificados. Ver: DCP_2_Filtrar_integridad_de_los_directorios.ods	General/Seguridad/Ossec/History/Filtros
EC 1.5 Opción Exportar datos .	El usuario selecciona la opción Exportar.	El sistema permite exportar los datos de los archivos modificados. Ver: DCP_3_Exportar_integridad_de_los_directorios.ods	General/Seguridad/Ossec/History/Exportar

DCP del requisito funcional: mostrar integridad de los directorios.

Descripción general: El requisito comienza cuando el usuario accede a la interfaz Ossec y selecciona la opción Mostrar de la lista de archivos modificados. El sistema muestra los elementos correspondientes del archivo especificado.

Tabla 7. DCP del requisito funcional mostar integridad de los directorios.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción de mostrar los datos de los archivos modificados.	Selecciona la opción de mostrar los datos de un archivo y consulta sus datos.	Muestra los siguientes datos del archivo: - Suma de verificación MD5. - Suma de verificación SHA1. - Tamaño. - Fecha de modificación.	General/Seguridad/Ossec/History/Mostrar

		Permite además: - Eliminar los datos de la institución. - Regresar al listado de archivos modificados.	
EC 1.2 Opción de regresar al listado de los archivos modificados.	El usuario selecciona la opción de Regresar al listado.	Regresa al listado de archivos modificados.	General/Seguridad/Ossec/History/ Mostrar/Regresar al listado
EC 1.3 Opción de eliminar el elemento.	Selecciona la opción de Eliminar.	El sistema brinda la posibilidad de eliminar la institución. Ver: DCP_4_Eliminar_integridad_de_los_directorios.ods	General/Seguridad/Ossec/History/ Eliminar

3.5.3 Pruebas de regresión

Las pruebas de regresión consisten en la repetición selectiva de pruebas para detectar fallos introducidos durante la modificación de un sistema o componente de un sistema. Se efectúan para comprobar que los cambios no han originado efectos adversos no intencionados o que se siguen cumpliendo los requisitos especificados (63).

Como parte de la estrategia de prueba sobre la solución desarrollada se efectuaron al inicio de las iteraciones 2, 3 y 4 las pruebas de regresión, que estuvieron dirigidas a detectar no conformidades (NC) que quedaran pendientes de la iteración anterior a estas. De esta manera se garantiza que se puedan iniciar de manera satisfactoria las iteraciones mencionadas teniendo en cuenta que fueron corregidas las NC pendientes de las iteraciones anteriores.

3.5.4 Pruebas de aceptación

Las pruebas de aceptación se caracterizan por la participación activa del usuario, que debe ejecutar los casos de prueba ayudado por miembros del equipo de pruebas. Están enfocadas a probar los requisitos de usuario, o mejor dicho a demostrar que no se cumplen los requisitos, los criterios de aceptación o el contrato. Si no se consigue demostrar esto el cliente deberá aceptar el producto (63).

Para verificar el correcto funcionamiento de la aplicación teniendo en cuenta los requisitos funcionales definidos, el cliente en conjunto con el equipo de desarrollo realizaron las pruebas de aceptación enfocadas al cumplimiento de cada requisito. Como resultado de esta revisión se detectó un error en el RF2, filtrar integridad de los directorios, el cual fue corregido en breve tiempo.

3.5.5 Resultados de las pruebas

Una vez concluida la implementación de la herramienta se detectaron un conjunto de (NC), las cuales fueron clasificadas en significativas y no significativas. En la tabla siguiente se muestran los resultados obtenidos luego de aplicadas todas las pruebas:

Tabla 8. Cantidad de no conformidades por iteración.

Clasificación de las no conformidades	**1ra Iteración**	**2da Iteración**	**3ra Iteración**	**4ta Iteración**
Significativas	6	7	3	0
No significativas	5	4	4	2
Total	11	11	7	2

Para un mejor entendimiento de los resultados, se muestra en el gráfico siguiente la cantidad de no conformidades detectadas por cada una de las iteraciones.

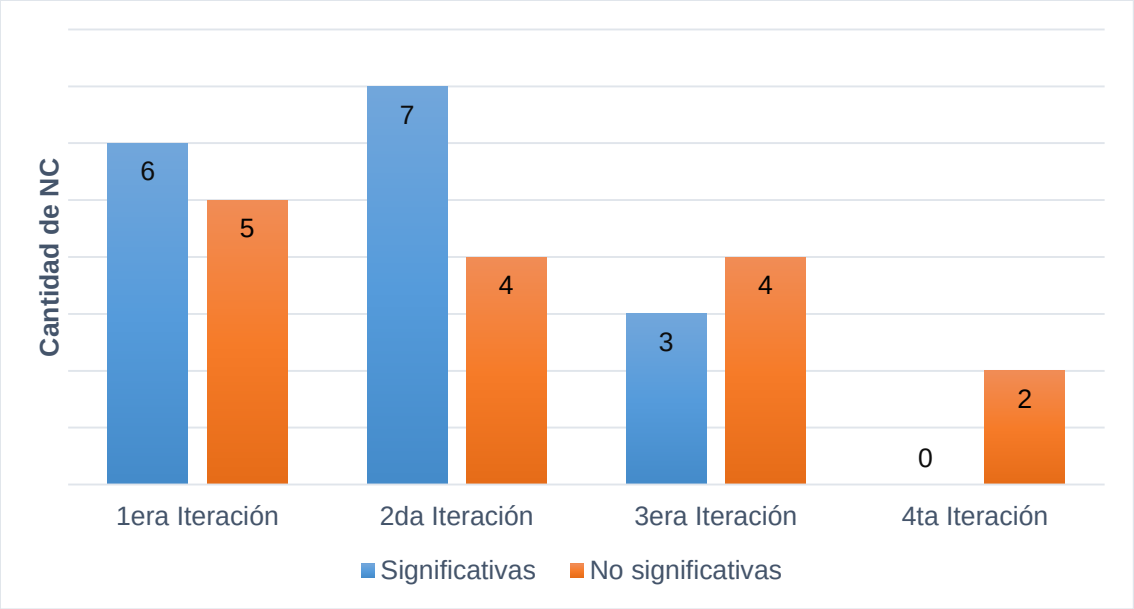

Ilustración 18. Resultados de las pruebas.

Las principales NC no significativas encontradas fueron errores ortográficos, tanto omisiones de tildes como cambios de mayúsculas por minúsculas. Las principales NC significativas encontradas fueron errores al ejecutar acciones como eliminar y mostrar tanto en las alertas como en los archivos modificados. Además, los filtros de tipo fecha presentaban problemas para filtrar y la acción exportar no se realizaba correctamente en una de sus opciones. En la última iteración se detectaron 2 NC las cuales se solucionaron en el mismo momento en que fueron detectadas no siendo necesario la realización de otra iteración. Después de concluida cada iteración se resolvieron las NC arrojadas, para garantizar así la satisfacción del cliente.

3.6 Conclusiones parciales

En el presente capítulo se desarrolló el diagrama de componentes definiendo los principales estereotipos utilizados para su confección y se describió la propuesta de solución por etapas especificando las clases, y entidades creadas. El método de prueba seleccionado permitió detectar los principales errores cometidos en la implementación de la herramienta, los cuales fueron corregidos garantizando su buen funcionamiento.

Conclusiones generales

La culminación de la presente investigación permitió arribar a las siguientes conclusiones:

- En el estudio del estado del arte se analizaron aplicaciones que permiten detectar cambios en los archivos de un directorio, lo que permitió identificar algunas funcionalidades básicas a incorporar en la aplicación para detectar cambios en el código fuente de SIGIES, así como las herramientas y tecnologías necesarias para su desarrollo.
- A través del análisis y el diseño se generaron los artefactos correspondientes, teniendo en cuenta la metodología AUP - UCI, los que guiaron el desarrollo de la herramienta para la detección de cambios en el código fuente de SIGIES.
- Las pruebas realizadas permitieron detectar y solucionar las deficiencias existentes en el funcionamiento de la herramienta, por lo que se obtuvo una aplicación que contribuye a la seguridad e integridad del código fuente de SIGIES.

Recomendaciones

A partir del desarrollo del presente trabajo de diploma se recomienda:

- Desarrollar funciones que permitan la manipulación desde la web de un grupo de servicios con que cuenta Ossec producto a que el acceso a estos servicios se realiza a través de un terminal.
- Identificar e implementar entidades que brinden diferentes opciones de configuración al usuario entre las cuales pudieran estar la notificación por correo electrónico y la especificación de los directorios que se van a monitorear.

Referencias bibliográficas

1. **CARNEIRO, Roberto, TOSCANO, Juan Carlos and DÍAZ, Tamara.** Los desafíos de las TIC para el cambio educativo. *Fundación Santillana: Madrid* [online]. 2009. [Accessed 16 December 2016]. Available from: http://www.academia.edu/download/37227526/cambio_educativo.pdf

2. **IBÁÑEZ, Jesús Salinas.** Innovación docente y uso de las TIC en la enseñanza universitaria. *RUSC. Universities and Knowledge Society Journal*. 2004. Vol. 1, no. 1, p. 3.

3. **SILVA, Pedro Horruitiner.** El modelo curricular de la educación superior cubana. *Pedagogía Universitaria* [online]. 2000. Vol. 5, no. 3. [Accessed 16 December 2016]. Available from: http://cvi.mes.edu.cu/peduniv/index.php/peduniv/article/download/162/159

4. **GONZÁLEZ, Yuleisy Pérez.** FORTES SIGIES Especificación de requisitos de software. La Habana. Cuba: sn.

5. **LÓPEZ, Óscar Andrés, PARRA, Misael Leonardo Prieto and MANZUR, Beatriz Acosta.** Arquitectura y Comunicaciones en un Sistema de Detección de Intrusos. In : *Primer Congreso Iberoamericano de Seguridad Informática CIBSI* [online]. 2002. [Accessed 15 May 2017]. Available from: http://www.academia.edu/download/5068050/pdf.pdf

6. **SANTOS, Rafael Castillo.** Detección de intrusos mediante técnicas de minería de datos. *Revista Clepsidra*. 2006. Vol. 2, no. 2, p. 31–44.

7. **ALFARO, Emilio José Mira.** Implantación de un Sistema de Detección de Intrusos en la Universidad de Valencia. *Ingeniería Informatica* [online]. 2002. [Accessed 15 May 2017]. Available from: http://www.rediris.es/cert/doc/pdf/ids-uv.pdf

8. **PÉREZ PORTO, JULIÁN, Gardey, Ana.** Definición de notificación — Definicion.de. *Definición.de* [online]. 2013. [Accessed 24 May 2017]. Available from: http://definicion.de/notificacion/

9. **Notificación.** *Definición MX* [online]. [Accessed 24 May 2017]. Available from: https://definicion.mx/notificacion/Definición de Notificación

10. **GUTIÉRREZ, Carlos Carrero.** Un sistema integrado de gestión e información de archivos. Aabadom: Boletín de la Asociación Asturiana de Bibliotecarios, Archiveros, Documentalistas y Museólogos. 2003. Vol. 14, no. 1, p. 4–12.

11. **MORERA, Remei Perpinya.** Instrumentos de selección de software para la gestión de archivos. *Bilduma*. 2000. Vol. 14, p. 301–333.

12. **WinAudit - Home.** [online]. [Accessed 17 January 2017]. Available from: https://winaudit.codeplex.com/

13. **systraq.** [online]. [Accessed 11 December 2016]. Available from: http://mdcc.cx/systraq/

14. **Debian -- Details of package diffmon.** [online]. [Accessed 22 January 2017]. Available from: https://packages.debian.org/sid/admin/diffmon

15. **AIDE Manual version 0.16.** [online]. [Accessed 22 January 2017]. Available from: http://aide.sourceforge.net/stable/manual.html

16. **Integrit File Verification System.** [online]. [Accessed 24 January 2017]. Available from: http://integrit.sourceforge.net/

17. **GitHub - Integrit is the most simple Tripwire alternative.** [online]. [Accessed 24 January 2017]. Available from: https://github.com/integrit/integrit

18. **LETELIER, Patricio.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [online]. 2006. [Accessed 3 January 2017]. Available from: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm

19. **The Agile Unified Process (AUP) Home Page**. [online]. [Accessed 17 January 2017]. Available from: http://www.ambysoft.com/unifiedprocess/agileUP.html

20. **Rodríguez Sánchez, Tamara**. Metodología de desarrollo para la Actividad productiva de la UCI. La Habana. Cuba: s.n.

21. **VIDAL, Cristian L., SCHMAL, Rodolfo F., RIVERO, Sabino and VILLARROEL, Rodolfo H.** Extensión del Diagrama de Secuencias UML (Lenguaje de Modelado Unificado) para el Modelado Orientado a Aspectos. *Información tecnológica*. 2012. Vol. 23, no. 6, p. 51–62.

22. **GARCÍA, Jesús Joaquin, ROSSI, Gustavo and MOREIRA, Ana.** UML: el lenguaje estándar para el modelado de software. *Novática: Revista de la Asociación de Técnicos de Informática*. 2004. No. 168, p. 4–5.

23. **RIDDELL, Jonathan.** *Umbrello UML Modeller* [online]. 2008. [Accessed 10 January 2017]. Available from: ftp://hackbbs.org/Dev/Methodologie/UML/Umbrello-UML-Modeler2.pdf

24. **Visual Paradigm para UML.** [online]. [Accessed 29 April 2017]. Available from: http://www.software.com.ar/p/visual-paradigm-para-uml

25. **CARAVACA-MORA, Oscar Mauricio.** Diseño de un Entorno de Desarrollo Integrado para una Unidad Controladora de Procesos. [online]. 2010. [Accessed 24 January 2017]. Available from: http://repositoriotec.tec.ac.cr/handle/2238/2598

26. **PHP: ¿Qué es PHP? - Manual.** [online]. [Accessed 24 January 2017]. Available from: http://php.net/manual/es/intro-whatis.php

27. **PHP: ¿Qué puede hacer PHP? - Manual.** [online]. [Accessed 24 January 2017]. Available from: http://php.net/manual/es/intro-whatcando.php

28. **LAPUENTE, María Jesús Lamarca and LAPUENTE, Chusa Lamarca.** HTML. [online]. [Accessed 29 May 2017]. Available from: http://www.hipertexto.info/documentos/html.htmLenguajes hipertextuales: HTML

29. **ÁLVAREZ, Miguel Ángel.** Etiquetas nuevas de HTML5. [online]. [Accessed 29 May 2017]. Available from: http://www.desarrolloweb.com/articulos/nuevas-etiquetas-html5.html

30. **CSS.** *Mozilla Developer Network* [online]. [Accessed 29 May 2017]. Available from: https://developer.mozilla.org/en-US/docs/Learn/CSS

31. **HERNANDEZ, Oscar.** CARACTERÍSTICAS DE LAS CSS. [online]. [Accessed 29 May 2017]. Available from: http://www.academia.edu/10934492/CARACTER%C3%8DSTICAS_DE_LAS_CS

32. **EGUILUZ, Javier.** Responsive Web Design. [online]. [Accessed 29 May 2017]. Available from: https://www.arkaitzgarro.com/responsive-web-design/index.html

33. **The open source codebase and curriculum.** Learn to code and help nonprofits [online]. JavaScript. freeCodeCamp, 2017. [Accessed 29 May 2017]. Available from: https://github.com/freeCodeCamp/freeCodeCamp

34. **ALEGSA, Leandro.** Definición de Framework de desarrollo. [online]. [Accessed 29 May 2017]. Available from: http://www.alegsa.com.ar/Dic/framework.php

35. **Labs, Sensio.** The Book for Symfony 2.0. 2013.

36. **Symfony 2.7 Documentation.** [online]. [Accessed 30 May 2017]. Available from: http://symfony.com/doc/2.7/index.html

37. **ACEDO, Jose.** ¿Qué es el Framework Bootstrap? Ventajas y Desventajas. [online]. [Accessed 30 May 2017]. Available from: http://programacion.jias.es/2015/05/web-%c2%bfque-es-el-framework-bootstrap-ventajas-desventajas/

38. **ÁLVAREZ, Miguel Ángel.** Introducción a jQuery. DesarrolloWeb.com [online]. [Accessed 30 May 2017]. Available from: http://www.desarrolloweb.com/articulos/introduccion-jquery.html

39. **Servidor web: definición, historia y programas.** [online]. [Accessed 22 January 2017]. Available from: https://www.1and1.es/digitalguide/servidores/know-how/servidor-web-definicion-historia-y-programas/

40. **Welcome to NGINX.** [online]. [Accessed 22 January 2017]. Available from: https://www.nginx.com/resources/wiki/

41. **Nginx Secure Web Server.** [online]. [Accessed 22 January 2017]. Available from: https://calomel.org/nginx.html

42. **Nginx | Grav Documentation.** [online]. [Accessed 22 January 2017]. Available from: https://learn.getgrav.org/webservers-hosting/local/nginx

43. **SÁNCHEZ, Jorge.** Diseño Conceptual de Bases de Datos. *Obtenido de http://www. jorgesanchez. net/bd/disenoBD. pdf* [online]. 2004. [Accessed 24 January 2017].

44. **Sobre PostgreSQL | www.postgresql.org.es.** [online]. [Accessed 24 January 2017]. Available from: http://www.postgresql.org.es/sobre_postgresql

45. **Aspectos Teóricos - Entorno de Desarrollo Integrado (IDE).** [online]. [Accessed 24 January 2017]. Available from: http://cu.globedia.com/aspectos-teoricos-entorno-desarrollo-integrado-ide

46. **PhpStorm IDE :: JetBrains PhpStorm.** [online]. [Accessed 25 January 2017]. Available from: https://www.jetbrains.com/phpstorm/?fromMenu

47. **PhpStorm IDE de programación web.** [online]. [Accessed 25 January 2017]. Available from: http://www.editoresdecodigo.com/2014/06/descargar-phpstorm-full-ide-para-php-y-mas.html

48. **NetBeans IDE entorno de desarrollo para lenguajes como Java PHP C/C++ Groovy.** [online]. [Accessed 24 January 2017]. Available from: https://www.genbetadev.com/herramientas/netbeans-1

49. **LARMAN, Diapositivas**. Modelo Del Dominio. *de UML y Patrones, Prentice Hall*. 2003. P. 23.

50. **Plantillas: Plan de Iteración | Tecnología y Synergix.** [online]. [Accessed 11 May 2017]. Available from: https://synergix.wordpress.com/2008/07/18/plantillas-plan-de-iteracion/

51. **Pressman, Roger S.** Ingenieria de Software. Un enfoque práctico. 2005. ISBN 0072853182.

52. **ROMERO, Yenisleidy Fernández and GONZÁLEZ, Yanette Díaz.** Patrón modelo-vista-controlador. *Revista Telem@tica*. 2012. Vol. 11, no. 1, p. 47–57.

53. **E. Gamma, R. Helm, R., Johnson, and J., Vlissides.** Design Patterns. [online], 1995. [Accessed 15 March 2017]. Available from: http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf.

54. **TABARES, Ricardo Botero**. Patrones Grasp y Anti-Patrones: un Enfoque Orientado a Objetos desde Lógica de Programación. *Entre Ciencia e Ingeniería*. 2011. No. 8, p. 161–173.

55. **GUERRERO, Carlos A., SUÁREZ, Johanna M. and GUTIÉRREZ, Luz E.** Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. *Información tecnológica*. 2013. Vol. 24, no. 3, p. 103–114.

56. **ApunteRUP.doc - rup.pdf.** [online]. [Accessed 16 March 2017]. Available from: http://dsc.itmorelia.edu.mx/~jcolivares/courses/pm10a/rup.pdf

57. **Modelo Entidad-Relación.** [online]. [Accessed 16 March 2017]. Available from: http://dis.um.es/~jfernand/0405/dbd/tema2.pdf

58. **JACOBSON, Ivar, BOOCH, Grady and RUMBAUGH, James.** El proceso unificado de desarrollo de software. Madrid : s.n., 2000.

59. **Estándares de codificación — Manual de Symfony2 en Español.** [online]. [Accessed 12 May 2017]. Available from: http://gitnacho.github.io/symfony-docs-es/contributing/code/standards.html

60. **Cabrera Mata, Yandri.** *FORTES SIGIES. Estandares de codificacion para PHP.* 2016.

61. **USAOLA, Macario Polo**. Pruebas del Software. [online]. [Accessed 13 May 2017]. Available from: https://pdfs.semanticscholar.org/191e/9b745defff8f26557e9c5e4d6da38d3420bd.pdf

62. **MARTÍNEZ, Eduardo Salazar.** Procedimiento para realizar pruebas de Caja Blanca. *Informática Jurídica* [online]. 1 January 2015. [Accessed 13 May 2017]. Available from:

http://www.informatica-juridica.com/trabajos/procedimiento-realizar-pruebas-caja-blanca/.

63. **Técnicas de Evaluación Dinámica.** [online]. [Accessed 14 June 2017]. Available from: http://www.lsi.us.es/docencia/get.php?id=361

Anexos

Anexo I. Descripción de requisitos por proceso

RF 2. Filtrar integridad de los directorios.

Tabla 9. Descripción del requisito funcional filtrar integridad de los directorios.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Filtrar integridad de los directorios	
1.	El usuario selecciona la opción Filtrar integridad de los directorios.
2.	El sistema debe permitir filtrar la integridad de los directorios, teniendo en cuenta los siguientes datos: • Dirección del archivo. • Tamaño (Kb). • Fecha de modificación. Y permite, además, realizar las siguientes opciones: • Buscar
1.	El usuario selecciona los filtros de búsqueda e introduce o selecciona los datos para Buscar la integridad de los directorios y selecciona la opción Buscar.
2.	El sistema muestra el listado de las categorías de acuerdo a los elementos de filtrado que se han seleccionado.
3.	Concluye así el requisito.
Pos-condiciones	
1.	Se filtró la integridad de los directorios satisfactoriamente.
Flujos alternativos	

Flujo alternativo 4.a No existen coincidencias		
1.	El sistema no encuentra coincidencias con los elementos de filtrado seleccionados y muestra el mensaje de información: No existen coincidencias.	
2.	Volver al paso 2 del flujo básico.	
Pos-condiciones		
1.	No se mostró el listado filtrado de categorías.	
Validaciones		
1.	NA	
Conceptos	NA	NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

RF 3. Exportar integridad de los directorios.

Tabla 10. Descripción del requisito funcional exportar integridad de los directorios.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Exportar integridad de los directorios	
1.	El usuario selecciona la opción Exportar.
2.	El sistema debe permitir que se exporte la integridad de los directorios, teniendo en cuenta los siguientes formatos: • PDF • XLS • DOCX
1.	El usuario selecciona la opción PDF.
2.	El sistema exporta los datos a formato .pdf.

3.	Concluye así el requisito.
Pos-condiciones	
1.	Se exportaron los datos satisfactoriamente.
Flujo alternativo 3.a Opción XLS	
1.	El usuario selecciona la opción XLS.
2.	El sistema exporta los datos a formato .xls.
3.	Concluye así el requisito.
Flujo alternativo 3.b Opción DOCX	
4.	El usuario selecciona la opción DOCX.
5.	El sistema exporta los datos a formato .docx.
6.	Concluye así el requisito.
Pos-condiciones	
1.	Se exportaron los datos satisfactoriamente.
Validaciones	
1.	NA

Conceptos	NA	NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

RF 4. Eliminar integridad de los directorios.

Tabla 11. Descripción del requisito funcional eliminar integridad de los directorios.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Eliminar integridad de los directorios	

1.	El usuario selecciona un elemento de la integridad de los directorios y la opción Eliminar desde el listado de la integridad de los directorios o desde la vista previa del propio elemento.
2.	El sistema muestra un mensaje de confirmación. Además, permite seleccionar las siguientes opciones: • Aceptar • Cancelar
1.	El usuario selecciona la opción Aceptar.
2.	El sistema elimina la integridad de los directorios.
3.	El sistema actualiza el listado y muestra el siguiente mensaje de información: La entidad se ha eliminado satisfactoriamente.
4.	Concluye así el requisito.
Pos-condiciones	
1.	Se eliminó la integridad de los directorios satisfactoriamente.
Flujos alternativos	
Flujo alternativo * Cancelar	
1.	El usuario selecciona la opción Cancelar.
2.	El sistema elimina los datos creados, regresa a la interfaz anterior y muestra el mensaje de información: La acción ha sido cancelada.
3.	Concluye así el requisito.
Pos-condiciones	
1.	No se elimina la integridad de los directorios.
Flujos alternativos	
Flujo alternativo * Cancelar	
1.	El usuario selecciona la opción Cancelar.
2.	El sistema elimina los datos creados, regresa a la interfaz anterior y muestra el mensaje de información: La acción ha sido cancelada.

3.	Concluye así el requisito.
Pos-condiciones	
1.	No se eliminan las entidades integridad de los directorios.
Validaciones	
1.	NA

Conceptos	NA	NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

RF 6. Listar alertas.

Tabla 12. Descripción del requisito funcional listar alertas.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Listar alertas	
1.	El usuario selecciona la opción Listar alertas.
2.	El sistema debe permitir listar las alertas existentes de forma ascendente o descendente mostrando los siguientes datos: • Fecha. • Ubicación. • Descripción. • Información.
3.	Concluye así el requisito.
Pos-condiciones	
1.	Se listó las alertas satisfactoriamente.
Flujos alternativos	

Flujo alternativo		
1.	N/A	
Pos-condiciones		
1.	NA	
Validaciones		
1.	NA	
Conceptos	NA	NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

RF 7. Filtrar alertas.

Tabla 13. Descripción del requisito funcional filtrar alertas.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Filtrar alertas	
1.	El usuario selecciona la opción Filtrar alertas.
2.	El sistema debe permitir filtrar alertas, teniendo en cuenta los siguientes datos: • Fecha. • Ubicación. • Descripción. • Información. Y permite, además, realizar las siguientes opciones: • Buscar
3.	El usuario selecciona los filtros de búsqueda e introduce o selecciona los datos para Buscar las alertas y selecciona la opción Buscar.

4.	El sistema muestra el listado de las categorías de acuerdo a los elementos de filtrado que se han seleccionado.		
5.	Concluye así el requisito.		
Pos-condiciones			
1.	Se realizó el filtrado de las alertas satisfactoriamente.		
Flujos alternativos			
Flujo alternativo 4.a No existen coincidencias			
1.	El sistema no encuentra coincidencias con los elementos de filtrado seleccionados y muestra el mensaje de información: No existen coincidencias.		
2.	Volver al paso 2 del flujo básico.		
Pos-condiciones			
1.	No se mostró el listado filtrado de las alertas.		
Validaciones			
1.	NA		
Conceptos	NA	NA	
Requisitos especiales			
Asuntos pendientes	NA		

RF 8. Exportar alertas.

Tabla 14. Descripción del requisito funcional exportar alertas.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Exportar alertas	
1.	El usuario selecciona la opción Exportar.
2.	El sistema debe permitir que se exporte la integridad de los directorios, teniendo en cuenta los siguientes formatos:

	• PDF • XLS • DOCX
3.	El usuario selecciona la opción PDF.
4.	El sistema exporta los datos a formato .pdf.
5.	Concluye así el requisito.
Pos-condiciones	
1.	Se exportaron los datos satisfactoriamente.
Flujo alternativo 3.a Opción XLS	
1.	El usuario selecciona la opción XLS.
2.	El sistema exporta los datos a formato .xls.
3.	Concluye así el requisito.
Flujo alternativo 3.b Opción DOCX	
Pos-condiciones	
1.	Se exportaron los datos satisfactoriamente.
Validaciones	
1.	NA

Conceptos	NA	NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

RF 9. Eliminar alertas.

Tabla 15. Descripción del requisito funcional eliminar alertas.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Eliminar alertas	
1.	El usuario selecciona un elemento de las alertas y la opción Eliminar desde el listado de las alertas o desde la vista previa del propio elemento.
2.	El sistema muestra un mensaje de confirmación. Además, permite seleccionar las siguientes opciones: • Aceptar • Cancelar
1.	El usuario selecciona la opción Aceptar.
2.	El sistema elimina la integridad de los directorios.
3.	El sistema actualiza el listado y muestra el siguiente mensaje de información: La entidad se ha eliminado satisfactoriamente.
4.	Concluye así el requisito.
Pos-condiciones	
1.	Se eliminó la alerta satisfactoriamente.
Flujos alternativos	
Flujo alternativo * Cancelar	
1.	El usuario selecciona la opción Cancelar.
2.	El sistema elimina los datos creados, regresa a la interfaz anterior y muestra el mensaje de información: La acción ha sido cancelada.
3.	Concluye así el requisito.
Pos-condiciones	
1.	No se elimina la alerta.

Validaciones		
1.	NA	
Conceptos	NA	NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

RF 10. Mostrar alertas.

Tabla 16. Descripción del requisito funcional mostrar alertas.

Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo de eventos	
Flujo básico Mostrar alertas	
1.	El usuario selecciona la opción Mostrar del listado de las alertas.
2.	El sistema debe mostrar los siguientes datos de las alertas: • Fecha. • ID de la regla. • Nivel. • Ubicación. • Descripción. • Información. Y permite, además, realizar las siguientes opciones: • Regresar al listado
3.	El usuario visualiza los datos y selecciona la opción Regresar al listado.
4.	El sistema regresa a la interfaz anterior.
5.	Concluye así el requisito.

Pos-condiciones		
1.	Se visualizaron los datos de las alertas satisfactoriamente.	
Flujos alternativos		
Pos-condiciones		
1.	NA	
Pos-condiciones		
1.	N/A	
Validaciones		
Conceptos	NA	NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

Anexo II. Diagramas de clases del análisis.

DCA 2. Filtrar integridad de los directorios.

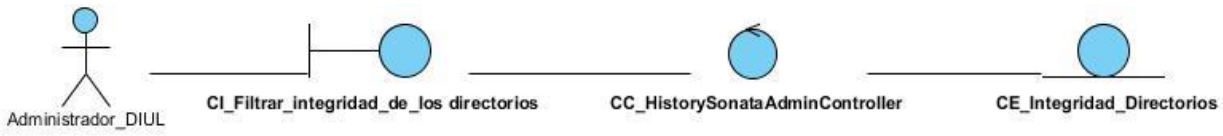

Ilustración 19. Diagrama de clases del análisis del requisito funcional filtrar integridad de los directorios.

DCA 3. Exportar integridad de los directorios.

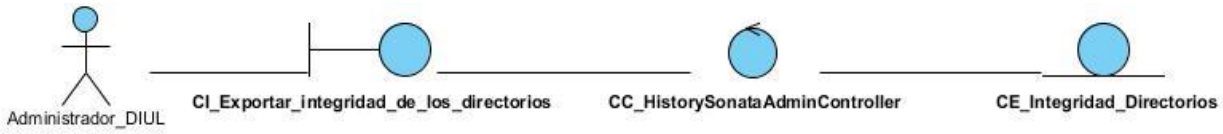

Ilustración 20. Diagrama de clases del análisis del requisito funcional exportar integridad de los directorios.

DCA 4. Eliminar integridad de los directorios.

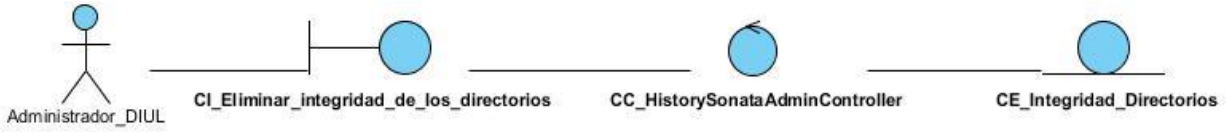

Ilustración 21. Diagrama de clases del análisis del requisito funcional eliminar integridad de los directorios.

DCA 6. Listar alertas.

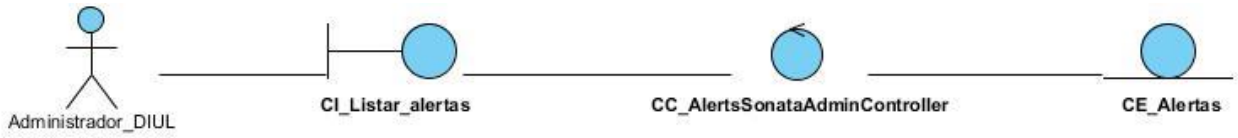

Ilustración 22. Diagrama de clases del análisis del requisito funcional listar alertas.

DCA 7. Filtrar alertas.

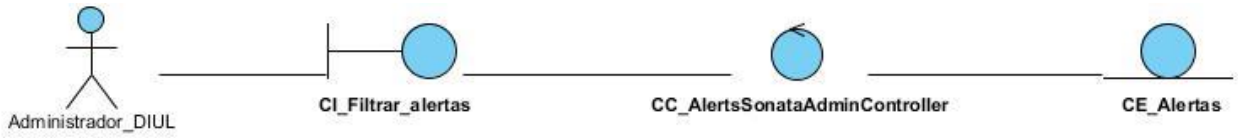

Ilustración 23. Diagrama de clases del análisis del requisito funcional filtrar alertas.

DCA 8. Exportar alertas.

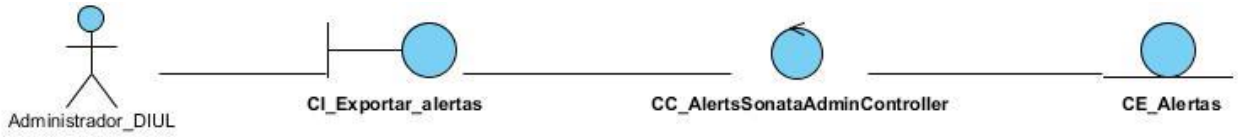

Ilustración 24. Diagrama de clases del análisis del requisito funcional exportar alertas.

DCA 9. Eliminar alertas.

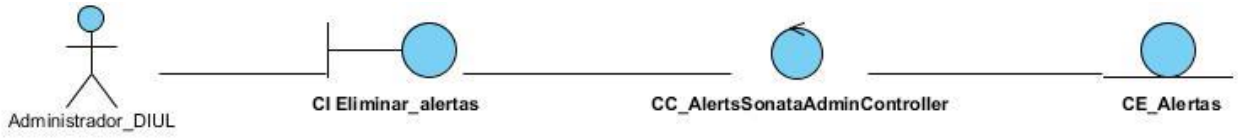

Ilustración 25. Diagrama de clases del análisis del requisito funcional eliminar alertas.

DCA 10. Mostrar alertas.

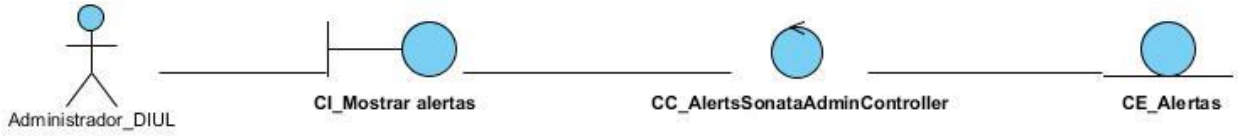

Ilustración 26. Diagrama de clases del análisis del requisito funcional mostrar alertas.

Anexo III. Diagramas colaboración.

DC 2. Filtrar integridad de los directorios.

Ilustración 27. Diagrama de colaboración del análisis del requisito funcional filtrar integridad de los directorios.

DC 3. Exportar integridad de los directorios.

Ilustración 28. Diagrama de colaboración del análisis del requisito funcional exportar integridad de los directorios.

DC 4. Eliminar integridad de los directorios.

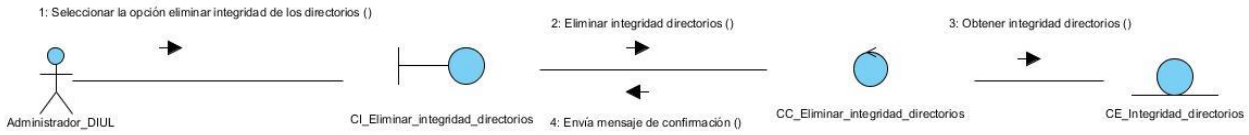

Ilustración 29. Diagrama de colaboración del análisis del requisito funcional eliminar integridad de los directorios.

DC 6. Listar alertas.

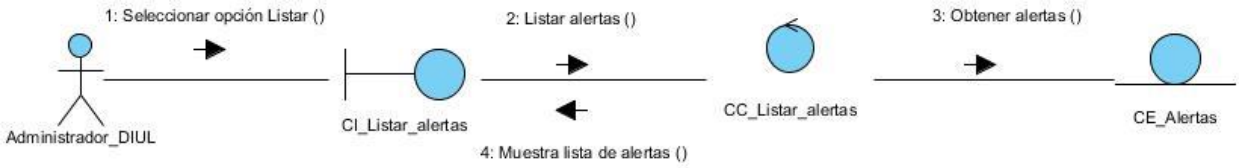

Ilustración 30. Diagrama de colaboración del análisis del requisito funcional listar alertas.

DC 7. Filtrar alertas.

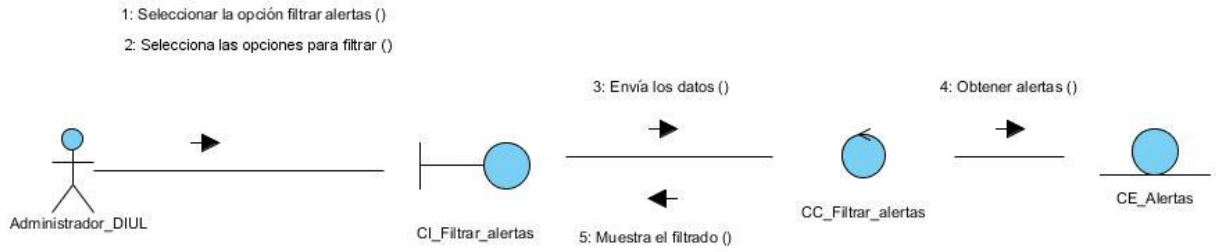

Ilustración 31. Diagrama de colaboración del análisis del requisito funcional filtrar alertas.

DC 8. Exportar alertas.

Ilustración 32. Diagrama de colaboración del análisis del requisito funcional exportar alertas.

DC 9. Eliminar alertas.

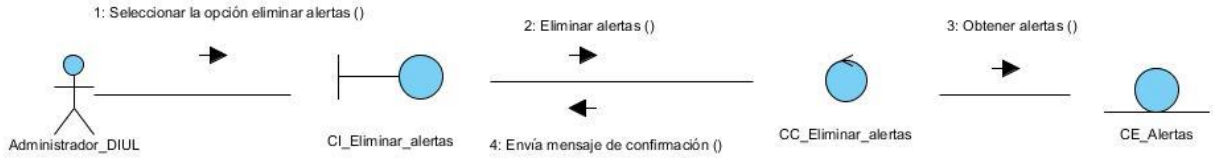

Ilustración 33. Diagrama de colaboración del análisis del requisito funcional eliminar alertas.

DC 10. Mostrar alertas.

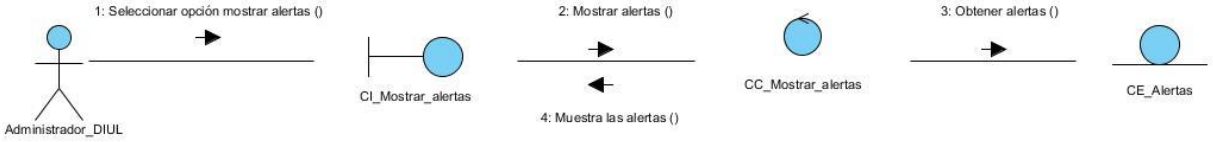

Ilustración 34. Diagrama de colaboración del análisis del requisito funcional mostrar alertas.

Anexo IV. Diagramas de clases del diseño.

DCD 2. Filtrar integridad de los directorios.

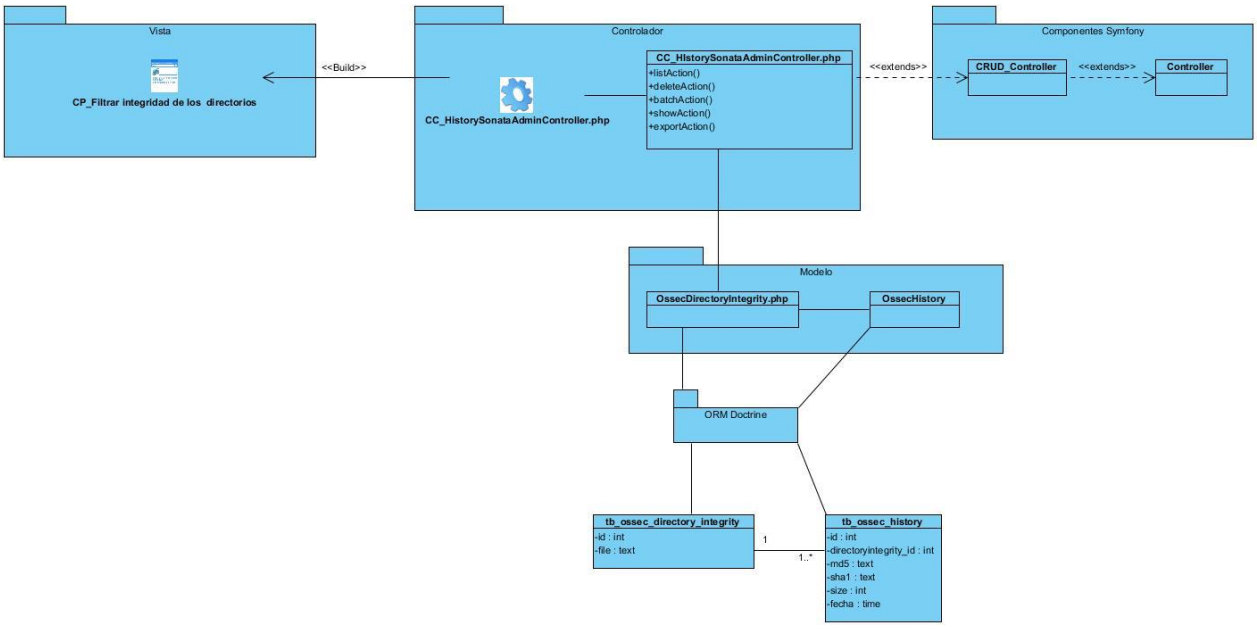

Ilustración 35. Diagrama de clases del diseño del requisito funcional filtrar integridad de los directorios.

DCD 3. Exportar integridad de los directorios.

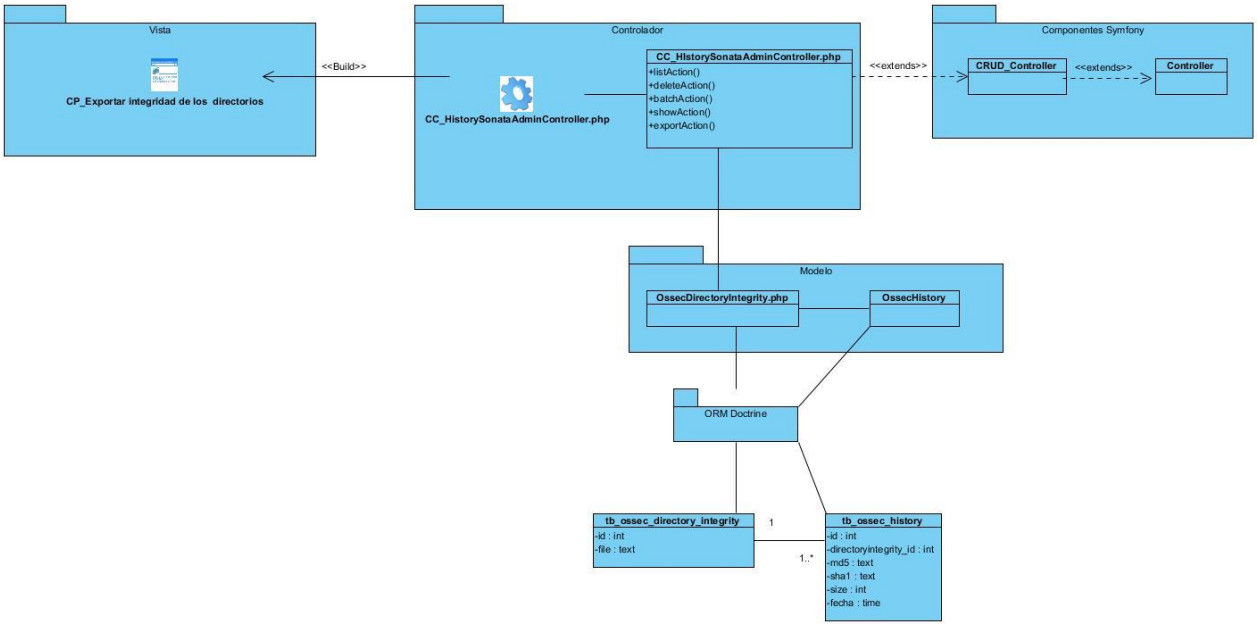

Ilustración 36. Diagrama de clases del diseño del requisito funcional exportar integridad de los directorios.

DCD 4. Eliminar integridad de los directorios.

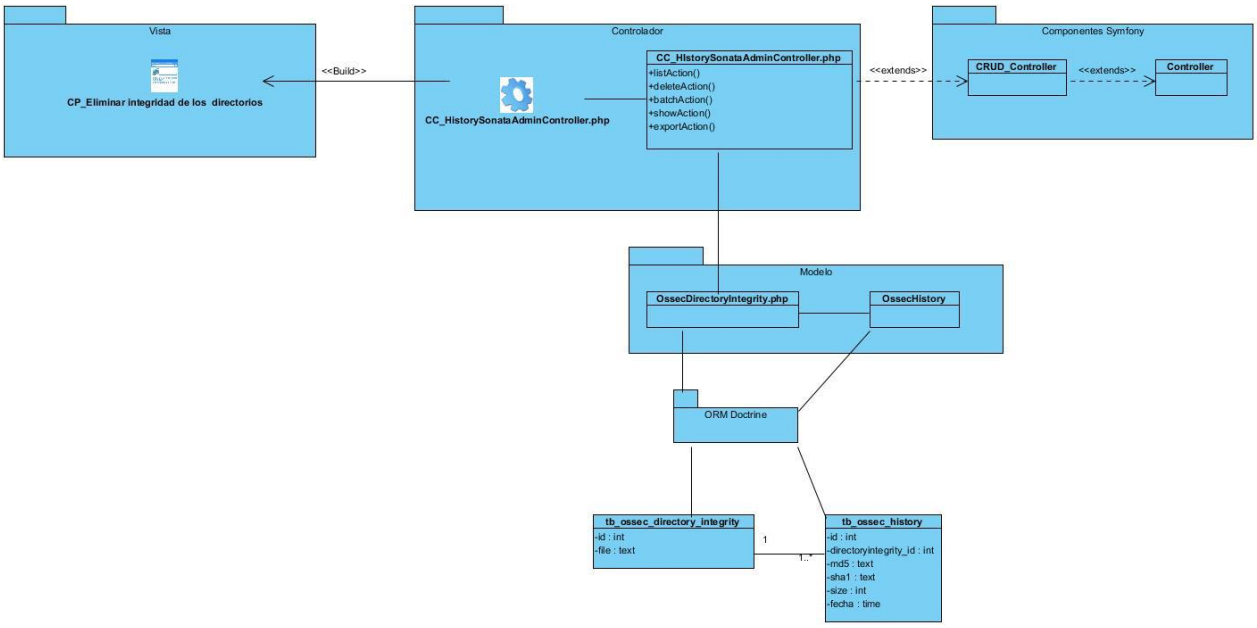

Ilustración 37. Diagrama de clases del diseño del requisito funcional eliminar integridad de los directorios.

DCD 6. Listar alertas.

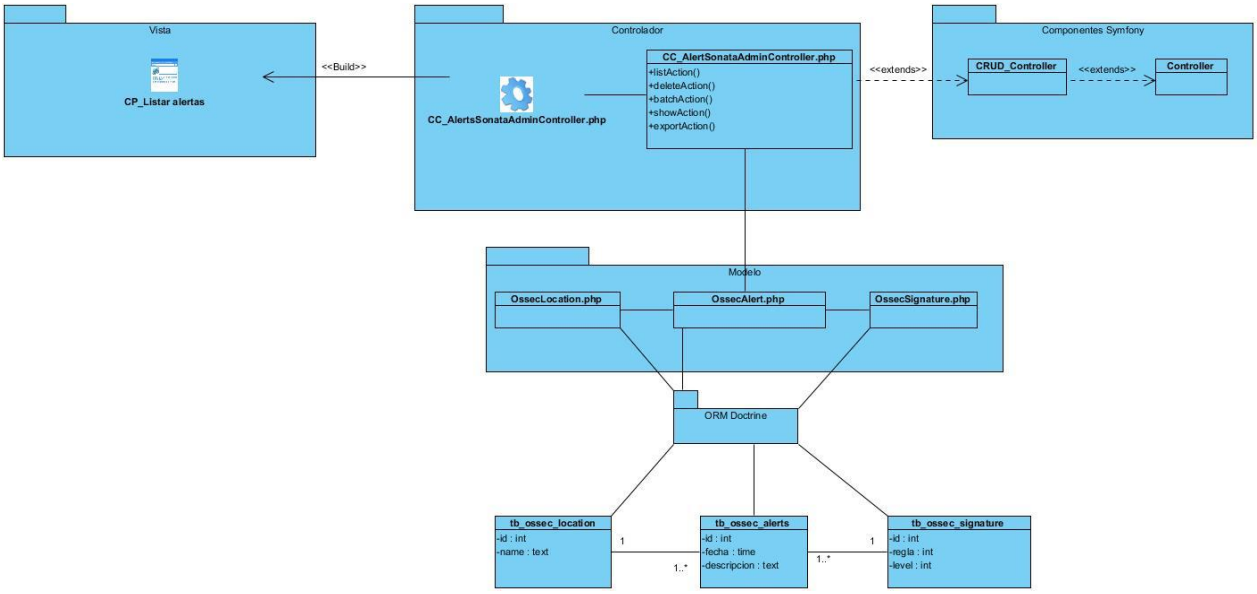

Ilustración 38. Diagrama de clases del diseño del requisito funcional listar alertas.

DCD 7. Filtrar alertas.

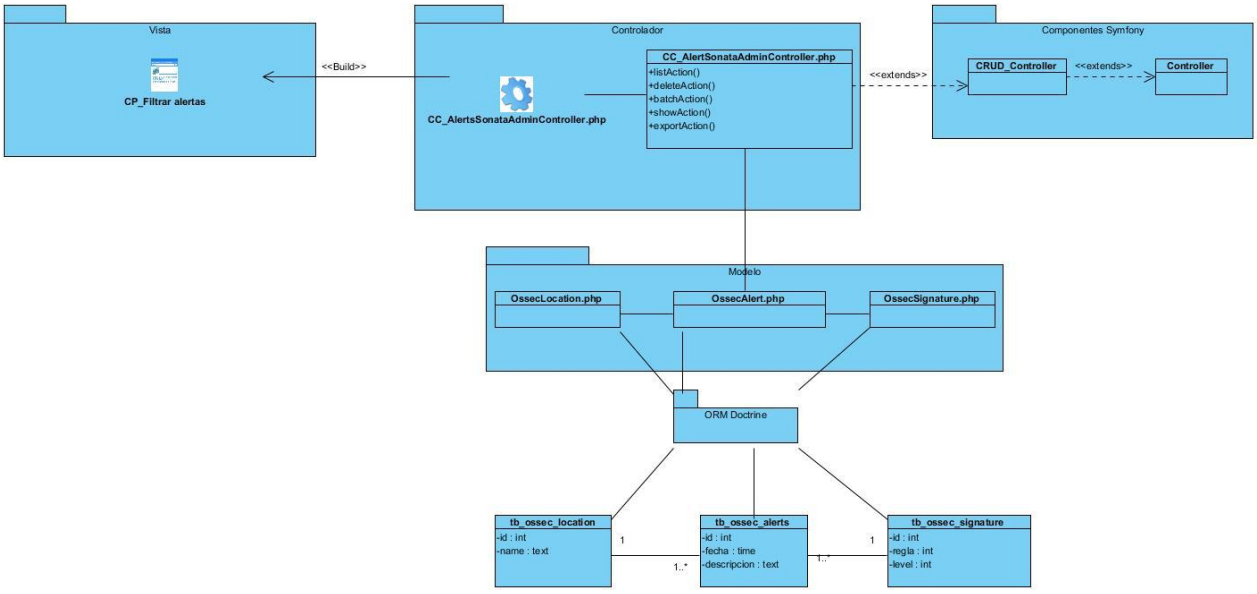

Ilustración 39. Diagrama de clases del diseño del requisito funcional filtrar alertas.

DCD 8. Exportar alertas.

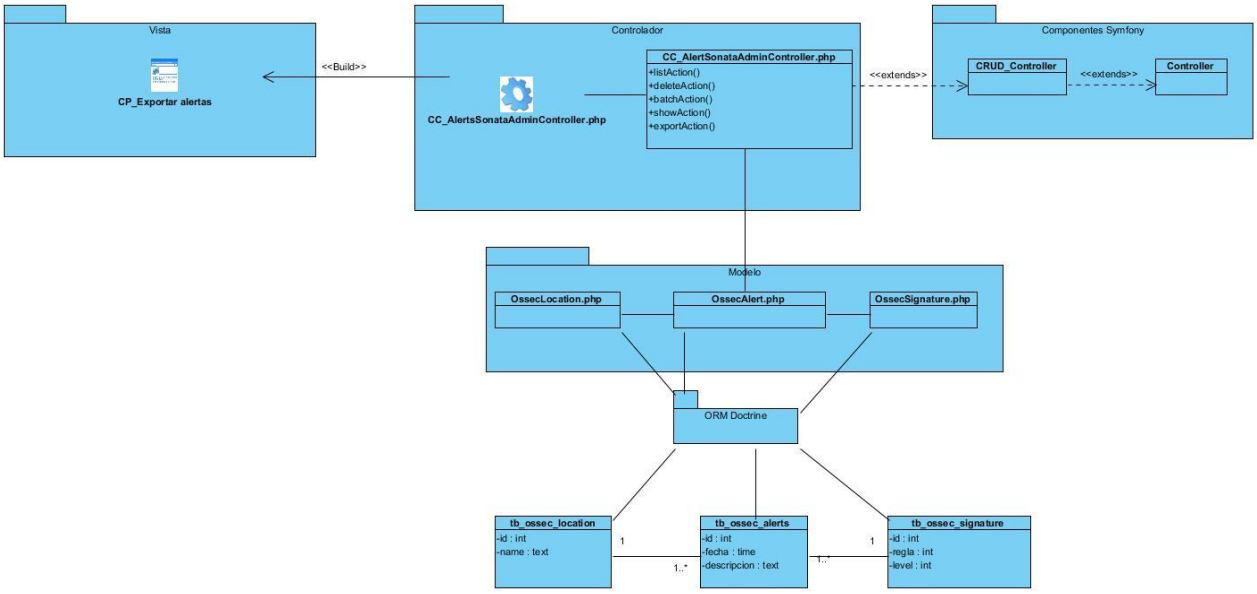

Ilustración 40. Diagrama de clases del diseño del requisito funcional exportar alertas.

DCD 9. Eliminar alertas.

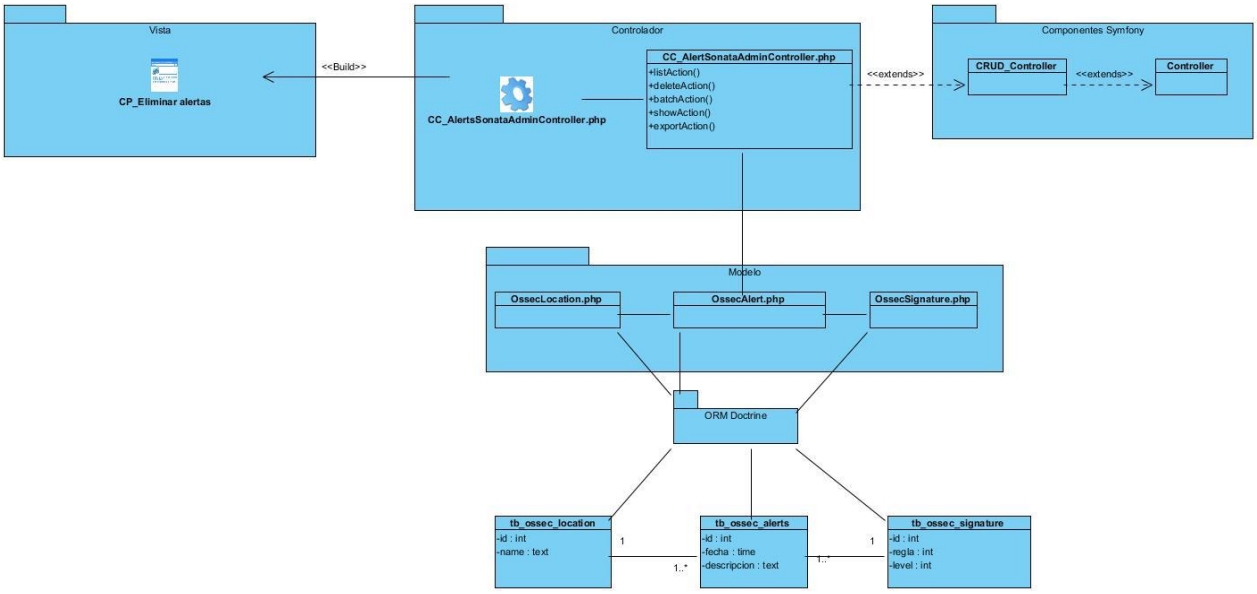

Ilustración 41. Diagrama de clases del diseño del requisito funcional eliminar alertas.

DCD 10. Mostrar alertas.

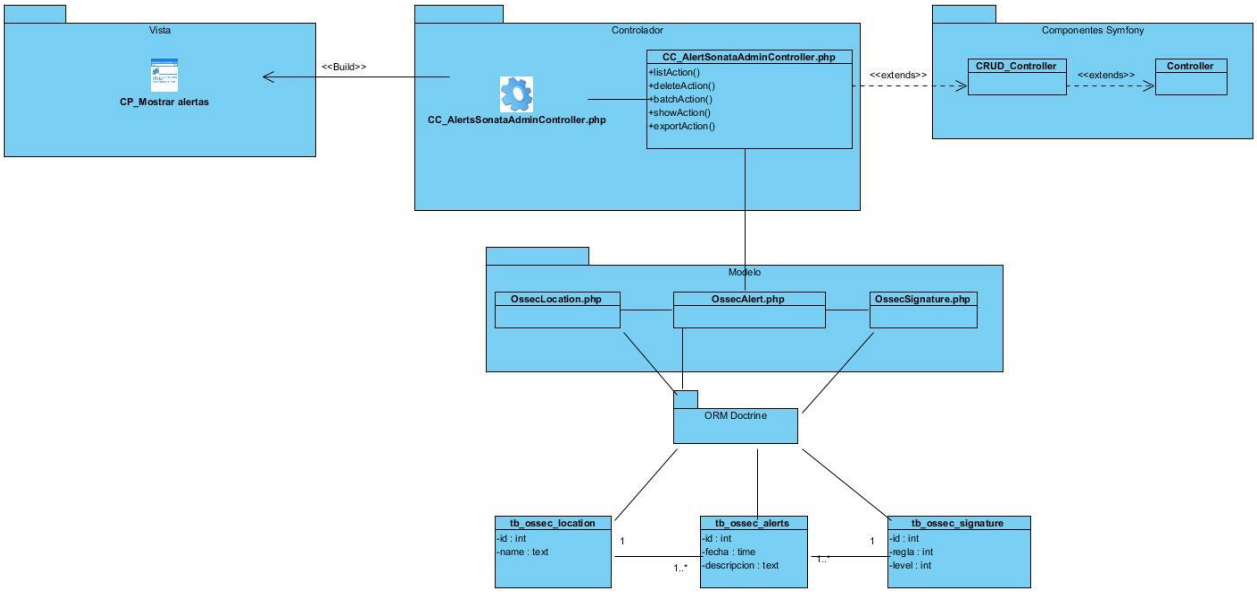

Ilustración 42. Diagrama de clases del diseño del requisito funcional mostrar alertas.

Anexo V. Diagramas de secuencia del diseño.

DSD 2. Filtrar integridad de los directorios.

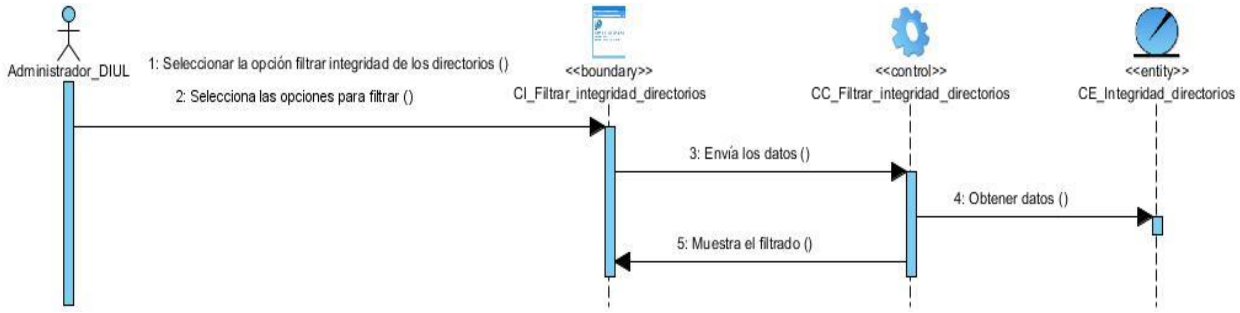

Ilustración 43. Diagrama de secuencia del diseño del requisito funcional filtrar integridad de los directorios.

DSD 3. Exportar integridad de los directorios.

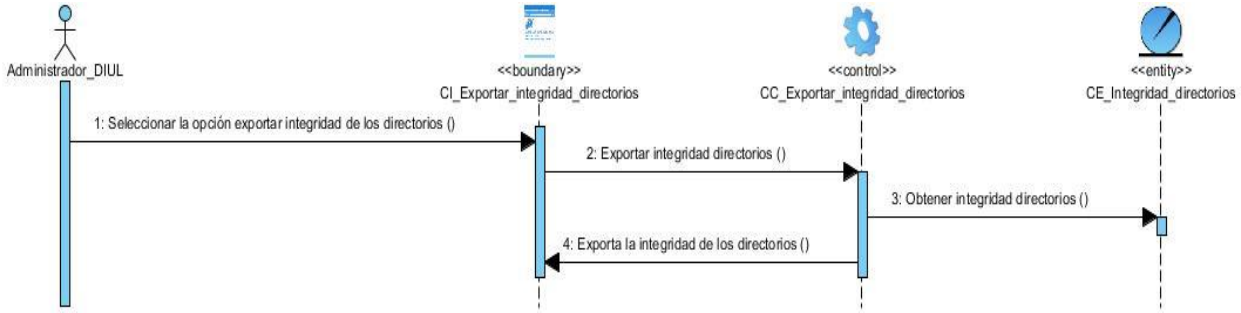

Ilustración 44. Diagrama de secuencia del diseño del requisito funcional exportar integridad de los directorios.

DSD 4. Eliminar integridad de los directorios.

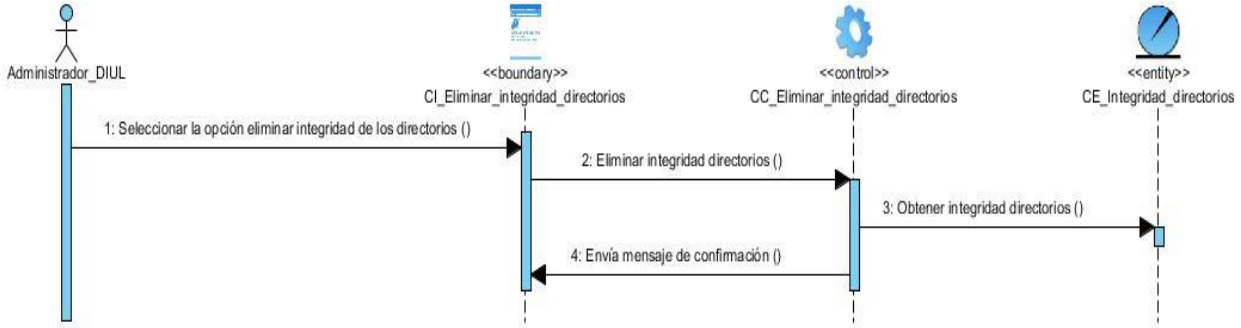

Ilustración 45. Diagrama de secuencia del diseño del requisito funcional eliminar integridad de los directorios.

DSD 6. Listar alertas.

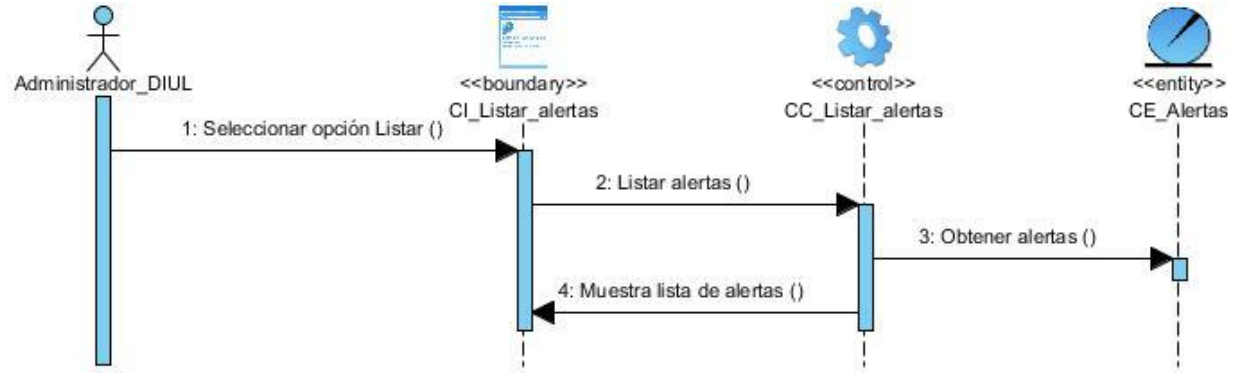

Ilustración 46. Diagrama de secuencia del diseño del requisito funcional listar alertas.

DSD 7. Filtrar alertas.

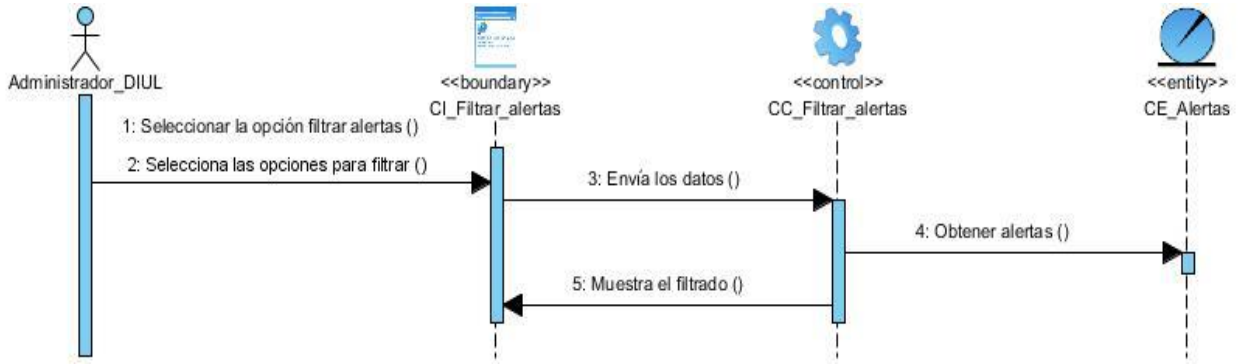

Ilustración 47. Diagrama de secuencia del diseño del requisito funcional filtrar alertas.

DSD 8. Exportar alertas.

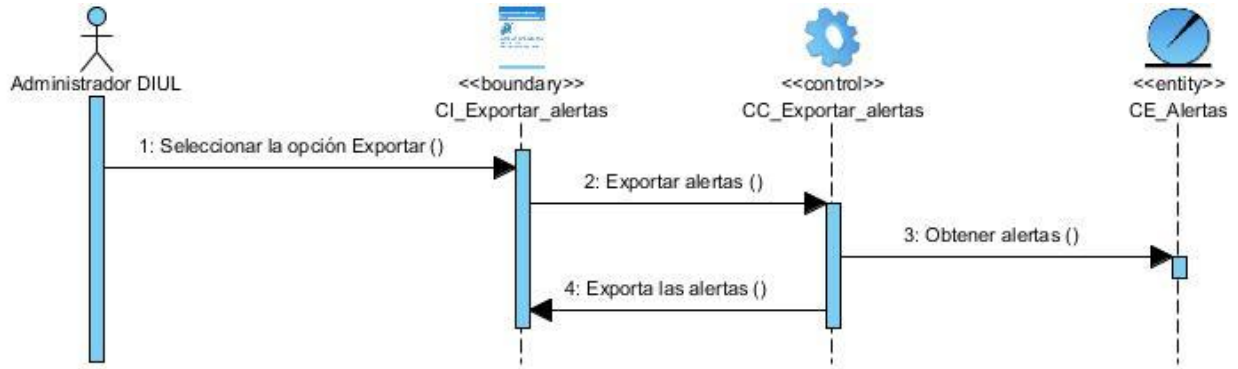

Ilustración 48. Diagrama de secuencia del diseño del requisito funcional exportar alertas.

DSD 9. Eliminar alertas.

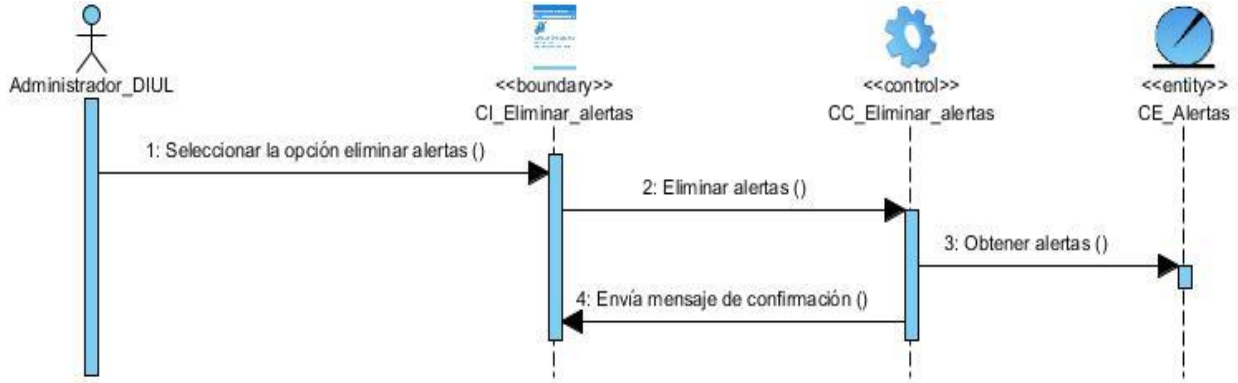

Ilustración 49. Diagrama de secuencia del diseño del requisito funcional eliminar alertas.

DSD 10. Mostrar alertas.

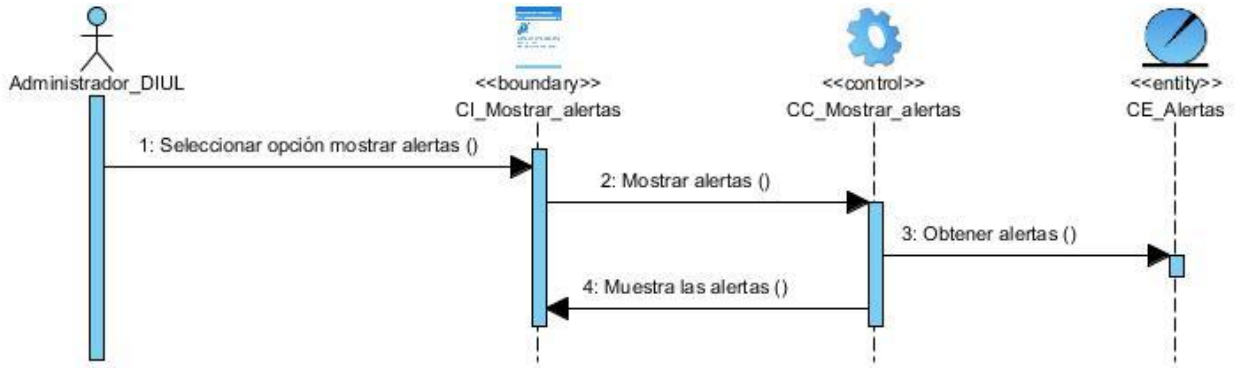

Ilustración 50. Diagrama de secuencia del diseño del requisito funcional mostrar alertas.

Anexo VI. Diseños de casos de prueba.

DCP 2. Filtrar integridad de los directorios.

Tabla 17. DCP del requisito funcional filtrar integridad de los directorios.

Escenario	Descripción	Criterio	Respuesta del sistema	Flujo central
EC 1.1 Opción Filtros	El usuario selecciona la opción Filtros.		El sistema debe permitir filtrar la integridad de los directorios teniendo en cuenta los siguientes datos: - Archivo - Tamaño (Kb) - Fecha	General/Seguridad/Ossec/History/Filtros
EC 1.2 Introducir criterios de búsqueda	El usuario selecciona el/ los filtros de búsqueda e introduce o selecciona los datos para buscar la integridad de los directorios.	N/A	El sistema muestra las siguientes opciones: - Filtrar - Restablecer - Filtros avanzados	General/Seguridad/Ossec/History/Filtros/ Criterio de búsqueda X

EC 1.3 Opción Filtrar	El usuario selecciona la opción Filtrar.	N/A	El sistema muestra el listado de la integridad de los directorios de acuerdo a los elementos de filtrado que se han seleccionado.	General/Seguridad/Ossec/History/Filtros/ Criterio de búsqueda X/Filtrar
EC 1.4 No existen coincidencias	Luego de seleccionar la opción Filtrar el sistema no encuentra coincidencias.	N/A	El sistema no encuentra coincidencias con los elementos de filtrado seleccionados y muestra el mensaje de información: No existen coincidencias.	General/Seguridad/Ossec/Alertas/Filtros/ Criterio de búsqueda X/Filtrar
EC 1.5 Restablecer	El usuario selecciona la opción Restablecer.	N/A	El sistema elimina los filtros seleccionados.	General/Seguridad/Ossec/History/Filtros/ Criterio de búsqueda X/Restablecer
EC 1.6 Filtros avanzados	El usuario selecciona la opción Filtros avanzados.	N/A	El sistema permite especificar cómo desea que se verifique cada criterio (contiene, no contiene, es igual a).	General/Seguridad/Ossec/History/Filtros/ Criterio de búsqueda X/Filtros avanzados
EC 1.7 Especificar cómo filtrar	El usuario específica para cada criterio como desea que se verifique.	N/A	Regresa al EC 1.2	General/Seguridad/Ossec/History/Filtros/ Criterio de búsqueda X/Filtros avanzados/ Filtro avanzado X
EC 1. 8 Datos incompletos	El usuario selecciona uno o más criterios de filtrado para realizar la búsqueda y lo(s) deja en blanco.	N/A	En caso de que el usuario deje en blanco uno o varios criterios de filtrado seleccionados, el sistema elimina los mismos y muestra los archivos modificados cuyos datos coinciden total o parcialmente con el resto de los criterios introducidos por el usuario. En caso de que el usuario deje	General/Seguridad/Ossec/History/Filtros/ Criterio de búsqueda X/Filtrar

			en blanco todos los criterios de filtrado seleccionados, el sistema muestra un listado con todos los archivos modificados en el sistema.	

DCP 3. Exportar integridad de los directorios.

Tabla 18. DCP del requisito funcional exportar integridad de los directorios.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Exportar	El usuario selecciona la opción Filtros.	El sistema debe permitir exportar la integridad de los directorios a los siguientes formatos: - PDF - XLS - DOCX	General/Seguridad/Ossec/History/Exportar
EC 1.2 Opción PDF	El usuario selecciona la opción PDF.	El sistema muestra una ventana donde le brinda al usuario la opción de abrir el archivo exportado o guardarlo en una ubicación específica.	General/Seguridad/Ossec/History/Exportar/PDF
EC 1.3 Opción XLS	El usuario selecciona la opción XLS.	El sistema muestra una ventana donde le brinda al usuario la opción de abrir el archivo exportado o guardarlo en una ubicación específica.	General/Seguridad/Ossec/History/Exportar/XLS
EC 1.4 Opción DOCX	El usuario selecciona la opción DOCX.	El sistema muestra una ventana donde le brinda al usuario la opción de abrir el archivo exportado o guardarlo en una ubicación específica.	General/Seguridad/Ossec/History/Exportar/DOCX

DCP 4. Eliminar integridad de los directorios.

Tabla 19. DCP del requisito funcional eliminar integridad de los directorios.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Eliminar archivos modificados	Selecciona la opción de eliminar un archivo modificado	Muestra un mensaje de confirmación. Y permite: - Aceptar. - Cancelar.	General/Seguridad/Ossec/History/Eliminar
EC 1.2 Opción Aceptar	Selecciona la opción Aceptar.	Elimina el archivo modificado. Regresa al listado de instituciones actualizado y muestra un mensaje de información.	General/Seguridad/Ossec/History/Eliminar/Aceptar
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar	Regresa a la vista anterior.	General/Seguridad/Ossec/History/Eliminar/Cancelar

DCP 6. Listar alertas.

Tabla 20. DCP del requisito funcional listar alertas.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Listar alertas	El usuario selecciona la opción Alertas.	El sistema debe permitir listar las alertas generadas por Ossec. En el listado se deben mostrar los siguientes datos: - Fecha. - Ubicación. - Descripción. - Información. Y permite además, realizar las siguientes opciones: - Mostrar - Eliminar - Filtros - Exportar	General/Seguridad/Ossec/Alertas

EC 1.2 Opción Mostrar	El usuario selecciona la opción Mostrar	El sistema permite visualizar los datos de los archivos modificados. Ver DCP_5_Mostrar_alertas.ods	General/Seguridad/Ossec/Alertas/Mostrar
EC 1.3 Opción Eliminar	El usuario selecciona la opción Eliminar.	El sistema permite eliminar los datos de los archivos modificados. Ver DCP_4_Eliminar_alertas.ods	General/Seguridad/Ossec/Alertas/Eliminar
EC 1.4 Opción Filtros	El usuario selecciona la opción Filtros.	El sistema permite filtrar los datos de los archivos modificados. Ver DCP_2_Filtrar_alertas.ds	General/Seguridad/Ossec/Alertas/Filtros
EC 1.5 Opción Exportar datos	El usuario selecciona la opción exportar.	El sistema permite exportar los datos de los archivos modificados. Ver DCP_3_Exportar_alertas.ods	General/Seguridad/Ossec/Alertas/Exportar

DCP 7. Filtrar alertas.

Tabla 21. DCP del requisito funcional filtrar alertas.

Escenario	Descripción	Criterio	Respuesta del sistema	Flujo central
EC 1.1 Opción Filtros	El usuario selecciona la opción Filtros.		El sistema debe permitir filtrar las alertas teniendo en cuenta los siguientes datos: - Fecha - Ubicación - Descripción - Información	General/Seguridad/Ossec/Alertas/Filtros
EC 1.2 Introducir criterios de búsqueda	El usuario selecciona el/ los filtros de búsqueda e introduce o selecciona los datos para buscar la integridad de los directorios.	N/A	El sistema muestra las siguientes opciones: - Filtrar - Restablecer - Filtros avanzados	General/Seguridad/Ossec/Alertas/Filtros/ Criterio de búsqueda X

EC 1.3 Opción Filtrar	El usuario selecciona la opción Filtrar.	N/A	El sistema muestra el listado de las alertas de acuerdo a los elementos de filtrado que se han seleccionado.	General/Seguridad/Ossec/Alertas/Filtros/ Criterio de búsqueda X/Filtrar
EC 1.4 No existen coincidencias	Luego de seleccionar la opción Filtrar el sistema no encuentra coincidencias.	N/A	El sistema no encuentra coincidencias con los elementos de filtrado seleccionados y muestra el mensaje de información: No existen coincidencias.	General/Seguridad/Ossec/Alertas/Filtros/ Criterio de búsqueda X/Filtrar
EC 1.5 Restablecer	El usuario selecciona la opción Restablecer.	N/A	El sistema elimina los filtros seleccionados.	General/Seguridad/Ossec/Alertas/Filtros/ Criterio de búsqueda X/Restablecer
EC 1.6 Filtros avanzados	El usuario selecciona la opción Filtros avanzados.	N/A	El sistema permite especificar cómo desea que se verifique cada criterio (contiene, no contiene, es igual a).	General/Seguridad/Ossec/Alertas/Filtros/ Criterio de búsqueda X/Filtros avanzados
EC 1.7 Especificar cómo filtrar	El usuario específica para cada criterio como desea que se verifique.	N/A	Regresa al EC 1.2	General/Seguridad/Ossec/Alertas/Filtros/ Criterio de búsqueda X/Filtros avanzados/ Filtro avanzado X
EC 1. 8 Datos incompletos	El usuario selecciona uno o más criterios de filtrado para realizar la búsqueda y lo(s) deja en blanco.	N/A	En caso de que el usuario deje en blanco uno o varios criterios de filtrado seleccionados, el sistema elimina los mismos y muestra las alertas cuyos datos coinciden total o parcialmente con el resto de los criterios	General/Seguridad/Ossec/Alertas/Filtros/ Criterio de búsqueda X/Filtrar

			introducidos por el usuario. En caso de que el usuario deje en blanco todos los criterios de filtrado seleccionados, el sistema muestra un listado con todas las alertas generadas por Ossec.	

DCP 8. Exportar alertas.

Tabla 22. DCP del requisito funcional exportar alertas.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Exportar	El usuario selecciona la opción Filtros.	El sistema debe permitir exportar las alertas a los siguientes formatos: - PDF - XLS - DOCX	General/Seguridad/Ossec/Alertas/Exportar
EC 1.2 Opción PDF	El usuario selecciona la opción PDF.	El sistema muestra una ventana donde le brinda al usuario la opción de abrir el archivo exportado o guardarlo en una ubicación específica.	General/Seguridad/Ossec/Alertas/Exportar/PDF
EC 1.3 Opción XLS	El usuario selecciona la opción XLS.	El sistema muestra una ventana donde le brinda al usuario la opción de abrir el archivo exportado o guardarlo en una ubicación específica.	General/Seguridad/Ossec/Alertas/Exportar/XLS
EC 1.4 Opción DOCX	El usuario selecciona la opción DOCX.	El sistema muestra una ventana donde le brinda al usuario la opción de abrir el archivo exportado o guardarlo en una ubicación específica.	General/Seguridad/Ossec/Alertas/Exportar/DOCX

DCP 9. Eliminar alertas.

Tabla 23. DCP del requisito funcional eliminar alertas.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Eliminar alertas	Selecciona la opción de eliminar una alerta	Muestra un mensaje de confirmación. Y permite: - Aceptar. - Cancelar.	General/Seguridad/Ossec/Alertas/Eliminar
EC 1.2 Opción Aceptar	Selecciona la opción Aceptar.	Elimina la alerta. Regresa al listado de instituciones actualizado y muestra un mensaje de información.	General/Seguridad/Ossec/Alertas/Eliminar/Aceptar
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.	Regresa a la vista anterior.	General/Seguridad/Ossec/Alertas/Eliminar/Cancelar

DCP 10. Mostrar alertas.

Tabla 24. DCP del requisito funcional mostrar alertas.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción mostrar de una alerta	Selecciona la opción de mostrar de una alerta	Muestra los siguientes datos de la alerta: - Fecha. - ID de la regla. - Nivel - Ubicación - Descripción - Información Permite además: - Eliminar los datos del archivo. - Regresar al listado de archivos modificados.	General/Seguridad/Ossec/Alertas

EC 1.2 Opción de regresar al listado de las alertas	El usuario selecciona la opción de Regresar al listado.	Regresa al listado de las alertas	General/Seguridad/Ossec/Alertas/Mostrar/Regresar al listado
EC 1.3 Opción de eliminar el elemento	Selecciona la opción de Eliminar.	El sistema brinda la posibilidad de eliminar la institución. Ver: DCP_9_Eliminar_alertas.ods	General/Seguridad/Ossec/Alertas/Mostrar/Regresar al listado