



# **Universidad de las Ciencias Informáticas**

## **Facultad 6**

Sistema para la gestión de la información generada en las auditorías a la seguridad informática en la Universidad de las Ciencias Informáticas.

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

**Autor:**

Jorge Luis Hernández Suárez

**Tutores:**

**Msc. Grethell Castillo Reyes**

**Ing. Aime Esther Guzmán Ramírez**

La Habana, junio del 2016

# Declaración de autoría

## **Declaración de autoría**

---

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Jorge Luis Hernández Suárez

\_\_\_\_\_  
Firma del Autor

Msc. Grethell Castillo Reyes

\_\_\_\_\_  
Firma del Tutor

Ing. Aimé Esther Guzmán Ramírez

\_\_\_\_\_  
Firma del Tutor

## Datos de Contacto

---

### Tutores

**Nombre:** Grethell

**Apellidos:** Castillo Reyes

**Centro de trabajo:** Universidad de las Ciencias Informáticas

**Correo electrónico:** [gcreyes@uci.cu](mailto:gcreyes@uci.cu)

**Año de graduación:** 2012

**Título de la especialidad:** Máster en Informática Aplicada

**Nombres:** Aimé Esther

**Apellidos:** Guzmán Ramírez

**Centro de trabajo:** Universidad de las Ciencias Informáticas

**Correo electrónico:** [aguzman@uci.cu](mailto:aguzman@uci.cu)

**Año de graduación:** 2010

**Título de la especialidad:** Ingeniera en Ciencias Informáticas

### Agradecimientos

---

*Un agradecimiento especial a mis padres, por darme la fuerza y el apoyo para concluir este trabajo de tesis.*

*A mis hermanos por alentarme para que luche por mis objetivos y por darme apoyo psicológicamente.*

*A mis tutoras porque además de guiarme en esta última etapa y ayudarme incondicionalmente, me brindaron su apoyo y conocimiento, me enseñaron muchas cosas y sobre todo por haber asumido esto que hoy estamos logrando.*

*A mis amigos que siempre han estado conmigo en los momentos buenos y malos, y por soportarme en los momentos de mucho stress.*

## Dedicatoria

---

*Le dedico el presente trabajo de diploma a todos los que de una forma u otra me ayudaron a la realización del mismo, especialmente a mis padres que me han apoyado en todo lo posible y me han motivado a seguir y luchar por mis objetivos.*

## Frases

---

*“Dios nos da la gracia para aceptar con serenidad las cosas que no se pueden modificar, valor para cambiar las cosas que se deben cambiar, y sabiduría para diferenciar unas de las otras”.*

*“La vida es una obra de teatro que no permite ensayos. Por eso, canta, ríe, baila, llora y vive intensamente cada momento de tu vida antes que el telón baje y la obra termine sin aplausos.”*



*Charles Spencer (Charlie) Chaplin*

# Resumen

## Resumen

---

Las auditorías a la seguridad informática son un proceso que está implícito en cualquier institución y además juegan un papel primordial para garantizar la seguridad en la red. En la Universidad de las Ciencias Informáticas se realiza esta actividad con el Sistema de Gestión de Software y Hardware (GRHS). Sin embargo no es posible corregir los problemas detectados desde el propio sistema, ni realizar instalaciones de programas de forma remota o reportes de incidencias. La presente investigación plantea una solución a la problemática a partir de la implementación del Sistema de control de las políticas de seguridad informática. El producto brinda una serie de vistas que facilitan la toma de decisiones oportunas, el monitoreo constante y retroactivo y la solución inmediata a los problemas detectados. Para la construcción del sistema se utilizó PHP 5 como lenguaje de programación, Symfony 2.7 como marco de trabajo, como gestor de base de datos Postgresql 9.3, como servidor web Apache 2.4 y Visual Paradigm como herramienta CASE. Todo el proceso de desarrollo estuvo guiado por la metodología ágil XP.

**Palabras claves:** Auditoría informática, gestión de la información, información, seguridad informática

# Abstract

## Abstract

---

*Computer audits to security are a process which is implicit in any institution and also play a key for ensuring network security role. At the University Informatics Science this activity is done with System Management Software and Hardware (GRHS). However it is not possible to correct the problems identified from the system itself, or perform software installations remotely or incidents reports. This research proposes a solution to the problem from the implementation of the control system security policies. The product provides a series of views that facilitate timely decision making, and retroactive constant monitoring and immediate solution to the problems identified. To develop the system was used PHP 5 as programming language, Symfony 2.7 as framework, Postgresql 9.3 as data base manager, Apache 2.4 as web server and Visual Paradigm as CASE tool. The development process was guided with the agile methodology XP.*

*Key Words: Computer audit, information, information management, informatic security*

# Índice

## Índice

---

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 .....</b>	<b>5</b>
1.1 INTRODUCCIÓN .....	5
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DE LA PROPUESTA DE SOLUCIÓN .....	5
1.3 CARACTERIZACIÓN DEL OBJETO DE ESTUDIO .....	6
1.4 SISTEMAS EXISTENTES PARA LA REALIZACIÓN DE AUDITORÍAS INFORMÁTICAS .....	7
<i>Net Support DNA</i> .....	7
<i>ACL</i> .....	7
<i>Nessus</i> .....	8
<i>OCS Inventory NG 2.0.3</i> .....	8
<i>GRHS</i> .....	8
1.5 SELECCIÓN DE TÉCNICAS Y HERRAMIENTAS.....	9
<i>Metodología de desarrollo</i> .....	9
<i>Selección de la metodología a emplear.</i> .....	9
Crystal Clear .....	9
Scrum .....	10
XP (eXtreme Programming) .....	10
<i>Lenguaje de programación</i> .....	11
PHP.....	11
<i>Herramientas de desarrollo</i> .....	12
<i>Framework de desarrollo</i> .....	12
<i>Sistema Gestor de Base de Datos</i> .....	14
PostgreSQL.....	14
<i>Servidor Web</i> .....	15
Apache .....	16
<i>CASE</i> .....	16
Visual Paradigm.....	16
1.6 CONCLUSIONES PARCIALES .....	17
<b>CAPÍTULO 2 .....</b>	<b>18</b>
1.1 INTRODUCCIÓN.....	18
1.2 PLANIFICACIÓN DE ROLES Y RESPONSABILIDADES METODOLOGÍA XP .....	18
1.3 MODELO DE DOMINIO.....	19
Descripción de las clases del dominio .....	20
1.4 LISTA DE RESERVA DEL PRODUCTO.....	20
1.5 REQUISITOS NO FUNCIONALES .....	22
1.6 HISTORIAS DE USUARIO Y TAREAS DE INGENIERÍA .....	22
1.7 PLAN DE RELEASE .....	39
1.8 ARQUITECTURA DE SOFTWARE .....	40
<i>Estilos arquitectónicos</i> .....	41
<i>Patrones de Diseño</i> .....	41
GRASP .....	42
GOF .....	42
1.9 MODELADO DEL DISEÑO .....	43
<i>Diagrama de Clases del Diseño</i> .....	43
Descripción del Diagrama de Diseño.....	44
<i>Modelo de Datos</i> .....	44
<i>Diagrama de componentes</i> .....	45

# Índice

<i>Diagrama de Despliegue</i> .....	46
1.10 CONCLUSIONES PARCIALES .....	47
<b>CAPÍTULO 3</b> .....	<b>48</b>
2.1 INTRODUCCIÓN .....	48
2.2 CASOS DE PRUEBAS DE ACEPTACIÓN .....	48
2.3 SISTEMA DESARROLLADO .....	52
2.4 CONCLUSIONES PARCIALES .....	53
<b>CONCLUSIONES</b> .....	<b>55</b>
<b>RECOMENDACIONES</b> .....	<b>56</b>
<b>REFERENCIAS</b> .....	<b>57</b>
<b>BIBLIOGRAFÍA</b> .....	<b>58</b>
<b>ANEXOS</b> .....	<b>60</b>

# Índice de figuras

## Índice de figuras

---

FIG #1.	MODELO DE DOMINIO (DEL AUTOR).....	19
FIG #2.	DIAGRAMA DE CLASES DEL DISEÑO ADICIONAR PCS (DEL AUTOR) .....	44
FIG #3.	MODELO DE DATOS (DEL AUTOR) .....	45
FIG #4.	DIAGRAMA DE COMPONENTES (DEL AUTOR) .....	46
FIG #5.	DIAGRAMA DE DESPLIEGUE (DEL AUTOR).....	46
FIG #6.	IMAGEN DEL SISTEMA QUE MUESTRA LA PANTALLA DE ACCESO (DEL AUTOR) .....	52
FIG #7.	IMAGEN DEL SISTEMA QUE MUESTRA UN LISTADO DE LOS LABORATORIOS DEL CENTRO (DEL AUTOR).....	53
FIG #8.	IMAGEN DEL SISTEMA QUE MUESTRA LA PANTALLA DE BIENVENIDA (DEL AUTOR) .....	53

# Índice de tablas

## Índice de tablas

---

TABLA # 1.	PLANIFICACIÓN POR ROLES (DEL AUTOR) .....	18
TABLA # 2.	LISTA DE RESERVA DEL PRODUCTO (DEL AUTOR) .....	20
TABLA # 3.	REQUISITOS NO FUNCIONALES (DEL AUTOR).....	22
TABLA # 4.	HISTORIA DE USUARIO: AUTENTICAR USUARIO (DEL AUTOR) .....	22
TABLA # 5.	HISTORIA DE USUARIO: GESTIONAR ESCANEADO (DEL AUTOR) .....	23
TABLA # 6.	HISTORIA DE USUARIO: EXPORTAR REPORTES PDF (DEL AUTOR) .....	24
TABLA # 2.	HISTORIA DE USUARIO: REALIZAR FILTRADO (DEL AUTOR).....	25
TABLA # 3.	HISTORIA DE USUARIO: GENERAR REPORTE (DEL AUTOR).....	26
TABLA # 4.	HISTORIA DE USUARIO: GENERAR NOTIFICACIÓN (DEL AUTOR) .....	27
TABLA # 5.	HISTORIA DE USUARIO: GESTIONAR LABORATORIO (DEL AUTOR).....	28
TABLA # 6.	HISTORIA DE USUARIO: GESTIONAR DATOS DE AUDITORÍA (DEL AUTOR) .....	30
TABLA # 7.	HISTORIA DE USUARIO: DESCONECTAR USUARIO (DEL AUTOR) .....	31
TABLA # 8.	HISTORIA DE USUARIO: GESTIONAR USUARIO (DEL AUTOR) .....	32
TABLA # 9.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 1_ AUTENTICAR USUARIO (DEL AUTOR).....	33
TABLA # 10.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 2_ REALIZAR ESCANEADO DE LAS MÁQUINAS DE UN ÁREA DETERMINADA (DEL AUTOR).....	33
TABLA # 11.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 2_ INSTALAR PAQUETES EN LAS ESTACIONES CLIENTES (DEL AUTOR) .....	34
TABLA # 12.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 2_ REALIZAR LA INSTALACIÓN DE PAQUETES DE FORMA REMOTA (DEL AUTOR).....	34
TABLA # 13.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 2_ ESTABLECER LAS POLÍTICAS DE SEGURIDAD QUE SE DESEAN CHEQUEAR AL REALIZAR EL ESCANEADO (DEL AUTOR).....	35
TABLA # 14.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 2_ REGISTRAR CAUSAS DE LAS VULNERABILIDADES (DEL AUTOR) .....	35
TABLA # 15.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 2_ PLANIFICAR ESCANEADO AUTOMÁTICO (DEL AUTOR) 35	35
TABLA # 16.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 7_ NOTIFICAR AL ASESOR DE SEGURIDAD EL FIN DEL ESCANEADO (DEL AUTOR) .....	36
TABLA # 17.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 7_ ENVIAR POR CORREO RESULTADO DEL ESCANEADO SEGÚN LA PLANIFICACIÓN (DEL AUTOR) .....	36
TABLA # 18.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 7_ NOTIFICAR ADVERTENCIA A LOS USUARIOS QUE INCIDAN EN UNA FALTA (DEL AUTOR) .....	37
TABLA # 19.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 11_ NOTIFICAR ADVERTENCIA A LOS USUARIOS QUE INCIDAN EN UNA FALTA (DEL AUTOR) .....	37
TABLA # 20.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 11_ ELIMINAR USUARIO (DEL AUTOR) .....	37
TABLA # 21.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 11_ EDITAR USUARIO (DEL AUTOR) .....	38

## Índice de tablas

TABLA # 22.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 11_ CONFIGURAR USUARIO (DEL AUTOR)...	38
TABLA # 23.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 11_ ASIGNAR PERMISOS A FUNCIONALIDADES POR ROLES (DEL AUTOR) .....	38
TABLA # 24.	TAREAS DE INGENIERÍA ASOCIADA A LA HISTORIA DE USUARIO 11_ ELIMINAR PERMISOS A FUNCIONALIDADES POR ROLES (DEL AUTOR) .....	39
TABLA # 25.	PLAN DE RELEASE (DEL AUTOR) .....	39
TABLA # 26.	CASO DE PRUEBA HISTORIA DE USUARIO 1-1_ AUTENTICAR USUARIO (DEL AUTOR) .....	48
TABLA # 27.	CASO DE PRUEBA HISTORIA DE USUARIO 2-1_ REALIZAR ESCANEADO DE LAS MÁQUINAS DE UN ÁREA DETERMINADA (DEL AUTOR) .....	49
TABLA # 28.	CASO DE PRUEBA HISTORIA DE USUARIO 2-4_ ESTABLECER LAS POLÍTICAS DE SEGURIDAD QUE SE DESEAN CHEQUEAR AL REALIZAR EL ESCANEADO (DEL AUTOR).....	49
TABLA # 29.	CASO DE PRUEBA HISTORIA DE USUARIO 2-4_ EXPORTAR REPORTES PDF (DEL AUTOR) .....	50
TABLA # 30.	CASO DE PRUEBA HISTORIA DE USUARIO 2-4_ REALIZAR FILTRADO (DEL AUTOR).....	50
TABLA # 31.	CASO DE PRUEBA HISTORIA DE USUARIO 2-4_ GESTIONAR LABORATORIO (DEL AUTOR).....	51
TABLA # 32.	CASO DE PRUEBA HISTORIA DE USUARIO 2-4_ DESCONECTAR USUARIO (DEL AUTOR) .....	51

## Introducción

---

La Informática, hoy en día, está vinculada a la gestión integral de la empresa, por ello las normas y estándares propiamente informáticos deben estar sometidos a sus particularidades. En consecuencia, las organizaciones informáticas forman parte de lo que se ha denominado *management*<sup>1</sup>. Cabe aclarar que la informática no gestiona propiamente la empresa, sino que ayuda en el proceso de toma de decisiones. Con el inmenso auge de esta ciencia y la constante necesidad de estar interconectados surge como prioridad establecer requerimientos que rijan la seguridad de la información.

La seguridad informática es el área que se enfoca en la protección de la infraestructura computacional y todas sus relaciones, especialmente la información contenida circulante. En las empresas con el fin de garantizar el uso apropiado de las tecnologías se establecen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes para minimizar los posibles riesgos (Fernández G, 2013). Debido a la importancia de proteger los activos de las organizaciones y monitorear el cumplimiento de las pautas definidas se llevan a cabo las Auditorías a la Seguridad Informática.

La auditoría persigue como objetivo principal realizar una evaluación exhaustiva de las actividades ejecutadas en la organización. Esto implica la detección de las debilidades y la identificación de amenazas como una definición clara de las causas de estas situaciones para posteriormente tomar las previsiones necesarias para disminuir, trasladar o eliminar dichas amenazas y fortalecer sus áreas débiles. De esta forma se contribuye a la seguridad de una red de computadoras.

La constante evolución del mundo empresarial, el crecimiento diario de la cantidad y la complejidad de las operaciones, limitan al propietario y al ejecutivo mismo la capacidad de controlar eficaz y eficientemente todas las actividades que suceden a su alrededor y que son necesarias para continuar trabajando. La Universidad de las Ciencias Informáticas (UCI), no queda exento de dicha problemática. Más aun cuando se encuentra en un proceso de madurez y crecimiento con respecto a este tema. En la actualidad cuenta con un Gestor de Recursos de Hardware y Software (GRHS) (Ramos Betancourt, y otros, 2012): "*una solución tecnológica integral para los procesos de captura, análisis y consulta de la información*", desarrollado en la propia universidad.

Sin embargo, la no existencia de una base de datos que almacene toda la información para ejecutar futuras consultas acarrea complicaciones para realizar búsquedas de incidencias en fechas pasadas y la

---

1

## Introducción

obtención de un resumen histórico de manera rápida. Otro inconveniente es la complejidad a la hora de realizar estadísticas de la información obtenidas de las auditorías a las estaciones de trabajo como: cuáles han sido las personas que más han violado las políticas de la organización, o sea, cuántas máquinas hoy no cuentan con el estándar definido. También representa una dificultad no poder realizar acciones ante incidencias de forma automática o no contar con una visualización gráfica del escaneo realizado.

En el análisis de todo lo antes expuesto se identifica como **problema a resolver**: ¿Cómo contribuir a elevar los niveles de centralización e integridad de la información asociada a los resultados obtenidos en las auditorías de seguridad informática en la Universidad de las Ciencias Informáticas?

Teniendo como **objeto de estudio**: El proceso de gestión de la información generada en las auditorías a la seguridad informática y el **campo de acción** se ve reflejado en el proceso de gestión de la información generada en las auditorías a la seguridad informática en la Universidad de las Ciencias Informáticas. Para darle solución al problema se define como **objetivo general**: desarrollar una aplicación informática para la gestión de las auditorías informáticas en la Universidad de las Ciencias Informáticas.

Como **preguntas científicas** se encuentran:

- ❖ ¿Cuáles son las bases que definirán la fundamentación teórica que demostrará el por qué de la realización de la investigación y la propuesta de solución?
- ❖ ¿Cómo guiar el diseño de la propuesta de solución a partir de la caracterización del proceso de gestión de la información generada a partir de las auditorías de seguridad informática realizadas en la Universidad de las Ciencias Informáticas?
- ❖ ¿Cómo caracterizar el desarrollo de la propuesta de solución y los resultados de las pruebas efectuadas para validar las funcionalidades implementadas?

Para el cumplimiento de los objetivos se plantearon las siguientes **tareas de la investigación**:

- ❖ Determinación de los conceptos asociados al dominio del problema de la investigación.
- ❖ Identificación de aplicaciones informáticas similares existentes en el ámbito nacional e internacional para la gestión de la información en auditorías de seguridad informática.
- ❖ Selección de tecnologías y herramientas a utilizar en el proceso de desarrollo del software.
- ❖ Caracterización del proceso de gestión de la información en las auditorías de seguridad informática en la Universidad de las Ciencias Informáticas.

## Introducción

- ❖ Caracterización de la propuesta de solución.
- ❖ Realización del diseño de la solución propuesta.
- ❖ Implementación de las funcionalidades de la solución propuesta.
- ❖ Validación de la solución mediante la realización de pruebas de software.

### **Resultados esperados:**

- ❖ El Sistema para la gestión de la información generada en las auditorías a la seguridad informática en la Universidad de las Ciencias Informáticas.
- ❖ El documento de investigación que avale la realización del sistema.
- ❖ Los artefactos generados durante el desarrollo de la investigación.

Los **métodos científicos** utilizados durante la investigación fueron:

### **Métodos teóricos:**

- ❖ El analítico – sintético se emplea para determinar conceptos asociados a la problemática planteada, así como en el estudio de las soluciones existentes y la selección de las herramientas y técnicas permitiendo establecer las relaciones que existen entre los elementos de cada contexto, además en el análisis y síntesis de la documentación utilizada para la bibliografía.

### **Métodos empíricos:**

- ❖ La observación se emplea para identificar algunas características en el proceso de gestión de información de auditorías a las estaciones de trabajo como las técnicas, herramientas y actividades empleadas y personas que intervienen.

El documento de investigación estará estructurado de la siguiente manera:

### **Capítulo 1. Fundamentación teórica.**

Se describen los términos y conceptos más importantes asociados al proceso de gestión de la información obtenida en las auditorías de seguridad informática, así como elementos esenciales de las aplicaciones existentes para realizar dicho proceso, sus ventajas y desventajas. Se exponen y valoran las tendencias y tecnologías actuales para el desarrollo de sistemas informáticos, fundamentándose las razones de la selección de la metodología, lenguaje de programación, servidor y gestor de base de datos a utilizar para la construcción de la propuesta de solución.

# Introducción

## **Capítulo 2. Características del sistema**

Son descritas las características de la propuesta de solución para sentar las bases para su correspondiente elaboración. Se analiza y diseña tanto el sistema como el negocio, a través de la metodología que se selecciona para guiar el camino de la implementación. Por último, quedan plasmados los elementos de la fase de diseño de la propuesta de solución, así como sus respectivos artefactos.

## **Capítulo 3. Validación y pruebas**

Se plantea la construcción de la solución propuesta en el Capítulo 2 a través de las disciplinas definidas por la metodología seleccionada. Además, se realizan las pruebas al sistema y se genera cada uno de los artefactos correspondientes. Como último elemento se muestra el resultado de la investigación, así como las características sobresalientes del producto.

## Capítulo 1

---

### 1.1 Introducción

En este capítulo se abordan diferentes conceptos asociados al problema planteado, que son útiles para un mejor entendimiento de la información, o sea, se profundiza en el objeto de estudio. Se realiza un estudio del estado del arte de algunas herramientas y tecnologías usadas actualmente, se describen algunas de las soluciones existentes en el mundo, enmarcándose en la situación problemática. Se realiza la selección de las herramientas a utilizar para el desarrollo del sistema.

### 1.2 Conceptos asociados al dominio de la propuesta de solución

**Información:** *"Forma social de existencia del conocimiento consolidada en una fuente determinada (...)*  
*La información es un recurso, un valor o un activo igual que cualquier otro y que como recurso tiene características que lo hacen similar o diferente a los demás, o sea, que se adquiere a un costo, posee valores, requiere del control de sus costos, tiene un ciclo de vida, puede procesarse y existen sustitutos para informaciones específicas"* (Aja Quiroga, 2002).

**Gestión de información:** *"Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada"* (Aja Quiroga, 2002).

El autor coincide con la definición brindada (Ponjuán, 1998) *"Es el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. Tiene como elemento básico, la gestión del ciclo de vida de este recurso y se desarrolla en cualquier organización. En particular, también se desarrolla en unidades especializadas que manejan este recurso en forma intensiva, llamadas unidades de información"*.

**Sistemas de gestión de información ERP:** *"es un sistema de información integral que soporta los procesos y la administración de recursos de una organización. El sistema consta de varias aplicaciones cuya integración soporta la administración de la cadena de suministro y las operaciones del día a día"* (Alvarado B, 2006).

O sea, es un sistema de ayuda a la toma de decisiones a partir de información de variadas fuentes que se encuentra ubicada en una única base de datos.

## Capítulo 1: Fundamentación teórica

**Seguridad informática:** *"conjunto de métodos y herramientas destinados a proteger la información y por ende los sistemas informáticos ante cualquier amenaza, un proceso donde además participan personas"* (Seguridad Informática y Cartografía, 2006).

*"La seguridad informática es la protección a los elementos de valor, llamado activos de una computadora o sistema informático, existen muchos tipos de activos, hardware, software, datos, personas, procesos o combinación de ellos, para determinar qué proteger, primero identificamos lo que tiene valor y para quién"* (Pfleeger, y otros, 2015).

Para la mayoría de los expertos el concepto de seguridad en la informática es utópico porque no existe un sistema 100% seguro. Para que un sistema se pueda definir como seguro debe tener estas cuatro características (Seguridad Informática y Cartografía, 2006):

- ❖ **Integridad:** los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.
- ❖ **Confidencialidad:** la información o los activos informáticos son accedidos solo por las personas autorizadas para hacerlo.
- ❖ **Disponibilidad:** los activos informáticos son accedidos por las personas autorizadas en el momento requerido.
- ❖ **Irrefutabilidad (No repudio):** el uso y/o modificación de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar dicha acción.

### 1.3 Caracterización del Objeto de Estudio

Caracterización actual sobre el proceso de gestión de la información obtenida a partir de las auditorías a la seguridad informática en la Universidad de las Ciencias Informáticas.

Para comprobar la correcta implementación de las políticas de Seguridad Informática en la UCI se emplea el GRHS, un importante sistema que realiza un levantamiento de requisitos de tipo *Software y/o Hardware* y muestra un informe con todo lo relacionado a las estaciones de trabajo. Dentro del monitoreo realizado se chequea la instalación y actualización del antivirus, el dominio al que pertenece la máquina, la presencia del corta fuego y muestra un listado de características como la capacidad del HDD, la RAM, las sesiones existentes y la dirección IP. Sin embargo, existen una serie de requisitos que no son contemplados por la herramienta:

## Capítulo 1: Fundamentación teórica

- ❖ La posibilidad de tener una consola interna en el sistema para la realización de determinadas acciones.
- ❖ Una vez comprobada una vulnerabilidad de software darle solución a través de la red (por ejemplo: si no existe ningún antivirus, ejecutar automáticamente la instalación a través de la red).
- ❖ Generar reporte de incidencias.
- ❖ Enviar por correo resultados del escaneo según la planificación realizada.
- ❖ Obtener los resultados de los reportes en formato PDF para análisis en reuniones.
- ❖ Verificar la correcta implementación de todas las políticas de seguridad informática.

Todos estos elementos son importantes para realizar una adecuada gestión de la información ya que brinda un mayor control y una rápida localización de elementos que apoyen la toma de decisiones, representado un ahorro de recursos y centralización de la información en las diversas estructuras de la organización.

### 1.4 Sistemas existentes para la realización de auditorías informáticas

#### **Net Support DNA**

*"Es una completa solución modular que ofrece inventario de hardware, software y gestión de licencias. Presenta avisos detallados y personalizables, como la medición y el control de uso de aplicaciones y de Internet. Posee actualización automática por consulta de distribución de software a través de una LAN (en español Red de Área Local) o una WAN (en español Red de Área Amplia). Proporciona una puerta de enlace de comunicaciones integrada que le permite interactuar con sus activos con toda seguridad, incluso a través de Internet y en cualquier lugar, todo ello sin necesidad de una VPN (en español Red Virtual Privada), ni cambios en la red existente o en la configuración del cortafuego. Se integra con Active Directory y cuenta con Helpdesk basado en TIL (en español Biblioteca de Infraestructuras de Tecnologías de Información). Esta herramienta corre bajo sistema operativo Windows"* (Torres Matos, y otros, 2011).

#### **ACL**

ACL es un producto muy poderoso y fácil de usar que posee múltiples aplicaciones debido a su combinación, acceso a datos, análisis y generación de reportes. Entre la gran variedad de análisis que posibilita se encuentran (The Eniac Corporation, 2015):

- ❖ Identificación de problemas de control.
- ❖ Evaluación de procesos y cumplimiento de estándares.

## Capítulo 1: Fundamentación teórica

- ❖ Análisis y detección de fraudes.
- ❖ Identificación de excepciones y anomalías.
- ❖ Localizar errores y posibles irregularidades.

### **Nessus**

Es un escáner de vulnerabilidades que funciona mediante un proceso de alta velocidad por el que encuentra los datos sensibles y trabaja con la auditoría de configuraciones y el perfil activo. Es de gran utilidad para los dispositivos de uso personal pero también para las grandes empresas que se manejan con equipos conectados en red, pues trabaja mediante el uso de zonas desmilitarizadas (DMZ, por sus siglas en inglés), capaz de auditar y analizar tanto los sistemas en red como los individuales a través del rastreo de archivos y parches de seguridad. Se adapta a diferentes plataformas y sistemas operativos. Opcionalmente, los resultados del escaneo pueden ser exportados como informes en varios formatos, como archivos en texto plano, XML y HTML. Los resultados también pueden ser guardados en una base de conocimiento para referencia en futuros escaneos de vulnerabilidades (Gómez, 2015).

### **OCS Inventory NG 2.0.3**

Herramienta multiplataforma que realiza el reporte de los inventarios y genera alarmas a través del correo electrónico. Como principal desventaja no permite el inventario a través de acceso remoto (Ramos Betancourt, y otros, 2012). *"También en el mapa de red que presenta, las estaciones de trabajo no están distribuidas por locales, sino que son mostradas en forma de listado, haciendo más compleja su gestión"* (Torres Matos, y otros, 2011).

### **GRHS**

Propuesta nacional desarrollada por la UCI, es una *"Aplicación cliente-servidor cuyo objetivo principal es centralizar en el servidor la información de hardware y software instalados en las estaciones clientes. La aplicación cliente se encarga de obtener la información de hardware y software instalados y así envía los datos recolectados al servidor, determinado si hubo cambios a partir del inventario obtenido anteriormente"* (Torres Matos, y otros, 2011). Es un sistema libre, multiplataforma y su principal función es que cuenta con un sistema de alertas.

El estudio de las herramientas mencionadas demuestra la amplia gama de programas dedicados a este fin, sin embargo, las necesidades expuestas al inicio de la investigación no son resueltas en su totalidad por ninguno de los sistemas. Net Support DNA no ejecuta acciones correctivas ante la detección de

# Capítulo 1: Fundamentación teórica

vulnerabilidades. ACL funciona solamente en sistema operativo Windows, dando contraste al proceso de informatización de la sociedad cubana. Net Support DNA y ACL son distribuidos bajo licencia propietaria, lo que implica un gasto adicional. Por su parte OCS Inventory NG no realiza el inventario desde un acceso remoto y el mapa de red que muestra no es de forma gráfica sino mediante un listado dificultando la gestión de la información. Por su parte del GRHS es el sistema que utiliza actualmente la UCI, pero presenta una serie de limitantes que fueron señalados al comienzo de la investigación.

## 1.5 Selección de técnicas y herramientas

### Metodología de desarrollo

*“Un proceso de software detallado y completo suele denominarse “Metodología”. Las metodologías se basan en una combinación de los modelos de proceso. Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucradas, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas) actividades del proceso de desarrollo”* (Departamento de Ingeniería de Software, 2007-2008).

Es importante destacar que se pueden clasificar en ágiles o pesadas y que son las características del cliente y equipo de trabajo los que requieren el empleo de una u otra. Se debe hacer uso de una metodología pesada cuando la captura de requisitos es guiada por una entrevista formal al cliente, los integrantes del grupo de trabajo forman un conjunto extenso y cuando existe la necesidad de documentar detalladamente cada elemento del negocio. Si por el contrario son pocas las personas involucradas en la creación del producto, los requisitos varían constantemente y el tiempo es limitado lo mejor es optar por una metodología ágil, caso ante el que se encuentra el autor de la presente investigación.

### Selección de la metodología a emplear.

#### **Crystal Clear**

Son una serie de metodologías que bajo el nombre de Crystal “presentan un enfoque ágil, con gran énfasis en la comunicación, y con cierta tolerancia que la hace ideal en los casos en que sea inaplicable la disciplina requerida por XP. (...) *Crystal maneja iteraciones cortas con feedback frecuente por parte de los usuarios/clientes, minimizando de esta forma la necesidad de productos intermedios”* (Calderón, y otros, 2007).

## Capítulo 1: Fundamentación teórica

*"Cockburn (creador) confiesa que cuando imaginó a Crystal Clear pensaba proporcionar un método ligero; comparado con XP, sin embargo, Cristal Clear resultó muy pesado, sin embargo, es más fácil de aprender e implementar"* (Calderón, y otros, 2007).

### **Scrum**

Es un marco de trabajo que brinda la flexibilidad de auto-gestionarte para tratar de eliminar la complejidad que genera la imposición de tareas, siendo el propio interesado en el desarrollo quien elige la tarea que considera es capaz de solucionar en menos tiempo. Permite aplicar la combinación de otras prácticas ingenieriles o reglas ya definidas. Consiste en la división del trabajo en pequeñas partes que formen una versión estable y verificable de la aplicación, que es implementada en un tiempo acordado por el equipo que no puede exceder las 4 semanas (Pérez, 2015). De esta forma se va incrementando el producto y es mucho más fácil adaptarse a nuevos cambios, en el próximo comienzo.

### **XP (eXtreme Programming)**

*"XP se centra en la continua retroalimentación entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como una metodología especialmente adecuada para proyectos con requisitos muy cambiantes e imprecisos, donde existe un alto riesgo técnico. Como metodología pragmática, recoge las que considera mejores prácticas para el desarrollo software, cuya aplicación disciplinada pretende disminuir la curva exponencial del costo del cambio a lo largo del proyecto. Se trata de doce prácticas: el juego de la planificación, entregas pequeñas, metáfora, diseño simple, pruebas, refactorización, programación en parejas, propiedad colectiva del código, integración continua, 40 horas por semana, cliente in-situ y estándares de programación"* (Rodríguez González, 2008).

Fases establecidas por la metodología XP según (Amaro Calderón, y otros, 2007) y (Letelier, y otros, 2003):

- ❖ Fase de Exploración: incluye todo el proceso de confección de las historias de usuarios, así como el entendimiento por parte del equipo de las necesidades del usuario. También se seleccionan las tecnologías apropiadas para el desarrollo y se plantea la arquitectura del sistema a desarrollar.
- ❖ Fase de Planificación de la entrega: momento en que el cliente establece la prioridad de las historias de usuario y el desarrollador estima el esfuerzo requerido para la implementación, lo que unido a la velocidad del equipo permite la planificación de las entregas y su contenido.
- ❖ Fase de Iteraciones: proceso mediante el cual el entregable pasa por varias iteraciones antes de entrar a la fase de producción

## Capítulo 1: Fundamentación teórica

- ❖ Fase de Producción: se realizan pruebas que garantizan que el producto a entregar es lo requerido por el usuario. Es el momento de revisar los cambios que propone el usuario para incluirlos en este momento o para posterior implementación.
- ❖ Fase de Mantenimiento: es el proceso de soporte que brinda el equipo cuando el producto se despliega y se encuentra en su ambiente real.
- ❖ Fase de Muerte del Proyecto: momento que llega cuando se ejecutaron todas las historias de usuario, dando paso a generar la documentación asociada del cierre del proyecto.

### **Roles que intervienen:**

- ❖ Encargado de seguimiento
- ❖ Gestor
- ❖ Consultor
- ❖ Programador
- ❖ Cliente
- ❖ Entrenador
- ❖ Probador (*Tester*)

Según el criterio del autor XP es la opción ideal para guiar su desarrollo ya que establece una serie de artefactos que dejan evidencia de los procesos llevados a cabo durante el presente trabajo, de forma tal que quede todo debidamente documentado para futuras versiones. Otro motivo es que para la elaboración se cuenta con poco tiempo y se tiene un conocimiento básico XP, por lo que si la elección fuese otra habría que invertir tiempo en aprender una nueva metodología. Además se basa en la retroalimentación continua en el cliente y el equipo que implementa la solución, trayendo como ventaja la satisfacción con el producto final.

### **Lenguaje de programación**

#### ***PHP***

La solución que se pretende desarrollar es una aplicación web, por la experiencia alcanzada en años precedentes y por las características que se mencionan a continuación el autor de la investigación opta por el lenguaje PHP.

Es un lenguaje de programación del lado del servidor dirigido a la creación de páginas web. Lenguaje de programación procedural con una sintaxis similar a la del lenguaje C, aunque actualmente puede utilizarse una sintaxis de programación orientada a objetos similar a la de Java.

Características principales de PHP (The PHP Group, 2016):

- ❖ Lenguaje de libre distribución por lo que es una alternativa viable para la sociedad cubana.

## Capítulo 1: Fundamentación teórica

- ❖ Lenguaje orientado a objeto brindando las bondades de la herencia, el polimorfismo y el encapsulamiento.
- ❖ Autenticación a través de HTTP, incluyendo un mecanismo para el manejo de cookies, sesiones, formularios y manipulación de ficheros.
- ❖ Empleo de ficheros remotos: *"Por ejemplo, se puede emplear para abrir un fichero en un servidor web remoto, analizar la salida de los datos deseados, y usar dichos datos en una consulta a una base de datos, o simplemente para mostrarlos en un estilo que haga juego con el del sitio web"*.
- ❖ Manejo de conexiones a través de sus cuatro posibles estados: NORMAL, ABORTED, TIMEOUT y ABORTED y TIMEOUT.
- ❖ Conexiones persistentes a bases de datos.
- ❖ Mecanismo de Recolección de basura.
- ❖ No requiere definición del tipo de variable.
- ❖ Basta documentación en su sitio oficial, con definiciones y ejemplos prácticos que le brindan valor agregado.
- ❖ Permite la conexión con la mayoría de los sistemas gestores de base de datos.

### Herramientas de desarrollo

Las herramientas de desarrollo de software permiten la creación de aplicaciones para sistemas concretos. Las más comunes incluyen técnicas de soporte para la detección de errores de programación, la generación automática de código y características que facilitan y agilizan el desarrollo de soluciones de software. Frecuentemente incluyen también códigos de ejemplo, notas técnicas y documentación de soporte. Una vez seleccionado el lenguaje de programación a utilizar se realiza un estudio para conocer qué herramienta se adecua más a la implementación, de forma tal que brinde las mayores ventajas.

### Framework de desarrollo

Un *framework* es un marco de trabajo que permite personalizar sus componentes brindando la posibilidad de desarrollar otro *software* partiendo de él. Según (Torres Matos, y otros, 2011) es *"una estructura conceptual y tecnológica de soporte definida, con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Incluye soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto"*.

## Capítulo 1: Fundamentación teórica

Ventajas de utilizar un *framework*:

- ❖ Agilizar el proceso de desarrollo.
- ❖ Utilizar código ya existente (reutilización).
- ❖ Aplicación de patrones de diseño que le proporcionan al código mayor legibilidad y valor.
- ❖ Permite el acceso a librería y bibliotecas existentes.

En la actualidad existen un gran número de *frameworks* para PHP algunos de ellos son CakePHP, Zend Framework, Phalcon, Symfony, PhpBB, Laravel y Drupal.

El sitio oficial de PHP en la Universidad de las Ciencias Informáticas (php.uci.cu, 2016) ofrece un listado de donde se encuentran algunos:

*"Phalcon (...) se instala como un módulo de PHP y está escrito en C. Aunque está escrito en C se utiliza como si se usasen clases normales, por lo que no debemos de preocuparnos por nada que no sea PHP. Phalcon también incorpora MVC y una cantidad enorme de funcionalidades extra como: ODM para Mongo, cifrado, motor de plantilla, gestión de HTTP, paginación, constructor y validador de formulario".*

*"Zend Framework es un framework fácil de entender, bien documentado y que cumple varias funciones de forma segura, eficiente y modular (si así se desea). Incorpora el modelo vista controlador, además de tener sistema de rutas, sistema de generación de formularios y sistema de traducción. Incorpora una capa de compatibilidad con bases de datos de distintos proveedores, para hacer los proyectos más uniformes y fácilmente portables".*

*"Symfony se ha convertido en uno de los frameworks más usados en la actualidad para el desarrollo de proyectos de casi cualquier tamaño. (...) pone en práctica el patrón de diseño MVC, además tiene componentes complementarios como sistema de plantillas, traducción, rutas (REST), sistema de ficheros y una lista bastante interesante".* Otra de las características de Symfony es que sus componentes se encuentran desacoplados, características que han aprovechado algunos homólogos como PhpBB, Laravel y Drupal para incorporarlos a su núcleo (php.uci.cu, 2016).

En el desarrollo del presente trabajo se escoge como *framework* de desarrollo **Symfony2**, debido a que, además de ser código abierto, es rápido, flexible y fácil de aprender, permite a los desarrolladores construir sistemas webs gracias a la amplia comunidad de desarrollo y vasta documentación. *"Symfony está diseñado para optimizar el desarrollo de las aplicaciones, dada sus características. Para empezar,*

## Capítulo 1: Fundamentación teórica

*separa la lógica de negocio de la del servidor y la presentación. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web" (Potencier, y otros, 2008).*

Entre sus características están (Potencier, y otros, 2008):

- ❖ Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- ❖ Independiente del sistema gestor de bases de datos.
- ❖ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ❖ Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- ❖ Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- ❖ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ❖ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- ❖ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

### **Sistema Gestor de Base de Datos**

Un Sistema Gestor de Base de Datos (SGBD) es un programa que brinda la posibilidad de definir y guardar base de datos para su posterior acceso. El SGBD que se emplea entra en la clasificación de modelo relacional, debido a que guarda las relaciones creadas entre las tablas.

### **PostgreSQL**

*"Es un gestor de bases de datos orientadas a objetos (SGBDOO o ORDBMS en sus siglas en inglés) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también*

## Capítulo 1: Fundamentación teórica

*por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales" (Gibert Ginestà, y otros, 2016). PostgreSQL se distribuye bajo licencia BSD, puede funcionar en múltiples plataformas (en general, en todas las modernas basadas en Unix) y, a partir de la próxima versión 8.0 (actualmente en su segunda beta), también en Windows de forma nativa. Para las versiones anteriores existen versiones binarias para este sistema operativo, pero no tienen respaldo oficial. Para la interacción con el sistema gestor se usa PgAdmin III, que a través de una interfaz amigable permite manejar características como creación de tablas, funciones y vistas.*

Características principales de PostgreSQL:

- ❖ La API de acceso al SGBD se encuentra disponible en PHP.
- ❖ Su administración se basa en usuarios y privilegios.
- ❖ Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC.
- ❖ Es altamente confiable en cuanto a estabilidad se refiere.
- ❖ Puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética (soundex), etc.
- ❖ Control de concurrencia multi-versión, lo que mejora sensiblemente las operaciones de bloqueo y transacciones en sistemas multi-usuario.
- ❖ Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.
- ❖ Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.

### **Servidor Web**

*"Un servidor web es un simple servidor de archivos donde los clientes se dirigen a este mediante el protocolo HTTP para obtener recursos. Cuando el servidor web recibe una petición HTTP, extrae simplemente de la petición el nombre del recurso solicitado, lo busca en el disco y lo envuelve dentro de una respuesta HTTP para transmitirlo al cliente. Un servidor no realiza ningún tratamiento en el recurso antes de transmitirlo al cliente. Por lo tanto puede transmitir de manera indiferente a un cliente una página HTML, una imagen, un archivo de sonido o incluso un archivo ejecutable" (Díaz Díaz, 2013).*

# Capítulo 1: Fundamentación teórica

## **Apache**

Servidor web multiplataforma, de ahí su amplio empleo a nivel mundial. Crea un proceso padre y tantos hijos como peticiones sean realizadas para dar respuesta a las tareas del cliente. Lo antes expresado unido a que realiza la comunicación mediante el protocolo HTTP/ HTTPS impactan directamente en la reducción del ancho de banda necesario para responder las peticiones.

Ventajas:

- ❖ Versatilidad: permite interactuar con sistemas gestores de BD, lenguajes de programación y diferentes módulos existentes o creados por los usuarios.
- ❖ Precio: un único sistema para varios servidores.
- ❖ Configuración: una configuración del sistema para todos los servidores.
- ❖ Actualizaciones: solo una vez.

Desventajas:

- ❖ Fragilidad: Un fallo en el sistema ocasiona la no disponibilidad de los servidores.
- ❖ Configuración: Un fallo en la configuración puede hacer que no funcione el proceso.
- ❖ Actualizaciones: Si es necesario detener el sistema para actualizarlo se pararán todos los servidores.
- ❖ Seguridad: afectación de todos los servidores ante problemas de seguridad.

## **CASE**

Las herramientas CASE brindan soporte para desarrollar y mantener *software*. Son herramientas individuales que ayudan al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software.

Para el desarrollo del presente trabajo se selecciona la herramienta CASE Visual Paradigm.

### **Visual Paradigm**

Es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, con mejores características y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso,

## Capítulo 1: Fundamentación teórica

generar código desde diagramas y generar documentación. La herramienta CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Características :

- ❖ Soporte de UML versión 2.1.
- ❖ Diagramas de Procesos de Negocio.
- ❖ Ingeniería inversa.
- ❖ Modelo a código, diagrama a código.
- ❖ Diagramas de flujo de datos.
- ❖ Soporte ORM.
- ❖ Generación de bases de datos.
- ❖ Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ❖ Distribución automática de diagramas.

### 1.6 Conclusiones parciales

En la elaboración del capítulo se arribó a la conclusión de que en una organización efectuar auditorías a la seguridad informática es primordial para garantizar la integridad, confidencialidad y disponibilidad del activo más importante. Tras una búsqueda exhaustiva de los sistemas existentes se concluye que estos no brindan una solución a la problemática planteada al principio de la investigación. Dicho estudio, además, sirve de apoyo para tener un mejor entendimiento de este tipo de aplicaciones y una guía de cómo realizar una propuesta solución. Además, se decide emplear XP como metodología para guiar el proceso de desarrollo.

### Capítulo 2

---

#### 1.1 Introducción

En el presente capítulo se modela el dominio para una mayor comprensión de la situación existente, además se enumeran los requisitos funcionales (RF) y los requisitos no funcionales (RNF) del sistema. Se realiza la priorización de los requisitos y diseña la solución más simple. A través del diseño se contribuye a su comprensión y la modelación asesora la implementación.

#### 1.2 Planificación de Roles y Responsabilidades metodología XP.

Tabla # 1. Planificación por roles (del autor)

Rol	Responsabilidades	Responsable
Encargado de seguimiento	Verifica el cumplimiento de las responsabilidades.  Responsable de la retroalimentación.  Da seguimiento al progreso de las iteraciones.	Grethell Castillo Reyes  Aimé E. Guzmán Ramírez
Gestor	Guía las reuniones.  Garantiza los recursos para el cumplimiento del proyecto.	Jorge Luis
Entrenador	Persona con conocimientos de la metodología que guía en el proceso.	Grethell Castillo Reyes  Aimé E. Guzmán Ramírez
Cliente	Encargado de establecer la Lista de Reserva del Producto (LRP).  Darle prioridad a los elementos de	Asesor del Centro GEYSED

## Capítulo 2: Características del Sistema

	la lista.	
Programador	Implementar el código fuente de la aplicación.  Describe las pruebas unitarias.	Jorge Luis
Consultor	Persona con conocimiento específico para resolver algún problema.	Yoandrys S. Pacheco Jérez
Tester	Realizar pruebas al software para validar el cumplimiento de las funcionalidades.	Jorge Luis

### 1.3 Modelo de dominio

XP define como uno de sus procesos el modelado de las historias del negocio, ante el insuficiente conocimiento de los procesos por el equipo se presenta a continuación el modelo de dominio, para un entendimiento más amplio de los eventos que ocurren en el escenario donde se implementa la solución.

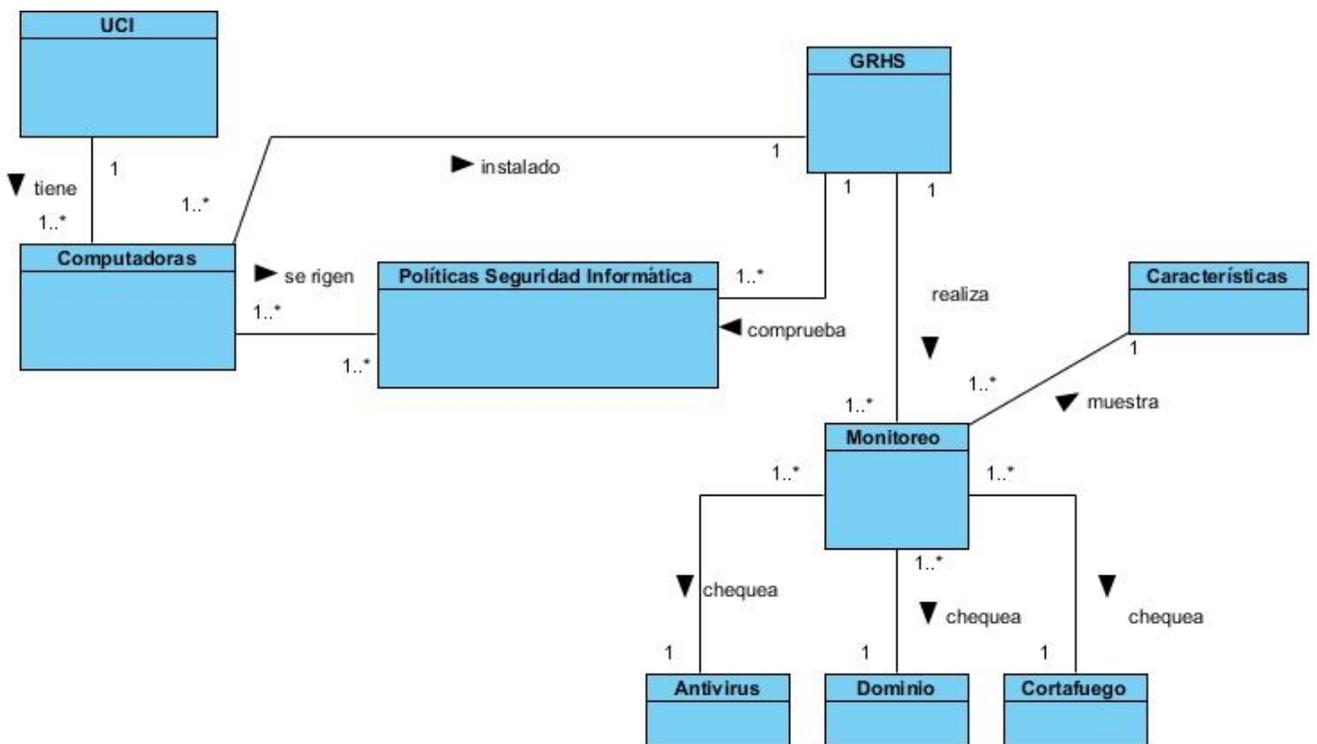


Fig #1. Modelo de Dominio (del autor)

## Capítulo 2: Características del Sistema

La Universidad de las Ciencias Informáticas cuenta con una cierta cantidad de computadoras que tienen instalado el Sistema de Gestión de Recursos Humanos (GRHS) para comprobar que todas cumplen con las Políticas de Seguridad Informáticas establecidas en la UCI. Dicho sistema realiza el monitoreo chequeando la existencia del antivirus, el cortafuego y que se encuentren en el dominio uci.cu. Las características son mostradas para su posterior análisis, favoreciendo así la toma de decisiones.

### ***Descripción de las clases del dominio***

- ❖ UCI: Universidad de las Ciencias Informáticas. Entidad única de su tipo en el país dedicada al desarrollo de aplicaciones informáticas.
- ❖ Computadores: se refiere al medio de trabajo de los especialistas de la UCI.
- ❖ Políticas de Seguridad Informática: conjunto de requerimientos que debe cumplir cada estación de trabajo en la UCI para garantizar la seguridad de la información.
- ❖ GRHS: herramienta empleada en el centro para la Gestión de Recursos de Hardware y Software.
- ❖ Monitoreo: proceso mediante el cual las computadoras del centro son escaneadas para velar por el cumplimiento de las Políticas de Seguridad Informática.
- ❖ Antivirus: programa que protege la información contenida en las computadoras de la infección y propagación de virus.
- ❖ Dominio: conectividad de todas las computadoras del Centro al dominio uci.cu.
- ❖ Cortafuego: programa que permite prohibir el acceso no autorizado.
- ❖ Características: resultado del escaneo realizado, donde se notifica el cumplimiento o no de las políticas.

### **1.4 Lista de Reserva del Producto**

Es el producto de trabajo que establece el orden y duración por la cual se registrará el progreso de creación. Esta lista tiende a crecer o sufrir modificaciones pues al comienzo del proyecto rara vez se tienen todos los elementos. La única condición es que solo puede ser alterada cuando ha concluido una iteración, nunca durante su transcurso.

Tabla # 2.

Lista de Reserva del Producto (del autor)

## Capítulo 2: Características del Sistema

Prioridad	Ítem *	Descripción	Estimación	Estimado por
<b>Alta</b>				
	RF1	Autenticar usuario	2 días	Jorge Luis
	RF2	Realizar escaneo de las máquinas de un área determinada	4 días	Jorge Luis
	RF3	Instalar paquetes en las estaciones clientes	5 días	Jorge Luis
	RF4	Realizar la instalación de paquetes de forma remota	2 días	Jorge Luis
	RF5	Establecer las políticas de seguridad que se desean chequear al realizar el escaneo.	4 días	Jorge Luis
	RF6	Crear laboratorio	2 días	Jorge Luis
	RF7	Eliminar laboratorio	1 día	Jorge Luis
	RF8	Modificar laboratorio	1 día	Jorge Luis
	RF9	Realizar reporte del estado de las vulnerabilidades	1 días	Jorge Luis
	RF10	Adicionar datos de auditoría en un área	5 días	Jorge Luis
	RF11	Modificar datos de auditoría en un área determinada	5 días	Jorge Luis
	RF12	Adicionar usuario	2 días	Jorge Luis
	RF13	Eliminar usuario	2 día	Jorge Luis
	RF14	Editar usuario	2 día	Jorge Luis
	RF15	Realizar reporte del software y hardware	1 día	Jorge Luis
<b>Media</b>				
	RF16	Notificar al asesor de seguridad fin del escaneo	2 días	Jorge Luis
	RF17	Enviar por correo electrónico el resultado del escaneo según la planificación	5 días	Jorge Luis
	RF18	Notificar advertencia a los usuarios que incidan en una falta	1 día	Jorge Luis
	RF19	Exportar reportes PDF.	2 días	Jorge Luis
	RF20	Filtrar por IP	1 día	Jorge Luis
	RF21	Filtrar por medio básico	5 días	Jorge Luis
	RF22	Filtrar por política	5 días	Jorge Luis
	RF23	Desconectar usuario	1 día	Jorge Luis
	RF24	Filtrar por fecha	2 días	Jorge Luis
	RF25	Filtrar por laboratorio	2 días	Jorge Luis
	RF26	Asignar permisos a funcionalidades por roles	2 días	Jorge Luis
	RF27	Eliminar permisos a funcionalidades por roles	2 días	Jorge Luis
	RF28	Planificar escaneo automático	2 días	Jorge Luis

## Capítulo 2: Características del Sistema

### 1.5 Requisitos no Funcionales

Tabla # 3. Requisitos No Funcionales (del autor)

Requisitos	Descripción
<b>Usabilidad</b>	Realizar reporte del estado del hardware y el software. El sistema está creado para ser utilizado por personas con conocimientos básicos de informática.
<b>Rendimiento</b>	El sistema debe menos de 5 segundos a la hora de procesar la información y dar respuesta a las peticiones de los usuarios
<b>Confidencialidad</b>	La información manejada por el sistema debe estar protegida ante el acceso no autorizado y la divulgación
<b>Disponibilidad</b>	El sistema debe estar disponible todo el tiempo para los usuarios autorizados El período entre fallos recuperables, como por ejemplo fallos en el servidor no debe exceder las 24 horas
<b>Restricciones de diseño</b>	El producto de software final debe diseñarse sobre una arquitectura modelo -vista-controlador Se utilizará como herramienta CASE Visual Paradigm para el modelado de la herramienta.
<b>Hardware (Estaciones cliente)</b>	Se requiere que tengan una tarjeta de red Al menos 128 MB de memoria RAM Procesador a 512 MHz como mínimo
<b>Hardware (Estaciones servidor)</b>	Se requiere una tarjeta de red El Servidor de Base de Datos tenga como mínimo 2GB de RAM y 40GB de disco duro

### 1.6 Historias de usuario y tareas de ingeniería

Para describir el trabajo a realizar de acorde a las especificidades del cliente la metodología seleccionada emplea el artefacto llamado historias de usuario y las descompone en tareas de ingeniería para su mejor comprensión. A continuación se observan las once historias de usuarios definidas y luego una pequeña muestra de tareas de ingeniería perteneciente a algunas de ellas. Para mayor información ver anexos del 1 al 16.

Tabla # 4. Historia de Usuario: Autenticar Usuario (del autor)

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Autenticar usuario.

## Capítulo 2: Características del Sistema

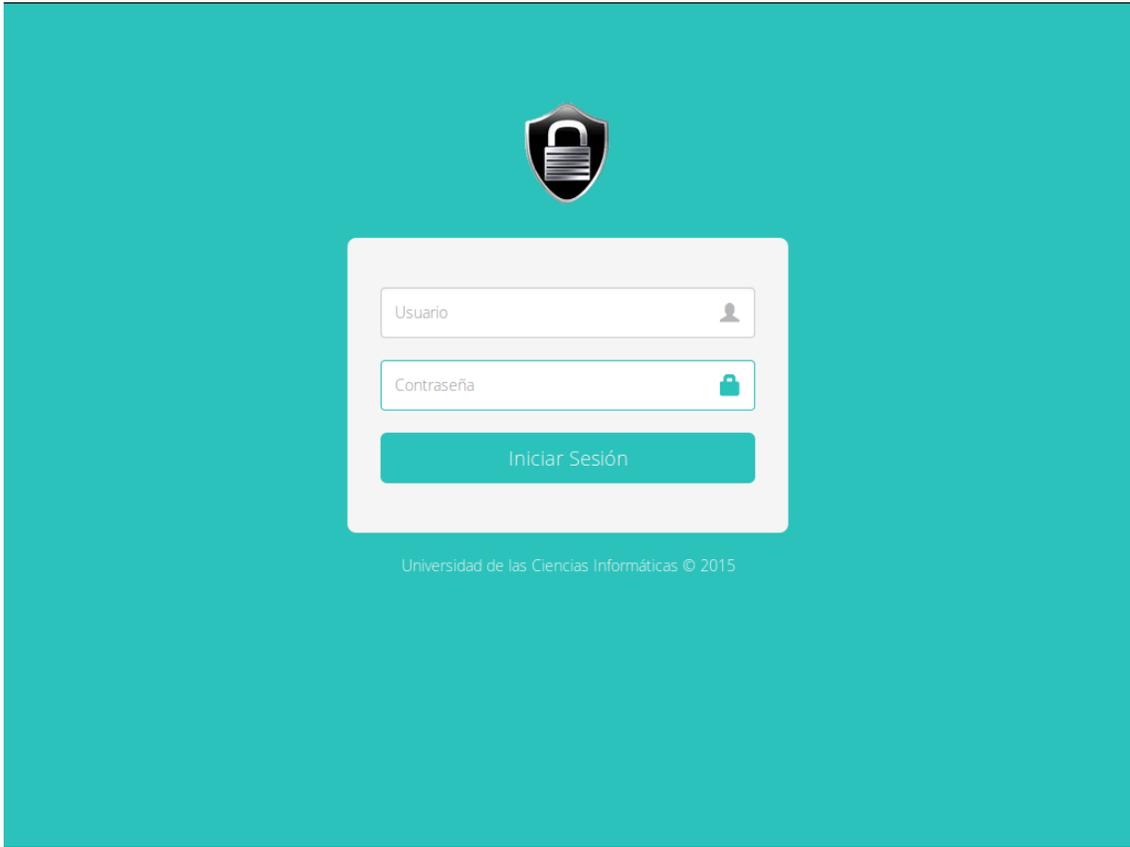
<b>Modificación de Historia de Usuario Número:</b> ninguna	
<b>Usuario:</b> Jorge Luis Hernández Suárez	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 3
<b>Descripción:</b> La presente historia de usuario tiene como objetivo ingresar un usuario al sistema para que solo tenga acceso a la información correspondiente.	
<b>Observaciones:</b> No aplica	
<b>Prototipo de interfaz:</b>	
 <p>Un prototipo de interfaz de usuario para el inicio de sesión. El fondo es de color verde azulado. En el centro hay un formulario blanco con un campo de texto etiquetado 'Usuario' con un ícono de persona a la derecha, un campo de texto etiquetado 'Contraseña' con un ícono de candado a la derecha, y un botón de color verde azulado con el texto 'Iniciar Sesión'. Encima del formulario hay un ícono de un candado dentro de un escudo. Debajo del formulario, en la parte inferior del formulario, se lee 'Universidad de las Ciencias Informáticas © 2015'.</p>	

Tabla # 5.

Historia de Usuario: Gestionar escaneo (del autor)

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Gestionar escaneo.
Modificación de Historia de Usuario Número: ninguna	

## Capítulo 2: Características del Sistema

<b>Usuario:</b> Jorge Luis Hernández Suárez	<b>Iteración Asignada:</b> 1
<b>Prioridad Negocio:</b> Alta	<b>Puntos Estimados:</b> 7
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 9
<b>Descripción:</b> realiza el escaneo a una computadora o varias computadoras de un área determinada, a través de la red.	
<b>Observaciones:</b> No aplica	

**Prototipo de interfaz:**

Escanear una computadora específica:

Listado de Computadoras + Registrar

Buscar: Laboratorio  IP  MAC  MB  Estado:  Activo:  Filtrar Limpiar

Computadoras 6

Laboratorio	IP	MB	SO-Activo	RAM	Microprocesador	Motherboard	Disco Duro	Actualizado	Estado	Click para analizar esta computadora.
Lab 306	10.54.15.248		Desconocido					22/06/2016 09:10:25		
Lab 306	10.54.15.249		Desconocido	0	imposible resolver el anfitrión geysed306-4 [sudo] password for prueba	imposible resolver el anfitrión geysed306-4 [sudo] password for prueba	160GB	22/06/2016 09:10:28		
Lab 306	10.54.15.250		Linux	4GB	Intel(R) Celeron(R) CPU G1830 @ 2.80GHz	PEGATRON CORPORATION	1000GB	22/06/2016 09:10:34	✓	
Lab 306	10.54.15.251		Desconocido	1GB	Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz	Intel Corporation	160GB	22/06/2016 09:10:31		

Escanear un área determinada:

Listado de Laboratorios + Registrar

Buscar: Nombre  Edificio  Subred  Estado:  Activo:  Filtrar Limpiar

Laboratorios 1

Nombre	Edificio	Subred	Actualizado	Estado	Click para analizar las pcs de este laboratorio.
Lab 306	Docente 4	10.54.15.*	22/06/2016 08:51:41	*	

Tabla # 6.

Historia de Usuario: Exportar reportes PDF (del autor)

## Capítulo 2: Características del Sistema

Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Exportar reportes PDF.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Jorge Luis Hernández Suárez	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 9
Riesgo en Desarrollo: Alto	Puntos Reales: 10
Descripción: devuelve un archivo PDF con los resultados obtenidos en el escaneo.	
Observaciones: No aplica	
Prototipo de interfaz:	

Tabla # 7. Historia de Usuario: Realizar filtrado (del autor)

Historia de Usuario	
Número: HU_5	Nombre Historia de Usuario: Realizar filtrado
Modificación de Historia de Usuario Número: ninguna	
Usuario: Jorge Luis Hernández Suárez	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3
Riesgo en Desarrollo: bajo	Puntos Reales: 1
Descripción: permite filtrar la información como sea más conveniente para el usuario atendiendo a un criterio especificado.	
Observaciones: No aplica	

## Capítulo 2: Características del Sistema

**Prototipo de interfaz:**

Listado de Laboratorios + Registrar

Laboratorios 1

Nombre	Edificio	Subred	Actualizado	Estado	Activo	Acciones
Lab 306	Docente 4	10.54.15.*	23/06/2016 14:56:14	*	✓	<input type="button" value="Ver"/> <input type="button" value="Actualizar"/> <input type="button" value="Configurar"/> <input type="button" value="Eliminar"/>

Tabla # 8. Historia de Usuario: Generar Reporte (del autor)

Historia de Usuario	
<b>Número:</b> HU_6	<b>Nombre Historia de Usuario:</b> Generar Reporte.
<b>Modificación de Historia de Usuario Número:</b> ninguna	
<b>Usuario:</b> Jorge Luis Hernández Suárez	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 5
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 4
<b>Descripción:</b> permite generar el reporte asociado a criterios relacionados con las vulnerabilidades detectadas	
<b>Observaciones:</b> No aplica	
<b>Prototipo de interfaz:</b>	

## Capítulo 2: Características del Sistema

Reporte generado por el sistema para el control de las políticas de seguridad

Reporte que detalla el estado de las PC.

Detalles de las PC. Total: 6. Bien: 3, Mal: 3

ID	Laboratorio	IP	MAC	MB	SO-Activo	RAM	Microprocesador	Motherboard	Disco Duro	Actualizado	Estado	Activo
8	Lab 306	10.54.15.252	00:1cc0:02:00:55		Linux	1GB	Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz	Intel Corporation	160GB	06/22/2016 09:10:25	Bien	Si
7	Lab 306	10.54.15.248			Desconocido					06/22/2016 09:10:25	Mal	No
5	Lab 306	10.54.15.249	00:1cc0:01:ad:20		Desconocido	0	imposible resolver el antihón geysed306-4 [sudo] password for prueba	imposible resolver el antihón geysed306-4 [sudo] password for prueba	160GB	06/22/2016 09:10:28	Mal	No
6	Lab 306	10.54.15.30	60:02:92:3d:7f:49		Linux	4GB	Intel(R) Celeron(R) CPU G1830 @ 2.80GHz	PEGATRON CORPORATION	1000GB	06/22/2016 09:10:28	Bien	Si
2	Lab 306	10.54.15.250	60:02:92:3d:7f:c0		Linux	4GB	Intel(R) Celeron(R) CPU G1830 @ 2.80GHz	PEGATRON CORPORATION	1000GB	06/22/2016 09:10:34	Bien	Si
3	Lab 306	10.54.15.251	00:1cc0:02:00:55		Desconocido	1GB	Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz	Intel Corporation	160GB	06/22/2016 09:10:31	Mal	No

Tabla # 9.

Historia de Usuario: Generar notificación (del autor)

Historia de Usuario	
Número: HU_7	Nombre Historia de Usuario: Generar notificación.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Jorge Luis Hernández Suárez	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 6
Riesgo en Desarrollo: Alto	Puntos Reales: 5
Descripción: una vez realizado el escaneo se debe notificar a los interesados sobre los resultados.	
Observaciones: para ello debe haberse realizado el escaneo.	
Prototipo de interfaz:	

## Capítulo 2: Características del Sistema

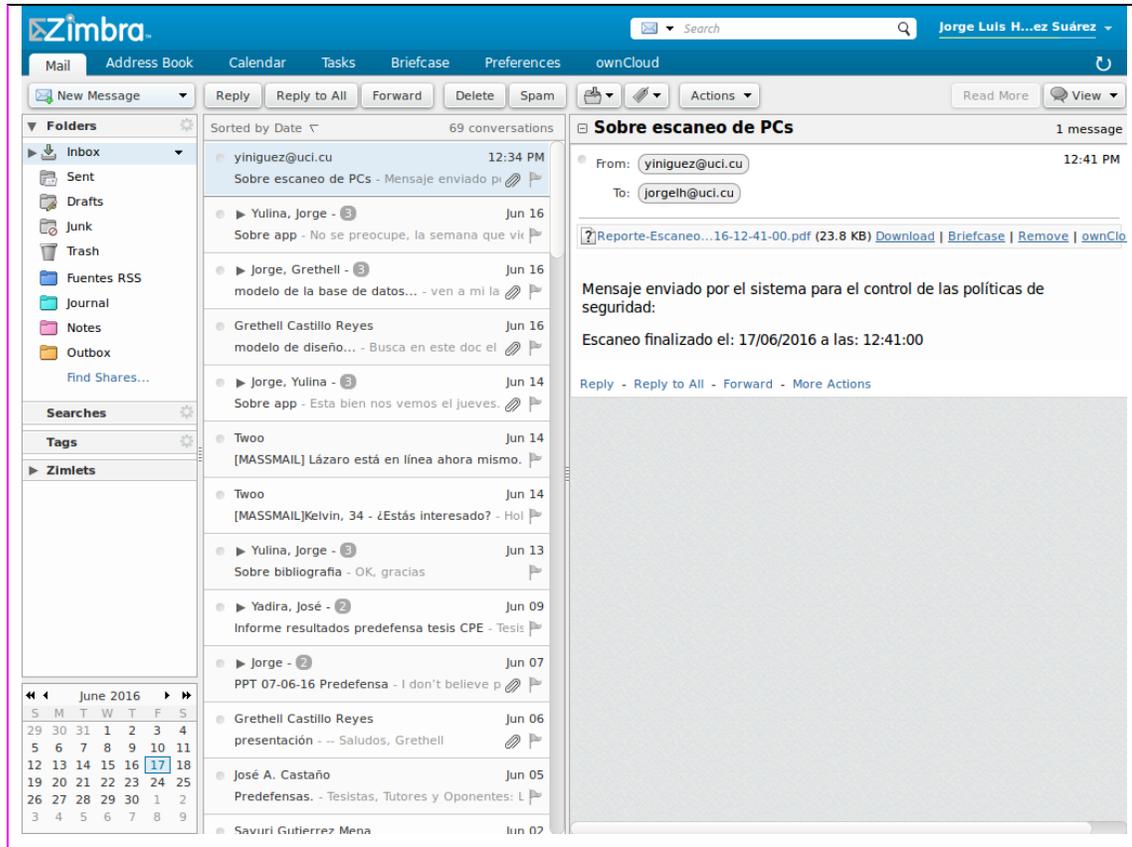


Tabla # 10. Historia de Usuario: Gestionar laboratorio (del autor)

Historia de Usuario	
Número: HU_8	Nombre Historia de Usuario: Gestionar laboratorio.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Jorge Luis Hernández Suárez	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3
Descripción: permite adicionar, eliminar, mostrar y modificar un laboratorio.	
Observaciones: No aplica	
Prototipo de interfaz: Adicionar laboratorio	

## Capítulo 2: Características del Sistema

**Sistema para el control de las políticas de seguridad.** Bienvenido, Administrador

Inicio Laboratorios Computadoras Reglas Reportes Administración Usuarios

Volver al listado de laboratorios / Registrar

**Nombre**  
Nombre:

**Estado**  
Bien

**Registrar** **Cancelar**

**Edificio**  
Edificio:

**Activo**  
Activo

**Subred**  
Subred:

**Modificar laboratorio**

Inicio Laboratorios Computadoras Reglas Reportes Administración Usuarios

Volver al listado de laboratorios / Editar Lab 306

**Nombre**  
Lab 306

**Estado**  
Mal

**Edificio**  
Docente 4

**Activo**  
Sí

**Subred**  
10.54.15.\*

**Guardar** **Cancelar**

Mostrar computadoras de un laboratorio

## Capítulo 2: Características del Sistema

Computadoras 7

Laboratorio	IP	MB	SO-Activo	RAM	Microprocesador	Motherboard	Disco Duro	Actualizado	Estado	Activo	Acciones
Lab 306	10.54.15.242		Linux	2GB	Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz	Intel Corporation	1000GB	23/06/2016 16:08:09	✓	✓	
Lab 306	10.54.15.248		Desconocido					22/06/2016 09:10:25			
Lab 306	10.54.15.249		Desconocido	0	imposible resolver el anfitrión geysed306-4 [sudo] password for prueba	imposible resolver el anfitrión geysed306-4 [sudo] password for prueba	160GB	22/06/2016 09:10:28			
Lab 306	10.54.15.250		Linux	4GB	Intel(R) Celeron(R) CPU G1830 @ 2.80GHz	PEGATRON CORPORATION	1000GB	22/06/2016 09:10:34	✓		
Lab 306	10.54.15.251		Desconocido	1GB	Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz	Intel Corporation	160GB	22/06/2016 09:10:31			
Lab 306	10.54.15.252		Linux	1GB	Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz	Intel Corporation	160GB	23/06/2016 16:07:41	✓	✓	
Lab 306	10.54.15.30		Linux	4GB	Intel(R) Celeron(R) CPU G1830 @ 2.80GHz	PEGATRON CORPORATION	1000GB	22/06/2016 09:10:28	✓		

Eliminar laboratorio

Sistema para el control de las políticas de seguridad - Mozilla Firefox

localhost:8100/app/user/labs/

Sistema para el control de las políticas de seguridad. Bienvenido, Administrador

Inicio Laboratorios Computadoras Reglas Reportes Administración Usuarios

Listado de Laboratorios + Registrar

Advertencia: Esta apunta de eliminar datos. Esta seguro que desea realizar esta operación?

Eliminar Cancelar

Nombre	Edificio	Subred	Actualizado	Estado	Activo	Acciones
Lab 306	Docente 4	10.54.15.*	23/06/2016 14:56:14	*	✓	
Lab 307	Docente 4	10.54.16.*	23/06/2016 16:09:51	*	✓	

Tabla # 11. Historia de Usuario: Gestionar datos de auditoría (del autor)

## Capítulo 2: Características del Sistema

Historia de Usuario																																																																																					
Número: HU_9	Nombre Historia de Usuario: Gestionar datos de auditoría.																																																																																				
Modificación de Historia de Usuario Número: ninguna																																																																																					
Usuario: Jorge Luis Hernández Suárez	Iteración Asignada: 2																																																																																				
Prioridad en Negocio: Alta	Puntos Estimados: 7																																																																																				
Riesgo en Desarrollo: Alto	Puntos Reales: 3																																																																																				
Descripción: permite adicionar, mostrar, eliminar, activar o desactivar y modificar las reglas de la auditoría.																																																																																					
Observaciones: No aplica																																																																																					
Prototipo de interfaz:																																																																																					
<p>The screenshot shows a web interface titled 'Listado de Reglas'. At the top, there is a navigation bar with 'Inicio', 'Laboratorios', 'Computadoras', 'Reglas', 'Reportes', 'Administración', and 'Usuarios'. Below the navigation bar, there is a 'Registrar' button highlighted in red and a link 'Crear nueva regla'. A search bar is present with fields for 'Nombre', 'Comando', 'Descripción', and 'Activ', along with 'Filtrar' and 'Limpiar' buttons. The main content is a table of rules with the following data:</p> <table border="1"> <thead> <tr> <th>Nombre</th> <th>Comando</th> <th>Descripcion</th> <th>Actualizado</th> <th>Como sudo</th> <th>Activo</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>Detectar instalación de windows.</td> <td>cat /root/grub/grub.cfg   grep -c "Window WINDOW window"</td> <td>Comando para detectar instalación de windows en la pc.</td> <td>16/06/2016 09:50:43</td> <td>Si</td> <td>✓</td> <td>⋮ ⓧ ⓧ</td> </tr> <tr> <td>Activar o desactivar regla</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Dominio</td> <td>/opt/llkewise/bin/lw-get-status   grep -i "dns domain"</td> <td>Comando para obtener el dominio del host.</td> <td>16/06/2016 09:50:43</td> <td>Si</td> <td>☑</td> <td>⋮ ⓧ ⓧ</td> </tr> <tr> <td>Estado básico del firewall.</td> <td>ufw status   grep -i "estado status"</td> <td>Comando para obtener el estado (habilitado o deshabilitado) del firewall.</td> <td>16/06/2016 09:50:43</td> <td>Si</td> <td>✓</td> <td>⋮ ⓧ ⓧ</td> </tr> <tr> <td>Estado general del firewall.</td> <td>ufw status</td> <td>Comando para obtener el estado del firewall de forma general.</td> <td>16/06/2016 09:50:43</td> <td>Si</td> <td>✓</td> <td>⋮ ⓧ ⓧ</td> </tr> <tr> <td>Grupo uci admin</td> <td>/opt/llkewise/bin/lwconfig --show RequireMembershipOf   grep -i "sec-uci-security"</td> <td>Comando para obtener el grupo de uci admin del host.</td> <td>16/06/2016 09:50:43</td> <td>Si</td> <td>✓</td> <td>⋮ ⓧ ⓧ</td> </tr> <tr> <td>Eliminar regla</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Hostname</td> <td>hostname</td> <td>Comando para obtener el nombre del host.</td> <td>16/06/2016 09:50:43</td> <td>No</td> <td>✓</td> <td>⋮ ⓧ ⓧ</td> </tr> <tr> <td>IP</td> <td>ifconfig   grep -i inet:   cut -d: -f2   awk '{print \$1}'</td> <td>Comando para obtener el IP del host.</td> <td>16/06/2016 09:50:43</td> <td>No</td> <td>*</td> <td>⋮ ⓧ ⓧ</td> </tr> <tr> <td>Paquetes no actualizados.</td> <td>apt-get -s upgrade   grep -m2 "upgraded"   actualizados"   cut -d: -f2   cut -d" " -f1</td> <td>Comando para obtener la cantidad de paquetes de software no actualizados.</td> <td>16/06/2016 09:50:43</td> <td>Si</td> <td>✓</td> <td>⋮ ⓧ ⓧ</td> </tr> <tr> <td>Software</td> <td>dpkg --get-selections</td> <td>Comando para obtener la lista de paquetes instalados en</td> <td>16/06/2016 09:50:43</td> <td>Si</td> <td>✓</td> <td>⋮ ⓧ ⓧ</td> </tr> </tbody> </table>		Nombre	Comando	Descripcion	Actualizado	Como sudo	Activo	Acciones	Detectar instalación de windows.	cat /root/grub/grub.cfg   grep -c "Window WINDOW window"	Comando para detectar instalación de windows en la pc.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ	Activar o desactivar regla							Dominio	/opt/llkewise/bin/lw-get-status   grep -i "dns domain"	Comando para obtener el dominio del host.	16/06/2016 09:50:43	Si	☑	⋮ ⓧ ⓧ	Estado básico del firewall.	ufw status   grep -i "estado status"	Comando para obtener el estado (habilitado o deshabilitado) del firewall.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ	Estado general del firewall.	ufw status	Comando para obtener el estado del firewall de forma general.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ	Grupo uci admin	/opt/llkewise/bin/lwconfig --show RequireMembershipOf   grep -i "sec-uci-security"	Comando para obtener el grupo de uci admin del host.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ	Eliminar regla							Hostname	hostname	Comando para obtener el nombre del host.	16/06/2016 09:50:43	No	✓	⋮ ⓧ ⓧ	IP	ifconfig   grep -i inet:   cut -d: -f2   awk '{print \$1}'	Comando para obtener el IP del host.	16/06/2016 09:50:43	No	*	⋮ ⓧ ⓧ	Paquetes no actualizados.	apt-get -s upgrade   grep -m2 "upgraded"   actualizados"   cut -d: -f2   cut -d" " -f1	Comando para obtener la cantidad de paquetes de software no actualizados.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ	Software	dpkg --get-selections	Comando para obtener la lista de paquetes instalados en	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ
Nombre	Comando	Descripcion	Actualizado	Como sudo	Activo	Acciones																																																																															
Detectar instalación de windows.	cat /root/grub/grub.cfg   grep -c "Window WINDOW window"	Comando para detectar instalación de windows en la pc.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ																																																																															
Activar o desactivar regla																																																																																					
Dominio	/opt/llkewise/bin/lw-get-status   grep -i "dns domain"	Comando para obtener el dominio del host.	16/06/2016 09:50:43	Si	☑	⋮ ⓧ ⓧ																																																																															
Estado básico del firewall.	ufw status   grep -i "estado status"	Comando para obtener el estado (habilitado o deshabilitado) del firewall.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ																																																																															
Estado general del firewall.	ufw status	Comando para obtener el estado del firewall de forma general.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ																																																																															
Grupo uci admin	/opt/llkewise/bin/lwconfig --show RequireMembershipOf   grep -i "sec-uci-security"	Comando para obtener el grupo de uci admin del host.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ																																																																															
Eliminar regla																																																																																					
Hostname	hostname	Comando para obtener el nombre del host.	16/06/2016 09:50:43	No	✓	⋮ ⓧ ⓧ																																																																															
IP	ifconfig   grep -i inet:   cut -d: -f2   awk '{print \$1}'	Comando para obtener el IP del host.	16/06/2016 09:50:43	No	*	⋮ ⓧ ⓧ																																																																															
Paquetes no actualizados.	apt-get -s upgrade   grep -m2 "upgraded"   actualizados"   cut -d: -f2   cut -d" " -f1	Comando para obtener la cantidad de paquetes de software no actualizados.	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ																																																																															
Software	dpkg --get-selections	Comando para obtener la lista de paquetes instalados en	16/06/2016 09:50:43	Si	✓	⋮ ⓧ ⓧ																																																																															

Tabla # 12.

Historia de Usuario: Desconectar usuario (del autor)

Historia de Usuario	
Número: HU_10	Nombre Historia de Usuario: Desconectar usuario.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Jorge Luis Hernández Suárez	Iteración Asignada: 3

## Capítulo 2: Características del Sistema

<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> permite al usuario, una vez concluido el uso del sistema, salir de él.	
<b>Observaciones:</b> No aplica	
<b>Prototipo de interfaz:</b>	

Tabla # 13. Historia de Usuario: Gestionar usuario (del autor)

Historia de Usuario	
<b>Número:</b> HU_11	<b>Nombre Historia de Usuario:</b> Gestionar usuario.
<b>Modificación de Historia de Usuario Número:</b> ninguna	
<b>Usuario:</b> Jorge Luis Hernández Suárez	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 6

## Capítulo 2: Características del Sistema

<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 6
<b>Descripción:</b> permite adicionar, eliminar y modificar un usuario y asignarle las responsabilidades según su rol.	
<b>Observaciones:</b> El administrador debe estar autenticado.	
<b>Prototipo de interfaz:</b>	
	

Tabla # 14. Tareas de Ingeniería asociada a la historia de usuario 1\_ Autenticar usuario (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.1	<b>Número Historia de Usuario:</b> HU_1
<b>Nombre Tarea:</b> Autenticar usuario.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 28-9-2015	<b>Fecha Fin:</b> 30-9-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> A través de esta funcionalidad el usuario accede al sistema y realiza las actividades definidas por el rol que desempeña.	

Tabla # 15. Tareas de Ingeniería asociada a la historia de usuario 2\_ Realizar escaneo de las máquinas de un área determinada (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.1	<b>Número Historia de Usuario:</b> HU_2

## Capítulo 2: Características del Sistema

<b>Nombre Tarea:</b> Realizar escaneo de las máquinas de un área determinada.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 1-10-2015	<b>Fecha Fin:</b> 5-10-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite detectar las vulnerabilidades en un área específica.	

Tabla # 16. Tareas de Ingeniería asociada a la historia de usuario 2\_ Instalar paquetes en las estaciones clientes (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.2	<b>Número Historia de Usuario:</b> HU_2
<b>Nombre Tarea:</b> Instalar paquetes en las estaciones clientes.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 15-10-2015	<b>Fecha Fin:</b> 21-10-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> esta funcionalidad le permite a un usuario sin privilegios una vez escanear su puesto de trabajo instalar, él mismo, los paquetes que den solución a los problemas detectados.	

Tabla # 17. Tareas de Ingeniería asociada a la historia de usuario 2\_ Realizar la instalación de paquetes de forma remota (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.3	<b>Número Historia de Usuario:</b> HU_2
<b>Nombre Tarea:</b> Realizar la instalación de paquetes de forma remota.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 22-10-2015	<b>Fecha Fin:</b> 24-10-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	

## Capítulo 2: Características del Sistema

**Descripción:** esta funcionalidad le permite a los asesores de tecnología una vez escaneado los puestos de trabajo instalar, vía remota, los paquetes que den solución a los problemas detectados.

Tabla # 18. Tareas de Ingeniería asociada a la historia de usuario 2\_ Establecer las políticas de seguridad que se desean chequear al realizar el escaneo (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 2.4</b>	<b>Número Historia de Usuario: HU_2</b>
<b>Nombre Tarea:</b> Establecer las políticas de seguridad que se desean chequear al realizar el escaneo.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 6-10-2015	<b>Fecha Fin:</b> 9-10-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> el sistema permite introducir las reglas por las cuales se realizará el escaneo.	

Tabla # 19. Tareas de Ingeniería asociada a la historia de usuario 2\_ Registrar causas de las vulnerabilidades (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 2.5</b>	<b>Número Historia de Usuario: HU_2</b>
<b>Nombre Tarea:</b> Registrar causas de las vulnerabilidades.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 12-10-2015	<b>Fecha Fin:</b> 12-10-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite guardar en base de datos los resultados de las auditorías informáticas realizadas.	

Tabla # 20. Tareas de Ingeniería asociada a la historia de usuario 2\_ Planificar escaneo automático (del autor)

## Capítulo 2: Características del Sistema

Tarea de Ingeniería	
<b>Número Tarea: 2.6</b>	<b>Número Historia de Usuario: HU_2</b>
<b>Nombre Tarea:</b> Planificar escaneo automático.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b>
<b>Fecha Inicio:</b> 13-10-2015	<b>Fecha Fin:</b> 14-10-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite planificar cuándo se desea escanear la red.	

Tabla # 21. Tareas de Ingeniería asociada a la historia de usuario 7\_ Notificar al asesor de seguridad el fin del escaneo (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 7.1</b>	<b>Número Historia de Usuario: HU_7</b>
<b>Nombre Tarea:</b> Notificar al asesor de seguridad fin del escaneo.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 12-1-2016	<b>Fecha Fin:</b> 13-1-2016
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> una vez concluido el escaneo se notifica por correo al asesor de seguridad que concluyó la tarea.	

Tabla # 22. Tareas de Ingeniería asociada a la historia de usuario 7\_ Enviar por correo resultado del escaneo según la planificación (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 7.2</b>	<b>Número Historia de Usuario: HU_7</b>
<b>Nombre Tarea:</b> Enviar por correo resultado del escaneo según la planificación.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 14-1-2016	<b>Fecha Fin:</b> 20-1-2016

## Capítulo 2: Características del Sistema

<b>Programador Responsable:</b> Jorge Luis Hernández Suárez
<b>Descripción:</b> el día planificado se realiza el escaneo y es notificado por correo.

Tabla # 23. Tareas de Ingeniería asociada a la historia de usuario 7\_ Notificar advertencia a los usuarios que incidan en una falta (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 7.3</b>	<b>Número Historia de Usuario: HU_7</b>
<b>Nombre Tarea:</b> Notificar advertencia a los usuarios que incidan en una falta.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 21-1-2016	<b>Fecha Fin:</b> 21-1-2016
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> si el sistema detecta una reincidencia notifica por correo a la persona que incurrió en el acto.	

Tabla # 24. Tareas de Ingeniería asociada a la historia de usuario 11\_ Notificar advertencia a los usuarios que incidan en una falta (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 11.1</b>	<b>Número Historia de Usuario: HU_11</b>
<b>Nombre Tarea:</b> Adicionar usuario.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 25-01-2015	<b>Fecha Fin:</b> 26-01-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite adicionar un usuario al sistema.	

Tabla # 25. Tareas de Ingeniería asociada a la historia de usuario 11\_ Eliminar usuario (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 11.2</b>	<b>Número Historia de Usuario: HU_11</b>

## Capítulo 2: Características del Sistema

<b>Nombre Tarea:</b> Eliminar usuario.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 27-01-2015	<b>Fecha Fin:</b> 28-01-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite eliminar un usuario del sistema.	

Tabla # 26. Tareas de Ingeniería asociada a la historia de usuario 11\_ Editar usuario (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 11.3	<b>Número Historia de Usuario:</b> HU_11
<b>Nombre Tarea:</b> Editar usuario.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 29-01-2015	<b>Fecha Fin:</b> 1-2-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite modificar un usuario del sistema.	

Tabla # 27. Tareas de Ingeniería asociada a la historia de usuario 11\_ Configurar usuario (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 11.4	<b>Número Historia de Usuario:</b> HU_11
<b>Nombre Tarea:</b> Configurar usuario.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 2-2-2015	<b>Fecha Fin:</b> 3-2-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite configurar las características del sistema para cada usuario.	

Tabla # 28. Tareas de Ingeniería asociada a la historia de usuario 11\_ Asignar permisos a funcionalidades por roles (del autor)

Tarea de Ingeniería
---------------------

## Capítulo 2: Características del Sistema

<b>Número Tarea: 11.5</b>	<b>Número Historia de Usuario: HU_11</b>
<b>Nombre Tarea:</b> Asignar permisos a funcionalidades por roles.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 4-2-2015	<b>Fecha Fin:</b> 5-2-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite configurar las funcionalidades a las que cada rol puede acceder.	

Tabla # 29. Tareas de Ingeniería asociada a la historia de usuario 11\_ Eliminar permisos a funcionalidades por roles (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 11.6</b>	<b>Número Historia de Usuario: HU_11</b>
<b>Nombre Tarea:</b> Eliminar permisos a funcionalidades por roles.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 5-2-2015	<b>Fecha Fin:</b> 8-2-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite eliminar un permiso otorgado a un rol para acceder a cierta funcionalidad.	

### 1.7 Plan de release

Tabla # 30. Plan de release (del autor)

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
1.	En esta iteración se realiza la implementación de los requisitos funcionales que se encargan de Gestionar laboratorio y Gestionar escaneo.	HU_8 y HU_2	4 semanas
2.	En esta iteración se realiza la implementación de los requisitos	HU_9 ,HU_6 y HU_5	4 semanas

## Capítulo 2: Características del Sistema

	funcionales que se encargan de Gestionar datos de auditoría, Generar Reporte y Realizar filtrado.		
3.	En esta iteración se realizará la implementación de los requisito funcionales que se encargan de Generar notificación, Autenticar usuario, Desconectar usuario y Gestionar rol.	HU_7, HU_1, HU_10, HU_11	4 semanas
4.	En esta iteración se realiza la implementación de el requisito funcional que se encarga de Exportar reportes PDF	HU_4 y HU_3	4 semanas

### 1.8 Arquitectura de software

La arquitectura de software: *"es una vista estructural de alto nivel, ocurre muy tempranamente en el ciclo de vida y define los estilos o grupos de estilos adecuados para cumplir con los requerimientos no funcionales"* (UCI, 2015).

- ❖ Basada en componentes

*"El desarrollo de software basado en componentes permite reutilizar piezas de código pre elaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión "* (Microsoft, 2016).

- ❖ Modelo Vista Controlador

*"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 un reducido espacio de tiempo"* (Fernández Romero, y otros, 2012).

Partes que lo componen (Fernández Romero, y otros, 2012):

**Modelo:** representa la capa de abstracción de los datos, o sea, se manejan los datos y las transformaciones que ocurran con ellos. No cuenta con la responsabilidad de mantener un enlace con las vistas, sino que es tarea del propio sistema notificar a las vistas cuando algo así suceda.

## Capítulo 2: Características del Sistema

**Vista:** representa la interfaz con la que interactuará el usuario, visualizando los datos del modelo.

**Controlador:** maneja las peticiones del usuario actuando sobre los datos y representándolo en la vista.

**Ventajas del MVC** (Torres Matos, y otros, 2011):

- ❖ *"Una clara separación entre los componentes de un programa; que permite implementarlos por separado".*
- ❖ *"Un API (Interfaz de Programación de Aplicaciones) muy bien definido; cualquiera que use el API, podrá reemplazar el modelo, la vista o el controlador, sin aparente dificultad".*
- ❖ *"La conexión entre el modelo y sus vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación".*

**Otras ventajas** (Fernández Romero, y otros, 2012):

- ❖ *"El uso de los frameworks basados en este patrón permite tener una separación lógica y física de los componentes de la aplicación, ya que por un lado se tienen los modelos, por otro lado las vistas y por otro lado los controladores"*
- ❖ *"Ofrece una elevada organización en el trabajo".*
- ❖ *"Estos frameworks poseen generadores que crean los archivos base de los modelos o vistas, para no tener que crear cada archivo relacionado a mano"*

### Estilos arquitectónicos

En un estudio realizado por (Reynoso, y otros, 2004) se definen como *"los más representativos y vigentes"* a los estilos a Flujo de Datos, Centrados en Datos, Estilos heterogéneos, *Peer-to-Peer* y el Llamada y Retorno, el cual será el empleado en el presente trabajo. Pues al decir de (Reynoso, y otros, 2004) *"Esta familia de estilos enfatiza la modificabilidad y la escalabilidad. Son los estilos más generalizados en sistemas en gran escala. Miembros de la familia son las arquitecturas de programa principal y subrutina, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objeto y los sistemas jerárquicos en capas"*.

### Patrones de Diseño

## Capítulo 2: Características del Sistema

*"Los patrones de diseño se han convertido en una técnica importante para el reuso del conocimiento de software. Cada patrón provee información sobre su diseño, describiendo las clases, métodos y relaciones que resuelven un problema de diseño en particular"* (Sandra Almeida, y otros, 2007).

Como se ha mencionado en este trabajo se emplea el *framework* de desarrollo Symfony. Marco de trabajo que trae como ventajas el uso de patrones de diseños que proporcionan una mayor calidad al código y supone una buena práctica para cualquier programador. El autor de la presente investigación emplea los patrones GRASP y GOF en la implementación del sistema.

### **GRASP**

- ❖ Experto: crea una capa de abstracción en el modelo, encapsula la lógica de los datos y genera las clases con funcionalidades comunes para las entidades. Por esta vía se crea una relación conteniendo la información de la tabla que representa. Esto es posible con la inclusión de Doctrine para el mapeo de la base de datos.
- ❖ Patrón Creador: permite crear objetos de una instancia como vía de acceso a la información del sistema. Se puede evidenciar su uso en la clase PcsController con el uso de objetos de clases que representan las entidades. Ejemplos de algunas funciones utilizadas en la clase PcsController son: createAction(), updateAction(), deleteAction()
- ❖ Bajo Acoplamiento: propicia la baja dependencia entre las clases del sistema, esto es apreciable en la no asociación del modelo con la vista o el controlador. Lo que implica un cambio imperceptible ante la necesidad de cambios. La clase PcsController hereda únicamente de Controller para alcanzar un bajo acoplamiento de clases.
- ❖ Alta Cohesión: trabajo por clase asignándole la responsabilidad necesaria a cada una de ellas, por lo que solo es usada para la tarea que fue creada. La clase PcsController, está formada por varias funcionalidades que están estrechamente relacionadas, siendo la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, crear instancias de objetos y acceder a las propiedades.
- ❖ Controlador: todas las peticiones son controladas por un controlador frontal, que es el único punto de entrada a la aplicación. Este patrón se evidencia en la clase app.php que es el punto de entrada de la aplicación.

### **GOF**

## Capítulo 2: Características del Sistema

- ❖ Acción (command): permite la creación de objetos con acciones encapsuladas para no duplicar implementación en presencia de dos eventos que dan respuesta a la misma funcionalidad.
- ❖ Decorador (decorator): patrón mediante el cual se pueden añadir funcionalidades dinámicamente, o sea, se modifican, agregan o retiran las propiedades durante su ejecución. Este patrón se evidencia en la clase `base.html.twig`, padre de todas las vistas, que contiene el Layout de la página. Este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página.
- ❖ Instancia única (singleton): garantiza que una clase sólo tenga una única instancia, previendo un punto de acceso global.

### 1.9 Modelado del diseño

El modelado del diseño pretende realizar un bosquejo inicial del sistema a desarrollar. Entre las prácticas de XP se encuentra la metáfora, "una historia compartida que describe cómo debería funcionar el sistema" (Letelier, y otros, 2003). Es la solución funcional más simple que tiene como salida el modelo de clases del diseño.

#### Diagrama de Clases del Diseño

## Capítulo 2: Características del Sistema

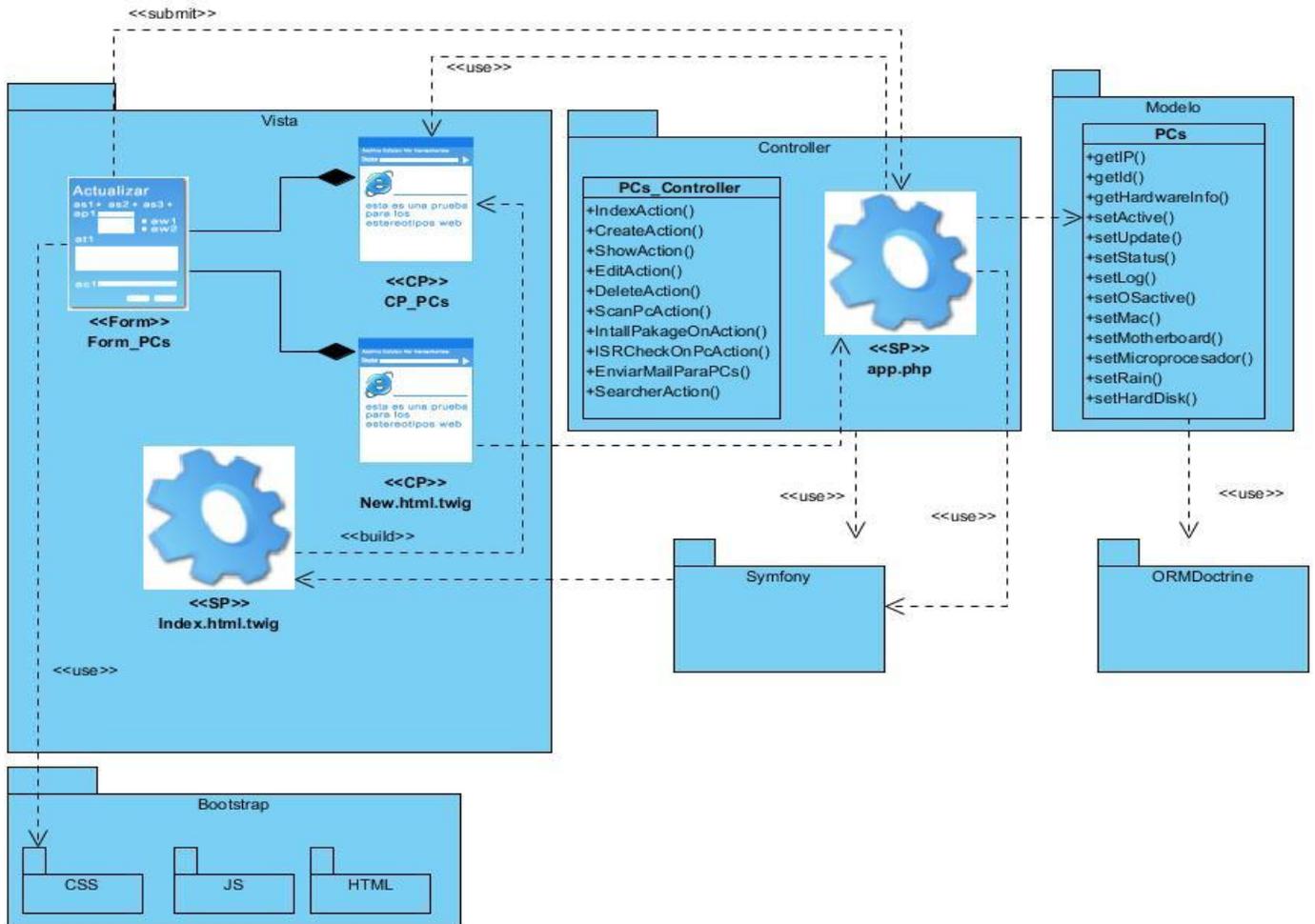


Fig #2. Diagrama de Clases del Diseño Adicionar PCs (del autor)

### Descripción del Diagrama de Diseño

Como se observa en el diagrama de clases de la Fig #2 el usuario accede a la aplicación para adicionar una o varias computadoras al listado de las ya existentes. Esta operación es realizada a través de la clase `PCs`, la cual muestra un formulario que es construido mediante el empleo de Bootstrap. La información introducida por el usuario es enviada al controlador que se encarga haciendo uso del método `CreateAction()` de enviarla al modelo que utiliza el ORM Doctrine.

### Modelo de Datos

Una de las características del sistema a construir es la necesidad de guardar la información para futuras consultas, actividad que solo puede ser realiza mediante la elaboración de una BD. El modelo de datos o diagrama entidad-relación ofrece la información de las tablas más significativas del negocio. En este caso se cuenta con las tablas usuario, labs, pcs, credentials, isr e isrdtail.

## Capítulo 2: Características del Sistema

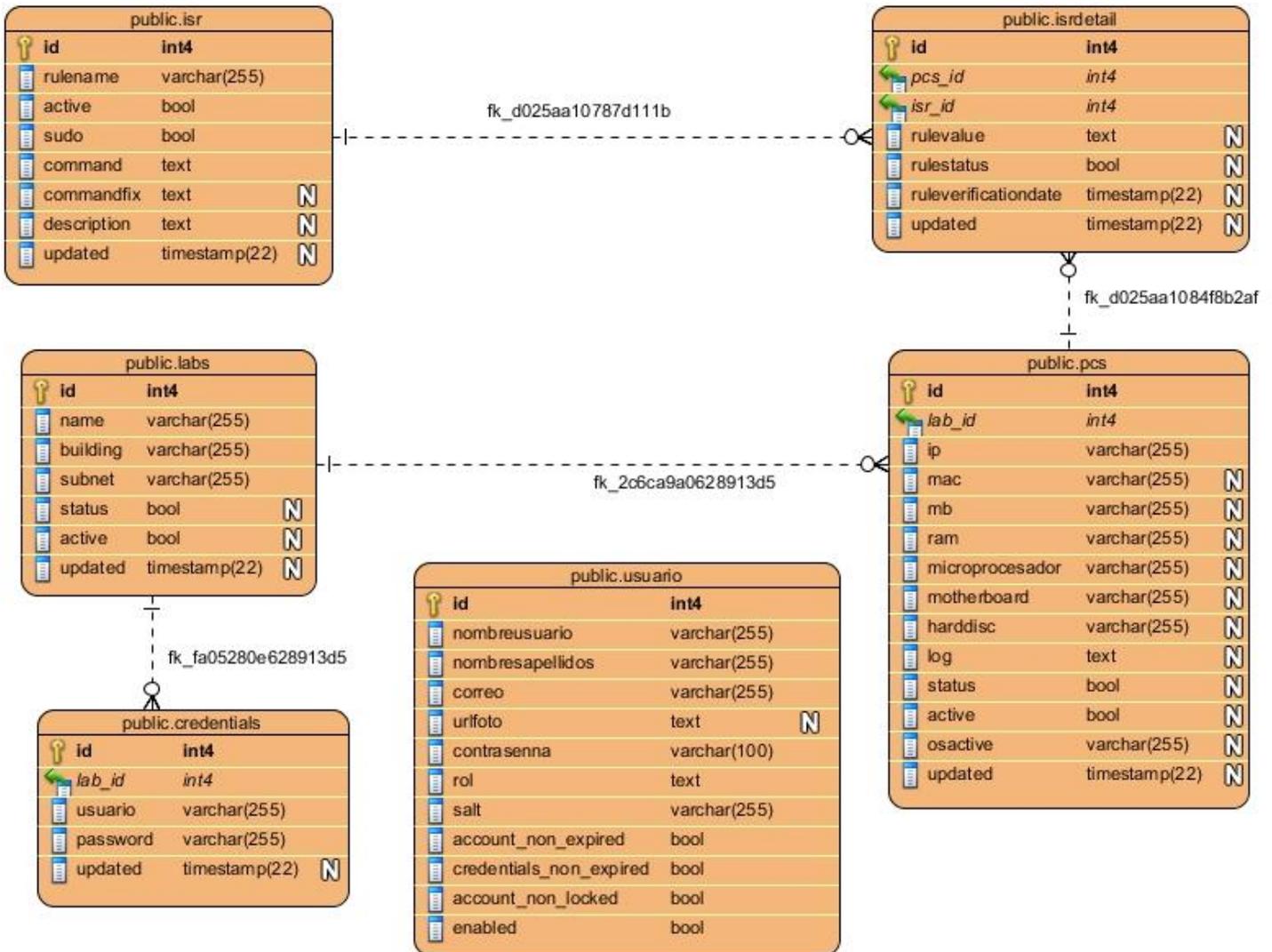


Fig #3. Modelo de datos (del autor)

### Diagrama de componentes

Para una mejor comprensión física del producto se presenta el diagrama de componentes y las relaciones que entre ellos existe. Su objetivo es modelar la vista estática y dinámica del sistema y para ello se relacionan las librerías, paquetes y archivos de forma tal que se puedan controlar las dependencias entre dichos componentes. En fin es un esquema que muestra el comportamiento de la solución.

## Capítulo 2: Características del Sistema

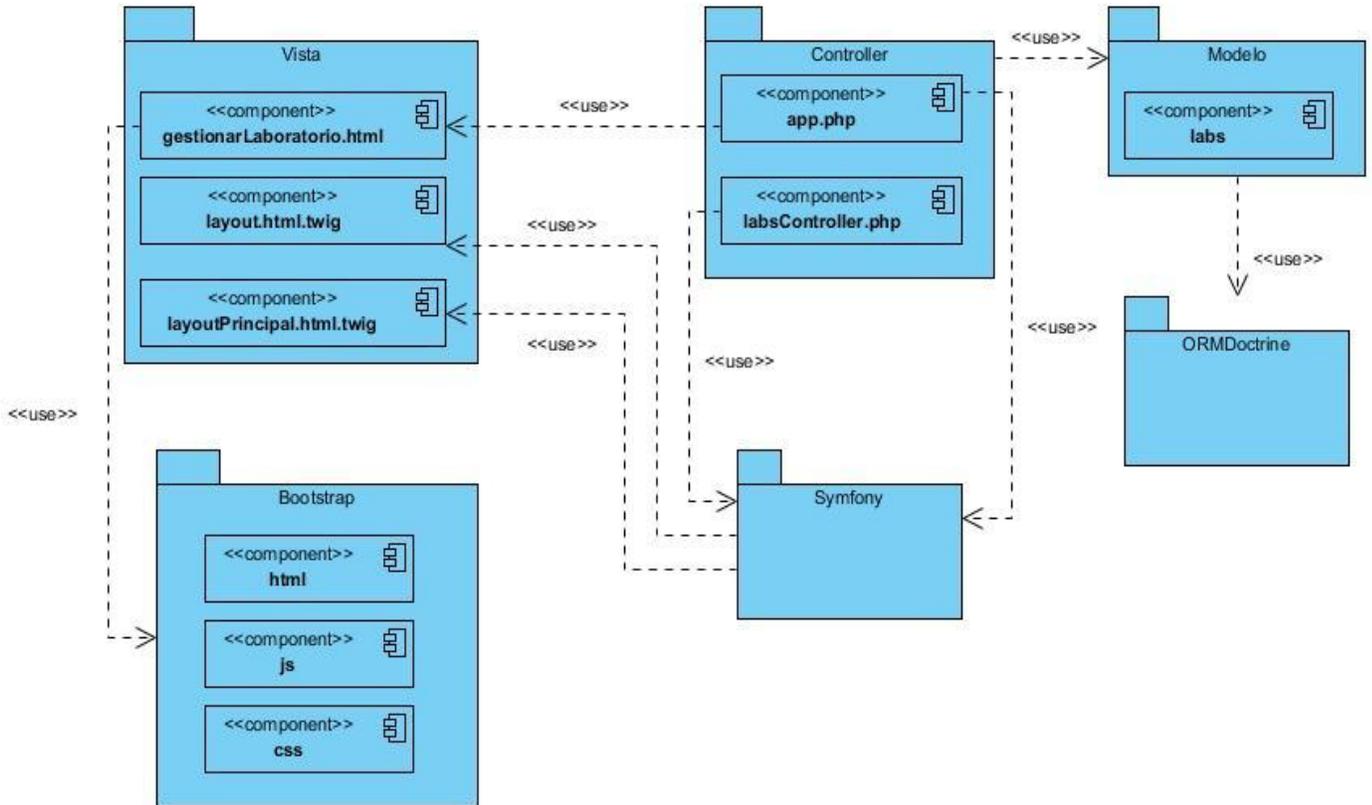


Fig #4. Diagrama de Componentes (del autor)

### Diagrama de Despliegue

El diagrama de despliegue muestra las relaciones de la distribución física de los nodos que componen el sistema. O sea, se puede decir que se muestra la arquitectura desde el punto de vista del despliegue. Los nodos son recursos de ejecución como los dispositivos de una computadora.

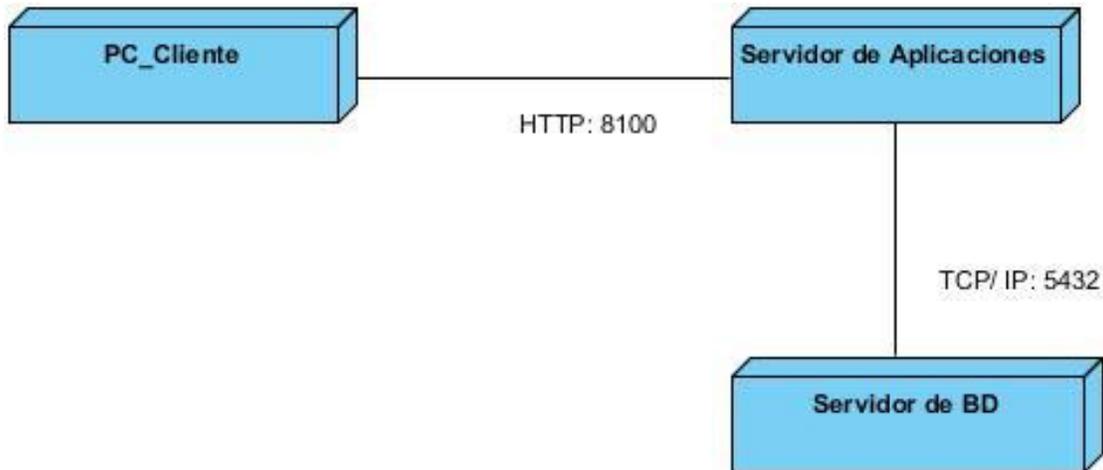


Fig #5. Diagrama de Despliegue (del autor)

## Capítulo 2: Características del Sistema

### 1.10 Conclusiones parciales

Todo proceso de desarrollo cuenta con una etapa de análisis y diseño donde, en dependencia de la metodología seleccionada, se generan los artefactos correspondientes. Cuando se decide optar por una metodología ágil, no se prescinde de la documentación, aunque es notable la disminución en los artefactos respecto a metodologías robustas. En el caso de la metodología XP los requisitos funcionales y no funcionales se describen a través de historias de usuario y se descomponen en tareas de ingeniería. Mediante la representación física de los componentes se realiza el modelado de diseño. Por esta vía se llega a la implementación del sistema.

### Capítulo 3

---

#### 2.1 Introducción

Este capítulo recoge el resultado de las pruebas aplicadas al sistema pues no se puede afirmar que un producto tiene la calidad requerida si no se le realizan pruebas que den muestra de ello. En este caso se emplean los casos de prueba de aceptación, donde se va probando la correcta respuesta del sistema ante las descripciones realizadas. En un último momento es mostrado el resultado de la investigación.

#### 2.2 Casos de pruebas de aceptación

Si se conoce la finalidad del producto y sus características se pueden realizar pruebas que demuestran su operatividad. Una de las características de las metodologías ágiles radica en realizar pruebas mientras se va avanzando en la implementación, cada vez que se culmina una iteración. En este caso se hará uso del tipo denominado de Caja Negra, que está enfocado en los requisitos funcionales. Enfocando el proceso en la técnica de casos de prueba de aceptación. Según (Gutiérrez, y otros, 2006) las pruebas de aceptación o funcionalidad son *"destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñada por el cliente final"*. De acuerdo a las historias de usuario y en función de validar las tareas de ingeniería se plantearon los casos de prueba de aceptación que a continuación se muestran.

Es importante destacar que en las metodologías que emplean caso de uso como parte de la captura de requisitos, se describen los campos necesarios para facilitar las descripciones de las pruebas, como son los escenarios, los flujos alternos, las precondiciones y las postcondiciones. Sin embargo las historias de usuario no brindan una información tan detallada. Es por tal motivo que es fundamental la colaboración del cliente en este momento tan crucial del desarrollo.

Tabla # 31. Caso de prueba historia de usuario 1-1\_ Autenticar usuario (del autor)

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU_1-1	<b>Nombre Historia de Usuario:</b> Autenticar usuario
<b>Nombre de la persona que realiza la prueba:</b> Yoandrys S. Pacheco Jérez	
<b>Descripción de la Prueba:</b> consiste en evaluar el requisito Autenticar usuario para comprobar la autenticidad de la información.	
<b>Condiciones de Ejecución:</b> acceso mediante el navegador a la url del sistema.	
<b>Entrada / Pasos de ejecución:</b> para evaluar el requisito es necesario:	

## Capítulo 3: Validación y pruebas

<ul style="list-style-type: none"> <li>- Introducir el nombre del usuario.</li> <li>- Introducir la contraseña del usuario.</li> <li>- Seleccionar el botón Iniciar Sesión.</li> </ul>
<b>Resultado Esperado:</b> el usuario accede al sistema.
<b>Evaluación de la Prueba:</b> Satisfactorio

Tabla # 32. Caso de prueba historia de usuario 2-1\_ Realizar escaneo de las máquinas de un área determinada (del autor)

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU_2-1	<b>Nombre Historia de Usuario:</b> Realizar escaneo de las máquinas de un área determinada.
<b>Nombre de la persona que realiza la prueba:</b> Yoandrys S. Pacheco Jérez	
<b>Descripción de la Prueba:</b> consiste realizar un escaneo en las pcs de un área determinada.	
<b>Condiciones de Ejecución:</b> debe estar autenticado.	
<b>Entrada / Pasos de ejecución:</b> para evaluar el requisito es necesario: <ul style="list-style-type: none"> <li>- Seleccionar en el menú la opción Laboratorio.</li> <li>- En la lista desplegable que aparece seleccionar Listado.</li> <li>- Se selecciona de la lista de locales del centro el que se desea escanear.</li> </ul>	
<b>Resultado Esperado:</b> la herramienta debe mostrar un listado con el escaneo realizado.	
<b>Evaluación de la Prueba:</b> Satisfactorio	

Tabla # 33. Caso de prueba historia de usuario 2-4\_ Establecer las políticas de seguridad que se desean chequear al realizar el escaneo (del autor)

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU_2-4	<b>Nombre Historia de Usuario:</b> Establecer las políticas de seguridad que se desean chequear al realizar el escaneo.
<b>Nombre de la persona que realiza la prueba:</b> Yoandrys S. Pacheco Jérez	
<b>Descripción de la Prueba:</b> consiste en evaluar el requisito Establecer las políticas de seguridad que se desean chequear al realizar el escaneo.	
<b>Condiciones de Ejecución:</b> debe estar autenticado.	

## Capítulo 3: Validación y pruebas

Ser administrador del sistema.
<b>Entrada / Pasos de ejecución:</b> para evaluar el requisito es necesario: <ul style="list-style-type: none"> <li>- Seleccionar en el menú la opción Reglas.</li> <li>- En la lista desplegable que aparece, seleccionar Listado.</li> <li>- Se selecciona el botón registrar.</li> </ul>
<b>Resultado Esperado:</b> nueva regla añadida
<b>Evaluación de la Prueba:</b> Satisfactorio

Tabla # 34. Caso de prueba historia de usuario 2-4\_ Exportar reportes PDF (del autor)

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU_3-1	<b>Nombre Historia de Usuario:</b> Exportar reportes PDF
<b>Nombre de la persona que realiza la prueba:</b> Yoandrys S. Pacheco Jérez	
<b>Descripción de la Prueba:</b> devuelve un archivo PDF con los resultados obtenidos en el escaneo.	
<b>Condiciones de Ejecución:</b> debe estar autenticado.	
<b>Entrada / Pasos de ejecución:</b> para evaluar el requisito es necesario: <ul style="list-style-type: none"> <li>- Seleccionar en el menú la opción Reportes.</li> <li>- En la lista desplegable que aparece seleccionar la opción de reporte deseado.</li> </ul>	
<b>Resultado Esperado:</b> la herramienta debe mostrar un archivo PDF con una determinada información.	
<b>Evaluación de la Prueba:</b> Satisfactorio	

Tabla # 35. Caso de prueba historia de usuario 2-4\_ Realizar filtrado (del autor)

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU_5-1	<b>Nombre Historia de Usuario:</b> Realizar filtrado
<b>Nombre de la persona que realiza la prueba:</b> Yoandrys S. Pacheco Jérez	
<b>Descripción de la Prueba:</b> se realiza un filtro de información especificado por el usuario.	
<b>Condiciones de Ejecución:</b> debe estar autenticado.	

## Capítulo 3: Validación y pruebas

<p><b>Entrada / Pasos de ejecución:</b> para evaluar el requisito es necesario:</p> <ul style="list-style-type: none"> <li>- Especificar tipo de filtro.</li> </ul>
<p><b>Resultado Esperado:</b> la herramienta debe mostrar la información solicitada por el usuario.</p>
<p><b>Evaluación de la Prueba:</b> Satisfactorio</p>

Tabla # 36. Caso de prueba historia de usuario 2-4\_ Gestionar laboratorio (del autor)

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU_3-1	<b>Nombre Historia de Usuario:</b> Gestionar laboratorio
<b>Nombre de la persona que realiza la prueba:</b> Yoandrys S. Pacheco Jérez	
<b>Descripción de la Prueba:</b> permite la agregación, eliminación o modificación de un laboratorio.	
<b>Condiciones de Ejecución:</b> debe estar autenticado y tener privilegios de administración.	
<p><b>Entrada / Pasos de ejecución:</b> para evaluar el requisito es necesario:</p> <ul style="list-style-type: none"> <li>- Seleccionar en el menú la opción Laboratorios.</li> <li>- En la lista desplegable que aparece seleccionar la opción Listado.</li> <li>- Seleccionar Registrar para adicionar un nuevo laboratorio o seleccionar Eliminar o Editar para la modificación o eliminación del mismo.</li> </ul>	
<b>Resultado Esperado:</b> se pudo realizar la gestión de un laboratorio.	
<b>Evaluación de la Prueba:</b> Satisfactorio	

Tabla # 37. Caso de prueba historia de usuario 2-4\_ Desconectar usuario (del autor)

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU_10-1	<b>Nombre Historia de Usuario:</b> Desconectar usuario
<b>Nombre de la persona que realiza la prueba:</b> Yoandrys S. Pacheco Jérez	
<b>Descripción de la Prueba:</b> permite cerrar la sesión.	
<b>Condiciones de Ejecución:</b> debe estar autenticado.	

## Capítulo 3: Validación y pruebas

**Entrada / Pasos de ejecución:** para evaluar el requisito es necesario:

- Seleccionar el menú Bienvenido usuario.
- En la lista desplegable que aparece seleccionar la opción Cerrar Sesión.

**Resultado Esperado:** la herramienta cierra la sesión autenticada y vuelve a la página principal.

**Evaluación de la Prueba:** Satisfactorio

### 2.3 Sistema desarrollado

A continuación, se presentan algunas imágenes del sistema:

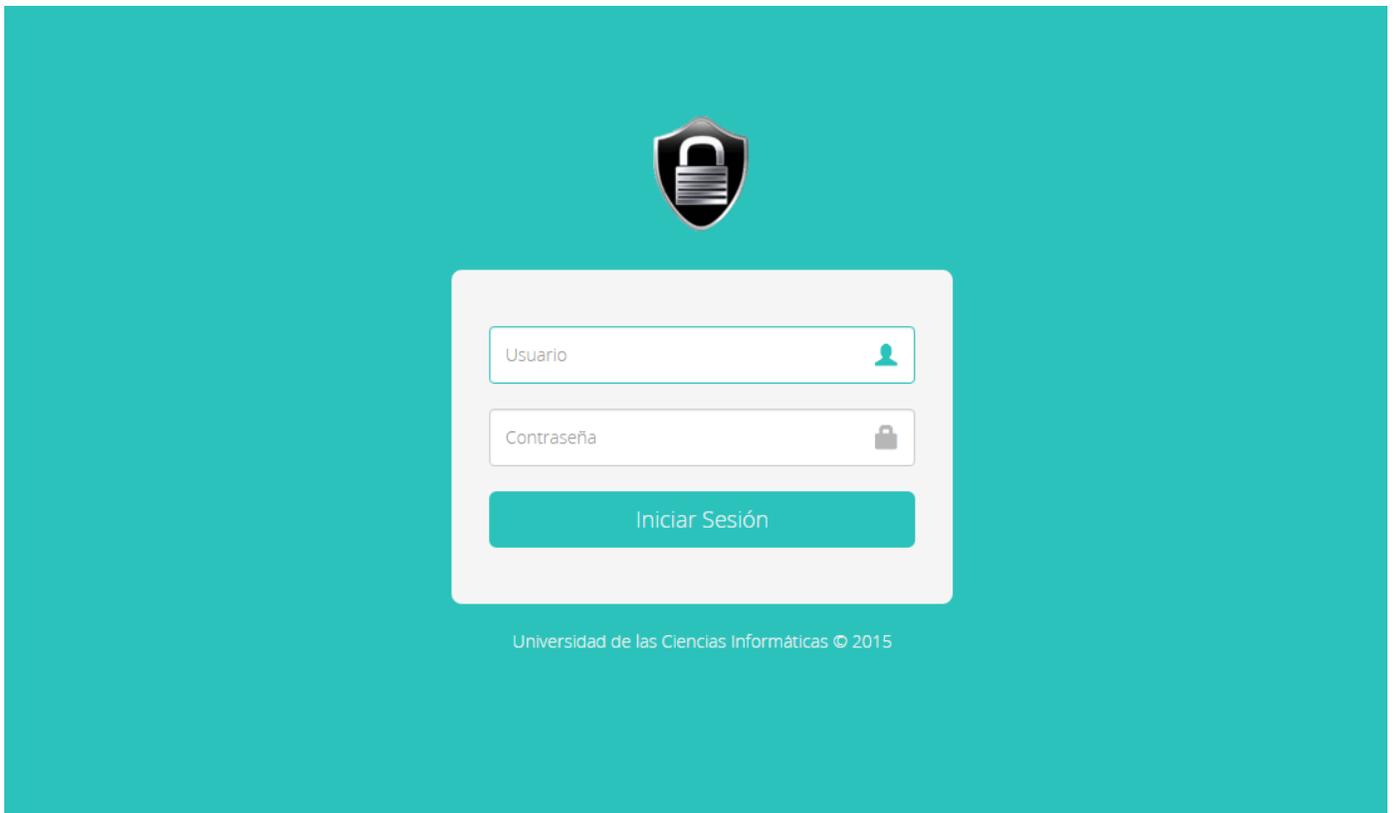


Fig #6. Imagen del sistema que muestra la pantalla de acceso (del autor)

## Capítulo 3: Validación y pruebas

Listado de Laboratorios + Registrar

Buscar:      🔍 Filtrar 🧼 Limpiar

Laboratorios 4

Acciones: 📄 Test1 📄 Test2 📄 Test2 📄 Test2

Nombre	Edificio	Subred	Actualizado	Estado	Activo	Acciones
Lab 303	docente 4	10.54.12.*	17/05/2016 09:09:22	*	✓	👁️ 📄 🗑️
lab 305	docente 4	10.54.14.*	16/05/2016 22:47:38	*	✓	👁️ 📄 🗑️
Lab307	Docente4	10.54.16.*	16/05/2016 22:39:42	*	✓	👁️ 📄 🗑️
Lab 401	Docente4	10.54.17.*	16/05/2016 22:41:23	*	✓	👁️ 📄 🗑️

Fig #7. Imagen del sistema que muestra un listado de los laboratorios del centro (del autor)



Fig #8. Imagen del sistema que muestra la pantalla de bienvenida (del autor)

La aplicación presenta un diseño sencillo, mostrando un interfaz común para todo el sistema, de forma tal que sea cómodo para el usuario final. El empleo de menú desplegables permite no sobrecargar las páginas. Se garantiza el acceso no deseado a través de la pantalla de autenticación.

### 2.4 Conclusiones parciales

No es posible desarrollar un sistema de calidad sin realizar pruebas que respalden esta afirmación. Las pruebas en el desarrollo ágil van aparejadas a la implementación, o sea, son parte del proceso de construcción. Los clientes tienen la oportunidad de probar la herramienta a través de las pruebas de

## Capítulo 3: Validación y pruebas

aceptación como parte de las llamadas pruebas de caja negra. La aplicación muestra un diseño sencillo y lo intuitivo de su manejo.

# Conclusiones

## Conclusiones

---

Una vez concluida la investigación se llegaron a las siguientes conclusiones:

- ❖ En la elaboración del capítulo 1 se arribó a la conclusión de que en una organización efectuar auditorías a la seguridad informática es primordial para garantizar la integridad, confidencialidad y disponibilidad del activo más importante.
- ❖ El estado del arte actual no brinda una solución a la problemática planteada al principio de la investigación. Sin embargo, el estudio realizado sirvió para tener un mejor entendimiento de este tipo de aplicaciones y una guía de cómo realizar una propuesta de solución.
- ❖ Se emplea XP como metodología para guiar el proceso de desarrollo. Los artefactos generados sirven para documentar de forma ágil el proceso de análisis y diseño.
- ❖ No es posible desarrollar un sistema de calidad sin realizar pruebas que demuestren que realmente cumple con las restricciones de los clientes.
- ❖ Se logró la implementación de un sistema que permite gestionar la información asociada a las auditorías de seguridad informática en la UCI y chequear su cumplimiento.

## Recomendaciones

### Recomendaciones

---

- ❖ Mostrar el resultado a través de gráficas con el objetivo de mejorar la toma de decisiones.
- ❖ Agregar nuevas funcionalidades que permitan robustecer el sistema.

### Referencias

---

- Aja Quiroga, Lourdes. 2002.** *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. 2002.
- Alvarado B, Javier. 2006.** *SISTEMAS DE PLANIFICACION DE RECURSOS EMPRESARIALES*. Cuenca : s.n., 2006.
- Calderón, Amaro, y otros. 2007.** *Metodologías Ágiles*. Trujillo : s.n., 2007.
- Departamento de Ingeniería de Software. 2007-2008.** *Introducción al proceso de desarrollo de software*. Habana : s.n., 2007-2008.
- Díaz Díaz, Adrian. 2013.** *XStormAlmiquiServer, servidor para el cliente de visualización web del SAINUX Almiquí*. Habana : s.n., 2013.
- Fernández G, José. 2013.** *Seguridad en Informática*. 2013.
- Gibert Ginestà, Marc y Pérez Mora, Oscar. 2016.** Universitat Oberta de Catalunya (UOC). *Universitat Oberta de Catalunya (UOC)*. [En línea] 2016. [http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\\_M2109\\_02152.pdf](http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf).
- Gómez, Daniel P. Valdés. 2015.** SlidePlayer. [En línea] 2015. <http://slideplayer.es/slide/4311994/>.
- Gutiérrez, J J, y otros. 2006.** *Pruebas del Sistema en Programación Extrema*. Sevilla : s.n., 2006.
- Letelier, Patricio y Penadés, María Carmen. 2003.** *Metodologías ágiles para el desarrollo de software: eXtreme Programming*. 2003.
- Microsoft . 2016.** Developer Network. [En línea] 2016. <https://msdn.microsoft.com/es-es/library/bb972268.aspx>.
- Patrón Modelo-Vista-Controlador. Fernández Romero, Yenisleidy y Díaz González, Yanette. 2012.** 2012, Telem@tica, pág. 57.
- Pérez, María José. 2015.** *Guía Comparativa de Metodologías Ágiles*. 2015.
- Pfleeger, Charles, Lawrence, Shari y Margulies, Jonathan. 2015.** *Security in Computing*. 2015.
- Ponjuán, Dante. 1998.** *Gestión de Información en la Organización (Principios, conceptos y aplicaciones)*. Santiago de Chile : s.n., 1998.
- Potencier, Fabien y Zaninotto, François. 2008.** *Symfony la guía definitiva*. s.l. : Eyrolles Collection, 2008.
- Ramos Betacourt, Surenis y León Rdoríguez, Ivette. 2012.** *Módulo de Alarmas y Acciones ante Incidencias*. Habana : s.n., 2012.
- Reynoso, Carlos y Kiccillof, Nicolás. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Buenos Aires : s.n., 2004.
- Rodríguez González, Pilar. 2008.** *Estudio de la aplicación de metodologías ágiles para la evolución de productos de software*. 2008.
- Sandra Almeida, Adriana y Perez Cavenago, Vanina. 2007.** *Arquitectura de Software: Estilos y Patrones*. Argentina : s.n., 2007.
- Seguridad Informática y Cartografía. 2006.** *Seguridad Informática y Cartografía*. 2006.
- The Eniac Corporation. 2015.** ACL. *La clave para Auditoría, Cumplimiento, Monitoreo Continuo y Detección de Fraudes*. Venezuela : s.n., 2015.
- The PHP Group. 2016.** PHP. [En línea] 2016. <http://php.net/>.
- Torres Matos, Yulissa y Morales González, Yanelis. 2011.** *Plataforma de Gestión de Hardware y Software. Módulo: Mapa tecnológico*. Habana : s.n., 2011.
- UCI. 2015.** Entorno Visual de Aprendizaje. [En línea] 25 de 07 de 2015. [eva.uci.cu](http://eva.uci.cu).

### Bibliografía

---

- Aja Quiroga, Lourdes. 2002.** *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. 2002.
- Alvarado B, Javier. 2006.** *SISTEMAS DE PLANIFICACION DE RECURSOS EMPRESARIALES*. Cuenca : s.n., 2006.
- Architecture paradigms and their influences and impacts on componente-based software systems.* **Wang, Guijun y Fung, Casey. 2004.** 2004. Proceedings of the 37th Hawaii Intrnational Conference on System Sciences.
- Bass, Len, Clements, Paul y Kazman, Rick. 1998.** *Software Architecture in Practice*. 1998.
- Calderón, Amaro, y otros. 2007.** *Metodologías Ágiles*. Trujillo : s.n., 2007.
- Departamento de Ingeniería de Software. 2007-2008.** *Introducción al proceso de desarrollo de software*. Habana : s.n., 2007-2008.
- Díaz Díaz, Adrian. 2013.** *XStormAlmiquiServer, servidor para el cliente de visualización web del SAINUX Almiquí*. Habana : s.n., 2013.
- Eguiluz, Javier. 2012.** *Desarrollo web ágil con Symfony2*. 2012.
- Fernández G, José. 2013.** *Seguridad en Informática*. 2013.
- Gibert Ginestà, Marc y Pérez Mora, Oscar. 2016.** Universitat Oberta de Catalunya (UOC). *Universitat Oberta de Catalunya (UOC)*. [En línea] 2016. [http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\\_M2109\\_02152.pdf](http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf).
- Gómez, Daniel P. Valdés. 2015.** SlidePlayer. [En línea] 2015. <http://slideplayer.es/slide/4311994/>.
- Gutiérrez, J J, y otros. 2006.** *Pruebas del Sistema en Programación Extrema*. Sevilla : s.n., 2006.
- Letelier, Patricio y Penadés, María Carmen. 2003.** *Metodologías ágiles para el desarrollo de software: eXtreme Programming*. 2003.
- Microsoft . 2016.** Developer Network. [En línea] 2016. <https://msdn.microsoft.com/es-es/library/bb972268.aspx>.
- Patrón Modelo-Vista-Controlador.* **Fernández Romero, Yenisleidy y Díaz González, Yanette. 2012.** 2012, Telem@tica, pág. 57.
- Pérez, María José. 2015.** *Guía Comparativa de Metodologías Ágiles*. 2015.
- Pfleeger, Charles, Lawrence, Shari y Margulies, Jonathan. 2015.** *Security in Computing*. 2015.
- Ponjuán, Dante. 1998.** *Gestión de Información en la Organización (Principios, conceptos y aplicaciones)*. Santiago de Chile : s.n., 1998.
- Potencier, Fabien y Zaninotto, François. 2008.** *Symfony la guía definitiva*. s.l. : Eyrolles Collection, 2008.
- Pressman, Roger S. 2001.** *Ingeniería de Software. Un enfoque Práctico*. 5ta. 2001.
- Ramos Betacourt, Surenis y León Rdoríguez, Ivette. 2012.** *Módulo de Alarmas y Acciones ante Incidencias*. Habana : s.n., 2012.
- Reynoso, Carlos y Kiccillof, Nicolás. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Buenos Aires : s.n., 2004.
- Rodríguez González, Pilar. 2008.** *Estudio de la aplicación de metodologías ágiles para la evolución de productos de software*. 2008.
- Sandra Almeida, Adriana y Perez Cavenago, Vanina. 2007.** *Arquitectura de Software: Estilos y Patrones*. Argentina : s.n., 2007.
- Seguridad Informática y Cartografía. 2006.** *Seguridad Informática y Cartografía*. 2006.
- Sommerville, Ian. 2005.** *Ingeniería de Software*. 7ma. Madrid : Pearson Educación. S.A, 2005.
- Tenable Network Security. 2016.** Tenable network security. [En línea] 2016. <http://www.tenable.com/es/nessus/>.
- The Eniac Corporation. 2015.** *ACL. La clave para Auditoría, Cumplimiento, Monitoreo Continuo y Detección de Fraudes*. Venezuela : s.n., 2015.
- The PHP Group. 2016.** PHP. [En línea] 2016. <http://php.net/>.

## Bibliografía

**Torres Matos, Yulissa y Morales González, Yanelis. 2011.** *Plataforma de Gestión de Hardware y Software. Módulo: Mapa tecnológico.* Habana : s.n., 2011.

**UCI. 2015.** Entorno Visual de Aprendizaje. [En línea] 25 de 07 de 2015. eva.uci.cu.

## Anexos

### Anexos

---

**Anexo 1.** Tareas de Ingeniería asociada la historia de usuario 3\_ Exportar reportes PDF (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 3.1</b>	<b>Número Historia de Usuario: HU_3</b>
<b>Nombre Tarea:</b> Exportar reportes PDF.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 9
<b>Fecha Inicio:</b> 15/02/2016	<b>Fecha Fin:</b> 19/02/2016
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> devuelve un archivo PDF con los resultados obtenidos en el escaneo.	

**Anexo 2.** Tareas de Ingeniería asociada la historia de usuario 5\_Filtrar por laboratorio (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 5.1</b>	<b>Número Historia de Usuario: HU_5</b>
<b>Nombre Tarea:</b> Filtrar por laboratorio.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 12-11-2015	<b>Fecha Fin:</b> 18-11-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite obtener la información relacionada al laboratorio proporcionado.	

**Anexo 3.** Tareas de Ingeniería asociada la historia de usuario 5\_Filtrar por IP (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 5.2</b>	<b>Número Historia de Usuario: HU_5</b>
<b>Nombre Tarea:</b> Filtrar por IP.	

## Anexos

<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 19-11-2015	<b>Fecha Fin:</b> 20-11-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite obtener la información relacionada al IP proporcionado.	

**Anexo 4.** Tareas de Ingeniería asociada la historia de usuario 5\_Filtrar por medio básico (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 5.3	<b>Número Historia de Usuario:</b> HU_5
<b>Nombre Tarea:</b> Filtrar por medio básico.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 24-11-2015	<b>Fecha Fin:</b> 26-11-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite obtener la información relacionada al medio básico proporcionado.	

**Anexo 5.** Tareas de Ingeniería asociada la historia de usuario 5\_Filtrar por política (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 5.4	<b>Número Historia de Usuario:</b> HU_5
<b>Nombre Tarea:</b> Filtrar por política.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 27-11-2015	<b>Fecha Fin:</b> 1-12-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite obtener la información relacionada a la política proporcionada.	

## Anexos

**Anexo 6.** Tareas de Ingeniería asociada la historia de usuario 5\_Filtrar por fecha (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 5.5</b>	<b>Número Historia de Usuario: HU_5</b>
<b>Nombre Tarea:</b> Filtrar por fecha.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 14/12/2015	<b>Fecha Fin:</b> 14/12/2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite obtenerla información relacionada a la fecha proporcionada.	

**Anexo 7.** Tareas de Ingeniería asociada la historia de usuario 6\_ Realizar reporte del estado de las vulnerabilidades (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 6.1</b>	<b>Número Historia de Usuario: HU_6</b>
<b>Nombre Tarea:</b> Realizar reporte del estado de las vulnerabilidades.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 9-11-2015	<b>Fecha Fin:</b> 9-11-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> obtiene un reporte de las vulnerabilidades detectadas en el escaneo realizado.	

**Anexo 8.** Tareas de Ingeniería asociada la historia de usuario 6\_ Realizar reporte del software y hardware (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 6.2</b>	<b>Número Historia de Usuario: HU_6</b>
<b>Nombre Tarea:</b> Realizar reporte del software y hardware.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1

## Anexos

<b>Fecha Inicio:</b> 10-11-2015	<b>Fecha Fin:</b> 10-11-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> obtiene un reporte de las vulnerabilidades asociadas a problemas de software y hardware.	

**Anexo 9.** Tareas de Ingeniería asociada la historia de usuario 6\_ Realizar reporte de reincidentes (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 6.3	<b>Número Historia de Usuario:</b> HU_6
<b>Nombre Tarea:</b> Realizar reporte de incidencias.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 11-11-2015	<b>Fecha Fin:</b> 11-11-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> obtiene un reporte de las incidencias detectadas.	

**Anexo 10.** Tareas de Ingeniería asociada la historia de usuario 8\_ Crear laboratorio (del autor)

Tarea de Ingeniería	
<b>Número Tarea:</b> 8.1	<b>Número Historia de Usuario:</b> HU_8
<b>Nombre Tarea:</b> Crear laboratorio.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 28-9-2015	<b>Fecha Fin:</b> 29-9-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite crear un laboratorio con la especificación de la subred que le corresponde.	

**Anexo 11.** Tareas de Ingeniería asociada la historia de usuario 8\_ Eliminar laboratorio (del autor)

Tarea de Ingeniería
---------------------

## Anexos

<b>Número Tarea: 8.2</b>	<b>Número Historia de Usuario: HU_8</b>
<b>Nombre Tarea:</b> Eliminar laboratorio.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 30-9-2015	<b>Fecha Fin:</b> 30-9-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite eliminar un laboratorio.	

**Anexo 12.** Tareas de Ingeniería asociada la historia de usuario 8\_ Modificar laboratorio (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 8.3</b>	<b>Número Historia de Usuario: HU_8</b>
<b>Nombre Tarea:</b> Modificar laboratorio.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 1-10-2015	<b>Fecha Fin:</b> 1-10-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite modificar las características definidas para un laboratorio.	

**Anexo 13.** Tareas de Ingeniería asociada la historia de usuario 9\_ Adicionar datos de auditoría en un área (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 9.1</b>	<b>Número Historia de Usuario: HU_9</b>
<b>Nombre Tarea:</b> Adicionar datos de auditoría en un área.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 26-10-2015	<b>Fecha Fin:</b> 30-10-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> el sistema permite adicionar información de un área determinada.	

## Anexos

**Anexo 14.** Tareas de Ingeniería asociada la historia de usuario 9\_ Modificar datos de auditoría en un área determinada (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 9.2</b>	<b>Número Historia de Usuario: HU_9</b>
<b>Nombre Tarea:</b> Modificar datos de auditoría en un área determinada	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 2-11-2015	<b>Fecha Fin:</b> 6-11-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> el sistema permite modificar la información de un área determinada.	

**Anexo 15.** Tareas de Ingeniería asociada la historia de usuario 10\_ Desconectar usuario (del autor)

Tarea de Ingeniería	
<b>Número Tarea: 10.1</b>	<b>Número Historia de Usuario: HU_10</b>
<b>Nombre Tarea:</b> Desconectar usuario.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 23-01-2015	<b>Fecha Fin:</b> 23-01-2015
<b>Programador Responsable:</b> Jorge Luis Hernández Suárez	
<b>Descripción:</b> permite al usuario una vez concluido el uso del sistema salir de él.	