



Facultad 4

Sistema para el control de activos informáticos de la facultad 4

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

Autores: Yasel Herrera Vive

Luis García Membribes

Tutores: Ing. Michel Miranda Cairo

Co-tutor: Ing. Renier Portelles Cobas

“La Habana, junio del 2014”



“La única lucha que se pierde es la que se abandona”

Ernesto Che Guevara

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del presente trabajo titulado: "Sistema para el control de activos informáticos de la facultad 4" y autorizamos a la Universidad de las Ciencias Informáticas y al Departamento de Tecnología de la facultad a usarlo en su beneficio.

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

Yasel Herrera Vive

Luis García Membrives

Ing. Michel Miranda Cairo

Tutor

Ing. Renier Portelles Cobas

Co-tutor

Agradecimientos

Yasel

Quiero agradecer a toda mi familia en especial a mis padres, que siempre estuvieron presente en cada momento de mi vida, me apoyaron incondicionalmente en todas mis decisiones y acciones, nunca me dieron la espalda a pesar del problema que haya tenido, siempre me dieron su amor y cariño, también quiero agradecer a mi hermana, mi sobrinita preciosa, a mi novia por todo el apoyo, ayuda, cariño, alegría que le ha dado a mi vida desde que la conocí y a mi prima Julie por toda su ayuda.

Agradecer a mi tutor Michel por toda su ayuda incondicional, a su esposa Diana también, a todo el Departamento de Tecnología de la facultad y a mi compañero de tesis.

Un agradecimiento a mi grupo 4502 en especial a Dailen, Annelys, Pavel, Yordanis, Remberto, Raúl, a Victor por todas las veces que lo moleste en el apartamento y siempre me brindó su mano y su amistad, a Enrique Mejías por su ayuda.

Un agradecimiento a mis amigos Zayli, Evis que estuvieron siempre presente en toda mi vida estudiantil, a Carlos Rafael García Caballero, Yasmani Bargallo, Michel Rivera y a todos mis amigos.

Luis

A mis padres, que han sido mi motor impulsor para seguir adelante, por haberme apoyado siempre, no dejar que perdiera las esperanzas y confiar en que era capaz de llegar tan lejos en la vida, le agradezco a mi novia por toda su ayuda y comprensión en este tiempo.

A toda mi familia en especial a Oxana y su esposo Luis Enrique que han sido mis segundos padres en esta trayectoria, a mis abuelos por preocuparse tanto por mí y darme tan buenos consejos.

A mi compañero de tesis por estar conmigo en las buenas y en las malas. A mi tutor por habernos guiado con toda su ayuda, comprensión y paciencia. A mis hermanos de la UCI Osmel, Willy por aceptarme como soy y acompañarme en los momentos difíciles. Además a mis compañeros de aula y de apartamento.

Dedicatoria

Dedico esta tesis especialmente a mis padres

Carmela y Marcelino, a mi hermana

Yissel, a mi sobrinita Camila,

a mis abuelas Paula y

Lourdes por darme

todo su amor.

Yasel

A las personas que más quiero en esta vida,

a mi querida madre, a mi padre,

a mi hermana y a mis

abuelos. A toda

mi familia.

Luis

Resumen

En el mundo de hoy las compañías y organizaciones son completamente dependientes de la tecnología para llevar a cabo sus objetivos, la seguridad informática toma un papel fundamental dado la necesidad de proteger la confidencialidad, disponibilidad e integridad de la información. El control de activos informáticos es uno de los aspectos de primer orden según los estándares internacionales y dado su complejidad en la facultad 4 de la Universidad de las Ciencias Informáticas (UCI) se ha convertido en un proceso demandante y en ocasiones ineficiente. Dado esta situación, la presente investigación asume como objetivo principal la implementación de una aplicación para la informatización del control de los activos informáticos de la facultad 4, basado en la metodología de desarrollo XP, utilizándose como herramientas para el desarrollo de la misma los framework, Bootstrap 3 y Symfony 2 para el entorno web, Nodejs para el servidor y como IDE de desarrollo Qt-creator para el cliente. Este sistema es capaz de buscar incidencias entre las computadoras, generar un expediente técnico y su historial de modificaciones de forma informatizada, brinda la posibilidad de imprimirlos ya con los datos correspondientes, cuenta con una gestión de áreas, la misma tiene incorporada un local, un responsable así como también se puede llevar un control de las computadoras rotas en dichas áreas, optimiza el trabajo llevado a cabo por el personal del Departamento de Tecnología de la facultad 4, así como el llenado de datos del expediente técnico de forma rápida y eficaz.

ÍNDICE

Agradecimientos	II
Dedicatoria.....	III
Resumen	IV
Introducción	1
Capítulo 1 Fundamentación teórica.....	6
1.1 Introducción	6
1.2 Conceptos asociados	6
1.2.1 Seguridad informática	6
1.2.2 Principales estándares de seguridad.....	6
1.2.3 Control de seguridad informática.....	8
1.2.4 Inventario de activos.	8
1.3 Análisis crítico de soluciones similares	9
1.3.1 OCSInventory-NG	10
1.3.2 GLPI.....	11
1.3.3 Resultados del análisis de soluciones	13
1.4 Metodologías de desarrollo	13
1.4.1 Extreme Programing (XP)	13
1.4.2 Scrum	14
1.4.3 Rational Unified Process (RUP)	14
1.4.4 Resultados del análisis de metodologías para el desarrollo	15
1.5 Herramientas y lenguajes de programación.....	16
1.5.1 Herramienta de modelado Visual Paradigm	16
1.5.2 Lenguaje de Modelado Unificado (UML)	16
1.6 Tecnologías del lado del servidor -PHP	17
1.7 Tecnologías del lado del cliente.....	17
1.7.1 Hyper Text Markup Language (HTML)	17
1.7.2 Javascript.....	17
1.7.3 Cascading Style Sheets (CSS).....	18
1.7.4 C++	18
1.8 Entorno de desarrollo integrado (IDE)	19
1.8.1 QT-Creator.....	19

1.8.2 PHP Storm 7.0	19
1.9 Framework de desarrollo	20
1.9.1 Symfony 2.3.5	20
1.9.2 Bootstrap 3.....	20
1.10 Servidor web	21
1.10.1 Apache.....	21
1.10.2 Nodejs.....	21
1.11 Sistema gestor de base de datos – MySQL.....	22
1.12 Propuesta y selección de las herramientas.....	22
1.13 Conclusiones del capítulo	22
Capítulo 2 Propuesta de solución.....	24
2.1 Introducción.....	24
2.1.1 Objetos a informatizar	24
2.1.2 Propuesta de solución.....	25
2.2 Personal relacionado con el sistema	25
2.3 Lista de reserva del producto.....	26
2.4 Aspectos no funcionales del sistema	27
2.5 Fase de Exploración	28
2.5.1 Historia de Usuario.....	28
2.6 Fase de planificación	37
2.6.1 Estimación del esfuerzo por HU	37
2.6.2 Plan de iteraciones.....	38
2.6.3 Plan de duración de las iteraciones.....	39
2.6.4 Plan de entrega.....	40
2.7 Patrones de diseño.....	41
2.7.1 Patrones GRASP	41
2.7.2 Patrones GOF	42
2.8 Estándar de implementación	43
2.9 Tarjetas CRC.....	44
2.10 Diagrama de componentes del subsistema UCInventoryWebInterface.....	45
2.11 Diagrama de despliegue.....	48
2.12 Conclusiones del capítulo	49
Capítulo 3 Pruebas.....	50

3.1 Introducción.....	50
3.2 Planificación de pruebas.....	50
3.2.1 Plan de pruebas por iteración.....	50
3.2.2 Resumen de resultados de la etapa de pruebas	55
3.3 Conclusiones.....	56
Conclusiones generales.....	57
Recomendaciones	58
Referencias.....	59
Glosario de términos.....	62
Anexos.....	64
Anexo 1: Casos de pruebas	64
Anexo 2: Expediente técnico de computadoras.....	71
Anexo 3: Manual de usuario para UCInventory	72
Anexo 4: No conformidades	78
Anexo 5: Patrones de diseño.....	79

ÍNDICE DE FIGURAS

Figura 1: OCS Inventory NG	11
Figura 2: GLPI.....	12
Figura 4: Relación de clases utilizando el patrón de diseño Experto	41
Figura 5: Relación de clases utilizando el patrón de diseño Experto	42
Figura 6: Relación de clases utilizando el patrón de diseño Creador	42
Figura 7: Estructura que utiliza el patrón de diseño Decorator	43
Figura 8: Instancias de clase que utiliza el estándar de implementación CamelCase	43
Figura 9: Diagrama de componente del subsistema UCInventoryWebInterface	47
Figura 10: Diagrama de despliegue	48
Figura 11: Resultados de las Pruebas de Aceptación en cada Iteración.	55
Figura 12: Expediente técnico.....	72
Figura 13: Login	74
Figura 14: Inicio	74
Figura 15: Crear área.....	75
Figura 16: Listar área.....	75
Figura 17: Editar área	75
Figura 18: Eliminar área.....	76
Figura 19: Listar computadoras nuevas	76
Figura 20: Asignar área.....	76
Figura 21: Asignar área.....	76
Figura 22: Asignar área.....	77
Figura 23: Generar expediente técnico	77
Figura 24: Generar expediente técnico	78
Figura 25: Historial de modificaciones.....	78
Figura 26: Patrón Decorator.....	79

ÍNDICE DE TABLAS

Tabla 1: Comparación entre herramientas existentes	12
Tabla 2: Comparación entre las metodologías	14
Tabla 3: Personal relacionado con el sistema	26
Tabla 4: HU-Autenticar usuario	29
Tabla 5: HU-Crear área.....	29
Tabla 6: HU-Eliminar área.....	30
Tabla 7: HU-Editar área	30
Tabla 8: HU-Listar área.....	30
Tabla 9: HU-Asignar área a una computadora	31
Tabla 10: HU-Generar expediente técnico	31
Tabla 11: HU-Adicionar datos al expediente técnico	32
Tabla 12: HU-Imprimir expediente técnico	32
Tabla 13: HU-Ver historial de modificaciones.....	32
Tabla 14: HU-Imprimir historial de modificaciones	33
Tabla 15: HU-Editar historial de modificaciones	33
Tabla 16: HU-Reportar como rota una computadora.....	34
Tabla 17: HU-Eliminar computadora rota	34
Tabla 18: HU-Adicionar computadora manualmente.....	34
Tabla 19: HU-Aceptar cambios en historial de modificaciones	35
Tabla 20: HU-Eliminar historial de modificaciones	35
Tabla 21: HU-Ver historial de fecha	35
Tabla 22: HU-Eliminar historial de fecha	36
Tabla 23: HU-Buscar incidencia en la computadora.....	36
Tabla 24: HU-Enviar notificación de correo electrónico	37
Tabla 25: Estimación de esfuerzo por HU	37
Tabla 26: Plan de iteraciones.....	38
Tabla 27: Plan de duración de las iteraciones.....	39
Tabla 28: Plan de entrega.....	40
Tabla 29: Tarjeta CRC Áreas	44
Tabla 30: Tarjeta CRC computadoraController	44
Tabla 31: Tarjeta CRC historialModificaciones.....	45
Tabla 32: Tarjeta CRC historialFecha	45
Tabla 33: Tarjeta CRC SecurityContext	45

Tabla 34: Prueba de aceptación. No 1	52
Tabla 35: Prueba de aceptación. No 2	53
Tabla 36: Prueba de aceptación. No 6	53
Tabla 37: Prueba de aceptación. No 7	54
Tabla 38: Prueba de aceptación. No 20	54
Tabla 39: Prueba de aceptación. No 3	64
Tabla 40: Prueba de aceptación. No 4	64
Tabla 41: Prueba de aceptación. No 5	65
Tabla 42: Prueba de aceptación. No 8	65
Tabla 43: Prueba de aceptación. No 9	66
Tabla 44: Prueba de aceptación. No 10	66
Tabla 45: Prueba de aceptación. No 11	67
Tabla 46: Prueba de aceptación. No 12	67
Tabla 47: Prueba de aceptación. No 13	68
Tabla 48: Prueba de aceptación. No 14	68
Tabla 49: Prueba de aceptación. No 15	69
Tabla 50: Prueba de aceptación. No 17	69
Tabla 51: Prueba de aceptación. No 18	70
Tabla 52: Prueba de aceptación. No 19	70
Tabla 53: Prueba de aceptación. No 20	71
Tabla 54: Prueba de aceptación. No 21	71

Introducción

En la actualidad las compañías y organizaciones son completamente dependientes de la tecnología para llevar a cabo sus objetivos. Actualmente toda información empleada es procesada, almacenada y transmitida en formato digital, esto facilita su uso y propagación. En un entorno en que el desarrollo tecnológico ha posibilitado la conexión a Internet desde cualquier lugar y mediante múltiples dispositivos electrónicos, esto provoca que los sistemas informáticos se encuentren constantemente expuestos a múltiples amenazas.

Uno de los principales riesgos, a los que se expone la información digital, son los ataques informáticos, principalmente aquellos que vinculan a los activos de una organización o institución. Estas amenazas pueden provocar la pérdida de disponibilidad, confidencialidad e integridad del contenido; las consecuencias implican graves pérdidas para las empresas y en muchas ocasiones se originan daños irreparables (1).

La Universidad de las Ciencias Informáticas es una institución que presenta uno de los sistemas de información más grandes del país, con un total de más de 6 mil estaciones de trabajo conectados en red, dando lugar a que se manifieste diversas amenazas. Una de las principales amenazas debido al impacto que causa sobre la misma es el robo de hardware.

EL control designado para la mitigación de estas amenazas consiste en la realización del inventario de activos. Todas las áreas de esta Universidad incluyendo la facultad 4 presentan algunos problemas en la realización de estos controles.

El inventario de activo a pesar de que existen software que informatizan este control, implica el levantamiento de muchos datos de manera manual. Además de que está establecido llevar el registro de todo el inventario en un expediente. Este proceso actualmente se realiza manualmente, haciendo que el trabajo del personal del Departamento de Tecnología sea un proceso complejo y engorroso, además provoca un conjunto de dificultades que confluyen en la pérdida, duplicidad y errores en la información registrada.

Según datos estadísticos con relación a incidentes de seguridad informática alcanzó un valor de 234 mil dólares en el año 2009. Teniendo en cuenta, además, el crecimiento exponencial de los programas malignos, los cuales aumentan por decenas de miles diariamente. De este modo

es posible afirmar que se hace necesario garantizar la confianza de los medios de cada establecimiento; el procedimiento se realizaría mediante la aplicación de un adecuado proceso de gestión que abarque todas las medidas necesarias (2).

Los controles de seguridad informática a establecer son muchos y muy variados. El estándar ISO/IEC 27001 propone más de 130 controles que están relacionados con aplicaciones, dispositivos tecnológicos, recursos humanos y cuestiones organizativas (3). La gran cantidad y variedad de controles que es necesario implementar, en un entorno donde la tecnología evoluciona a una gran velocidad, hace que la gestión de la seguridad informática sea un proceso complejo, en el que hay que lograr una adecuada armonía entre tecnología, personas y procedimientos (4).

Una de las vías para lograr que la gestión de la seguridad informática sea un proceso menos complejo y más efectivo, en un entorno de constantes amenazas de seguridad, y teniendo en cuenta la gran cantidad de medidas a implementar; es la automatización de controles de seguridad informática, basadas en un grupo de controles computacionales a realizar para mitigar las vulnerabilidades tanto del sistema como del hardware, algunos de estos son:

- Inventario de activos.
- Propiedad de los activos.
- Uso aceptable de los activos.
- Etiquetado y manipulado de la información.

En la actualidad existen una serie de sistemas de gestión de inventarios y su principal utilidad es que permiten tener un control completo de cualquier componente informático tanto en términos de hardware como de software. La gestión informatizada de un control de seguridad informática implica que la operación, monitorización y revisión del mismo se realizan de forma automática, mediante sistemas informáticos o herramientas de hardware sin que se produzca intervención humana en la realización de estas acciones.

Hay varias herramientas de gestión de inventario informático o informatización de la seguridad informática, aunque en el campo de la seguridad informática existen pocas investigaciones que abordan esta temática. La mayoría de los estudios académicos enfocan el tema desde una perspectiva puntual, refiriéndose a la informatización de controles específicos de seguridad informática, algunas de ellas son privativas y otras libres, por citar algunas: Total Network, Inventory Admin Express, ManagePC, GLPI y OCSInventory NG (6). Estas herramientas

internacionales son utilizadas por muchas entidades, facilitan el trabajo de llevar en cuenta los componentes físicos que tienen cada ordenador dentro de esa empresa.

En el Departamento de Tecnología de la facultad 4, se trabaja con OCSInventory NG y GLPI. La primera es una herramienta que permite realizar el inventario de hardware y software de los equipos de cómputo de una entidad de manera automatizada, lo que no puede realizar GLPI. La misma recopila toda la información de los equipos que forman parte de la red pero esto se realiza a través de un plugin instalado. Estas pueden utilizarse para visualizar el inventario a través de una interfaz web e incluye la posibilidad de instalar aplicaciones en los equipos de acuerdo a criterios de búsqueda definidos. También, posee la opción de escanear la red y de instalar aplicaciones remotamente. Un estudio realizado sobre estas herramientas arribó a detectar ciertas desventajas, las mismas provocan que el proceso de inventario de activos sea engorroso para la facultad, entre ellas se destacan:

- Carece de registro de historial.
- Carece de una gestión de área.
- El diseño de la aplicación no está orientado a objeto.
- Los reportes que realiza no tienen los formatos que hacen falta y no son personalizables.
- El proceso del levantamiento de información no es automatizado.
- Los datos se llenan manualmente.

Derivado de la situación anteriormente expuesta se define el siguiente **problema de la investigación**: ¿Cómo contribuir al control de los activos informáticos presentes en las distintas áreas de la facultad 4 de la UCI?

Como **objeto de estudio** se define el proceso de gestión de controles de seguridad informática y como **campo de acción** desarrollo de sistemas para el control automatizado de los activos informáticos en la facultad 4 de la UCI.

Para resolver el **problema de investigación** trazado se precisó como **objetivo general**: desarrollar un sistema para la informatización del control de los activos informáticos de la facultad 4.

A partir del **objetivo general** definido se derivan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.

- Realizar el diseño del modelo y la arquitectura para la gestión automatizada e integrada de controles de los activos informáticos.
- Implementar la aplicación necesaria del lado del cliente y el servidor para la realización de las operaciones de los controles automatizados de los activos informáticos.
- Realizar pruebas que permitan llevar a cabo el control de los activos informáticos.

Para dar respuesta a la interrogante presentada en este trabajo y con los objetivos trazados se plantea el cumplimiento de las siguientes **tareas de la investigación**:

- Estudio del proceso de inventario de hardware en la facultad 4.
- Levantamiento de controles relacionados con activos informáticos según la norma ISO 27001.
- Estudio sobre el funcionamiento de OCSInventory NG.
- Investigación sobre el servidor de Nodejs y el framework Symfony 2.3.5.
- Selección de la metodología y las herramientas.
- Realizar un levantamiento de requisitos para esta aplicación.
- Realizar el análisis y diseño de la aplicación.
- Desarrollar los requisitos gestionar área, gestionar computadora, gestionar reporte y hacer el cliente de Linux.

Los métodos utilizados en la elaboración de este trabajo con el fin de darle solución a las tareas de investigación fueron los siguientes:

En el nivel teórico:

- **Análítico-sintético:** este método se utiliza para analizar las diferentes definiciones, documentaciones y aplicaciones relacionada con la gestión automatizada del control de inventario de hardware para así tener un mejor conocimiento de lo que se define para hacer en el sistema.
- **Análisis histórico-lógico:** este método se utiliza para tener conocimiento del aumento de sistemas desarrollados con herramientas libres, así como otras temáticas relacionadas con la investigación.
- **Modelación:** este método se empleará con el objetivo de describir cómo funcionan los procesos del sistema y en el diseño de este a través del lenguaje de modelado unificado (UML).

Métodos empíricos:

- **Observación:** se utiliza para observar las actividades que se desarrollan en el Departamento de Tecnología de la facultad 4 y de esta manera determinar los procesos que se realizan.

Como **posibles resultados** se espera:

- Un sistema informatizado de los activos informáticos, integrando herramientas de seguridad informática y la automatización de un grupo de tareas que implica la operación, actuación o autorregulación independiente, sin intervención humana.
- Documentación del sistema del control de activos según la metodología de desarrollo que se seleccione.

El trabajo de diploma se encuentra estructurado de la siguiente forma:

Capítulo 1: Fundamentación Teórica:

Este capítulo contiene una base teórica para entender el problema planteado, donde se realiza un análisis sobre los sistemas de control de activos informáticos, además se hace una breve descripción de las tecnologías, herramientas y lenguajes a utilizar en el contexto del presente trabajo.

Capítulo 2: Propuesta de Solución:

En el presente capítulo se define la lista de reserva del producto y los aspectos no funcionales del sistema que debe tener y cumplir respectivamente la aplicación para su correcto funcionamiento. También se realizan las historias de usuario por cada aspecto funcional que tiene el sistema, un plan de entrega y se elaboran las tarjetas CRC, se explica la arquitectura empleada y los principales patrones de diseño utilizados.

Capítulo 3: Pruebas:

Se describe el proceso de detección y corrección de posibles errores existentes en la aplicación con la finalidad de garantizar su buen funcionamiento.

Capítulo **1** Fundamentación teórica

1.1 Introducción

En este capítulo se expondrán los conceptos teóricos que sustentan al proceso de gestión de controles de seguridad informática, específicamente el relacionado con la realización del inventario de hardware. Se presenta un examen crítico de sistemas o soluciones con respecto al tema, una propuesta y selección de las herramientas, tecnologías, metodologías y lenguajes que contribuyen al control automatizado de los activos informáticos presentes en las distintas áreas de la facultad 4 de la UCI.

1.2 Conceptos asociados

1.2.1 Seguridad informática

Actualmente existen un conjunto de definiciones que argumentan y ejemplifican el concepto de seguridad informática: la misma constituye un conjunto de métodos y herramientas destinados a proteger la información; por ende, los sistemas informáticos, ante cualquier amenaza, constituyen un proceso en el cual también participan personas (5). Por otra parte el término es catalogado, de igual modo como disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático (7).

1.2.2 Principales estándares de seguridad

Existen un conjunto de estándares y regulaciones relacionados con temas de seguridad informática que constituyen normas certificables y gozan de gran aceptación a nivel internacional. A continuación se detallan, a criterio de los autores de la presente investigación, las más importantes.

ISO/IEC 27000

La ISO ha reservado la serie de numeración 27000 para las normas relacionadas con sistemas de gestión de seguridad de la información. En el año 2005 se incluyó en ella la primera numeración de la serie ISO 27001. En próximos años está prevista la incorporación de nuevas

normas que supongan un apoyo para las organizaciones que implanten y certifiquen un SGSI según ISO 27001.

Entre otras, se encuentran:

- 27000 (términos y definiciones).
- 27002 (objetivos de control y controles, Contiene 39 objetivos de control y 133 controles, agrupados en 11 dominios.)
- 27003 (guía de implantación de un SGSI)
- 27004 (métricas y técnicas de medida de la efectividad de un SGSI)
- 27005 (guía para la gestión del riesgo de seguridad de la información)
- 27006 (proceso de acreditación de entidades de certificación y el registro de SGSI) (3).

ISO/IEC 27001

La ISO/IEC 27001 es un estándar diseñado para la gestión de la seguridad de la información utilizable por cualquier tipo de organización, pública o privada, grande o pequeña. Para la adecuada gestión de la seguridad de la información, es necesario implantar un sistema que aborde esta tarea de una forma metódica, documentada y basada en unos objetivos claros de seguridad y una evaluación de los riesgos a los que está sometida la información de la organización.

Este estándar se le aplica a organizaciones que procesen o manejen información y que requieran protegerla, como es la impresa o escrita en papel, almacenada electrónicamente, enviada por correo u otro medio electrónico, mostrada en videos corporativos y verbal expresada en conversaciones (9)(10).

El estándar ISO/IEC 27001 define como implantar un sistema de gestión de seguridad de la información que aporta a la organización interesantes beneficios:

- Aporta a la empresa un valor añadido, dando a nuestros clientes una mayor credibilidad, ya que contar con esta certificación, asegura que tenemos un proceso adecuado para la gestión de la información.
- Ofrece una metodología para llevar a cabo un análisis y gestión de riesgos.
- Garantiza la implantación de medidas de seguridad consistente, eficiente y apropiada al valor de la información protegida.

- Contempla planes de contingencia ante cualquier tipo de incidencia (pérdidas de datos, incendio, robo, terrorismo, etc.)
- Puede ser utilizada como herramienta de diferenciación frente a la competencia.
- Mejora la conciencia del personal en todo lo que se refiere a la seguridad y a sus responsabilidades dentro de la organización.

UNIT-ISO/IEC 27002

El estándar reconoce que el nivel de la seguridad que se puede alcanzar con medios técnicos es limitado. El nivel requerido de la seguridad establecida con la determinación de los niveles de riesgo y de costos asociados a las brechas de seguridad, contra los costos de implementar medidas de seguridad se debe conducir siempre por controles y procedimientos apropiados para la gestión. La gestión de la seguridad de la información requiere, como mínimo, participación de todos los empleados de la organización. Puede también requerir la participación de accionistas, proveedores, terceros y clientes.

La UNIT-ISO identifica los controles que forman el punto de partida para la seguridad de la información. Cubre los factores críticos del éxito; la organización de la seguridad de la información; la gestión de los activos; la seguridad ligada al personal; seguridad física y del entorno; la gestión de las comunicaciones y operaciones; la adquisición de sistemas de información; desarrollo y mantenimiento; gestión de incidentes; la gestión de la continuidad del negocio y conformidad. Está destinado a convertirse en una herramienta esencial para las organizaciones de cualquier tipo y tamaño, sean públicas o privadas (11).

1.2.3 Control de seguridad informática

Los controles de seguridad informática son mecanismos que se utilizan para inspeccionar los accesos y privilegios a los recursos indicados. Es responsabilidad del dueño del activo sobre el que se le aplican los controles establecer los parámetros requeridos para la disponibilidad, confidencialidad e integridad.

1.2.4 Inventario de activos.

Según la Resolución 127/07 del Ministerio de la Informática y las Comunicaciones de Cuba, se establece en el artículo 14 de la sección primera del Reglamento de Seguridad para las Tecnologías de la Información, la necesidad de que todos los bienes informáticos de una entidad deberán estar identificados y controlados, para lo cual se conformará y mantendrá

actualizado un inventario de estos incluyendo sus componentes y las especificaciones técnicas de aquellos que pudieran ser suplantados (12).

El inventario de activos, según la ISO/IEC 27002, es el encargado de identificar todos los activos, debiéndose elaborar y mantener un registro de los activos más importantes. Una entidad debe identificar todos los activos y documentar la importancia de estos (13). El inventario incluye toda la información necesaria para poder recuperarse de un desastre, incluyendo el tipo de activo, formato, ubicación, información de respaldo, información de licencias y un valor comercial. No se debe duplicar innecesariamente la información contenida en otros inventarios. Además, se debe acordar y documentar la propiedad y la clasificación de la propiedad para cada uno de los activos, estableciendo responsables para cada uno. Basado en la importancia de estos, su valor comercial y su clasificación de seguridad, se debe identificar los niveles de protección que se controlarán. Existen cinco tipos de activos fundamentales, los cuales son (13):

- Información: base de datos y archivos de datos, contratos y acuerdos, documentación del sistema, información de investigaciones, manuales del usuario, material de capacitación, procedimientos operacionales o de soporte, planes de continuidad del negocio, acuerdos para contingencias, rastros de auditoría e información archivada.
- Activos de software: software de aplicación, software del sistema, herramientas de desarrollo y utilidades.
- Activos físicos: equipo de cómputo, equipo de comunicación, medios removibles y otros equipos.
- Servicios: servicios de computación y comunicación, servicios generales; por ejemplo, calefacción, iluminación, energía y aire acondicionado, personas y sus calificaciones, capacidades y experiencia.
- Intangibles: tales como la reputación y la imagen de la organización.

Esta investigación centrará su atención fundamentalmente en el control de inventario referente a los activos físicos, en particular, según lo descrito en el campo de acción, a las computadoras que se encuentran en los locales de la facultad 4 de la UCI.

1.3 Análisis crítico de soluciones similares

A través de la realización del estudio de los antecedentes se pudieron identificar algunas soluciones que informatizan en alguna medida el control de inventario de activos. Estos

sistemas contienen una serie de deficiencias por la cual su uso es poco factible en el Departamento de Tecnología de la facultad 4, estas herramientas son:

1.3.1 OCSInventory-NG

Es un sistema de inventario y distribución de paquetes con interfaz web. Su arquitectura es de cliente-servidor. OCSInventory NG es de libre distribución, permite a los usuarios administrar el inventario de sus activos de tecnología informática. La misma recopila información sobre el hardware y software de los equipos que forman parte de la red.

Características:

- Filtros de búsqueda: por cantidad de memoria, Sistema Operativo.
- Ingreso de campos personalizados.
- Organización de los datos por los campos definidos.
- Compatible con GLPI para inventario automatizado.
- Licencia GNU/GPL (14).

Ventajas:

- Disponible en varios idiomas.
- El agente es soportado por la mayoría de los Sistemas Operativos: Windows (desde 95 en adelante), Linux y Mac OS.

Desventajas:

- Carece de registro de historial.
- Carece de una gestión de área.
- Interfaz poca intuitiva.
- Carece de un reporte de incidencia.
- La implementación de la aplicación no se desarrolla con el paradigma de programación orientada a objeto.

Account info: TAG	Last inventory	Computer	User	Operating system	RAM (MB)	CPU (MHz)	Select	Delete
VRIP-Asesor	2014-04-04 05:07:29	ANTONIOG	Antonio	Microsoft Windows XP Professional	1024	2197	<input type="checkbox"/>	<input type="checkbox"/>
Lab102-d4-pc08	2014-04-04 02:26:50	I202-d5-0016		Distribución cubana de GNU/Linux	975	2200	<input type="checkbox"/>	<input type="checkbox"/>
Lab102-d4-pc08	2014-04-04 02:25:40	I202-d5-0016		Distribución cubana de GNU/Linux	488	2800	<input type="checkbox"/>	<input type="checkbox"/>
Lab102-d4-pc08	2014-04-04 02:25:24	D05-104-PC01z		Distribución cubana de GNU/Linux	993	2800	<input type="checkbox"/>	<input type="checkbox"/>
Lab102-d4-pc08	2014-04-04 02:25:23	I202-d5-004	docencia	Distribución cubana de GNU/Linux	992	1599	<input type="checkbox"/>	<input type="checkbox"/>
Lab102-d4-pc08	2014-04-04 02:25:22	I202-d5-0014		Distribución cubana de GNU/Linux	993	2800	<input type="checkbox"/>	<input type="checkbox"/>
Lab102-d4-pc08	2014-04-04 02:25:22	I202-d5-008		Distribución cubana de GNU/Linux	993	2800	<input type="checkbox"/>	<input type="checkbox"/>

Figura 1: OCS Inventory NG

1.3.2 GLPI

Es una herramienta web que ofrece una gestión del inventario informático de una empresa, pudiendo definir tanto el hardware a todos los niveles, sean monitores, impresoras, ordenadores, como el software instalado e incluyendo datos como las licencias. Esta aplicación es libre y multiplataforma (15).

Ventajas:

- Permite generar informes.
- Detecta los dispositivos conectados a la red.

Desventajas:

- El proceso del levantamiento de información no es automatizado.
- Los datos se llenan manualmente.
- No cuenta con un historial de modificaciones.

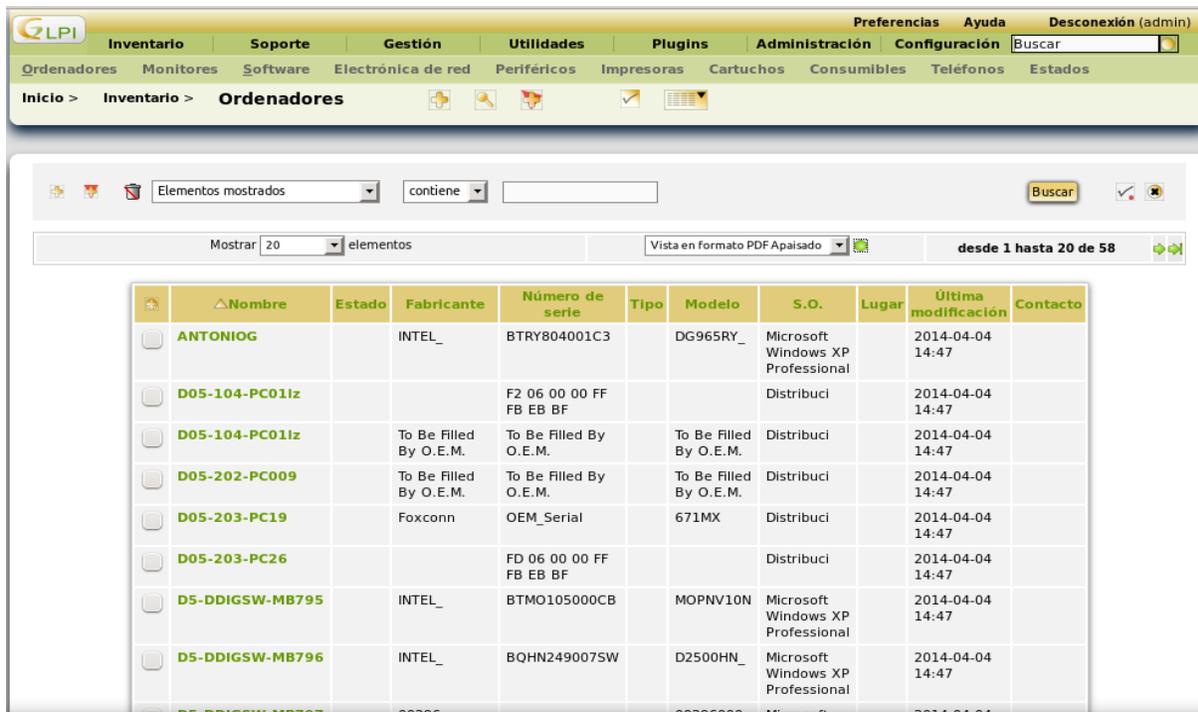


Figura 2: GLPI

Tabla 1: Comparación entre herramientas existentes

Características	GLPI	OCSInventory
Propósito	Inventario de Hardware y Software	Inventario de Hardware y Software
Automatización	NO(plugin OCSInventory NG)	Si
Activos	<ul style="list-style-type: none"> - Motherboard - Procesador - Memoria - Disco duro - Unidad óptica - Fuente - Tarjeta de video - Tarjeta de red - Monitor - Teclado - Mouse - Speaker 	<ul style="list-style-type: none"> - Motherboard - Procesador - Memoria - Disco duro - Unidad óptica - Fuente - Tarjeta de video - Tarjeta de red

	<ul style="list-style-type: none"> - UPS - Impresora - Scanner - Teléfono - Televisores 	
Historial	No	No
Personabilidad	Media	Baja
Reportes de incidentes	Si (Manual)	No

1.3.3 Resultados del análisis de soluciones

Derivado de lo anteriormente expuesto, referente a herramientas similares, se llegó a la conclusión que ninguna de las aplicaciones antes mencionadas cuentan con un historial de reportes, es de vital importancia saber si una computadora ha tenido modificaciones en sus componentes. Ninguna de estas dos aplicaciones consta con una gestión de áreas, ni con unas de las características más importantes llevadas a cabo en la Universidad de las Ciencias Informáticas como es la automatización del expediente técnico y su historial de modificaciones ya que son propias de la misma.

1.4 Metodologías de desarrollo

Una metodología de desarrollo de software es un conjunto de fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación que los desarrolladores deben seguir para garantizar la calidad de un producto informático. En la actualidad existe una serie de metodologías de desarrollo con particularidades específicas, entre las más conocidas están:

1.4.1 Extreme Programming (XP)

Kent Beck, el padre de XP, la describe como una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. Ésta se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico(16)(17). XP brinda la posibilidad de que los costos se reduzcan en el cambio de etapas del sistema, además de que

el trabajo con la documentación no se hace engorroso ya que el plazo del proyecto es corto producto a la reutilización de código.

1.4.2 Scrum

Es una metodología que permite definir un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral, prototipos, etc.) mediante la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar la poca predictibilidad y el riesgo a niveles aceptables.

Scrum se encarga de maximizar la realimentación sobre el desarrollo pudiendo corregir problemas y mitigar riesgos de forma temprana. Su uso se está extendiendo cada vez más dentro de la comunidad de metodologías ágiles, siendo combinado con otras como XP para completar sus carencias. No propone el uso de ninguna práctica de desarrollo en particular; sin embargo, es habitual emplearlo como un framework ágil de administración de proyectos que puede ser combinado con cualquiera de las metodologías mencionadas (18). Scrum es la metodología ágil que va trabajando iteración a iteración hasta llegar a la entrega final, al finalizar cada iteración se realiza una entrega funcional las cuales van a ir siendo utilizada por el cliente antes de recibir la entrega final del producto.

1.4.3 Rational Unified Process (RUP)

Es un proceso de ingeniería de software, que hace una propuesta orientada por disciplinas para lograr las tareas y responsabilidades de una organización que desarrolla software. Su meta principal es asegurar la producción de software de alta calidad que cumpla con las necesidades de los usuarios, con una planeación y presupuesto predecible. RUP está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental, esta metodología es conceptualmente amplia y diversa, está orientada a objetos y consta de cuatro fases por la que pasa el software: Inicio, Elaboración, Construcción y Transición (19).

Tabla 2: Comparación entre las metodologías

Características	XP	RUP	SCRUM
Costo	Reduce el costo del cambio en las etapas de vida del sistema.	Un cambio en las etapas de vida del sistema incrementaría notablemente el	Reduce los costos del plan interno de mejora de procesos.

		costo.	
Trabajadores	Se requiere un grupo pequeño de programadores para trabajar con esta metodología.	Requiere un grupo grande de programadores para trabajar con esta metodología.	Requiere un grupo pequeño de programadores desde 1-9 para trabajar con esta metodología.
Obtención de requisitos	Historia de usuario	Casos de uso	Resultados tangibles
Documentación	Poca	Mucha	Poca
Duración de Proyecto	Corta	Larga	Corto
Reutilización del código	Si	Si	Si
Simplicidad en el diseño	Si	No	Si
Diseño simple	Si	No	Si
Evaluación del estado de proyecto	Corto	Largo	Corto
Accesibilidad al código fuente por parte del cliente	Mucha	Poca	Mucha

1.4.4 Resultados del análisis de metodologías para el desarrollo

La mayoría de los proyectos de software fracasan porque exceden sus plazos, superan su presupuesto, no se ajustan a las auténticas necesidades del cliente, presentan una calidad deficiente o en muchos casos, son abortados.

Debido a las carencias de las herramientas estudiadas se decide proseguir con el desarrollo de un sistema que cumpla con las necesidades del Departamento de Tecnología de la facultad 4. Este proceso debe ser guiado por una metodología de desarrollo que se ajuste a las características del proyecto. En el caso de esta investigación se opta por el uso de la metodología de desarrollo XP debido a la envergadura de la solución que se propone, las características del equipo de trabajo y que se cuenta además con el cliente como parte de este equipo.

1.5 Herramientas y lenguajes de programación

1.5.1 Herramienta de modelado Visual Paradigm

Visual Paradigm es una herramienta de lenguaje modelado unificado (UML) profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejoras y reduce el costo. Permite dibujar todo tipo de diagramas de clases, código inverso, así como generar código y documentación. Esta herramienta UML cuenta con una serie de características (20):

- Ingeniería inversa - código a modelo, código a diagrama.
- Generación de código - modelo a código, diagrama a código.
- Editor de detalles de casos de uso - Entorno todo en uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas de flujo de datos.
- Soporte ORM - Generación de objetos Java desde la base de datos.
- Generación de base de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML (20).

La herramienta CASE¹ Visual Paradigm es muy importante en el desarrollo de sistemas ya que brinda la posibilidad de realizar una serie de diagramas, para que el cliente tenga un mejor entendimiento de lo que se quiere hacer en el sistema, al igual los programadores a la hora de implementar deben de regirse por lo modelado.

1.5.2 Lenguaje de Modelado Unificado (UML)

UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Este ofrece un estándar para describir un plano del modelo, incluyendo aspectos conceptuales tales como procesos de negocios, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y

¹ Ingeniería de Software Asistida por Computadora

componentes de software reutilizables. Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software y detallar los artefactos en el sistema. En otras palabras, es el lenguaje donde está descrito el modelo (21).

1.6 Tecnologías del lado del servidor -PHP

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Una de las características más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos como Oracle, PostgreSQL, MySQL. Soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. También se pueden abrir sockets de red directos (raw sockets) e interactuar con otros protocolos (22). Es especialmente creado para el desarrollo de páginas web dinámicas.

1.7 Tecnologías del lado del cliente

Las tecnologías del lado del cliente son aquellas que pueden ser directamente interpretadas por el navegador y no necesitan un pre-tratamiento. El navegador web es llamado cliente web debido a que realiza tareas de solicitud y consumo de servicios. A continuación se mencionarán las tecnologías del lado del cliente a emplear.

1.7.1 Hyper Text Markup Language (HTML)

HTML es un lenguaje de composición, documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque indica como desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. Define cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos, así como otros recursos de Internet, como FTP y Gopher (23).

1.7.2 Javascript

Javascript es un lenguaje de programación del lado del cliente, se encarga de soportar la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript se pueden crear efectos en las páginas y definir interactividades con el usuario. El navegador del cliente se encarga de interpretar las instrucciones de este lenguaje y ejecutarlas para realizar los

efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. Javascript es el siguiente paso, después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos (24).

Este lenguaje del lado del cliente brinda la posibilidad de que el sistema implementado tenga la seguridad necesaria del lado del cliente para que no sea invadido por personas que no tenga permiso a la aplicación. Permite validar todos aquellos campos que contenga el HTML de la aplicación para un mejor aseguramiento de los datos entrados.

1.7.3 Cascading Style Sheets (CSS)

CSS es un lenguaje principalmente utilizado por parte de los navegadores web y por los programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc. CSS se basa en intentar separar lo que es la estructura del documento HTML de su presentación. Por decirlo de alguna manera la página web sería lo que hay debajo es decir el contenido y CSS sería un cristal de color que hace que el contenido se vea de una forma u otra (25).

Su uso trae como principales ventajas el control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario (26). CSS está vinculado con la página HTML, brinda la interfaz del sistema, mejor posicionamiento, sombreado, efectos de los elementos y otra serie de cualidades.

1.7.4 C++

C++ es un lenguaje imperativo orientado a objetos derivado de C. En realidad un súper conjunto de C, que nació para añadirle cualidades y características de las que carecía. El resultado es que como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten un estilo de programación con alto nivel de abstracción (27).

1.8 Entorno de desarrollo integrado (IDE)

Un entorno de desarrollo integrado (IDE) es una aplicación de software, proporciona servicios integrales a los programadores de software. Normalmente está compuesto por un editor de código fuente, un compilador y un intérprete, la automatización de generación de herramientas y un depurador. Los IDE pueden ser aplicaciones que funcionan por sí solas o pueden ser parte de aplicaciones existentes, poseen entre sus características más importantes compilar más de un lenguaje de programación. Para el desarrollo de la propuesta de solución se hará uso de los IDE:

1.8.1 QT-Creator

Qt-creator es un IDE para el desarrollo de aplicaciones multiplataforma. La función más conocida de Qt es la creación de interfaces de usuario, sin embargo no se limita sólo a esto. Provee varias clases para facilitar ciertas tareas de programación. Tiene una buena comunicación con bases de datos, manejo de cadenas de caracteres y soporte para XML. Ofrece aplicaciones para facilitar y agilizar las tareas de desarrollo. Está disponible para los sistemas operativos Linux, Mac y Windows, permite al desarrollador crear aplicaciones para múltiples sistemas o plataformas móviles (28)(29).

1.8.2 PHP Storm 7.0

Es un potente entorno de desarrollo integrado (IDE), especialmente diseñado a fin de proporcionar a los desarrolladores de HTML, Javascript y PHP todas las herramientas necesarias para este trabajo, esta herramienta cuenta con una serie de funcionalidades las cuales son:

- Motor de inferencia de variables y atributos PHP más sofisticado.
- Nueva vista del patrón de arquitectura MVC para proyectos en Symfony2.
- Posibilidad de especificar de manera muy sencilla su propio PHP, así como el conjunto de reglas de programación, para su proyecto.
- Numerosas incorporaciones en cuanto a compatibilidad con bases de datos creadas en SQL, incluyendo refactorización de esquemas de bases de datos a tiempo real, edición de procedimientos almacenados, generación de scripts de migración a nuevos esquemas y función de completado de código SQL mejorada.
- Modo de edición a tiempo real para códigos en PHP/HTML/CSS/JS y nueva función de pre visualización en el navegador.

- Provee mejoras para la edición de Javascript y CSS (30).

1.9 Framework de desarrollo

Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, pueden servir de base para la organización y desarrollo de software. Éstos pueden incluir soportes de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. Tomando así los framework de desarrollo.

1.9.1 Symfony 2.3.5

Symfony es un completo framework de desarrollo de aplicaciones web diseñado para optimizar gracias a sus características basado en el patrón de diseño Modelo-Vista-Controlador. Separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Este framework es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas (Unix, Linux, etc.) como en plataformas Windows y es independiente del sistema gestor de base de datos (31). Es lo suficientemente estable como para desarrollar aplicaciones a largo plazo. Su código es fácil de leer e incluye comentarios de phpDocumentor permitiendo un mantenimiento muy sencillo. Emplea una potente línea de comandos que facilita la generación de código, lo cual contribuye a ahorrar tiempo de trabajo.

1.9.2 Bootstrap 3

Bootstrap es un framework que permite crear interfaces web con CSS y Javascript, adapta la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, esto se denomina diseño adaptativo o diseño sensible. Ofrece todas las posibilidades de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: botones, menús desplegables, formularios incluyendo todos sus elementos e integración JQuery para ofrecer ventanas.

1.10 Servidor web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS. Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

- Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
- Recibe una petición.
- Busca el recurso.
- Envía el recurso utilizando la misma conexión por la que recibió petición.
- Vuelve al segundo punto.

1.10.1 Apache

Apache es un servidor web que se carga como un servicio más del sistema operativo cuando está activo, convierte la máquina en un servidor capaz de enviar contenido a cualquier navegador. Para comprobar si el servidor funciona se debe abrir un navegador y escribir localhost como dirección. Un mensaje se informará del éxito en la instalación. A partir de ese momento se debe copiar el contenido web que se quiera servir a un directorio (32). Este servidor brinda la posibilidad de cargar aplicaciones desarrolladas en Symfony para enviar los datos a navegadores web.

1.10.2 Nodejs

Es un intérprete Javascript del lado del servidor basado en eventos que cambia la noción de cómo debería trabajar un servidor. Su meta es permitir a un programador construir aplicaciones altamente escalables y escribir código que maneje decenas de miles de conexiones simultáneas en una sola máquina física. Nodejs es adecuado cuando se necesita hacer muchas peticiones al mismo tiempo, sobre todo muchas operaciones como (acceso a ficheros, bases de datos a la vez y es especialmente bueno para aplicaciones en tiempo real).

Ventajas de Nodejs

- Está basado en eventos, así que toda la filosofía asíncrona que se utiliza con AJAX en el cliente se puede pasar al servidor.
- Permite utilizar el mismo lenguaje (javascript) tanto en el cliente como en el servidor.

- Permite hacer en el servidor todo lo que se necesita (acceso a ficheros, a bases de datos, conexiones de clientes) (33).

Nodejs cuenta con gran velocidad debido a un motor V8 para compilar código javascript. Es de código abierto y se ejecuta en varios sistemas operativos como son (Windows, Linux y Mac). Además las conexiones de red son concurrentes permitiendo que este nunca se quede en punto muerto a la hora de entrar o salir.

1.11 Sistema gestor de base de datos – MySQL

MySQL preferentemente es usado para desarrollos Web 2.0 y Enterprise 2.0. Soporta gran cantidad de tipos de datos para columnas. Dispone de gran cantidad de lenguajes(C++, Java, PHP, etc.). Posee gran portabilidad entre sistemas y soporta hasta 32 índices por tabla (35). Este se puede instalar en una serie de sistemas operativos, también es uno de los gestores con mayor rendimiento producto a la velocidad de realizar las operaciones, es de código abierto y funciona en ordenadores de bajos recursos sin ningún problema.

1.12 Propuesta y selección de las herramientas

Para desarrollar un sistema que se ajuste a las necesidades identificadas en el Departamento de Tecnología de la facultad 4 se selecciona la metodología ágil XP ya que se ajusta a las características del proyecto a realizar y del equipo de trabajo. Para la implementación del sistema se utilizará el lenguaje de programación C++, javascript, css, html por el lado de cliente y php, javascript por el lado del servidor. Para el almacenamiento de la información del sistema se utilizará el gestor de base de datos MySQL que constituye una opción fiable que se integra con los lenguajes y marcos de trabajo seleccionados.

1.13 Conclusiones del capítulo

Con la realización de este capítulo se pudo observar que el control de activos de hardware es una actividad importante para la protección de los medios físicos de las organizaciones y que existen un conjunto de normas, estándares y regulaciones que lo rigen.

Se pudo determinar además que existen sistemas que automatizan este control, pero al evaluar sus funcionalidades en base a las necesidades a cubrir con esta investigación, estas presentan carencias importantes, no obstante se pudo detectar un conjunto de características que pudieran ser reutilizadas en el desarrollo de una solución propia.

Con el objetivo de realizar el desarrollo de una propuesta de solución al problema planteado se estudiaron un conjunto de metodologías, lenguajes, herramientas y tecnologías, concluyendo con la selección de las más indicadas según las necesidades y características del equipo de trabajo.

Capítulo 2 Propuesta de solución

2.1 Introducción

Actualmente la información del Departamento de Tecnología de la facultad 4 es almacenada y distribuida en formato digital y se lleva de forma manual, lo que puede provocar la pérdida de información. Hasta el momento no se cuenta con un sistema que permita informatizar los reportes que se llevan a cabo sobre la gestión de hardware por cada computadora de los laboratorios, aulas, salones y oficinas de la facultad, permitiendo que los administradores del sistema estén constantemente actualizados de cualquier incidencia ocurrida. Por tal motivo se decidió desarrollar un sistema basándose en las fases de la metodología XP.

En este capítulo se describirá el flujo actual para el control de activos informáticos de la facultad 4. Se hará referencia a los procesos informatizados, se especificará la lista de reserva del producto, los aspectos no funcionales del mismo y se detallarán las Historias de usuario (HU) para establecer luego el orden en que serán implementadas atendiendo a su prioridad. Además, se detallarán los aspectos importantes tenidos en cuenta para la implementación de la solución propuesta.

El sistema a desarrollar llamado UCInventory, está dividido en tres subsistemas, (UCInventoryManager); una aplicación que tendrá la funcionalidad de generar un reporte por computadora, donde el mismo sería enviado al servidor (UCInventoryServer); este se encargará de capturar el reporte, buscar incidencias en las computadoras reportadas y enviar notificaciones en tiempo real por correo electrónico, de esta forma el administrador gestionaría los datos a través de una interfaz web (UCInventoryWebInterface).

2.1.1 Objetos a informatizar

Para llevar a cabo la gestión de información del Departamento de Tecnología de la facultad 4, existen una serie de procesos que deben ser informatizados, estos son:

- 1-El expediente técnico: planilla orientada por la Vicerrectoría de Tecnología de la UCI para el control de activos informáticos.
- 2-La gestión de área: la información de hardware se encuentra dividida en los locales dentro del área de la facultad 4.

3-Asignar área a computadoras: cada computadora de la facultad 4 debe encontrarse asignada a un local.

2.1.2 Propuesta de solución

Debido a los problemas mencionados anteriormente sobre el trabajo manual que se lleva a cabo sobre los datos por cada computadora, se decidió realizar un sistema que estará dividido en tres subsistemas los cuales son:

- ❖ Interfaz web (UCInventoryWebInterface) desarrollada con el framework Symfony que permite al sistema concentrar toda la información referente al proceso de inventario por cada ordenador en la aplicación, brindando la posibilidad de gestionar áreas, donde se podrán crear, ver, editar y eliminar. El sistema contará con una gestión de computadoras, donde se podrá adicionar una computadora en estado rota manualmente, observar los datos pertenecientes a las mismas, gestionar su expediente técnico así como su historial de modificaciones, tendrá la opción de imprimir este expediente y asignarle un área.
- ❖ Servidor (UCInventoryServer) desarrollado con Nodejs que se encargará de capturar todos los reportes de inventario enviados por las computadoras clientes, distribuidas en aulas, oficinas, salones y laboratorios. Buscará incidencias en los reportes de cada computadora comparándolos con los equivalentes que existen en la base de datos, almacenará el contenido del mismo y enviará notificaciones por correo electrónico en caso de haber ocurrido una incidencia perteneciente a un ordenador.
- ❖ Cliente (UCInventoryManager) desarrollado con Qt-creator utilizando el lenguaje de programación C++, el mismo estará instalado en las computadoras que se encuentran distribuidas en los locales, se encargará de generar un reporte de inventario enviándolo hacia el servidor, tendrá un archivo de configuración permitiéndole al administrador modificar la dirección del servidor, el puerto y el intervalo de tiempo de envío del reporte.

2.2 Personal relacionado con el sistema

Uno de los puntos fundamentales que se debe tener en cuenta al desarrollar un sistema es la especificación de la audiencia a la que está dirigido el mismo, delimitando el nivel de interacción entre los diferentes usuarios. Los roles y las funciones que realizarán estos usuarios en el sistema mostrados en la Tabla 3. El sistema se diseñará para la gestión automatizada del control de activos de hardware y del expediente técnico por parte del Departamento de

Tecnología de la facultad 4, donde tanto el especialista de hardware, el asesor de seguridad informática y el jefe de departamento serán los encargados de interactuar con el mismo.

Tabla 3: Personal relacionado con el sistema

Personal relacionado con el sistema	Descripción
Administrador	Además de gestionar la información del sistema es el encargado de administrar todas las computadoras de la facultad 4, darle soporte y mantenimiento a la aplicación.

2.3 Lista de reserva del producto

Para darle cumplimiento a los objetivos propuestos, el sistema debe ser capaz de cumplir con los siguientes aspectos funcionales, los cuales describen los servicios o funcionalidades que este brindará:

- 1- Autenticar Usuario
- 2- Crear área
- 3- Eliminar área
- 4- Editar área
- 5- Listar área
- 6- Asignar computadoras a un área.
- 7- Generar expediente técnico
- 8- Adicionar datos al expediente técnico
- 9- Imprimir expediente técnico
- 10- Ver historial de modificaciones
- 11- Imprimir historial de modificaciones
- 12- Editar Historial de modificaciones
- 13- Reportar como rota una computadora
- 14- Eliminar computadoras rotas
- 15- Adicionar computadora rota manualmente
- 16- Aceptar cambio en historial de modificaciones
- 17- Eliminar historial de modificaciones
- 18- Ver historial de fecha
- 19- Eliminar historial de fecha

20- Buscar incidencias en la computadora

21- Enviar notificación de correo electrónico

2.4 Aspectos no funcionales del sistema

Los aspectos no funcionales se refieren directamente a las propiedades emergentes del sistema, es decir, a las características que este debe tener.

Restricciones de software

- ❖ Usar el lenguaje de programación PHP versión 5.4.12 o superior.
- ❖ Usar el gestor de Base de Datos MySQL versión 5.6.12 o superior.
- ❖ Utilizar como servidor web apache para la interfaz web.
- ❖ Utilizar como servidor de gestión de reportes Nodejs.
- ❖ Desarrollar bajo el framework Symfony 2.3.5, la interfaz web.
- ❖ Desarrollar el cliente bajo el IDE de desarrollo Qt Creator utilizando como lenguaje de programación C++.
- ❖ Desarrollar el servidor de gestión de reportes bajo la tecnología Nodejs utilizando como lenguaje de programación JavaScript.

Interfaz

El diseño de interfaz visual debe ser intuitivo para los usuarios que interactúan con la aplicación, permitir el fácil entendimiento de las funcionalidades que el mismo brinda, además de poseer colores adecuados para una mejor interacción entre el usuario y la aplicación.

Software

- ❖ Computadoras clientes con sistema operativo Linux.
- ❖ Usar un navegador web como: Mozilla Firefox (versión 27.0.1 o superior).

Hardware

- ❖ PC Pentium 4 o superior.
- ❖ Para el servidor 2GB de memoria RAM o superior.
- ❖ Para el cliente 512 MB de memoria RAM o superior.

Rendimiento

- ❖ Rapidez y eficiencia tanto en los tiempos de respuesta como en la velocidad de procesamiento.

Seguridad

- ❖ **Confidencialidad:** sólo podrá acceder al sistema, aquella persona que se autentique en el mismo.
- ❖ **Integridad:** sólo podrá acceder a la información del sistema el administrador encargado.
- ❖ **Disponibilidad:** el sistema debe funcionar permanentemente para que de esta forma el administrador tenga constancia de lo que sucede en ese instante y en horas posteriores.

2.5 Fase de Exploración

Esta fase define el alcance general del proyecto. Los programadores estiman el tiempo de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias y podrían variar cuando se analicen más en detalle en cada iteración (34). Durante esta fase el trabajo estuvo encaminado a determinar las historias de usuario necesarias para desarrollar un sistema que diera solución al problema planteado.

2.5.1 Historia de Usuario

Las Historias de usuario (HU) son las técnicas utilizadas por la metodología de desarrollo XP para especificar los aspectos funcionales del sistema. Estas son lo suficientemente comprensibles y aclaradas para que los programadores puedan desarrollarlas en apenas unas semanas, los autores de estas HU proponen una planilla que contienen los siguientes datos:

Nombre: nombre de la HU.

Número: número de la HU.

Usuario: el usuario del sistema o el protagonista de la HU.

Iteración: es a la iteración que va a pertenecer la HU.

Prioridad: es la importancia que tiene para el cliente o para el grupo de desarrollo (baja, media, alta).

Complejidad: es el nivel de dificultad que establece el desarrollador a la hora de implementar (baja, media, alta).

Punto estimado: es la estimación definida por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias.

Descripción: es un pequeño resumen de la HU donde se describen las acciones del mismo.

Observación: es la nota de interés.

Tabla 4: HU-Authenticar usuario

Historia de Usuario	
Nombre: Autenticar usuario	Número: 1
Usuario: Administrador	
Prioridad: Alta	Iteración: 1
Complejidad: Alta	Punto Estimado: 1
Descripción: Para poder acceder al sistema se necesita poder autenticarse. Para ello el usuario debe proveer su nombre y su contraseña.	
Observación: Sólo se van a poder autenticar los administradores.	

Tabla 5: HU-Crear área

Historia de Usuario	
Nombre: Crear área	Número: 2
Usuario: Administrador	
Prioridad: Alta	Iteración: 1
Complejidad: Media	Punto Estimado: 0.5
Descripción: El administrador para crear un área debe de llenar los campos siguientes: área de responsabilidad, cantidad de computadoras con que va a contar, el local y el responsable. Luego se presiona el botón crear y se creará un área.	
Observación: Se adiciona una nueva área al sistema.	

Tabla 6: HU-Eliminar área

Historia de Usuario	
Nombre: Eliminar área	Número: 3
Usuario: Administrador	
Prioridad: Media	Iteración: 1
Complejidad: Baja	Punto Estimado: 0.5
Descripción: El administrador podrá eliminar un área cuando esta ya no se encuentre disponible para el sistema, para poder eliminar un área se encuentra dentro de sus opciones un botón (Eliminar área) al ser presionado este visualizará una confirmación, al aceptar el área será eliminada.	
Observación: Se elimina el área completamente del sistema.	

Tabla 7: HU-Editar área

Historia de Usuario	
Nombre: Editar área	Número: 4
Usuario: Administrador	
Prioridad: Media	Iteración: 1
Complejidad: Media	Punto Estimado: 0.5
Descripción: Para editar un área se encuentra dentro de las opciones un botón (Editar área) al presionar este botón se visualizarán los datos anteriores de los cuales está compuesta el área, brindando la opción de editarlos.	
Observación: Se pueden modificar los campos de cada área.	

Tabla 8: HU-Listar área

Historia de Usuario	
Nombre: Listar área	Número: 5
Usuario: Administrador	
Prioridad: Baja	Iteración: 1

Complejidad: Media	Punto Estimado: 0.5
Descripción: Permite que el administrador tenga en cuenta la cantidad de locales pertenecientes al área que hay en el sistema. Además de tener como funcionalidades el editar esta área y eliminarla.	
Observación: Se mostrarán las áreas creadas en el sistema.	

Tabla 9: HU-Asignar área a una computadora

Historia de Usuario	
Nombre: Asignar área a computadoras	Número: 6
Usuario: Administrador	
Prioridad: Media	Iteración: 2
Complejidad: Media	Punto Estimado: 1
Descripción: Cada computadora sea de estado nueva, asignada o rota se le asignará un local, dentro de las opciones de computadoras se encuentra el botón (Asignar área), cuando es presionado se le muestra al usuario un listado de todos los locales creados en el sistema.	
Observación: El campo área de las computadoras contendrá el nombre del área asignada.	

Tabla 10: HU-Generar expediente técnico

Historia de Usuario	
Nombre: Generar expediente técnico	Número: 7
Usuario: Administrador	
Prioridad: Alta	Iteración: 2
Complejidad: Alta	Punto Estimado: 0.5
Descripción: Para cada computadora sea de estado nueva, asignada o rota se le brinda la opción de generar o visualizar su expediente técnico, en este se pueden visualizar todos los datos referente a esa computadora ya sea de hardware o periféricos, brindando la opción de llenar algunos campos de forma manual.	
Observación: Se muestra una interfaz que contiene una planilla con un botón adicionar datos, ver historial de modificaciones e imprimir.	

Tabla 11: HU-Adicionar datos al expediente técnico

Historia de Usuario	
Nombre: Adicionar datos al expediente técnico	Número: 8
Usuario: Administrador	
Prioridad: Alta	Iteración: 2
Complejidad: Media	Punto Estimado: 0.5
Descripción: Una vez visualizado el expediente técnico, este tiene algunos campos vacíos los cuales deben ser llenado manualmente por el usuario, una vez completado todos los campos se presiona el botón (Adicionar datos) y estos datos pasan a formar parte del expediente de esa computadora.	
Observación: Se mostrara el expediente técnico con todos los datos llenos.	

Tabla 12: HU-Imprimir expediente técnico

Historia de Usuario	
Nombre: Imprimir expediente técnico	Número: 9
Usuario: Administrador	
Prioridad: Media	Iteración: 2
Complejidad: Media	Punto Estimado: 0.5
Descripción: Dentro de las opciones de expediente técnico se encuentra un botón (Imprimir), el cual mostrará un expediente listo para imprimir.	
Observación: Se podrá imprimir el expediente que es el encargado de recopilar toda la información que posee una computadora.	

Tabla 13: HU-Ver historial de modificaciones

Historia de Usuario	
Nombre: Ver historial de modificaciones	Número: 10
Usuario: Administrador	

Prioridad: Baja	Iteración: 2
Complejidad: Media	Punto Estimado: 0.5
Descripción: Esta funcionalidad forma parte del expediente técnico referente a una computadora, cada expediente tiene la opción de (Ver historial de modificaciones), esta muestra todos los cambios que se le ha presentado a la computadora analizada en ese momento.	
Observación: Se puede observar todas las modificaciones que se le han realizado a las computadoras ordenadas por fecha.	

Tabla 14: HU-Imprimir historial de modificaciones

Historia de Usuario	
Nombre: Imprimir historial de modificaciones	Número: 11
Usuario: Administrador	
Prioridad: Media	Iteración: 3
Complejidad: Media	Punto Estimado: 0.5
Descripción: Dentro de las opciones de historial de modificaciones se encuentra un botón (Imprimir), el cual mostrará un historial listo para imprimir.	
Observación: Se genera la copia de impresión del historial.	

Tabla 15: HU-Editar historial de modificaciones

Historia de Usuario	
Nombre: Editar historial de modificaciones	Número: 12
Usuario: Administrador	
Prioridad: Alta	Iteración: 3
Complejidad: Media	Punto Estimado: 0.5
Descripción: Para editar un historial de modificaciones, es necesario que la computadora que se está observando en ese momento contenga alguna incidencia o alguna modificación, brindando dentro de sus opciones la opción (Editar datos), en esta opción se podrán editar los datos de las modificaciones o incidencias que ha sufrido la computadora, como su número de orden gat y su nuevo número de sello.	

Observación: Se edita los cambios según el tipo de incidencia ocurrido.
--

Tabla 16: HU-Reportar como rota una computadora

Historia de Usuario	
Nombre: Reportar como rota una computadora	Número: 13
Usuario: Administrador	
Prioridad: Media	Iteración: 3
Complejidad: Baja	Punto Estimado: 0.5
Descripción: Cuando una computadora por cualquier motivo deja de funcionar por lo que no se reportará al sistema, se brindará la opción de reportar como rota, la misma se encuentra dentro de las opciones de computadoras.	
Observación: La computadora pasa a estado de rota.	

Tabla 17: HU-Eliminar computadora rota

Historia de Usuario	
Nombre: Eliminar computadora rota	Número: 14
Usuario: Administrador	
Prioridad: Alta	Iteración: 3
Complejidad: Alta	Punto Estimado: 0.5
Descripción: Se elimina aquella computadora que se encuentre en estado roto y se le dará baja del sistema.	
Observación: No se encontrará más en ningún área de la facultad.	

Tabla 18: HU-Adicionar computadora manualmente

Historia de Usuario	
Nombre: Adicionar computadora manualmente	Número: 15
Usuario: Administrador	
Prioridad: Media	Iteración: 3

Complejidad: Media	Punto Estimado: 0.5
Descripción: Esta funcionalidad es para adicionar al sistema una computadora rota, que no pueda reportarse pero pertenece a un área.	
Observación: Se agrega una nueva computadora en estado de rota.	

Tabla 19: HU-Aceptar cambios en historial de modificaciones

Historia de Usuario	
Nombre: Aceptar cambios en historial de modificaciones	Número: 16
Usuario: Administrador	
Prioridad: Alta	Iteración: 3
Complejidad: Media	Punto Estimado: 0.5
Descripción: Esta funcionalidad es la encargada de aceptar los cambios presentados en el historial de modificaciones de la computadora por el usuario, una vez que ya los cambios sean vistos y analizados, brindando la opción de aceptarlos.	
Observación: Se aceptan los cambios y dejan de ser editables.	

Tabla 20: HU-Eliminar historial de modificaciones

Historia de Usuario	
Nombre: Eliminar historial de modificaciones	Número: 17
Usuario: Administrador	
Prioridad: Media	Iteración: 3
Complejidad: Media	Punto Estimado: 0.5
Descripción: Se eliminarán los historiales que coincidan en incidencias que pertenezcan a una misma máquina. Esta opción aparecerá en su historial de modificaciones.	
Observación: La existencia de reportes de una máquina con la misma incidencia.	

Tabla 21: HU-Ver historial de fecha

Historia de Usuario	
Nombre: Ver historial de fecha	Número: 18

Usuario: Administrador	
Prioridad: Media	Iteración: 3
Complejidad: Media	Punto Estimado: 0.5
Descripción: Cada computadora tiene la opción de ver su historial de fechas, en esta se visualizan todas las veces que esta computadora se ha reportado al sistema llevando una constancia de las veces que se reportó en el día.	
Observación: Se observan las fechas en que esa computadora se reportó.	

Tabla 22: HU-Eliminar historial de fecha

Historia de Usuario	
Nombre: Eliminar historial de fecha	Número: 19
Usuario: Administrador	
Prioridad: Media	Iteración: 3
Complejidad: Media	Punto Estimado: 0.5
Descripción: Cada computadora tiene la opción de eliminar su historial de fechas, en caso de que la computadora se haya reportado en el día sin incidencias.	
Observación: Se podrá eliminar el historial de fechas por computadora.	

Tabla 23: HU-Buscar incidencia en la computadora

Historia de Usuario	
Nombre: Buscar incidencia en la computadora	Número: 20
Usuario: Sistema	
Prioridad: Alta	Iteración: 3
Complejidad: Alta	Punto Estimado: 1
Descripción: Esta funcionalidad forma parte del servidor nodejs, cuando una computadora es reportada, se comparan los datos viejos de su reporte con los nuevos en busca de incidencia.	
Observación: Se buscará incidencias en la computadora.	

Tabla 24: HU-Enviar notificación de correo electrónico

Historia de Usuario	
Nombre: Enviar notificación de correo electrónico	Número: 21
Usuario: Sistema	
Prioridad: Alta	Iteración: 3
Complejidad: Alta	Punto Estimado: 0.5
Descripción: Cuando una computadora es reportada con incidencias, el sistema le informará o notificará mediante correos electrónicos al administrador, la incidencia ocurrida.	
Observación: Se envía notificaciones a correos electrónicos.	

2.6 Fase de planificación

La planificación es una fase donde se entrevistan el cliente con el gerente y el grupo de desarrolladores para definir el orden en que se desarrollan las historias de usuario, se toma como acuerdo el tiempo para cada una de las HU. El resultado de esta fase es un plan de entrega que no puede excederse de los tres meses.

2.6.1 Estimación del esfuerzo por HU

Para la creación del sistema propuesto, se realizó una estimación para cada HU identificada, a continuación (Tabla 25) se muestran los resultados obtenidos.

Tabla 25: Estimación de esfuerzo por HU

Número	Historia de usuario	Puntos de estimación
1	Autenticar usuario	1
2	Crear área	0.5
3	Eliminar área	0.5
4	Editar área	0.5
5	Listar área	0.5
6	Asignar área a computadoras	1
7	Generar expediente técnico	0.5

8	Adicionar datos al expediente técnico	0.5
9	Imprimir expediente técnico	0.5
10	Ver historial de modificaciones	0.5
11	Imprimir historial de modificaciones	0.5
12	Editar Historial de modificaciones	0.5
13	Reportar como rota una computadora	0.5
14	Eliminar computadoras rotas	0.5
15	Adicionar computadora manualmente	0.5
16	Aceptar cambio en historial de modificaciones	0.5
17	Eliminar historial de modificaciones	0.5
18	Ver historial de fecha	0.5
19	Eliminar historial de fecha	0.5
20	Buscar incidencias en la computadora	1
21	Enviar notificación de correo electrónico	0.5

2.6.2 Plan de iteraciones

La metodología XP define que los proyectos constan de tres etapas en adelante, las cuales toman el nombre de iteraciones, denominándose metodología iterativa. Cada una de las iteraciones contiene un conjunto de HU a desarrollar, estas cuentan con un tiempo estimado de una a tres semanas, se obtiene así una versión funcional del producto. Teniendo en cuenta la lista de reserva del producto para el proyecto y el tiempo de duración para el desarrollo del sistema, donde se determinó dividirlo en tres iteraciones, las cuales son:

Tabla 26: Plan de iteraciones

Iteraciones	Plan de iteraciones
Iteración 1	En la primera iteración se implementarán las funcionalidades básicas del sistema, teniendo presente las HU 1, 2, 3, 4 y 5. Con la culminación de esta iteración se obtiene la primera versión del producto donde el administrador podrá autenticarse y gestionar áreas.

Iteración 2	En esta fase se implementarán las HU asignar área a computadoras, generar expediente técnico, adicionar datos al expediente técnico, imprimir expediente técnico y ver historial de modificaciones , además de corregir cualquier error e inconformidad detectada en las historias de usuario de la primera iteración y se realizarían las pruebas de software a esta segunda iteración. También se le hace una entrega de estas dos iteraciones al cliente a ver si está conforme con el trabajo realizado.
Iteración 3	En esta tercera iteración se desarrollan las últimas HU del sistema, como imprimir historial de modificaciones, editar historial de modificaciones, reportar como rota una computadora, eliminar computadoras rotas, adicionar computadora manualmente, aceptar cambio en historial de modificaciones, eliminar historial de modificaciones, ver, eliminar historial de fecha, buscar incidencias, enviar notificación por correo. Estas HU serán desarrolladas para hacer una entrega final del producto de tal forma que el cliente quede conforme con el trabajo realizado.

2.6.3 Plan de duración de las iteraciones

El plan de duración de las iteraciones según XP es realizado para conocer el tiempo que se demora cada iteración del proyecto, las historias de usuario que serán implementadas en estas iteraciones. Este define el orden de las HU por iteraciones y el tiempo que demora cada una, permite conocer la duración que tarda el sistema en desarrollarse.

Tabla 27: Plan de duración de las iteraciones

Iteraciones	Orden de las HU	Duración
Iteración 1	<ol style="list-style-type: none"> 1. Autenticar usuario 2. Crear área 3. Eliminar área 4. Editar área 5. Listar área 	3 semanas
	<ol style="list-style-type: none"> 1. Asignar área a computadoras 2. Generar expediente técnico 	

Iteración 2	<ul style="list-style-type: none"> 3. Adicionar datos al expediente técnico 4. Imprimir expediente técnico 5. Ver historial de modificaciones 	3 semanas
Iteración 3	<ul style="list-style-type: none"> 1. Imprimir historial de modificaciones 2. Editar historial de modificaciones 3. Reportar como rota una computadora 4. Eliminar computadoras rotas 5. Adicionar computadora manualmente 6. Aceptar cambio en historial de modificaciones 7. Eliminar historial de modificaciones 8. Ver historial de fecha 9. Eliminar historial de fecha 10. Buscar incidencias en la computadora 11. Enviar notificación de correo electrónico 	6 semanas

2.6.4 Plan de entrega

El principal propósito del plan de entrega es agrupar una serie de HU para conformar una entrega de las mismas, así como su orden por iteraciones. Este cronograma se realiza en función de las estimaciones de tiempo que se debe consumir en la realización del proyecto.

Tabla 28: Plan de entrega

Historia de usuario	Primera iteración	Segunda iteración	Tercera iteración
V 0.1	25-01-2014		
V 0.5		28-02-2014	
V 1.0			01-04-2014

2.7 Patrones de diseño

Es una solución de problemas que ocurren durante el desarrollo del software, estos ayudan a entender las soluciones del problema con un término igual, lo que permite un mejor entendimiento.

Los patrones de diseño se dividen en dos grandes grupos los patrones generales de software para asignar responsabilidades (GRASP²) y los patrones que pertenecen a la llamada pandilla de los cuatro (GOF³). A continuación se explica cómo fueron utilizados en el sistema.

2.7.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (35). De estos patrones se destacan para el diseño del sistema los siguientes:

Experto: Una clase se considera experta si contiene toda la información necesaria para realizar la labor de lo que se le encomendó. Además, para el cumplimiento de esta responsabilidad requiere a menudo información distribuida en varias clases, un ejemplo es la clase `computadoraController.php`, esta clase tiene la responsabilidad de trabajar con todos los datos de una computadora, es el centro de todas las otras clases (36). Ella controla todo el flujo de información almacenado en otras clases como la clase `areas.php`, ya que una computadora consta de la asignación de una.

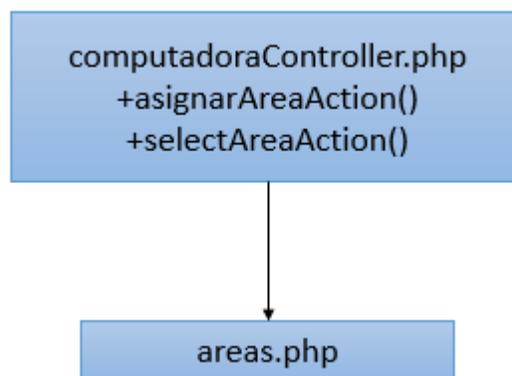


Figura 4: Relación de clases utilizando el patrón de diseño Experto

² Del inglés General Responsibility Assignment Software Patterns

³ Del inglés Gang of Four. Agrupa a los patrones Object Pool, Abstract Factory, Builder y Factory Method.

Una computadora también tiene como información un historial de modificaciones, estos datos aparecen en la clase `historialModificaciones.php` y se gestionan los mismos a través de la clase `computadoraController.php`.

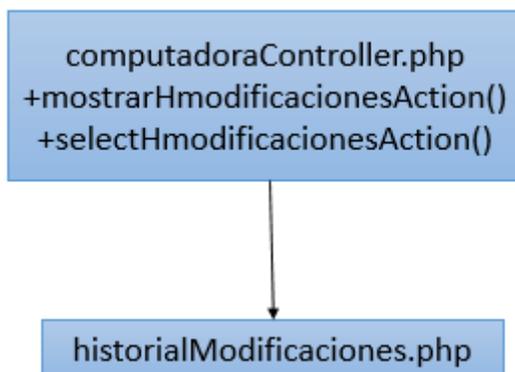


Figura 5: Relación de clases utilizando el patrón de diseño Experto

Creador: Se utiliza cuando se quiere, a partir de una clase con alta jerarquía, obtener clases descendientes o instancias a partir de clases obtenidas. Este patrón tiene dos tipos de clases, las abstractas y las concretas, define dos tipos de estructuras dentro de la jerarquía, estructura AKO(A King of) y APO(A Punt of) (37). La estructura APO se ve vigente en las clases `hardware.php` y `historialModificaciones.php`, ya que estas clases son una parte de `computadora.php`.

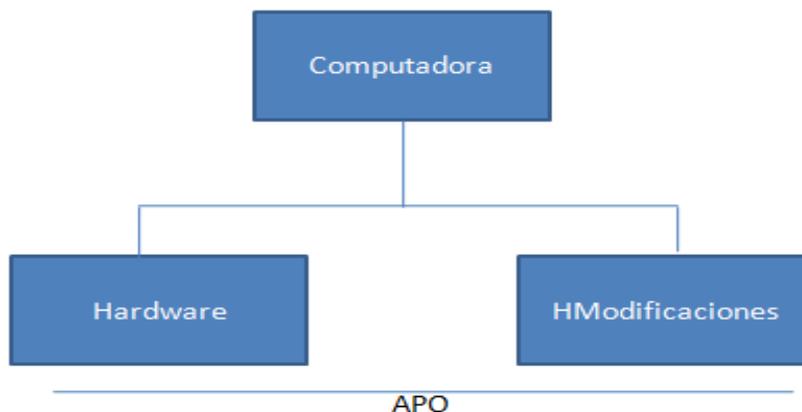


Figura 6: Relación de clases utilizando el patrón de diseño Creador

2.7.2 Patrones GOF

Estos patrones son aquellos que describen la comunicación entre objetos y clases adaptada a la resolución de problemas generales de diseño en un contexto particular. Se centran en una solución a un problema concreto en el ámbito de la programación orientada a objetos, se

propone una solución genérica de clases y relaciones para resolver dicho problema. El framework Symfony utiliza una serie de patrones GOF como son:

En la categoría Estructurales:

Decorator (Envoltorio): Añade funcionalidad a una clase, dinámicamente. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, evita su repetición en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación de este patrón de diseño (38). El mismo está vigente en el directorio app\Resources\views.(Anexo 5: figura 26)

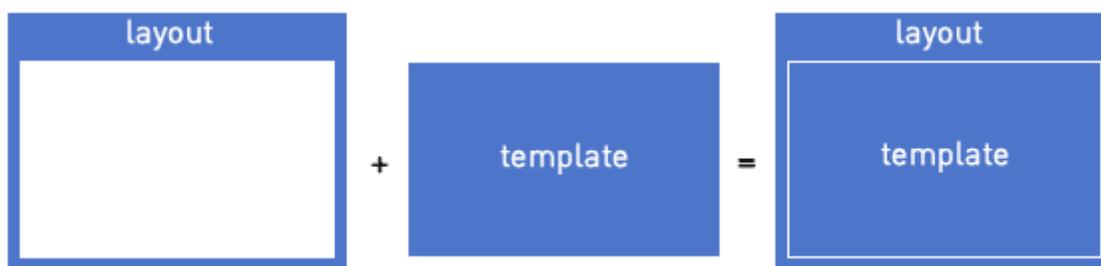


Figura 7: Estructura que utiliza el patrón de diseño Decorator

2.8 Estándar de implementación

El estándar empleado en el desarrollo de la aplicación es el CamelCase, este consiste en la práctica de escribir frases o palabras compuestas, se eliminan los espacios y se escribe en mayúscula o minúscula la primera letra de cada palabra. Existen dos tipos de CamelCase, el UpperCamelCase que es cuando la primera letra de todas está escrita en mayúscula y el lowerCamelCase, cuando la primera letra de todas está escrita en minúscula, de estos dos se utiliza el segundo respectivamente, un ejemplo de su uso se muestra en la imagen a continuación (39):

```
public function computadoraRotaAction($id) {  
  
public function listadoComputadorasRotasAction() {  
  
public function listadoComputadorasRotasAreasAction($valor) {  
  
public function aceptarCambioAction($id) {
```

Figura 8: Instancias de clase que utiliza el estándar de implementación CamelCase

2.9 Tarjetas CRC

Las tarjetas CRC son clases bien estructuradas con responsabilidades bien definidas. Estas son simples, adaptables y tienen características que facilitan la interacción entre el cliente y el equipo de desarrollo. Las tarjetas CRC generadas para el diseño de la solución propuesta se muestran a continuación.

Tabla 29: Tarjeta CRC Áreas

Áreas	
Responsabilidades	Colaboraciones
Crear área Listar área Editar área Eliminar área	computadoraController

Tabla 30: Tarjeta CRC computadoraController

computadoraController	
Responsabilidades	Colaboraciones
Asignar área a computadoras Generar expediente Adicionar datos a expediente técnico Imprimir expediente técnico Ver historial de modificaciones Ver historial de fecha Eliminar historial de fecha Imprimir historial de modificaciones Reportar como rota una computadora Eliminar computadora rota Adicionar computadora manualmente	historialModificaciones historialFecha hardware

Tabla 31: Tarjeta CRC historialModificaciones

historialModificaciones	
Responsabilidades	Colaboraciones
Editar historial de modificaciones	

Tabla 32: Tarjeta CRC historialFecha

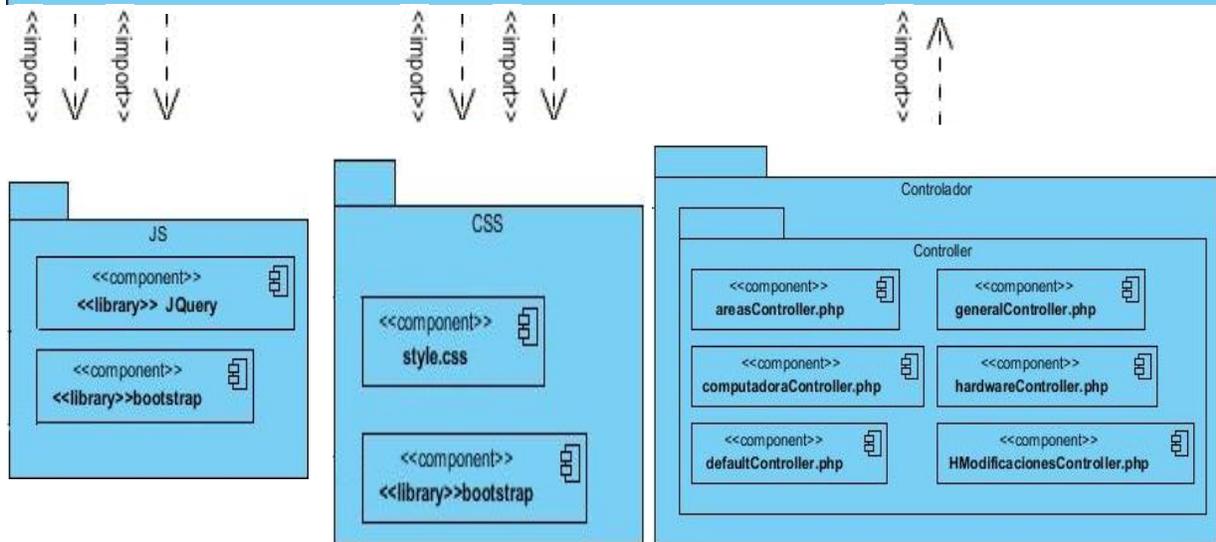
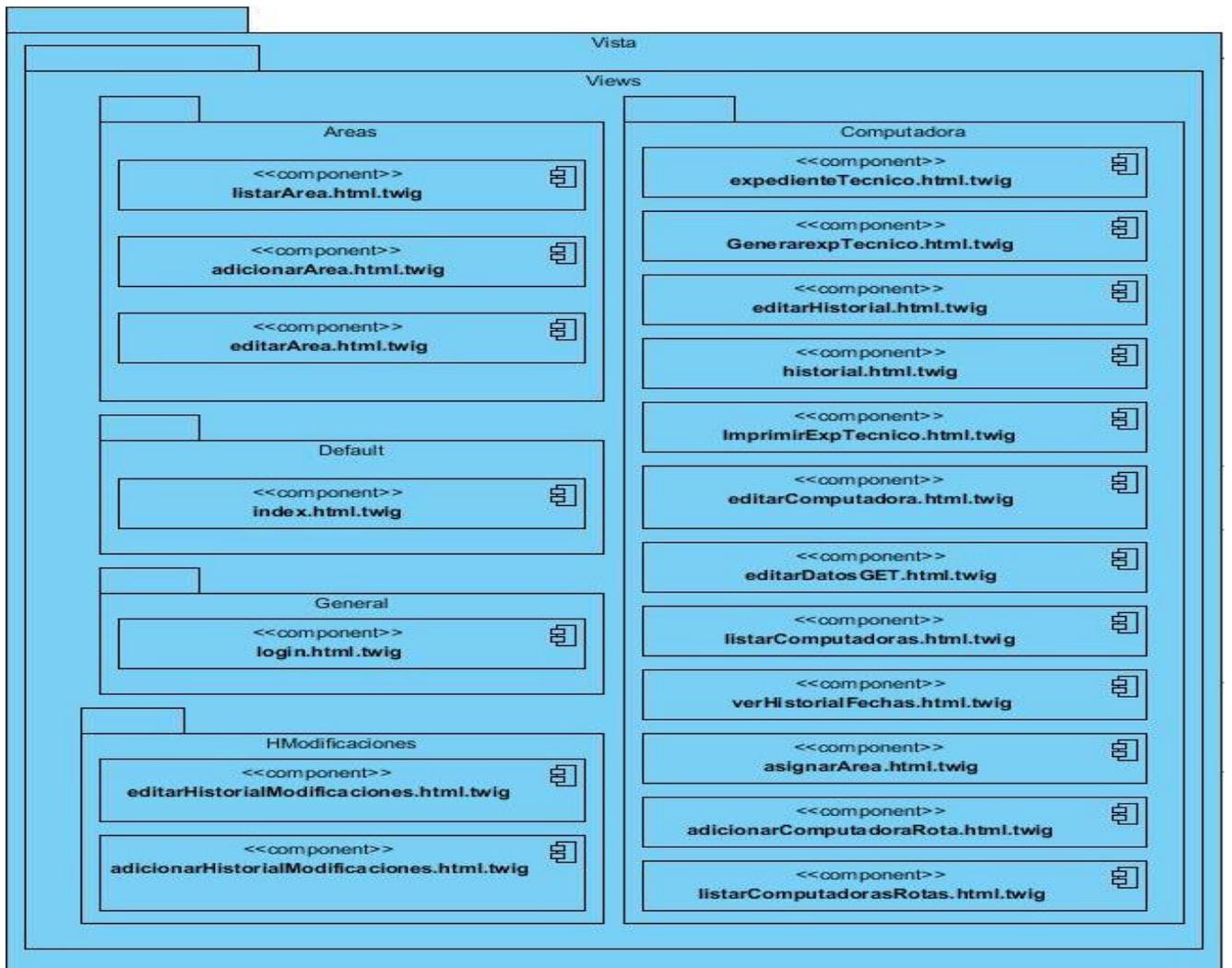
historialFecha	
Responsabilidades	Colaboradores
Ver historial de fecha	

Tabla 33: Tarjeta CRC SecurityContext

SecurityContext	
Responsabilidades	Colaboradores
Autenticar Usuario	

2.10 Diagrama de componentes del subsistema UCInventoryWebInterface

El diagrama de componentes muestra la relación entre componentes de software, sus dependencias, su comunicación, su ubicación y otras condiciones. En el diagrama a continuación se visualiza el diagrama de componentes del sistema de interfaz web a crear, su estructura y organización.



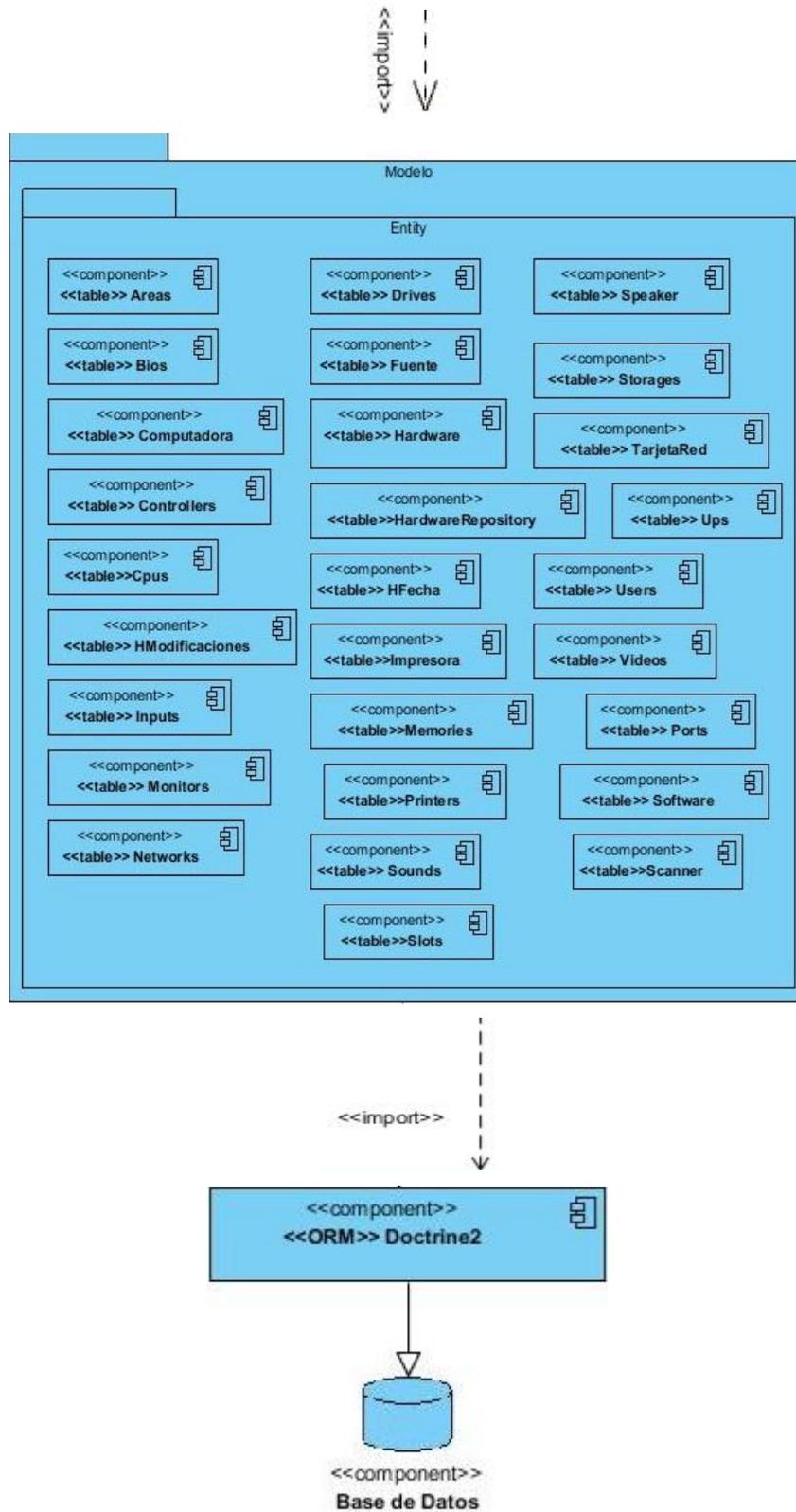


Figura 9: Diagrama de componente del subsistema UCInventoryWebInterface

2.11 Diagrama de despliegue

El modelo de despliegue provee un modelo detallado de la forma en la que los componentes se despliegan a lo largo de la infraestructura del sistema. Es un mapa específico de la instalación física del sistema. Un diagrama de despliegue ilustra el despliegue físico del sistema en un ambiente de producción (o prueba). Muestra dónde se ubicarán los componentes, en qué servidores, máquinas o hardware. Puede ilustrar vínculos de red, ancho de banda de LAN, etc. En el diagrama a continuación se muestra como queda distribuido el sistema a crear.

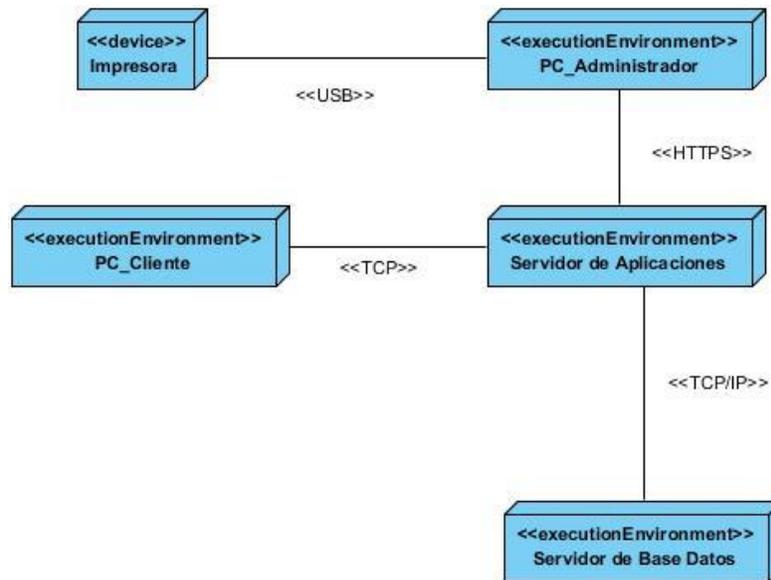


Figura 10: Diagrama de despliegue

PC cliente: este nodo representa a las computadoras de las áreas que reportarán al sistema, un informe con sus datos, de hardware, software y periféricos.

Servidores de aplicaciones: representa varias estaciones donde estarán montados los servidores Apache y Nodejs sobre el cual estará ejecutándose UCInventory.

Servidor de base de datos: representa el servidor donde estará el SGBD MySQL que dará respuesta a las peticiones hechas por la aplicación.

PC administrador: este nodo representa la computadora del administrador encargado de mantener el sistema.

Impresora: este nodo representa al periférico impresora que estará conectado con la computadora del administrador brindándole un servicio de impresión de datos.

2.12 Conclusiones del capítulo

En el presente capítulo se evidencia el desarrollo de un sistema compuesto por 3 aplicaciones que cumplen con los requerimientos identificados durante la fase de exploración propuesta por la metodología XP.

A través de la identificación de las historias de usuarios, el plan de iteraciones y entregas se cumplió con el cronograma planteado.

La elaboración de las tarjetas CRC junto a varios diagramas como el de componente y despliegue ayudaron a un mejor entendimiento de la arquitectura del sistema durante la fase de desarrollo, alcanzándose una productividad elevada del equipo de desarrollo.

Se obtuvo la aplicación cliente encargada de la creación del reporte de hardware, el servidor y la web interface que permitirá a los especialistas del Departamento de Tecnología de la facultad 4 el control de los activos informáticos presentes en el área, así como la generación de los expedientes técnicos de las pc.

Capítulo 3 Pruebas

3.1 Introducción

En este capítulo se presentan los casos de pruebas identificados durante la fase de planificación de la metodología XP y los resultados de la realización de los diferentes tipos de pruebas utilizadas durante el desarrollo del sistema.

3.2 Planificación de pruebas

El proceso de pruebas es uno de los pilares fundamentales de la metodología XP, el cual ayuda al cliente a verificar y concretar las funcionalidades de las HU, por lo que favorece la comunicación entre el cliente y el equipo de desarrollo. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar los efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, se escriben antes que el código y son las encargadas de verificar el mismo de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. En el caso particular de esta investigación, fue necesario incluir la realización de pruebas de integración debido a que el sistema está compuesto por tres subsistemas mencionados anteriormente, aunque funcionan de manera separada necesitan integrarse para la obtención de la información requerida por el cliente.

3.2.1 Plan de pruebas por iteración

Pruebas unitarias

De acuerdo con lo que plantea la metodología XP, las pruebas unitarias o pruebas de unidad consisten en comprobaciones manuales o automatizadas desarrolladas por los programadores. Las cuales se realizan para verificar que el código correspondiente al sistema se comporta de manera esperada. Las pruebas unitarias proporcionan beneficios tales como:

- ✓ Brindan al programador una inmediata retroalimentación de cómo está realizando su trabajo.

- ✓ El programador puede realizar cambios de forma segura respaldada por efectivos casos de prueba.
- ✓ Permite saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo.

Las pruebas unitarias se le aplicaron al sistema en todo momento del desarrollo, con el objetivo de que el funcionamiento del código fuese el correcto.

Pruebas de integración

Con el desarrollo de cada sistema se realizaron pruebas para ver su funcionamiento por separado para confirmar que estos en su totalidad funcionaban sin problemas, por tanto se aplicó a estos sistemas las pruebas de integración, estas verifican directamente el acoplamiento paulatino de los sistemas, este tipo de prueba maneja dos conceptos fundamentales: integración descendente e integración ascendente, durante el desarrollo del sistema fue necesario realizar pruebas de integración descendente, que consisten en verificar directamente la implementación de los sistemas, desde el principal hacia los inferiores de manera jerárquica. El modo de funcionamiento consiste en que el sistema de control principal maneje toda la información de las pruebas y las distribuya directamente a los sistemas que se encuentren inmediatamente subordinados o conectados. Estas pruebas se realizaron al concluir el desarrollo del servidor UCInventoryServer en la tercera iteración con el objetivo de verificar el flujo de información que debe existir entre los subsistemas UCInventoryManager y UCInventoryWebInterface.

Pruebas de aceptación

Las pruebas de aceptación son validaciones que se le realizan al sistema para ver si su funcionamiento está acorde a lo esperado. Utilizan el método de caja negra y se planifican a partir de la HU, pueden tener varios casos de pruebas hasta que quede asegurado la aceptación del sistema desarrollado, las cuales fueron realizadas y aceptadas por el cliente ya que forma parte del equipo de desarrollo.

Un sistema está completamente aceptable cuando quedan satisfechos todos los aspectos funcionales especificados por el cliente dentro de la lista de reserva del producto, teniendo en cuenta además los aspectos no funcionales, así como a los distintos recursos del sistema.

Para la realización de las pruebas se planificaron un total de 21 casos los cuales fueron divididos y aplicados al final de cada iteración según correspondían.

Los campos de un caso de aceptación de prueba son los siguientes:

Código: el código estará dividido en dos partes, la primera (HU [número]) será la historia de usuario a la que se le estará realizando el caso de prueba. La segunda (P [número]) pertenece al número de prueba realizada.

Historia de usuario: nombre de la historia de usuario a la que se le está realizando el caso de prueba.

Nombre: descripción más detallada del nombre de la historia de usuario.

Iteración: iteración a la que pertenece la historia de usuario.

Descripción: breve descripción del propósito de la prueba.

Condiciones de ejecución: condiciones especiales que deben tenerse en cuenta para ejecutar el caso de prueba.

Entrada/pasos de ejecución: entradas o funciones que deben ejecutarse para realizar el caso de prueba.

Resultados esperados: salida u objetivo que debe cumplir la funcionalidad a la que se le realiza el caso de prueba.

Evaluación de las pruebas: evaluación de éxito del caso de prueba. Prueba satisfactoria en caso de éxito o prueba insatisfactoria en caso de fallo.

A continuación, aparecen algunas de las pruebas de aceptación realizadas a la solución propuesta, el resto se puede encontrar en los anexos:

Iteración 1:

Tabla 34: Prueba de aceptación. No 1

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: Autenticar usuario
Nombre: Autenticación para ingresar al sistema.	Iteración: 1
Descripción: Prueba para la funcionalidad de autenticarse, para ingresar al sistema	
Condiciones de ejecución: Conocer el usuario y la contraseña del sistema.	
Entrada/Pasos de ejecución: Conocer la dirección del sistema. Escribir usuario. Escribir contraseña. Presionar el botón (Iniciar sesión).	
Resultados esperados: Los datos insertados son correctos.	
Evaluación de la prueba: Resultado satisfactorio.	

Prototipo de interfaz de usuario: Anexo

Tabla 35: Prueba de aceptación. No 2

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: Crear área
Nombre: Crear un área en el sistema.	Iteración: 1
Descripción: Prueba para la funcionalidad crear área.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Gestión de áreas). Seleccionar la opción (Crear áreas). Llenar los campos que conforman un área. Campo área de responsabilidad. Campo cantidad de computadoras. Campo local. Campo responsable. Presionar el botón (Adicionar).	
Resultados esperados: Se ha creado el área satisfactoriamente.	
Evaluación de la prueba: Resultado satisfactorio.	

Iteración 2:

Tabla 36: Prueba de aceptación. No 6

Caso de Prueba de Aceptación	
Código: HU6_P6	Historia de Usuario: Asignar área a computadoras
Nombre: Asignarle un área a cada computadora reportada en el sistema.	Iteración: 2
Descripción: Se realizará la prueba a la funcionalidad asignar área a una computadora que se haya reportado en el sistema.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Computadora nueva o Computadora asignada). Seleccionar la opción (Asignar área o icono naranja) dentro de las opciones de	

computadoras.
Resultados esperados: Se muestran todas las áreas creadas en el sistema.
Evaluación de la prueba: Resultado satisfactorio.

Tabla 37: Prueba de aceptación. No 7

Caso de Prueba de Aceptación	
Código: HU7_P7	Historia de Usuario: Generar expediente técnico
Nombre: Generar el expediente técnico automatizado de una computadora	Iteración: 2
Descripción: Se realizará la prueba a la funcionalidad generar expediente técnico que el sistema debe llenarlo automáticamente.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Computadora nueva). Seleccionar la opción (Generar Expediente) dentro de las opciones de computadoras. Llenar los campos que no contengan datos.	
Resultados esperados: Se muestran todas las áreas creadas en el sistema.	
Evaluación de la prueba: Resultado satisfactorio.	

Iteración 3:

Tabla 38: Prueba de aceptación. No 20

Caso de Prueba de Aceptación	
Código: HU16_P16	Historia de Usuario: Aceptar cambio en historial de modificaciones
Nombre: Aceptar cambio en historial de modificaciones	Iteración: 3
Descripción: Se realizará la prueba a la funcionalidad aceptar cambio en historial de modificaciones	
Condiciones de ejecución: Estar autenticado en el sistema. Computadora analizada debe tener al menos una incidencia.	
Entrada/Pasos de ejecución:	

Seleccionar la opción (Computadora nueva, asignada o rota).
 Seleccionar la opción (Generar Expediente) dentro de las opciones de computadoras.
 Seleccionar la opción (Ver Historial de Modificaciones).
 Seleccionar la opción (Aceptar cambio) dentro de las opciones de incidencia de computadoras.

Resultados esperados: Cambios aceptados satisfactoriamente.

Evaluación de la prueba: Resultado satisfactorio.

3.2.2 Resumen de resultados de la etapa de pruebas

Luego de realizada todas las pruebas necesarias para el desarrollo del sistema, se encontró un total de 25 no conformidades, entre las que se destacan:

- ❖ Las letras del índex no cumplen con el tamaño definido por el cliente.
- ❖ Dentro de la funcionalidad crear área, el campo nombre contiene “::”.
- ❖ Los campos del expediente técnico, número de serie admiten letras y sólo pueden contener enteros.
- ❖ Cuando se eliminaba una computadora o un área no existía una confirmación de seguridad.
- ❖ Dentro de la funcionalidad gestionar área se encuentra editar área la cual no mostraba los datos anteriores de la misma para que su actualización fuera factible.
- ❖ No existía un eliminar incidencias dentro del historial de incidencia de una computadora.

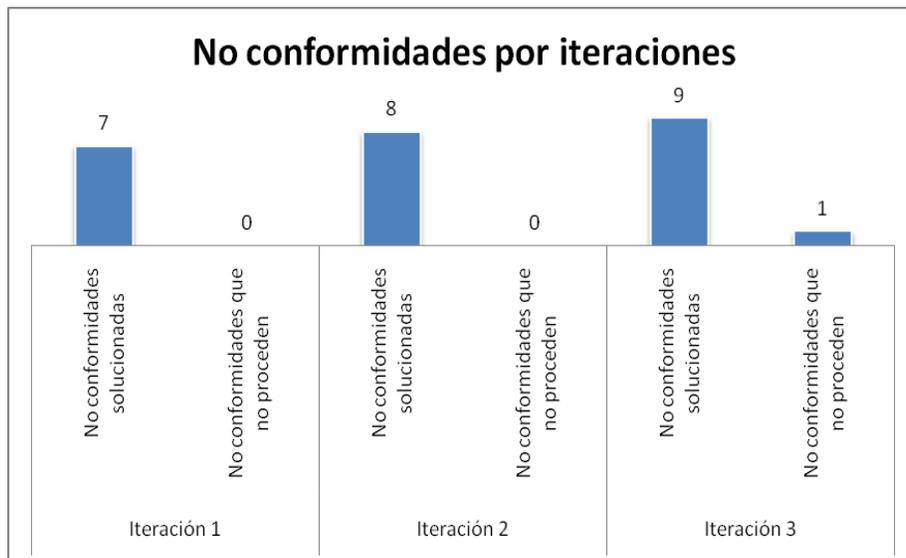


Figura 11: Resultados de las Pruebas de Aceptación en cada Iteración.

A 20 de las no conformidades antes descritas se le realizaron los cambios pertinentes ya que las restantes no proceden con lo cual el cliente quedó satisfecho con el sistema creado.

3.3 Conclusiones

En el presente capítulo se llevó a cabo el desarrollo de las pruebas del sistema propuesto por la metodología XP. Luego de cada iteración se realizaron pruebas de integración, unitarias y de aceptación, las mismas identificaron una lista de no conformidades las cuales fueron resuelta satisfactoriamente quedando así el sistema libre de problemas.

Conclusiones generales

Sobre el desarrollo de esta investigación se puede concluir que:

- ❖ El marco teórico de la investigación posibilitó un mejor entendimiento de los conceptos, regulaciones y estándares asociados al campo de acción de esta investigación.
- ❖ El análisis de soluciones similares permitió entender el nivel de automatización del control de activos informáticos a nivel internacional y nacional, aportando además elementos que fueron integrados a la solución desarrollada en esta investigación.
- ❖ Se logró obtener una aplicación que permite el control de manera automática de los activos informáticos a través de un proceso de desarrollo guiado por una metodología eficaz como lo es XP.
- ❖ El proceso de pruebas ayudó a elevar la calidad del sistema y la detección de errores de manera temprana antes de su despliegue.

Recomendaciones

Una vez concluida la investigación y basándose en las experiencias acumuladas a lo largo del desarrollo del sistema, se proponen las siguientes recomendaciones:

- ❖ Implementar una gestión de usuarios, rol y una vista para que algunas personas que no sean administradores puedan acceder al sistema con los permisos correspondientes.
- ❖ Continuar el ciclo del desarrollo del sistema, creando nuevas funcionalidades que aporten a la automatización del inventario de activos.
- ❖ Hacerlo extensible a otras áreas de la universidad.
- ❖ Generación de reportes personalizados para administrativos de las áreas.

Referencias

1. CSI Computer Crime and Security Survey 2010/2011 | Computer Security Institute. In: [online]. [Accessed 19 febrero 2014]. Available from: <http://gocsi.com/survey>.
2. Kaspersky Security Bulletin. Statistics 2011 - Securelist. In: [online]. [Accessed 20 febrero 2014]. Available from: http://www.securelist.com/en/analysis/204792216/Kaspersky_Security_Bulletin_Statistics_2011.
3. ISO27000.es - El portal de ISO 27001 en español. Gestión de Seguridad de la Información. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.iso27000.es/iso27000.html>.
4. RODRIGO WERLINGER, KIRSTIE HAWKEY y KONSTANTIN BEZNOV. An integrated view of human, organizational, and technological challenges of IT security management. In: *Information Management & Computer Security*. 2009, Vol. Vol. 17, pp. 4-19. no. 1
5. Untitled Document. In: [online]. [Accessed 20 febrero 2014]. Available from: http://www.bc.gov.cu/anteriores/RevistaBCC/2011/revista_2_2011/Seguridad%20inform%C3%A1tica.html.
6. Mis diez herramientas esenciales para administrar sistemas - Bitelia. In: [online]. [Accessed 21 febrero 2014]. Available from: <http://bitelia.com/2011/01/diez-herramientas-esenciales-administrar-sistemas>.
7. 1. Definición de Seguridad Informática | Gestión de Riesgo en la Seguridad Informática. In: [online]. [Accessed 20 febrero 2014]. Available from: http://protejete.wordpress.com/gdr_principal/definicion_si/.
8. Real Academia Española. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.rae.es/>.
9. FREDERIC MAMPOU. *Sistema de Gestió de Seguridad de la informació*. S.l.: s.n.
10. ISO/IEC 27001. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.nyce.org.mx/index.php/sistemas/iso-27001>.
11. UNIT onLine. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.unit.org.uy/iso27000/iso17799.php>.
12. - *Res 127 2007 MIC Reglamento de Seguridad Informatica.pdf* [online]. S.l.: s.n. [Accessed 20 mayo 2014]. Available from: <file:///C:/Users/Jessie/Desktop/Res%20127%202007%20MIC%20Reglamento%20de%20Seguridad%20Informatica.pdf>.

13. ISO/IEC 17799. *ESTÁNDAR INTERNACIONAL*. 15 junio 2005. S.l.: s.n. Tecnología de la Información – Técnicas de seguridad – Código para la práctica de la gestión de la seguridad de la información
14. OCS Inventory NG - Interfase S.A. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.interfase.com.uy/index.php/impsistemas/ocsinventory>.
15. GLPI - Gestionnaire libre de parc informatique. In: [online]. [Accessed 21 febrero 2014]. Available from: <http://www.glpi-project.org/>.
16. BECK, K. *Extreme Programming Explained*. 1999. S.l.: s.n. Pearson Education
17. Extreme Programming: A Gentle Introduction. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.extremeprogramming.org/>.
18. *Metodologías Agiles - METODOLOGIAS AGILES.pdf* [online]. S.l.: s.n. [Accessed 20 febrero 2014]. Available from: <http://www.sisman.utm.edu.ec/libros/FACULTAD%20DE%20CIENCIAS%20ZOOT%20C3%89CNICAS/CARRERA%20DE%20INGENIER%20C3%8DA%20EN%20INFORMATICA%20AGROPECUARIA/07/INGENIERIA%20DEL%20SOFTWARE%20I/METODOLOGIAS%20AGILES.pdf>.
19. Modelo RUP | Ing. en Software. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://softwarerecopilation.wordpress.com/modelo-rup/>.
20. Affordable UML Tool with Full UML, SysML and ERD Supports. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.visual-paradigm.com/product/vpuml/editions/modeler.jsp>.
21. Qué es uml, PARA QUE SIRVE, PASOS. In: [online]. [Accessed 21 febrero 2014]. Available from: <http://www.slideshare.net/myle22/qu-es-uml-para-que-sirve-pasos>.
22. *Manual_de_PHP.pdf* [online]. S.l.: s.n. [Accessed 20 mayo 2014]. Available from: file:///C:/Users/Jessie/Desktop/Manual_de_PHP.pdf.
23. *HTMLGuiaCompleta.pdf* [online]. S.l.: s.n. [Accessed 20 mayo 2014]. Available from: <file:///C:/Users/Jessie/Desktop/HTMLGuiaCompleta.pdf>.
24. Características de Java Script. In: [online]. [Accessed 20 febrero 2014]. Available from: http://www.uazuay.edu.ec/estudios/sistemas/lenguaje_iii/MANualJavaScript/caracteristicas.htm.
25. *Microsoft Word - CU010011D selector CSS id reglas precedencia ejemplos.docx - index.php* [online]. S.l.: s.n. [Accessed 20 febrero 2014]. Available from: http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=631.
26. Qué es y para qué sirve el lenguaje CSS (Cascading Style Sheets - Hojas de Estilo). In: [online]. [Accessed 20 febrero 2014]. Available from: http://www.aprenderaprogramar.com/index.php?option=com_content&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&Itemid=163.

27. *manualcpp.pdf* [online]. S.l.: s.n. [Accessed 21 febrero 2014]. Available from: <http://mat21.etsii.upm.es/ayudainf/aprendainf/Cpp/manualcpp.pdf>.
28. Resultados de la búsqueda. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.codigoqt.com/index.php?action=search2>.
29. *Microsoft Word - INTRODUCCIÓN AL QT Y AL QT DESIGNER - http://www.unicauca.edu.co/deic/Documentos/INTRODUCCI%D3N%20AL%20QT%20CREATOR.pdf* [online]. S.l.: s.n. [Accessed 22 febrero 2014]. Available from: <http://www.unicauca.edu.co/deic/Documentos/INTRODUCCI%D3N%20AL%20QT%20CREATOR.pdf>.
30. PhpStorm descarga gratis. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.softpedia.es/programa-PhpStorm-161476.html>.
31. Symfony 2 | Symfony para Todos. In: [online]. [Accessed 17 marzo 2014]. Available from: <http://symfony.cubava.cu/introduccion/symfony-2/>.
32. Apache HTTP Server - Descargar. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://apache-http-server.softonic.com/>.
33. Ventajas de utilizar Node.js. In: [online]. [Accessed 20 febrero 2014]. Available from: <http://www.pixelovers.com/ventajas-utilizar-nodejs-1953900>.
34. ING. JOSÉ JOSKOWICZ. *Reglas y Prácticas en eXtreme Programming*. S.l.: s.n.
35. *08-Patrones1.ppt - 08-Patrones.pdf* [online]. S.l.: s.n. [Accessed 2 abril 2014]. Available from: <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
36. INGENIERIA DE SOFTWARE: Patrones de diseño y frameworks. In: [online]. [Accessed 8 abril 2014]. Available from: <http://ingdesw2.blogspot.com/p/patrones-de-diseno-y-frameworks.html>.
37. INGENIERIA DE SOFTWARE: Patrones de diseño y frameworks. In: [online]. [Accessed 8 abril 2014]. Available from: <http://ingdesw2.blogspot.com/p/patrones-de-diseno-y-frameworks.html>.
38. Re: [symfony-users] I am working with symfony, and i need more info about de dassing patterns - Thomas Rabaix - com.googlegroups.symfony-users - MarkMail. In: [online]. [Accessed 8 abril 2014]. Available from: <file:///C:/Users/Jessi/Desktop/Patrones/Re%20%20%5Bsymfony-users%5D%20I%20am%20working%20with%20symfony,%20and%20i%20need%20more%20info%20about%20de%20dassing%20patterns%20-%20Thomas%20Rabaix%20-%20com.googlegroups.symfony-users%20-%20MarkMail.htm>.
39. Guía de estilo de programación. In: [online]. [Accessed 8 abril 2014]. Available from: <http://beosman.org/docs/guia-estilo/>.

Glosario de términos

Programación orientada por eventos

La programación orientada por eventos es un paradigma de programación en el que el flujo del programa está determinado por eventos o mensajes desde otros programas o hilos de ejecución. Las aplicaciones desarrolladas con programación dirigida por eventos implementan un bucle principal o main loop donde se ejecutan las dos secciones principales de la aplicación: El selector de eventos y el manejador de eventos.

Activo informático

Es todo elemento que compone todo el proceso de la comunicación, partiendo desde la información, emisor, medio de transmisión y receptor.

Riesgo

Son las probabilidades de que las amenazas exploten los puntos débiles o vulnerabilidades causando daños y pérdidas, y afectando completamente la Integridad, Confidencialidad y Disponibilidad.

Seguridad

Es una práctica orientada hacia la eliminación de las vulnerabilidades para evitar o reducir las posibilidades que las potenciales amenazas se concreten.

Confidencialidad

Es la propiedad de la información, por la que se garantiza que está accesible únicamente a personal autorizado a acceder a dicha información. La confidencialidad ha sido definida por la Organización Internacional de Estandarización (ISO) en la norma ISO-17799 como "garantizar que la información es accesible sólo para aquellos autorizados a tener acceso" y es una de las piedras angulares de la seguridad de la información. La confidencialidad es uno de los objetivos de diseño de muchos criptosistemas, hecha posible en la práctica gracias a las técnicas de criptografía moderna.

Integridad

Es la propiedad que busca mantener los datos libres de modificaciones no autorizadas. No es igual a integridad referencial en base de datos. A groso modo, la integridad es el mantener con exactitud la información tal cual fue generada, sin ser manipulada o alterada por personas o

procesos no autorizados. La violación de integridad se presenta cuando un empleado, programa o proceso (por accidente o con mala intención) modifica o borra los datos importantes que son parte de la información, así mismo hace que su contenido permanezca inalterado a menos que sea modificado por personal autorizado, y esta modificación sea registrada, asegurando su precisión y confiabilidad. La integridad de un mensaje se obtiene adjuntándole otro conjunto de datos de comprobación de la integridad: la firma digital. Es uno de los pilares fundamentales de la seguridad de la información.

Disponibilidad

Es la característica, cualidad o condición de la información de encontrarse a disposición de quienes deben acceder a ella, ya sean personas, procesos o aplicaciones. A groso modo, la disponibilidad es el acceso a la información y a los sistemas por personas autorizadas en el momento que así lo requieran. En el caso de los sistemas informáticos utilizados para almacenar y procesar la información, los controles de seguridad utilizada para protegerlo, y los canales de comunicación protegidos que se utilizan para acceder a ella deben estar funcionando correctamente. La Alta disponibilidad sistemas objetivo debe estar disponible en todo momento, evitando interrupciones del servicio debido a cortes de energía, fallos de hardware, y actualizaciones del sistema.

V8 (motor JavaScript)

Es un motor de código abierto para JavaScript creado por Google, siendo su programador jefe Lars Bak. Está escrito en C++ y es usado en Google Chrome. Puede funcionar de manera individual (standalone) o incorporada a cualquier aplicación C++.

Plugin

Es una noción que no forma parte del diccionario de la Real Academia Española (RAE). Se trata de un concepto de la lengua inglesa que puede entenderse como “inserción” y que se emplea en el campo de la informática. Un plugin es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software. En nuestro idioma, por lo tanto, puede nombrarse al plugin como un complemento.

Anexos

Anexo 1: Casos de pruebas

Tabla 39: Prueba de aceptación. No 3

Caso de Prueba de Aceptación	
Código: HU3_P3	Historia de Usuario: Eliminar área
Nombre: Eliminar un área en el sistema.	Iteración: 1
Descripción: Prueba para la funcionalidad eliminar un área.	
Condiciones de ejecución: Estar autenticado en el sistema. Debe existir al menos un área en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Gestión de áreas). Seleccionar la opción (Listar áreas). Presionar el botón (Eliminar área).	
Resultados esperados: Se ha eliminado el área satisfactoriamente.	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 40: Prueba de aceptación. No 4

Caso de Prueba de Aceptación	
Código: HU4_P4	Historia de Usuario: Editar área
Nombre: Editar un área en el sistema.	Iteración: 1
Descripción: Prueba para la funcionalidad editar área.	
Condiciones de ejecución: Estar autenticado en el sistema. Debe existir al menos un área en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Gestión de áreas). Seleccionar la opción (Listar áreas). Presionar el botón (Editar área).	

<p>Entrar los datos: Área de responsabilidad. Local. Responsable Cantidad de computadoras</p>
Resultados esperados: Se ha editado el área satisfactoriamente.
Evaluación de la prueba: Resultado satisfactorio.

Tabla 41: Prueba de aceptación. No 5

Caso de Prueba de Aceptación	
Código: HU5_P	Historia de Usuario: Listar área
Nombre: Listar todas las áreas del sistema.	Iteración: 1
Descripción: Prueba a la funcionalidad listar todas las áreas del sistema.	
Condiciones de ejecución: Estar autenticado en el sistema. Debe existir al menos un área en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Gestión de áreas). Seleccionar la opción (Listar áreas).	
Resultados esperados: Listado de todas las áreas satisfactoriamente.	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 42: Prueba de aceptación. No 8

Caso de Prueba de Aceptación	
Código: HU8_P8	Historia de Usuario: Adicionar datos al expediente técnico
Nombre: Adicionar datos al expediente técnico.	Iteración: 2
Descripción: Se realizará la prueba a la funcionalidad Adicionar datos al expediente técnico de cada computadora que tiene el sistema.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución:	

<p>Seleccionar la opción (Generar Expediente) en computadoras nuevas.</p> <p>Llenar los campos vacíos como ejemplo:</p> <p>No. Sello</p> <p>No. Inventario</p> <p>Puesto</p> <p>Presionar el botón (Insertar datos).</p>
Resultados esperados: La computadora ahora pasa a un estado de asignada.
Evaluación de la prueba: Resultado satisfactorio.

Tabla 43: Prueba de aceptación. No 9

Caso de Prueba de Aceptación	
Código: HU9_P9	Historia de Usuario: Imprimir expediente técnico
Nombre: Imprimir expediente técnico.	Iteración: 2
Descripción: Prueba a la funcionalidad imprimir expediente técnico de cada computadora perteneciente al sistema.	
Condiciones de ejecución:	
Estar autenticado en el sistema.	
Todos los datos del historial deben estar completados.	
Entrada/Pasos de ejecución:	
Seleccionar la opción (Generar Expediente) en computadoras nuevas.	
Presionar el botón (Imprimir expediente).	
Resultados esperados: La computadora ahora pasa a un estado de asignada.	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 44: Prueba de aceptación. No 10

Caso de Prueba de Aceptación	
Código: HU10_P10	Historia de Usuario: Ver historial de modificaciones
Nombre: Ver historial de modificaciones.	Iteración: 2
Descripción: Prueba a la funcionalidad ver historial de modificaciones.	
Condiciones de ejecución:	
Estar autenticado en el sistema.	
Entrada/Pasos de ejecución:	
Seleccionar la opción (Generar Expediente) en computadoras nuevas.	

Presionar el botón (Ver Historial modificaciones).
Resultados esperados: Listado de modificaciones e incidencias de la computadora.
Evaluación de la prueba: Resultado satisfactorio.

Tabla 45: Prueba de aceptación. No 11

Caso de Prueba de Aceptación	
Código: HU11_P11	Historia de Usuario: Imprimir historial de modificaciones
Nombre: Imprimir historial de modificaciones	Iteración: 3
Descripción: Prueba a la funcionalidad imprimir historial de modificaciones de cada computadora perteneciente al sistema.	
Condiciones de ejecución: Estar autenticado en el sistema. Tener editado los datos de modificaciones.	
Entrada/Pasos de ejecución: Seleccionar la opción (Generar Expediente) en computadoras nuevas. Presionar el botón (Ver Historial modificaciones). Presionar el botón Imprimir.	
Resultados esperados: Datos imprimidos satisfactoriamente.	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 46: Prueba de aceptación. No 12

Caso de Prueba de Aceptación	
Código: HU12_P12	Historia de Usuario: Editar historial de modificaciones
Nombre: Editar historial de modificaciones	Iteración: 3
Descripción: Se realizará la prueba a la funcionalidad editar historial de modificaciones.	
Condiciones de ejecución: Estar autenticado en el sistema. Computadora analizada debe tener al menos una incidencia.	
Entrada/Pasos de ejecución: Seleccionar la opción (Generar Expediente) en computadoras nuevas. Presionar el botón (Ver Historial modificaciones). Presionar el botón (Editar Incidencia) dentro de las opciones de incidencias. Llenar los campos	

No. Gat
No. Sello
Incidencia
Presionar botón (Aceptar cambio).
Resultados esperados: Datos editados satisfactoriamente.
Evaluación de la prueba: Resultado satisfactorio.

Tabla 47: Prueba de aceptación. No 13

Caso de Prueba de Aceptación	
Código: HU13_P13	Historia de Usuario: Reportar como rota una computadora
Nombre: Reportar como rota una computadora	Iteración: 3
Descripción: Prueba a la funcionalidad reportar como rota una computadora.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Computadora asignada). Presionar el botón (Reportar como rota) dentro de las opciones de computadora asignada.	
Resultados esperados: La computadora ha pasado a un estado de rota	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 48: Prueba de aceptación. No 14

Caso de Prueba de Aceptación	
Código: HU14_P14	Historia de Usuario: Eliminar computadora rota
Nombre: Eliminar computadora rota	Iteración: 3
Descripción: Prueba a la funcionalidad eliminar computadora rota.	
Condiciones de ejecución: Estar autenticado en el sistema. Computadora en estado rota.	
Entrada/Pasos de ejecución: Seleccionar la opción (Computadora rota). Presionar el botón (Eliminar) dentro de las opciones de computadora rota.	
Resultados esperados: La computadora se ha borrado satisfactoriamente.	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 49: Prueba de aceptación. No 15

Caso de Prueba de Aceptación	
Código: HU15_P15	Historia de Usuario: Adicionar computadora manualmente
Nombre: Adicionar computadora manualmente en estado de rota.	Iteración: 3
Descripción: Prueba a la funcionalidad adicionar computadora manualmente en estado de rota.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Insertar computadora rota). Llenar los campos. Nombre de la computadora. Macc. Estado. Incidencia. No. Inventario. No. Sello. No. Serie. Presionar botón (Adicionar).	
Resultados esperados: Se ha insertado la computadora satisfactoriamente.	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 50: Prueba de aceptación. No 17

Caso de Prueba de Aceptación	
Código: HU17_P17	Historia de Usuario: Eliminar historial de modificaciones
Nombre: Eliminar historial de modificaciones	Iteración: 3
Descripción: Prueba a la funcionalidad eliminar historial de modificaciones.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Generar Expediente) en computadoras nuevas o	

<p>Seleccionar la opción (Ver Expediente) en computadoras asignadas o rotas.</p> <p>Presionar el botón (Ver Historial modificaciones).</p> <p>Presionar el botón (Eliminar Incidencia) dentro de las opciones de incidencias.</p>
<p>Resultados esperados: Se ha eliminado la incidencia satisfactoriamente.</p>
<p>Evaluación de la prueba: Resultado satisfactorio.</p>

Tabla 51: Prueba de aceptación. No 18

Caso de Prueba de Aceptación	
Código: HU18_P18	Historia de Usuario: Ver historial de fecha
Nombre: Ver historial de fecha de cada computadora	Iteración: 3
Descripción: Se realizará la prueba a la funcionalidad ver historial de fecha de cada uno de los ordenadores pertenecientes al sistema.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Computadora Asignada). Seleccionar la opción (Ver Historial de fechas) dentro de las opciones de computadoras asignadas.	
Resultados esperados: Se visualiza todas las veces que se ha reportado la computadora.	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 52: Prueba de aceptación. No 19

Caso de Prueba de Aceptación	
Código: HU19_P19	Historia de Usuario: Eliminar historial de fecha
Nombre: Eliminar historial de fecha de cada computadora	Iteración: 3
Descripción: Se realizará la prueba a la funcionalidad eliminar los historiales de fecha perteneciente a computadoras que se registran en el sistema.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución: Seleccionar la opción (Computadora Asignada). Seleccionar la opción (Ver Historial de fechas) dentro de las opciones de computadoras asignadas.	

Seleccionar la opción (Eliminar historial).
Resultados esperados: Se eliminan todas las fechas de reportes de esa computadora
Evaluación de la prueba: Resultado satisfactorio.

Tabla 53: Prueba de aceptación. No 20

Caso de Prueba de Aceptación	
Código: HU20_P20	Historia de Usuario: Buscar incidencia en la computadora
Nombre: Buscar incidencia en la computadora	Iteración: 3
Descripción: Prueba a la funcionalidad buscar incidencia en la computadora.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución:	
Resultados esperados: Se buscará incidencias en la computadora.	
Evaluación de la prueba: Resultado satisfactorio.	

Tabla 54: Prueba de aceptación. No 21

Caso de Prueba de Aceptación	
Código: HU21_P21	Historia de Usuario: Enviar notificación de correo electrónico
Nombre: Enviar notificación de correo electrónico	Iteración: 3
Descripción: Se realizará prueba a la funcionalidad enviar notificación al correo mediante el sistema.	
Condiciones de ejecución: Estar autenticado en el sistema.	
Entrada/Pasos de ejecución:	
Resultados esperados: Se envía notificaciones a correos electrónicos.	
Evaluación de la prueba: Resultado satisfactorio.	

Anexo 2: Expediente técnico de computadoras

Primero se debe configurar el UIMConfig.ini de la siguiente manera:

Host = IP del servidor al cual se reportaría la computadora cliente.

Port = Viene predefinido 9098 el cual deberíamos dejar así.

Pass = Contraseña del sudo.

OCSAgentProgramDataPath = Dirección donde está el UIMSrv copiado /reportes

SendReportEach = Intervalo de tiempo de envío del reporte en segundos.

Escribir en consola `cd /ruta` donde está el UIMSrv luego se ejecutan los siguientes comandos

`./UIMSrv -i` para instalar el demonio.

`./UIMSrv -s` para ejecutar el servicio o mandar a correr la aplicación.

`./UIMSrv -t` para parar el servicio.

`./UIMSrv -u` para desinstalar el demonio.

Seguido de la instalación del UIMSrv se pasa a la configuración del script **UIMSrv.sh**:

Primero configurar el fichero UIMSrv.sh en (ruta) se actualiza y se pone la ruta en donde se encuentra el programa a ejecutar.

Para que inicie con el sistema de Linux

En consola de Linux (Terminal)

1- Dar los permisos **chmod 777 -R /ruta/UIMSrv.sh**

2- **nano /etc/rc.local**

2.1- Configurar ese fichero

2.2- **sh /ruta/UIMSrv.sh**

2.3- Remover el parametro **-e**

2.4- **sleep 100**

Para que inicie con la sección

1- Ir a la dirección

1.1- **cd /home/ (usuario) / (entorno de Linux)/ Autostart**

1.2- Copiar el script en esa dirección

1.2.1- **cp UIMSrv.sh/home/(usuario)/(entornoLinux)/Autostart**

UCInventoryServer Manual de instalación para Linux:

En la computadora que será designada para el servidor se debe tener instalado el siguiente programa:

sudo apt-get install node

sudo apt-get install nodejs

Para correr el servidor se debe situar en la carpeta donde se encuentra el script, luego escribir en consola (terminal)

node UCInventoryServer.js

UCInventoryWebInterface (InventarioActivo) Manual de instalación y modo de uso para Linux:

Copiar la aplicación web para la carpeta var/www/ luego abrir un navegador ejemplo Mozilla Firefox y escribir localhost seguido del puerto en caso de que se le haya creado un virtual host tener en cuenta que debe estar ejecutándose el servidor Apache.

La primera ventana que se visualizará es el login se debe autenticar para poder ingresar al sistema.



Figura 13: Login

Una vez que se haya ingresado los datos se mostrara la siguiente interfaz.

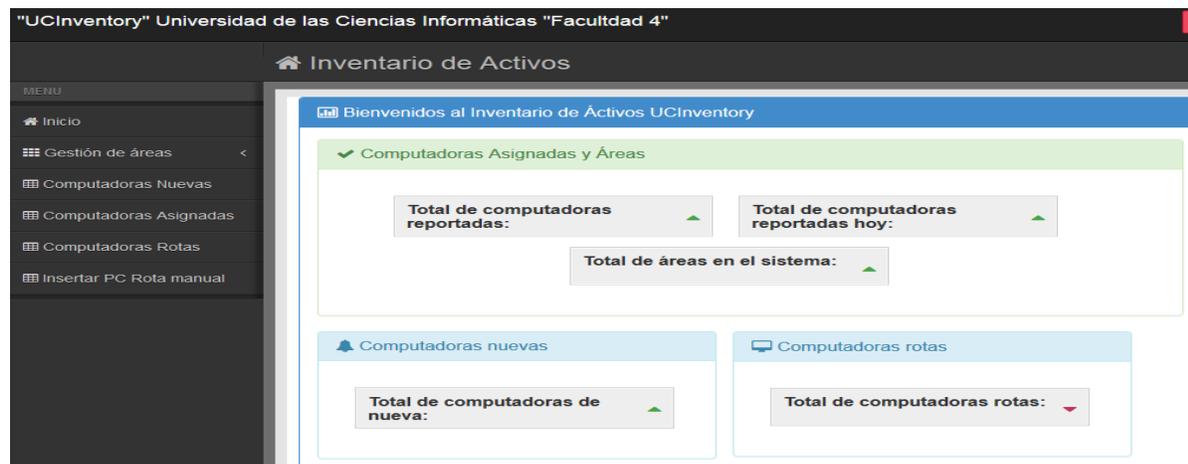


Figura 14: Inicio

Para crear un área en el sistema.

MENU

- Inicio
- Gestión de áreas <
- > Crear Áreas**
- > Listar Areas
- Computadoras Nuevas
- Computadoras Asignadas
- Computadoras Rotas
- Insertar PC Rota manual

Crear área

Área de responsabilidad :

Cantidad de computadoras:

Local:

Responsable:

✓ Adicionar ↻ Cancelar

Figura 15: Crear área

Listar áreas.

Gestión de áreas

Áreas creadas en el sistema.

Área de responsabilidad	Cantidad de PC	Local	Responsable	Acciones
Facultad 4	30	Laboratorio 207	Tecnico	 

Figura 16: Listar área

Editar las áreas.

Gestión de áreas

Áreas creadas en el sistema.

Área de responsabilidad	Cantidad de PC	Local	Responsable	Acciones
Facultad 4	30	Laboratorio 207	Tecnico	 

Figura 17: Editar área

Eliminar áreas.

Área de responsabilidad	Cantidad de PC	Local	Responsable	Acciones
Facultad 4	30	Laboratorio 207	Tecnico	 

Figura 18: Eliminar área

Gestión de computadoras nuevas, aquí se visualizan todas las computadoras reportadas al sistema la primera vez o que no se le haya generado su expediente, en esta funcionalidad se puede generar el expediente técnico, así como asignarle un área.

PC	Nombre de la Computadora	Estado	Sistema Operativo	Asignar Área	Generar Expediente
	D5-L207-PC24	Nueva	Microsoft Windows 8 Pro		<input type="button" value="Generar Expediente"/>

Figura 19: Listar computadoras nuevas

Funcionalidad Asignar área.

PC	Nombre de la Computadora	Estado	Sistema Operativo	Asignar Área	Generar Expediente
	D5-L207-PC24	Nueva	Microsoft Windows 8 Pro		<input type="button" value="Generar Expediente"/>

Figura 20: Asignar área

Área de responsabilidad	Local	Responsable
Facultad 4	<input type="text" value="Laboratorio 207"/>	Tecnico

Figura 21: Asignar área

Escoja el área a seleccionar

Área de responsabilidad	Local	Responsable
Facultad 4	Laboratorio 207	Tecnico

[Cancelar](#)

Figura 22: Asignar área

Generar expediente técnico.

Computadoras Reportadas en estado nueva.

PC	Nombre de la Computadora	Estado	Sistema Operativo	Asignar Área	Generar Expediente
	D5-L207-PC24	Nueva	Microsoft Windows 8 Pro		Generar Expediente

[Cancelar](#)

Figura 23: Generar expediente técnico

Expediente Técnico Computadora: [D5-L207-PC24] Sistema Operativo: [Microsoft Windows 8 Pro]

[Ver Historial de Modificaciones](#) [Cancelar](#)

**VICERRECTORIA DE TECNOLOGIA
EXPEDIENTE TECNICO**

DATOS GENERALES

Área de Responsabilidad: Facultad 4		
Local: Laboratorio 207	<input type="text" value="Puesto"/>	Puesto:
Responsable: Tecnico		
<input type="text" value="No. Inventario"/>	No. Inventario:	<input type="text" value="No. Serie"/>
	No. Serie:	<input type="text" value="No. Sello"/>
	No. Sello:	

DATOS TECNICOS

MOTHERBOARD	Fabricante: INTEL_	Modelo: DG965RY_	<input type="text" value="No. S"/>	No. Serie:
PROCESADOR	<input type="text" value="Modelo"/>	Modelo:	<input type="text" value="No. S"/>	No. Serie: Modelo: Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz [2 core(s) x64]
MEMORIA	Total: 512	Tipo: DDR2	Slot: 4	
DISCO DURO	Fabricante: Standard disk drives	Modelo: //PHYSICALDRIVE0	No. Serie: 6RA8FG8Q	
UNIDAD OPTICA	Fabricante: Standard CD-ROM drives	Modelo: HL-DT-ST DVD-RAM GSA-H55N SCSI CdRom Device	<input type="text" value="No. S"/>	No. Serie:
FUENTE	<input type="text" value="Modelo"/>	Modelo:	<input type="text" value="serie"/>	No. Serie:

OBSERVACIONES

FECHA Y FIRMA DEL RESPONSABLE

Figura 24: Generar expediente técnico

Historial de Modificaciones

Expediente Técnico/Historial de Modificaciones de la computadora D5-L207-PC24



VICERRECTORIA DE TECNOLOGIA

EXPEDIENTE TECNICO

HISTORIAL DE MODIFICACIONES

Fecha	Descripción	No. Orden GAT	No. Sello	Opciones

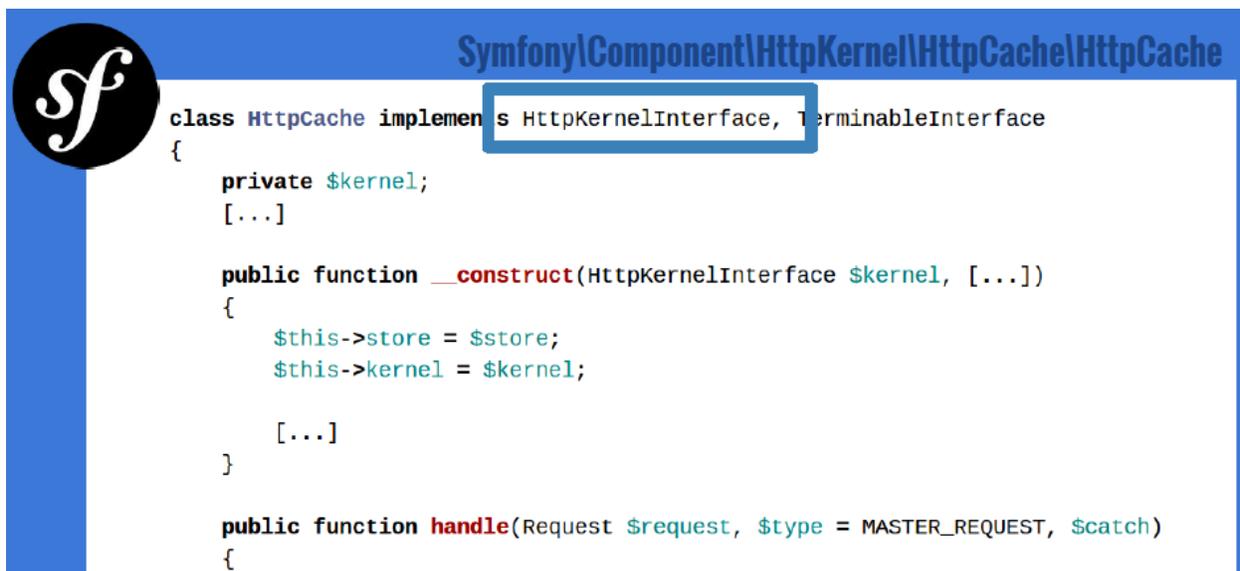
Figura 25: Historial de modificaciones

Anexo 4: No conformidades

- ✓ En la interfaz principal, el menú que se encuentra en la parte izquierda la funcionalidad gestión de áreas la palabra gestión la tiene sin tilde.
- ✓ En la funcionalidad crear área se encuentra el nombre del campo ``área de responsabilidad`` donde la palabra área se encuentra sin tilde
- ✓ En la funcionalidad eliminar computadora no existía una confirmación que le informe al administrador si se ha eliminado o no
- ✓ La funcionalidad generar expediente técnico el botón generar expediente no funciona.
- ✓ Cuando se van a listar las computadoras nuevas no se ve ninguna lista de estas.
- ✓ En el expediente técnico no se actualizan los datos.
- ✓ En la funcionalidad asignar área el botón cancelar no funciona.

- ✓ El expediente técnico tiene falta de ortografía en el nombre que tiene en la parte superior derecha como por ejemplo técnico y tecnología.
- ✓ La funcionalidad que me aparece en el menú principal computadoras rotas no funciona.
- ✓ La funcionalidad de imprimir el historial de modificaciones no imprime ese historial lo que me imprime es el expediente técnico.
- ✓ En historial de modificaciones la palabra descripción no contiene tilde.
- ✓ En la funcionalidad computadoras asignadas y listar computadora nueva el icono de la máquina que aparece no varía siempre esta del mismo color.
- ✓ No existe un botón cancelar dentro de la funcionalidad editar área.
- ✓ No existe un botón cancelar en la confirmación eliminar computadora.
- ✓ No existe una confirmación al reportar una computadora como rota.
- ✓ Cuando se inserta una computadora manualmente no existe la opción de asignarle un área.
- ✓ No existe una opción que limpie el historial de fecha del día.
- ✓ Cuando se aceptan los cambios de incidencia no se actualiza el nuevo número de sello.
- ✓ Cuando se inserta una computadora manualmente la mac no es un valor único.

Anexo 5: Patrones de diseño



```

Symfony\Component\HttpKernel\HttpCache\HttpCache
class HttpCache implements HttpKernelInterface, TerminableInterface
{
    private $kernel;
    [...]

    public function __construct(HttpKernelInterface $kernel, [...])
    {
        $this->store = $store;
        $this->kernel = $kernel;

        [...]
    }

    public function handle(Request $request, $type = MASTER_REQUEST, $catch)
    {

```

Figura 26: Patrón Decorator