

Universidad de las Ciencias Informáticas
Facultad 1



Título: Personalización de la herramienta de gestión de inventarios
OCS Inventory NG.

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

Autores: Diana Lilian Antúnez Ginarte
Mario Lázaro García Pérez

Tutores: Ing. Gladys Marsi Peñalver Romero
Ing. Eduardo Alejandro Cuesta Llanes

Ciudad de la Habana, Cuba.

Mayo, 2011

“Año 53 de la Revolución”

Pensamiento



"El hombre debe transformarse al mismo tiempo que la producción progresa; no realizaríamos una tarea adecuada si fuéramos tan solo productores de artículos, de materias primas y no fuéramos al mismo tiempo productores de hombres."

Declaración de autoría

Declaramos ser los únicos autores de este trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de ____ del año 2011.

Diana Lilian Antúnez Ginarte

Mario Lázaro García Pérez

Ing. Gladys Marsi Peñalver Romero

Ing. Eduardo Alejandro Cuesta Llanes

Agradecimientos

Haber llegado hasta aquí y ver cumplido el sueño que tanto hemos deseado, no hubiera sido posible sin el apoyo de todas aquellas personas que desde que éramos niños y hasta los días de hoy han estado de una forma u otra a nuestro lado; nuestros familiares, amigos y profesores. También gracias a esta Revolución por habernos permitido estudiar una carrera en la cual podemos realizarnos como profesionales.

Diana:

A mis queridos padres, a quien les debo la vida, los que con su esmero hicieron posible este sueño. A mi querida hermana, por ser mi guía y siempre ser mi negra linda, sabes que yo te adoro. A todos los que de una u otra forma han colaborado en la realización de este trabajo.

Mario:

A mis padres que han sido mis guías y mi ejemplo a seguir, gracias por sus consejos, por cuidar y estar pendientes de cada uno de mis pasos. Por siempre estar ahí cuando los necesito, por ayudarme siempre con mis problemas y apoyarme cada día.

A mis hermanos Sandra y Norvelio que también han sido parte importante de mi vida y con los que he podido contar para todo lo que me he propuesto en la vida.

A todos mis amigos quienes siempre han estado conmigo en los buenos y malos momentos escuchándome y aconsejándome, en especial a Vismar por su incondicional ayuda en la realización de este trabajo, a Eduardo por su apoyo desde que estábamos en el pre hasta los días de hoy, a Lianet que siempre se preocupó por mi y a la cual considero como una hermana, a Arasay por siempre estar dispuesta cada vez que la necesito, a Ander por soportarme todo este tiempo y brindarme su ayuda.

A todos mis compañeros de aula y proyecto que de una forma o otra contribuyeron en este trabajo, a Odaimy, Yaniris y mi querida compañera de tesis Diana que compartieron conmigo los tensos días de realización de este trabajo.

A todos mis profesores que han contribuido a mi formación durante estos 5 años. En especial a Yoandy quien a sido mi ejemplo a seguir desde que era mi Alumno Ayudante.

Por último y no menos importante a mis tutores Gladys y Eduardo por sabernos guiar y ayudar con las dudas que nos han surgido.

Dedicatoria

A todos nuestros seres queridos, en especial nuestros padres. A la universidad que aportó en nuestra formación.

Resumen

El proceso de migración a Software Libre llevado a cabo por el proyecto Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS), requiere como primer paso la realización de un levantamiento informático con el fin de obtener información sobre el hardware y software de cualquier ordenador. Para ello se hace uso de la herramienta de gestión de inventarios OSC Inventroy NG. Dicha herramienta requiere de la implementación de un módulo de control de cambios así como de otras funcionalidades para su mejor funcionamiento.

En la presente investigación se realiza una personalización a OCS Inventory NG con la integración de un módulo que gestiona el control de cambios surgiendo de esta forma el sistema Gestión de Inventarios de Tecnología Informática (GITI). Para ello se realizó un estudio de diferentes sistemas de gestión de inventarios, identificando características y funcionalidades que fueron incorporadas a la herramienta. Se identificaron las tecnologías, herramientas, lenguajes de programación y la metodología de desarrollo de software que guiaron el desarrollo del sistema. Además se identificaron y documentaron las funcionalidades que el sistema debía cumplir acorde a los requisitos establecidos por el cliente.

Palabras Claves: control de cambios, desarrollo de software, inventario de hardware y software, sistemas de gestión de inventarios.

Índice

Resumen	VI
Introducción	1
Capítulo 1: Fundamentación teórica.	5
1.1 Conceptos asociados al dominio del problema	5
1.2 Análisis de los sistemas de gestión de inventarios informáticos que implementan el control de cambios	6
1.4 Valoración del estudio realizado	18
1.5 Metodología de desarrollo de software	18
1.5.1 Metodologías ágiles	19
1.6 Lenguajes, tecnologías y herramientas de desarrollo utilizadas	20
1.6.1 Lenguajes	20
1.6.2 Tecnologías	22
1.6.3 Herramientas	25
Capítulo 2: GITI, personalización de la herramienta OCS Inventory NG.	29
2.1 Características funcionales del sistema	29
2.2.1 Planificación del proyecto por roles	32
2.3 Captura de Requisitos	33
2.3.1 LRP	33
2.4 Historia de Usuario (HU)	36
2.4.1 HU Gestionar notificaciones	36
2.4.2 HU Dar de baja a un equipo	38
2.4.3 HU Gestionar envío de correo electrotécnico	39
2.4.4 HU Gestionar inventario manual	40
2.4.5 HU Reportes en PDF	43
2.5 Plan de Release	43
2.6 Lista de Riesgo	44
2.7 Arquitectura y Modelo del diseño	46
2.8 Modelo de Datos	47
2.9 Tareas de Ingenierías	47
2.9.1 Tareas de ingenierías asociadas a la Historia de Usuario Gestionar notificaciones	48
2.9.2 Tareas de Ingenierías asociadas a la Historia de Usuario Gestionar inventario manual	49
2.9.3 Tareas de Ingenierías asociadas a la Historia de Usuario Dar de baja a un equipo	50
2.9.4 Tareas de Ingenierías asociadas la Historia de Usuario Gestionar envío de correo electrónico	50
2.9.5 Tareas de Ingenierías asociadas la Historia de Usuario Reportes en PDF	51
Capítulo 3: Pruebas al Sistema.	52
3.1 Casos de pruebas de Aceptación	52
3.2 Resultados obtenidos	56
Conclusiones	58
Recomendaciones	59
Referencias Bibliográficas	60
Bibliografía	62
Anexos	66
Glosario de Términos	69

Introducción

Actualmente la humanidad vive en un mundo prácticamente privatizado donde el Software Libre (SWL) y el Código Abierto han cobrado auge como una alternativa a países e instituciones que no pueden tener acceso a las caras tecnologías privadas. Promoviendo principios como el trabajo en equipo, la no privatización del código fuente y la libertad de acceso a la información, el SWL constituye actualmente un amplio movimiento tecnológico que es considerado por muchos como el principal enemigo del sistema capitalista. Para un país como Cuba, que se encuentra bloqueado económicamente por el gobierno de los Estados Unidos; nación donde tienen sede las principales empresas productoras de software privado del planeta, el uso de las tecnologías libres viene a ser más que una alternativa, una obligación en aras de impulsar el desarrollo de soluciones informáticas que no solo solventen los problemas del país sino que puedan ser exportadas al exterior convirtiéndose en una fuente de ingresos para el mismo.

Ante la necesidad de migrar hacia el uso del Software Libre y valorando la importancia de lograr una correcta adaptación de todas las entidades, ministerios y empresas nacionales que aún no lo han hecho, el proyecto Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS) perteneciente al departamento de Migración y Soporte del Centro de Software Libre (CESOL) integrado a la Facultad 1 en la Universidad de las Ciencias Informáticas (UCI), asumió la responsabilidad de llevar a cabo este proceso de migración. Para ello, el proyecto propone automatizar todos los procesos y tareas que se realizan durante la migración, mediante el desarrollo de una plataforma que integre todas las herramientas necesarias.

Este proceso se rige por la Metodología cubana para la migración a Software Libre y Código Abierto, propuesta en el año 2008, la cual está compuesta por 4 etapas fundamentales: *Preparación, Migración Parcial, Migración Total y Consolidación* [1]. En la etapa de Preparación, esta metodología propone recopilar información del estado actual de los activos informáticos de la institución, lo cual implica realizar un levantamiento tanto del software como del hardware de cada una de las estaciones de trabajo presentes en la misma, siendo de vital importancia pues permitirá definir elementos que posteriormente serán cruciales en las etapas siguientes.

Los sistemas encargados de realizar este tipo de tareas reciben el nombre de sistemas de gestión de inventarios y su principal utilidad es que permiten tener un control organizado y completo de cualquier ente informático tanto en términos de hardware incorporado como de software instalado. En la actualidad existen varias herramientas de gestión de inventario informático, algunas de ellas privadas o libres, por citar algunas: LiveState Discovery, Unicenter TNG, Software Asset Management (SAM), LogInventory, Everest Corporate Edition, OpenViewPCWizar, ProactivaNET,

OCS Inventory NG. Estas guardan la información recopilada en diferentes formatos que se almacenan en su sistema de base de datos, emitiendo informes que pueden ser procesados posteriormente. Después de una búsqueda de todas las soluciones similares existentes a nivel mundial, el proyecto SIMAYS escogió la herramienta de gestión de inventarios *Open Computer and Software Inventory Next Generation* (OCS Inventory NG) [2], para sustentar el proceso de recopilación de información en las diversas instituciones y entidades involucradas en la migración.

OCS Inventory NG es una herramienta de libre distribución que permite a los usuarios administrar el inventario de sus activos de tecnología informática. La misma recopila información sobre el hardware y software de equipos que forman parte de la red donde se instala. Esta puede 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. Además, posee opciones como escaneo de la red o instalación de aplicaciones remotamente. Después de su puesta en práctica dentro del proyecto SIMAYS durante más de un año de trabajo, la herramienta tiene una limitación muy propia de la mayoría de los sistemas de gestión de inventarios de la actualidad; no provee una funcionalidad que permita mantener un control de los cambios ocurridos tras haber realizado varios inventarios en una misma estación de trabajo.

Partiendo de todo lo anteriormente planteado se identifica el siguiente **problema científico**:

¿Cómo lograr que la herramienta OCS Inventory NG lleve a cabo el control de cambios?

Como **objeto de estudio** de la presente investigación se tiene: los sistemas de gestión de inventarios, siendo el **campo de acción**: la herramienta OCS Inventory NG empleada en la migración a Software Libre por el proyecto SIMAYS.

Para dar solución al problema existente se tiene como **objetivo general**: desarrollar un módulo para el control de cambios en la herramienta OCS Inventory NG.

Derivándose de este los siguientes **objetivos específicos**:

- Realizar un estudio de los sistemas de gestión de inventarios que implementan el control de cambios.
- Identificar las tecnologías y estándares que puedan ser útiles en la implementación de la solución propuesta.
- Implementar el módulo de control de cambios para la herramienta OCS Inventory NG.

- Realizar pruebas al módulo implementado.

Idea a defender:

El desarrollo de un módulo para el control de cambios en la herramienta OCS Inventory NG aumentará la eficiencia de la misma, dotándola de las funcionalidades necesarias para ser usada en la migración a SWL propuesta por el proyecto SIMAYS.

Para satisfacer los objetivos específicos se planifican las siguientes **tareas de investigación:**

- Análisis de la bibliografía existente sobre las herramientas de gestión de inventarios que implementan la gestión de cambios.
- Estudio de la metodología de desarrollo a utilizar.
- Análisis y diseño del proceso de control de cambios para la herramienta OCS Inventory NG.
- Implementación del módulo diseñado e integración del mismo con la herramienta OCS Inventory NG.
- Elaboración y ejecución de un plan de pruebas al módulo implementado.

La investigación estará sustentada en los siguientes **métodos científicos:**

Teóricos:

El empleo del método **histórico-lógico** posibilitará la comprensión lógica del objeto de estudio haciendo un análisis riguroso de sus antecedentes y el proceso evolutivo por el cual han transitado todas las tecnologías relacionadas con la gestión de inventario y el control de cambios.

Como técnica de recopilación de información se utilizará la **entrevista** con el propósito de obtener información, experiencias, ideas y puntos de vistas, que contribuyan al desarrollo de la investigación y aporten conocimientos específicos del tema.

Las entrevistas se realizarán de forma no estructuradas con el cliente, con el fin de entender el proceso de personalización que se le realizará a la herramienta así como profundizar en los detalles de las nuevas funcionalidades de la misma.

El documento está estructurado de la siguiente manera:

Capítulo 1: Fundamentación teórica: En este capítulo se realiza un estudio acerca de los sistemas de gestión de inventarios existentes en la actualidad que implementan el control de cambios, así como sus principales características y funcionalidades. Se abordaran varios conceptos claves. Contiene además la argumentación de la Metodología de Desarrollo y las tecnologías empleadas para el desarrollo de la solución propuesta.

Capítulo 2: GITI, personalización de la herramienta OCS Inventory NG: Se presenta la propuesta de solución al sistema. Todas las características de la herramienta OCS Inventory NG. Se realiza la captura de requisitos funcionales y no funcionales. Se priorizan las historias de usuario a implementar así como las tareas asociadas a cada una de ellas, se realizan los diagramas de diseño, componente y datos. Se elabora la lista de riesgos.

Capítulo 3: Pruebas al sistema: Se realizan las pruebas necesarias para poder validar que las funcionalidades implementadas dan cumplimiento al sistema desarrollado. Se realiza un plan de pruebas, un grupo de pruebas de aceptación así como el análisis del impacto de la solución propuesta.

Capítulo 1: Fundamentación teórica.

El proceso de migración a SWL de las instituciones y empresas nacionales se rige por una metodología de migración. Durante este proceso el uso de OCS Inventory NG como herramienta de gestión de inventarios juega el papel principal por lo que se hace necesario que la misma provea como funcionalidad primordial un módulo que gestione el control de cambios, además de otras funcionalidades tales como: gestionar el inventario manualmente, dar de baja a un equipo y generar reportes en formato PDF.

En el presente capítulo se realiza un estudio de las principales herramientas de gestión de inventarios que implementan el control de cambios, con el fin de obtener información a ser utilizada para la personalización de la herramienta OCS Inventory NG, además de definir una serie de tecnologías y herramientas, que guiadas por una metodología de desarrollo permitan la implementación de las funcionalidades exigidas por el proyecto SIMAYS para el sistema.

Para lograr una mejor comprensión de los diferentes términos que se emplearán en la investigación a lo largo del capítulo se hace necesario definir una serie de conceptos.

1.1 Conceptos asociados al dominio del problema

Inventario de software

Seguimiento del software que está instalado en las estaciones de trabajo y otros dispositivos, registrando además las actualizaciones por las que ha transitado.

Inventario de hardware

Seguimiento de las características físicas de activos informáticos, tales como computadoras, impresoras, monitores, teléfonos; incluyendo componentes como la configuración de memoria, tarjeta gráfica, tarjeta de red, o el contenido del procesador.

Parques informáticos

Zona o lugar de concentración de un conjunto de equipos informáticos de los que dispone una colectividad.

Sistemas de gestión de inventarios

Los sistemas de gestión de inventarios son herramientas que permiten un control de la

información de cualquier ordenador brindando y almacenando información sobre el hardware y software del mismo, en vez del tradicional inventario manual. Dicha información va desde que tipo de tarjeta de video usa la computadora, las licencias de software instaladas y en uso, así como sistema operativo instalado.

Luego de definir los conceptos básicos se realiza un estudio basado en los sistemas de gestión de inventarios que implementan el control de cambios.

1.2 Análisis de los sistemas de gestión de inventarios informáticos que implementan el control de cambios

Actualmente existen gran variedad de herramientas que gestionan inventarios informáticos, cada una con características y funcionalidades comunes, tales como:

- Disminución en el coste de obtención de información.
- Los datos sobre el nivel de uso de las aplicaciones instaladas permiten al administrador del sistema, tomar decisiones fundamentadas sobre compras de licencias o renovación de mantenimientos.
- La detección temprana de configuraciones inadecuadas, antivirus sin actualizar o programas no autorizados, minimizan el impacto de estos potenciales huecos de seguridad en la red y asegura la conformidad del parque informático frente a posibles auditorías.
- Los usuarios del sistema ven como mejora la productividad del sistema informático, al minimizarse el número de incidencias y el tiempo medio de resolución de las mismas.

Estas herramientas proporcionan al usuario los datos necesarios sobre sus ordenadores.

ProactivaNET, *Everest Corporate Edition* y *Sistema de Inventario de Hardware y Software (SIHS)*, forman parte del gran grupo de herramientas que no solo gestionan inventarios sino que incluyen además una gestión de incidencias o control de cambios. A continuación serán expuestas algunas de sus características principales.

ProactivaNET

ProactivaNET permite conocer al instante y de manera sencilla el inventario de los equipos, sus licencias y configuración, simplificando enormemente la labor de administración. Incluye herramientas con las que se puede automatizar labores claves para la correcta administración de

la red de ordenadores.

Contiene una serie de características y funcionalidades tales como:

- Detección de versiones de antivirus no actualizadas.
- Informes de las licencias de software instalado y su nivel de uso. Detección de software no autorizado.
- Informes sobre versiones de librerías (dlls) y ejecutables.
- Consultas flexibles sobre la configuración del equipo y detección de claves de registro.
- Organización automática de los ordenadores en sus diferentes ubicaciones (por departamentos o localizaciones, siguiendo una estructura jerárquica).
- Registro de garantías y otros datos administrativos de cada equipo.
- Extremadamente ligero en la implantación, volumen de tráfico de datos y puesta en servicio.
- Flexibilidad total en la implantación, en redes distribuidas geográficamente, con conectividad de bajo ancho de banda a través del Transmission Control Protocol and Internet Protocol (TCP/IP) se adapta a cualquier tipología de red.
- Mínimamente intrusivo, únicamente se instala en el servidor y no hay módulos residentes en los ordenadores que puedan afectar su funcionamiento.
- Conectividad total, pues su base de datos en Structured Query Language (SQL) puede ser integrada con cualquier sistema de gestión [3].
- Brinda soporte y administración de sistemas.
- Puede integrarse con herramientas como Service Desk y Administración Remota. Service Desk facilita la gestión de incidencias, problemas, cambios y entregas desde el registro inicial hasta su cierre incorporando estándares internacionales de buenas prácticas como ITIL/ISO20000¹, formando entre ambas una herramienta básica para la optimización de cualquier servicio de Informática [4]. Mientras que Administración Remota permite evitar

1 Estándar para la gestión de Servicios de Tecnologías Informáticas

desplazamientos innecesarios usando la herramienta de control remoto; despliega parches y nuevas configuraciones a través de la distribución de software y administra remotamente su red a través de la transferencia de ficheros.

En la Figura 1 se representa la integración de Service Desk con ProactivaNET Inventario. La aplicación de Service Desk tiene comunicación bidireccional con los datos de ProactivaNET Inventario: de esta forma, es posible acceder a los datos de los equipos relacionados con la asistencia en curso, así como ver las asistencias que han sido relacionadas con un equipo concreto. A partir del usuario que reporta la incidencia, es posible obtener de forma inmediata su localización geográfica y el ordenador habitual del usuario produciéndose la integración con el inventario de red. Esta integración da la posibilidad de acceder directamente a todo el detalle del equipo o del software que genera la incidencia; los últimos cambios y alertas generados por ese equipo son accesibles desde la propia asistencia, lo que permite en muchos casos un diagnóstico rápido y remoto del problema, que asegura una resolución óptima y eficiente. Los datos administrativos más relevantes como listados de usuarios, localizaciones, etcétera. también son compartidos por ambas aplicaciones [5].



Figura 1: Integración de Service Desk-Inventario.

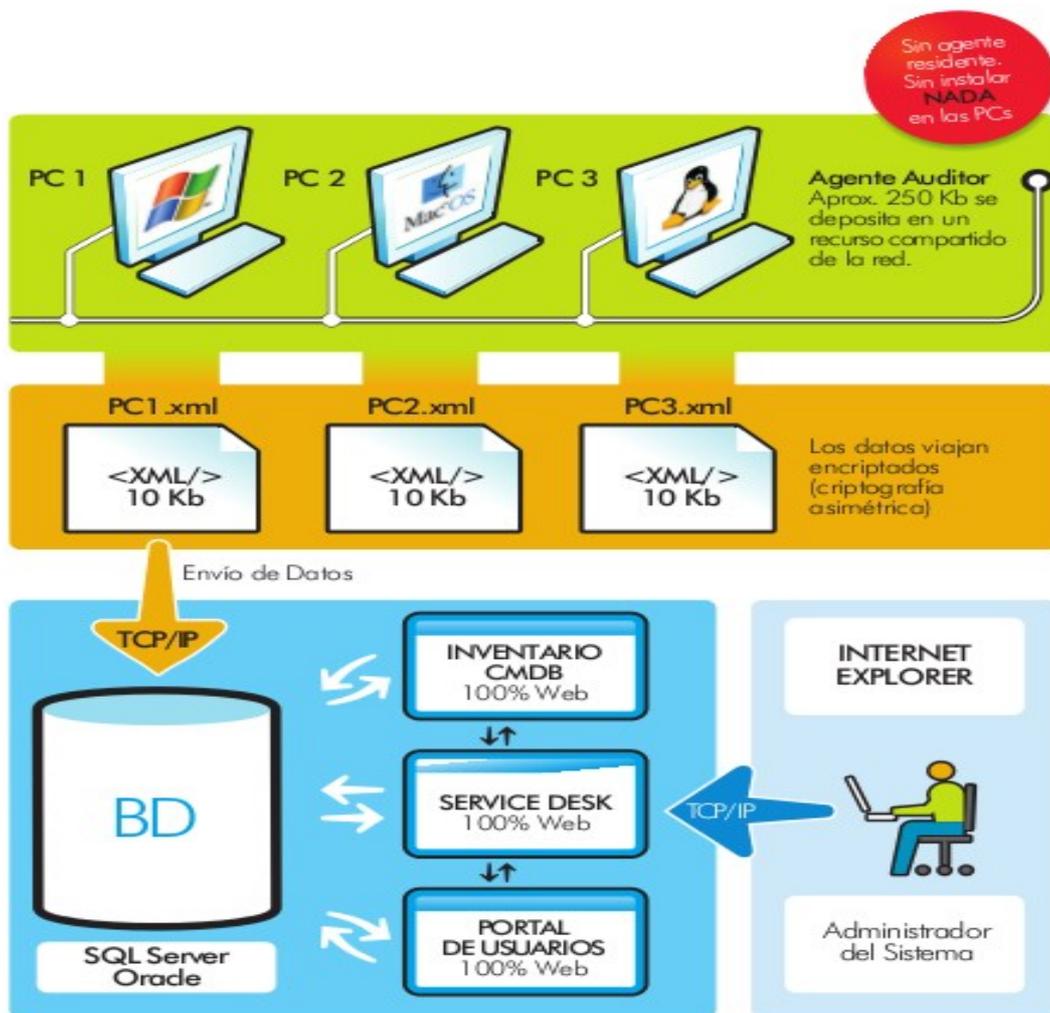


Figura 2: Esquema de componentes y conexiones internas de ProactivaNET.

En la Figura 2 se describen las conexiones y componentes de ProactivaNET. El sistema está conformado por agentes auditores (estos se depositan en un recurso compartido de la red) los cuales se encargan de auditar cada uno de los ordenadores de una red, una base de datos relacional donde se almacena toda la información del parque informático (los datos viajan encriptados en formato Extensible Markup Language (XML)), una aplicación web para explotar la información de la base de datos y un visor cliente para acceder a la aplicación web.

- Además con la integración con Service Desk, ProactivaNET provee una de las funcionalidades más importantes en un inventario de hardware y software: "Alertas sobre cambios en el parque informático o gestión de incidencias (hardware, software y configuración)".

Este módulo ayuda a organizar un servicio de soporte de alta profesionalidad y entre sus

funcionalidades y prácticas más destacadas se encuentran: integración con los datos del inventario de red y la Configuration Management Database (CMDB), un portal de usuarios para reportar las incidencias de manera directa [6].

En la Figura 3 se esquematiza el proceso de gestión de incidencias. Las incidencias pueden ser reportadas a través del portal de usuario que provee la herramienta, vía teléfono, correo electrónico o fax, las mismas son atendidas por técnicos capacitados en el tema. La nueva incidencia reportada transita por un proceso de identificación, registro, categorización, priorización, investigación y diagnóstico, resolución y recuperación y por último cierre.

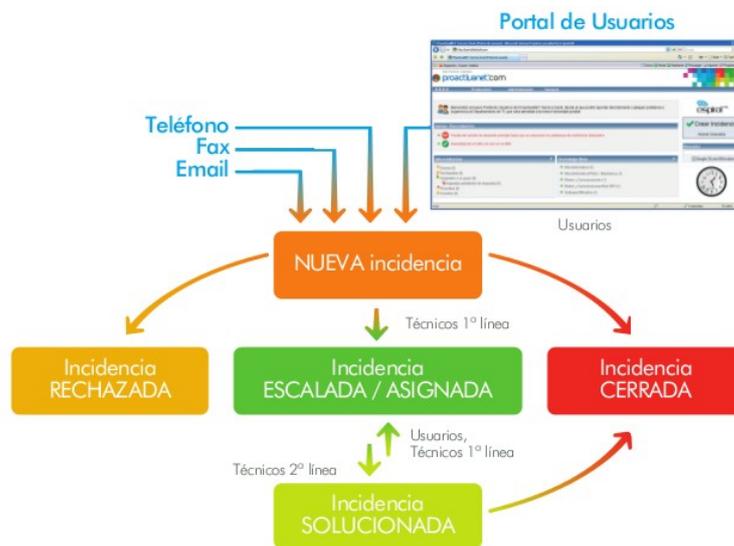


Figura 3: Módulo de Gestión de Incidencias.

En la Figura 4 se esquematiza el funcionamiento del módulo de Gestión de Problemas. Los problemas son causados, generalmente, por incidentes recurrentes de los que se desconocen las causas, o bien por incidentes aislados. La Gestión de Problemas se preocupa de encontrar y analizar las causas subyacentes que han provocado el incidente. La apertura de un problema, será similar a la de una incidencia, donde se deberá categorizar, priorizar, asignar un técnico e introducir una descripción para el problema. Se verifica si es conocido o no el error detectado, luego se hará uso de la Base de Conocimientos (KB) de Errores Conocidos, la cual permite el almacenamiento centralizado de errores conocidos y las resoluciones de incidencias y problemas. Propone Peticiones de Cambio (RFC) para que éstos sean implementados y una vez seleccionada la solución definitiva, ProactivaNET autorizaría la comunicación a Gestión de Cambios y el alta de las RFCs oportunas y es cerrado el problema.

razón, resultado, riesgos, recursos, responsables, relación y además se evalúa el impacto.

- Priorización y Riesgo: primero se usará el impacto que puede provocar (el impacto determinará la categoría del cambio: menor, medio o mayor). A continuación se le dará una prioridad al cambio.
- Análisis y cierre: El CAB realizará la Revisión Post Implementación (PIR), garantizando que el cambio ha alcanzado sus objetivos, que todos estén satisfechos y no haya efectos secundarios imprevistos, quedando todo bien documentado.

La integración de todos los módulos junto con la detección automática de cambios y alertas en su infraestructura, hace de ProactivaNET una fuente de información básica para el resto de procesos de Gestión del Servicio [9].

Everest Corporate Edition

Everest Corporate Edition es una herramienta que ejecuta una auditoría de red automática, control de cambios del sistema y solución de monitoreo de red para pequeñas y grandes empresas. Ofrece información precisa y capacidad de diagnóstico, tales como mejoras de memoria, monitoreo de hardware y una información de hardware a bajo nivel y de forma centralizada.

Esta herramienta brinda información acerca de:

Hardware:

- Motherboard y Central Processing Unit (CPU).
- Adaptadores de videos y monitor.
- Dispositivos de almacenamiento.
- Adaptadores de red, multimedia y dispositivos de entrada.
- Servicios Universal Serial Bus (USB).
- Software instalado.
- Alarmas por cambios de hardware o software.

Auditorías de red:

- Interfaz línea de comando.
- Auditoría de software.
- Administrador de auditoría.
- Administrador de base de datos.

Software:

- Sistema operativo.
- Servidor, usuarios, grupos.
- Red.
- Programas instalados.

Diagnósticos:

- Monitoreo de hardware.
- Pruebas de rendimiento.

Reportes:

- Asistente de reportes.
- Formatos de reportes.
- Reporte por correo electrónico e impresión.

Además utiliza las últimas tecnologías incluyendo XML y HyperText Markup Language (HTML) en la presentación de informes.

Al ofrecer varias instancias de auditoría permite a los clientes mantener un archivo de activos de información, y también ofrece una forma única para detectar con precisión cualquier cambio de software o hardware en la empresa [10].

SIHS

SIHS es un software de obtención de información de hardware y software en una red de computadoras, desarrollado en la UCI. El sistema está dividido en dos partes fundamentales, un servidor (recibe los inventarios enviados por los clientes y almacena la información en una base de datos donde se realizan una serie de pasos para analizar la información y dar una respuesta al inventario enviado) y un cliente o agente (destinado a la recolección de información del hardware, software de cualquier ordenador, así como la configuración de la computadora agente que es aquella en la que está instalado el agente o cliente del sistema de inventario de hardware y software). SIHS es compatible con los sistemas operativos GNU/Linux y Windows. Utiliza PostgreSQL como sistema gestor de base de datos y está programado en lenguaje Python.

El sistema tiene como proceso base Inventariar hardware y software (ver Figura 6). El mismo cuenta con un instalador (I1), código fuente (I2), el estado de la red (C1) debe estar activado, con un Cliente (M1) y un Servidor (M2), para poder inventariar (AO), teniendo como subprocesos acciones (O1), reportes de cambios (O2), reporte de estado real inventario (O3), y por último almacenar inventario (O4). Esta serie de subprocesos tienen como resultado final la creación de un inventario o alerta de correo electrónico ocasionada por la detección de incidencias sobre cualquier cambio efectuado dentro del inventario.

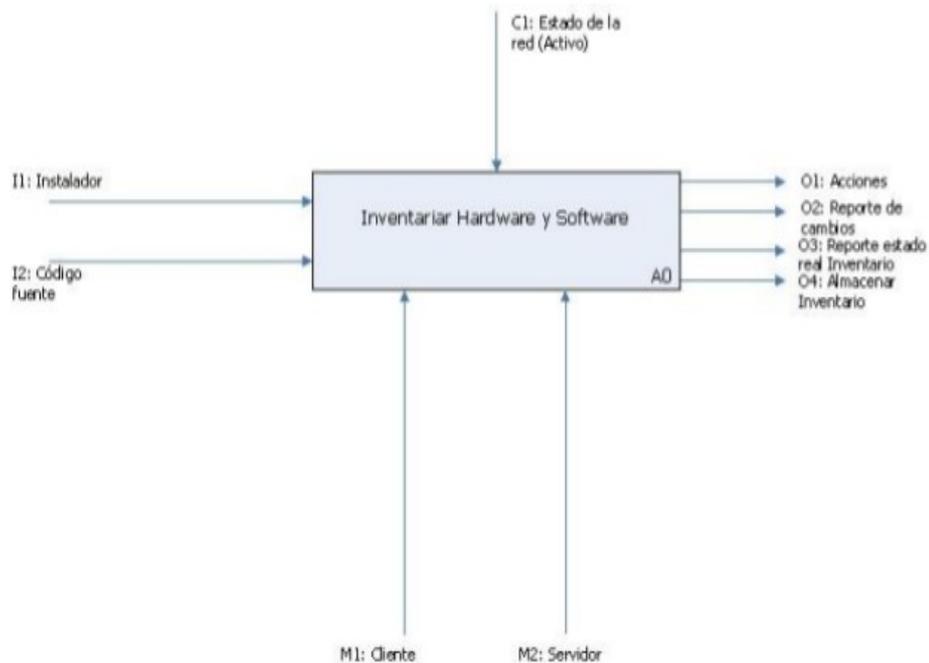


Figura 6: Proceso base Inventariar Hardware y Software.

El proceso base Inventariar hardware y software no es más que la intervención de una aplicación cliente y otra servidor. La cliente instalada en cualquier ordenador puede conocer los componentes del hardware y software que tiene instalado la computadora. Además puede identificar los cambios ocurridos a través de reportes y un sistema de alarmas que informa las violaciones a través de correo electrónico.

En la Figura 7 se muestra de forma más detallada este proceso:

Instalar cliente: consiste en instalar una aplicación cliente que obtiene toda la información del ordenador donde se instala para enviarla luego a la aplicación servidor.

Generar inventario: genera un inventario con toda la información de la máquina que posteriormente es enviado a una etapa de comprobación al sistema servidor.

Comprobar inventario: el cliente obtiene el inventario generado y procede a comprobarlo con el viejo inventario guardado en la cache, en caso de no coincidir se genera un reporte con los cambios.

Enviar Información: el cliente establece conexión con el servidor enviando el inventario o el reporte de los cambios el cual es almacenado en una base de datos y realiza una determinada acción en correspondencia con los datos obtenidos.

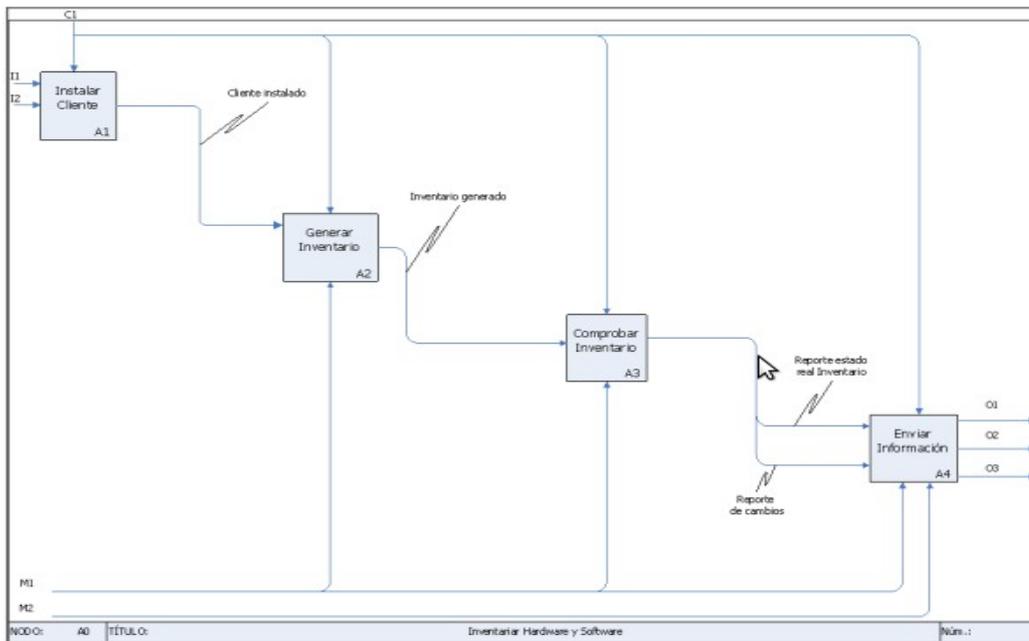


Figura 7: Modelado más detallado Inventariar Hardware y Software.

Además el sistema presenta un módulo de obtención de información (ver Figura 8) el cual consiste en ser una aplicación cliente-servidor. El objetivo principal de este módulo es centralizar en el servidor la información del hardware y software instalados en las estaciones clientes, dicha información es enviada al servidor y este determina si hubo cambios en el hardware o el software a partir del inventario obtenido anteriormente con el fin de que si existe algún cambio en el inventario, denominado *incidencia*, se ejecutará una determinada acción, pudiendo ser un correo electrónico, alertando al administrador de los cambios realizados.

Este módulo esta conformado por tres procesos descritos a continuación:

Comparar Inventario: Luego de que el cliente genere el inventario local, es comparado con el reporte viejo almacenado en la máquina Cliente, en caso de encontrar cambios este genera un Reporte de cambios.

Actualizar Inventario Local: Este proceso se ejecuta en caso de que se encuentren cambios, procediendo entonces a actualizar el inventario local.

Generar Reportes: Después de obtener el Inventario general o el Reporte con los cambios, el Cliente genera un inventario de cambios que es considerado como un reporte, que luego es enviado al servidor.

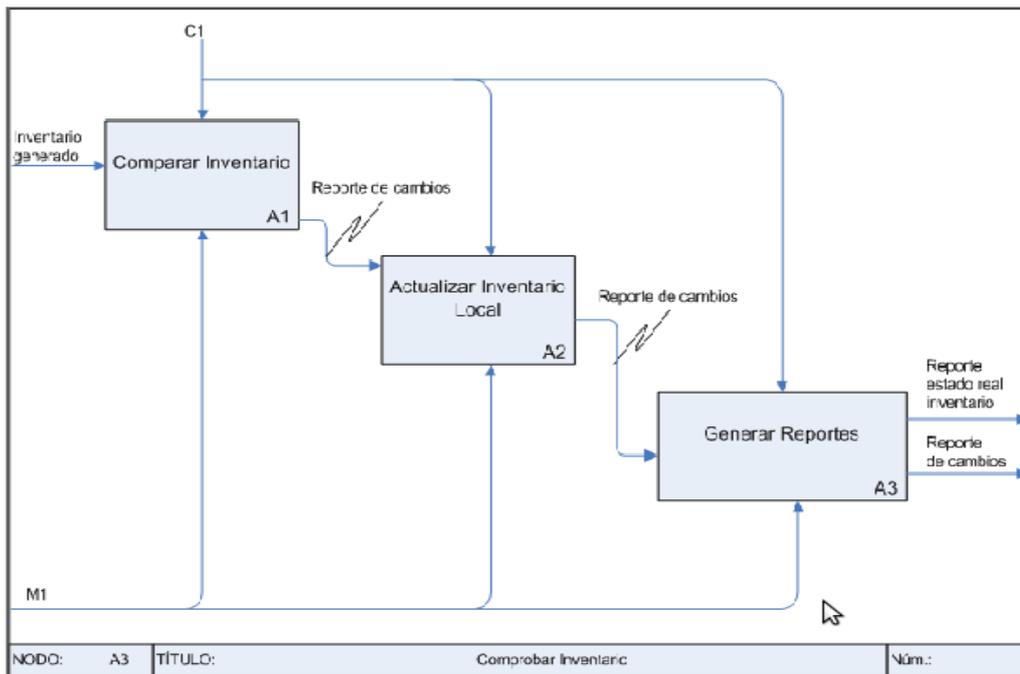


Figura 8: Proceso Comprobar Inventario.

También el módulo permite la organización de los activos inventariados por categorías que responden a las necesidades del administrador de red, alertas de incidencias, gestión de períodos de cambios autorizados donde el administrador podrá autorizar en una hora y fecha determinada cambios sobre el periféricos específicos para que los técnicos o personas responsables del mantenimiento de los ordenadores puedan realizar cambios de hardware sobre estos sin que el sistema los detecte como incidencias, gestión de reportes donde por determinados criterios de búsqueda los usuarios del sistema podrán obtener información, guardarlas como reportes y posteriormente exportarlas a formato PDF [11].

1.4 Valoración del estudio realizado

Luego de estudiar estos tres sistemas de gestión de inventarios y analizar las funcionalidades que proveen, enfocando el análisis en la gestión de incidencias y control de cambios, arrojó como resultado final, que este proceso en las tres herramientas no está de acorde con las características de OCS Inventory NG pues difieren en cuanto a arquitectura y modo de funcionamiento a pesar de que comparten la funcionalidad principal de todo sistema de gestión de inventarios. Además tanto ProactivaNET como Everest Corporate Edition son herramientas que trabajan sobre Windows, siendo esto el principal impedimento de su instalación en el proyecto durante la etapa de estudio y análisis de las mismas.

En los tres sistemas el proceso de control de cambios es bastante complejo. ProactivaNET para gestionar incidencias necesita integrarse con otras herramientas (Service Desk), pues la misma

utiliza los módulos Gestión de Incidencias, Gestión de Problemas y Gestión de Cambios y Entregas (ver Epígrafe 1.2) para hacer del inventario un paquete más completo. En Evesrest Corporate Edition la documentación era muy escasa y solo se tomó en cuenta la estructura de la información a visualizar, mientras que en SIHS el control de cambios es atendido por el módulo Obtención de Información que al igual que en ProactivaNET tiene dependencias de otros procesos como es el caso del proceso base Inventariar Hardware y Software del cual se derivan una serie de subprocesos (ver Epígrafe 1.2), convirtiéndose en algo complejo para su utilización en OCS Inventory NG. De igual forma, el análisis brindó información de posibles soluciones para la implementación del módulo de control de cambios en OCS Inventory NG.

1.5 Metodología de desarrollo de software

Una metodología de desarrollo de software no es más que un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Actualmente existen muchas metodologías de desarrollo entre ellas las llamadas metodologías tradicionales que prácticamente son obsoletas pues las mismas producen documentación que no fluye, en vez de un sistema de trabajo que cubra las necesidades del negocio y otras más actuales como son las metodologías modernas dentro de las cuales podemos encontrar las metodologías ágiles que surgen como alternativa a las metodologías tradicionales. Las mismas dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo iterativo e incremental del software pero con iteraciones muy cortas. Este enfoque muestra su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad. Es aplicable en proyectos de corta duración.

Se selecciona la metodología ágil de desarrollo SXP, por ser la definida por el proyecto en el desarrollo de sistemas similares.

1.5.1 Metodologías ágiles

SXP

SXP es un híbrido de metodologías ágiles que toma las mejores prácticas de las metodologías SCRUM y XP además de regirse por los lineamientos de calidad de la UCI. Logrando una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos. Teniendo como premisa una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Consta de 4 fases principales:

- Planificación-Definición: se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo: es donde se realiza la implementación del sistema hasta que este listo para ser entregado.
- Entrega: se entrega el producto (puesta en marcha).
- Mantenimiento: es donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto (LRP), definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añada una nueva funcionalidad.

Es una metodología que se caracteriza por ser iterativa e incremental con pequeñas mejoras unas tras otras, basada en Historia de Usuario (HU), está atenta al cambio y permite que el equipo de programación se mantenga en frecuente interacción con el cliente o usuario. Esta especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo [12].

Luego de determinada, entendida y analizada la metodología de desarrollo, se hace necesario definir las tecnologías, herramientas y lenguajes de programación que apoyarán el proceso de desarrollo del sistema.

1.6 Lenguajes, tecnologías y herramientas de desarrollo utilizadas

Los lenguajes, tecnologías y herramientas seleccionadas para lograr un desarrollo correcto del sistema, con el fin de obtener un producto adecuado a las exigencias del cliente son las siguientes:

1.6.1 Lenguajes

Los lenguajes de programación son herramientas que nos permiten crear programas y software. Facilitan la tarea de programación y representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona.

A continuación se muestran los lenguajes de programación seleccionados:

HTML

Es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. La mayoría de los documentos tienen estructuras comunes (títulos, párrafos, listas, etcétera) que van a ser definidas por este lenguaje mediante tags (etiquetas). Cualquier cosa que no sea una tag es parte del documento mismo. Este lenguaje no describe la apariencia del diseño de un documento sino que ofrece a cada plataforma que le de formato según su capacidad y la de su navegador (tamaño de la pantalla, fuentes que tiene instaladas, etcétera). HTML tiene dos ventajas que lo hacen prácticamente imprescindible a la hora de diseñar una presentación web: su compatibilidad y su facilidad de aprendizaje debido al reducido número de tags que usa. HTML tiene dos ventajas que lo hacen prácticamente imprescindible a la hora de diseñar una presentación web: Su compatibilidad y su facilidad de aprendizaje debido al reducido número de tags que usa [13]. El entorno para trabajar HTML es simplemente un procesador de texto. El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html. Estos documentos pueden ser mostrados por los navegadores o "browsers" de páginas Web en Internet.

Hypertext Pre-processor (PHP)

Es un lenguaje de programación interpretado de libre distribución, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. Permite la autenticación HTTP con PHP, manejo de sesiones, manejo de conexiones, entre otros. Algunas de las funciones que provee se encuentran: manipulación de formatos de audio, extensiones relacionadas con fecha y hora, extensiones de base de datos, procesado generación de imágenes, extensiones relacionadas con correo electrónico.

Por la amplia y activa comunidad de desarrolladores que existe es posible encontrar gran cantidad de documentación y ejemplos de su desarrollo en la web. Este lenguaje tiene como ventajas que

es multiplataforma y fácil de aprender.

Cascading Style Sheets (CSS)

Es un mecanismo simple que describe como se va a mostrar un documento en la pantalla, o como se va a imprimir, o incluso como va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos [14]. 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.

Perl

Es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que actualmente es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo web, programación en red, desarrollo de interfaz gráfica de usuario y más. Entre sus principales características tenemos que es muy fácil de usar, soporta tanto la programación estructurada como la programación orientada a objeto y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles. Perl es software libre y está licenciado bajo la Licencia Artística y la GNU General Public License. Está actualmente soportado por la mayoría de las plataformas existentes [15].

JavaScript

Es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta 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 podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación

podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica [16].

Unified Modeling Language (UML)

Es el lenguaje de modelado de sistemas de software. Proporciona una base importante para el modelo impulsado por el Object Management Group (OMG) de Arquitectura , que unifica todas las etapas de desarrollo e integración de modelos de negocio, a través de modelos arquitectónicos y la aplicación, para el desarrollo, implementación, mantenimiento y la evolución. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software [17].

1.6.2 Tecnologías

Asynchronous JavaScript And XML (AJAX)

Es un lenguaje permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano. Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor [18].

Sistema Gestor de Base Datos (SGBD)

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. El propósito general de los mismos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los mismos.

Entre las principales funciones que debe realizar un SGBD se encuentran la definición de los datos, manipulación de los datos, garantizar la seguridad e integridad de los datos y la gestión de las transacciones así como el acceso concurrente [19].

La aplicación OCS Inventory NG trabaja con el gestor de base de datos **MySQL Server** en su versión 5.1.

MySQL se ha convertido en la base de datos de código abierto más popular debido a su alto rendimiento, alta fiabilidad y facilidad de uso. También es la base de datos de elección para una

nueva generación de aplicaciones basadas en la pila LAMP (Linux, Apache, MySQL, PHP / Perl / Python). Muchos de los más grandes y las organizaciones de más rápido crecimiento del mundo, incluyendo Facebook, Google, Adobe, Alcatel Lucent y Zappos se basan en MySQL para ahorrar tiempo y dinero en sus grandes volúmenes de sitios Web, los sistemas críticos de negocio y paquetes de software.

Entre sus principales características se encuentran:

- Está escrito en los lenguajes C y en C++.
- Probado con un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas (Linux, MAC OS X, Sun Solaris, Windows).
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales, posibilitando la adaptación del Sistema Gestor de Base de Datos a casos concretos.
- Relativamente sencillo de añadir otro sistema de almacenamiento.
- Un sistema de reserva de memoria muy rápido basado en hilos (threads).
- Velocidad de respuesta destacable.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Administración basada en un esquema de usuarios y privilegios.
- Las funciones SQL están implementadas usando una librería altamente optimizada.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor, además de estar disponible como biblioteca y puede ser incrustado en aplicaciones independientes los que pueden usarse por sí mismas o en entornos donde no hay red disponible.
- Las opciones de conexión abarcan TCP/IP, sockets y el estándar de acceso Open Database Connectivity (ODBC).
- Soporta los diversos tipos de columnas del estándar SQL.

MySQL en operaciones de lectura sobre las bases de datos es mucho más rápido y constituye la

mejor solución para las aplicaciones web dinámicas donde la rapidez en la presentación de la información es crucial [20].

Servidores Web

Un servidor web o servidor HTTP es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales, síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones. Entre los más populares, importantes y usados se encuentran Apache, Internet Information Services (IIS) y Sun Java System Web Server.

Debido a las características que presenta el sistema OCS Inventory NG y las características del entorno de desarrollo del mismo, Apache se convierte en el mejor candidato para ser usado como servidor web. Es el servidor web por excelencia para sistemas operativos GNU/Linux, y constituye la alternativa libre en el hospedaje de aplicaciones para la web. A continuación se presenta un pequeño resumen sobre este servidor con sus características más importantes:

Apache

El servidor Apache surge a partir de un servidor web del National Center for Supercomputing Applications (NCSA) de la Universidad de Illinois como corrección a errores y mejoras importantes al producto inicial. Por los resultados logrados, es fundado el Grupo Apache, potenciador del nuevo producto al que se le hicieron mejoras en su arquitectura logrando modularidad. En 1999 los desarrolladores fundan la Apache Software Foundation, organización que da soporte legal al desarrollo de Apache [21]. Entre sus principales características están:

- Soporte para el protocolo HTTP 1.1.
- Soporte para Common Gateway Interface (CGI).
- Soporte para FastCGI.
- Soporte de Host Virtuales.
- Soporte de autenticación HTTP.
- Perl integrado.

- Soporte de scripts PHP.
- Soporte de servlets de Java.
- Servidor proxy integrado.
- Estado del servidor y adaptación de registros.
- Soporte de Server Side Include (SSI).
- Soporte de Secured Socket Layer (SSL).
- Módulos multiproceso.
- Portable en tiempo de ejecución [22].
- Código abierto.

1.6.3 Herramientas

Las herramientas seleccionadas para sustentar el proceso de desarrollo del sistema son las siguientes:

Firebug

Es una extensión de Firefox creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura Document Object Model (DOM)), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea e inline.

Web Developer

Es un complemento del navegador web Firefox que nos añade un menú y una barra de herramientas con las que podremos, habilitar, validar, desactivar o editar, CSS, JavaScript, formularios, cookies, enlaces.que contenga la página que estamos visualizando. La extensión está disponible para Firefox y Chrome, y se ejecutará en cualquier plataforma que soporte estos navegadores incluyendo Windows, Mac OS X y Linux.

Netbeans

Netbeans es un Integrated Development Environment (IDE) gratuito de código abierto, con soporte para PHP y características visuales para el desarrollo web, el cual se integra muy bien con sistemas de control de versiones y sistema gestores de bases de datos como MySQL y PostgreSQL y funciona sobre plataformas como Linux, MAC OS X, Solaris y Windows.

PHPMysqladmin

Es una herramienta libre escrita en PHP con el objetivo de administrar servidores MySQL a través de la web. Soporta un amplio rango de operaciones sobre las bases de datos, tales como la administración de bases de datos, tablas, registros, usuarios y permisos, ejecución de consultas SQL e importación y exportación de datos en múltiples formatos. Posee una amplia documentación.

Visual Paradigm for UML

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Es una herramienta Computer Aided Software Engineering (CASE) potente y fácil de utilizar, que permite el modelado visual UML, propiciando la rápida construcción de aplicaciones de calidad. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación, además es una herramienta colaborativa, pues soporta múltiples usuarios trabajando sobre el mismo proyecto.

Entre sus características más importantes sobresalen:

- Soporte para varias versiones de UML.
- Brinda una versión gratuita y otra comercial.
- Permite la ingeniería inversa (código a modelo, código a diagrama).
- Permite la generación de código (modelo a código, diagrama a código).
- Permite exportar el diseño de la base de datos para varios gestores.
- Es un producto multiplataforma.

RapidSVN

RapidSVN es un cliente de interfaz gráfica para la comunicación con servidores Subversion. Está escrito en C++ y distribuido bajo licencia GPL. Este facilita el versionado de ficheros, desde una

interfaz sencilla e intuitiva y se encuentra disponible para plataformas Windows, Linux, MAC OS X y Solaris. Es una herramienta rápida y eficiente.

Geany

Geany es un editor de texto ligero con características básicas de entorno de desarrollo integrado (IDE). Está disponible para distintos sistemas operativos, como GNU/Linux, Mac OS X, Solaris y Microsoft Windows. Es distribuido como software libre bajo la Licencia Pública General de GNU. Tiene soporte para muchos lenguajes de programación distintos, como C, C++, Java, JavaScript, PHP, HTML, CSS, Python, Perl, Ruby, Fortran, Pascal y Haskell. Algunas de las características más destacadas de Geany son: autocompletado, soporte multidocumento, soporte de proyectos, coloreado de sintaxis y emulador de terminal incrustado.

Conclusiones parciales

En el desarrollo de este capítulo se trataron conceptos generales acerca de los sistemas de gestión de inventarios con el propósito de ubicar al lector en el contexto del problema. Fueron analizados algunos sistemas de inventarios con las características y funcionalidades que proveen con el fin de encontrar funcionalidades que ayuden a la personalización de la herramienta centrada principalmente en el control de cambios. Se abordaron elementos de la metodología SXP que guiará el proceso de desarrollo de software, así como también fueron expuestas las principales características de las tecnologías y lenguajes que serán usados y se definieron además un conjunto de herramientas de soporte al ciclo de desarrollo.

Capítulo 2: GITI, personalización de la herramienta OCS Inventory NG.

En este capítulo se presentan las características del sistema y se inicia el desarrollo de la solución propuesta guiada por la metodología de desarrollo SXP. Se realiza la captura de requisitos del sistema, actividad de vital importancia para la implementación del módulo de control de cambios. También se definen una serie de tareas asociadas al desarrollo de la aplicación quedando documentadas en historias de usuarios, tareas de ingenierías, plan de release.

2.1 Características funcionales del sistema

OCS Inventory NG es una herramienta de gestión de inventarios de libre distribución que permite a los usuarios administrar el inventario de sus activos de tecnología informática. Recopila información acerca del software y hardware de equipos en una red. Esta puede utilizarse para visualizar y realizar consultas del inventario a través de una interfaz web y es capaz de detectar dispositivos de la red tales como switch, routers e impresoras.

Al ejecutar la consola de administración bajo el sistema operativo GNU/Linux, si se encuentran disponibles nmap² y nmblookup³, permite escanear una dirección IP o una subred para así obtener la información detallada de los equipos no inventariados. Además incluye la capacidad de distribuir o desplegar paquetes en los equipos clientes.

Arquitectura de OCS Inventory NG

OCS Inventory NG utiliza un agente que ejecuta el inventario en los equipos cliente, un servidor de administración, que centraliza los resultados del inventario, permitiendo su visualización y la creación de paquetes de distribución y una interfaz web. A continuación se realiza una descripción más completa de estos elementos:

- Agentes: Se utilizan para recoger el máximo de la información posible. Son programas que se instalan en los equipos a inventariar, que periódicamente conectan con el servidor de comunicación enviando el inventario en formato XML. Existen clientes para varios sistemas operativos como Windows, GNU/Linux, Mac Os X, Sun Solaris, IBM AIX.

2 Herramienta de código abierto para exploración de red y auditorías de seguridad.

3 Resuelve los nombres NetBIOS (Network Basic Input/Output System) en direcciones IP. Difunde su consulta en la subred local hasta que la máquina objetivo contesta.

- Servidor: El servidor de administración utiliza como servidor web Apache, como gestor de base de datos MySQL y como lenguaje de programación Perl. Es multiplataforma gracias a su simple diseño y el uso de la librería mod_perl.
- Interfaz web: Una interfaz web opcional escrita en PHP que ofrece servicios complementarios.

Las comunicaciones entre los agentes y el servidor de administración se realiza mediante los protocolos HTTP/HTTPS. Todos los datos tienen formato XML para reducir el promedio de tráfico de la red. Los agentes pueden ser instalados en los equipos cliente. Bajo el sistema operativo Linux, el agente debe ser instalado manualmente.

Presenta un servidor de administración, que está conformado por 4 componentes principales:

- Servidor de base de datos; es el encargado de almacenar la información de los inventarios. Puede ser MySQL 4.1 o superior.
- Servidor de comunicaciones: se encarga de las comunicaciones mediante el protocolo de comunicación HTTP entre los agentes y el servidor de base de datos. Éste necesita Apache Web Server en sus versiones 1.3.X/2.X, está desarrollado en PERL como módulo del Apache, permitiendo un mayor rendimiento, pues los scripts PERL son compilados cuando inicia Apache.
- Servidor de distribución: se encarga de almacenar toda la configuración de la distribución de paquetes, para esto se requiere HTTPS. Necesita cualquier servidor web con SSL habilitado.
- Consola de administración: permite a los administradores consultar el servidor de base de datos a través de cualquier navegador web. Está desarrollada en PHP 4.1 o superior y es ejecutada en el servidor web Apache. También requiere soporte ZIP y GD (librería gráfica que utiliza php5) para usar la distribución de software.

Estos 4 componentes se pueden alojar en un solo equipo o en diferentes equipos para permitir el equilibrio de carga. Para más de 10000 equipos inventariados, es mejor utilizar al menos 2 servidores diferentes, 1 para el servidor de base de datos y servidor de comunicación y otro para una réplica de la base de datos, el servidor de administración y servidor de distribución (ver Figura 9).

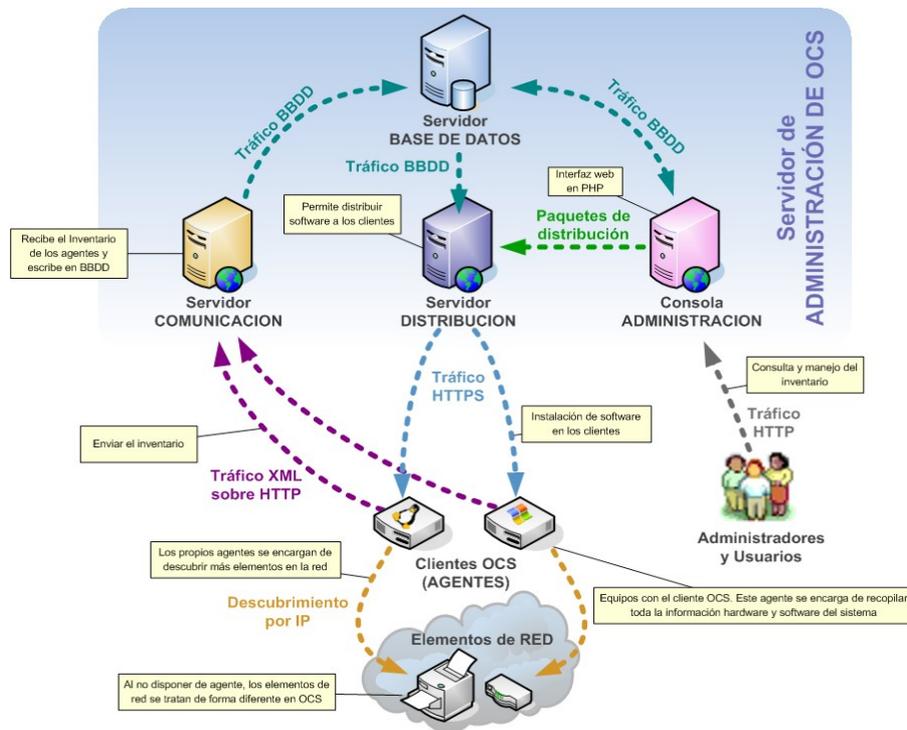


Figura 9: Arquitectura de OCS Inventory NG.

OCS Inventory NG soporta casi todas las plataformas disponibles, tales como Linux, Windows, Mac Os X, Sun, IBM, AIX. De su integración con la aplicación Gestión Libre del Parque Informático (GLPI) se obtiene una herramienta muy potente para la administración de todos los recursos de la red, permitiendo así a un administrador llevar un seguimiento muy completo a su red, y evitando la pérdida de componentes de hardware del equipo monitorizado o daños en los mismos.

Se ha podido apreciar OCS inventory NG es una herramienta que provee varias funcionalidades similares a las herramientas analizadas en el capítulo 1, exceptuando el control de cambios. Conociendo la estructura que presenta la herramienta se procede a presentar la propuesta de solución.

2.2 Descripción de la propuesta de solución

Como se ha venido abordando hasta ahora en los contenidos expuestos, que al sistema se le hace necesario la implementación de un módulo de control de cambios, con el fin de que notifique cada vez que exista un cambio tras haber realizado varios inventarios en una misma estación de trabajo. Además al sistema se le incorporarán otras funcionalidades para hacer de la herramienta lo más completa y ajustada a las exigencias del cliente para su posterior uso en el proceso de migración a SWL.

Como propuesta de solución, primeramente se propone un cambio de identidad para el sistema,

pasando el mismo a llamarse Gestión de Inventarios de Tecnología Informática (GITI), además se le implementarán las siguientes funcionalidades:

1. Módulo Control de cambios (incluye las siguientes funcionalidades):

- Insertar Notificación.
- Modificar Notificación.
- Mostrar Notificación.
- Eliminar Notificación.

2. Dar baja a un equipo.

3. Módulo Gestionar inventario manual (incluye las siguientes funcionalidades):

- Insertar Inventario.
- Modificar Inventario.
- Mostrar Inventario.
- Eliminar Inventario.

4. Módulo Reportes en PDF (permite generar reportes de toda la información en formato pdf).

5. Módulo Gestionar envío de correo electrónico (incluye las siguientes funcionalidades):

- Configurar correo electrónico.
- Enviar correo electrónicos.

2.2.1 Planificación del proyecto por roles

Rol	Responsabilidad	Nombre y Apellidos
Gerente	Es el responsable de tomar las decisiones finales, participa en la definición de objetivos y requerimientos. Tiene la responsabilidad de	Mario Lázaro García Pérez

	controlar el progreso del software	
Cliente	Asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración, participa en la concepción inicial del sistema. Contribuye a definir las historias de usuario y los casos de prueba de aceptación.	Ing. Yoandy Pérez Villazón
Miembros del Equipo		
Programador	Define las tareas de ingeniería. Produce el código del sistema. Selecciona el estándar de programación. Confecciona los Manuales de usuario y de desarrollo.	Mario Lázaro García Pérez Diana Lilian Antúnez Ginarte
Analista	Escribe la concepción del sistema y las historias de usuario. Crea el Modelo de historia de usuario del negocio y la LRP. Asigna la prioridad a las historias de usuario.	Diana Lilian Antúnez Ginarte
Tester	Escribe los casos de prueba de aceptación. Ejecuta las pruebas, es responsable de las herramientas de soporte para pruebas.	Mario Lázaro García Pérez Diana Lilian Antúnez Ginarte
Arquitecto	Se vincula directamente con el analista y el diseñador, tiene que ver con la estructura y el diseño del sistema.	Mario Lázaro García Pérez

2.3 Captura de Requisitos

Para desarrollar el sistema como próximo paso se realizó la captura de requisitos. Para ello se partió de las entrevistas realizadas con el cliente, de las cuales se obtuvo una descripción del sistema, generándose de esta la Lista de Reserva del Producto (LRP).

2.3.1 LRP

Es una lista priorizada que define el trabajo que se realiza en el proyecto. Esta puede crecer y modificarse a medida que aumenta el proyecto teniendo como restricción que solo puede ser

cambiada entre Sprint (Iteración) con el fin de que el proyecto final sea más concreto, útil y competitivo posible.

A continuación se presenta la LRP de la solución propuesta:

RF (Requisitos Funcionales)				
Asignado a	Ítem *	Descripción	Estimación	Estimado por
Sistema: Gestión de inventarios OCS Inventory NG				
Prioridad		Alta		
Mario Lázaro García Pérez	1	Insertar notificaciones.	1 día	Diana Lilian Antúnez Ginarte
Mario Lázaro García Pérez	2	Eliminar notificaciones.	5 horas	Diana Lilian Antúnez Ginarte
Mario Lázaro García Pérez	3	Mostrar notificaciones.	8 horas	Diana Lilian Antúnez Ginarte
Mario Lázaro García Pérez	4	Modificar notificaciones.	2 días	Diana Lilian Antúnez Ginarte
Prioridad		Media		
Diana Lilian Antúnez Ginarte	5	Dar de baja a un equipo.	1 día	Diana Lilian Antúnez Ginarte
Mario Lázaro García Pérez	6	Insertar inventario manual.	1día	Diana Lilian Antúnez Ginarte
Mario Lázaro García Pérez	7	Modificar inventario manual.	2días	Diana Lilian Antúnez Ginarte
Mario Lázaro García Pérez	8	Eliminar inventario manual.	1día	Diana Lilian Antúnez Ginarte
Mario Lázaro García Pérez	9	Mostrar inventario manual.	1día	Diana Lilian Antúnez Ginarte
Prioridad		Baja		

Diana Lilian Antúnez Ginarte	10	Configurar servidor saliente de correo electrónico.	1 día	Diana Lilian Antúnez Ginarte
Diana Lilian Antúnez Ginarte	12	Enviar correo electrónico.	1 día	Diana Lilian Antúnez Ginarte
Diana Lilian Antúnez Ginarte	13	Reportes en PDF	1 día	Diana Lilian Antúnez Ginarte
RNF (Requisitos No Funcionales)				
	14	Todo deberá ser desarrollado en lenguaje PHP, Perl.		
	15	Utilizar como servidor web Apache y servidor de bases de datos MySQL Server.		
	16	El sistema debe responder en un tiempo relativamente rápido a las peticiones del usuario (menos de 5 segundos)		
	17	El sistema debe garantizar la seguridad a través de la autenticación de los usuarios que administren la herramienta.		
	18	El acceso a las funcionalidades y el contenido publicado estará protegido por permisos de acceso según roles definidos en el sistema.		

	19	Debe presentar y mantener una interfaz amigable, interactiva e intuitiva con el cliente.		
	20	Para el análisis y el diseño del sistema debe ser utilizada la metodología SXP.		
	21	Se entregará al cliente el instalador y un manual de instalación.		

A partir de tener definida la LRP como próximo paso se realiza una especificación de requisitos, para ello se hace uso de la Historia de Usuario, escritas por los clientes pero siempre con la ayuda de los analistas del sistema, rol principal en esta tarea.

2.4 Historia de Usuario (HU)

Son técnicas utilizadas para especificar los requisitos del software. Las mismas son escritas por los clientes como tareas que el sistema debe hacer. Su construcción depende principalmente de la habilidad que se tenga para definir las. Sirven también para estimar tiempos de desarrollo y son la base para las pruebas funcionales.

2.4.1 HU Gestionar notificaciones

Historia de Usuario	
Número: 01	Nombre Historia de Usuario: Gestionar notificaciones
Modificación de Historia de Usuario Número: 1	
Usuario: Mario Lázaro García Pérez	Iteración Asignada: 3
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 3

Descripción:

Cuando se realiza un nuevo inventario a una misma estación de trabajo el sistema debe notificar si existe algún cambio tanto en términos de hardware como de software con respecto al inventario anterior. Estas notificaciones las genera el sistema y las envía vía correo electrónico al equipo de migración que labora en la entidad en cuestión. Además se podrá modificar, eliminar y visualizar las notificaciones.

Observaciones:

a) Se visualizarán por fecha, mostrando además los siguientes atributos:

1. Fecha.
2. ID.
3. Computador.
4. Componente.
5. Evento(Añadido/Modificado/Eliminado).
6. Campo.
7. Valor anterior.
8. Valor actual.
9. Leída.
10. Borrar.
11. Editar.
12. Seleccionar.

b) Para modificar alguna notificación seleccionada se ajustarían el atributo: Leída. Además solo se puede modificar una sola notificación a la vez.

c) Se pueden eliminar varias notificaciones a la vez o seleccionar solo una.

Prototipos de interfaz:

Mostrar: ▼

Restringir vista: ▼

Adicionar columna: ▼

2 Resultados

Fecha	Computador	Componente	Evento	Campo	Valor anterior	Valor actual	Borrar	Editar	Seleccionar
2011-04-04 21:54:10	pc-12	/dev/sda4	Modificado	FREE	6413	6385			<input type="checkbox"/>
2011-04-04 21:54:10	pc-12	firefox	Modificado	VERSION	3.6	4.01			<input type="checkbox"/>

Figura 10: Mostrar notificaciones.

Modificar notificación

ID: 1

Fecha: 2011-04-04 21:54:10

Componente: acpi-support

Evento: Modificado

Leída: ▼




Figura 11: Modificar notificaciones.

Fecha	Computador	Componente	Evento	Campo	Valor anterior	Valor actual	Borrar	Editar	Seleccionar
2011-04-04 21:54:10	pc-12	/dev/sda4	Modificado	FREE	6413	6385			<input type="checkbox"/>
2011-04-04 21:54:10	pc-12	firefox	Modificado	VERSION	3.6	4.01			<input type="checkbox"/>



Figura 12: Eliminar notificaciones.

2.4.2 HU Dar de baja a un equipo

Historia de Usuario	
Número: 02	Nombre Historia de Usuario: Dar de baja a un equipo.
Modificación de Historia de Usuario Número: 1	
Usuario: Diana Lilian Antúñez Ginarte	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 2
Descripción: Se dará baja al equipo que no se desee que forme parte del inventario real.	
Observaciones: Permitirá dar baja a uno o varios computadores.	
Prototipo de interfaz:	

TAG	Último inventario	Computador	Nombre usuario	Sistema Operativo	RAM(MB)	CPU(MHz)	Borrar	Seleccionar
NA	2011-04-04 21:54:10	pc-12	mlgarcia	Ubuntu 10.10	976	2200	X	<input type="checkbox"/>
NA	2011-04-05 11:37:00	pc-2	dlantunez	Ubuntu 10.04	976	2200	X	<input type="checkbox"/>



Figura 13: Dar de baja.

2.4.3 HU Gestionar envío de correo electrónico

Historia de Usuario	
Número: 03	Nombre Historia de Usuario: Gestionar envío de correo electrónico
Modificación de Historia de Usuario Número: 1	
Usuario: Diana Lilian Antúnez Ginarte	Iteración Asignada: 4
Prioridad en Negocio: Baja	Puntos Estimados: 2
Riesgo en Desarrollo: Media	Puntos Reales: 2
Descripción: Las notificaciones se enviarán vía correo electrónico al administrador del sistema o a algún miembro del equipo de migración en cuanto se detecte un cambio en el inventario.	
Observaciones: a) Se configura el servidor saliente de correo electrónico con los siguientes atributos: <ol style="list-style-type: none"> 1. Dirección del servidor SMTP. 2. Puertos de conexión del servidor. 3. Nombre del usuario para la conexión al servidor SMTP. 4. Contraseña para la conexión al servidor SMTP. b) Se configura el o los destinatarios con los siguientes atributos: <ol style="list-style-type: none"> 1. Tipo. 2. Grupo. 3. Nombre. 4. Apellido. 5. Correo electrónico. 6. Recibir notificación. 7. Comentarios. 	

8. Contraseña.

c) Se envía correo electrónico en cuanto el sistema detecte algún cambio en el reporte ya establecido.

Prototipos de interfaz:

The screenshot shows a configuration form for an SMTP server. It consists of four rows, each with a label and a description on the left, and a text input field on the right. The labels and descriptions are: SMTP_SERVER (Dirección del servidor SMTP), SMTP_PORT (Puerto de conexión del servidor SMTP), SMTP_USER (Nombre de usuario para la conexión al servidor SMTP), and SMTP_PASSWD (Contraseña para la conexión al servidor SMTP). The input fields contain the values: smtp.uci.cu, 25, mlgarcia@estudiantes.uci.cu, and a masked password (*****). Below the input fields is a button labeled 'Actualizar'.

Figura 14: Configurar servidor de correo electrónico.

The screenshot shows a user configuration form. It has several fields: ID usuario (mlgarcia), Tipo (Súper administradores), Grupo (a dropdown menu with a plus sign), Nombre (Mario Lázaro), Apellido (García Pérez), Correo electrónico (mlgarcia@estudiantes.uci.cu), Recibir notificación (checked checkbox), Comentarios (Cuenta de súper administrador), and Contraseña (masked with *****). At the bottom of the form are two circular icons: a green checkmark and a red X.

Figura 15: Configurar destinatario.

2.4.4 HU Gestionar inventario manual

Historia de Usuario	
Número: 04	Nombre Historia de Usuario: Gestionar inventario manual

Modificación de Historia de Usuario Número: 1	
Usuario: Mario Lázaro García Pérez	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 4
Descripción: Se entrarán los datos para el inventario de forma manual, así como también se podrá modificar, eliminar y visualizar los mismos.	
Observaciones:	
Prototipo de interfaz:	
Computador	
ID:	2
Nombre del computador:	<input type="text"/>
Nombre usuario:	<input type="text"/>
Sistema Operativo:	<input type="text"/>
Tipo de procesador:	<input type="text"/>
Cantidad de procesador:	<input type="text"/>
Velocidad del procesador (MHz):	<input type="text"/>
BIOS	
Fabricante:	<input type="text"/>
Modelo:	<input type="text"/>
Sonido	
Fabricante:	<input type="text"/>
Tarjeta de video	
Chipset:	<input type="text"/>
Memoria (MB):	<input type="text"/>
Resolución:	<input type="text"/>
Monitor	
Fabricante:	<input type="text"/>
Tipo:	<input type="text"/>
Almacenamiento	
Modelo:	<input type="text"/> +
Tipo:	<input type="text"/>
Capacidad (MB):	<input type="text"/>
Memoria (s)	
Capacidad (MB):	<input type="text"/> +
Velocidad (MHz):	<input type="text"/>
Red (es)	
Dirección IP:	<input type="text"/> +
Tipo:	Ethernet ▾
<input type="button" value="Aceptar"/>	

Figura 16: Insertar inventario.

Computador	
ID:	1
Nombre del computador:	pc-12
Nombre usuario:	mlgarcia
Sistema Operativo:	Ubuntu 10.10
Tipo de procesador:	Intel(R) Core(TM)2 Duo CPU
Cantidad de procesador:	2
Velocidad del procesador (MHz):	2200
BIOS	
Fabricante:	Intel Corp.
Modelo:	
Sonido	
Fabricante:	Intel Corporation 82801H (IC
Tarjeta de video	
Chipset:	VGA compatible controller
Memoria (MB):	512
Resolución:	1024x768
Monitor	
Fabricante:	AOC
Tipo:	
Almacenamiento	
Modelo:	CDDVDW SH-S223C
Tipo:	cdrom
Capacidad (MB):	0
Modelo:	ST3160815AS
Tipo:	disk
Capacidad (MB):	152628
Memoria (s)	
Capacidad (MB):	512
Velocidad (MHz):	667 MHz (1.5 ns)
Red (es)	
Dirección IP:	10.33.2.11
Tipo:	Ethernet
Dirección IP:	127.0.0.0
Tipo:	Local

Aceptar

Figura 17: Modificar inventario.

Mostrar: ▼

Restringir vista: ▼ **Filtrar** 

Adicionar columna: ▼ 

2 Resultados  

Último inventario	Computador	Nombre de usuario	Sistema Operativo	Versión	Borrar	Editar
2011-04-04 21:54:10	pc-12	mlgarcia	Ubuntu 10.10	3.112ubuntu1		
2011-04-04 22:30:23	dlantunez-desktop	dlantunez	Ubuntu 10.10	0.137		

Figura 18: Mostrar inventario.

Ultimo inventario	Computador	Nombre de usuario	Sistema Operativo	Versión	Borrar	Editar
2011-04-04 21:54:10	pc-12	mlgarcia	Ubuntu 10.10	3.112ubuntu1		
2011-04-04 22:30:23	dlantunez-desktop	dlantunez	Ubuntu 10.10	0.137		

Figura 19: Eliminar inventario.

2.4.5 HU Reportes en PDF

Historia de Usuario	
Número: 05	Nombre Historia de Usuario: Reportes en PDF.
Modificación de Historia de Usuario Número: 1	
Usuario: Diana Lilian Antúnez Ginarte	Iteración Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 2
Descripción: Permitir descargar y mostrar los reportes de inventario en formato PDF.	
Observaciones:	
Prototipo de interfaz: No contiene prototipo de interfaz.	

Todas las HU llevan una prioridad para su implementación. La prioridad sirve para la planificación de las mismas en el Plan de Release.

2.5 Plan de Release

Se recogen las iteraciones a realizar con sus características, además del orden de las historias de usuario con la planificación estimada para su implementación.

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	Desarrollo de las Historias de Usuario de alta prioridad.	1	6 semanas

2	Desarrollo de las Historias de Usuario que tienen prioridad media e integrar los componentes desarrollados en iteraciones anterior.	3,4	4 semanas
3	Desarrollo de las Historias de Usuario de prioridad baja e integrar los componentes desarrollados en iteraciones anterior.	2, 5	2 semanas

2.6 Lista de Riesgo

En todo sistema en desarrollo existen un conjunto de riesgos los cuales son la posible causa al fracaso del producto en desarrollo, comprometiendo la entrega en tiempo y la calidad requerida del sistema, tema a tener en cuenta por parte del líder del proyecto y el gerente mientras dure el proceso de desarrollo.

SXP, propone documentar estos riesgos en un la Plantilla Lista de Riesgos identificando en la misma los riesgos que puedan incidir en el desarrollo del software así como las estrategias trazadas para mitigarlos.

Riesgo	Tipos de riesgos	Impacto	Descripción	Probabilidad	Efectos	Mitigación del riesgo
1	Estimación.	Desarrollo.	Complejidad de las tecnologías a utilizar.	Alta.	Tolerables.	Obteniendo libros más específicos del tema.
2	Personal.	Retraso en la entrega del producto.	Ausencia de un programador, analista, documentador	Bajo.	Serios.	Se evita programando bien las tareas de cada cual sin crear

						dependencia de un trabajo con otro y teniendo más de una persona preparada en cada tema.
3	Tecnológico.	Pérdida de la información valiosa.	Rotura de una máquina donde se encuentra información importante.	Bajo.	Serios.	Se evita actualizando toda la información vital del proyecto en el sistema de control de versiones.
4	Tecnológico.	Retraso de las actividades.	Afectación del fluido eléctrico.	Baja.	Tolerables.	Buscar horarios alternativos para trabajar.
5	Requerimientos.	Dudas en la calidad de nuestro trabajo, modificar requisitos volviendo a fases anteriores.	Inconformidad en la arquitectura y diseño del producto.	Baja.	Tolerables.	Comunicación constante con los clientes logrando un mayor entendimiento y claridad en lo que realmente desea.
6	Tecnológico.	Productividad afectada.	Se desconecta la red constantemente.	Alta.	Serios.	Comunicación al SCRUM del problema.
7	Personal.	Productividad	Analista no	Muy alta.	Altos.	Tesistas

		ad afectada.	domina la metodología SXP.			asumen parte del análisis y tienen apoyo.
8	Personal.	Se afecta la calidad del producto.	Desarrollador es no realizan las pruebas una vez implementada la funcionalidad, sino que lo dejan para el final del proceso de desarrollo, en la fase de entrega.	Alta.	Serios.	Gerente chequea casos de prueba con los desarrolladores al culminar cada iteración.

2.7 Arquitectura y Modelo del diseño

La Arquitectura de Software establece los fundamentos para que analistas, diseñadores, programadores, trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. Estas indican la estructura, funcionamiento e interacción entre las partes del software.

OCS Inventory NG generalmente presenta una arquitectura cliente-servidor y dentro de la consola de administración donde se implementaron todas las funcionalidades mantiene una arquitectura en n capas (ver Anexo1) con un reparto claro de funciones, en el caso de OCS Inventory NG la cantidad de capas que presenta son 3: una capa para la representación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

En la capa de presentación se encuentran las clases index, html_header y footer donde se hacen llamadas a las funcionalidades para presentar las vistas en el navegador. En la capa controladora se agruparon las clases header, ms_notifications, ms_all_computer, ms_manual_inventory, ms_pdf, ms_soft_pdf y function_table_html correspondiendo la clase ms_manual_inventory al

Módulo Generar inventario manual, ms_notifications al Módulo Control de cambios, ms_pdf y ms_soft_pdf al Módulo Reportes en pdf y el resto de las clases para la utilización de funcionalidades necesarias, estas se relacionan con las clases de la capa acceso a datos, lo que permite abstraerse del sistema gestor de bases de datos que se utilice.

Los paquetes y componentes en OCS Inventory NG están representados en el Anexo2, de manera que pueda verse la estructura lógica del sistema. En el paquete bakend se encuentran los archivos necesarios para la autenticación en el sistema. El paquete require contiene todas las funciones utilizadas por los distintos módulos. El paquete plugins incluye todos los módulos del sistema. Para una mayor comprensión ver Anexo2.

2.8 Modelo de Datos

OCS Inventory NG está compuesto por una base de datos conformada por 92 tablas, que contiene los datos relacionados a los reportes de inventario. Con la implementación del módulo que gestiona el control de cambios en la herramienta, y la funcionalidad Gestionar inventario manual ha sido necesario agregar nuevas tablas y nuevos atributos a tablas ya existentes.

En el Anexo 3 se presentan las tablas principales. En las tablas **hardware** y **operator** se agregaron nuevos atributos, mientras que en **config** por ser una tabla con valores estáticos solo se usó para insertar datos nuevos. Mientras que las nuevas agregadas a la base de datos son: **resources**, **resources_type**, **change_type**, **change_history**, **notifications** y **notifications_event**.

Teniendo estructurado el sistema y haber definido los módulos que van a ser modificados y sobre los cuales se va a implementar y haber insertado en la base de datos las tablas necesarias, para desarrollar las funcionalidades requeridas al sistema, se procede a implementar las historias de usuario. Para ello se elaboran una serie de tareas de ingenierías, que no son más que una serie de actividades a seguir para poder conformar correctamente las HU definidas para el sistema.

2.9 Tareas de Ingenierías

A continuación se definen las actividades asociadas a las historias de usuario. En las mismas se determina el programador asignado a cada tarea así como el tiempo necesario para su realización.

2.9.1 Tareas de ingenierías asociadas a la Historia de Usuario Gestionar notificaciones

Tarea de Ingeniería	
Número Tarea: 01	Número Historia de Usuario: 01
Nombre Tarea: Instalar herramienta de gestión de inventario OCS Inventory NG	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 17/09/2010	Fecha Fin: 20/09/2010
Programador Responsable: Mario Lázaro García Pérez y Diana Lilian Antúnez Ginarte	
Descripción: Se instala la herramienta para comenzar los primeros estudios de la misma.	

Tarea de Ingeniería	
Número Tarea: 02	Número Historia de Usuario: 01
Nombre Tarea: Estudiar estructura lógica de la herramienta	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 21/09/2011	Fecha Fin: 24/09/2011
Programador Responsable: Mario Lázaro García Pérez y Diana Lilian Antúnez Ginarte	
Descripción: Estudio de todos los componentes asociados a la herramienta así como la estructura del sistema de base de datos.	

Tarea de Ingeniería	
Número Tarea: 03	Número Historia de Usuario: 01
Nombre Tarea: Crear tablas de la base de datos de la herramienta	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 18/01/2011	Fecha Fin: 22/01/2011
Programador Responsable: Mario Lázaro García Pérez	
Descripción: Se crean las tablas necesarias para la implementación de las historias de usuarios.	

Tarea de Ingeniería	
Número Tarea: 04	Número Historia de Usuario: 01
Nombre Tarea: Estudio del lenguaje de programación Perl	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 18/01/2011	Fecha Fin: 31/03/2011
Programador Responsable: Diana Lilian Antúnez Ginarte y Mario Lázaro García Pérez	
Descripción: Se realiza estudio del lenguaje para la implementación del insertar notificación.	

2.9.2 Tareas de Ingenierías asociadas a la Historia de Usuario Gestionar inventario manual

Tarea de Ingeniería	
Número Tarea: 05	Número Historia de Usuario: 04
Nombre Tarea: Insertar datos manualmente	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 16/02/2011	Fecha Fin: 25/03/2011
Programador Responsable: Mario Lázaro García Pérez	
Descripción: Se insertan datos manualmente para la implementación de la historia de usuario	

correspondiente.

2.9.3 Tareas de Ingenierías asociadas a la Historia de Usuario Dar de baja a un equipo

Tarea de Ingeniería	
Número Tarea: 06	Número Historia de Usuario: 02
Nombre Tarea: Eliminar computador inventariado	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 22/02/2011	Fecha Fin: 25/02/2011
Programador Responsable: Diana Lilian Antúnez Ginarte	
Descripción: Se implementa la funcionalidad dar baja a PC.	

2.9.4 Tareas de Ingenierías asociadas la Historia de Usuario Gestionar envío de correo electrónico

Tarea de Ingeniería	
Número Tarea: 07	Número Historia de Usuario: 03
Nombre Tarea: Estudio de los módulos de Perl y librerías PHP para el envío de correo electrónico	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 22/02/2011	Fecha Fin: 04/03/2011
Programador Responsable: Diana Lilian Antúnez Ginarte y Mario Lázaro García Pérez	
Descripción: Se realiza estudio de los módulos de Perl Email::Sender::Simple, Email::Sender::Transport::SMTP::TLS y Email::Simple::Creator y la librería phpmailer de PHP para el envío de correo electrónico.	

2.9.5 Tareas de Ingenierías asociadas la Historia de Usuario Reportes en PDF

Tarea de Ingeniería	
Número Tarea: 08	Número Historia de Usuario: 05
Nombre Tarea: Estudiar y trabajar con la librería ezpdf .	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 18/04/2011	Fecha Fin: 21/04/2011
Programador Responsable: Diana Lilian Antúnez Ginarte y Mario Lázaro García Pérez	
Descripción: Se realiza estudio de la librería ezpdf .	

Conclusiones parciales

En este capítulo se definieron los requisitos funcionales que el sistema debe ser capaz de brindar, a través de las historias de usuarios que el desarrollador debe cumplir por medio de una serie de tareas de ingenierías asociadas a cada historia de usuario. Se recogen las iteraciones en que serán desarrolladas las funcionalidades de la solución propuesta, tiempo que se demorará la implementación de las historias de usuario y la confección de los principales diagramas que guiaron el proceso de desarrollo así como un cambio de identidad en el sistema siendo renombrado el mismo como GITI.

Capítulo 3: Pruebas al Sistema.

Las pruebas que se le realizan a todo producto de software son cruciales en el ciclo de desarrollo del mismo, permitiendo verificar la seguridad del producto así como ganar en calidad. El presente capítulo expone las pruebas realizadas a la herramienta OCS Inventory NG después de implementada la propuesta de solución.

3.1 Casos de pruebas de Aceptación

A continuación se describe en detalles un conjunto de pruebas de aceptación asociadas a las Historias de Usuario pertenecientes a los módulos implementados a OCS Inventory NG, con el fin de verificar que la propuesta de solución cumple con el funcionamiento esperado, permitiendo así su aceptación por parte del usuario.

3.1.1 Casos de pruebas de aceptación para la Historia de Usuario: HU-01

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-01-01	Nombre Historia de Usuario: Gestionar notificación
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón	
Descripción de la Prueba: Se procede a modificar notificación	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y debe existir al menos una notificación.	
Entrada / Pasos de ejecución: Clic sobre la opción editar mostrada en la tabla (imagen en forma de lápiz). Se modifica los datos correctamente, se da clic en el botón aplicar.	
Resultado Esperado: Se confirma si la acción se realizó correctamente y se muestra la tabla con la modificación realizada.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-01-02	Nombre Historia de Usuario: Gestionar notificación
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón	
Descripción de la Prueba: Se procede a eliminar notificación.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y debe existir al menos una notificación.	

Entrada / Pasos de ejecución: Clic sobre la opción borrar (representada por una X en rojo) o seleccionar una o varias notificaciones y dar clic sobre el icono que se encuentra debajo de la tabla (eliminar seleccionados).
Resultado Esperado: Se elimina la notificación de las 2 formas que se establecieron.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-01-03	Nombre Historia de Usuario: Gestionar notificación
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón.	
Descripción de la Prueba: Se procede a visualizar las notificaciones.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y debe existir al menos una notificación.	
Entrada / Pasos de ejecución: Clic sobre icono de notificaciones.	
Resultado Esperado: Muestra una tabla con los datos asociados a las notificaciones.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-01-04	Nombre Historia de Usuario: Gestionar notificación
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón.	
Descripción de la Prueba: Se inserta notificación.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y existir al menos un inventario.	
Entrada / Pasos de ejecución: Realizar nuevo inventario a la misma PC.	
Resultado Esperado: Muestra en el icono de notificaciones, la cantidad.	
Evaluación de la Prueba: Satisfactoria.	

3.1.2 Casos de pruebas de aceptación para la Historia de Usuario: HU-2

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-02-05	Nombre Historia de Usuario: Dar de Baja a un equipo
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón	
Descripción de la Prueba: Se procede a eliminar PC del inventario.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y debe existir al menos un reporte de inventario.	

Entrada / Pasos de ejecución: Clic sobre la opción borrar (representada por una imagen en formad X en roja) o seleccionar una o varias PC y dar clic sobre el icono que se encuentra debajo de la tabla(eliminar seleccionados).
Resultado Esperado: Muestra una tabla con los datos asociados a las notificaciones.
Evaluación de la Prueba: Satisfactoria.

3.1.3 Casos de pruebas de aceptación para la Historia de Usuario: HU-3

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-03-06	Nombre Historia de Usuario: Gestionar envío de correo electrónico
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón	
Descripción de la Prueba: Enviar correo cuando se inserte una notificación.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y deben ocurrir cambios en el inventario.	
Entrada / Pasos de ejecución: Realizar un nuevo inventario a la misma PC, se registra un cambio en el inventario (notificación), envía correo electrónico automáticamente.	
Resultado Esperado: El correo llega correctamente al destinatario.	
Evaluación de la Prueba: Satisfactoria.	

3.1.4 Casos de pruebas de aceptación para la historia de usuario: HU-4

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-04-07	Nombre Historia de Usuario: Gestionar inventario manual
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón	
Descripción de la Prueba: Insertar inventario manual.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose.	
Entrada / Pasos de ejecución: Clic sobre el icono Inventario Manual y seleccionar la opción Nuevo Inventario . Se llenan los campos correspondientes y se da clic en el botón Aceptar .	
Resultado Esperado: Confirma el registro del inventario correctamente y muestra en la tabla de inventario el nuevo registro realizado.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-04-08	Nombre Historia de Usuario: Gestionar inventario manual

Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón
Descripción de la Prueba: Mostrar inventario manual.
Condiciones de Ejecución: La aplicación debe estar ejecutándose y debe existir al menos un inventario.
Entrada / Pasos de ejecución: Clic sobre el icono Inventario Manual y seleccionar la opción Computador .
Resultado Esperado: Muestra una tabla con los computadores inventariados.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-04-09	Nombre Historia de Usuario: Gestionar inventario manual
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón	
Descripción de la Prueba: Eliminar inventario manual.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y debe existir al menos reporte de inventario.	
Entrada / Pasos de ejecución: Dar clic en la X en roja que aparece en la tabla en la columna Borrar .	
Resultado Esperado: Confirma que la acción se ha realizado correctamente y muestra la tabla con el cambio realizado.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-04-10	Nombre Historia de Usuario: Gestionar inventario manual
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón	
Descripción de la Prueba: Modificar inventario manual.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y debe existir al menos un reporte de inventario.	
Entrada / Pasos de ejecución: Dar clic en la imagen en forma de lápiz que aparece en la tabla en la columna Actualizar . Se muestra un formulario con todos los datos asociados al inventario, se modifican los necesarios y se da clic en el botón Aceptar .	
Resultado Esperado: Se confirma que la actualización ha sido realizada correctamente.	
Evaluación de la Prueba: Satisfactoria.	

3.1.5 Casos de pruebas de aceptación para la historia de usuario: HU-5

Caso de Prueba de Aceptación	
Código Caso de Prueba: PHGIO-05-10	Nombre Historia de Usuario: Reportes en PDF
Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón	
Descripción de la Prueba: Se procede a mostrar los reportes en pdf.	
Condiciones de Ejecución: La aplicación debe estar ejecutándose y debe existir al menos un inventario.	
Entrada / Pasos de ejecución: Clic sobre el icono PDF (se puede descargar o abrir directamente el documento), Aceptar .	
Resultado Esperado: Descarga correctamente el archivo. Muestra documento con reportes.	
Evaluación de la Prueba: Satisfactoria.	

3.2 Resultados obtenidos

Con el desarrollo del módulo de control de cambios en la herramienta OSC Inventory NG, se propicia un mejor manejo de la información obtenida cuando se realiza un inventario. La personalización realizada a la herramienta aumenta la calidad y eficiencia en los reportes de inventario, así como su utilidad durante el proceso de migración pues la misma es capaz de notificar siempre que ocurra un cambio tras haber realizado un inventario en la entidad en la que se este trabajando.

Las principales funcionalidades que provee la aplicación después de su perzonalización son:

- Inventariar manualmente (insertar, modificar, eliminar y mostrar inventario).
- Notificar sobre algún cambio en un inventario ya realizado.
- Configurar envío de correo electrónico.
- Enviar notificación vía correo.
- Poder eliminar, modificar y visualizar las notificaciones.
- Dar baja a una o varias PC.
- Permitir ver y descargar reportes y notificaciones en formato PDF.

Conclusiones parciales

En este capítulo se mostraron los pasos para probar el funcionamiento de la herramienta OCS Inventory NG tras haber realizado la personalización de la misma poniendo en práctica cada una de las funcionalidades implementadas, garantizando así la entrega de un producto que responde a las necesidades del cliente. Además se describieron las pruebas que corroboran la calidad del producto pues hace lo que se supone que tiene que hacer.

Conclusiones

Con la culminación del presente trabajo de diploma se dio cumplimiento a cada uno de los objetivos trazados, destacándose de manera general los siguientes aspectos:

- Fueron analizados diferentes sistemas de gestión de inventario, estudio que permitió identificar características y funcionalidades que se ajustan a las necesidades exigidas por SIMAYS.
- Con el uso de las herramientas, tecnologías y lenguajes de programación que soportaron el proceso de desarrollo guiado por la metodología SXP, se obtiene un producto que puede enfrentarse a cambios en los requisitos.
- Fueron documentadas las funcionalidades que cumple el sistema para su posterior desarrollo, concluyendo que su implementación se encuentran en correspondencia con las necesidades del cliente.
- El diseño y ejecución de un conjunto de pruebas funcionales realizadas al culminar cada nueva funcionalidad aseguraron la calidad del producto final.
- Se desarrolló un instalador del sistema para poder instalar el mismo en cualquier lugar, así como también se elaboró un manual de su funcionamiento e instalación.

Recomendaciones

Al concluir la presente investigación se recomienda:

- Desplegar el producto con el objetivo de asegurar su correcto funcionamiento.
- Utilizar el producto en un proceso de migración real.
- Implementar servicios web para que pueda ser integrada a la Plataforma de Migración.

Referencias Bibliográficas

[1] Yoandy Pérez Villazón, Metodología para la migración a Software Libre de las universidades del Ministerio Superior (MES). Universidad de las Ciencias Informáticas. 2008.

[2] Lisandra Cala Hernández, Propuesta de Integración y Nuevas Herramientas para el Desarrollo de La Plataforma Cubana de Migración a Software Libre. Universidad de las Ciencias Informáticas. 2010.

[3] ProactivaNET® Inventario. [citado 11 Octubre 2010]. Disponible en: <http://www.proactivanet.com/proactivanet-inventario>.

[4] ProactivaNET® Service Desk. [citado 15 Octubre 2010]. Disponible en: <http://www.proactivanet.com/proactivanet-service-desk>.

[5] Integración con el inventario de red. [citado 11 Octubre 2010]. Disponible en: <http://www.proactivanet.com/integracion-con-el-inventario-de-red>.

[6] ProactivaNET® Service Desk - Gestión de incidencias. [citado 11 Octubre 2010]. Disponible en: <http://www.proactivanet.com/proactivanet-service-desk-gestion-de-incidencias>.

[7] ProactivaNET® Service Desk - Gestión de Problemas. [citado 11 Octubre 2010]. Disponible en: <http://www.proactivanet.com/proactivanet-service-desk-gestion-de-problemas>.

[8] ProactivaNET® Service Desk – Gestión de Cambios y Entregas. [citado 11 Octubre 2010]. Disponible en: <http://www.proactivanet.com/proactivanet-service-desk-gestion-de-cambios-y-entregas>.

[9] Alertas y eventos para el mantenimiento de red. [citado 12 Octubre 2010]. Disponible en: <http://www.proactivanet.com/alertas-y-eventos-para-el-mantenimiento-de-red>.

[10] EVEREST Corporate Edition v5.50. [citado 20 Octubre 2011]. Disponible en: <http://www.abcdatos.com/programas/programa/g178.html>.

[11] Camilo D. González, Alié Castillo Ruíz, Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Sistema de Inventario de Hardware y Software. Módulo Obtención de Información, Universidad de las Ciencias Informáticas. 2008. 78.

[12] Gladys Marsi Peñalver Romero, MA-GMPR-UR2 Metodología ágil para proyectos de software

libre. Universidad de las Ciencias Informáticas. 2008.

[13] El lenguaje HTML. Manual de HTML. Tutorial de HTML. WebEstilo. 2010. Disponible en: <http://www.webestilo.com/html/cap2a.phtml>.

[14] Guía Breve de CSS. 2010. Disponible en: <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.

[15] Perl Básico - Parte 1 - Perl En Español. 2010. Disponible en: http://perlenespanol.com/tutoriales/aprendiendo_perl/perl_basico_parte_1.html.

[16] Qué es Javascript. 2010. Disponible en : <http://www.desarrolloweb.com/articulos/25.php>.

[17] IBM - Rational - Unified Modeling Language - UML Resource Center. 2010. Disponible en: <http://www-01.ibm.com/software/rational/uml/>.

[18] Capítulo 1. Introducción a AJAX | Introducción a AJAX | LibrosWeb.es. [citado 23 Noviembre 2010]. Disponible en: <http://www.librosweb.es/ajax/capitulo1.html>.

[19] Sistemas gestores de bases de datos. 2010. Disponible en: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

[20] MySQL :: Why MySQL? 2010. Disponible en: <http://www.mysql.com/why-mysql/>.

[21] C. Mateu, Desarrollo de aplicaciones web, Eureka Media, 2004. Disponilbe en: www.apimadrid.org/media/libros/16-Desarrollo_web.pdf.

[22] Servidor de Aplicaciones Web. [citado 11 Noviembre 2010]. Disponilbe en: <http://es.scribd.com/doc/19217037/Servidor-de-Aplicaciones-Web>.

Bibliografía

1. Yoandy Pérez Villazón, Metodología para la migración a Software Libre de las universidades del Ministerio Superior (MES). Universidad de las Ciencias Informáticas. 2008.
2. Ramón Paumier Samón, Yoandy Pérez Villazón, and Abel Meneses Abad. GUÍA CUBANA DE MIGRACIÓN A SOFTWARE LIBRE. 2008.
3. Sayda Coello González, Rolando Alfredo Hernández León. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. [Ciudad de la Habana]: EDUNIV, 2002.
4. Lisandra Cala Hernández, Propuesta de Integración y Nuevas Herramientas para el Desarrollo de La Plataforma Cubana de Migración a Software Libre. Universidad de las Ciencias Informáticas, 2010.
5. Gladys Marsi Peñalver Romero, MA-GMPR-UR2 Metodología ágil para proyectos de software libre. Universidad de las Ciencias Informáticas. 2008.
6. Abel Meneses Abad y Gladys Marsi Peñalver Romero, "SXP, metodología ágil para proyectos de software libre". Universidad de las Ciencias Informáticas. 2009.
7. Inventario de Software y Hardware - EVEREST, LOGINVENTORY, SYMANTEC DISCOVERY. [citado 20 Octubre 2010]. Disponible en: <http://www.trustation.com/inventario.htm>.
8. Análisis sistema - Programas para Windows: Softonic. [citado 15 Octubre 2010]. Disponible en: <http://www.softonic.com/windows/analisis-sistema>.
9. EVEREST Corporate Edition v5.50. [citado 20 Octubre 2010]. Disponible en: <http://www.abcdatos.com/programas/programa/g178.html>.
10. EVEREST v5.50 Is Now Available | Lavalys.com. [citado 20 Octubre 2010]. Disponible en: <http://www.lavalys.com/community/blog/2010/04/everest-v550-now-available/>.
11. ProactivaNET®. [citado 10 Octubre 2010]. Disponible en: <http://www.proactivanet.com/>.
12. Alertas y eventos para el mantenimiento de red. [citado 12 Octubre 2010]. Disponible en:

- <http://www.proactivanet.com/alertas-y-eventos-para-el-mantenimiento-de-red>.
13. Éxitos. [citado 16 Octubre 2010]. Disponible en: <http://www.proactivanet.com/exitos/list>.
 14. ProactivaNET® Service Desk. [citado 15 Octubre 2010]. Disponible en: <http://www.proactivanet.com/proactivanet-service-desk>.
 15. Auditoría e inventario de red. [citado 12 Octubre 2010]. Disponible en: <http://www.proactivanet.com/auditoria-e-inventario-automatico>.
 16. ProactivaNET® Control Remoto. [citado 11 Octubre 2010]. Disponible en: <http://www.proactivanet.com/proactivanet-control-remoto>.
 17. Descripción Técnica. [citado 11 Octubre 2010]. Disponible en: <http://www.proactivanet.com/descripcion-tecnica>.
 18. OCS Inventory NG - Welcome to OCS Inventory NG web site ! [citado 5 Octubre 2010]. Disponible en: <http://www.ocsinventory-ng.org>.
 19. W3C HTML. [citado 25 Noviembre 2010]. Disponible en: <http://www.w3.org/html>.
 20. C. Mateu, Desarrollo de aplicaciones web, Eureka Media. 2004. Disponible en: www.apimadrid.org/media/libros/16-Desarrollo_web.pdf.
 21. Guía Breve de CSS. 2010. Disponible en: <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.
 22. Camilo D. González, Alié Castillo Ruíz, Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Sistema de Inventario de Hardware y Software. Módulo Obtención de Información, Universidad de las Ciencias Informáticas. 2008.
 23. Departamento de Ingeniería de Software, UCI, "Introducción a la Ingeniería de Software."
 24. PHP: Manual de PHP - Manual. [citado 25 Noviembre 2010]. Disponible en: <http://www.php.net/manual/es/index.php>.
 25. Diseño Arquitectura Del Software. [citado 12 Febrero 2011]. Disponible en: <http://www.mitecnologico.com/Main/Dise%F1oArquitecturaDelSoftware%3E>.
 26. Servidor de Aplicaciones Web. [citado 25 Noviembre 2010]. Disponible en:

<http://es.scribd.com/doc/19217037/Servidor-de-Aplicaciones-Web>.

27. Lenguajes de programación, Lenguajes de programación. [citado Noviembre 2010]. Disponible en: <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
28. Descripción Técnica. [citado 11 Octubre 2010]. Disponible en: <http://www.proactivanet.com/descripcion-tecnica>.
29. PHP: Características, Manual. 2010. Disponible en: <http://www.php.net/manual/es/features.php>.
30. PHP: Referencia de funciones, Manual. 2010. Disponible en: <http://www.php.net/manual/es/funcref.php>.
31. Rafael Andrés Leaña, and Luis Carlos Beltrán. La Arquitectura De Netbeans. 2008. [citado 15 Noviembre 2010]. Disponible en: <http://www.slideshare.net/ralphkui/la-arquitectura-de-netbeans-v2>.
32. Metodologías de desarrollo software. [citado 15 Noviembre 2010]. Disponible en: <http://es.scribd.com/doc/2050925/metodologias-de-desarrollo-software>.
33. Guía de referencia de Nmap (Página de manual). [citado 7 Octubre 2010]. Disponible en: <http://nmap.org/man/es/>.
34. APACHE: Introducción a 2/4. [citado 25 Noviembre 2010]. Disponible en: <http://www.htmlpoint.com/apache/01a.htm>.
35. CASE. [citado 15 Noviembre 2010]. Disponible en: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
36. Servidor de Aplicaciones Web. [citado 25 Noviembre 2010]. Disponible en: <http://es.scribd.com/doc/19217037/Servidor-de-Aplicaciones-Web>.
37. Documentation:Plugins - OCS Inventory NG. [citado 5 Octubre 2010]. Disponible en: <http://wiki.ocsinventory-ng.org/index.php/Documentation:Plugins>.
38. 10.00_OCSNG_InventarioHardwareSoftware.pdf - Ésta es una idea de Google Docs. [citado 5 Octubre 2010]. Disponible en: http://docs.google.com/viewer?a=v&q=cache:8u6u85MZqoQJ:https://isl.unal.edu.co/fl18/conferencias/10.00_OCSNG_Inve

[ntarioHardwareSoftware.pdf+arquitectura+%2B+ocsreports&hl=es&pid=bl&srcid=ADGEESg-AifA7YT8723x-JoNamrJBPoYkSNBcZZ27eiwTyqueLYUj14citCTMEhzP1LAW2uF_7tsHyb9DtB_wvmSNHVfnsJnJYf0NPllcw6u5eASCecjS8fjz4lvOJyR2WzYw_eDc9Z8&sig=AHIEtbT5IX6u5SVTgH2iA4rh2KqGqkJRuA>.](#)

39. Main Page/es - OCS Inventory NG. [citado 5 Octubre 2010]. Disponible en: <http://wiki.ocsinventory-ng.org/index.php/Main_Page/es>.
40. Object Management Group - UML. [citado 25 Noviembre 2010]. Disponible en: <<http://www.uml.org/>>.
41. Welcome to NetBeans. [citado 25 Noviembre 2010]. Disponible en: <<http://netbeans.org/>>.
42. phpMyAdmin. [citado 25 Noviembre 2010]. Disponible en: <http://www.phpmyadmin.net/home_page/index.php>.
43. VISUALPARADIGM.COM. [citado 25 Noviembre 2010]. Disponible en: <<http://www.visualparadigm.com/>>.
44. RapidSVN. [citado 15 Noviembre 2010]. Disponible en: <<http://www.rapidsvn.org/>>.
45. What is Firebug? : Firebug. [citado 15 Noviembre 2010]. Disponible en: <<http://getfirebug.com/whatisfirebug>>.
46. FireBug - Descargar. [citado 15 Noviembre 2010]. Disponible en: <<http://firebug.softonic.com/>>.
47. Web Developer. [citado 15 Noviembre 2010]. Disponible en: <<http://chrispederick.com/work/web-developer/>>.
48. WebDeveloper: Herramientas para desarrollo web (II) « Historial de un navegante. [citado 15 Noviembre 2010]. Disponible en: <<http://historialweb.wordpress.com/2009/11/04/herramientas-desarrollo-web-ii/>>.
49. Geany : Home Page. [citado 20 Noviembre 2010]. Disponible en: <<http://www.geany.org/>>.
50. Capítulo 1. Introducción a AJAX | Introducción a AJAX | LibrosWeb.es. [citado 20 Noviembre 2010]. Disponible en: <<http://www.librosweb.es/ajax/capitulo1.html>>.

Anexos

Anexo 1

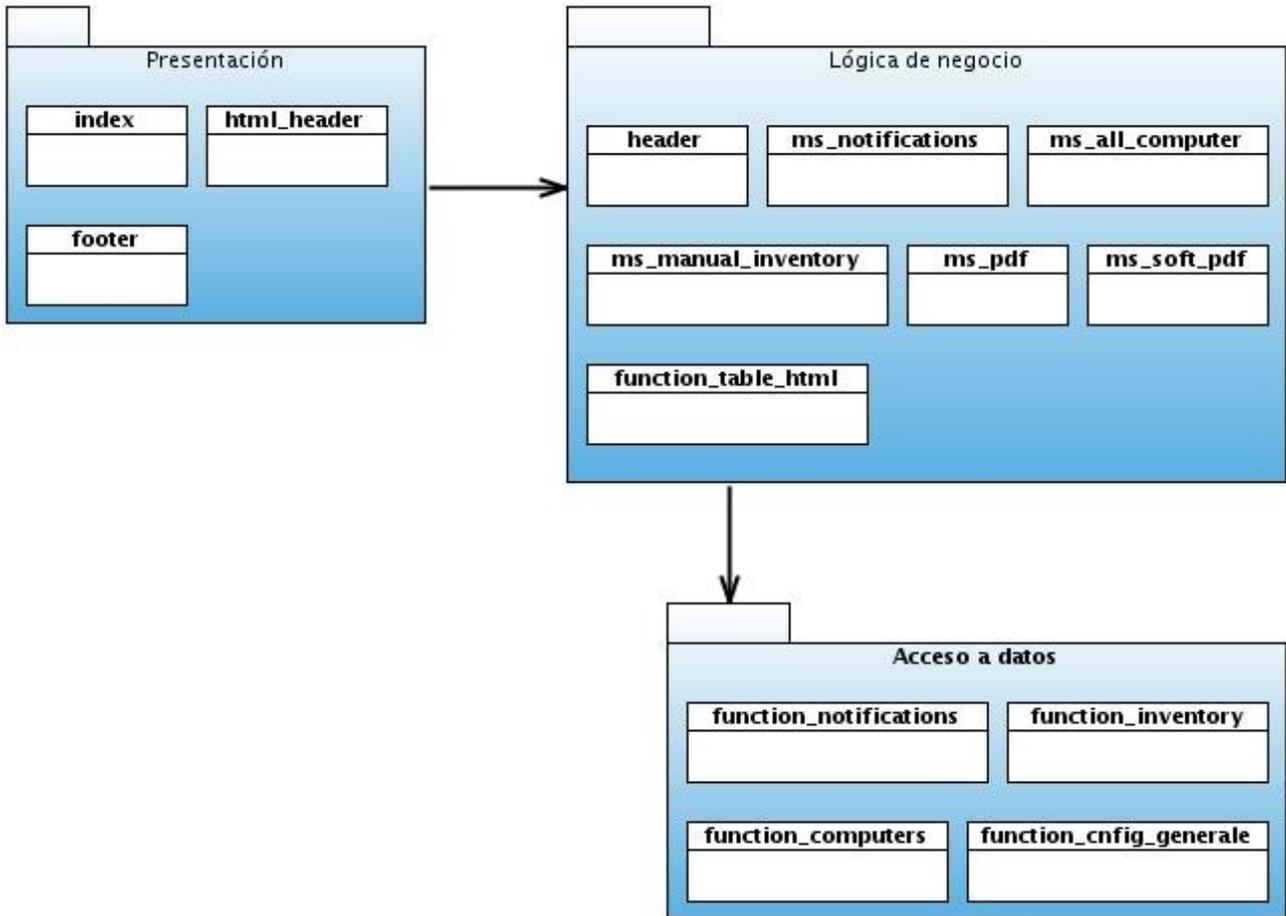


Figura 20: Diagrama de clases.

Anexo 2

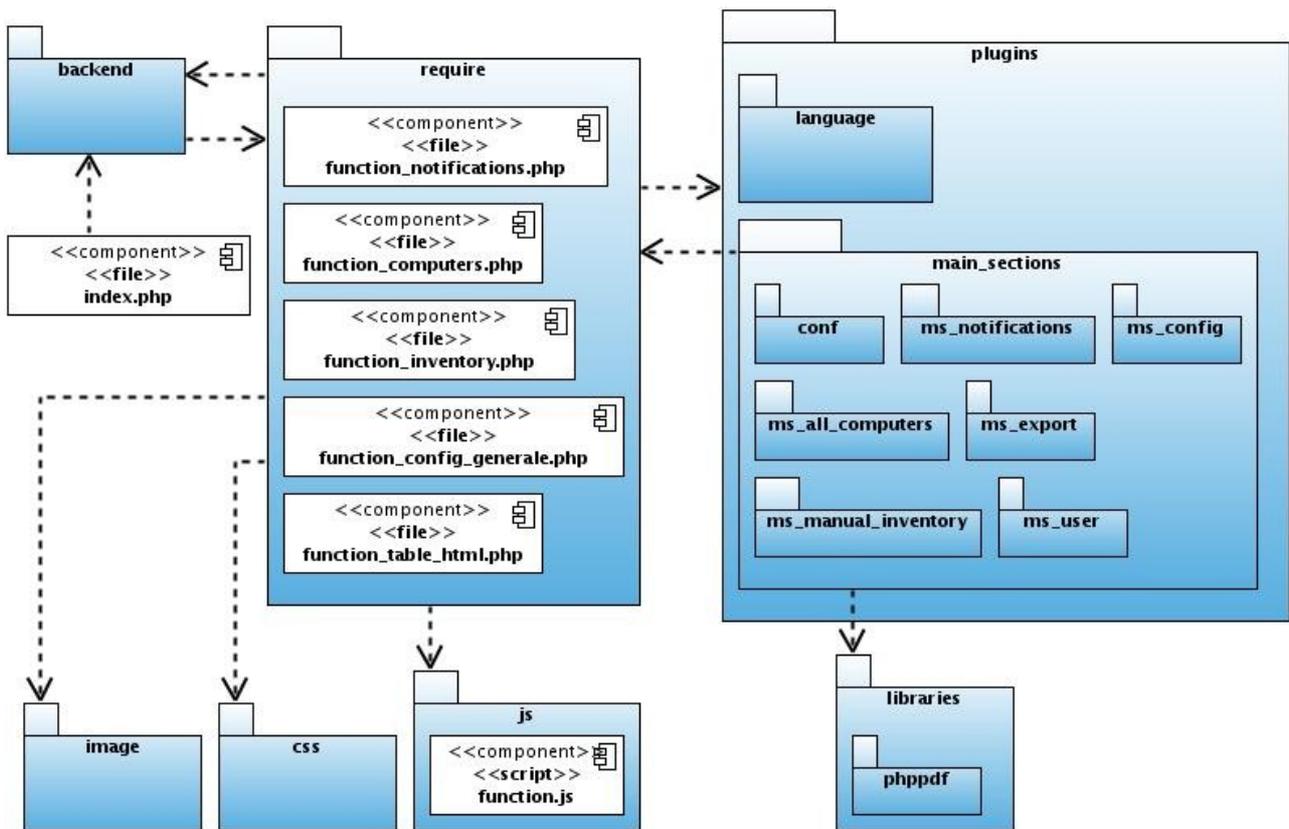


Figura 21: Diagrama de componentes.

Anexo 3

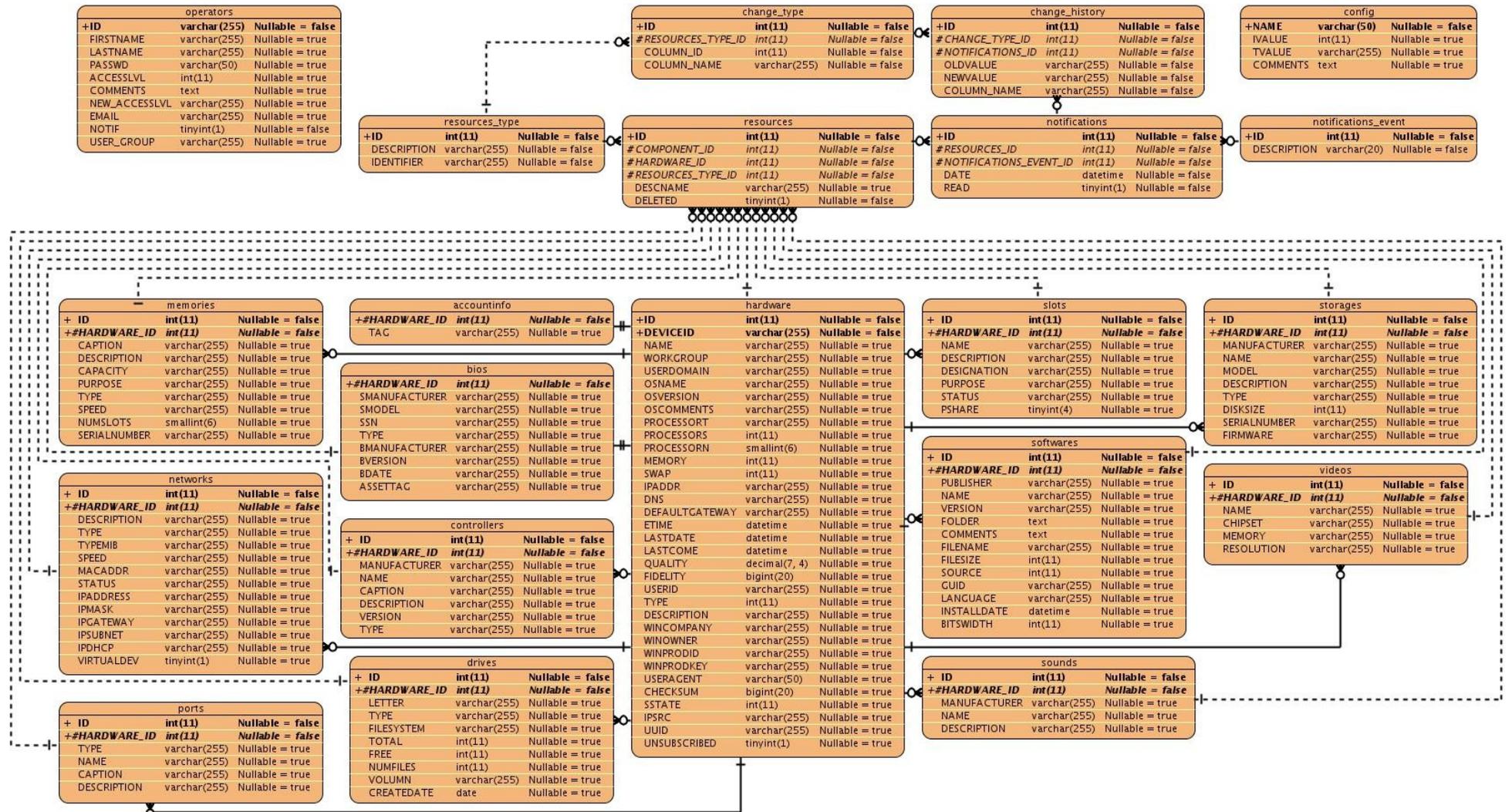


Figura 22: Diagrama de datos.

Glosario de Términos

ASP (Active Server Page): lenguaje de programación web desarrollado por la compañía Microsoft, que permite la creación de páginas web dinámicas a través de un conjunto de scripts interpretados del lado del servidor.

Almacenamiento transaccionales: uso de motores de almacenamiento que trabajan con tablas transaccionales (tienen requerimientos significativamente mayores para memoria y espacio de disco y mayor carga de CPU). Estos motores permiten una mejor concurrencia para tablas que tienen varias actualizaciones concurrentes con lecturas. Puede combinar varios comandos y aceptarlos todos al mismo tiempo con el comando COMMIT (si autocommit está desactivado). Puede ejecutar ROLLBACK para ignorar los cambios (si autocommit está desactivado). Si falla una actualización, todos los cambios se deshacen. (Con tablas no transaccionales, todos los cambios son permanentes.)

Almacenamiento no transaccionales: uso de motores de almacenamiento que trabajan con tablas no transaccionales, siguen un paradigma diferente para integridad de datos llamado "operaciones atómicas" las cuales a menudo ofrecen integridad comparable con mejor rendimiento. Proporciona almacenamiento y recuperación de datos rápida, así como posibilidad de búsquedas fulltext.

Código Abierto: término con el que se conoce al software distribuido y desarrollado libremente.

Common Gateway Interface (CGI): interfaz de entrada común es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa.

Escalabilidad: en Ingeniería Informática y redes se refiere a la cualidad que posee un sistema informático de crecer en servicios ofrecidos sin perder calidad o responder a nuevos flujos de trabajo o circunstancias cambiantes de manera eficiente.

eXtremme Programing (XP): es 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 propiciando un buen clima de trabajo.

FastCGI: es un protocolo para interconectar programas interactivos con un servidor web. El

principal objetivo de FastCGI es reducir la carga asociada con el hecho de interconectar el servidor web y los programas CGI, permitiéndole a un servidor atender más peticiones a la vez.

Interfaz gráfica de usuario: es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

Herramientas CASE: son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas.

Hipertextos: conjunto estructurado de textos, gráficos y otros elementos enlazados entre sí.

Hardware: corresponde a todas las partes tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos; sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

Intranet: una red de ordenadores privados, que proporciona los servicios de red necesarios a nivel de empresa u organización.

ITIL/ISO20000: ITIL (Information Technology Infrastructure Library) es una colección de libros que proporcionan una guía de las mejores prácticas para la entrega y el soporte de los servicios tecnológicos. ISO 20000 se limita a la recolección de requerimientos para una gestión de servicios de calidad. Por tanto, podemos utilizar la información presentada en la ISO 20000 para verificar si una organización esta cumpliendo con las prácticas propuestas por ITIL.

Programación Estructurada: es una forma de escribir programas de ordenador de manera clara. Para ello utiliza únicamente tres estructuras: secuencia, selección e iteración; siendo innecesario el uso de la instrucción o instrucciones de transferencia incondicional (GOTO, EXIT FUNCTION, EXIT SUB o múltiples RETURN).

Programación Funcional: es un paradigma de programación declarativa basado en la utilización de funciones aritméticas que no maneja datos mutables o de estado.

Portable Document Format (PDF): formato de archivos desarrollado por Adobe Systems.

PC: terminología usada para referirse a una computadora.

Release: se refiere a una versión funcional de un producto software.

Sockets: es un método para la comunicación entre un programa del cliente y un programa del servidor en una red. Un socket se define como el punto final en una conexión.

Servlets: es un programa que se ejecuta en un servidor.

SCRUM: define un marco de trabajo ágil para la gestión de proyectos, basado en iteraciones y entregas incrementales de desarrollo de un producto o servicio. Especialmente indicado para proyectos con requisitos cambiantes.

Software: programas del ordenador. Se refiere a instrucciones que se ejecutan por medio de un equipo de cómputo.

Servidor Web: software que suministra páginas Web en respuesta a las peticiones de los navegadores Web.

Servidor FTP: es un programa especial que se ejecuta en un servidor conectado normalmente en Internet (aunque puede estar conectado en otros tipos de redes, LAN, MAN, etc.). La función del mismo es permitir el desplazamiento de datos entre diferentes servidores / ordenadores.

Secure Sockets Layer (SSL): Protocolo de Capa de Conexión Segura es un protocolo criptográfico que proporciona comunicación segura por una red, comúnmente Internet.

Server Side Include (SSI): es un simple lenguaje interpretado de secuencias de comandos del lado del servidor utilizado casi exclusivamente para la web. El uso más frecuente de SSI es para incluir el contenido de uno o más archivos en una página web en un servidor web.

Scripts: es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

Tablas Hash: es una estructura de datos que asocia llaves o claves con valores.

Web dinámicas: permiten crear aplicaciones dentro de la propia Web, otorgando una mayor interactividad con el navegante. Aplicaciones dinámicas como encuestas y votaciones, foros de soporte, libros de visita, envío de e-mails inteligentes, reserva de productos, pedidos on-line, atención al cliente personalizada

ZIP: es un concepto que se utiliza en la informática para nombrar a ciertos formatos de

almacenamiento, ya sean físicos o virtuales. A nivel de software, el zip se utiliza para la compresión de datos (documentos de texto, imágenes, programas, etcétera.) sin pérdida de calidad.