



## Facultad 2

Trabajo de Diploma para optar por el título de Inge-  
niero en Ciencias Informáticas

# Plugin de integración de datos de GRHS a GLPI.

**Autores:**

Félix Rafael Benavides Mendoza.  
Rafael Arismende Barinas Ricardo.

**Tutores:** Ing. Osbel Ledesma Peral.  
Ing. Claudia Liset Stincer Torre

La Habana 2015

“Año 57 de la Revolución”

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Telemática (TLM) de la Facultad 2 y a la Universidad de las Ciencias Informáticas (UCI) a hacer el uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de junio de 2015

---

Firma del Autor

Rafael Arismende Barinas Ricardo

---

Firma del Autor

Félix Rafael Benavides Mendoza

---

Firma del Tutor

Ing.Osbel Ledesma Peral

---

Firma del Tutor

Ing. Claudia Liset Stincer Torres

FRASE

spanish-quotes | tumblr

“TODOS SOMOS UNOS  
GENIOS. PERO SI JUZGAS  
A UN PEZ POR SU HABILIDAD  
DE ESCALAR UN ÁRBOL,  
VIVIRÁ SU VIDA ENTERA  
CREYENDO QUE ES  
ESTÚPIDO”

-Albert Einstein

## AGRADECIMIENTOS

*Félix Rafael*

*Les agradezco a mis padres, Magalis y Armando, porque me han apoyado siempre en las decisiones que tomo y sin ellos no estaría hoy aquí.*

*A mis abuelos Rafael, Hilda, Gladis y Félix Mendoza este último lamentablemente no se encuentra entre nosotros, pero siempre estaré agradecido con todos por estar siempre al tanto de lo que me pasaba y por sus cariños.*

*A mi hermana Niuvis, por todo el apoyo que me ha brindado durante mi carrera y por ser sobre todo una excelente hermana.*

*A mi sobrinita Doralis que aunque es muy pequeña todavía, siempre me ha cautivado con su ternura.*

*A toda mi familia.*

*A mi compañero de tesis por ser un buen amigo tanto en las buenas como en las malas.*

*A mis amigos de la infancia y a todos los compañeros y amigos con los que he compartido en los cinco años de carrera que no han cambiado nunca y que espero logren sus metas.*

*A mis profesores de todos estos años y a mis tutores, Liset y Osbel por todo el apoyo y la ayuda que nos han brindado.*

## AGRADECIMIENTOS

*Rafael*

*Le agradezco a mi madre ella es la culpable de que me gradué y por su apoyo en cada momento de mi vida*

*A mi papa que hace 9 años que no se encuentra conmigo pero que me ha educado y permitido ser una mejor persona*

*A mi tía lleyly, mi primo Alex y Adrián, mis abuelos y a mi hermanito Leinier ustedes son las razones por la cual me levanto todos los días*

*A Yisel por ser amiga, madre y tutora*

*A todos los profesores que han influido en mi preparación en especial a mis tutores Osbel y Claudia por su ayuda en esta tesis*

*Uno muy especial a mi compañero de tesis por el estrés que cogimos, por ser mi amigo y por irle al peor equipo del mundo*

*A toda mi familia y amigos gracias por todo su apoyo brindado*

## **RESUMEN.**

En el centro de TLM perteneciente a la facultad 2 de la Universidad de las Ciencias Informáticas se desarrolló el Gestor de Recursos de Hardware y Software, encargado de monitorear los dispositivos conectados a una red de ordenadores y mostrar el resultado en una interfaz web. Por otro lado existe GLPI aplicación que permite mostrar los datos referentes a los inventarios de hardware y software. Este trabajo se basa en el desarrollo de un plugin que permita la integración de los datos gestionados por GRHS en GLPI.

Con la culminación del presente trabajo de diploma se puede visualizar los datos generados por GRHS en la interfaz web de GLPI de una forma más organizada. El desarrollo del plugin está basado en la metodología de desarrollo ágil XP, obteniendo los diferentes artefactos que se generan en sus diferentes fases. Se utiliza PHPStorm como IDE de desarrollo, seleccionando PHP como lenguaje de programación y PostgreSQL y MySQL como gestores de base de datos.

**Palabras claves: GLPI, GRHS, hardware, integración de datos, software**

## ÍNDICE

Resumen.....	5
Introducción.....	13
Capítulo 1 Fundamentación teórica del plugin para la integración de datos de GRHS a GLPI .....	17
1.1.    Introducción .....	17
1.2.    Conceptos asociados al dominio del problema.....	17
1.2.1. Hardware .....	17
1.2.2. Software.....	17
1.2.3 Plugin.....	17
1.2.4 Inventario .....	17
1.2.4 Integración de datos.....	17
1.3    Técnicas de integración de datos.....	18
1.3.1 Replicación de datos.....	18
1.3.2 Integración de información empresarial (EII).....	18
1.3.3 Integración de aplicaciones empresariales (EAI).....	19
1.3.4 Extracción, transformación y carga de datos (ETL).....	20
1.4    Herramientas para la integración y migración de datos de PostgreSQL a MySQL .....	20
1.4.1 DBConvert for MySQL & PostgreSQL .....	20
1.4.2 Navicat Premium 9.0 .....	21
1.4.3 ESF Database Migration Toolkit Professional Editon.....	21
1.4.4 Oracle Data Integrator .....	21
1.5    Sistema de Gestor de Recursos de Hardware y Software.....	22
1.6    GLPI.....	23
1.7    Metodología de desarrollo de software.....	24
1.8    Lenguajes de programación .....	25
1.8.1 Lenguajes de programación del lado del servidor.....	25
PHP 5 (Acrónimo de Hypertext Preprocessor) .....	25
1.8.1 Lenguajes de programación del lado del cliente.....	26
CSS 3 .....	26

JavaScript.....	26
HTML5.....	26
1.8.2 Framework .....	27
Ajax.....	27
1.8.3 Motor de plantilla Smarty.....	27
1.9 IDE de desarrollo .....	27
1.9.1 PHPStrom 7.....	27
1.11 Sistema gestor de bases de datos .....	28
1.11.1. PostgreSQL 9.1.....	28
1.11.2. MySQL 4.1.2.....	29
1.10 Servidor web .....	29
1.12 Conclusiones del capítulo 1.....	30
Capítulo 2.Características del Sistema y las fases de exploración, planificación, diseño del plugin para la integración de datos de GRHS a GLPI.....	31
2.1. Introducción.....	31
2.2. Propuesta de solución.....	31
Descripción del Módulo de Integración .....	32
Descripción del Módulo: Agentes.....	32
Descripción del Módulo: Incidencias .....	32
Descripción del Módulo: Acción .....	32
Descripción del Módulo: Trazas.....	32
2.3. Funcionalidades del sistema .....	32
Configurar integración.....	32
Mostrar el inventario .....	32
Mostrar las incidencias.....	33
Mostar las acciones .....	33
Mostrar los agentes .....	33
Mostrar las trazas .....	33
Mostrar usuarios y grupos del sistema.....	33
2.4. Características no funcionales.....	33



Usabilidad.....	33
Legales.....	33
Interfaz.....	34
Requisitos de software.....	34
2.5 Fase de exploración.....	34
2.6 Historia de usuarios.....	34
2.7 Fase de planificación.....	36
2.7.1 Estimación de esfuerzos por historia de usuario.....	36
2.7.2 Plan de iteraciones.....	36
2.7.3 Iteración 1.....	37
2.7.4 Iteración 2.....	37
2.7.5 Iteración 3.....	37
2.7.6 Duración de las iteraciones.....	37
2.7.7 Plan de entregas del producto.....	37
2.8 Fase de diseño.....	38
2.9 Arquitectura.....	38
2.9.1 Arquitectura cliente servidor.....	38
2.10 Patrón arquitectónico.....	39
2.10.1 Modelo vista controlador.....	39
2.10.2 Representación de las capas de la arquitectura.....	40
2.11 Patrones de diseño.....	42
Patrones Grasp.....	42
Patrones GoF.....	43
2.12 Tarjetas CRC.....	43
2.13 Conclusiones del capítulo 2.....	46
Capítulo 3. Implementación y pruebas del plugin para la integración de datos de GRHS a GLPI.....	48
3.1 Introducción.....	48
3.2 Iteración 1.....	48
3.3 Iteración 2.....	48
3.3 Iteración 3.....	49

3.4 Tareas de ingeniería .....	49
3.5 Pruebas.....	50
3.5.1 Pruebas unitarias .....	50
3.5.2 Pruebas de aceptación.....	52
Conclusiones del capítulo 3 .....	54
Conclusiones Generales .....	55
Recomendaciones .....	55
REFERENCIAS BIBLIOGRÁFICAS .....	56
BIBLIOGRAFÍA .....	59
Anexos .....	62
Anexo 1: Historias de usuarios.....	62
Anexo 2: Tareas de ingeniería. ....	65
Anexo 3: Pruebas de aceptación.....	71
Anexo 4: Pruebas unitarias. ....	77
Anexo 5: Prototipo de interfaz. ....	78

## ÍNDICE DE TABLAS

Tabla 1 Mostrar Módulo de integración.....	35
Tabla 2 Estimación de esfuerzo por historia de usuarios .....	36
Tabla 3 Plan de duración de las iteraciones .....	37
Tabla 4 Plan de entrega .....	37
Tabla 5 Tarjeta CRC PluginGrhsConfiguracion.....	43
Tabla 6 Tarjeta CRC PluginGrhsAccion. ....	44
Tabla 7 Tarjeta CRC PluginGrhsTraza. ....	44
Tabla 8 Tarjeta CRC PluginGrhsAgente.....	45
Tabla 9 Tarjeta CRC PluginGrhsIncidencia. ....	45
Tabla 10 Tarjeta CRC PluginGrhsSincronizadorOrdenadores. ....	45
Tabla 11 Tarjeta CRC PluginGrhsPgsqlDb.....	46
Tabla 12 UH Perteneiente a la Primera Iteración.....	48
Tabla 13: Perteneiente a la Segunda Iteración. ....	48
Tabla 14: UH Perteneiente a la Tercera Iteración.....	49
Tabla 15: UH Perteneiente a la UH configurar integración.....	49
Tabla 16: Caso de prueba de aceptación HU1-P1. ....	53
Tabla 17: Mostrar el inventario.....	62
Tabla 18: Mostrar las incidencias. ....	62
Tabla 19: Mostrar las acciones. ....	62
Tabla 20: Mostrar los agentes .....	63
Tabla 21: Mostrar las trazas. ....	63
Tabla 22: Mostrar usuarios y grupos del sistema.....	63
Tabla 23: Perteneiente a la UH Configurar integración.....	65
Tabla 24: Perteneiente a la UH Configurar integración.....	65
Tabla 25: Perteneiente a la UH Mostrar el inventario.....	65
Tabla 26: Perteneiente a la UH Mostrar las incidencias. ....	66
Tabla 27: Perteneiente a la UH Mostrar las acciones. ....	66
Tabla 28: Perteneiente a la UH Mostrar las acciones. ....	66

Tabla 29: Perteneciente a la UH Mostrar las acciones. ....	67
Tabla 30: Perteneciente a la UH Mostrar los agentes. ....	67
Tabla 31: Perteneciente a la UH Mostrar los agentes. ....	67
Tabla 32: Perteneciente a la UH Mostrar los agentes. ....	68
Tabla 33: Perteneciente a la UH Mostrar las trazas. ....	68
Tabla 34: Perteneciente a la UH Mostrar las trazas. ....	69
Tabla 35: Perteneciente a la UH Mostrar usuarios y grupos del sistema. ....	69
Tabla 36: Perteneciente a la UH Mostrar usuarios y grupos del sistema. ....	69
Tabla 37: Caso de prueba de aceptación HU1-P2. ....	71
Tabla 38: Caso de prueba de aceptación HU2-P1. ....	71
Tabla 39: Caso de prueba de aceptación HU2-P2. ....	71
Tabla 40: Caso de prueba de aceptación HU2-P3. ....	72
Tabla 41: Caso de prueba de aceptación HU3-P1. ....	72
Tabla 42: Caso de prueba de aceptación HU3-P2. ....	73
Tabla 43: Caso de prueba de aceptación HU3-P3. ....	73
Tabla 44: Caso de prueba de aceptación HU4-P1. ....	73
Tabla 45: Caso de prueba de aceptación HU6-P1. ....	74
Tabla 46: Caso de prueba de aceptación HU6-P2. ....	74
Tabla 47: Caso de prueba de aceptación HU6-P3. ....	75
Tabla 48: Caso de prueba de aceptación HU7-P1. ....	75
Tabla 49: Caso de prueba de aceptación HU7-P2. ....	75

## ÍNDICE DE FIGURAS

Ilustración 1: Propuesta del Sistema .....	31
Ilustración 2: Arquitectura Cliente-Servidor .....	39
Ilustración 3: Patrón Modelo-Vista-Controlador .....	40
Ilustración 4: Prueba unitaria modelo anti-virus.....	77
Ilustración 5: Prueba unitaria modelo controlador .....	77
Ilustración 6: Prueba unitaria modelo disco duro .....	77
Ilustración 7: Prototipo interfaz configuración general.....	78
Ilustración 8: Prototipo de interfaz configuración de la conexión con la BD .....	78

## **INTRODUCCIÓN.**

Desde el comienzo de la historia de la humanidad, uno de los factores que ha constituido un elemento vital para la evolución y desarrollo de la misma es la comunicación. En esencia, en una comunicación se transmite información desde una persona a otra o de un elemento cualquiera a otro (1).

En sus comienzos, las ciencias de las Telecomunicaciones y de la Informática desarrollaron sus caminos de forma independiente sin embargo a raíz de la evolución de las computadoras esto cambió. Así en las dos primeras generaciones de computadoras, estas eran máquinas capaces de ejecutar solamente un proceso y generalmente con dispositivos periféricos relativamente limitados, en la tercera generación de computadoras se dieron pasos de avances como fue la reducción de los equipos gracias al circuito integrado. Ya en la cuarta generación es cuando empieza a hacerse latente la confluencia entre las Telecomunicaciones y la Informática. Esta unión da origen al concepto de Teleinformática o Telemática.

La teleinformática puede definirse como el conjunto de máquinas, técnicas y métodos relacionados entre sí que permiten el proceso de datos a distancia y que participan en la convergencia entre las telecomunicaciones y la informática (2).

En los últimos años el número de equipos que forman parte de una red de ordenadores de una organización empresarial, centros tecnológicos y educativos por mencionar solo algunos ha ido en crecimiento a un ritmo acelerado. Esta situación hace necesario la utilización de herramientas de gestión que permitan entre otras funcionalidades inventariar los recursos de hardware y software. Las empresas son conscientes de la necesidad de manejar información precisa del control del mencionado inventario; tradicionalmente el inventario se realizaba de forma manual provocando imprecisiones e inconsistencias.

Son muchos los motivos por los cuales una organización necesita un inventario de los recursos de hardware y software. Actualmente existen una gran variedad de herramientas capaces de realizar dicha función, entre las que se pueden mencionar las siguientes; NetSupport DNA (Dynamic Network Administration), Total Network Inventory, VEO, OCS Inventory, Gestión Libre de Equipo informático (GLPI).

GLPI es una aplicación de código abierto, muy útil en empresas con varias sedes, para gestionar el inventario. Es solamente la interfaz de administración de un sistema de inventario, ya que toma sus datos de otras herramientas como OCS Inventory NG. Incluye datos administrativos

como pueden ser: períodos de validez de las licencias de software, garantías y tiempos de caducidad de las mismas, o datos de ubicación de los equipos (3). Una de sus principales ventajas es que permite la integración de otras herramientas capaces de realiza en inventario de hardware y software como OCS-Inventory NG a través de un sistema de plugin, entiéndase por plugin “programas auxiliares que permiten a sistemas mayores extender sus capacidades normales o aportar una función, generalmente muy específica, de manera que no se afecten las funciones ya existentes ni se complique el desarrollo del programa principal.”

La Universidad de las Ciencias Informáticas (UCI), fue creada por el Comandante en Jefe Fidel Castro Ruz en el fulgor de la Batalla de Ideas librada por el pueblo. Actualmente la UCI cuenta con diversas facultades las cuales tienen a su dirección centros productivos. El Centro de Telemática perteneciente a la Facultad 2, entre sus proyectos, desarrolló el Gestor de Recursos de Hardware y Software (GRHS).

GRHS es un sistema que funciona a partir de un agente instalado en las PC de la entidad, que envía la información a un gestor, donde se procesa y almacena. Además cuenta con una aplicación web que tributa a la visualización de los datos, la realización de reportes e informes, entre otras características. Esta aplicación permite una mayor facilidad para la gestión de los recursos así como la obtención de informes sobre información tecnológica, evita la duplicación y pérdida de la información, entre otros beneficios.

GLPI presenta un número de características las cuales no están presente en GRHS. El mismo permite realizar búsquedas por filtros, donde se especifica que tipo se quiere buscar, ya sea una placa base, un procesador, un disco duro o tarjetas de sonido solo por mencionar algunos, luego de seleccionar que tipo de componente existen otros filtros para realizar la búsqueda por nombre, fabricante etc. En lo que respecta a la información GLPI muestra la misma más organizada permitiendo ver los diferentes datos de los ordenadores a través de pestañas y sin la necesidad de redireccionar hacia otras páginas. Otras de las características existente en GLPI es que permite incorporar datos a sus interfaces, tienen un conjunto de datos los cuales no aparecen en GRHS entre ellos a los usuarios se le puede incorporar teléfono, móvil y matrícula esta función también es permitida para las demás información que se importa. Además a las computadoras inventariadas se le puede adicionar información referente al responsable como contacto y su número para contactarlo. Por último la aplicación permite eliminar componentes con solo señalar un monitor, una placa base u otro componente o usuario e ir a acciones y señalar elimi-

nar la aplicación borra al mismo de su base de datos. Por lo antes mencionado surge como **problema a resolver** ¿Cómo integrar los datos de GRHS a GLPI?

Definiendo como **objeto de estudio de esta investigación**: El proceso de integración de datos.

El **objetivo general es**: Desarrollar un plugin para GLPI que permita la integración de datos de GRHS hacia GLPI.

El **campo de acción** de la presente investigación se enmarca en la integración de datos de GRHS hacia GLPI.

Para darle cumplimiento al objetivo general se elaboraron las siguientes **Tareas de Investigación**:

- ✓ Elaboración del marco teórico de la investigación para definir conceptos fundamentales en la comprensión de la investigación.
- ✓ Realización del estudio del estado del arte de las técnicas de integración de datos para elegir el adecuado.
- ✓ Análisis de la metodología de desarrollo a utilizar para la descripción de los artefactos que se generan durante la implementación del plugin de integración para GRHS y GLPI.
- ✓ Selección de las tecnologías y herramientas para desarrollar plugin de integración para GRHS y GLPI.
- ✓ Identificación y análisis de las funcionalidades del plugin que permita la integración de GRHS y GLPI para su implementación.
- ✓ Estudio de los métodos de pruebas existentes para asegurar que el módulo cumpla con los requerimientos solicitados por el cliente.

Definiéndose como **idea a defender**: el desarrollo de un plugin que permita la integración de datos de GRHS hacia GLPI permitirá visualizar de forma más organizada la información de GRHS.

Para la realización del presente trabajo se utilizaron los siguientes métodos de la investigación científica:

**Métodos teóricos**: Permiten revelar las relaciones esenciales del objeto de la investigación, no observables directamente. Participan en la etapa de asimilación de hechos, fenómenos y procesos en la construcción del modelo e hipótesis de la investigación.



Entre los métodos teóricos estudiados se usaron los siguientes:

**Análisis-Sintésis:** Este método permitirá analizar las teorías presentadas en las bibliografías consultadas y extraer los elementos más importantes relacionados con el objeto de estudio, necesarios para dar solución al problema planteado.

**Histórico-lógico:** Permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Se utiliza este método ya que para darle cumplimiento a los objetivos se trazaron algunas tareas de la investigación como estudiar las herramientas informáticas y lenguajes que se utilizarán en el desarrollo de dicho plugin.

**Métodos empíricos:** Revela y explica las características fenomenológicas del objeto. Este se emplea fundamentalmente en la primera etapa de acumulación de información empírica y en la comprobación experimental de la hipótesis del trabajo.

Entre los métodos empíricos estudiados fueron usados los siguientes:

**Entrevista:** La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información. Se realizan entrevistas a los clientes para obtener información y entender mejor los procesos relacionados con la gestión de información en la aplicación GLPI.

El presente trabajo investigativo está compuesto por 4 capítulos:

**Capítulo 1. Fundamentación teórica del plugin para la integración de datos de GRHS a GLPI:** En este capítulo se realiza un estudio de las principales técnicas de integración de datos así como de las herramientas capaces de realizar el proceso de integración y migración de datos, se describen los principales conceptos a tratar en la investigación. Se define la metodología de desarrollo, herramientas y lenguajes de programación en la implementación del plugin.

**Capítulo 2. Características del sistema, exploración, planificación y diseño del plugin para la integración de datos de GRHS a GLPI:** En el presente capítulo se explica la solución al problema planteado, se exponen las características funcionales del plugin mediante las historias de usuarios (HU). Se estiman las HUs para su implementación, se selecciona la arquitectura del plugin, los patrones de diseños empleados y se redactan las tarjetas CRC.

**Capítulo 3. Implementación y prueba del plugin para la integración de datos de GRHS a GLPI:** En el presente capítulo se elaboran las tareas de ingeniería que responden a cada HU. Se realizan pruebas para verificar que el plugin cumpla con las necesidades del cliente.

# **CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA DEL PLUGIN PARA LA INTEGRACIÓN DE DATOS DE GRHS A GLPI**

## **1.1. Introducción**

A la hora de realizar una investigación para desarrollar una aplicación informática es importante realizar un estudio de las tecnologías y herramientas a usar en la misma, así como la metodología de desarrollo de software a usar y definir los conceptos fundamentales a tratar en la investigación. En el presente capítulo se realiza un estudio de las principales técnicas de integración de datos y herramientas capaces de realizar la integración y migración de datos, tecnología, herramientas y metodología de desarrollo de software usada para el desarrollo del plugin que permita la integración de datos de GRHS hacia GLPI.

## **1.2. Conceptos asociados al dominio del problema**

### **1.2.1. Hardware**

Conjunto de elementos físicos empleados para el tratamiento de datos. Corresponde a las partes tangibles de un dispositivo electrónico: componentes electrónicos, electromecánicos y mecánicos; cables, gabinetes, periféricos y cualquier otro elemento físico (4).

### **1.2.2. Software**

Conjunto de programas, procedimientos o reglas y eventualmente de documentación, relativo al funcionamiento de un conjunto de tratamientos de datos (4).

### **1.2.3 Plugin**

Los plug-in's son programas auxiliares que permiten a sistemas mayores extender sus capacidades normales o aportar una función, generalmente muy específica, de manera que no se afecten las funciones ya existentes ni se complique el desarrollo del programa principal (5).

### **1.2.4 Inventario**

Es la verificación periódica de las existencias de materiales, equipo, muebles e inmuebles con que cuenta una dependencia o entidad (6).

### **1.2.4 Integración de datos**

La integración de datos se puede definir como el proceso de combinar datos que residen en diferentes fuentes y permitirle al usuario final tener una vista unificada de todos sus datos. La habilidad de transformar datos interdepartamentales de fuentes heterogéneas en un plan de acción que se convertido en un reto y en una ventaja competitiva para compañías que requieran la integración de datos (7).

### **1.3 Técnicas de integración de datos**

En la actualidad existen diferentes técnicas de integración de herramientas las mismas permiten que diferentes sistemas informáticos puedan compartir funcionalidades. La integración de los sistemas informáticos puede traer varias ventajas como (8):

- ✓ Creación de actividades de negocio coordinadas, manejadas por diferentes personas de modo que se pueda formar un proceso de negocio unificado.
- ✓ Permite derribar las barreras entre las unidades de la organización.
- ✓ Reduce la duplicación de esfuerzos.
- ✓ Reduce los costos en desarrollo y mantenimiento de tecnologías.

A continuación se exponen algunos métodos de integración de herramientas.

#### **1.3.1 Replicación de datos**

La Replicación de bases de datos es una técnica de integración que se basa en la creación y mantenimiento de múltiples copias de una misma base de datos. En la mayoría de las implementaciones de replicación, un servidor mantiene la copia primaria de la base de datos y servidores adicionales mantienen las copias esclavas de la misma (White, Colin) (9).

No se decidió utilizar la replicación de datos ya que es una de las técnicas más antiguas de integración y presenta la desventaja que en caso que ocurra una falla en una de las réplicas restablecer el servicio podría llevar un gran consumo de tiempo, además cuando se realice una actualización en el estado de la aplicación, dicha actualización debe realizarse en todas las réplicas, y esto debe hacerse de una manera ordenada para que todas ellas mantengan un estado consistente.

#### **1.3.2 Integración de información empresarial (EII)**

EII permite la integración de datos procedentes de múltiples sistemas en una representación única, coherente y orientada hacia la visualización y manipulación de los datos. Los datos son agregados, reestructurados y renombrados, si es necesario, y luego presentados al usuario.

Proporcionar un enfoque unificado de los datos procedentes de sistemas disímiles conlleva un conjunto de requisitos y restricciones. En primer lugar, los datos deben ser accesibles en "tiempo real", esto significa que es necesario acceder a los sistemas directamente en lugar de acceder a datos previamente capturados que pueden no estar actualizados. En segundo lugar, la consistencia de los datos entre diferentes sistemas, por ejemplo su semántica y significado, de-

be resolverse pues diferentes sistemas pueden representar los mismos datos con diferentes nombres y formatos que son relevantes solo para sus respectivos usos, pero que requieren algún tipo de relación para el usuario final a fin de serles útil. EII, teniendo en cuenta los datos involucrados, elimina las duplicaciones, comprueba la validez, adapta los nombres, reformatea los valores, etc. Sin EII todo esto, por lo general, es realizado manualmente por el usuario de los sistemas (10).

Para lograr la integración real de información es necesaria la implementación de una arquitectura orientada a servicios (SOA). Esto proporciona un mecanismo de acceso universal a todos los sistemas a través de servicios Web y una representación de datos universal a través de XML. Una arquitectura SOA apoya la integración y el desarrollo de la estructuración de la información, los sistemas transaccionales y los sistemas basados en contenido.

No se escogió EII como técnica de integración porque a la hora de realizar una consulta a dos tablas muy grandes de orígenes de datos diferentes, cada tabla se convertirá a un XML aumentando su tamaño aproximadamente 3 veces, esto provoca dos grandes problemas, primeramente una gran cantidad de datos se mueve a través de la red y la segunda que la operación de unión no podrá ser optimizada (10).

### **1.3.3 Integración de aplicaciones empresariales (EAI)**

La Integración de aplicaciones empresariales (EAI por sus siglas en inglés) se define como el uso de principios de la arquitectura de software y sistemas informáticos para integrar un conjunto de informaciones y procesos en una organización. Las aplicaciones de gestión de información normalmente no se pueden comunicar entre sí para compartir datos o reglas de negocio. Por esta razón, a estas aplicaciones a veces se les denomina islas de automatización o silos de información. Esta falta de comunicación conduce a la ineficiencia debido a que se duplican datos idénticos en lugares diferentes provocando, además, que sea más difícil la automatización de procesos en apariencia simples (11).

Empleando EAI se logra la integración de aplicaciones en una única organización, permitiendo simplificar y automatizar los procesos de negocio evitando, al mismo tiempo, tener que realizar cambios radicales en las aplicaciones existentes y sus estructuras de datos. De esta forma, EAI representa el intercambio de datos y procesos de negocios relacionados entre cualquier aplicación u origen de datos en una organización.

Entre las ventajas de los EAI se encuentran:

- ✓ Posibilita el intercambio de información entre los sistemas en tiempo real.
- ✓ Permite encadenar los procesos de negocio y ayuda a incrementar la eficiencia organizacional.
- ✓ Mantiene la integridad de la información entre varios sistemas

Sin duda la técnica de EAI resulta una excelente opción para integrar aplicaciones en tiempos reales pero no fue seleccionada ya que presenta elevados costos de desarrollo para pequeños y medianos negocios, además su implementación requiere mucho tiempo y recursos.

### **1.3.4 Extracción, transformación y carga de datos (ETL)**

La técnica Extracción transformación y carga de datos (Extract, Transform and Load), como su nombre lo indica extrae información de un sistema fuente, transforma esos datos para satisfacer los requerimientos del negocio y carga el resultado en el sistema destino. Tanto la fuente como el destino son generalmente bases de datos y archivos. La transformación puede implicar la reestructuración y reconciliación del registro de datos, limpieza del contenido de datos (es decir, revisados por si existen discrepancias y eliminación de datos obviamente falsos), o agregación del contenido de datos. Esto sucede mediante una serie de procedimientos especiales que permiten obtener un formato unificado común y mejorado. Sólo después de la revisión y unificación de los datos estos son cargados. Esta técnica se encarga de la integración de datos, no de aplicaciones, y obtiene los datos directamente de las bases de datos (12).

Teniendo en cuenta las características antes menciona esta técnica de integración es la elegida para la creación del plugin ya que permite la extracción de datos de una base de datos hacia otra permitiendo la transformación y carga de la misma.

## **1.4 Herramientas para la integración y migración de datos de PostgreSQL a MySQL**

### **1.4.1 DBConvert for MySQL & PostgreSQL**

Es una herramienta de integración de base de datos que permite convertir de MySQL a PostgreSQL, MySQL a PostgreSQL Dump, MySQL a MySQL Dump, PostgreSQL a MySQL, entre otros. Opera con una base de datos completa o solo con las tablas necesarias, campos, índices y llaves foráneas. Esta herramienta no está basada en software libre, por lo que en su versión de evaluación solamente migran 50 registros por tabla (13).

### **1.4.2 Navicat Premium 9.0**

Es una herramienta que une a todo los miembros de la familia Navicat para administración de base de datos y desarrolladores. Permite que los usuarios se conecten a cuatro de las bases de datos más comunes: MySQL, Oracle, PostgreSQL y SQLite desde una única aplicación. Entre una de sus funcionalidades se encuentra la transferencia de datos entre los diferentes gestores a los que se conecta, permitiendo migrar bases de datos entre ellos (14). Sin embargo presenta algunas limitaciones entre las que destacan:

- ✓ No migran las llaves foráneas.
- ✓ No migran los índices.
- ✓ Es una herramienta propietaria.

### **1.4.3 ESF Database Migration Toolkit Professional Editon**

Es un programa que permite convertir entre diferentes formatos de base de datos. Puede conectarse directamente a MySQL, SQL Server, PostgreSQL, SQLite, Oracle, Access, Excel, Paradox, Lotus, dBase, Visual FoxPro, texto y otros formatos y convertir entre estos formatos (SQL Server a MySQL, MySQL a Acces, entre otros). Permite conectarse a la base de datos origen, seleccionar las tablas y a continuación, convertir y copiar al destino. También ofrece todas las opciones de conversión necesarias teniendo en cuenta las peculiaridades de los dos formatos de entrada y salida de la base de datos, además de incluir compatibilidad para las claves primarias de las tablas, índices e identificación automática. Es compatible con todos los caracteres Unicode (UTF-8, LATIN, CP1250, ASCII, entre otros). Esta herramienta al migrar la estructura de la base de datos no mantiene los campos que se encuentran con letras mayúsculas, cambiándolos a minúscula. Debido a su carácter propietario, para aquellos usuarios que no posean licencias, la herramienta migra los campos de tipo cadena una T en la primera letra (15).

### **1.4.4 Oracle Data Integrator**

Es una plataforma de integración completa que cubre los requisitos de integración de datos. Maneja alto volumen, provee lotes de alto desempeño a procesos dirigidos a eventos, a servicios de integración basados en una arquitectura orientada a servicios y con la capacidad de procesar eventos en tiempo real (16).

Oracle Data Integrator maneja múltiples necesidades empresariales referentes a la integración de datos:

- ✓ Data warehousing e inteligencia de negocios tiene la capacidad de manejar grandes volúmenes de datos con un desempeño óptimo para cargar data warehouse y data mart. Maneja cargas incrementales, integridad de datos, reglas de negocio y consistencia.
- ✓ Arquitectura orientada a servicios provee la funcionalidad de invocar servicios externos para propósitos de integración e implementar servicios de integración y transformación integrados a una arquitectura orientada a servicios.
- ✓ Master data management es una combinación de aplicaciones y tecnologías que consolidan, limpian, mejora los datos maestros de la empresa y los sincroniza con aplicaciones, procesos de negocio y herramientas analíticas como Oracle BIEE+.

El Oracle Data Integrator va más allá de un ETL tradicional, donde se extraen los datos de diferentes fuentes, se transforman en el proceso y luego se cargan a su destino. La funcionalidad ELT cambia en el donde y como la transformación se ejecuta y se apalanca del motor de base de datos.

Después de realizar un estudio de las herramientas expuestas capaces de realizar la integración y migración de datos se arriba a la conclusión que ninguna ofrece una solución al problema planteado ya que presentan algunas desventajas, no son capaces de integrar y migrar toda la información de las bases de datos, algunas no incluye las llaves foráneas ni índices, además varias son herramientas propietarias y dirigida más a la elaboración de un data warehouse.

Por lo antes planteado se decidió desarrollar un plugin que posibilite integrar la información que genera GRHS y ser mostrada en la aplicación GLPI.

### **1.5 Sistema de Gestor de Recursos de Hardware y Software.**

GRHS es un sistema que permite la captura, análisis y consulta de la información de los activos informáticos en una red. Es una plataforma cliente-servidor, que posee un diseño modular, y utiliza el protocolo HTTPS para la comunicación entre estos. Posee, además versiones para Windows y GNU/Linux. Es una herramienta desarrollada con tecnologías libres. Aporta una gran cantidad de información sobre el inventario de los recursos de hardware y software de una red ya que integra varias bibliotecas para la obtención de la misma. Permite la ejecución de acciones posibilitando tomar el control de los ordenadores en caso de incidencia. (17)

Este sistema está compuesto por una aplicación servidor encargada de procesar y almacenar la información obtenida por los agentes, una aplicación web donde se mostrará toda la información obtenida por el sistema agente-gestor y donde se podrán configurar las alarmas que se

activarán ante las diferentes incidencias y a quién se le enviará la información de la misma; y una aplicación agente o cliente GRHS encargada de la obtención de la información del hardware y el software de las computadoras donde se encuentre instalado, así como detectar algún cambio en la configuración de dicho computador, informarlo al gestor y realizar alguna acción según corresponda.

La implantación de esta solución permite al centro poseer un sistema para el proceso de captura, análisis y consulta de la información de los activos informáticos de su red de computadoras, para un mejor control de los recursos y facilitar la toma de decisiones respecto a los mismos.  
(17)

## **1.6 GLPI**

GLPI es una aplicación de código abierto, muy útil en empresas con varias sedes, para gestionar el inventario. Es solamente la interfaz de administración de un sistema de inventario, ya que toma sus datos de otras herramientas como OCS Inventory NG. Incluye datos administrativos como pueden ser: períodos de validez de las licencias de software, garantías y tiempos de caducidad de las mismas, o datos de ubicación de los equipos. Posibilita efectuar búsquedas parametrizadas y ordenadas sobre inventarios a computadoras, material de red, impresoras, monitores, periféricos externos y software. Soporta las plataformas GNU/Linux y Windows(3).

Aplicación totalmente web que ataca los principales problemas de la gestión del inventario informático: la administración de los recursos de hardware, software, usuarios, suministros e incidencias. La sección de Inventario permite listar y consultar el inventario de hardware y software del parque informático. Posibilita efectuar búsquedas parametrizadas y ordenadas, sobre uno de los tipos de inventario (computadores, material de red, impresoras, monitores, periféricos externos, software); permite desplegar la ficha que contiene los datos particulares de cada elemento inventariado, o agregar un elemento. Luego da la opción de modificar o eliminar cada elemento.

La gestión y la historia de las acciones de mantenimiento y de los procedimientos del inventario son elementos que este maneja. Aplicación dinámica y está conectado directamente a los usuarios brindándole una interfaz donde pueden escribir solicitudes a los técnicos para el servicio de mantenimiento y la indexación de un problema encontrado con uno de los recursos técnicos a los que tienen acceso.

Características fundamentales:



- ✓ Multilingüe (gestión de 22 idiomas disponibles).
- ✓ Gestión de usuarios.
- ✓ Módulo de búsquedas complejas.
- ✓ Menús configurables.
- ✓ Importa datos de uno o varios servidores.

## **1.7 Metodología de desarrollo de software**

El desarrollo de software no es una tarea fácil. Prueba de ello son las diferentes metodologías que se utilizan en la actualidad, las mismas guían el proceso de desarrollo de un software. Por una parte están aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Por otra parte están las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software 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 pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generan un amplio debate entre sus seguidores y quienes por convencimiento no las ven como alternativa para las metodologías tradicionales (18).

Debido a que se necesita estar preparado para cambios durante el proyecto y que se cuenta con un grupo de desarrollo pequeño, donde existen pocos roles, se generan pocos artefactos y se desea hacer menos énfasis en la arquitectura del software, se ha tomado la decisión de utilizar para el desarrollo del plugin una metodología ágil.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Esta metodología se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes La satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las

necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. (18)

Se selecciona la metodología ágil de desarrollo XP porque:

- ✓ Los programadores trabajan en parejas, lo cual disminuye el número de errores durante la programación.
- ✓ Los requisitos pueden cambiar durante el desarrollo del subsistema.
- ✓ Cada integrante del equipo de desarrollo debe conocer cualquier parte del código.
- ✓ Se cuenta con poco tiempo para el desarrollo del plugin.
- ✓ El cliente forma parte del equipo de desarrollo, lo cual ayuda a corregir a tiempo los errores que puedan surgir.

## **1.8 Lenguajes de programación**

Los lenguajes de programación son un conjunto de reglas, herramientas y condiciones que nos permiten crear programas o aplicaciones dentro de una computadora. (19).

### **1.8.1 Lenguajes de programación del lado del servidor**

Los lenguajes del lado del servidor proporcionan un entorno rápido de creación de scripts y soporte para los estándares más importantes. Además de las aplicaciones tradicionales de bases de datos, las aplicaciones dinámicas de Internet prometen la integración de las comunicaciones bidireccionales y los datos en tiempo real en las mismas, en este sentido, PHP se perfila como un ejemplo necesario del lado del servidor.

#### **PHP 5 (Acrónimo de Hypertext Preprocessor)**

PHP es un lenguaje "del lado del servidor" (esto significa que PHP funciona en un servidor remoto que procesa la página web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas web dinámicas. Permite una serie de funcionalidades tan extraordinarias que se ha convertido en el favorito de millones de programadores en todo el mundo (20).

Características de PHP:

- ✓ Dispone de una conexión propia a varios sistemas de base de datos como: MySQL, PostgreSQL, etc.
- ✓ Gratuito, versátil y multiplataforma.

- ✓ Gran popularidad. Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código.
- ✓ Enorme eficiencia. Con escaso mantenimiento y un servidor gratuito (en este caso, Apache), puede soportar sin problema millones de visitas diarias.
- ✓ Permite usar técnicas de programación orientada a objetos.

Se decidió utilizar el lenguaje antes mencionado por sus características y porque la aplicación GLPI está implementada en PHP.

### **1.8.1 Lenguajes de programación del lado del cliente**

Las tecnologías del lado del cliente están orientadas preferentemente, como su nombre indica, para ejecutarse en los puestos cliente.

#### **CSS 3**

Son las siglas de Cascading Style Sheets, en español Hojas de Estilo en Cascada. Permite crear páginas web. Gracias a esta los desarrolladores se involucran en los resultados finales de la página, pudiendo incluir márgenes, tipos de letra, fondos, colores e incluso pueden definir los estilos en un archivo externo. Si en algún momento se quiere cambiar alguno de los estilos, automáticamente se actualizan todas las páginas vinculadas al sitio (21) .

Este estilo propone varias ventajas dentro de las cuales cabe destacar:

- ✓ Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- ✓ El documento HTML en sí mismo es más claro de entender y se consigue reducir su tamaño (siempre y cuando no se utilice estilo en línea).

#### **JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (22).

#### **HTML5**

HTML 5 es una nueva versión del lenguaje de marcas de hipertexto (HTML por sus siglas en inglés), ofrece herramientas para la administración efectiva de datos. Facilita el desarrollo de aplicaciones para diferentes navegadores web. También introduce nuevas etiquetas y mejoras,

controles de formulario, soporte de bases de datos y una velocidad de procesamiento significativamente más rápida (23).

### **1.8.2 Framework**

Una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación (24).

#### **Ajax**

El término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML". Se decidió usar AJAX ya que las aplicaciones construidas con el mismo 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. AJAX es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. Java Script es el lenguaje en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante una petición en segundo plano que viene con formato XML (25).

### **1.8.3 Motor de plantilla Smarty**

Smarty es un motor de plantillas para PHP. Más específicamente, esta herramienta facilita la manera de separar la aplicación lógica y el contenido en la presentación. La mejor descripción está en una situación donde la aplicación del programador y la plantilla del diseñador juegan diferentes roles, o en la mayoría de los casos no la misma persona (26).

## **1.9 IDE de desarrollo**

Un entorno de desarrollo integrado o IDE (Integrated Development Environment), es un conjunto de herramientas de programación integradas en un único entorno, entre las que destacan el editor de código, un compilador, un depurador, una consola y en algún caso un constructor de interfaz gráfica. Este puede servir a un único lenguaje de programación o varios (27).

### **1.9.1 PHPStrom 7**

PHPStrom es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de aplicación más completos de la actualidad, permite editar código no solo de lenguaje de programación PHP como lo indica su nombre. Vale destacar que PHPStrom permite la visualización del código en la misma interfaz del IDE y es compatible hasta con navegadores (28).

Entre sus principales funcionalidades se encuentran:

- ✓ Proporciona un fácil autocompletado de código.
- ✓ Soporta el trabajo con PHP 5.5
- ✓ Sintaxis abreviada.

## **1.11 Sistema gestor de bases de datos**

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario, y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y un lenguaje de consulta (29).

### **1.11.1. PostgreSQL 9.1**

El Sistema Gestor de Base de Datos Relacionales Orientados a Objetos conocido como PostgreSQL es uno de los sistemas gestores de base datos de código abierto más avanzado de hoy en día ofreciendo control de concurrencia multiversión, soportando casi todas las sintaxis SQL( incluyendo consultas , transacciones y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación entre los que se encuentran C, C++, PHP, Java, Perl, Tcl y Python. (30).

Características de PostgreSQL:

- ✓ Soporta distintos tipos de datos, además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP.), cadenas de bits, etc.
- ✓ Incorpora una estructura de datos array.
- ✓ Permite la declaración de funciones propias.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Se decidió utilizar PostgreSQL porque la base de datos de GRHS trabaja con ese gestor.

### 1.11.2. MySQL 4.1.2

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones, MySQL es usado por muchos sitios web grandes y populares, MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones, consta de varias características como son (31):

- ✓ Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- ✓ Disponibilidad en gran cantidad de plataformas y sistemas.
- ✓ Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones.
- ✓ Transacciones y claves foráneas.
- ✓ Conectividad segura.
- ✓ Replicación.
- ✓ Búsqueda de indexación de campos de texto.

Se decidió utilizar MySQL ya que la base de datos de GLPI trabaja con este mismo gestor de bases de datos.

### 1.10 Servidor web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y 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 (32).

### Apache 2

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que

ayude en su configuración. Entre sus más interesantes ventajas se puede mencionar su carácter modular, lo cual permite incrementar o quitar los módulos que posee en función de las necesidades específicas del usuario. Es además multi-plataforma lo que implica que podamos hacer uso de él en varios sistemas operativos (33).

Se utiliza como servidor web Apache ya que es el usado por la aplicación GLPI.

### **1.12 Conclusiones del capítulo 1**

En el presente capítulo se definieron los principales conceptos de la investigación científica dejando así las bases teóricas del trabajo de diploma. Se realizó el estado del arte de las herramientas capaces de realizar el proceso de integración y migración de datos arribando a la conclusión que ninguna ofrece una respuesta al problema planteado. También se estudiaron las técnicas de integración de datos. Se seleccionó XP como metodología de desarrollo de software. El IDE de programación PHPStorm por las características que presenta, PHP como lenguaje de programación ya que GLPI se encuentra implementado en dicho lenguaje y PostgreSQL y MySQL como gestores de base de datos por ser las bases de datos de GRHS y GLPI.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA Y LAS FASES DE EXPLORACIÓN, PLANIFICACIÓN, DISEÑO DEL PLUGIN PARA LA INTEGRACIÓN DE DATOS DE GRHS A GLPI

### 2.1. Introducción

En este capítulo se explica la propuesta de solución al problema planteado, haciendo énfasis en el proceso de integración de datos de GRHS a GLPI. Se mencionan las funcionalidades definidas por el cliente. Además se generan las historias de usuario artefacto que corresponde a la fase de exploración de la metodología de desarrollo de software seleccionada. Se aborda la fase de planificación donde se elabora el plan de entrega del producto. También se trata la fase de diseño donde se expone la arquitectura, los patrones de diseño y las tarjetas CRC.

### 2.2. Propuesta de solución

Para darle solución al problema planteado se realiza un plugin que permita integrar los datos de GRHS a GLPI. Desde la máquina cliente se realizará una petición al servidor de GLPI, que no mostrará ningún dato hasta que sea instalado y configurado el plugin. Después de realizar las acciones antes mencionadas se establecerá la conexión con la base de datos de GRHS, se importaran los datos y se mostrará la información almacenada en el cliente de GLPI. El plugin posibilita visualizar los datos generados por GRHS en la interfaz de GLPI. En la siguiente imagen se ilustra lo antes planteado.



Ilustración 1: Propuesta del Sistema



### **Descripción del Módulo de Integración**

Se encarga de establecer la conexión con la base de datos de GRHS además de importar los datos de GRHS y limpiar los enlaces de GLPI y GRHS.

### **Descripción del Módulo: Agentes**

Un agente en GRHS es una representación lógica de las computadoras inventariadas, el módulo se muestra una tabla con información acerca de las estadísticas del total de host o computadoras y en otras dos tablas separadas se mostrarán los conectores activos e inactivos.

### **Descripción del Módulo: Incidencias**

Entiéndase como incidencia a un cambio no autorizado por la entidad, en el módulo de incidencias se mostrará un conjunto de datos relacionados con las incidencias ocurridas en una computadora teniendo en cuenta el tipo de identificador, componente, tipo de incidencia, estado, nivel, fecha y localización de la misma.

### **Descripción del Módulo: Acción**

Muestra las estadísticas con respecto al total de alarmas y total de controles entiéndase que una alarma en GRHS son los mensajes enviados cuando se detectan incidencias y los controles son acciones ejecutadas por el administrador de la red sobre las computadoras inventariadas.

### **Descripción del Módulo: Trazas**

En dicho módulo podrá visualizar las peticiones de registro las cuales están compuestas por el identificador, la fecha, dirección ip, puerto y plataforma para así llevar un control de los mismos. En otra tabla podrá ver la comunicación con Agentes donde se registrará la fecha, dirección ip, puerto y parámetros.

## **2.3. Funcionalidades del sistema**

### **Configurar integración**

- ✓ Configurar servicio.
- ✓ Importar bases de datos de GRHS.
- ✓ Limpiar enlace de GRHS y GLPI.

### **Mostrar el inventario**

- ✓ Mostrar inventario.

**Mostrar las incidencias**

- ✓ Mostrar incidencias.

**Mostrar las acciones**

- ✓ Mostrar total de alarmas y controles.
- ✓ Mostrar alarmas.
- ✓ Mostrar controles.

**Mostrar los agentes**

- ✓ Mostrar estadísticas.
- ✓ Mostrar colectores activos.
- ✓ Mostrar colectores inactivos.

**Mostrar las trazas**

- ✓ Mostrar peticiones de registro.
- ✓ Mostrar comunicación con agentes.

**Mostrar usuarios y grupos del sistema**

- ✓ Mostrar usuarios de GRHS.
- ✓ Mostrar grupos de GRHS.

**2.4. Características no funcionales.**

Las características no funcionales no es más que las cualidades o propiedades que el software o producto debe tener para su correcto funcionamiento. Además de recibir una preparación previa de como interactuar con el mismo.

**Usabilidad**

El usuario que interactuará con el sistema deberá poseer un conocimiento medio de la especialidad de informática.

**Legales**

La Universidad de las Ciencias Informáticas posee la propiedad intelectual del plugin desarrollado.

## Interfaz

El plugin desarrollado presenta la interfaz de la aplicación web GLPI donde se muestra la información generada por GRHS.

## Requisitos de software

Se requiere un servidor Apache con soporte para PHP 4 como mínimo, aunque se recomienda PHP 5, así como una base de datos de MySQL 4.1.2 y PostgreSQL 9.1, es compatible con navegadores como Mozilla Firefox 30.0+, Chrome 20.0+ e Internet Explorer 6+.

## 2.5 Fase de exploración

En esta fase los clientes realizan las historias de usuario que desean que estén para la primera entrega. Cada historia describe una de las funcionalidades que el programa tendrá. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, las tecnologías y las prácticas a ser utilizadas durante el proyecto. En algunos casos se utiliza un prototipo para testear la nueva tecnología y explorar algunos aspectos de la arquitectura a ser implementada. La duración de esta fase puede extenderse desde unas pocas semanas a varios meses dependiendo de la adaptación del equipo de desarrollo (34).

## 2.6 Historia de usuarios

Las historias de usuario representan una breve descripción del comportamiento del sistema, se realiza una por cada característica principal del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de pruebas de aceptación(22).

Las HU serán representadas mediante tablas divididas por las siguientes secciones:

- ✓ **Número:** Número de la historia de usuario incremental en el tiempo.
- ✓ **Nombre de Historia de Usuario:** El nombre de la historia de usuario.
- ✓ **Usuario:** Persona que va a interactuar con la HU.
- ✓ **Programadores Responsables:** Involucrados en el desarrollo de la HU.
- ✓ **Iteración Asignada:** número de la iteración.
- ✓ **Prioridad en Negocio:**

Alta: Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

Media: Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

Baja. Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

### Riesgo en desarrollo:

Alta: Las historias de usuarios que en caso de tener algún error de implementación, puedan afectar la disponibilidad del sistema, tienen riesgo de desarrollo alto.

Media: Las historias de usuarios que puedan presentar errores y retrasan la entrega de la versión, tienen riesgo de desarrollo medio.

Baja: Las historias de usuario que puedan presentar errores, pero estos son tratados con facilidad y no afectan en desarrollo del proyecto, tienen riesgo de desarrollo bajo.

- ✓ **Puntos estimados:** Tiempo estimado que se demorará el desarrollo de la HU.
- ✓ **Puntos reales:** Tiempo que se demoró en realidad el desarrollo de la HU.
- ✓ **Descripción:** Breve descripción de la HU.
- ✓ **Observaciones:** Señalamiento o advertencia del sistema.

A continuación un ejemplo de una historia de usuario perteneciente al desarrollo del plugin.

**Tabla 1 Mostrar Módulo de integración**

Historia de Usuario:	
<b>Número:</b> 2	<b>Nombre de la Historia de Usuario:</b> Configurar integración
<b>Riesgo en Desarrollo :</b> Alto	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Programador Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Permite establecer la conexión con la base de datos de GRHS.	
<b>Observaciones:</b> En esta interfaz se debe especificar puerto, base de datos y contraseña de la misma.	

El resto de las historias de usuarios pueden consultarse en el [Anexo I.](#)

## 2.7 Fase de planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses (35). Esta fase dura unos pocos días. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad de proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

### 2.7.1 Estimación de esfuerzos por historia de usuario

Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las HU generalmente valen de 1 a 3 puntos.

**Tabla 2 Estimación de esfuerzo por historia de usuarios**

No.HU	Historia de Usuarios (HU)	Puntos estimados
1	Configurar integración	2
2	Mostrar el inventario	1
3	Mostrar las incidencias	1
4	Mostrar las acciones	2
5	Mostrar los agentes	1
6	Mostrar las trazas	1
7	Mostrar usuarios y grupos del sistema	1

### 2.7.2 Plan de iteraciones

Luego de haber visto el esfuerzo para cada una de las historias de usuario se procede a la etapa de planificación de la implementación de la aplicación. Para dar cumplimiento a dicha etapa hemos decidido dividir la misma en 3 iteraciones que se explicarán a continuación.

### 2.7.3 Iteración 1

En esta iteración se tiene en cuenta las historias de usuarios que tengan prioridad del negocio alta y se obtendrá la primera versión de la aplicación donde el cliente podrá probar algunas funcionalidades como, configurar integración, mostrar el inventario e incidencias.

### 2.7.4 Iteración 2

En esta segunda iteración se implementarán las historias de usuarios de prioridad media. Con la implementación de las historias de usuario se contará con un producto más completo el cual será presentado al cliente para su aprobación y posibles cambios en el mismo.

### 2.7.5 Iteración 3

En la presente iteración se implementarán las historias de usuario con prioridad baja logrando una versión final del producto. El mismo será presentado al cliente para su aprobación y despliegue.

### 2.7.6 Duración de las iteraciones

En el siguiente plan se establecerá las historias de usuario que deben de ser implementada con la iteración que le corresponde y se muestra el tiempo de duración de cada iteración.

**Tabla 3 Plan de duración de las iteraciones**

Iteración	Historia de usuario a implementar	Duración
Iteración 1	Configurar integración Mostrar el inventario Mostrar las incidencias	4 semanas
Iteración 2	Mostrar las acciones Mostrar los agentes	3 semanas
Iteración 3	Mostrar las trazas Mostrar usuarios y grupos del sistema	2 semanas

### 2.7.7 Plan de entregas del producto

El plan de entrega está compuesto por las iteraciones de no más de 3 semanas. A continuación veremos el plan de entrega para el sistema de integración de GRHS y GLPI.

**Tabla 4 Plan de entrega**

Sistema	Fin Iteración 1	Fin Iteración 2	Fin iteración 3
Integración de GRHS y GLPI	27/02/2015	20/03/2015	03/04/2015

## 2.8 Fase de diseño

Según la metodología XP sugiere que se deben realizar diseños simples y sencillos. Hay que hacerlo lo más sencillo posible para procurar conseguir un diseño entendible a la hora de programarlo lo que traerá un considerable ahorro de tiempo en la implementación. Esta metodología estimula el uso de tarjetas CRC.

## 2.9 Arquitectura

Una arquitectura de software define la estructura del sistema, la cual adquiere gran importancia debido a que las representaciones de las arquitecturas de software permiten la comunicación entre todas las partes interesadas en el desarrollo de un sistema de cómputo; constituye en modelo comprensible de cómo está estructurado el sistema (36).

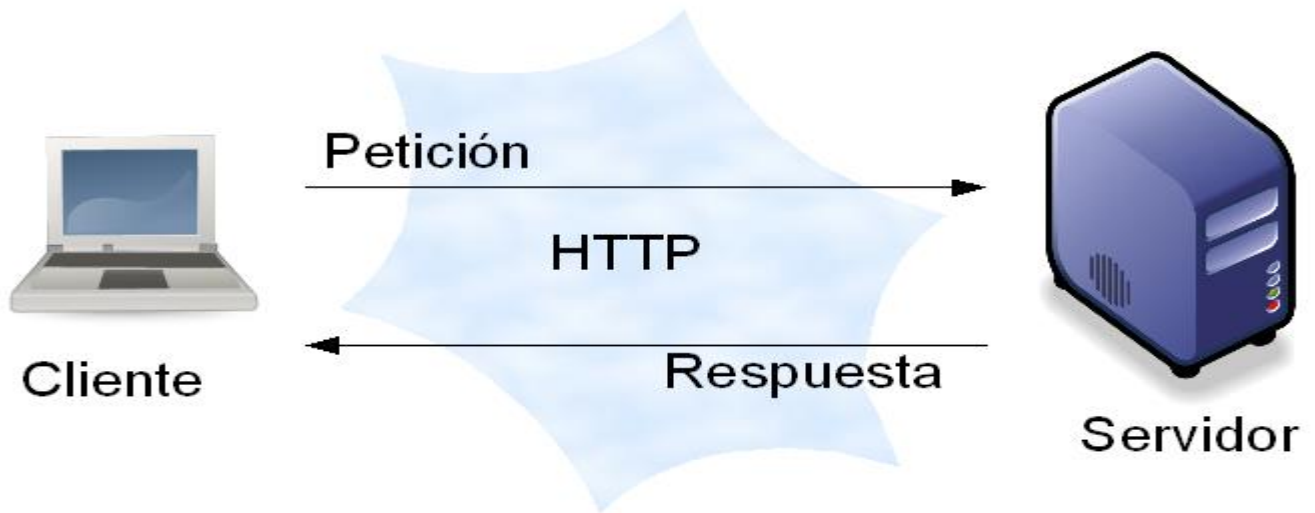
### 2.9.1 Arquitectura cliente servidor

Los clientes (o programas que representan entidades que necesitan servicios) y los servidores (o programas que proporcionan servicios) son objetos separados desde un punto de vista lógico y que se comunican a través de una red de comunicaciones para realizar una o varias tareas de forma conjunta(25).

Principales Características:

- ✓ Protocolos asimétricos: hay una relación muchos a uno entre los clientes y un servidor. Los Clientes siempre inician un diálogo mediante la solicitud de un servicio. Los Servidores esperan pasivamente por las solicitudes de los clientes.
- ✓ Encapsulación de servicios: el servidor es un especialista, cuando se le entrega un mensaje solicitando un servicio, él determina cómo conseguir hacer el trabajo. Los servidores se pueden actualizar sin afectar a los clientes en tanto que la interfaz pública de mensajes que se utilice por ambos lados, permanezca sin cambio.
- ✓ Integridad: el código y los datos de un servidor se mantienen centralizados, lo que origina que el mantenimiento sea más barato y la protección de la integridad de datos com-

partidos. Al mismo tiempo, los clientes mantienen su independencia. A continuación se muestra en la ilustración la mencionada arquitectura para una mejor comprensión.



**Ilustración 2: Arquitectura Cliente-Servidor**

Para el desarrollo del plugin se emplea la arquitectura cliente-servidor pues cada usuario autorizado podrá acceder a la aplicación y podrá operar sobre el mismo realizando peticiones al servidor donde se encontrará alojado dicho sistema.

## **2.10 Patrón arquitectónico**

Un patrón arquitectónico de software describe un problema particular y recurrente del diseño, que surge en un contexto específico y presenta un esquema genérico y probado de su solución (37). Representa un diseño organizativo estructural fundamental para guiar el desarrollo del software. Para el desarrollo del subsistema el patrón a seguir es el Modelo –Vista –Controlador (MVC según sus siglas en inglés).

### **2.10.1 Modelo vista controlador**

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la ac-



tualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo (38).

La capa modelo es la encargada de:

- ✓ Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- ✓ Define reglas de negocio (la funcionalidad del sistema).
- ✓ Notificará a las vistas los cambios que en los datos pueda producir un agente externo si se está ante un modelo activo.

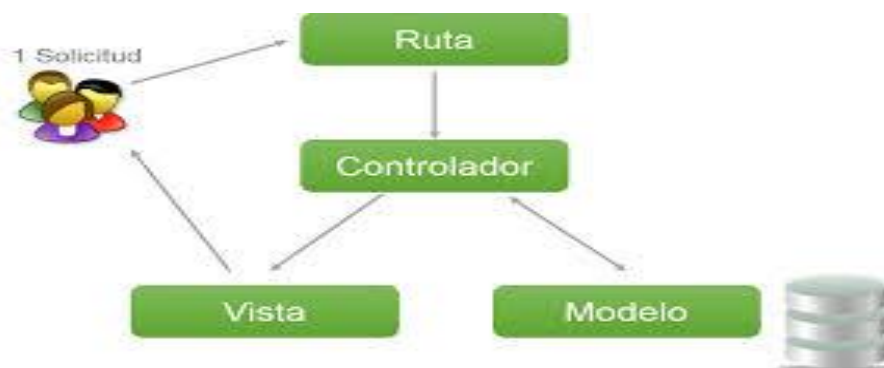
Mientras que las vistas tienen como responsabilidad:

- ✓ Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.
- ✓ Tienen un registro de su controlador asociado.

Algunas de las funcionalidades de las clases controladoras son:

- ✓ Recibir los eventos de entrada.
- ✓ Contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas.

Para una mejor comprensión del patrón modelo-vista-controlador a continuación una ilustración con su funcionamiento.



**Ilustración 3: Patrón Modelo-Vista-Controlador**

### 2.10.2 Representación de las capas de la arquitectura

A continuación se describen las clases situadas en cada una de las capas de la arquitectura del patrón MVC.

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo. Seguidamente las clases pertenecientes a la capa modelo.

- ✓ antivirus.php
- ✓ controlador.php
- ✓ discoDuro.php
- ✓ fabricante.php
- ✓ memoria.php
- ✓ monitor. php

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo. A continuación se pueden ver las clases pertenecientes a la capa vista.

- ✓ configuración.tpl
- ✓ vistaTablaPaginado.tpl
- ✓ vistaTabla.tpl

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. Seguidamente las clases pertenecientes a la capa controladora.

- ✓ acción.form.php
- ✓ agentes.form.php
- ✓ configuracion.form.php

- ✓ incidencias.form.php
- ✓ trazas.form.php

## 2.11 Patrones de diseño

Los patrones de diseño son un conjunto de prácticas de óptimo diseño que se utilizan para abordar problemas recurrentes en la programación orientada a objetos. Un patrón de diseño puede considerarse como un documento que define una estructura de clases que aborda una situación particular (39).

### Patrones Grasp

Los patrones GRASP son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Más explícitamente los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (39).

A continuación algunos de los patrones usados en la implementación de la aplicación.

#### ✓ **Experto**

**Problema:** ¿Cuál es el principio más básico para añadir responsabilidades en una clase?

**Solución:** Asignar responsabilidades al experto de la información, es decir, a la clase que tiene la información necesaria para llevar la tarea a cabo.

#### ✓ **Creador**

**Problema:** ¿Quién debe ser responsable en la creación de una nueva instancia de una clase?

**Solución:** Una clase B tiene la responsabilidad para crear una instancia de la clase A si:

- B agrega objetos de A.
- B contiene objetos de A.
- B almacena objetos de A.
- B usa objetos de A.
- B tiene los datos necesarios para inicializar a A cuando este es creado.

#### ✓ **Alta cohesión**

**Problema:** ¿Cómo lograr que la complejidad sea lo más manejable posible? Alta cohesión: lo relacionadas que están las responsabilidades de una clase, o una clase con responsabilidades altamente relacionadas y que no lleva a cabo gran cantidad de trabajo.

**Solución:** Asignar responsabilidades procurando que la cohesión sea lo más alta posible.

#### ✓ **Bajo acoplamiento:**

**Problema:** ¿Cómo soportar baja dependencia e incrementar la reutilización? Acoplamiento indica que tan fuertemente está conectada una clase con otra, tiene conocimiento de, o influye sobre otra clase, una clase con bajo acoplamiento no depende de otras clases.

**Solución:** Asignar responsabilidades de tal manera que el acoplamiento sea el menor posible.

#### ✓ **Controlador**

**Problema:** ¿Quién debe manejar eventos del sistema? Un controlador es un objeto responsable del manejo de los eventos del sistema, que no pertenece a la interfaz del usuario, el controlador recibe la solicitud del servicio desde la capa GUI (Interfaz Gráfica de Usuario) y coordina su realización delegando a otros objetos.

**Solución:** Responsabilidades para el manejo de mensajes de eventos del sistema a una clase que:

- Representa al conjunto del sistema o negocio (Controlador Fachada).
- Representa algo del mundo real que está activo (Controlador de Roles).
- Representa un administrador artificioso para todos los eventos del sistema (Controlador Caso de Uso).

### **Patrones GoF**

Por sus siglas en inglés Gang of Four. Llamados así debido a que fueron cuatro autores los que escribieron el libro “Design Patterns” que ilustra sus funciones.

**Singleton:** Patrón creacional diseñado para restringir la creación de objetos pertenecientes a una clase. Su objetivo es garantizar que una determinada clase solo tenga una instancia y proporcionar un punto de acceso global a esta. Es un patrón muy sencillo de diseñar y a menudo es implementado por otros patrones.

## **2.12 Tarjetas CRC**

El uso de las tarjetas CRC (Clases, Responsabilidad y Colaboración) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas CRC representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

### **Tabla 5 Tarjeta CRC PluginGrhsConfiguracion.**

<b>Tarjeta CRC:</b>	
<b>Clase:</b> PluginGrhsConfiguracion	
<b>Responsabilidad:</b>	<b>Colaboraciones:</b>
<ul style="list-style-type: none"> <li>• Verificar que los datos para establecer la conexión con la base de datos de GRHS sean los correctos.</li> <li>• Guardar los datos.</li> <li>• Cargar los datos.</li> <li>• Verificar que se establezca la conexión</li> </ul>	<ul style="list-style-type: none"> <li>• Pgsqldb.php.</li> </ul>

**Tabla 6 Tarjeta CRC PluginGrhsAccion.**

<b>Tarjeta CRC:</b>	
<b>Clase:</b> PluginGrhsAccion	
<b>Responsabilidad:</b>	<b>Colaboraciones:</b>
<ul style="list-style-type: none"> <li>• Obtener los datos de la base de datos con respecto a las estadísticas.</li> <li>• Obtener los datos de la base de datos con respecto a las alarmas.</li> <li>• Obtener los datos de la base de datos con respecto a los controles.</li> </ul>	<ul style="list-style-type: none"> <li>• Pgsqldb.php.</li> </ul>

**Tabla 7 Tarjeta CRC PluginGrhsTraza.**

<b>Tarjeta CRC:</b>	
<b>Clase:</b> PluginGrhsTraza	

<b>Responsabilidad:</b> <ul style="list-style-type: none"> <li>• Obtener los datos de la base de datos con respecto a las peticiones de registro.</li> <li>• Obtener los datos de la base de datos con respecto a la comunicación de agentes.</li> </ul>	<b>Colaboraciones:</b> <ul style="list-style-type: none"> <li>• Pgsqldb.php.</li> </ul>
--	---

**Tabla 8 Tarjeta CRC PluginGrhsAgente.**

<b>Tarjeta CRC:</b>	
<b>Clase:</b> PluginGrhsAgente	
<b>Responsabilidad:</b> <ul style="list-style-type: none"> <li>• Obtener los datos de la base de datos con respecto a las estadísticas de los agentes.</li> <li>• Obtener los datos de la base de datos con respecto a los colectores activos.</li> <li>• Obtener los datos de la base de datos con respecto a los colectores inactivos.</li> </ul>	<b>Colaboraciones:</b> <ul style="list-style-type: none"> <li>• Pgsqldb.php.</li> </ul>

**Tabla 9 Tarjeta CRC PluginGrhsIncidencia.**

<b>Tarjeta CRC:</b>	
<b>Clase:</b> PluginGrhsIncidencia	
<b>Responsabilidad:</b> <ul style="list-style-type: none"> <li>• Obtener los datos de la base de datos con respecto a las incidencias.</li> </ul>	<b>Colaboraciones:</b> <ul style="list-style-type: none"> <li>• Pgsqldb.php.</li> </ul>

**Tabla 10 Tarjeta CRC PluginGrhsSincronizadorOrdenadores.**

<b>Tarjeta CRC:</b>	
<b>Clase:</b> PluginGrhsSincronizadorOrdenadores	
<b>Responsabilidad:</b>	<b>Colaboraciones:</b>
<ul style="list-style-type: none"> <li>• Importar todos los componentes de una pc.</li> </ul>	<ul style="list-style-type: none"> <li>• DiscoDuro.php.</li> <li>• LectorÓptico.php.</li> <li>• Memoria.php.</li> <li>• Monitor.php.</li> <li>• Mouse.php.</li> <li>• Procesador.php.</li> <li>• TarjetadeRed.php.</li> <li>• Teclado.php.</li> <li>• Pgsqldb.php.</li> </ul>

**Tabla 11 Tarjeta CRC PluginGrhsPgsqldb.**

<b>Tarjeta CRC:</b>	
<b>Clase:</b> PluginGrhsPgsqldb	
<b>Responsabilidad:</b>	<b>Colaboraciones:</b>
<ul style="list-style-type: none"> <li>• Cargar la configuración para establecer la conexión con la base de datos.</li> </ul>	

### **2.13 Conclusiones del capítulo 2.**

En el presente capítulo se realizó una descripción de la propuesta de solución, se redactaron las HU que deben ser implementadas, se realizó la estimación por esfuerzo, se definieron las iteraciones y se elabora el plan de entrega para definir las prioridades de las HU y la duración del desarrollo del plugin. Se describió y se explicó la arquitectura y se seleccionó los patrones

de diseños Grasp y Gof empleados, dando muestra de buenas prácticas de programación, por último se realiza la elaboración de las tarjetas CRC con el objetivo de guiar al programador al desarrollo orientado a objetos.



## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL PLUGIN PARA LA INTEGRACIÓN DE DATOS DE GRHS A GLPI

### 3.1 Introducción

Según la metodología XP la implementación de un software debe realizarse de forma iterativa, obteniendo al final de cada una de estas iteraciones un producto funcional el cual deberá ser mostrado al cliente para retroalimentar a los programadores y obtener un mejor software. En el siguiente capítulo se mostrarán las 3 iteraciones realizadas para integrar ambas herramientas así como las tareas de ingeniería de cada historia de usuario por cada iteración, además se realizarán las pruebas de aceptación realizadas al producto. Teniendo en cuenta la planificación realizada se llevaron a cabo tres iteraciones las cuales dejaremos a continuación con una descripción de las mismas.

### 3.2 Iteración 1

En la siguiente iteración se mostrarán las historias de usuarios con una prioridad para el negocio alta dichas historias de usuarios son: configurar integración, mostrar el inventario e incidencias, al terminar la implementación de dichas historias de usuarios se presentará al cliente para obtener una rápida retroalimentación de este.

**Tabla 12 UH Perteneciente a la Primera Iteración.**

Historia de usuario	Estimación	Real
Configurar integración	2	2
Mostrar el inventario	1	1
Mostrar las incidencias	1	1

### 3.3 Iteración 2

En la siguiente iteración se aborda las historias de usuario de prioridad medias tales como: mostrar las acciones y mostrar los agentes, al terminar la implementación de las mismas se obtendrá un producto más completo el cual será presentado al cliente para una retroalimentación.

**Tabla 13: Perteneciente a la Segunda Iteración.**

Historia de usuario	Estimación	Real
Mostrar las acciones	2	2
Mostrar los agentes	1	1

### 3.3 Iteración 3

En la siguiente iteración se aborda las historias de usuario de prioridad baja tales como: mostrar las trazas y mostrar usuarios y grupos del sistema, al terminar la implementación de las mismas se obtendrá un producto más completo el cual será presentado al cliente para una retroalimentación.

**Tabla 14: UH Pertenciente a la Tercera Iteración.**

Historia de usuario	Estimación	Real
Mostrar las trazas	1	1
Mostrar usuarios y grupos del sistema	1	1

### 3.4 Tareas de ingeniería

Como se planteó la metodología de desarrollo XP orienta a la implementación de forma iterativa. Durante cada iteración se implementa un número de historias de usuarios definidas por el cliente y descritas por el equipo de desarrollo. En la fase de implementación las historias de usuarios son divididas en tareas de ingeniería las que son asignadas a los desarrolladores para su implementación.

Las tareas de ingeniería son representadas mediante tablas con los siguientes campos:

- ✓ **Nombre de la tarea:** nombre que identifica a la tarea de ingeniería.
- ✓ **Número de la tarea:** número que identifica a la tarea, los números deben ser de forma consecutiva.
- ✓ **Número de historia:** número de la historia de usuario a la cual pertenece la tarea.
- ✓ **Tipo de tarea:** se especifica el tipo de tarea, Desarrollo, Corrección o Mejora.
- ✓ **Puntos estimados:** duración en semanas de la tarea a desarrollar.
- ✓ **Responsables:** desarrollador(es) responsable de realizar la tarea.
- ✓ **Descripción:** breve síntesis de la tarea.

A continuación un ejemplo de una tarea de ingeniería perteneciente a la primera iteración.

**Tabla 15: UH Pertenciente a la UH configurar integración.**

Tarea	
<b>Nombre de tarea:</b> Configurar servicio	
<b>Número de tarea :</b> 1	<b>Número de historia:</b> 1
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio :</b> 02-02-2015	<b>Fecha de fin :</b> 02-02-2015

<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.
<b>Descripción:</b> El cliente selecciona configurar servicio que le permite conectarse a la base de datos de GRHS, llena los campos de host, puerto, nombre de la base de datos, usuario y contraseña.

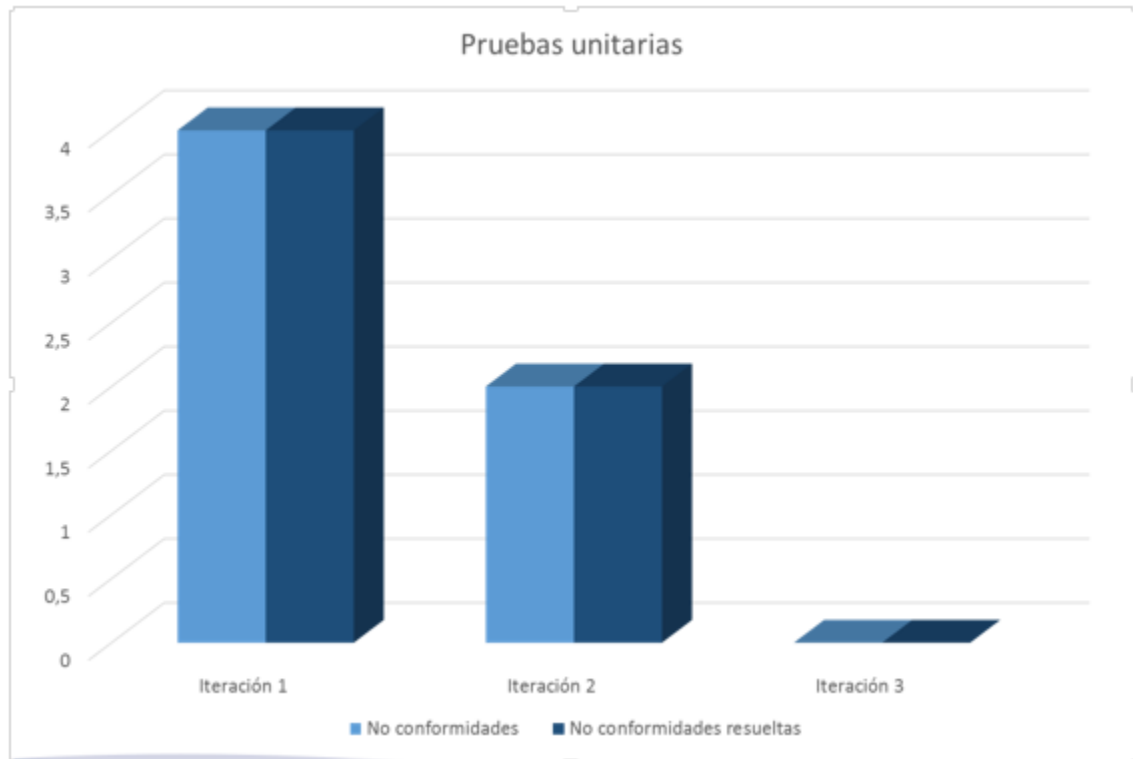
El resto de las tareas de ingeniería definidas por el equipo de desarrollo se encuentran en el [Anexo II.](#)

### **3.5 Pruebas**

Uno de los pilares de XP es el proceso de pruebas. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aplicación y se detección. También permite la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones. En XP existen dos tipos de pruebas las unitarias y las pruebas de aceptación (39).

#### **3.5.1 Pruebas unitarias**

Las pruebas unitarias aseguran que un único de la aplicación produce una salida correcta para una única entrada. Este tipo de pruebas validan la forma en que las funciones y métodos en cada caso particular. Las pruebas unitarias se encargan de un único caso cada vez, lo que significa que un único método necesite varias pruebas unitarias y su funcionamiento varía en función de contexto (35). Para la realización de las pruebas unitarias se seleccionó el framework PHPUnit, se seleccionaron 18 fragmentos de código y se detectaron 6 no conformidades las cuales fueron resueltas.



En la iteración 1 se detectaron 8 fragmentos de código y se detectaron 4 errores de lógica algorítmica.

En la iteración 2 se seleccionaron 6 fragmentos de código y se detectaron 2 errores de lógica algorítmica.

En la iteración 3 se seleccionaron 4 fragmentos de código y no se detectaron errores.

A continuación se muestra una prueba unitaria realizada primera iteración.

```

test
Done: 12 of 12 (0.102 s)
/usr/bin/php /tmp/ide-phpunit.php --no-configuration PluginGrhsUsuarioTest /usr/share/gli/plugins/grhs/pruebas/PluginGrhsUsuarioTest.php
Testing started at 22:27 ...
PHPUnit 3.7.28 by Sebastian Bergmann.

Time: 116 ms, Memory: 8.50Mb
OK (12 tests, 12 assertions)
Process finished with exit code 0

```

#### Ilustración 4: Prueba unitaria

El resto de las pruebas de unitarias definida por el equipo de desarrollo se encuentran en el

## Anexo IV.

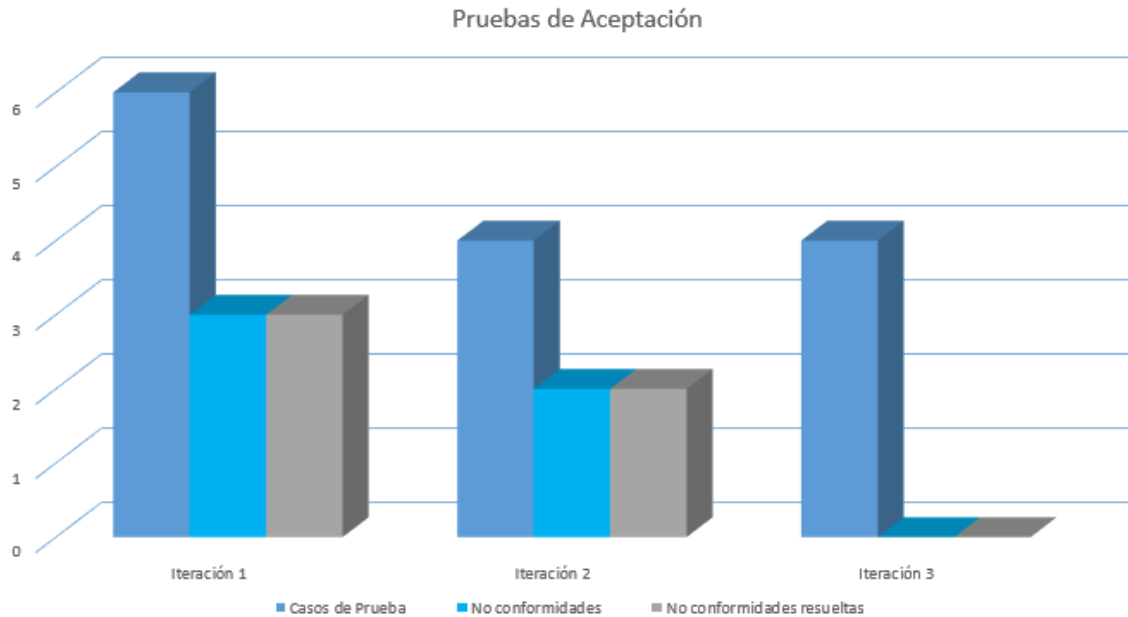
### 3.5.2 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios en cada ciclo de la iteración de desarrollo. El cliente debe especificar uno o diferentes escenarios para considerar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como "pruebas de caja negra" y los clientes son los encargados de verificar que los resultados de dichas pruebas sean los correctos (35).

Las pruebas de aceptación son representadas por tablas con los siguientes campos:

- ✓ **Código:** describe el número de la historia de usuario y la prueba.
- ✓ **Historia de usuario:** nombre de la historia de usuario a la cual se le realiza la prueba.
- ✓ **Nombre de la persona que realiza la prueba:** nombre de la persona del equipo de desarrollo encargada de realizar la prueba.
- ✓ **Descripción:** breve descripción de la funcionalidad del sistema.
- ✓ **Condiciones de ejecución:** condiciones que deben de cumplirse antes de comenzar las pruebas.
- ✓ **Entrada/Pasos de ejecución:** condiciones que se van a verificar en las pruebas.
- ✓ **Resultado esperado:** resultado que se espera con la realización de las pruebas.
- ✓ **Evaluación de la prueba:** clasificación de las pruebas según el resultado esperado.

Se realizaron casos de pruebas de aceptación a las 7 HU determinando no conformidades, en la siguiente ilustración se muestra el número de casos de pruebas por iteraciones así como la cantidad de no conformidades detectadas y resueltas.



En la iteración 1 se realizaron 6 casos de pruebas detectando las siguientes no conformidades:

- ✓ No se mostraban las incidencias.
- ✓ Mostraba programas no instalados en las computadoras.
- ✓ No señalaba la ausencia de campos vacíos.

En la iteración 2 se realizaron 4 casos de pruebas detectándose las siguientes no conformidades:

- ✓ No se mostraban los colectores activos.
- ✓ No se mostraban los colectores inactivos.

En la iteración 3 se realizaron 4 casos de pruebas y no se encontraron no conformidades.

**Tabla 16: Caso de prueba de aceptación HU1-P1.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU1-P1	<b>Historia de usuario:</b> Mostrar usuarios y grupos de GRHS.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar los usuarios existentes en GRHS.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a cada usuario.	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz usuarios los datos del mismo (usua-	

rio, apellido, correo electrónico).

**Evaluación de la prueba: Satisfactoria.**

El resto de las pruebas de aceptación definida por el equipo de desarrollo se encuentran en el [Anexo III](#).

### **Conclusiones del capítulo 3**

En este capítulo se elaboraron las tareas de ingeniería correspondiente a cada HU con el objetivo de que los desarrolladores puedan implementarlas. Se realizaron las pruebas que propone la metodología XP para poder probar que el plugin desarrollado funciona correctamente. Para ello se realizaron las pruebas unitarias con el framework PHPUnit las cuales arrojaron no conformidades que fueron corregidas, por otra parte se llevaron a cabo las pruebas de aceptación a las 3 iteraciones definidas en las fases anteriores, detectando no conformidades que fueron arregladas en el transcurso de desarrollo del plugin.

## **Conclusiones Generales**

En la solución se describió el funcionamiento del plugin para GLPI con el objetivo de mostrar de forma más organizada la información generada por GRHS. Una vez finalizada la misma arribaron a las siguientes conclusiones:

- ✓ Se realizó un estudio de las principales técnicas de integración de datos existentes escogiendo ETL como la adecuada.
- ✓ Se realizó un estudio de las principales herramientas capaces de realizar el proceso de integración y migración de datos, arribando a la conclusión que ninguno de ellos ofrece una solución al problema planteado.
- ✓ El plugin desarrollado puede ser usada en GLPI permitiendo integrar y mostrar en a interfaz de este toda la información generada por GRHS.
- ✓ La selección de las herramientas y tecnologías usadas en el desarrollo del plugin de integración de GRHS con GLPI fueron seleccionadas teniendo en cuenta las solicitudes y restricciones del cliente.
- ✓ Se cumplió con las exigencias dictadas por la metodología de desarrollo de software XP generando los artefactos necesarios para el desarrollo diseño e implementación del plugin de integración.

Luego del análisis anterior se expone que el siguiente trabajo de diploma cumple con los objetivos propuestos.

## **Recomendaciones**

Luego de haber analizado los resultados del presente trabajo se recomienda:

- ✓ Añadir el plugin de configuración al plugin desarrollado.



## REFERENCIAS BIBLIOGRÁFICAS.

1. ALCALDE LAUCHARRO, E. y TOMAS GARCIA, J. [sin fecha]. *Introducción a la Te-  
leinformática*. S.l.: S.A. MCGRAW-HILL. ISBN 9788448101404.
2. Telecomunicaciones. [2015]. [Consulta: 15 noviembre 2014]. Disponible en:  
<http://es.slideshare.net/ChiaLucci/telecomunicaciones-33655154>.
3. GLPI - Gestionnaire libre de parc informatique. [2002-2015]. [Consulta 15 de no-  
viembre del 2014]. Disponible en: <http://www.glpi-project.org/>.
4. BERTHET, C. *Manual de Informática*. S.l.: El Ateneo. ISBN 950-02-5251.
5. masadelante. [1999-2015]. [Citado el: 25 de noviembre de 2014.]  
<https://www.masadelante.com/faqs/plug-in>.
6. Oracle. *Oracle*. [En línea] [Consulta: 12 de diciembre de 2014.]  
<http://www.oracle.com/ocom/groups>.
7. RAMOS, R.A. Tecnológico. [en línea]. [Consulta: 2 febrero 2015]. Disponible en:  
<http://mitecnologico.com/Main/Tecnologico>.
8. AZÁN BASALLO, Y. y DÍAZ ESTRADA, A. *Una experiencia sobre integración de  
aplicaciones informáticas*. S.l.: s.n.
9. White, Colin. *Data Integration: Using ETL, EAI, and EII Tools to Create an Integrat-  
ed Enterprise*. 2009
10. Taylor, J. *Enterprise Information Integration: A New Definition*. 2004 [Consulta: 15  
noviembre 2014]; Available from: <http://www.information>
11. Gable, J., *Enterprise application integration*. *Information Management Journal*. ,  
2002 March/April 2009.
12. Kimball, Ralph. *The Data Warehouse ETL Toolkit*. 2009.
13. DB CONVERT. [En línea] 2001-2011. [Citado el: 10 de noviembre de  
2014], <http://dbconvert.com/convert-mysql-to-postgresql-pro.php>.
14. Navicat. [En línea] 1999 – 2014. [Citado el: 5 de noviembre de 2014].  
<http://www.navicat.com/es/whatisnavicat>.
15. ESF Database Migration Toolkit Professional Editon SOFTPEDIA. [En línea] 2001-  
2011. [Citado el: 20 de Octubre de 2014] [http://www.softpedia.es/programa-ESF-  
Database-Convert-Professional-105802.html](http://www.softpedia.es/programa-ESF-Database-Convert-Professional-105802.html).
16. Oracle. *Oracle*. [En línea] [Citado el: 12 de diciembre de 2014.]  
<http://www.oracle.com/ocom/groups>.

17. Huice, Ing. Odaysa Rodriguez. Programa de Mejora 0208\_Proyecto Técnico v1.0 Proyecto Gestión de Recursos de Hardware y Software.
18. LETELIER, P. y PENADÉS, M. del C. [sin fecha]. *Metodologías Ágiles para el desarrollo de software*. S.l.: s.n.
19. BONANATA, M. [sin fecha]. *Programación y Algoritmo*. S.l.: MP Ediciones SA.
20. GALLEGO, J.A. [sin fecha]. *Desarrollo WEB con PHP y MySQL*. S.l.: s.n. ISBN 84-415-1525.
21. TORRES, A. [sin fecha]. Lenguajes del lado servidor o cliente. [en línea]. [Consulta: 2 enero 2015]. Disponible en: [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html).
22. Capítulo 1. Introducción (Introducción a JavaScript). [En línea] [Citado el: 15 de noviembre de 2014.] [http://librosweb.es/javascript/capitulo\\_1.html](http://librosweb.es/javascript/capitulo_1.html). java Scrip
23. developerWorks. Fundamentos HTML 5 Parte 1. [En línea] [Citado el: 9 de Diciembre de 2014.] <http://www.ibm.com/developerworks/ssa/web/library/wa-html5fundamentals/.Framawork>
24. Introducción a AJAX. [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: <https://librosweb.es/libro/ajax/>.
25. Smarty. [En línea] 2002-2015. [Citado el: 14 de diciembre de 2014.] <http://www.smarty.net/docsv2/es/what.is.smarty.tpl>. de la Torre, Aníbal. Adelat. [En línea]
26. GALLEGO, J.A. [sin fecha]. *Desarrollo WEB con PHP y MySQL*. S.l.: s.n. ISBN 84-415-1525.
27. Editores de Código. [En línea] 2014. [Citado el: 12 de febrero de 2015.] <http://www.editoresdecodigo.com/2014/06/descargar-phpstorm-full-ide-para-php-y-mas.html>. .
28. Portal de la Udelar | Recursos Informáticos. [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: <http://www.universidad.edu.uy/renderPage/index/pageld/436>.
29. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
30. universidad de Santo Tomas. [En línea] 2013. [Citado el: 23 de enero de 2015.] <http://www.ustamed.edu.co/sistemas/index.php/frentes/sistemas/soporte-a-servidores> [sin fecha]. [Consulta: 2 abril 2015 b]. Disponible en: <http://netpecos.org/?hg=0&nr=0>.

31. MOUSQUES, G. *Metodología XP*. S.l.: s.n.
32. *Microsoft Word - XP - Jose Joskowicz.doc - XP - Jose Joskowicz.pdf* [en línea], [sin fecha]. S.l.: s.n. [Consulta: 2 junio 2015]. Disponible en: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
33. CERVANTES, H. [sin fecha]. *Arquitectura de Software | SG*. [en línea]. [Consulta: 2 enero 2015]. Disponible en: <http://sg.com.mx/revista/27/arquitectura-software>.
34. Capítulo 2. Explorando el interior de Symfony (Symfony 1.0, la guía definitiva). [en línea] [sin fecha]. [Consulta: 3 febrero 2015]. Disponible en: [http://librosweb.es/libro/symfony\\_1\\_0/capitulo\\_2.html](http://librosweb.es/libro/symfony_1_0/capitulo_2.html).
35. Posgrado en Ciencia e Ingeniería de la Computación. [en línea]. [Consulta: 3 abril 2015]. Disponible en: <http://www.mcc.unam.mx/>.
36. El patrón MVC (Symfony 1.0, la guía definitiva). [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: [http://librosweb.es/libro/symfony\\_1\\_0/capitulo\\_2/el\\_patron\\_mvc.html](http://librosweb.es/libro/symfony_1_0/capitulo_2/el_patron_mvc.html).
37. Patrones de diseño. [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: <http://es.kioskea.net/contents/224-patrones-de-diseno>.
38. MARCELLO, V. y ASTUDILLO, H. [sin fecha]. *Fundamentos de la Ingeniería de Software*. S.l.: s.n.
39. TORRES, J., ESCALONA, M., MEJIAS, M. y GUTIERREZ, J. *Pruebas del Sistema en Programación extrema*. S.l.: s.n.

## BIBLIOGRAFÍA.

1. ALCALDE LAUCHARRO, E. y TOMAS GARCIA, J. [sin fecha]. *Introducción a la Te-  
leinformática*. S.l.: S.A. MCGRAW-HILL. ISBN 9788448101404.
2. Telecomunicaciones.[2015].[Consulta: 15 noviembre 2014]. Disponible en:  
<http://es.slideshare.net/ChiaLucci/telecomunicaciones-33655154>.
3. GLPI - Gestionnaire libre de parc informatique. [2002-2015]. [Consulta15 de no-  
viembre del 2014]. Disponible en: <http://www.glpi-project.org/>.
4. BERTHET, C. *Manual de Informática*. S.l.: El Ateneo. ISBN 950-02-5251.
5. masadelante. [1999-2015].[Citado el: 25 de noviembre de 2014.]  
<https://www.masadelante.com/faqs/plug-in>.
6. Oracle. *Oracle*. [En línea] [Consulta: 12 de diciembre de 2014.]  
<http://www.oracle.com/ocom/groups>.
7. RAMOS, R.A.Tecnológico. [en línea]. [Consulta: 2 febrero 2015]. Disponible en:  
<http://mitecnologico.com/Main/Tecnologico>.
8. AZÁN BASALLO, Y. y DÍAZ ESTRADA, A. *Una experiencia sobre integración de  
aplicaciones informáticas*. S.l.: s.n.
9. White, Colin. *Data Integration: Using ETL, EAI, and EII Tools to Create an Integrat-  
ed Enterprise*.2009
10. Taylor, J. *Enterprise Information Integration: A New Definition*. 2004 [Consulta: 15  
noviembre 2014]; Available from:<http://www.information>
11. Gable, J., *Enterprise application integration*. *Information Management Journal*. ,  
2002 March/April 2009.
12. Kimball, Ralph.*The Data Warehouse ETL Toolkit*. 2009.
13. DB CONVERT. [En línea] 2001-2011. [Citado el: 10 de noviembre de  
2014],<http://dbconvert.com/convert-mysql-to-postgresql-pro.php>.
14. Navicat. [En línea] 1999 – 2014. [Citado el: 5 de noviembre de 2014].  
<http://www.navicat.com/es/whatisnavicat>.
15. ESF Database Migration Toolkit Professional Editon SOFTPEDIA. [En línea] 2001-  
2011. [Citado el: 20 de Octubre de 2014] [http://www.softpedia.es/programa-ESF-  
Database-Convert-Professional-105802.html](http://www.softpedia.es/programa-ESF-Database-Convert-Professional-105802.html).
16. Oracle. *Oracle*. [En línea] [Citado el: 12 de diciembre de 2014.]  
<http://www.oracle.com/ocom/groups>.

17. Huice, Ing. Odaysa Rodriguez. Programa de Mejora 0208\_Proyecto Técnico v1.0 Proyecto Gestión de Recursos de Hardware y Software.
18. LETELIER, P. y PENADÉS, M. del C. [sin fecha]. *Metodologías Ágiles para el desarrollo de software*. S.l.: s.n.
19. BONANATA, M. [sin fecha]. *Programación y Algoritmo*. S.l.: MP Ediciones SA.
20. GALLEGO, J.A. [sin fecha]. *Desarrollo WEB con PHP y MySQL*. S.l.: s.n. ISBN 84-415-1525.
21. TORRES, A. [sin fecha]. Lenguajes del lado servidor o cliente. [en línea]. [Consulta: 2 enero 2015]. Disponible en: [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html).
22. Capítulo 1. Introducción (Introducción a JavaScript). [En línea] [Citado el: 15 de noviembre de 2014.] [http://librosweb.es/javascript/capitulo\\_1.html](http://librosweb.es/javascript/capitulo_1.html). java Scrip
23. developerWorks. Fundamentos HTML 5 Parte 1. [En línea] [Citado el: 9 de Diciembre de 2014.] <http://www.ibm.com/developerworks/ssa/web/library/wa-html5fundamentals/.Framawork>
24. Introducción a AJAX. [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: <https://librosweb.es/libro/ajax/>.
25. Smarty. [En línea] 2002-2015. [Citado el: 14 de diciembre de 2014.] <http://www.smarty.net/docsv2/es/what.is.smarty.tpl>. de la Torre, Aníbal. Adelat. [En línea]
26. GALLEGO, J.A. [sin fecha]. *Desarrollo WEB con PHP y MySQL*. S.l.: s.n. ISBN 84-415-1525.
27. Editores de Código. [En línea] 2014. [Citado el: 12 de febrero de 2015.] <http://www.editoresdecodigo.com/2014/06/descargar-phpstorm-full-ide-para-php-y-mas.html>. .
28. Portal de la Udelar | Recursos Informáticos. [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: <http://www.universidad.edu.uy/renderPage/index/pageld/436>.
29. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
30. universidad de Santo Tomas. [En línea] 2013. [Citado el: 23 de enero de 2015.] <http://www.ustamed.edu.co/sistemas/index.php/frentes/sistemas/soporte-a-servidores> [sin fecha]. [Consulta: 2 abril 2015 b]. Disponible en: <http://netpecos.org/?hg=0&nr=0>.

31. MOUSQUES, G. *Metodología XP*. S.l.: s.n.
32. *Microsoft Word - XP - Jose Joskowicz.doc - XP - Jose Joskowicz.pdf* [en línea], [sin fecha]. S.l.: s.n. [Consulta: 2 junio 2015]. Disponible en: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
33. CERVANTES, H. [sin fecha]. *Arquitectura de Software | SG*. [en línea]. [Consulta: 2 enero 2015]. Disponible en: <http://sg.com.mx/revista/27/arquitectura-software>.
34. Capítulo 2. Explorando el interior de Symfony (Symfony 1.0, la guía definitiva). [en línea] [sin fecha]. [Consulta: 3 febrero 2015]. Disponible en: [http://librosweb.es/libro/symfony\\_1\\_0/capitulo\\_2.html](http://librosweb.es/libro/symfony_1_0/capitulo_2.html).
35. Posgrado en Ciencia e Ingeniería de la Computación. [en línea]. [Consulta: 3 abril 2015]. Disponible en: <http://www.mcc.unam.mx/>.
36. El patrón MVC (Symfony 1.0, la guía definitiva). [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: [http://librosweb.es/libro/symfony\\_1\\_0/capitulo\\_2/el\\_patron\\_mvc.html](http://librosweb.es/libro/symfony_1_0/capitulo_2/el_patron_mvc.html).
37. Patrones de diseño. [en línea] [sin fecha]. [Consulta: 2 abril 2015]. Disponible en: <http://es.kioskea.net/contents/224-patrones-de-diseno>.
38. MARCELLO, V. y ASTUDILLO, H. [sin fecha]. *Fundamentos de la Ingeniería de Software*. S.l.: s.n.
39. TORRES, J., ESCALONA, M., MEJIAS, M. y GUTIERREZ, J. *Pruebas del Sistema en Programación extrema*. S.l.: s.n.
40. Larman, Craig. *UML y Patrones. Segunda Edición*. s.l.: Prentice Hall.
41. BECK, F. M. Y. K. *Planeando en Programación Extrema*. 2000.
42. Escribano, Gerardo Fernández. *Introducción a Extreme Programming*.

## ANEXOS

### Anexo 1: Historias de usuarios

Tabla 17: Mostrar el inventario.

Historia de Usuario:	
<b>Número: 2</b>	<b>Nombre de la Historia de Usuario:</b> Mostrar el inventario
<b>Riesgo en Desarrollo :</b> Alto	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Programador Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Muestra el inventario de los recursos de hardware y software.	
<b>Observaciones:</b>	

Tabla 18: Mostrar las incidencias.

Historia de Usuario:	
<b>Número: 3</b>	<b>Nombre de la Historia de Usuario:</b> Mostrar las incidencias
<b>Riesgo en Desarrollo :</b> Alto	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Programador Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Muestra un conjunto de datos acerca de las incidencias ocurridas en una computadora.	
<b>Observaciones:</b>	

Tabla 19: Mostrar las acciones.

Historia de Usuario:	
<b>Número: 4</b>	<b>Nombre de la Historia de Usuario:</b> Mostrar las acciones
<b>Riesgo en Desarrollo :</b> Media	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2

<b>Programador Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.
<b>Descripción:</b> Muestra el total de alarmas y controles realizados.
<b>Observaciones:</b>

**Tabla 20: Mostrar los agentes**

<b>Historia de Usuario:</b>	
<b>Número: 5</b>	<b>Nombre de la Historia de Usuario:</b> Mostrar los agentes
<b>Riesgo en Desarrollo :</b> Media	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Programador Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Permite visualizar la información acerca del total de host o computadoras además de mostrar los conectores activos e inactivos del sistema.	
<b>Observaciones:</b>	

**Tabla 21: Mostrar las trazas.**

<b>Historia de Usuario:</b>	
<b>Número: 6</b>	<b>Nombre de la Historia de Usuario:</b> Mostrar las trazas
<b>Riesgo en Desarrollo :</b> Baja	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Baja	<b>Puntos Estimados:</b> 1
<b>Programador Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Muestra las peticiones de registro y la comunicación con el agente.	
<b>Observaciones:</b>	

**Tabla 22: Mostrar usuarios y grupos del sistema.**

<b>Historia de Usuario:</b>	
<b>Número: 7</b>	<b>Nombre de la Historia de Usuario:</b> Mostrar usuarios y grupos del sistema



<b>Riesgo en Desarrollo :</b> Baja	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Baja	<b>Puntos Estimados:</b> 1
<b>Programador Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Permite mostrar los usuarios y los grupos existentes en GRHS.	
<b>Observaciones:</b>	

## Anexo 2: Tareas de ingeniería.

Tabla 23: Pertenciente a la UH Configurar integración

Tarea	
<b>Nombre de tarea:</b> Importar BD de GRHS	
<b>Número de tarea :</b> 2	<b>Número de historia:</b> 1
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 09-02-2015	<b>Fecha de fin:</b> 11-02-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> El usuario selecciona configurar integración, importas bases de datos de GRHS y seguidamente el sistema es capaz de importar todo los datos que se van a mostrar desde GRHS hacia GLPI.	

Tabla 24: Pertenciente a la UH Configurar integración.

Tarea	
<b>Nombre de tarea:</b> Limpiar enlace entre GRHS y GLPI.	
<b>Número de tarea :</b> 3	<b>Número de historia:</b> 1
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 12-02-2015	<b>Fecha de fin:</b> 13-02-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> El usuario selecciona configurar integración, limpiar enlace entre GRHS y GLPI y el sistema borra todos los registros de GRHS existentes en GLPI.	

Tabla 25: Pertenciente a la UH Mostrar el inventario.

Tarea	
<b>Nombre de tarea:</b> Mostrar el inventario	
<b>Número de tarea :</b> 1	<b>Número de historia:</b> 2
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 16-02-2015	<b>Fecha de fin:</b> 20-02-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de importar los datos los usuarios pueden ver la información referente a los componentes de hardware y software, se debe importar y mostrar todo lo referente de un	

ordenador (monitores, tarjetas de red, placa base, memoria, procesadores, programas, conexiones, tarjetas gráficas y mostrarlo en las interfaces que presenta GLPI para estos componentes).

**Tabla 26: Pertenece a la UH Mostrar las incidencias.**

Tarea	
<b>Nombre de tarea:</b> Mostrar las incidencias	
<b>Número de tarea :</b> 1	<b>Número de historia:</b> 3
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 23-02-2015	<b>Fecha de fin:</b> 27-02-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos, el usuario selecciona el módulo de incidencias y el sistema la muestra la información referente a las incidencias teniendo en cuenta identificador, componente, tipo de incidencia, estado, nivel, fecha y localización.	

**Tabla 27: Pertenece a la UH Mostrar las acciones.**

Tarea	
<b>Nombre de tarea:</b> Mostrar total de alarmas y controles	
<b>Número de tarea :</b> 1	<b>Número de historia:</b> 4
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 02-03-2015	<b>Fecha de fin:</b> 06-03-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el módulo de acciones el sistema muestra la información teniendo en cuanta total de alarmas y controles.	

**Tabla 28: Pertenece a la UH Mostrar las acciones.**

Tarea	
<b>Nombre de tarea:</b> Mostrar alarmas	
<b>Número de tarea :</b> 2	<b>Número de historia:</b> 4
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.6

<b>Fecha de inicio:</b> 09-03-2015	<b>Fecha de fin:</b> 11-03-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el módulo de acciones el sistema muestra la información referente a las alarmas con los siguientes datos remitente, destinatario, tipo de mensaje y fecha.	

**Tabla 29: Pertenece a la UH Mostrar las acciones.**

Tarea	
<b>Nombre de tarea:</b> Mostrar controles	
<b>Número de tarea :</b> 3	<b>Número de historia:</b> 4
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 12-03-2015	<b>Fecha de fin:</b> 13-03-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el módulo de acciones el sistema muestra la información referente a el total de controles con los siguientes datos remitente, destinatario, tipo de mensaje y fecha.	

**Tabla 30: Pertenece a la UH Mostrar los agentes.**

Tarea	
<b>Nombre de tarea:</b> Mostrar estadísticas	
<b>Número de tarea :</b> 1	<b>Número de historia:</b> 5
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 16-03-2015	<b>Fecha de fin:</b> 18-03-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el módulo de agentes el sistema muestra la información de las estadísticas teniendo en cuenta total de host o computadoras, total de colectores tantos activos como inactivos.	

**Tabla 31: Pertenece a la UH Mostrar los agentes.**

Tarea	
<b>Nombre de tarea:</b> Mostrar colectores activos	
<b>Número de tarea :</b> 2	<b>Número de historia:</b> 5
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 19-03-2015	<b>Fecha de fin:</b> 19-03-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el módulo de agentes, el sistema muestra la información referente a los colectores activos teniendo en cuenta identificador, dirección ip, última conexión y localización.	

**Tabla 32: Pertenciente a la UH Mostrar los agentes.**

Tarea	
<b>Nombre de tarea:</b> Mostrar colectores inactivos	
<b>Número de tarea :</b> 3	<b>Número de historia:</b> 4
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 20-03-2015	<b>Fecha de fin:</b> 20-03-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el módulo de agentes, el sistema muestra la información referente a los colectores inactivos teniendo en cuenta identificador, dirección ip, última conexión y localización.	

**Tabla 33: Pertenciente a la UH Mostrar las trazas.**

Tarea	
<b>Nombre de tarea:</b> Mostrar peticiones de registro	
<b>Número de tarea :</b> 1	<b>Número de historia:</b> 6
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 23-03-2015	<b>Fecha de fin:</b> 25-03-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el módulo de trazas, el sistema muestra la información	

referente a las peticiones de registro teniendo en cuenta identificador, fecha, dirección ip, puerto y plataforma.

**Tabla 34: Pertenece a la UH Mostrar las trazas.**

<b>Tarea</b>	
<b>Nombre de tarea:</b> Mostrar comunicación con agentes	
<b>Número de tarea :</b> 2	<b>Número de historia:</b> 6
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 26-03-2015	<b>Fecha de fin:</b> 27-03-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavides.	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el módulo de trazas, el sistema muestra la información referente a la comunicación con agentes teniendo en cuenta fecha, dirección ip, puerto y parámetros.	

**Tabla 35: Pertenece a la UH Mostrar usuarios y grupos del sistema.**

<b>Tarea</b>	
<b>Nombre de tarea:</b> Mostrar usuarios de GRHS.	
<b>Número de tarea :</b> 1	<b>Número de historia:</b> 7
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 30-03-2015	<b>Fecha de fin:</b> 01-04-2015
<b>Responsable:</b> Rafael Barinas Ricardo y Félix Rafael Benavidez	
<b>Descripción:</b> Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el campo usuarios y puede visualizar los usuarios tanto de GRHS como de GLPI los cuales poseen usuario, apellido, correo electrónico, teléfono lugar y si está activo.	

**Tabla 36: Pertenece a la UH Mostrar usuarios y grupos del sistema.**

<b>Tarea</b>	
<b>Nombre de tarea:</b> Mostrar grupos de GRHS.	
<b>Número de tarea :</b> 2	<b>Número de historia:</b> 1
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 02-04-2015	<b>Fecha de fin:</b> 03-04-2015

**Responsable:** Rafael Barinas Ricardo y Félix Rafael Benavides.

**Descripción:** Luego de configurar la integración con a base de datos de GRHS y haber importados los datos el usuario selecciona el campo grupos y puede visualizar los grupos tanto de GRHS como de GLPI los cuales poseen nombre completo y los comentarios.

### Anexo 3: Pruebas de aceptación.

Tabla 37: Caso de prueba de aceptación HU1-P2.

Casos de Pruebas de Aceptación	
<b>Código:</b> HU1-P2	<b>Historia de usuario:</b> Mostrar usuarios y grupos de GRHS.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar los grupos existentes en GRHS.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a cada grupo.	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz grupos los datos del mismo (nombre completo, comentarios).	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 38: Caso de prueba de aceptación HU2-P1.

Casos de Pruebas de Aceptación	
<b>Código:</b> HU2-P1	<b>Historia de usuario:</b> Gestionar integración.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Establecer la conexión con la base de datos de GRHS con datos correctos.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión.	
<b>Entrada/Pasos de ejecución:</b> El usuario introduce los siguientes datos (host, usuario, puerto, nombre de la base de datos, contraseña).	
<b>Resultado esperado:</b> La aplicación verifica que los datos sean los correctos y de serlos establece la conexión con éxito.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 39: Caso de prueba de aceptación HU2-P2.

Casos de Pruebas de Aceptación	
<b>Código:</b> HU2-P2	<b>Historia de usuario:</b> Gestionar integración.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	



<b>Descripción:</b> Establecer la conexión con la base de datos de GRHS con datos incorrectos.
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión.
<b>Entrada/Pasos de ejecución:</b> El usuario introduce los siguientes datos (host, usuario, puerto, nombre de la base de datos, contraseña).
<b>Resultado esperado:</b> La aplicación verifica que los datos sean los correctos y de no serlos muestra un error.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 40: Caso de prueba de aceptación HU2-P3.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU2-P3	<b>Historia de usuario:</b> Gestionar integración.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Establecer la conexión con la base de datos de GRHS con campos vacíos.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión.	
<b>Entrada/Pasos de ejecución:</b> El usuario introduce algunos datos dejando campos vacíos.	
<b>Resultado esperado:</b> La aplicación señala el campo vacío y pide que sea rellenado.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 41: Caso de prueba de aceptación HU3-P1.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU3-P1	<b>Historia de usuario:</b> Mostrar agentes.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar la estadísticas de los agentes.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a las estadísticas de los agentes	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz agente las estadísticas de estos (variable, cantidad).	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 42: Caso de prueba de aceptación HU3-P2.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU3-P2	<b>Historia de usuario:</b> Mostrar agentes.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar colectores activos.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a los colectores activos.	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz agente todo los datos de los colectores activos (identificador, dirección ip, ultima conexión y localización).	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 43: Caso de prueba de aceptación HU3-P3.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU3-P3	<b>Historia de usuario:</b> Mostrar agentes.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar colectores inactivos.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a los colectores inactivos.	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz agente todo los datos de los colectores inactivos (identificador, dirección ip, ultima conexión y localización).	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 44: Caso de prueba de aceptación HU4-P1.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU4-P1	<b>Historia de usuario:</b> Mostrar incidencias.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar las incidencias existentes.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los	

datos de GRHS para GLPI.
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a las incidencias.
<b>Resultado esperado:</b> La aplicación muestra en la interfaz incidencias todos los datos de las incidencias (identificador, componente, tipo de incidencia, estado, nivel, fecha y localización).
<b>Evaluación de la prueba: Satisfactoria.</b>

**Tabla 45: Caso de prueba de aceptación HU6-P1.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código: HU6-P1</b>	<b>Historia de usuario:</b> Mostrar acción.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar las estadísticas relacionadas con el total de alarmas y controles.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a las estadísticas del total de alarmas y controles.	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz acción todos los datos de las estadísticas referentes al total de alarmas y controles (variable, cantidad).	
<b>Evaluación de la prueba: Satisfactoria.</b>	

**Tabla 46: Caso de prueba de aceptación HU6-P2.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código: HU6-P2</b>	<b>Historia de usuario:</b> Mostrar acción.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar las alarmas.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a las alarmas.	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz acción todos los datos de las estadísticas alarmas (remitente, destinatario, tipo de mensaje y fecha).	
<b>Evaluación de la prueba: Satisfactoria.</b>	

**Tabla 47: Caso de prueba de aceptación HU6-P3.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU6-P3	<b>Historia de usuario:</b> Mostrar acción.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar las peticiones de registro.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a las controles.	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz acción todos los datos de las estadísticas controles (remitente, destinatario, tipo de mensaje y fecha).	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 48: Caso de prueba de aceptación HU7-P1.**

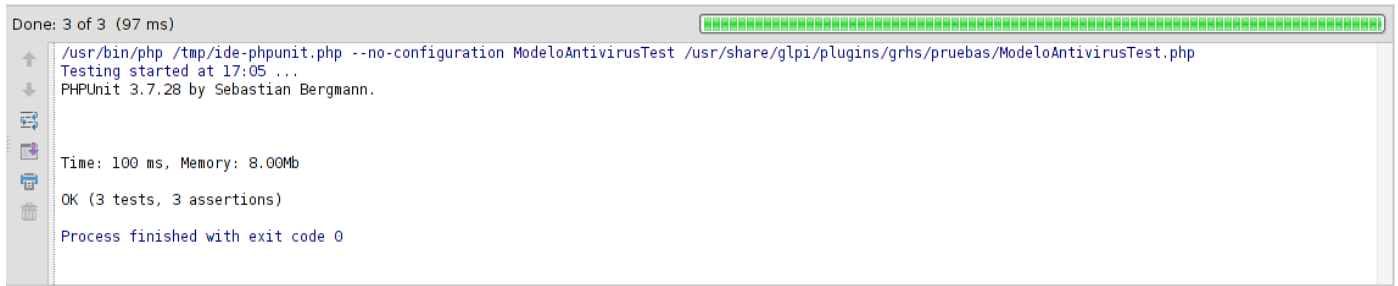
<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU7-P1	<b>Historia de usuario:</b> Mostrar trazas.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	
<b>Descripción:</b> Mostrar las comunicaciones con los agentes.	
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.	
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a la comunicación con agentes.	
<b>Resultado esperado:</b> La aplicación muestra en la interfaz trazas todos los datos de comunicación con agentes (fecha, dirección ip, puerto y parámetros).	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 49: Caso de prueba de aceptación HU7-P2.**

<b>Casos de Pruebas de Aceptación</b>	
<b>Código:</b> HU7-P2	<b>Historia de usuario:</b> Mostrar trazas.
<b>Nombre de la persona que realiza la prueba:</b> Rafael Arismende Barinas Ricardo.	

<b>Descripción:</b> Mostrar las peticiones de registro.
<b>Condiciones de ejecución:</b> Haber establecido la configuración de la conexión e importar los datos de GRHS para GLPI.
<b>Entrada/Pasos de ejecución:</b> Verificar que la interfaz muestre los datos correspondientes a las peticiones de registros.
<b>Resultado esperado:</b> La aplicación muestra en la interfaz trazas todos los datos de las peticiones de registro (identificador, fecha, dirección ip, puerto y plataforma).
<b>Evaluación de la prueba:</b> Satisfactoria.

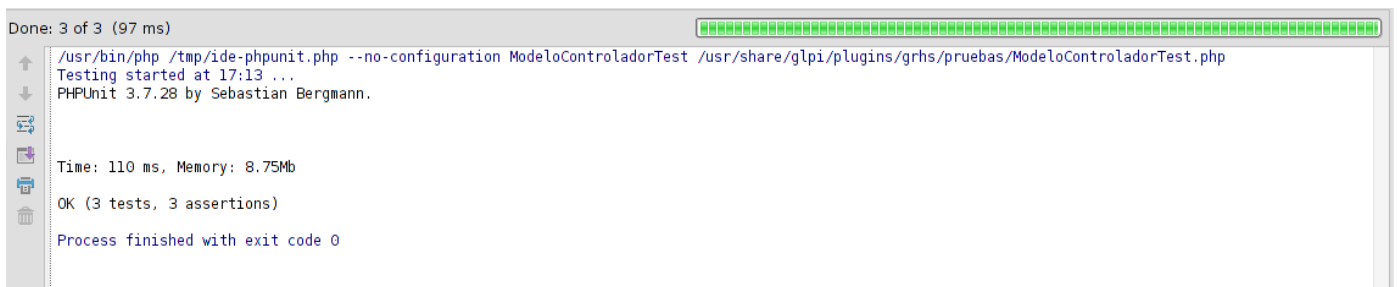
## Anexo 4: Pruebas unitarias.



```
Done: 3 of 3 (97 ms)
↑ /usr/bin/php /tmp/ide-phpunit.php --no-configuration ModeloAntivirusTest /usr/share/glpi/plugins/grhs/pruebas/ModeloAntivirusTest.php
↓ Testing started at 17:05 ...
PHPUnit 3.7.28 by Sebastian Bergmann.

Time: 100 ms, Memory: 8.00Mb
OK (3 tests, 3 assertions)
Process finished with exit code 0
```

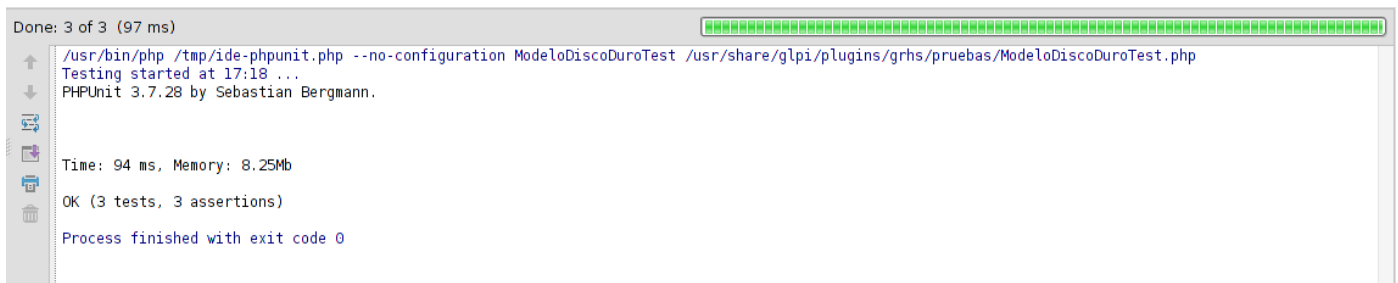
### Ilustración 4: Prueba unitaria modelo anti-virus



```
Done: 3 of 3 (97 ms)
↑ /usr/bin/php /tmp/ide-phpunit.php --no-configuration ModeloControladorTest /usr/share/glpi/plugins/grhs/pruebas/ModeloControladorTest.php
↓ Testing started at 17:13 ...
PHPUnit 3.7.28 by Sebastian Bergmann.

Time: 110 ms, Memory: 8.75Mb
OK (3 tests, 3 assertions)
Process finished with exit code 0
```

### Ilustración 5: Prueba unitaria modelo controlador



```
Done: 3 of 3 (97 ms)
↑ /usr/bin/php /tmp/ide-phpunit.php --no-configuration ModeloDiscoDuroTest /usr/share/glpi/plugins/grhs/pruebas/ModeloDiscoDuroTest.php
↓ Testing started at 17:18 ...
PHPUnit 3.7.28 by Sebastian Bergmann.

Time: 94 ms, Memory: 8.25Mb
OK (3 tests, 3 assertions)
Process finished with exit code 0
```

### Ilustración 6: Prueba unitaria modelo disco duro

## Anexo 5: Prototipo de interfaz.



**Ilustración 7: Prototipo interfaz configuración general**

The screenshot shows a form titled 'Configuración de vínculo con GRHS'. It contains five input fields arranged in two rows. The first row has 'Host' (localhost), 'Puerto' (5432), and 'Nombre de la base de datos' (grhs\_new). The second row has 'Usuario' (postgres) and 'Contraseña' (masked with dots). A green 'Guardar' button is positioned at the bottom right of the form.

**Ilustración 8: Prototipo de interfaz configuración de la conexión con la BD**