



Universidad de las Ciencias Informáticas

## SISTEMA PARA EL DIAGNÓSTICO DE COMPATIBILIDAD DE HARDWARE PARA EL SISTEMA OPERATIVO NOVA

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

**Autores:** Adrian Crúz Machin  
Roberto Menéndez Ruiz

**Tutor:** Ing. Angel Goñi Oramas  
Ing. Eduardo Alejandro Cuesta LLanes

*Ciudad de la Habana, 2012*

## Frase



*“La UCI tiene la misión de graduar profesionales comprometidos con la Patria, ser fieles a su tiempo, al ideario antiimperialista y latinoamericanista del más universal de los cubanos.”*

A handwritten signature in black ink, which appears to be 'Fidel Castro', written on a light-colored rectangular background.

## **Declaración de Autoría**

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

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año 2012.

Adrian Crúz Machin

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

Roberto Menéndez Ruiz

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

Ing. Angel Goñi Oramas

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

Ing. Eduardo A. Cuesta LLanes

\_\_\_\_\_  
Firma del Co-Tutor

## Agradecimientos

### Machin

*A mis tutores Angel Goñi Oramas y Eduardo Alejandro Cuesta brindarme su apoyo siempre y la información necesaria para realizar esta investigación.*

*Agradecerle a mi amor y futura esposa Lisandra Cala Hernández, por su entrega total durante estos 5 años de carrera y para la creación de este trabajo, por guiarme siempre, por su paciencia y dedicación, por estar a mi lado cuando más lo necesitaba, fueran momentos buenos o malos, por permitirme ser una persona mejor y por enseñarme a enfrentar los problemas que surgen en el camino de la vida como estudiante universitario, a ti mi amorcito lindo un beso con todo en amor del mundo.*

*A mis compañeros del proyecto, en especial a Susana, Firvida, Puente y Miranda por ayudarme y brindarme sus conocimientos, a Alexander por brindarme su tecnología y al rey del Inkscape Nelio.*

*A mis a todos mis compañeros de grupo, en especial a todos los que nos hemos mantenido estudiando juntos desde nuestro primer año en la antigua facultad 10.*

*A mi mamá por ser la mamá más linda que tiene este mundo, por depositar toda su confianza en mí y permitirme ser su orgullo.*

*A Víctor por su apoyo incondicional.*

*A mis hermanitas, a Tati por darme todo su apoyo y ayuda incondicional junto a su esposo Dany, a Araymita por ser tan pegajosa conmigo y por preguntarme siempre “tata cuando me vas a llevar a la piscina de la UCI”.*

*A alguien muy especial a la única abuelita que me queda en este mundo, que esperaba con muchísimas ansias este momento.*

*A mi tío Rubén por darme un empujoncito cuando más lo necesitaba y su ayuda incondicional.*

*A mi segunda mamá Cristy porque siempre he podido contar con ella para todo, por apoyarme siempre, a Picholo por ser mi segundo papá y apoyarme de igual forma, a Aliomar, Mamahaya, tío Vicente por estar siempre atentos.*

*A todas las personas allegadas a Damarys y Geraldo por brindarme su conocimiento, a Israel y Manuel por brindarme su tecnología al igual que a Yuremis y Arian.*

*A todos mis amigos de la universidad...*

*Mis más sinceros agradecimientos a todas aquellas personas que me brindaron su ayuda incondicional y aportaron su granito de arena.*

## **Roberto**

*Le agradezco a mi madre y a mi hermana por tanta preocupación y comprensión en el transcurso de estos cinco años.*

*A los todos aquellos del Grupo 1, que sobrevivieron a la locura de estos años de estudiante.*

*A los que se convirtieron en la familia de la UCI (Reidiel, Héctor, Ricardo, Tomás, Guillermo, el tocayo...) por su apoyo durante estos años de estudios universitarios.*

*A Lisandra por la ayuda prestada en el momento en que más se necesitaba.*

*A todos muchas gracias.*

## **Dedicatoria**

### **Machin**

*A Fidel Castro Ruz y la Revolución Cubana por permitirme ser parte de la Primera Universidad de la Batalla de Ideas.*

*A mi mamá y mi abuelita que son lo que más quiero en el mundo.*

*A mis hermanas por ayudarme en todo.*

*A mi novia Lisandra por su incansable apoyo.*

*A toda mi familia por estar siempre para mí.*

### **Roberto**

*A mi familia que siempre estuvo cuando las cosas salían bien, principalmente en los casos en que el resultado no era el esperado.*

*Especialmente a mi padre, aunque no lo pueda tener ya conmigo.*

## Resumen

Ante la necesidad de migrar Cuba hacia Aplicaciones de Código Abierto y valorando la importancia de obtener una correcta adaptación y aceptación de la migración, se propone mejorar la eficiencia del proceso de diagnóstico de compatibilidad de hardware para el sistema operativo Nova. El Sistema de Certificación y Homologación de Hardware (SICEH) se encarga de almacenar la información de los componentes con su categoría de certificación y dispositivos con características semejantes ya certificados.

Los usuarios pueden interactuar con SICEH de forma manual, sin embargo necesitan una herramienta que desde su estación de trabajo le permita conocer la compatibilidad del sistema operativo Nova con los componentes de su computadora, para facilitar el proceso de certificación y homologación de hardware.

En el presente trabajo se realiza un estudio de herramientas para la recopilación de información en los sistemas operativos GNU/Linux teniendo en cuenta sus características y dispositivos que obtiene, así como, lenguajes y herramientas para su desarrollo. Como resultado se implementa el Sistema de Diagnóstico de Compatibilidad de Hardware (SistDCH), que permite realizar un levantamiento de la información de los dispositivos existentes en una computadora, diagnostica los componentes obtenidos asignándole una categoría de certificación, además de realizar la búsqueda de homólogos para los dispositivos no compatibles, mediante el consumo de los servicios de SICEH.

**Palabras clave:** certificación, compatibilidad, diagnóstico, hardware.

# Índice

<b>Introducción</b> .....	<b>10</b>
<b>Capítulo 1. Diagnóstico y Compatibilidad de Hardware</b> .....	<b>14</b>
1.1. Definición del proceso de diagnóstico de compatibilidad de hardware en los sistemas operativos GNU/Linux. ....	14
1.1.2. Proceso de diagnóstico de compatibilidad de Hardware en los Sistemas Operativos GNU/Linux. ....	14
1.2. Herramientas más usadas para la recopilación de información de hardware.....	16
a. HardInfo.....	16
b. Gnome Device Manager .....	17
c. SysInfo .....	18
d. Lshw .....	19
e. Agente OCS Inventory Next Generation .....	20
1.3. Tablas de comparación entre las herramientas para la recopilación de información de hardware. ....	21
1.4. Propuesta de solución a las irregularidades que presenta Lshw.....	26
1.4.1. Comandos para la obtención de los dispositivos no detectados por Lshw. ....	27
1.5. Metodología para el desarrollo del software.....	27
1.6. Herramientas, tecnologías y lenguaje a utilizar.....	29
1.6.1. Lenguaje de programación. ....	29
1.6.2. Herramientas. ....	30
1.7. Estudio de los aspectos arquitectónicos para el diseño del sistema. ....	32
1.8. Protocolo de comunicación entre sistemas.....	33
Conclusiones parciales.....	34
<b>Capítulo 2. Definición y desarrollo del Sistema de Diagnóstico de compatibilidad de Hardware</b> .....	<b>35</b>
2.1. Sistema de diagnóstico de compatibilidad de hardware propuesto. ....	35
2.2. Proceso de desarrollo del proyecto basado en SXP. ....	36
2.3. Modelo de dominio. ....	37
2.4. Lista de Reserva del Producto (LRP).....	38
2.5. Historias de Usuario.....	40
2.5.1. Historia de usuario gestionar levantamiento de la información del hardware. ....	40
2.5.2. Historia de usuario obtener diagnóstico de compatibilidad del sistema.....	43

---

2.5.3. Historia de usuario buscar homólogos según características específicas del hardware.....	45
2.5.4. Historia de usuario imprimir reporte del levantamiento de hardware. ....	48
2.5.5. Historia de usuario imprimir reporte del diagnóstico de hardware. ....	49
2.5.6. Historia de usuario imprimir reporte de la homologación de los dispositivos. ....	51
2.6. Plan de Releases.....	52
2.7. Diseño con metáforas. ....	52
2.8. Diagrama de componentes. ....	54
Conclusiones parciales.....	55
<b>Capítulo 3: Diseño y realización de pruebas a SistDCH. ....</b>	<b>56</b>
3.1. Aplicación del SistDCH en un escenario real .....	56
3.2. Casos de prueba. ....	60
<input type="checkbox"/> Casos de prueba para HU gestionar levantamiento de la información del hardware.....	60
<input type="checkbox"/> Casos de prueba para HU Obtener diagnóstico de compatibilidad de hardware.....	63
<input type="checkbox"/> Casos de prueba para HU buscar homólogos según características específicas del hardware.....	64
<input type="checkbox"/> Casos de prueba para HU imprimir reporte del levantamiento del hardware.....	66
<input type="checkbox"/> Casos de prueba para HU imprimir reporte del diagnóstico de hardware.....	67
<input type="checkbox"/> Casos de prueba para HU imprimir reporte de la homologación de los dispositivos. ....	67
3.3. Resultados.....	69
Conclusiones parciales.....	69
<b>Conclusiones .....</b>	<b>70</b>
<b>Recomendaciones .....</b>	<b>71</b>
<b>Referencias bibliográficas .....</b>	<b>72</b>
<b>Bibliografía.....</b>	<b>75</b>
<b>Anexos.....</b>	<b>78</b>
<b>Glosario de términos. ....</b>	<b>81</b>

## Introducción

Con el estadio tecnológico hace alrededor de 2.500.000 billones de años se definió como tecnología a todos aquellos instrumentos creados por el ser humano, ayudando a adaptarse a sus necesidades (1). Al transcurrir las décadas fueron aumentando y sofisticándose con el conocimiento científico; lo que desató una larga carrera por alcanzar mayor dominio sobre las tecnologías, infiriendo en todas las esferas de la sociedad y desempeñando un papel fundamental.

En la actualidad son muchas las posibilidades que brindan las tecnologías de la información y las ventajas que ofrecen para los países de menos recursos. Facilitando la comunicación entre las personas e instituciones, no solo a nivel de la comunidad sino también a nivel mundial. Se ha conseguido eliminar barreras espaciales y temporales<sup>1</sup>, que hace poco tiempo se consideraban infranqueables.

Cuba, ha defendido siempre el concepto de que el uso masivo de las Tecnologías de la Información y las Comunicaciones (TIC) no es un fin, sino una herramienta poderosa para lograr el desarrollo (2). Como resultado de la política de bloqueo económico y financiero que el gobierno de Estados Unidos mantiene sobre Cuba, y ante el dominio de Microsoft sobre el mercado internacional de sistemas operativos, para el país resulta una utopía mantenerse a la par del modelo consumista impuesto por las economías capitalistas en el área de las TIC. Súmese a esto el riesgo que implica para la seguridad y la soberanía nacional la dependencia absoluta de productos extranjeros que día a día juegan un papel cada vez más importante en las esferas gubernamentales, productivas y sociales (3).

El uso de las tecnologías libres más que una alternativa, es una necesidad en aras de impulsar el desarrollo de soluciones informáticas que solventen los problemas del país. Para lograr la soberanía tecnológica<sup>2</sup>, en abril del año 2004 el Consejo de Ministros adoptó el acuerdo 084/2004 indicando al Ministerio de Informática y las Comunicaciones (MIC), el proceso paulatino de migración a código abierto (4). Se comienza desde entonces a trabajar en una estrategia para el uso de Software Libre y Aplicaciones de Código Abierto, que a diferencia del software propietario<sup>3</sup>, sus licencias aseguran que el código pueda ser modificado y mejorado por cualquier persona o grupo de personas con las habilidades correctas, el conocimiento es de dominio público (5).

---

<sup>1</sup> Las personas pueden comunicarse y colaborar de forma síncrona o asíncrona, sin limitaciones espacio-temporales, mediante las redes de datos como medios (Intranet e Internet), las aplicaciones hipertextuales como soporte (correo electrónico, web, chat) (6).

<sup>2</sup> Soberanía Tecnológica: capacidad de un país para desarrollarse en dicho campo en forma autónoma. No supone independencia absoluta, sino capacidad decisonal sobre su uso y desarrollo (7).

<sup>3</sup> Software Privativo: programas o aplicaciones en el que el usuario no puede acceder al código fuente o tiene un acceso restringido y, por tanto, se ve limitado en sus posibilidades de uso, modificación y redistribución (8).

Ante la necesidad de migrar hacia aplicaciones de código abierto y valorando la importancia de obtener una correcta adaptación y aceptación de todas las entidades, ministerios y empresas nacionales que aún no han migrado, el departamento de Migración y Soporte (MS), perteneciente al Centro de Software Libre (CESOL), en la Universidad de las Ciencias Informáticas (UCI), asumió la responsabilidad de llevar a cabo la migración (9). Proponiendo automatizar todas las actividades que se realizan, mediante el desarrollo de una plataforma que integre todas las herramientas necesarias (4).

La Guía Cubana para la Migración a Software Libre, en la cual se define la metodología para implementar la migración eficientemente propone cuatro etapas fundamentales: Preparación, Migración Parcial, Migración Total y Consolidación por las que se debe transitar. Asociado a cada una de las etapas existen flujos de trabajo que definen las tareas que se deben ejecutar en el proceso de migración. Una de las tareas del flujo de evaluación que tiene mayor peso en la etapa de preparación, es el análisis del estado actual de los sistemas informáticos de la institución en cuanto a hardware, con el propósito de garantizar su grado de compatibilidad (9), proponiendo realizar la Certificación y Homologación de Hardware (CHH) para garantizar la compatibilidad de los componentes con el sistema operativo Nova GNU/Linux.

En la CHH se deben obtener las características de todos los componentes con los que cuenta la computadora, generando un conjunto de listas, enmarcando los distintos tipos de compatibilidad. Concluido el proceso, los usuarios pueden dirigirse a aquellas aplicaciones existentes donde está el resultado de la certificación y realizar una búsqueda para verificar la compatibilidad de sus componentes. Estos encuentran trabajoso conocer de forma manual el grado de compatibilidad de sus estaciones de trabajo con el sistema operativo Nova, surgiendo la necesidad de garantizar desde sus estaciones de trabajo, la obtención de un diagnóstico de forma sencilla. Dada la **situación problemática** planteada anteriormente surge el siguiente **problema a resolver**: ¿Cómo contribuir a mejorar la eficiencia del proceso de diagnóstico de compatibilidad de hardware para el sistema operativo Nova?

El **objeto de estudio** de la investigación recae en el proceso de diagnóstico de compatibilidad de hardware, teniendo como **campo de acción** el proceso de diagnóstico de compatibilidad de hardware en sistemas operativos GNU/Linux.

Para el desarrollo de la investigación se trazó como **objetivo general** desarrollar la herramienta SistDCH que pueda contribuir a aumentar la eficiencia del proceso de diagnóstico de compatibilidad de hardware para el sistema operativo Nova.

Para poder dar cumplimiento al objetivo general anteriormente planteado se han trazado los siguientes **objetivos específicos**:

1. Construir el marco teórico de la investigación, a partir del análisis realizado sobre los sistemas de diagnóstico de compatibilidad de hardware en sistemas operativos GNU/Linux.
2. Desarrollar el sistema SistDCH.
3. Realizar pruebas de aceptación a SistDCH en el sistema operativo GNU/Linux Nova.

Los anteriores objetivos se concretan en las siguientes **tareas de la investigación**:

1. Revisión y Análisis de la documentación de los sistemas similares más utilizados a nivel mundial.
2. Selección de las herramientas y Metodologías de Desarrollo necesarias para el desarrollo del sistema.
3. Análisis y definición de las funcionalidades del sistema a partir de la descripción del proceso de negocio y los requerimientos funcionales y no funcionales.
4. Diseño y descripción de la arquitectura del sistema SistDCH.
5. Realización de pruebas de aceptación al sistema.

Por lo que se plantea como **idea a defender**:

Con el desarrollo de SistDCH se puede contribuir a aumentar la eficiencia en el proceso de diagnóstico de compatibilidad de hardware en el sistema operativo Nova.

Para el desarrollo de esta investigación se utilizó el siguiente **método teórico**:

**Analítico-Sintético**: está integrado por el desarrollo del análisis y la síntesis, mediante el cual se descompone un objeto, fenómeno o proceso en los principales elementos que lo integran para analizar, valorar y conocer sus particularidades, y simultáneamente a través de la síntesis, se integran vistos en su interrelación como un todo. Este método fue de gran utilidad para analizar cada uno de los sistemas existentes y realizar un estudio minucioso del asunto. Permitiendo hacer un resumen de las bibliografías consultadas en el desarrollo de la investigación, definiendo ideas concretas que se apliquen a la propuesta de solución.

El presente trabajo de diploma está compuesto por la Introducción, 3 Capítulos, Conclusiones, Recomendaciones, Referencias bibliográficas, Bibliografía, Anexos y Glosario de términos, todo ello bien estructurados y de forma organizada para un buen entendimiento del lector.

**Capítulo 1:** Diagnóstico y Compatibilidad de Hardware

En este capítulo se presentan los resultados del análisis documental realizado sobre los sistemas informáticos existentes vinculados al campo de acción, que poseen similares características o que pretenden desempeñar una labor semejante al sistema que se desea desarrollar, además de mejoras potenciales que se puedan agregar a la solución propuesta. También se realiza un estudio sobre la metodología, herramientas y tecnologías, que justifican la selección empleada en la construcción de la aplicación informática.

**Capítulo 2:** Definición y desarrollo del Sistema de Diagnóstico de compatibilidad de Hardware.

En este capítulo se presenta la descripción de las Historias de Usuario identificadas. Se realiza una explicación detallada del proceso de planificación, donde se utilizaron estas para realizar el análisis y la estimación del tiempo de desarrollo de cada una, a través de un proceso iterativo que se ajustó en todo momento a la velocidad de trabajo del equipo de desarrollo.

**Capítulo 3:**

En este capítulo se realiza un diagnóstico y compatibilidad de hardware de una computadora con el sistema operativo Nova, haciendo uso de la aplicación desarrollada en la presente investigación. Se aplican las pruebas a cada una de las historias de usuario correspondientes a cada iteración definida. Tienen como objetivo validar que los casos de prueba de aceptación al sistema cumplan con el funcionamiento esperado, permitiendo al cliente determinar su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

## Capítulo 1. Diagnóstico y Compatibilidad de Hardware.

En este capítulo se presentan los resultados del análisis documental realizado sobre los sistemas informáticos existentes vinculados al campo de acción, que poseen similares características o que pretenden desempeñar una labor semejante al sistema que se desea desarrollar, además de mejoras potenciales que se puedan agregar a la solución propuesta. También se realiza un estudio sobre la metodología, herramientas y tecnologías, que justifican la selección empleada en la construcción de la aplicación informática.

### 1.1. Definición del proceso de diagnóstico de compatibilidad de hardware en los sistemas operativos GNU/Linux.

Un **proceso** es un conjunto de actividades o eventos que se realizan con un determinado fin (10).

Según la definición de **diagnóstico** en un contexto médico y ajustado a la presente investigación, es el resultado del análisis que se realiza en una primera instancia, permite conocer las características específicas de una situación determinada para así poder actuar en consecuencia, sugiriendo tratamiento o no (11).

Según la RAE<sup>4</sup> compatible en términos informáticos es que un sistema pueda funcionar directamente con otro dispositivo, aparato o programa. Definiendo como **compatibilidad** la posibilidad de hacer funcionar un programa informático en un ordenador o conjuntamente con otros programas sin necesidad de hacer modificaciones (12).

Definición de proceso de diagnóstico y compatibilidad de hardware:

Teniendo en cuenta los conceptos descritos anteriormente se define **el proceso de diagnóstico de compatibilidad de hardware en los sistemas operativos GNU/Linux**, como una serie de pasos progresivos e interdependientes con acciones dirigidas a conocer las características específicas del hardware, que posibilita el correcto funcionamiento de los Sistemas Operativos GNU/Linux, teniendo como fin que los programas funcionen sin necesidad de hacer ninguna modificación o determinar las características específicas de la situación para así poder actuar en consecuencia.

#### 1.1.2. Proceso de diagnóstico de compatibilidad de Hardware en los Sistemas Operativos GNU/Linux.

En la etapa de preparación de la Metodología Cubana para la Migración a Aplicaciones de Código

---

<sup>4</sup> RAE: Real Academia de la Lengua Española.  
[http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=compatibilidad](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=compatibilidad).

Abierto se realiza el proceso de CHH, clasificando los dispositivos de los equipamientos ofimáticos. El mismo genera un conjunto de listas que enmarcan los tres tipos de compatibilidad (hardware compatible, no compatible y compatible con recomendaciones) (13). Una vez obtenidas las listas se guardan en la base de datos del Sistema de Certificación de Hardware (SICEH), concluyendo de esta manera el proceso de CHH.

Concluido el proceso, los usuarios pueden obtener las características de sus dispositivos mediante una herramienta para la obtención de información del hardware. Se dirigen a SICEH y realizan una búsqueda para ver el nivel de compatibilidad del sistema operativo con respecto a su equipamiento ofimático. Una vez obtenido el resultado de compatibilidad, pueden existir partes del hardware de su ordenador que no son compatibles. Existiendo la necesidad de realizar una nueva búsqueda para encontrar un homólogo del componente que no le es funcional. (Ver figura 1.1)

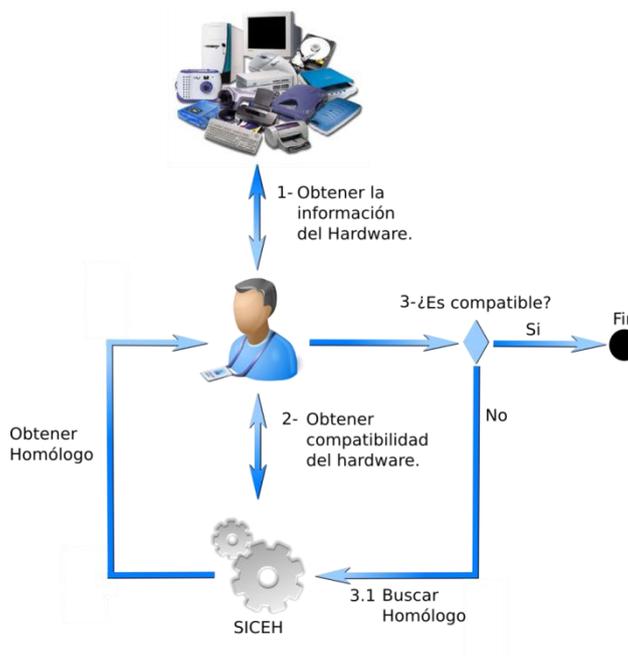


Figura 1.1. Descripción del Diagnóstico de Compatibilidad de Hardware.

A continuación se enumeran los pasos que rigen el proceso de diagnóstico de hardware para el sistema operativo Nova:

1. Tener conocimiento del hardware con el que cuenta el equipamiento ofimático.
2. Realizar una búsqueda en SICEH para identificar el grado de compatibilidad.
3. De existir al menos un dispositivo no compatible:
  - 3.1. Realizar una nueva búsqueda para encontrar un hardware con características

similares (Homologación).

Definido el proceso, se realizó una búsqueda bibliográfica para encontrar una herramienta o sistema que pudiera automatizar estos pasos lógicos, no encontrándose un sistema capaz de realizar el proceso de diagnóstico de compatibilidad de hardware de forma automatizada en los sistemas GNU/Linux. Teniendo en cuenta que existen herramientas capaces de ayudar a cubrir una de las necesidades que se describen en el paso 1 (recopilación de la información del hardware) se decide realizar un estudio de las más usadas.

Una vez identificadas las herramientas para la recopilación de la información de hardware existentes en los sistemas GNU/Linux, se instalarán en el sistema operativo Nova para comprobar su funcionamiento y características, con el objetivo de obtener la idónea para tener conocimiento del hardware con el que cuenta el equipamiento ofimático.

## **1.2. Herramientas más usadas para la recopilación de información de hardware.**

Las herramientas de recopilación de información de hardware están especialmente diseñadas a realizar reportes del hardware existente en un equipamiento ofimático. Concluido su período de instalación, visualiza la información detallada del hardware instalado en la computadora (procesador, memoria, RAM, sistema operativo, configuración de red, entre otros). Con el uso de estas herramientas se percibe a simple vista la facilidad de obtener la información del hardware, en una computadora de manera automatizada.

En la actualidad existen aplicaciones informáticas que visualizan y manipulan reportes con la información referente al hardware. Es importante tener en cuenta las herramientas existentes para estudiar su comportamiento cuando se va a realizar el proceso de visualización y generación de reportes.

### **Herramientas para la recopilación de información de hardware:**

#### **a. HardInfo**

Es una herramienta libre, de código abierto y gratuito. Permite obtener información acerca del hardware de un ordenador. Está publicado bajo *GNU Public License* (GPL). Implementa las librerías gráficas GTK+ para mostrar los controles de la interfaz como se muestra en la figura 1.2, lo que le permite integrarse con los escritorios GNOME y Xfce.

A través de dicha herramienta se puede observar la información referente a los dispositivos PCI, USB, impresoras y módulos del núcleo, así como del procesador, interfaces de red, directorios

compartidos, sistema operativo y de ficheros. Brinda la posibilidad de generar reportes y guardarlos en .HTML o en .TXT (14).

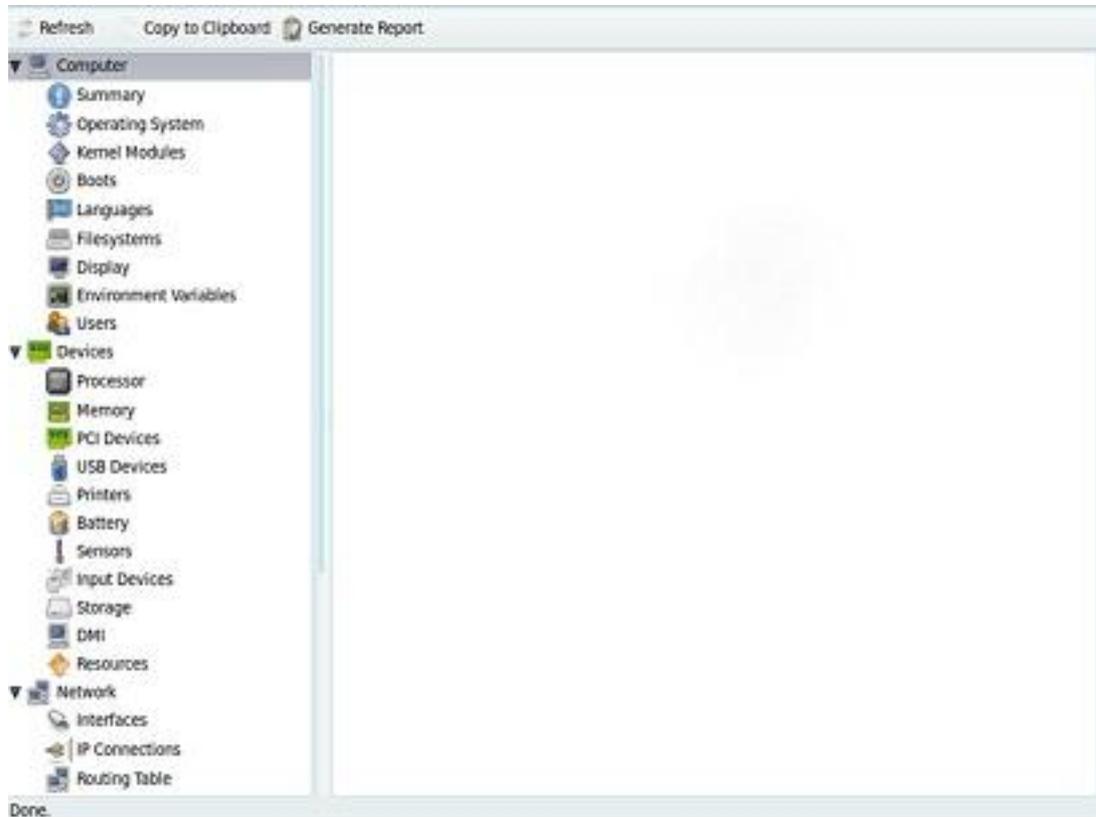


Figura 1.2: Interfaz de la herramienta de recopilación de la información del hardware HardInfo.

Los autores del presente trabajo opinan que HardInfo muestra algunos dispositivos carentes de características para garantizar la compatibilidad en el proceso de migración en la etapa de preparación, como es el caso del BIOS y la tarjeta madre que no muestran el número de serie y la versión respectivamente, por citar algunos ejemplos. A pesar de no contar con todos los requisitos necesarios, es una herramienta bien estructurada y fácil de usar para los usuarios.

## b. Gnome Device Manager

“Es un programa de GNOME para gestionar dispositivos y controladores. Está inspirado en el *hal-device-manager*, del proyecto HAL<sup>5</sup>, pero reescrito en C por eficiencia y un punto de vista que hace que gestione los dispositivos en vez de mostrar sólo la información” (15). (ver figura 1.3). Desarrollado para diferentes arquitecturas como amd64, armel, i387 y ia6. Realiza un análisis amplio con el fin de determinar las capacidades del hardware.

Aunque no permite generar y almacenar reportes, si un dispositivo en la PC o un dispositivo

<sup>5</sup> Hardware Abstraction Layer: era un proyecto de software que proveía una capa de abstracción de hardware para sistemas Unix-like.

periférico no funcionan correctamente, es útil usar *Gnome Device Manager* para encontrarlo. Luego tomando el nombre del hardware y realizando una búsqueda en Google o en foros populares de Linux se puede encontrar una solución (16).

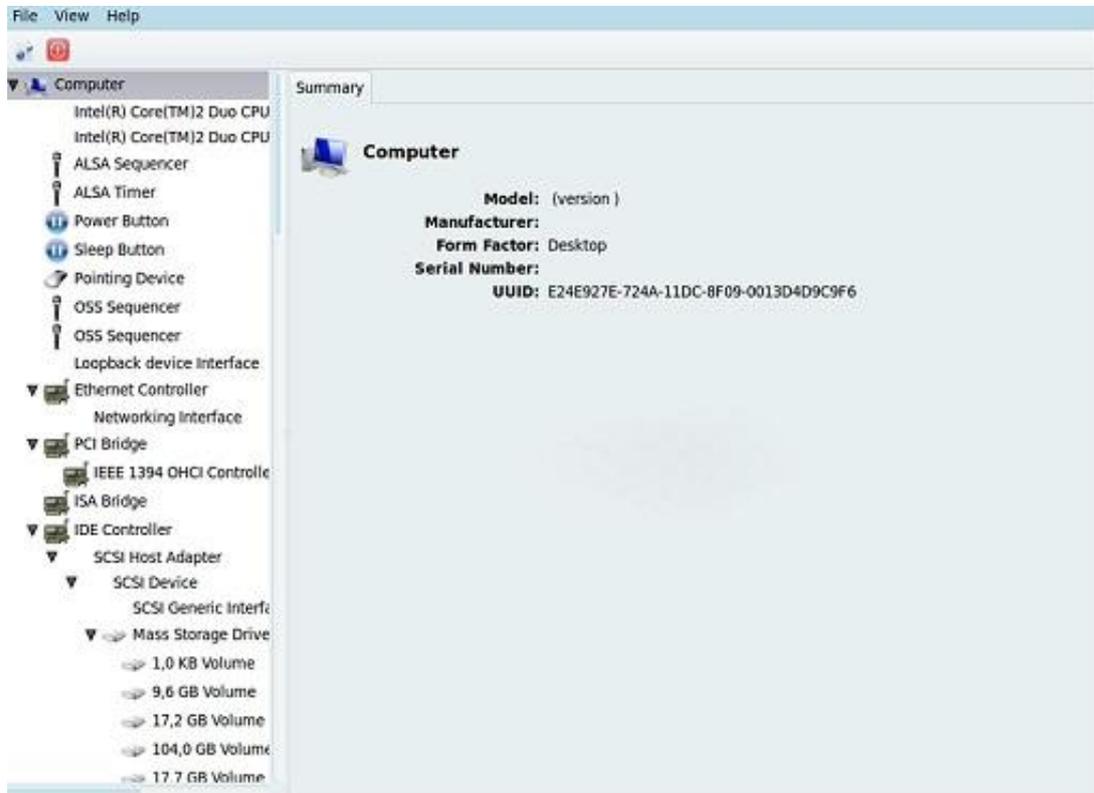


Figura 1.3: Interfaz de la herramienta de recopilación de la información del hardware *Gnome Device Manager*.

Gnome Device Manager es una herramienta que muestra la información de cada dispositivo de forma poco estructural. Al usuario se le hace difícil encontrar las características principales de cada componente, porque no aparece con una estructura adecuada. No visualiza la información del monitor.

### c. SysInfo

Proporciona una forma sencilla de obtener las estadísticas generales del sistema. Conociendo la información referente al sistema operativo, módulo del núcleo, controladores de puertos USB, controlador de audio, estado de la memoria RAM y de intercambio (17) (ver figura 1.4).

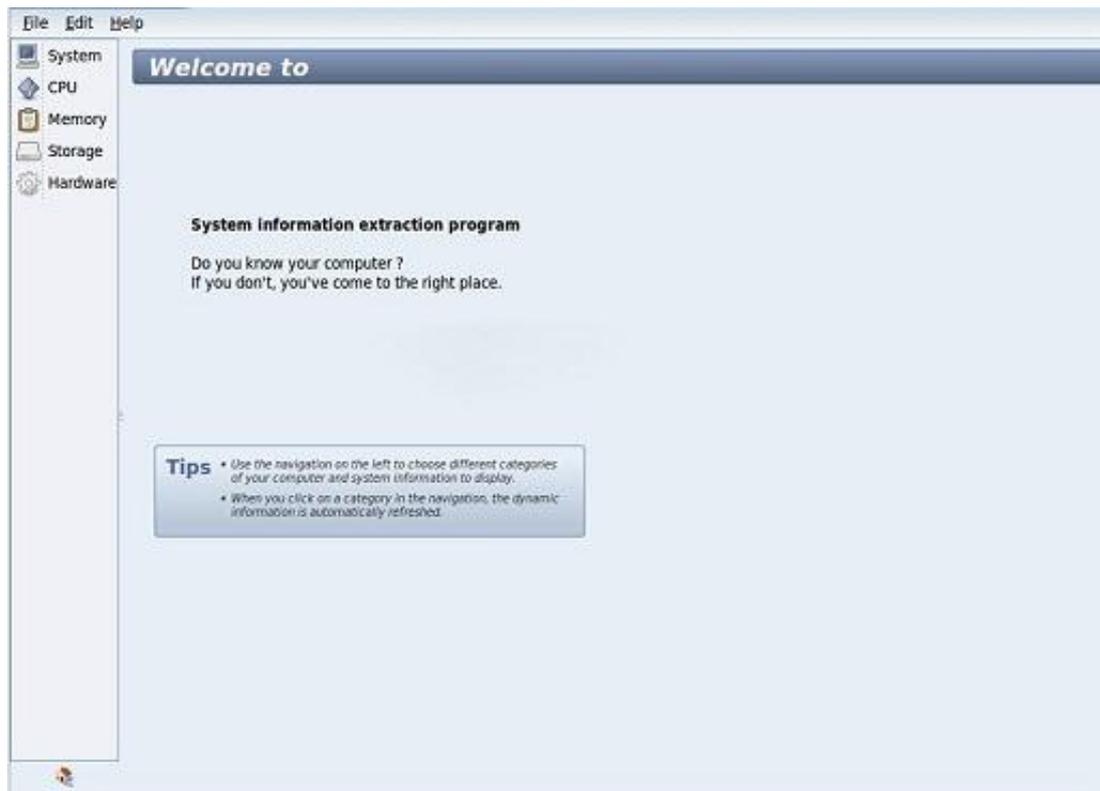


Figura 1.4: Interfaz de la herramienta de recopilación de la información del hardware SysInfo.

Es una herramienta que muestra una información muy elemental sobre los componentes de la computadora. Además, no cubre las necesidades de la investigación para la actividad de compatibilidad de software en el proceso de migración.

#### d. Lshw

Es un comando de Linux que ofrece detalles de todo el hardware de la computadora. Los datos facilitados por el Lshw ofrecen toda la gama de procesadores, memoria, slots, sonido integrado, tarjeta de vídeo entre otras. Para obtener todos los detalles del hardware, es necesario ejecutar el comando lshw como root (o súper-usuario). Contiene gran cantidad de información sobre todos los aspectos del hardware de la computadora.

Lshw es un comando que ofrece una salida en múltiples formatos de archivo. Por ejemplo, se puede conseguir la salida en HTML, XML, texto, entre otros. Tiene una interfaz gráfica de usuario equivalente, disponible en el paquete lshw-gtk (ver figura 1.5). Este complemento visual proporciona los detalles al usuario de manera amistosa y estructurada (18).

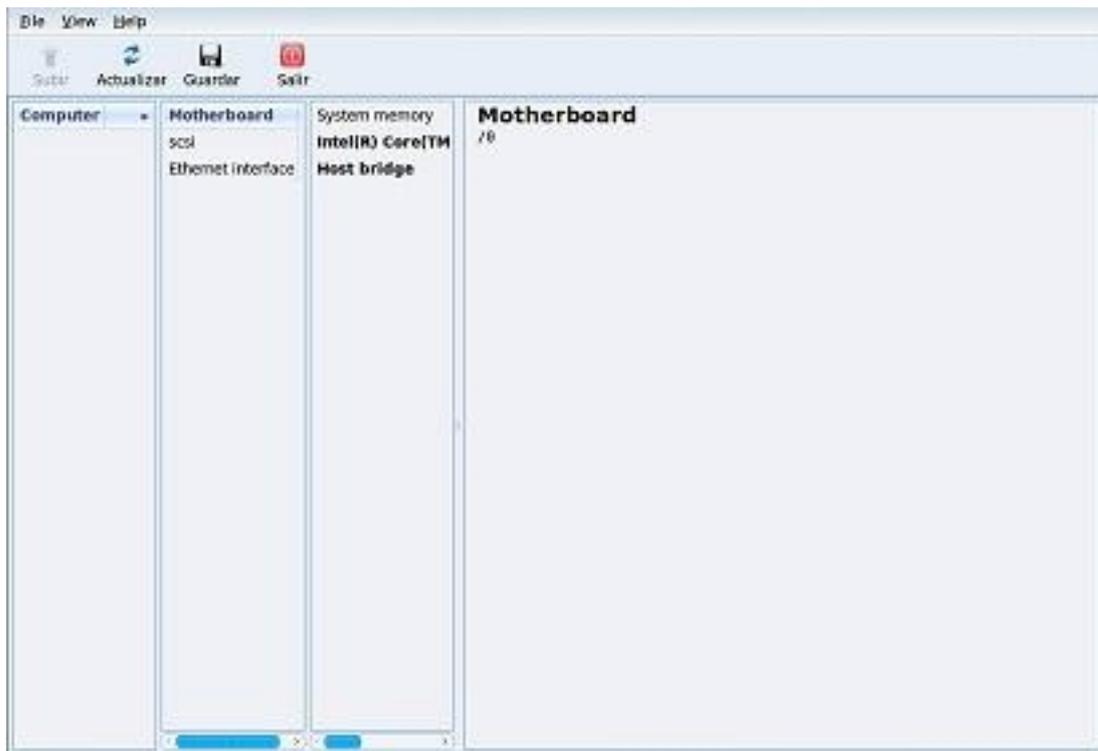


Figura 1.5: Interfaz visual del comando Lshw para la recopilación de la información de hardware.

Lshw aún ejecutado como súper usuario no muestra los dispositivos de entrada, ni las características del monitor. A diferencia de las estudiadas anteriormente permite exportar los resultados a diferentes formatos. Mantiene una estructura uniforme al mostrar las características de cada componente, obteniendo la suficiente información para tenerla en cuenta a la hora de darle solución al objetivo propuesto.

### e. Agente OCS Inventory Next Generation

OCS Inventory NG es una herramienta liberada bajo licencia GNU GPLv2. Permite mantener un control de los programas y hardware instalado en las máquinas en una red, ofrece un reporte del hardware y software instalados, el cual puede ser visualizado a través de una interfaz web. Es una aplicación cliente/servidor, el servidor se ejecuta sobre APACHE/MYSQL/PHP/PERL, bajo los sistemas operativos GNU/Linux o Windows NT.

Recopila información mediante OCSAgent (agente que envía el inventario al servidor OCSInventory). El cliente se puede ejecutar en los sistemas operativos Microsoft Windows 95/98/Me/NT4/2000/XP/2003 o en GNU/Linux.

Tiene las siguientes características fundamentales (19):

1. Información pertinente de los inventarios.
2. Sistema de despliegue de gran alcance que permite la distribución de la instalación de software o scripts en las computadoras, sin sobrecargar la red.
3. Consola Web de administración.
4. Los sistemas operativos múltiples de apoyo, incluyendo Microsoft Windows, GNU/Linux, BSD, Solaris de Sun, IBM AIX, HP-UX, Mac OS X.
5. Arquitectura con las normas vigentes, HTTP / HTTPS y formato de datos XML.
6. Servicio Web accesible a través de la interfaz SOAP.
7. Plugins de apoyo a través de API.
8. Red de exploración y sincronización con GLPI.

Ha sido usada por el departamento de Migración y Soporte, mostrando excelentes resultados en cuanto a los reportes solicitados de hardware y software de las computadoras que se desean migrar en las entidades involucradas en el proceso (20).

OCSAgent es una herramienta muy poderosa para tener el control y el inventario de hardware y software de las estaciones de trabajo para un sistema operativo privativo, no ocurriendo así en los sistemas GNU/Linux. Obtiene información suficiente, pero no comparable con las herramientas de recopilación de información destinadas a estos.

### **1.3. Tablas de comparación entre las herramientas para la recopilación de información de hardware.**

Para culminar el estudio de las herramientas más usadas en la recopilación de información de hardware, se hace necesaria una comparación de las 5 resultantes teniendo en cuenta las características, dispositivos y funcionalidades. Durante el desarrollo de la misma se irán descartando las herramientas que no cumplen con el objetivo trazado al comenzar su estudio.

La selección de la herramienta idónea se realizará mediante tres tablas comparativas:

1. **Tabla 1.1** Características que recoge en al menos un dispositivo.

2. **Tabla 1.2** Dispositivos y funcionalidades que obtienen.
3. **Tabla 1.3** Propuesta de la herramienta que formará parte del desarrollo de la investigación en cuanto a características que obtiene de cada dispositivo.

Con el fin de tomar una decisión final sobre la herramienta idónea en el proceso de migración en la etapa de preparación que garantice la compatibilidad del sistema operativo GNU/Linux Nova, se tienen en cuenta características que son elementales en SICEH para emitir una respuesta sobre el grado de compatibilidad.

Características fundamentales que debe tener la herramienta para su selección:

1. Generar reportes.
2. Obtener de cada dispositivo:
  - Modelo.
  - Versión.
  - Fabricante.
  - Tipo de Dispositivo.

### **Tablas Comparativas**

a. Para la siguiente tabla comparativa se tiene en cuenta las herramientas en cuanto a las características que recoge en al menos un dispositivo.

<b>Herramienta</b>	<b>Características</b>														<b>Total</b>	<b>Orden de prioridad</b>
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>		
HardInfo	X	X	X	X	-	X	X	X	-	-	-	-	-	-	7	3
Gnome Device Manager	X	X	X	X	X	X	X	X	-	-	-	-	X	-	9	2
SysInfo	-	X	X	X	-	-	X	-	-	-	-	-	-	-	4	5
Lshw	X	X	X	X	X	X	X	X	X	X	X	X	X	-	13	1
Agente OCS Inventory	X	X	X	X	X	-	X	-	-	-	-	-	-	-	6	4

Tabla 1.1: Elección mediante las características que recoge en al menos un dispositivo.

**Leyenda:** La (X) representa la presencia de la característica en al menos un dispositivo de la

herramienta que se encuentra en comparación, y el (-) que no aparece en ningún dispositivo dicha característica.

**Numeración de las características a tener en cuenta en la Tabla 1**

- |                          |                |
|--------------------------|----------------|
| 1. Producto              | 8. Drivers     |
| 2. Tipo                  | 9. Resources   |
| 3. Vendedor o Fabricante | 10. Whidth     |
| 4. Versión o Modelo      | 11. Clock      |
| 5. Serial                | 12. Slot       |
| 6. Compatibilidad        | 13. Bus Info   |
| 7. Tamaño                | 14. Resolución |

Resultado obtenido de la tabla 1.1:

Durante la realización de la tabla se pudo verificar que no todas las herramientas obtienen de cada dispositivo la misma cantidad de características. Por lo general, no todos tienen uniformidad a la hora de obtenerlas, siendo el comando **Lshw** el que muestra, para todos sus dispositivos, la mayor cantidad de características uniformemente. Se descartan a las herramientas restantes.

b. En la tabla comparativa 1.2 se tiene en cuenta las herramientas en cuanto a dispositivos y funcionalidades que recoge.

Herramienta	Dispositivos y funcionalidades														Total	Orden de prioridad
	1	2	3	4	5	6	7	8	9	10	11	12	13	14		
HardInfo	X	X	X	X	X	X	X	X	X	-	X	X	X	X	12	1
Gnome Device Manager	-	X	X	X	X	X	-	-	-	-	X	-	X	-	7	5
SysInfo	-	-	X	X	X	X	-	-	X	-	X		X	X	8	4
Lshw	-	-	X	X	X	X	X	X	X	X	X	-	X	X	11	2
Agente OCS Inventory	-	-	X	X	X	X	X	-	X	-	X	X	X	X	9	3

Tabla 1.2: Elección mediante a los dispositivos que obtiene y funcionalidades.

**Leyenda:** La (X) representa la presencia de la característica en al menos un dispositivo de la herramienta que se encuentra en comparación, y el (-) que no aparece en ningún dispositivo dicha característica.

**Numeración de los dispositivos que obtiene y funcionalidades**

- |            |                |
|------------|----------------|
| 1. Monitor | 8. MotherBoard |
| 2. Mouse   | 9. RAM         |

- |                     |                      |
|---------------------|----------------------|
| 3. Procesador       | 10. FireWire         |
| 4. Tarjeta de Vídeo | 11. Network          |
| 5. Tarjeta de Audio | 12. Impresora        |
| 6. Disco Duro       | 13. Controladores    |
| 7. BIOS             | 14. Generar Reportes |

Resultado obtenido de la tabla 1.2:

En esta tabla se realizó una comparación entre las herramientas, para analizar cuál obtiene la mayor cantidad de dispositivos y la funcionalidad de generar reportes, siendo la característica número uno que debe cumplir SICEH para dar un resultado de compatibilidad. Se obtiene como resultado a **HardInfo** y al comando **Lshw**, por ser los que más dispositivos obtienen y brindan la funcionalidad de generar reportes. Se descartan a las herramientas restantes.

c. La tabla 1.3 se realiza con el objetivo de llegar a la herramienta que formará parte del desarrollo de la investigación en cuanto a cantidad y uniformidad de las características que obtiene de cada dispositivo.

Teniendo en cuenta los resultados obtenidos de las tablas 1.1 y 1.2 se puede apreciar que la herramienta HardInfo y el comando Lshw brindan mayor información sobre el hardware de una computadora, siendo las adecuadas para realizar la comparación final. El resultado se alcanzará basándose en la que adquiera la mayor cantidad de características en los dispositivos más comunes y la uniformidad con que lo realiza. Se obtendrá la propuesta que formará parte del desarrollo de la investigación.

Tarjeta Madre	
HardInfo	Lshw
<ul style="list-style-type: none"> <li>Nombre</li> <li>Vendedor</li> </ul>	<ul style="list-style-type: none"> <li>Descripción</li> <li>Producto</li> <li>Vendedor</li> <li>Id Físico</li> <li>Versión</li> <li>Serie</li> <li>Slot</li> </ul>
CPU o Procesador	
HardInfo	Lshw
<ul style="list-style-type: none"> <li>Nombre</li> <li>Familia, Modelo o Stepping.</li> <li>Vendedor</li> </ul> Configuración: <ul style="list-style-type: none"> <li>Tamaño de la Caché</li> </ul>	<ul style="list-style-type: none"> <li>Descripción</li> <li>Producto</li> <li>Vendedor</li> <li>Id Físico</li> <li>Bus Info</li> </ul>

<ul style="list-style-type: none"> <li>• Frecuencia</li> <li>• Bogomips</li> <li>• Byte Order</li> </ul> <p>Features:</p> <ul style="list-style-type: none"> <li>• FDIV Bug</li> <li>• HTL Bug</li> <li>• FOOF Bug</li> <li>• Coma Bug</li> <li>• Has FPU</li> </ul> <p>Caché:</p> <ul style="list-style-type: none"> <li>• Lever1 (data)</li> <li>• Lever2 (instruction)</li> <li>• Lever3 (unified)</li> </ul> <p>Compatibilidad.</p>	<ul style="list-style-type: none"> <li>• Versión</li> <li>• Serie</li> <li>• Slot</li> <li>• Size</li> <li>• Capacidad</li> <li>• Width</li> <li>• Clock</li> <li>• Compatibilidad</li> </ul> <p>Caché:</p> <ul style="list-style-type: none"> <li>• Descripción</li> <li>• Id Físico</li> <li>• Slot</li> <li>• Size</li> <li>• Capacidad</li> <li>• Compatibilidad</li> </ul> <p>Lógico CPU:</p> <ul style="list-style-type: none"> <li>• Descripción</li> <li>• Id Físico</li> <li>• Width</li> <li>• Compatibilidad</li> </ul>
<b>Memoria</b>	
<b>HardInfo</b>	<b>Lshw</b>
<ul style="list-style-type: none"> <li>• Evalúa el uso de la memoria.</li> </ul>	<ul style="list-style-type: none"> <li>• Descripción</li> <li>• Id Físico</li> <li>• Slot</li> <li>• Size</li> </ul> <p>Bank:</p> <ul style="list-style-type: none"> <li>• Descripción</li> <li>• Producto</li> <li>• Vendedor</li> <li>• Id Físico</li> <li>• Bus Info</li> <li>• Serie</li> <li>• Slot</li> <li>• Size</li> <li>• Width</li> <li>• Clock</li> </ul>
<b>BIOS</b>	
<b>HardInfo</b>	<b>Lshw</b>
<ul style="list-style-type: none"> <li>• Fecha</li> <li>• Vendedor</li> <li>• Versión</li> </ul>	<ul style="list-style-type: none"> <li>• Descripción</li> <li>• Vendedor</li> <li>• Versión</li> <li>• Fecha</li> <li>• Size</li> <li>• Capacidad</li> <li>• Compatibilidad</li> </ul>

Disco Duro	
HardInfo	Lshw
<ul style="list-style-type: none"> <li>• Modelo</li> <li>• Vendedor</li> <li>• Tipo</li> <li>• Revisión</li> </ul> SCSI Controlador: <ul style="list-style-type: none"> <li>• Controlador</li> <li>• Channel</li> <li>• Id</li> <li>• LUN</li> </ul>	<ul style="list-style-type: none"> <li>• Descripción</li> <li>• Producto</li> <li>• Vendedor</li> <li>• Id Físico</li> <li>• Bus Info</li> <li>• Nombre Lógico</li> <li>• Versión</li> <li>• Serie</li> <li>• Size</li> <li>• Compatibilidad</li> <li>• Configuración</li> </ul>

Tabla 1.3: Hardinfo y Lshw en cuanto a características que obtiene de cada dispositivo.

#### Resultado obtenido de la tabla 1.3:

En la comparación realizada entre la herramienta Hardinfo y el comando Lshw, se obtuvo como resultado que **Lshw**, brinda una mayor información, un contenido más descriptivo, con una mejor estructura lógica y uniforme de la información para cada componente. Se adecúa a las necesidades de la investigación.

### **1.4. Propuesta de solución a las irregularidades que presenta Lshw.**

El comando Lshw para la recopilación de la información del hardware de un equipamiento ofimático, fue el resultado del estudio realizado con el objetivo de obtener la idónea para satisfacer las necesidades descritas y garantizar la compatibilidad. En las tablas comparativas, se pudo apreciar que Lshw a pesar de ser la más óptima, no realiza una captura de todos los dispositivos. Por tanto se decide buscar una solución que minimice la cantidad de dispositivos no detectados.

#### **Dispositivos que no detecta Lshw:**

1. Dispositivos de entrada.
  - Teclado, ratón, escáner y webcam.
2. Dispositivos de salida:
  - Monitor, impresora y fax.

### 1.4.1. Comandos para la obtención de los dispositivos no detectados por Lshw.

Es necesario saber las características del monitor en una herramienta de recopilación de información de hardware. El comando que se utiliza para este trabajo es **xDPYinfo**. Cuando es descrito en la terminal del sistema operativo Nova, el resultado es lanzado en la salida estándar, brindando información de las unidades organizativas de la pantalla actual, como el número de versión X, la resolución y profundidad de color de todas las pantallas actuales y así sucesivamente (21).

En la obtención de los dispositivos periféricos de entrada (teclado, ratón, escáner y webcam) y los de salida como es el caso de la impresora y el fax, se aplica el comando **usb-devices**. Para obtener ayuda sobre el comando en cuestión se utilizó `man` (22), describiendo a **usb-devices** como una secuencia de comandos shell (bash) que se pueden utilizar para mostrar los detalles de los buses USB en el sistema y los dispositivos conectados a ellos.

## 1.5. Metodología para el desarrollo del software.

El desarrollo de software es una tarea difícil de controlar y como resultado a este problema ha surgido una alternativa, las metodologías. Estas imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente.

Una metodología de desarrollo de software es un conjunto de métodos que permiten sistematizar actividades. Estas indican cómo hacer las cosas a través de procedimientos bien descritos (23).

Hoy en día existen numerosas propuestas metodológicas que inciden con distintos énfasis en las diferentes etapas del proceso de desarrollo de software. Para el desarrollo de la presente investigación se propone la metodología **SXP**, compuesta por las mejores prácticas de las metodologías XP y Scrum. Propuesta en el 2008 por la Ingeniera Gladys Marsi Peñalver Romero; aprobada y puesta en práctica en varios de los proyectos que trabajan con software libre y código abierto, en la UCI.

En dicha metodología, Scrum juega un papel fundamental en la parte de la planificación del proyecto, debido a que es una forma de gestionar proyectos de software, no es una metodología de análisis, ni de diseño, sino más bien una metodología de gestión del trabajo. Mientras que de XP se aprovecha su concepción de estar encaminada al desarrollo. Consiste en una programación extrema, cuya peculiaridad es tener como parte del equipo al usuario final (24).

**Fases de la metodología SXP:**

1. **Planificación-Definición:** Se define la concepción inicial del sistema a desarrollar, establece la visión y se realiza el aseguramiento del proyecto.
2. **Desarrollo:** Se realiza la implementación del sistema.
3. **Entrega:** Puesta en marcha.
4. **Mantenimiento:** Soporte para el cliente.

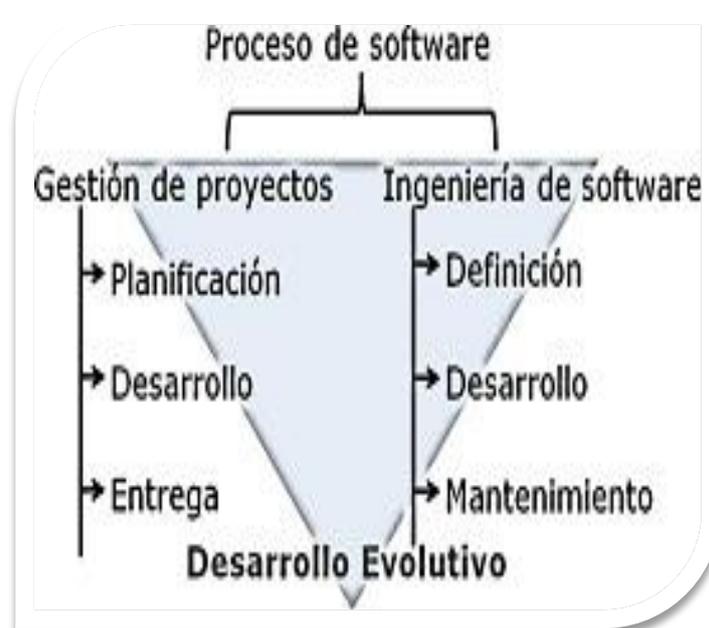


Figura 1.5 Esquema de trabajo SXP

En cada una de estas cuatro fases se realizan numerosas actividades tales como el Levantamiento de Requisitos, la priorización de la Lista de Reserva del Producto (LRP), Definición de las Historias de Usuario, Diseño, Implementación, Pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, permitiendo mejorar el diseño cada vez que se le añada una nueva funcionalidad.

Es una metodología que se caracteriza por ser iterativa e incremental con pequeñas mejoras unas tras otras, basada en Historia de Usuario (HU), está atenta al cambio y permite que el equipo de programación se mantenga en frecuente interacción con el cliente o usuario. Está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos

imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo.

## 1.6. Herramientas, tecnologías y lenguaje a utilizar.

Para el desarrollo de la presente investigación se hace necesario realizar un estudio de herramientas, tecnologías, librerías, y lenguajes de programación para lograr con éxito las tareas trazadas y facilitar el desarrollo de la aplicación, asegurando las condiciones para crear el producto. A continuación se realiza un resumen de las principales características de los mismos.

### 1.6.1. Lenguaje de programación.

#### ➤ Python

Es un lenguaje interpretado de programación, multiplataforma y flexible. Su implementación está bajo la licencia *Python Software Foundation License*, de código abierto. Posee una sintaxis muy clara y legible. Orientado a objetos, permite crear programas con componentes reutilizables. Usa conteo referencial<sup>6</sup> y tipado dinámico<sup>7</sup> para el manejo de memoria (25).

Al ser un lenguaje interpretado, implica el ahorro considerable de tiempo durante el desarrollo de un programa ya que no necesita de compilar ni enlazar. Permite escribir programas muy compactos y legibles. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa (23).

#### Otras características de este lenguaje son:

- Los tipos de datos en Python permiten expresar operaciones más complejas en sentencias cortas.
- El agrupamiento de sentencias se realiza por indentación<sup>8</sup>, en lugar de llaves de principio y fin.
- No se necesita de la declaración de variables.
- Soporta herencia múltiple y polimorfismo.

---

<sup>6</sup> El término conteo referencial se centra en el algoritmo de colección de basura en memoria implementado en Python (26).

<sup>7</sup> Tipado dinámico: una variable puede tomar diferentes tipos de datos durante la ejecución del programa (26).

<sup>8</sup> Indentación: Forma que usa Python para agrupar declaraciones (27).

- Tiene una gran biblioteca estándar, usada para una diversidad de tareas.
- En Python todo es un objeto (incluso las clases).
- El principal objetivo que persigue este lenguaje es la facilidad, tanto de lectura, como de diseño.
- Es un lenguaje potente, seguro, flexible, pero con una gran cantidad de módulos para todas las necesidades que pueden ser útiles a la hora de programar con Python.

### 1.6.2. Herramientas.

Una herramienta informática es una aplicación empleada para la construcción de otros programas o aplicaciones. Diseñadas y fabricadas para cumplir uno o más propósitos, teniendo una función técnica (28). A continuación se describen las herramientas a utilizar en el desarrollo de la presente investigación.

#### ➤ Glade

Es una herramienta RAD<sup>9</sup>, permite el desarrollo rápido y fácil de interfaces de usuario para GTK+ y el entorno de escritorio GNOME. Las interfaces de usuario diseñadas en Glade se guardan como archivos XML, y mediante el uso de GTK + GtkBuilder objetos de estos pueden ser cargados por aplicaciones de forma dinámica según sea necesario. Mediante el uso de GtkBuilder, los archivos XML de Glade se pueden utilizar en numerosos lenguajes de programación como C, C + +, C #, Vala, Java, Perl, Python y otros. Glade es un software libre publicado bajo la licencia GNU GPL (29).

#### ➤ NetBean 6.5

Es un IDE<sup>10</sup> de código abierto que cuenta con una gran comunidad de desarrolladores y colaboradores, creado y patrocinado por Sun Microsystem. El código fuente está disponible para su reutilización de acuerdo con la *Common Development and Distribution License* (CDDL) v1.0 y la *GNU General Public License* (GPL) v2. Es fácil de usar, instalar y puede ser ejecutado en múltiples plataformas como Windows, Linux, Mac OS X y Solaris. Posee un número importante de módulos para extender las funcionalidades del IDE (30).

Entre las principales características que brinda se aprecia el completamiento automático de código python, así como coloreado de código sintáctico y semántico. Permite el depurado del

---

<sup>9</sup> RAD: Desarrollo Rápido de Aplicaciones.

<sup>10</sup> IDE: Entorno Integrado de Desarrollo.

código y resaltado de sintaxis.

### ➤ **Visual Paradigm**

Para modelado, es una herramienta UML<sup>11</sup> profesional que soporta el ciclo de vida completo del desarrollo de software. Es una herramienta *Computer Aided Software Engineering* (CASE) potente y fácil de utilizar, permite el modelado visual UML, propiciando la rápida construcción de aplicaciones de calidad. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es una herramienta colaborativa, pues soporta múltiples usuarios trabajando sobre el mismo proyecto (30).

Entre sus características más importantes sobresalen las siguientes:

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

### ➤ **RapidSVN**

Es un cliente de interfaz gráfica para la comunicación con servidores Subversion. Está escrito en C++ y distribuido bajo licencia GPL. Facilita el versionado de ficheros, desde una interfaz sencilla e intuitiva y se encuentra disponible para plataformas Windows, Linux, MAC OS X y Solaris. Es una herramienta rápida y eficiente (31).

Características de RapidSVN:

- Simple: proporciona una interfaz fácil de usar para las funciones de Subversion.
- Eficiente: sencilla para los principiantes, pero lo suficientemente flexible como para aumentar la productividad de los usuarios de Subversion con experiencia.
- Portable: se ejecuta en cualquier plataforma en la que Subversion y wxWidgets puede ejecutar Linux, Windows, Mac OS / X, Solaris, etc.
- Rápido: completamente escrito en C + +.

---

<sup>11</sup> UML: Lenguaje Unificado de Modelado.

## **1.7. Estudio de los aspectos arquitectónicos para el diseño del sistema.**

El diseño arquitectónico representa la estructura de datos y los componentes del programa que se requieren para construir un sistema basado en computadora. Constituye el estilo arquitectónico que tendrá el sistema, la estructura y las propiedades de los componentes que ese sistema comprende, y las interrelaciones que tienen lugar entre todos los componentes arquitectónicos del sistema (32). Los tres aspectos fundamentales a tener en cuenta en el diseño de la arquitectura de un software son: Estilo Arquitectónico, Patrón Arquitectónico y Patrón de Diseño que representan, de lo general a lo particular, los niveles de abstracción que componen la Arquitectura de Software (Modelo vista Controlador).

Los estilos arquitectónicos según Pressman (32), se enmarcan en cinco categorías generales: Arquitecturas centradas de datos, Arquitecturas de flujo de datos, Arquitecturas de llamada y retorno (permite construir una estructura de programa relativamente fácil de modificar y ajustar a escala), Arquitecturas orientadas a objetos y Arquitecturas estratificadas, que son analizadas generalmente con el fin de combinar varios estilos arquitectónicos para obtener la estructura que mejor se ajusta a los requisitos del cliente y a las normas de calidad. Los estilos fueron formulados por primera vez cuando el escenario tecnológico era sustancialmente distinto del que se manifiesta hoy en día.

El trabajo de diploma “Modelo para la ayuda en la toma de decisiones relativas a estilos y patrones arquitectónicos” centró su investigación en desarrollar un modelo para la toma de decisiones relativas a estilos y patrones arquitectónicos basado en los atributos de calidad definidos para el producto de software a desarrollar, obteniendo dentro de su desarrollo la relación entre estilos y patrones arquitectónicos. Agrupa en la categoría de Arquitecturas de llamada y retorno a las arquitecturas siguientes: Arquitectura Orientada a Objetos, Arquitectura en Capas (Arquitectura en 3 capas), Arquitectura Modelo-Vista-Controlador, Arquitecturas basadas en Componentes (33).

En la Arquitectura en Capas (Arquitectura en 3 capas o N-Layer en inglés), el desarrollo se puede realizar en varias capas. El estilo 3 capas se dividen en las capas de presentación, lógica de negocio y acceso a datos, como se muestra en la figura 1.6. Estas tienen funcionalidades propias que permiten diseñar aplicaciones con un alto nivel de seguridad.

La **capa de usuario** (capa de presentación) muestra el sistema al usuario, captura la información y le comunica la información al usuario. Se comunica únicamente con la capa de negocio.

Por otro lado, la **capa de negocio** (lógica del negocio) recibe las peticiones del usuario. Es donde se establecen las reglas que deben cumplirse, se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados; y con la capa de datos, para almacenar o recuperar datos.

En la **capa de datos** (acceso a datos), residen los datos y es la encargada de acceder a los mismos. Recibe solicitudes ya sean para el almacenamiento o recuperación de información desde la capa de negocio.



Figura 1.6 Diseño Arquitectónico del Sistema.

En la presente investigación se propone utilizar la Arquitectura N Capas y un nivel perteneciente, el estilo arquitectónico y patrón arquitectónico N Capas, específicamente se emplearán 3 Capas para el diseño de la solución. Este estilo pertenece a la categoría de Arquitecturas de llamada y retorno según Pressman (32). También se aplicarán los patrones de diseño Experto en información y Controlador pertenecientes a GRASP<sup>12</sup> (*General Responsibility Assignment Software Patterns*, según sus siglas en inglés). El estilo de código de programación que se utilizará en el desarrollo del software será PEP-8<sup>13</sup> (34).

## 1.8. Protocolo de comunicación entre sistemas.

Se nombra protocolo de comunicación a un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red por medio de intercambio de mensajes. Pueden ser implementados por hardware, software o una combinación de ambos. En la capa de acceso a

<sup>12</sup> PEP-8: Patrón de software para asignación de responsabilidades de uso general.

<sup>13</sup> Python Enhancement Proposal: Propuesta de mejora en Python.

datos para la comunicación con el sistema SICEH se utilizará como protocolo de comunicación SOAP (35).

➤ **SOAP**

Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos. No está asociado con ningún lenguaje, por lo que la implementación de la API se deja al lenguaje de la programación. SOAP no se encuentra frecuentemente asociado a ningún protocolo de transporte, su especificación no describe como deberían asociar los mensajes de SOAP con HTTP. La mayoría de los sistemas de objetos distribuidos se pueden extender, no están atados a ninguna infraestructura. Permite la interoperabilidad entre múltiples entornos, SOAP se desarrolló sobre los estándares existentes de la industria del software, por lo que las aplicaciones que se ejecuten en plataformas con dichos estándares pueden comunicarse mediante mensajes SOAP con aplicaciones que se ejecutan en otras plataformas (36).

## **Conclusiones parciales**

En el capítulo recién concluido se abordaron algunos conceptos relacionados con el objetivo de la investigación ayudando a definir el proceso de diagnóstico de compatibilidad del hardware con el sistema operativo Nova. Se realizó un estudio de las herramientas de recopilación de la información para los sistemas operativos GNU/Linux, una vez identificadas se instalaron en el sistema operativo Nova, seleccionando los comandos Lshw, xdpinfo y usb-devices como apoyo al levantamiento de información del hardware. Esto permitió concluir que es necesario el desarrollo de una herramienta que permita realizar el proceso de Diagnóstico de Compatibilidad de Hardware para el sistema operativo Nova. Se realizó la selección de la metodología que guiará el proceso de desarrollo de la aplicación. Además, se documentaron las tecnologías, herramientas y lenguaje a emplear.

## **Capítulo 2. Definición y desarrollo del Sistema de Diagnóstico de compatibilidad de Hardware.**

En este capítulo se presenta la descripción de las Historias de Usuario identificadas. Se realiza una explicación detallada del proceso de planificación, donde se utilizaron estas para realizar el análisis y la estimación del tiempo de desarrollo de cada una, a través de un proceso iterativo que se ajustó en todo momento a la velocidad de trabajo del equipo de desarrollo

### **2.1. Sistema de diagnóstico de compatibilidad de hardware propuesto.**

Para el desarrollo de SistDCH se tiene en cuenta el proceso de diagnóstico de compatibilidad de hardware para el sistema operativo GNU/Linux Nova descrito en el Capítulo 1, epígrafe 1.1. En el mismo se define el concepto y se describen los pasos fundamentales que rigen el proceso.

En la propuesta de solución al problema el sistema debe cumplir una serie de requisitos. Para el desarrollo se utiliza el lenguaje de programación Python de conjunto con la librería Gtk y Webkit para visualizar la información al usuario e interactuar con él, apoyándose en el entorno integrado de desarrollo Netbeans para realizar la implementación y Glade para la creación de la interfaz visual. Como guía del desarrollo del software se utiliza la Metodología SXP y Visual Paradigm como herramienta de modelado.

Para lograr un mejor diseño de la arquitectura del software se utilizará la arquitectura N Capas y un nivel perteneciente, como estilo arquitectónico y patrón arquitectónico N Capas (N Layer - 3 Capas). Para un mejor diseño de la arquitectura del software se hará uso de los patrones de diseño Experto en información y Controlador pertenecientes a GRASP. El estilo de código de programación que se utilizará en el desarrollo del software será PEP – 8.

El sistema realizará un levantamiento de información del hardware del equipamiento ofimático donde se ejecute, apoyado de los comandos Lshw, xdpinfo y usb-devices para la recopilación de la información, o puede cargar un reporte antes exportado por el propio sistema. Una vez obtenido el mismo se consumirán los servicios que brinda SICEH mediante el protocolo de comunicación SOAP (ver figura 2.1), donde se enviará el reporte para diagnosticar (determinar la categoría de certificación) o para homologar (permite al usuario seleccionar los dispositivos a homologar).

SistDCH le mostrará al usuario el resultado del diagnóstico realizado, encontrándose la información detallada de la funcionalidad de los dispositivos, la categoría de certificación y sus recomendaciones. La homologación del hardware, por lo general se ejecuta cuando un dispositivo

no es compatible. Para homologar se consume otro servicio de SICEH enviándole los dispositivos que el usuario decide. Obtenida la respuesta se muestra en pantalla un listado con los dispositivos que contienen características semejantes y certificados como compatibles. SistDCH permite exportar el levantamiento de la información del hardware del equipamiento ofimático e importarlo nuevamente.

### Servicios a consumir que brinda SICEH:

1. **ImportHardware:** importa el hardware al sistema SICEH y obtiene como respuesta el Id de cada dispositivo.
2. **GetOperativeSystemIdByUid:** se envía a SICEH las características del sistema operativo instalado y devuelve su Id.
3. **GetHardwareCompatibility:** Se envía los Id obtenidos en el servicio 1 y 2, se obtiene el Id de la categoría de certificación de cada uno de los Id enviados.
4. **ListCertificationCategoryForOperativeSystem:** Se envía el resultado del servicio 2 y devuelve todas las categorías de certificación existentes para el sistema operativo.
5. **HomologateHardware:** Se envían los Id obtenidos en el servicio 1 y 2 a SICEH y la cantidad que se desea retornar, se obtiene como respuesta los dispositivos homólogos a los enviados.



Figura 2.1 Servicios de SICEH.

## 2.2. Proceso de desarrollo del proyecto basado en SXP.

Para llevar de forma organizada y efectiva el proceso de desarrollo, se toma como base en la metodología ágil SXP, la creación de artefactos (tabla de roles, historias de usuario, modelos auxiliares) que permiten un desarrollo centrado en subtareas, permitiendo mayor rapidez en el

cumplimiento de las tareas asignadas.

Rol	Responsabilidad	Nombre
<b>Gerente</b>	Dirige y controla las tareas del equipo. Toma las decisiones finales. Participa en la selección de objetivos y requerimientos. Controla el progreso y da seguimiento a cada iteración. Evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes.	Angel Goñi Oramas Eduardo A. Cuesta LLanes
<b>Cliente</b>	Participa en las tareas que involucran la lista de reserva del producto.	Yoandy Pérez Villazón
<b>Programadores</b>	Elabora el código de las nuevas funcionalidades a implementar. Escribe las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y el resto del equipo.	Adrian Cruz Machin Roberto Menéndez Ruiz
<b>Analista</b>	Es el encargado de escribir las historias de usuario y las pruebas funcionales para validar su implementación.	Adrian Cruz Machin
<b>Diseñadores</b>	Encargado del diseño del sistema, de los prototipos de interfaces máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción.	Roberto Menéndez Ruiz
<b>Encargado de pruebas.</b>	Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.	Adrian Cruz Machin Roberto Menéndez Ruiz
<b>Arquitecto</b>	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema. Ayuda en el diseño de las metáforas.	Adrian Cruz Machin Roberto Menéndez Ruiz

Tabla 2.1: Planificación del proyecto por roles.

### 2.3. Modelo de dominio.

Un sistema por pequeño que parezca no lo hace sencillo, por ello se establece una técnica para la especificación de los requisitos más importantes del sistema, que va a dar soporte al negocio, “el modelo del negocio”. En dependencia de la situación o escenario se determina si es necesario un modelo completo del negocio o de lo contrario se procede a definir el modelo conceptual o modelo de dominio (ver figura2.2).

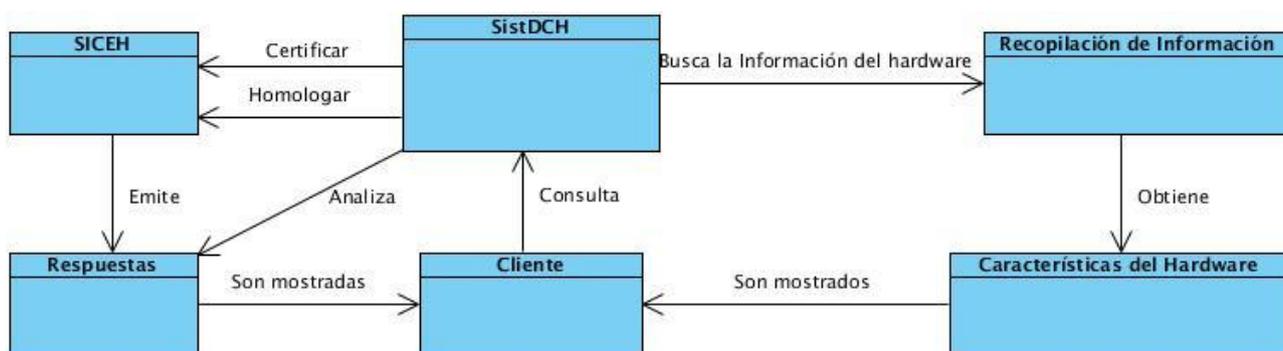


Figura 2.2: Modelo de Dominio de SistDCH.

## 2.4. Lista de Reserva del Producto (LRP).

La Lista de Reserva del Producto está conformada por una lista priorizada de requisitos que define el trabajo que se va a realizar en el proyecto. Determina el orden en que se le irá dando cumplimiento durante las iteraciones a cada requerimiento recogido, según la prioridad establecida en cada uno; abarca las características que el producto debe tener.

En la siguiente tabla 2.2 se recogen en cada una de las filas los requisitos identificados del producto. Se muestra el nombre del desarrollador que dará cumplimiento al requisito, el número identificativo del requisito, la descripción del requisito, el tiempo estimado para su cumplimiento y el nombre del estimador respectivamente.

RF (Requisitos Funcionales)				
Prioridad	Ítem	Descripción	Estimación	Estimado por
<b>Subsistema: SistDCH</b>				
<b>Muy Alta</b>				
Adrian Cruz Machin.	1	Mostrar el grado de compatibilidad del sistema operativo Nova con el hardware mediante el resultado del diagnóstico.	4 horas	Adrian Cruz Machin.
<b>Alta</b>				
Roberto Menéndez Ruíz	2	Realizar levantamiento de hardware.	2.5 horas	Roberto Menéndez Ruíz

Roberto Menéndez Ruíz	3	Enviar la información recolectada en el levantamiento de hardware a SICEH para ser diagnosticada.	3 horas	Roberto Menéndez Ruíz
Adrian Cruz Machin.	4	Enviar la información de los dispositivos que son compatibles a SICEH para buscar un Homólogo.	3 Horas	Adrian Cruz Machin.
<b>Media</b>				
Adrian Cruz Machin.	5	Mostrar la información recolectada en el levantamiento de hardware.	2 horas	Adrian Cruz Machin.
Roberto Menéndez Ruíz	6	Mostrar la información de los dispositivos devueltos después de consumir el servicio para la homologación.	3 horas	Roberto Menéndez Ruíz
<b>Baja</b>				
Adrian Cruz Machin.	7	Guardar el reporte efectuado.	2 horas	Adrian Cruz Machin.
Roberto Menéndez Ruíz	8	Importar reporte efectuado.	2 horas	Roberto Menéndez Ruíz
<b>RNF (Requisitos No Funcionales)</b>				
Diseño	9	Usar como lenguaje de programación python.		
	10	Utilizar GTK y Webkit para la interfaz de usuario.		
	11	Usar protocolo SOAP para la comunicación entre aplicaciones.		
Usabilidad	12	Interfaz amigable al usuario.		
Eficiencia	13	La aplicación debe responder en un tiempo relativamente corto de acuerdo a las condiciones en la que se ejecuta.		

Tabla 2.2: Lista de reserva del producto

## 2.5. Historias de Usuario

A continuación se exponen las Historias de Usuarios correspondientes al sistema a desarrollar, así como las interfaces relacionadas con estas. A pesar de que el cliente tiene un amplio dominio de las características del sistema, es válido destacar que es sólo una planificación inicial la cual puede cambiar para ir adecuándola a sus necesidades y nuevas propuestas.

### 2.5.1. Historia de usuario gestionar levantamiento de la información del hardware.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Gestionar levantamiento de la información del hardware.
Modificación de Historia de Usuario Número: 1	
Usuario: Adrian Cruz Machin.	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta.	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3
Descripción: Consiste en realizar un levantamiento de la información de los dispositivos existentes en la computadora, mostrar, exportar e importar el levantamiento.	
Observaciones: El usuario interactuara con la aplicación mediante una interfaz creada con GTK.	
Prototipo de interface:	
Realizar levantamiento de hardware:	
	
Mostrar levantamiento de hardware:	



**Exportar levantamiento de hardware:**



**Importar levantamiento de hardware:**



**Tareas de ingeniería**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.1	<b>Número Historia de Usuario:</b> HU_1
<b>Nombre Tarea:</b> Diseñar el prototipo de interfaz para la aplicación.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0,1
<b>Fecha Inicio:</b> 6/11/2011	<b>Fecha Fin:</b> 8/11/2011
<b>Programador Responsable:</b> Adrian Cruz Machin.	
<b>Descripción:</b> Se diseña una interfaz que permita visualizar los componentes para interactuar con el sistema y mostrar los resultados.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 1.2	<b>Número Historia de Usuario:</b> HU_1

<b>Nombre Tarea:</b> Realizar levantamiento de la información del hardware.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 8/11/2011	<b>Fecha Fin:</b> 14/11/2011
<b>Programador Responsable:</b> Adrian Cruz Machin.	
<b>Descripción:</b> Utilizar los comandos Lshw, xdpinfo y usb-devices para realizar el levantamiento de la información.	

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1.3	<b>Número Historia de Usuario:</b> HU_1
<b>Nombre Tarea:</b> Mostrar el levantamiento de la información del hardware.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 14/11/2011	<b>Fecha Fin:</b> 18/11/2011
<b>Programador Responsable:</b> Adrian Cruz Machin.	
<b>Descripción:</b> Implementar la funcionalidad que permita mostrar el levantamiento de hardware que se realizan con los comandos Lshw, xdpinfo y usb-devices.	

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1.4	<b>Número Historia de Usuario:</b> HU_1
<b>Nombre Tarea:</b> Clasificar la información de los dispositivos.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 18/11/2011	<b>Fecha Fin:</b> 22/11/2011
<b>Programador Responsable:</b> Adrian Cruz Machin.	
<b>Descripción:</b> Implementar una funcionalidad que permita agrupar los dispositivos teniendo en cuenta su clasificación.	

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1.5	<b>Número Historia de Usuario:</b> HU_1
<b>Nombre Tarea:</b> Exportar reporte de hardware.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 22/11/2011	<b>Fecha Fin:</b> 28/12/2011
<b>Programador Responsable:</b> Roberto Menéndez Ruíz.	
<b>Descripción:</b> Implementar una funcionalidad que permita exportar el reporte de hardware realizado.	

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1.6	<b>Número Historia de Usuario:</b> HU_1

<b>Nombre Tarea:</b> Importar reporte de hardware.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 28/11/2011	<b>Fecha Fin:</b> 5/12/2011
<b>Programador Responsable:</b> Roberto Menéndez Ruíz.	
<b>Descripción:</b> Implementar una funcionalidad que permita importar el reporte de hardware realizado.	

### 2.5.2. Historia de usuario obtener diagnóstico de compatibilidad del sistema.

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre Historia de Usuario:</b> Obtener diagnóstico de compatibilidad del sistema.
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Roberto Menéndez Ruíz.	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Muy Alta.	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Alto.	<b>Puntos Reales:</b> 2
<b>Descripción:</b> Consiste en consumir el servicio 1, 2, 3 y 4 de SICEH para obtener la compatibilidad de los dispositivos con el sistema operativo Nova, mostrando el resultado en la interfaz de usuario de SistDCH.	
<b>Observaciones:</b> El resultado se mostrará después de realizarse la Historia de Usuario número 1.	
<b>Prototipo de interface:</b>	
<b>Realizar diagnóstico de compatibilidad:</b>	
	
<b>Mostrar diagnóstico de compatibilidad:</b>	



### Tareas de ingeniería

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.1	<b>Número Historia de Usuario:</b> HU_2
<b>Nombre Tarea:</b> Consumir servicio 3 de SICEH.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 14/01/2012	<b>Fecha Fin:</b> 19/01/2012
<b>Programador Responsable:</b> Roberto Menéndez Ruíz.	
<b>Descripción:</b> Implementar una funcionalidad que permita conectarse con SICEH para consumir el servicio 3 para obtener la compatibilidad.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.2	<b>Número Historia de Usuario:</b> HU_2
<b>Nombre Tarea:</b> Consumir el servicio 4 para obtener el listado de las categorías para el sistema operativo Nova.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 22/01/2012	<b>Fecha Fin:</b> 29/01/2012
<b>Programador Responsable:</b> Roberto Menéndez Ruíz	
<b>Descripción:</b> Se consumirá el servicio 4 del sistema SICEH enviando el levantamiento de la información del hardware o el reporte.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.4	<b>Número Historia de Usuario:</b> HU_2

<b>Nombre Tarea:</b> Obtener grado de compatibilidad del sistema.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 22/01/2012	<b>Fecha Fin:</b> 29/01/2012
<b>Programador Responsable:</b> Adrian Cruz Machin.	
<b>Descripción:</b> Se consumirán los servicios 1, 2, 3 y 4 del sistema SICEH enviando el levantamiento de la información del hardware o el reporte.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.5	<b>Número Historia de Usuario:</b> HU_2
<b>Nombre Tarea:</b> Mostrar el diagnóstico.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 22/01/2012	<b>Fecha Fin:</b> 5/02/2012
<b>Programador Responsable:</b> Roberto Menéndez Ruíz	
<b>Descripción:</b> Se muestran los dispositivos existentes con su funcionalidad, categoría de certificación y sus recomendaciones.	

### 2.5.3. Historia de usuario buscar homólogos según características específicas del hardware.

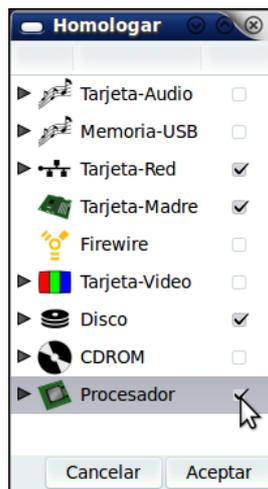
Historia de Usuario	
<b>Número:</b> 3	<b>Nombre Historia de Usuario:</b> Buscar homólogos según características específicas del hardware.
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Roberto Menéndez Ruíz.	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta.	<b>Puntos Estimados:</b> 3
<b>Riesgo en Desarrollo:</b> Alto.	<b>Puntos Reales:</b>
<b>Descripción:</b> Se buscarán equivalentes para los dispositivos que se encuentran instalados en la computadora. Muestra dispositivos con características semejantes.	
<b>Observaciones:</b> Este resultado se mostrará después de realizarse la Historia de Usuario número 1.	

Prototipo de interface:

Realizar homologación:



Seleccionar dispositivos a homologar:



Mostrar homologación:



## Tareas de ingeniería

Tarea de Ingeniería	
<b>Número Tarea:</b> 3.1	<b>Número Historia de Usuario:</b> HU_3
<b>Nombre Tarea:</b> Mostrar al usuario el listado de los dispositivos a homologar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 16/02/2012	<b>Fecha Fin:</b> 23/02/2012
<b>Programador Responsable:</b> Adrian Cruz Machin.	
<b>Descripción:</b> Implementar una funcionalidad que le muestre al usuario la lista de los dispositivos a homologar en una nueva ventana.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 3.2	<b>Número Historia de Usuario:</b> HU_3
<b>Nombre Tarea:</b> Seleccionar los dispositivos a homologar.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 23/02/2012	<b>Fecha Fin:</b> 1/03/2012
<b>Programador Responsable:</b> Adrian Cruz Machin	
<b>Descripción:</b> Implementar una funcionalidad que le permita al usuario seleccionar los dispositivos que desea homologar.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 3.3	<b>Número Historia de Usuario:</b> HU_3
<b>Nombre Tarea:</b> Implementar la funcionalidad que permita homologar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 1/03/2012	<b>Fecha Fin:</b> 8/03/2012
<b>Programador Responsable:</b> Roberto Menéndez Ruíz	
<b>Descripción:</b> Se consumirá el servicio 5 de SICEH enviando los datos de los dispositivos seleccionados por el usuario.	

Tarea de Ingeniería	
<b>Número Tarea:</b> 3.4	<b>Número Historia de Usuario:</b> HU_3
<b>Nombre Tarea:</b> Mostrar resultados de la homologación.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 8/03/2012.	<b>Fecha Fin:</b> 15/03/2012.
<b>Programador Responsable:</b> Roberto Menéndez Ruíz	
<b>Descripción:</b> Implementar una funcionalidad que permita mostrar al usuario el resultado de la homologación realizada.	

## 2.5.4. Historia de usuario imprimir reporte del levantamiento de hardware.

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre Historia de Usuario:</b> Imprimir reporte del levantamiento de hardware.
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Roberto Menéndez Ruíz.	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Baja.	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Implementar una funcionalidad que permita al usuario imprimir el reporte del levantamiento de hardware.	
<b>Observaciones:</b> Este resultado se mostrará después de realizarse la Historia de Usuario número 1.	
<b>Prototipo de interface:</b>	
<b>Realizar impresión:</b>	
 <p>The screenshot shows a Windows application window titled "Diagnóstico y Homologación de Hardware". The menu bar includes "Archivo", "Acciones", "Preferencias", and "Ayuda". The "Acciones" menu is open, displaying options: "Importar" (Ctrl+I), "Exportar" (Ctrl+S), "Imprimir" (Ctrl+P), and "Salir" (Ctrl+Q). A mouse cursor is hovering over the "Imprimir" option. In the background, a network card specification is visible: "Network:0", "Capacidad: 1 GB/s", and "Producto: 82566DC Gigabit Network".</p>	
<b>Imprimir reporte de levantamiento de información del hardware:</b>	



### Tareas de ingeniería

Tarea de Ingeniería	
<b>Número Tarea:</b> 4.1	<b>Número Historia de Usuario:</b> HU_4
<b>Nombre Tarea:</b> Imprimir reporte del levantamiento del hardware.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 15/03/2012.	<b>Fecha Fin:</b> 22/03/2012.
<b>Programador Responsable:</b> Roberto Menéndez Ruíz	
<b>Descripción:</b> Implementar una funcionalidad que permita al usuario imprimir el reporte del levantamiento de hardware.	

### 2.5.5. Historia de usuario imprimir reporte del diagnóstico de hardware.

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre Historia de Usuario:</b> Imprimir reporte del diagnóstico de compatibilidad del hardware.
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Roberto Menéndez Ruíz.	<b>Iteración Asignada:</b> 4

<b>Prioridad en Negocio:</b> Baja.	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Implementar una funcionalidad que permita al usuario imprimir reporte del diagnóstico de hardware.	
<b>Observaciones:</b> Este resultado se mostrará después de realizarse la Historia de Usuario número 2.	
<b>Prototipo de interface:</b>	
<b>Imprimir reporte del diagnóstico de compatibilidad:</b>	
	

### Tareas de ingeniería

Tarea de Ingeniería	
<b>Número Tarea:</b> 5.1	<b>Número Historia de Usuario:</b> HU_5
<b>Nombre Tarea:</b> Imprimir reporte del diagnóstico de hardware.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 22/03/2012.	<b>Fecha Fin:</b> 29/03/2012.
<b>Programador Responsable:</b> Roberto Menéndez Ruíz	
<b>Descripción:</b> Implementar una funcionalidad que permita al usuario imprimir reporte del diagnóstico de hardware.	

## 2.5.6. Historia de usuario imprimir reporte de la homologación de los dispositivos.

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre Historia de Usuario:</b> Imprimir reporte de la homologación de los dispositivos.
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Adrian Cruz Machin.	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Baja.	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Implementar una funcionalidad que permita al usuario imprimir reporte de la homologación de los dispositivos.	
<b>Observaciones:</b> Este resultado se mostrará después de realizarse la Historia de Usuario número 3.	
<b>Prototipo de interface:</b>	
<b>Imprimir reporte de homologación:</b>	
	

### Tareas de ingeniería

Tarea de Ingeniería	
<b>Número Tarea:</b> 6.1	<b>Número Historia de Usuario:</b> HU_6

<b>Nombre Tarea:</b> Imprimir reportes del resultado de la homologación.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 29/03/2012.	<b>Fecha Fin:</b> 5/04/2012.
<b>Programador Responsable:</b> Adrian Cruz Machin.	
<b>Descripción:</b> Implementar una funcionalidad que permita al usuario imprimir reporte del resultado de la homologación.	

## 2.6. Plan de Releases

De acuerdo a las funcionalidades descritas en las Historias de Usuarios y en concordancia con la prioridad asignada para su realización, se planificaron 3 iteraciones que recogen el desarrollo de SistDCH.

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
2	En esta iteración se desarrollarán las historias de usuario que tienen prioridad muy alta.	HU_1, HU_2.	8 semanas
3	En esta iteración se desarrollarán las historias de usuario de alta prioridad y se irán integrando con las ya realizadas.	HU_3	4 semanas
4	En esta iteración se desarrollarán las historias de usuario de baja prioridad y se integrarán con todas las anteriores para conformar SistDCH.	HU_4, HU_5, HU_6	4 semanas

## 2.7. Diseño con metáforas.

Las metáforas conforman el vocabulario para realizar la descripción del problema, conformando el diseño de solución para determinados momentos del proyecto, generando el Modelo de Diseño, que está integrado por un Diagrama de Paquetes. Este diagrama muestra los elementos físicos del sistema así como las relaciones existentes entre ellos. Muestran las dependencias lógicas entre paquetes de software (componentes, código fuente, entre otros).

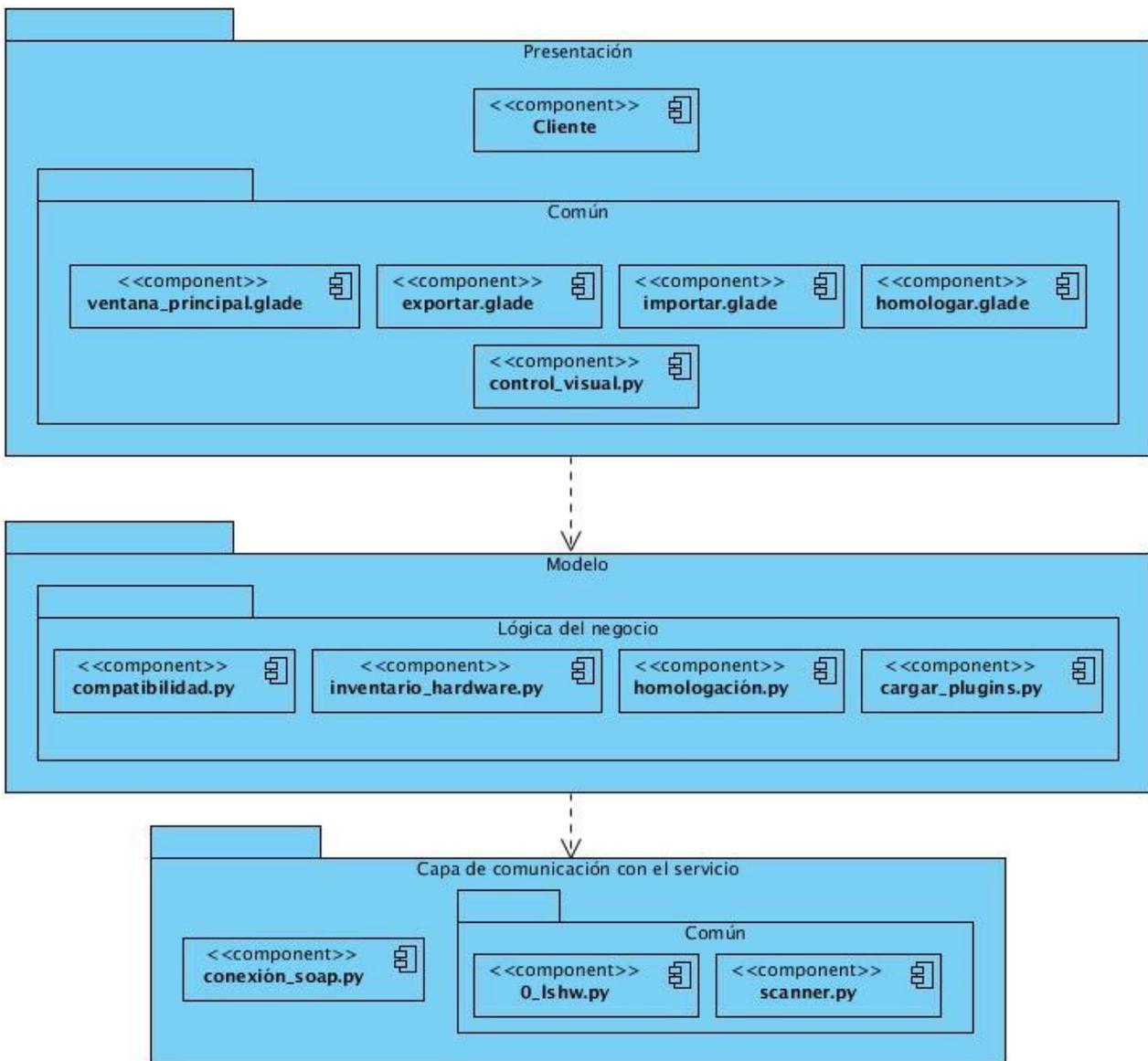


Figura 2.3 Diagrama de paquetes de SistDCH.

➤ **Descripción del diseño:**

En la **capa de presentación** se muestra el componente Cliente representando la interfaz de usuario. Tiene relación con el paquete común integrado por `control_visual.py` encargado de cargar los componentes `ventana_principal.glade` que muestra el acceso a todas las funcionalidades, `exportar.glade` e `importar.glade` permite exportar e importar los resultados y `homologar.glade` como componente auxiliar que permite seleccionar los dispositivos a homologar.

La **capa de modelo** integra todas las clases y funcionalidades relacionadas con el negocio que se encuentran divididas en los paquetes principales que integrarán las funcionalidades de SistDCH.

Esta capa se relaciona con la **capa de comunicación con el servicio**, posibilitando la obtención de la información necesaria.

## 2.8. Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros.

A continuación se presente el diagrama de componentes para SistDCH.

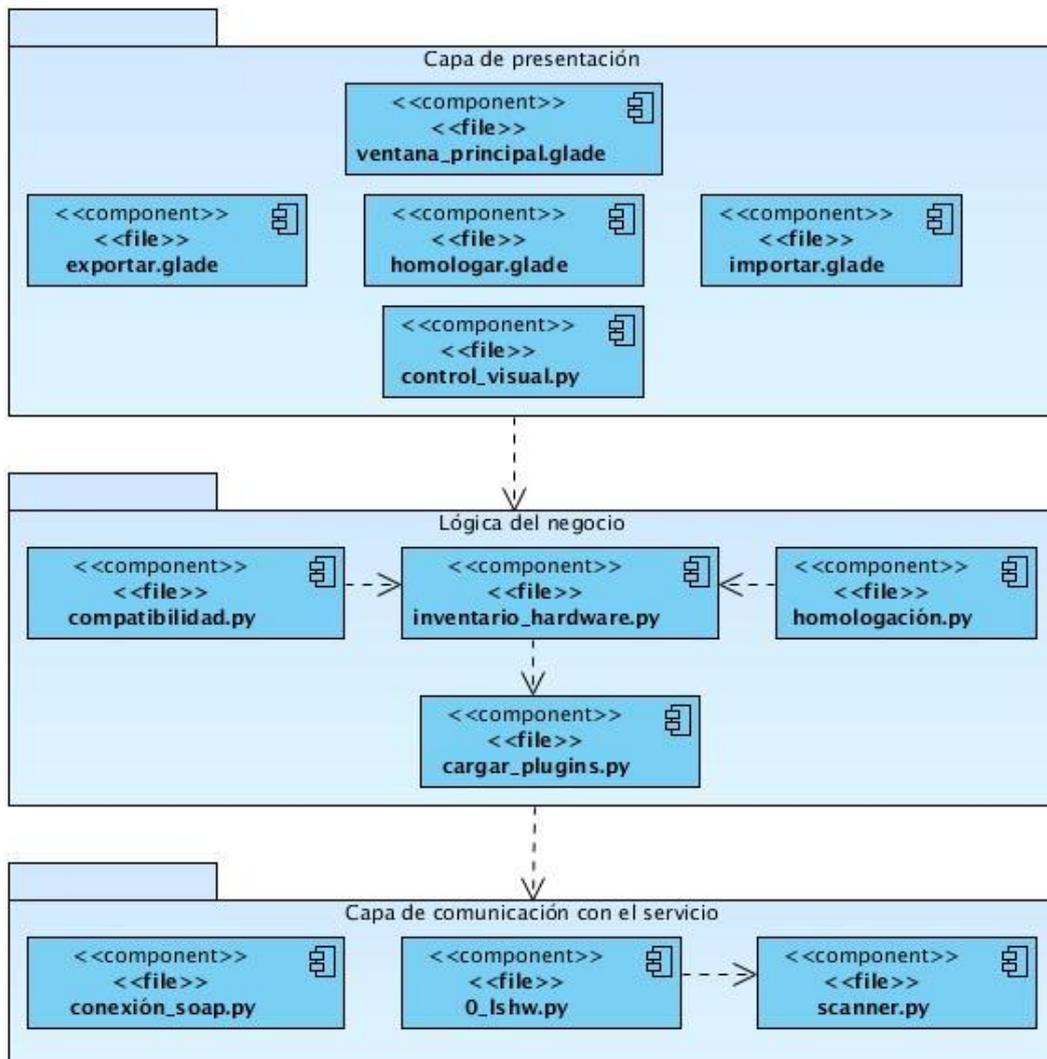


Figura 2.4 Diagrama de Componentes de SistDCH.

➤ **Descripción de las capas de SistDCH:**

La **primera** capa del sistema nombrada Capa de presentación, contiene la interfaz visual, mostrando los diferentes contenidos que gestiona el sistema.

La **segunda** capa nombrada Lógica del negocio contendrá los componentes cargar\_plugins.py encargado de cargar los componentes de la capa inferior para obtener la información referente a los dispositivos, inventario\_hardware.py contiene toda la información recopilada por el componente anterior. Compatibilidad.py depende de inventario\_hardware.py para gestionar el diagnóstico de hardware, al igual que homologación.py para obtener el resultado de la homologación.

En la **tercera** capa nombrada Capa de comunicación con el servicio, contendrá los componentes conexion\_soap.py encargado de realizar la conexión para consumir los servicios mediante el protocolo SOAP y 0\_lshw.py realiza el levantamiento de hardware, auxiliándose de scanner.py para hacer un filtrado de la información recopilada de cada dispositivo.

## **Conclusiones parciales**

En este capítulo se realizó la descripción de la propuesta de solución como factor fundamental para la obtención de un producto que cumpla las expectativas de clientes y desarrolladores. Se capturaron los requisitos del software definidos por el cliente. Además se realizó toda la planificación y el diseño del software; y en la misma medida se fue desarrollando el producto.

## Capítulo 3: Diseño y realización de pruebas a SistDCH.

Una etapa fundamental en el ciclo de desarrollo del software es la etapa de pruebas. Proceso que permite verificar y revelar la calidad del producto. En este capítulo se realiza un diagnóstico y compatibilidad de hardware de una computadora con el sistema operativo Nova, haciendo uso de la aplicación desarrollada en la presente investigación. Se aplican las pruebas a cada una de las historias de usuario correspondientes a cada iteración definida. Tienen como objetivo validar que los casos de prueba de aceptación al sistema cumplan con el funcionamiento esperado, permitiendo al cliente determinar su aceptación, desde el punto de vista de su funcionalidad y rendimiento. De igual forma se muestran las funcionalidades que hasta la fecha implementa el sistema.

### 3.1. Aplicación del SistDCH en un escenario real

A continuación se muestran los pasos necesarios para realizar un diagnóstico y determinar la compatibilidad del sistema operativo Nova con los dispositivos existentes en la computadora, con el objetivo de mostrar la facilidad y agilidad a la hora de realizar este proceso con la aplicación SistDCH desarrollada. Para ello se da solución a una problemática planteada en una empresa en proceso de migración.

Siguiendo los pasos de la Metodología de Migración, una de las tareas en el flujo de Evaluación que tiene mayor peso en la etapa de Preparación es el análisis del estado actual de los equipamientos informáticos de la institución en cuanto al hardware, con el propósito de garantizar su grado de compatibilidad con el sistema operativo Nova y lograr que la migración fluya sin contratiempos en etapas posteriores, donde se lleva a cabo la instalación del sistema.

Una vez obtenido el inventario del software instalado en la computadora mediante la herramienta GITI<sup>14</sup> se utiliza la aplicación SistDCH para realizar el levantamiento del hardware, diagnóstico y homologación. En el repositorio de Nova se encuentra disponible la aplicación SistDHC, esta se instala con privilegios de súper usuario mediante el comando *apt-get install sistdch*.

#### ➤ Pasos:

Primeramente el usuario debe realizar el levantamiento de hardware, seleccionando el elemento de menú Acciones → Escanear. Esta funcionalidad brinda una información detallada de los dispositivos existentes en la computadora, conociendo de cada uno el nombre, modelo o versión,

---

<sup>14</sup> Personalización de OCSInventory, realizada en el proyecto Plataforma de Migración del departamento Migración y Soporte del Centro CESOL (17).

producto, entre otros, tal como se muestra en la figura 3.1.

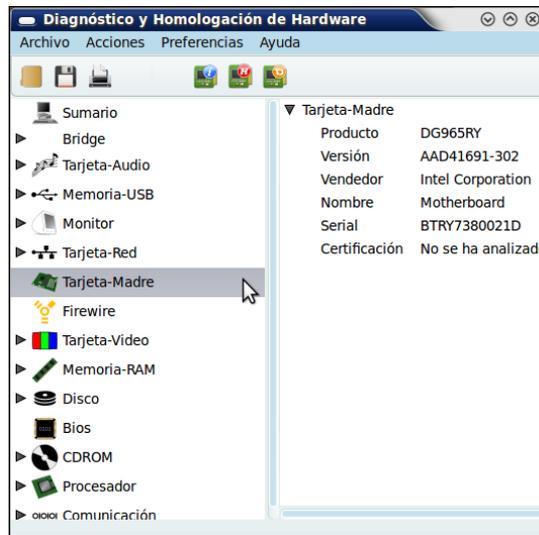


Figura 3.1 Levantamiento de la información.

Seguidamente el usuario puede conocer si los dispositivos existentes son compatibles con el sistema operativo Nova, seleccionando la opción Acciones → Diagnosticar que se encuentra en el menú o realizar una selección múltiple y dirigirse al icono  en la barra de acceso rápido. El sistema entonces muestra de cada uno de los dispositivos una categoría de compatibilidad con el sistema operativo: compatible, parcialmente compatible o incompatible. Si el mismo no se encuentra certificado en SICEH el usuario obtiene como respuesta “Dispositivo no certificado” (ver figura 3.2)



Figura 3.2 Resultado del diagnóstico de compatibilidad.

El usuario para obtener dispositivos que sean compatibles con el sistema operativo Nova, con características semejantes a los ya existentes se dirige al elemento de menú Acciones →

Homologar, desplegando una ventana auxiliar que le permite seleccionar los dispositivos indistintamente de su grado de compatibilidad (ver figura 3.3). Una vez seleccionados se da clic en el botón inferior Aceptar, ofreciendo un listado con todos los homólogos existentes en SICEH para cada dispositivo (ver figura 3.4).

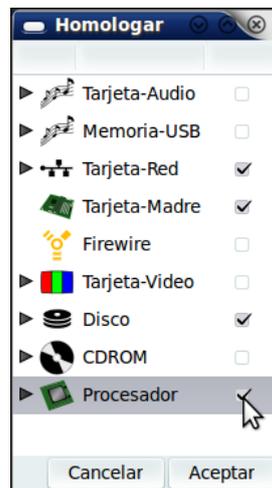


Figura 3.3 Seleccionar dispositivos a homologar.



Figura 3.4 Resultado de la homologación.

Si el usuario desea guardar el reporte efectuado para consultarlo posteriormente sin necesidad de ejecutar nuevamente la aplicación, puede dirigirse al elemento de menú Archivo → Imprimir o al icono de acceso rápido . Esta opción permite imprimir el levantamiento de la información del

hardware una vez escaneado la computadora (ver figura 3.5), el resultado del diagnóstico de compatibilidad para cada dispositivo (ver figura 3.6) y como se mostró anteriormente en la figura 3.4, el resultado de la homologación.



Figura 3.5 Imprimir el levantamiento de la información del hardware.



Figura 3.6 Imprimir el levantamiento de la información del hardware con el diagnóstico de compatibilidad realizado.

Una vez concluido el proceso de diagnóstico de compatibilidad de hardware haciendo uso de la aplicación SistDCH se evidencia que para obtener las respuestas se necesita solamente un usuario con dominio en el uso de la aplicación, sin tener que realizar una búsqueda exhaustiva en internet o utilizar SICEH como se hacía anteriormente, lo que reduce en gran medida el número de recursos y esfuerzos humanos. Se tiene a disposición del mismo una única aplicación, amigable y fácil de usar para obtener información indispensable para llevar a cabo un proceso de migración exitoso.

Por todo lo expuesto anteriormente se evidencia que el uso de esta aplicación contribuye a aumentar la eficiencia con que se realiza el proceso de diagnóstico de compatibilidad de hardware con el sistema operativo Nova, siendo anteriormente complejo, al no contar el usuario con una herramienta capaz de realizar el proceso de forma eficiente desde su estación de trabajo.

### 3.2. Casos de prueba.

Las pruebas de aceptación fueron definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final correspondieron al cliente. La utilización de estas, proporcionaron grandes ventajas, permitiendo a los programadores principalmente medir la calidad de su trabajo y garantizar la entrega de un producto con calidad y en correspondencia con las necesidades del cliente. Se definieron casos de prueba para todas las historias de usuario, a continuación se dan a conocer las pruebas que se realizaron a cada una de las historias de usuario con las que cuenta SistDCH para el diagnóstico, compatibilidad y homologación del hardware.

➤ **Casos de prueba para HU gestionar levantamiento de la información del hardware.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> SistDCH-1-1	<b>Nombre Historia de Usuario:</b> Gestionar levantamiento de la información del hardware.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	
<b>Descripción de la Prueba:</b> Esta prueba consiste en realizar el levantamiento de la información del hardware sin permisos de administración.	
<b>Condiciones de Ejecución:</b> Debe estar instalado un sistema operativo GNU/Linux Nova. Instalar en el sistema operativo la herramienta Lshw y los comandos xdpinfo y usb-device.	
<b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar la opción de Escanear para obtener las características de los dispositivos en el equipamiento ofimático.	

<p>La selección se puede realizar de dos variantes:</p> <ol style="list-style-type: none"> <li>1. Una vez en la aplicación SistDCH dirigirse al menú (Acciones→ Escanear).</li> <li>2. Seleccionar en la barra de herramienta de acceso rápido el icono .</li> </ol>
<p><b>Resultado esperado:</b> El sistema obtuvo todos los componentes del equipamiento ofimático con sus características en las dos variantes probadas.</p>
<p><b>Evaluación de la prueba:</b> Satisfactoria.</p>

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b> SistDCH-1-2</p>	<p><b>Nombre Historia de Usuario:</b> Gestionar levantamiento de la información del hardware.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.</p>	
<p><b>Descripción de la Prueba:</b> Esta prueba consiste en realizar el levantamiento de la información del hardware con permisos de administración.</p>	
<p><b>Condiciones de Ejecución:</b> Debe estar instalado un sistema operativo GNU/Linux Nova. Instalar en el sistema operativo la herramienta Lshw y los comandos xdpinfo y usb-device.</p>	
<p><b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar la opción de Escanear para obtener las características de los dispositivos en el equipamiento ofimático.</p> <p>La selección se puede realizar de dos variantes:</p> <ol style="list-style-type: none"> <li>1. Una vez en la aplicación SistDCH dirigirse al menú (Acciones→ Escanear).</li> <li>2. Seleccionar en la barra de herramienta de acceso rápido el icono .</li> </ol>	
<p><b>Resultado esperado:</b> El sistema obtuvo todos los componentes del equipamiento ofimático con sus características en las dos variantes probadas.</p>	
<p><b>Evaluación de la prueba:</b> Satisfactoria.</p>	

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b> SistDCH-1-3</p>	<p><b>Nombre Historia de Usuario:</b> Gestionar levantamiento de la información del hardware.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.</p>	
<p><b>Descripción de la Prueba:</b> Esta prueba consiste en listar todos los componentes del equipamiento ofimático con sus características.</p>	
<p><b>Condiciones de Ejecución:</b> Debe estar instalado un sistema operativo GNU/Linux Nova.</p>	

Instalar en el sistema operativo la herramienta Lshw y los comandos xdpinfo y usb-device.
<p><b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar la opción de Escanear para obtener las características de los dispositivos en el equipamiento ofimático.</p> <p>La selección se puede realizar de dos variantes:</p> <ol style="list-style-type: none"> <li>1. Una vez en la aplicación SistDCH dirigirse al menú (Acciones→ Escanear).</li> <li>2. Seleccionar en la barra de herramienta de acceso rápido el icono .</li> </ol>
<b>Resultado esperado:</b> El sistema lista todos los componentes del equipamiento ofimático con sus características en la interfaz SistDCH.
<b>Evaluación de la prueba:</b> Satisfactoria.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> SistDCH-1-4	<b>Nombre Historia de Usuario:</b> Gestionar levantamiento de la información del hardware.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	
<b>Descripción de la Prueba:</b> Esta prueba consiste en exportar el reporte del hardware con toda la información de los componentes en formato .xml.	
<b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear o Importar.	
<p><b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar la opción de Exportar para guardar el archivo.xml con la información recolectada por el sistema.</p> <p>La selección se puede realizar de dos variantes:</p> <ol style="list-style-type: none"> <li>1. Una vez en la aplicación SistDCH dirigirse al menú (Archivo→ Exportar).</li> <li>2. Seleccionar en la barra de herramienta de acceso rápido el icono .</li> </ol>	
<b>Resultado esperado:</b> El sistema exporto la información de los dispositivos en formato .xml.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> SistDCH-1-5	<b>Nombre Historia de Usuario:</b> Gestionar levantamiento de la información del hardware.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	

<b>Descripción de la Prueba:</b> Esta prueba consiste en importar el reporte del hardware con toda la información de los componentes en formato .xml.
<b>Condiciones de Ejecución:</b> El reporte debió ser generado anteriormente por la aplicación SistDCH.
<b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar la opción de Importar para guardar el archivo.xml con la información recolectada por el sistema.  La selección se puede realizar de dos variantes: 1. Una vez en la aplicación SistDCH dirigirse al menú (Archivo→ Importar). 2. Seleccionar en la barra de herramienta de acceso rápido el icono  .
<b>Resultado esperado:</b> El sistema carga la información del .xml y la muestra.
<b>Evaluación de la prueba:</b> Satisfactoria.

➤ **Casos de prueba para HU Obtener diagnóstico de compatibilidad de hardware.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> SistDCH-2-1	<b>Nombre Historia de Usuario:</b> Obtener diagnóstico de compatibilidad de hardware.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	
<b>Descripción de la Prueba:</b> Esta prueba consiste en consumir los servicios necesarios de SICEH para obtener la (s) categoría(s) de certificación.	
<b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado al menos un dispositivo.	
<b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar la opción de Diagnosticar para obtener la categoría de certificación.  La selección se puede realizar de dos variantes: 1. Una vez en la aplicación SistDCH dirigirse al menú (Acciones→ Diagnosticar) para obtener un resultado de todos los dispositivos existentes en el equipamiento ofimático. 2. Marcar el dispositivo que se desea diagnosticar y seleccionar en la barra de herramienta de acceso rápido el icono  , para obtener el resultado de un dispositivo en específico.	
<b>Resultado esperado:</b> El sistema obtuvo la categoría de los dispositivos que estaban certificados en el sistema de SICEH.	

**Evaluación de la prueba:** Satisfactoria.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> SistDCH-2-2	<b>Nombre Historia de Usuario:</b> Obtener diagnóstico de compatibilidad de hardware.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	
<b>Descripción de la Prueba:</b> Esta prueba consiste en consumir los servicios necesarios de SICEH para mostrar el resultado del diagnóstico.	
<b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado al menos un dispositivo.	
<b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar uno o más dispositivos en el sistema y seleccionar la opción de Diagnosticar para obtener el resultado del diagnóstico.  La selección se puede realizar de dos variantes: 1. Una vez en la aplicación SistDCH dirigirse al menú (Acciones→ Diagnosticar) para obtener un resultado de todos los dispositivos existentes en el equipamiento ofimático. 2. Marcar el dispositivo que se desea diagnosticar y seleccionar en la barra de herramienta de acceso rápido el icono  , para obtener el resultado de un dispositivo en específico.	
<b>Resultado esperado:</b> Se muestra en el sistema el resultado del diagnóstico en las dos variantes.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

- **Casos de prueba para HU buscar homólogos según características específicas del hardware.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> SistDCH-3-1	<b>Nombre Historia de Usuario:</b> Buscar homólogos según características específicas del hardware.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	
<b>Descripción de la Prueba:</b> Esta prueba consiste en mostrar el listado de los dispositivos a homologar en una ventana auxiliar para permitirle su selección.	
<b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado al menos un dispositivo.	
<b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar la opción de Homologar para	

<p>mostrar la ventana auxiliar.</p> <p>Una vez en la aplicación SistDCH dirigirse al menú (Acciones→ Homologar) para obtener un resultado de todos los dispositivos existentes en el equipamiento ofimático.</p>
<p><b>Resultado esperado:</b> Se muestran los dispositivos en la ventana auxiliar para permitir su selección.</p>
<p><b>Evaluación de la prueba:</b> Satisfactoria.</p>

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b> SistDCH-3-2</p>	<p><b>Nombre Historia de Usuario:</b> Buscar homólogos según características específicas del hardware.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.</p>	
<p><b>Descripción de la Prueba:</b> Esta prueba consiste en consumir el servicio 5 de SICEH para obtener los homólogos de un dispositivo.</p>	
<p><b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado al menos un dispositivo.</p>	
<p><b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar uno o más dispositivos en el sistema y seleccionar la opción de Homologar.</p> <p>Una vez en la aplicación SistDCH marcar el o los dispositivos que se desean homologar y seleccionar en la barra de herramienta de acceso rápido el icono  .</p>	
<p><b>Resultado esperado:</b> El sistema obtuvo el listado de los dispositivos que se desean homologar.</p>	
<p><b>Evaluación de la prueba:</b> Satisfactoria.</p>	

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b> SistDCH-3-3</p>	<p><b>Nombre Historia de Usuario:</b> Buscar homólogos según características específicas del hardware.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.</p>	
<p><b>Descripción de la Prueba:</b> Esta prueba consiste en consumir el servicio 5 de SICEH para mostrar los homólogos de un dispositivo.</p>	
<p><b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado al menos un dispositivo.</p>	
<p><b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar uno o más dispositivos en el sistema y seleccionar la opción de Homologar.</p> <p>Una vez en la aplicación SistDCH marcar el o los dispositivos que se desean homologar y</p>	

seleccionar en la barra de herramienta de acceso rápido el icono  .
<b>Resultado esperado:</b> El sistema muestra un webkit con el listado de los dispositivos enviados y se encuentran registrados con homólogos en SICEH.
<b>Evaluación de la prueba:</b> Satisfactoria.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> SistDCH-3-4	<b>Nombre Historia de Usuario:</b> Buscar homólogos según características específicas del hardware.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	
<b>Descripción de la Prueba:</b> Esta prueba consiste en consumir el servicio 5 de SICEH para mostrar los homólogos de un dispositivo.	
<b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado al menos un dispositivo.	
<b>Entrada /Pasos de ejecución:</b> La entrada consiste en seleccionar en la ventana auxiliar los dispositivos a Homologar. Una vez en la ventana auxiliar dirigirse a los dispositivos listados y seleccionar los que se deseen homologar <input checked="" type="checkbox"/> , dar clic en (Aceptar) para obtener el resultado de la homologación.	
<b>Resultado esperado:</b> El sistema muestra un webkit con el listado de los dispositivos homólogos en SICEH.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

➤ Casos de prueba para HU imprimir reporte del levantamiento del hardware.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> SistDCH-4-1	<b>Nombre Historia de Usuario:</b> Imprimir reporte del levantamiento de hardware.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	
<b>Descripción de la Prueba:</b> Esta prueba consiste en permitir imprimir el reporte del levantamiento de hardware.	
<b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado al menos un dispositivo.	
<b>Entrada /Pasos de ejecución</b> La entrada consiste en seleccionar la opción de Imprimir para	

<p>obtener el reporte del levantamiento de hardware. La selección se puede realizar de dos variantes:</p> <ol style="list-style-type: none"> <li>1. Una vez en la aplicación SistDCH dirigirse al menú (Archivo→ Imprimir).</li> <li>2. Seleccionar en la barra de herramienta de acceso rápido el icono .</li> </ol>
<p><b>Resultado esperado:</b> Se muestra mediante webkit el resultado del levantamiento y la opción de impresión.</p>
<p><b>Evaluación de la prueba:</b> Satisfactoria.</p>

➤ **Casos de prueba para HU imprimir reporte del diagnóstico de hardware.**

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b> SistDCH-5-1</p>	<p><b>Nombre Historia de Usuario:</b> Imprimir reporte del diagnóstico de hardware.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.</p>	
<p><b>Descripción de la Prueba:</b> Esta prueba consiste en permitir imprimir el reporte del diagnóstico de hardware como una más de las características del levantamiento.</p>	
<p><b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y Diagnosticar.</p>	
<p><b>Entrada /Pasos de ejecución</b> La entrada consiste en seleccionar la opción de imprimir el reporte del diagnóstico de hardware como una más de las características del levantamiento. La selección se puede realizar de dos variantes:</p> <ol style="list-style-type: none"> <li>1. Una vez en la aplicación SistDCH dirigirse al menú (Archivo→ Imprimir).</li> <li>2. Seleccionar en la barra de herramienta de acceso rápido el icono .</li> </ol>	
<p><b>Resultado esperado:</b> Se muestra mediante webkit el resultado del diagnóstico de hardware como una más de las características del levantamiento.</p>	
<p><b>Evaluación de la prueba:</b> Satisfactoria.</p>	

➤ **Casos de prueba para HU imprimir reporte de la homologación de los dispositivos.**

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b> SistDCH-6-1</p>	<p><b>Nombre Historia de Usuario:</b> Imprimir reporte de la homologación de los dispositivos.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.</p>	
<p><b>Descripción de la Prueba:</b> Esta prueba consiste en permitir imprimir el reporte de la</p>	

homologación de los dispositivos enviados a SICEH por la barra de herramientas.
<b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado al menos un dispositivo a homologar.
<b>Entrada /Pasos de ejecución</b> La entrada consiste en seleccionar en la aplicación al menos un dispositivo a homologar. 1. Una vez en la aplicación SistDCH dirigirse a la barra de herramientas de acceso rápido, en el icono  . 2. Se muestra el resultado en un webkit, dando la opción de Imprimir en la parte superior con el icono  .
<b>Resultado esperado:</b> Se muestra mediante webkit el resultado de la homologación y la opción de impresión
<b>Evaluación de la prueba:</b> Satisfactoria.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> SistDCH-6-2	<b>Nombre Historia de Usuario:</b> Imprimir reporte de la homologación de los dispositivos.
<b>Nombre de la persona que realiza la prueba:</b> Adrian Cruz Machin.	
<b>Descripción de la Prueba:</b> Esta prueba consiste en permitir imprimir el reporte de la homologación de los dispositivos enviados a SICEH mediante una ventana auxiliar.	
<b>Condiciones de Ejecución:</b> Haber ejecutado anteriormente en la aplicación la opción de Escanear y seleccionado en la ventana auxiliar al menos un dispositivo.	
<b>Entrada /Pasos de ejecución</b> La entrada consiste en seleccionar la opción de Homologar en la barra de menú, obteniendo la ventana auxiliar Homologar que permitirá seleccionar nuevos dispositivos. 1. Una vez en la aplicación SistDCH dirigirse al menú (Acciones→ Homologar). 2. Mostrada la ventana auxiliar seleccionar los dispositivos <input checked="" type="checkbox"/> y dar la opción de (Aceptar) al final de la selección. 3. Se muestra el resultado de la homologación en un webkit, dando la opción de imprimir en la parte superior con el icono  .	
<b>Resultado esperado:</b> Se muestra mediante webkit el resultado de la homologación y la opción de impresión.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

### 3.3. Resultados

- Se obtiene una información detallada de los dispositivos existentes en la computadora.
- Permite importar un archivo generado por el propio sistema SistDCH.
- Brinda un diagnóstico por selección múltiple o de todos los dispositivos existentes según lo desee el usuario.
- Permite homologar los dispositivos compatibles y no compatibles para obtener dispositivos con características semejantes a los existentes y compatibles con el sistema operativo Nova.
- Permite exportar en formato PDF los reportes obtenidos mediante el levantamiento de la información del hardware, diagnóstico u homologación.
- Permite exportar a un archivo XML el resultado del levantamiento de la información del hardware y el diagnóstico.
- Permite imprimir el levantamiento de la información del hardware, diagnóstico u homologación.

### Conclusiones parciales

En este capítulo se mostraron los pasos para llevar a cabo el proceso de Diagnóstico de Compatibilidad de Hardware para el sistema operativo Nova en un escenario real, utilizando la aplicación desarrollada; donde se pusieron en práctica cada una de las funcionalidades implementadas, demostrando de esta forma la agilidad y facilidad con que se realiza este proceso. Además se describieron las pruebas que corroboran la calidad del producto; y como resultado de estas se muestran un conjunto de funcionalidades validadas que garantizan la entrega de un producto que responde a las necesidades del cliente.

## Conclusiones

Como resultado de los temas abordados en esta investigación, se obtuvieron las siguientes conclusiones, por cada una de las tareas de investigación:

1. El diagnóstico de compatibilidad del hardware con el SO GNU/Linux Nova es un proceso definido, que requiere la ejecución de tres pasos fundamentales.
2. Para realizar el levantamiento de hardware es necesario vincular los comandos Lshw, xdpinfo y usb-devices.
3. La implementación de las funcionalidades correspondientes a las seis historias de usuario cubren la totalidad de requisitos definidos.
4. El uso del SistDCH contribuye a aumentar la eficiencia en el proceso de diagnóstico de compatibilidad de hardware para el sistema operativo GNU/Linux Nova.
5. SistDCH aporta beneficios a los usuarios en proceso de migración, pues permite realizar un levantamiento del hardware de su estación de trabajo, diagnosticar los dispositivos para identificar la categoría de certificación y encontrar dispositivos con características semejantes y compatibles con el sistema operativo GNU/Linux Nova.

## **Recomendaciones**

Como parte de esta investigación, se definen las siguientes recomendaciones:

1. Brindar el diagnóstico de los dispositivos no certificados en SICEH para que este sirva de base de conocimiento al sistema SICEH.
2. Mejorar el rendimiento de la aplicación utilizando la programación multi-hilo en el consumo de servicios y el levantamiento de la información del hardware.
3. Adaptar el sistema para que pueda ser aplicado en otros sistemas operativos como Windows y GNU/Linux.

## Referencias bibliográficas

1. **Alex Fernández y Blasco Gómez.** <http://www.arqueolab.org/el-nacimiento-de-la-tecnologia>. [En línea] Arqueolab. [Citado el: 23 de octubre de 2011.] <http://www.arqueolab.org/el-nacimiento-de-la-tecnologia>.
2. **MinRex.** Sitio del Ministerio de Relaciones Exteriores de Cuba. [En línea] Cuba, 2005. [Citado el: 28 de octubre de 2011.] [www.cubaminrex.cu/sociedad\\_informacion/cuba\\_si/Masivo\\_TIC.htm](http://www.cubaminrex.cu/sociedad_informacion/cuba_si/Masivo_TIC.htm).
3. **Fuentes, Allan Pierra.** *Conceptualización y Reestructuración Estratégica de la Distribución Cubana de GNU/Linux "Nova"*. La Habana, Cuba : s.n., octubre, 2011.
4. **Hernández, Lisandra Cala.** *Propuesta de integración y nuevas herramientas para el desarrollo de la Plataforma Cubana de Migración a Software Libre.* . La Habana, Cuba : s.n., 2010.
5. **Salmón, Ramón Paumier.** *Metodología para la migración a software libre de la Universidad de las Ciencias Informáticas.* La Habana, Cuba : s.n., 2007.
6. **Rosario, Jimmy.** Observatorio para la CiberSociedad 2012. *La Tecnología de la Información y la Comunicación (TIC)*. [En línea] 2012. [Citado el: 28 de octubre de 2011.] <http://www.cibersociedad.net/archivo/articulo.php?art=218>.
7. **Troyano.** Troyano. [En línea] 24 de agosto de 2009. [Citado el: 28 de octubre de 2011.] <http://www.estrellaroja.info/?p=209>.
8. **GNU.** Categorías de software libre y no libre. [En línea] 30 de 12 de 2011. [Citado el: 29 de octubre de 2011.] <http://www.gnu.org/philosophy/categories.es.html>.
9. **Villazón, Yoandy Pérez.** *Metodología para la migración a Software Libre de las universidades del Ministerio Superior (MES)*. La Habana, Cuba : s.n., 2008.
10. **Dictionary.com.** Dictionary.com. [En línea] 2012. [Citado el: 1 de noviembre de 2011.] <http://dictionary.reference.com/browse/process>.
11. **Definición ABC.** [En línea] 2012. [Citado el: 1 de noviembre de 2011.] <http://www.definicionabc.com/general/diagnostico.php>.
12. **The Free Dictionary.** [En línea] 2012. [Citado el: 3 de noviembre de 2011.] <http://es.thefreedictionary.com/compatibilidad>.
13. **Carbajo, Ricardo R.** *Sistema de Certificación y Homologación de Hardware.* La Habana, Cuba : s.n., 2012.
14. **Pereira, Leandro.** HardInfo. [En línea] 2009. [Citado el: 10 de noviembre de 2011.] <http://www.guia-ubuntu.org/index.php?title=Hardinfo>.
15. **Packages Debian.** [En línea] 2011. [Citado el: 10 de noviembre de 2011.] <http://packages.debian.org/es/squeeze/gnome-device-manager>.
16. **Bright Hub.** [En línea] 2011. [Citado el: 11 de noviembre de 2011.] <http://www.brighthub.com/computing/linux/articles/41828.aspx..>

17. **Haas, Juergen.** About.com. [En línea] 2011. [Citado el: 13 de noviembre de 2011.] [http://linux.about.com/library/cmd/blcmdl2\\_sysinfo.htm](http://linux.about.com/library/cmd/blcmdl2_sysinfo.htm), s.l. : 2011.
18. **Linux Amigos.** [En línea] 2009. [Citado el: 15 de noviembre de 2011.] <http://linuxandfriends.com/2009/02/23/lshw-command-list-hardware-information-in-linux>.
19. **SOFTPEDIA.** [En línea] 27 de julio de 2011. [Citado el: 19 de noviembre de 2011.] <http://www.softpedia.com/get/Others/Finances-Business/OCS-Inventory-NG-Agent.shtml>.
20. **Diana Lilian Antúnez Ginarte y Mario Lázaro García.** *Personalización de la herramienta de gestión de inventarios OCS Inventory NG.* La Habana, Cuba : s.n., mayo, 2011.
21. **ITRestauracion.** [En línea] 27 de diciembre de 2010. [Citado el: 24 de noviembre de 2011.] <http://www.itrestauración.com.ar/?p=1238>.
22. **González, Camilo Denis.** doutesdeslibertas. [En línea] 21 de febrero de 2011. [Citado el: 5 de diciembre de 2011.] <http://linux.die.net/man/1/usb-devices>.
23. **Eduardo Alejandro Cuesta Llanes y Yurenia Blanco Hernández.** *MED: Módulo para la edición y depuración de código Octave integrado al ambiente EIDMAT.* La habana, Cuba : s.n., 2010.
24. **Romero, Gladys Marsi Peñalver.** *Metodología SXP.* La Habana, Cuba : s.n., 2008.
25. **Python.** Python Programming Language. [En línea] 2011. <http://python.org.com>.
26. **P.S.Foundation.** Python Programming Language. [En línea] [Citado el: 8 de diciembre de 2011.] <http://www.python.org>.
27. **Python.** *Tutorial de Python* . julio 21, 2009. Relase 2.5.2.
28. **MasterMagazine.** Definición de Herramienta. *Significado y definición de Herramienta.* [En línea] 2004. [Citado el: 10 de enero de 2012.] <http://www.mastermagazine.info/termino/5234.php>.
29. **Somos Libres.** [En línea] 2011. [Citado el: 12 de enero de 2012.] <http://www.somoslibres.org/modules.php?name=News&file=article&sid=2529>.
30. **Netbeans.** [En línea] 2011. [Citado el: 14 de enero de 2012.] <http://netbeans.org>.
31. **RapidSVN.** [En línea] 2011. [Citado el: 20 de enero de 2012.] <http://www.rapidsvn.org>.
32. **Pressman.** *Ingeniería de Software un enfoque práctico.* 2007, Quinta Edición.
33. **Díaz Pastrana, Dainerys y Pérez Pérez, Joisel .** *Modelo para la ayuda en la toma de decisiones relativas a estilos y patrones arquitectónicos.* La Habana, Cuba : s.n., 2008, junio.
34. **Python.** Python dev. [En línea] 2008. [Citado el: 7 de febrero de 2012.] <http://www.python.org/dev/peps/pep-0008/>.
35. **EcuRed.** [En línea] 2011. [Citado el: 20 de febrero de 2012.] [http://www.ecured.cu/idex.php/Protocolo\\_simple\\_de\\_acceso\\_a\\_objetos\\_\(SOAP\)](http://www.ecured.cu/idex.php/Protocolo_simple_de_acceso_a_objetos_(SOAP)).

36. **C., Benjamín González.** Desarrolloweb.com. [En línea] 7 de julio de 2004. [Citado el: 1 de marzo de 2012.] <http://www.Desarrolloweb.com/articulos/1557.php>.

## Bibliografía

[En línea] [Citado el: 3 de marzo de 2012.]

[http://www.usc.es/estaticos/atic/sistemas/cursolinux/avanzado\\_2a\\_ed/cups.pdf](http://www.usc.es/estaticos/atic/sistemas/cursolinux/avanzado_2a_ed/cups.pdf).

**Alex Fernández y Blasco Gómez.** <http://www.arqueolab.org/el-nacimiento-de-la-tecnologia>. [En línea] Arqueolab. [Citado el: 23 de octubre de 2011.] <http://www.arqueolab.org/el-nacimiento-de-la-tecnologia>.

**Bright Hub.** [En línea] 2011. [Citado el: 11 de noviembre de 2011.] <http://www.brighthub.com/computing/linux/articles/41828.aspx>.

**C., Benjamín González.** Desarrolloweb.com. [En línea] 7 de julio de 2004. [Citado el: 1 de marzo de 2012.] <http://www.Desarrolloweb.com/articulos/1557.php>.

**Carbajo, Ricardo R.** *Sistema de Certificación y Homologación de Hardware*. La Habana, Cuba : s.n., 2012.

**CUPS.** [En línea] [Citado el: 3 de marzo de 2012.] <http://www.cups.org/>.

**Definición ABC.** [En línea] 2012. [Citado el: 1 de noviembre de 2011.] <http://www.definicionabc.com/general/diagnostico.php>.

**Diana Lilian Antúnez Ginarte y Mario Lázaro García.** *Personalización de la herramienta de gestión de inventarios OCS Inventory NG*. La Habana, Cuba : s.n., mayo, 2011.

**Díaz Pastrana, Dainerys y Pérez Pérez, Joisel .** *Modelo para la ayuda en la toma de decisiones relativas a estilos y patrones arquitectónicos*. La Habana, Cuba : s.n., 2008, junio.

**Dictionary.com.** Dictionary.com. [En línea] 2012. [Citado el: 1 de noviembre de 2011.] <http://dictionary.reference.com/browse/process>.

**EcuRed.** [En línea] 25 de noviembre de 2010. [Citado el: 5 de mayo de 2012.] <http://www.gnu.org/software/pythonwebkit/>.

**EcuRed.** [En línea] 2011. [Citado el: 20 de febrero de 2012.] [http://www.ecured.cu/idex.php/Protocolo\\_simple\\_de\\_acceso\\_a\\_objetos\\_\(SOAP\)](http://www.ecured.cu/idex.php/Protocolo_simple_de_acceso_a_objetos_(SOAP)).

**Eduardo Alejandro Cuesta Llanes y Yurenia Blanco Hernández.** *MED: Módulo para la edición y depuración de código Octave integrado al ambiente EIDMAT*. La habana, Cuba : s.n., 2010.

**El blog de Marcelo.** *Navegador Simple con Python+Webkit/GTK*. [En línea] 19 de noviembre de 2009. [Citado el: 8 de abril de 2012.] <http://blog.marcelofernandez.info/2009/11/navegador-simple-con-pywebkitgtk/>.

**Fuentes, Allan Pierra.** *Conceptualización y Reestructuración Estratégica de la Distribución Cubana de GNU/Linux "Nova"*. La Habana, Cuba : s.n., octubre, 2011.

**GNU.** Categorías de software libre y no libre. [En línea] 30 de 12 de 2011. [Citado el: 29 de octubre de 2011.] <http://www.gnu.org/philosophy/categories.es.html>.

- González, Camilo Denis.** doutesdeslibertas. [En línea] 21 de febrero de 2011. [Citado el: 5 de diciembre de 2011.] <http://linux.die.net/man/1/usb-devices>.
- Haas, Juergen.** About.com. [En línea] 2011. [Citado el: 13 de noviembre de 2011.] [http://linux.about.com/library/cmd/blcmdl2\\_sysinfo.htm](http://linux.about.com/library/cmd/blcmdl2_sysinfo.htm), s.l. : 2011.
- Hernández, Lisandra Cala.** *Propuesta de integración y nuevas herramientas para el desarrollo de la Plataforma Cubana de Migración a Software Libre.* . La Habana, Cuba : s.n., 2010.
- ITRestauracion.** [En línea] 27 de diciembre de 2010. [Citado el: 24 de noviembre de 2011.] <http://www.itrestauración.com.ar/?p=1238>.
- Linux Amigos.** [En línea] 2009. [Citado el: 15 de noviembre de 2011.] <http://linuxandfriends.com/2009/02/23/lshw-command-list-hardware-information-in-linux>.
- Lorente, Cesar de la Torre.** *Guía\_Arquitectura\_N-Capas\_DDD\_NET\_4.* España : s.n., 2010.
- MasterMagazine.** Definición de Herramienta. *Significado y definición de Herramienta.* [En línea] 2004. [Citado el: 10 de enero de 2012.] <http://www.mastermagazine.info/termino/5234.php>.
- MinRex.** Sitio del Ministerio de Relaciones Exteriores de Cuba. [En línea] Cuba, 2005. [Citado el: 28 de octubre de 2011.] [www.cubaminrex.cu/sociedad\\_informacion/cuba\\_si/Masivo\\_TIC.htm](http://www.cubaminrex.cu/sociedad_informacion/cuba_si/Masivo_TIC.htm).
- Netbeans.** [En línea] 2011. [Citado el: 14 de enero de 2012.] <http://netbeans.org>.
- Packages Debian.** [En línea] 2011. [Citado el: 10 de noviembre de 2011.] <http://packages.debian.org/es/squeeze/gnome-device-manager>.
- ParadigmaOO.** *El paradigma de la Programación Orientada a Objetos.* <http://eugeniabahit.blogspot.com>.
- Patrones de diseño.** [En línea] 2001. [Citado el: 4 de febrero de 2012.] <http://www.google.com/cu/url?sa=t&rct=j&q=arquitecturas+de+software&source=web&cd=2&ved=0CGIQFjAB&url=http%3A%2F%2Fwww.sceu.frba.utn.edu.ar%2Fe-learning%2Fcursos-a-distancia%2FInformatica-y-Sistemas%2FArquitecturas-de-Software%2Ftemario.html&ei=qAq6T96uAo>.
- Pereira, Leandro.** HardInfo. [En línea] 2009. [Citado el: 10 de noviembre de 2011.] <http://www.guia-ubuntu.org/index.php?title=Hardinfo>.
- Pressman.** *Ingeniería de Software un enfoque práctico.* 2007, Quinta Edición.
- P.S.Foundation.** Python Programming Language. [En línea] [Citado el: 8 de diciembre de 2011.] <http://www.python.org>.
- Python.** [En línea] 11 de diciembre de 2010. [Citado el: 24 de abril de 2012.] <http://wiki.python.org/moin/WebKit>.
- Python.** [En línea] 2010. [Citado el: 20 de enero de 2012.] <http://www.pygtk.org/>.
- Python.** Python dev. [En línea] 2008. [Citado el: 7 de febrero de 2012.] <http://www.python.org/dev/peps/pep-0008/>.

**Python.** Python Programming Language. [En línea] 2011. <http://python.org.com>.

**Python.** *Tutorial de Python* . julio 21, 2009. Relase 2.5.2.

**RapidSVN.** [En línea] 2011. [Citado el: 20 de enero de 2012.] <http://www.rapidsvn.org>.

**Reynoso, Carlos Billy.** <http://eugeniabahit.blogspot.com>. UNIVERSIDAD DE BUENOS AIRES : s.n., 2004.

**Romero, Gladys Marsi Peñalver.** *Metodología SXP*. La Habana, Cuba : s.n., 2008.

**Rosario, Jimmy.** Observatorio para la CiberSociedad 2012. *La Tecnología de la Información y la Comunicación (TIC)*. [En línea] 2012. [Citado el: 28 de octubre de 2011.]  
<http://www.cibersociedad.net/archivo/articulo.php?art=218>.

**Salmón, Ramón Paumier.** *Metodología para la migración a software libre de la Universidad de las Ciencias Informáticas*. La Habana, Cuba : s.n., 2007.

**SOFTPEDIA.** [En línea] 27 de julio de 2011. [Citado el: 19 de noviembre de 2011.]  
<http://www.softpedia.com/get/Others/Finances-Business/OCS-Inventory-NG-Agent.shtml>.

**Somos Libres.** [En línea] 2011. [Citado el: 12 de enero de 2012.]  
<http://www.somoslibres.org/modules.php?name=News&file=article&sid=2529>.

**The Free Dictionary.** [En línea] 2012. [Citado el: 3 de noviembre de 2011.]  
<http://es.thefreedictionary.com/compatibilidad>.

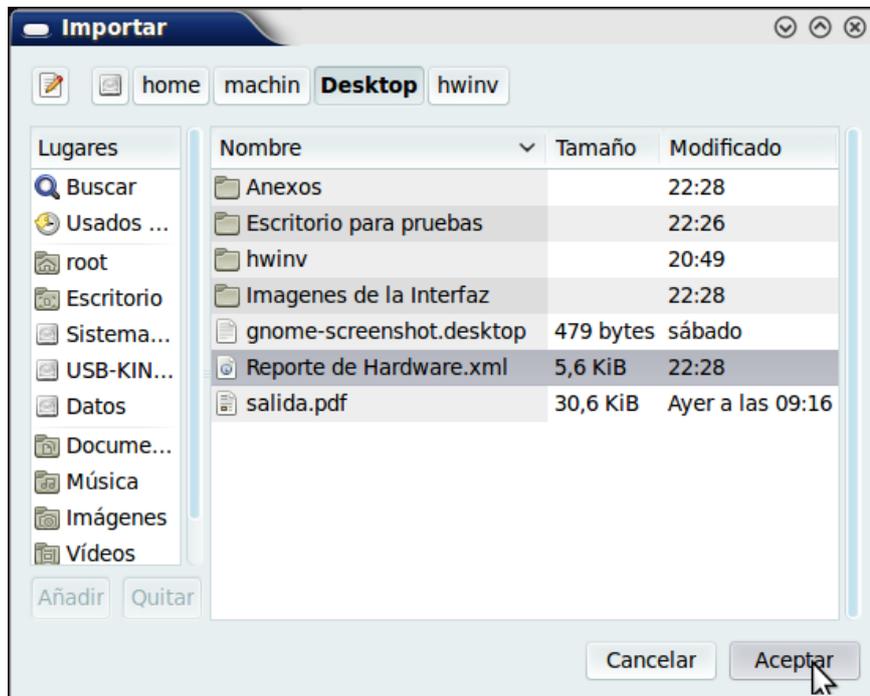
**The Gtk Class Reference.** *Gtk.TreeView*. [En línea] <http://www.pygtk.org/docs/pygtk/class-gtktreeview.html>.

**Troyano.** Troyano. [En línea] 24 de agosto de 2009. [Citado el: 28 de octubre de 2011.]  
<http://www.estrellaroja.info/?p=209>.

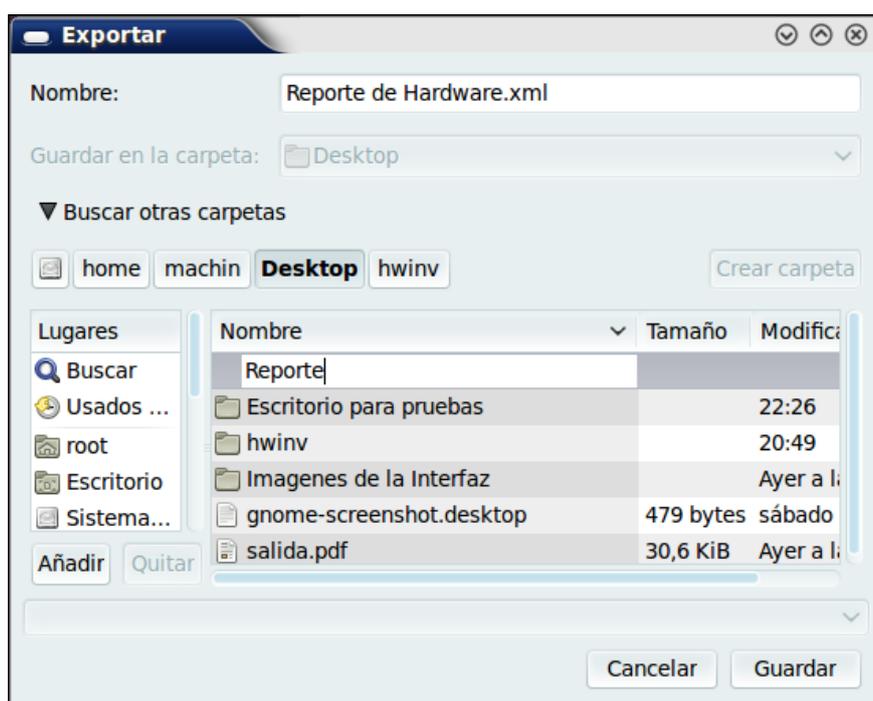
**Villazón, Yoandy Pérez.** *Metodología para la migración a Software Libre de las universidades del Ministerio Superior (MES)*. La Habana, Cuba : s.n., 2008.

## Anexos

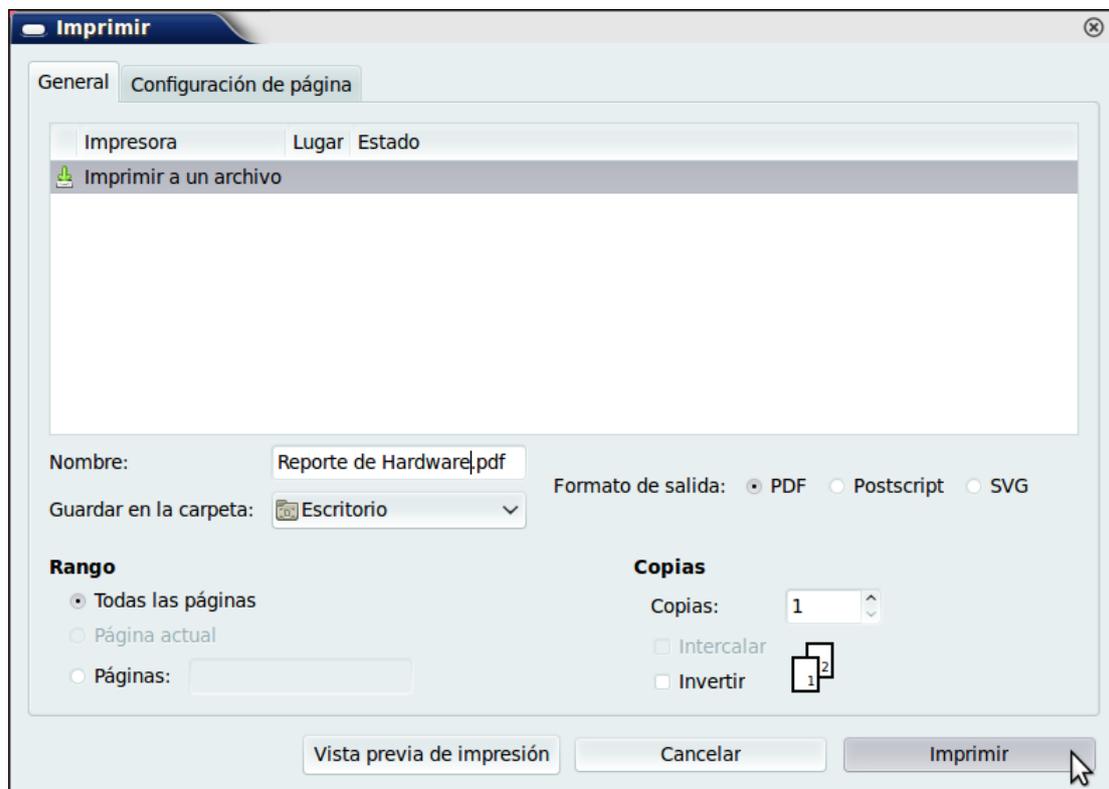
### 1. Importar un reporte antes realizado por SistDCH.



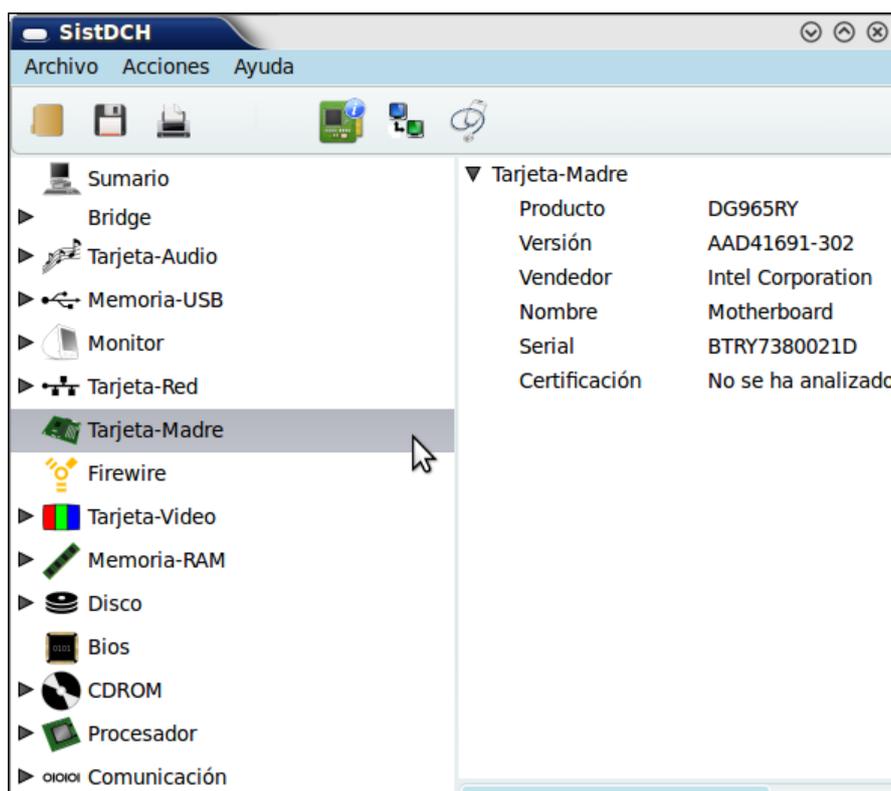
### 2. Exportar el levantamiento realizado a la estación de trabajo.



### 3. Imprimir levantamiento de hardware y guardarlo en formato .pdf.



### 4. Aplicación SistDCH.



## 5. Lista de Riesgos

Riesgo	Tiempos de Riesgo	Impacto	Descripción	Probabilidad	Efectos	Mitigación de Riesgo
1	Estimación	Desarrollo	Complejidad de las tecnologías a utilizar.	Alta	Tolerables	Obteniendo libros específicos sobre el tema.
2	Tecnológico	Retraso del trabajo.	Rotura de una computadora donde se encuentra información importante.	Alta	Serios	Realización de salvallas frecuentes de la información.
3	Personal	Retraso de las actividades	Enfermedad de uno de los miembros del equipo.	Alta	Tolerables	Capacitar varias personas en cada tema.
4	Tecnológico	Retraso en las actividades	Afectación del fluido eléctrico.	Baja	Tolerancia	Buscar horarios alternativos para trabajar.
5	Tecnológico	Productividad.	Desconexión de la red constantemente.	Alta	Serios	Comunicar al jefe de proyecto.

## **Glosario de términos.**

**BSD:** Berkeley Software Distribution (en español, "distribución de software berkeley") es un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

**Debuggers:** depurador, programa especial utilizado para encontrar errores en otros programas.

**Dirección IP:** dirección de un ordenador dentro de una red con protocolo TCP/IP.

**GNOME:** es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix y derivados Unix como GNU/Linux, BSD o Solaris; compuesto enteramente de software libre.

**GPL:** es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**GNU:** proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.

**HAL:** Hardware Abstraction Layer era un proyecto de software que proveía una capa de abstracción de hardware para sistemas Unix-like.

**HTTP:** Hypertext Transfer Protocol, en español protocolo de transferencia de hipertexto.

**Historias de Usuario (HU):** secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

**IDE:** Entorno de desarrollo integrado o en inglés Integrated Development Environment ('IDE'). Es un programa compuesto por un conjunto de herramientas para un programador desde el que se pueden editar programas, compilarlos y depurarlos.

**Instalar:** incorporar a la computadora un programa o dispositivo para ser utilizado.

**Ítem:** palabra latina adoptada y difundida por el inglés. Significa cada uno de los artículos que constituyen un escrito. Como sustantivo se utiliza para designar los mismos artículos. Los angloparlantes usan intensamente la palabra para designar también a un artículo en el sentido de objeto unitario.

**Java:** lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible.

**MySQL:** es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual.

**Núcleo:** parte esencial de un sistema operativo que provee los servicios más básicos del sistema. Se encarga de gestionar los recursos como el acceso seguro al hardware de la computadora, determina qué programa accederá a un determinado hardware, si dos o más programas quieren usarlo al mismo tiempo, entre otras.

**Privativo:** el software no libre (también llamado software propietario, software privativo, software privado, software con propietario o software de propiedad) se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo, cuyo código fuente no está disponible.

**Prototipo:** ejemplar original o primer molde en que se fabrica una figura u otra cosa. Producto terminado al que se llega en la fase de investigación y desarrollo, pero que no es, todavía, el definitivo que se comercializa.

**Python:** lenguaje de programación creado por Guido van Rossum en el año 1990. Es comparado habitualmente con TCL, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation.

**RAD:** Desarrollo Rápido de Aplicaciones.

**RAM:** se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo.

**Releases:** nueva versión de una aplicación informática.

**Servidor:** computadora central de un sistema de red que provee servicios y recursos (programas, comunicaciones, archivos, etc.) a otras computadoras (clientes) conectadas a ella.

**Soberanía Tecnológica:** capacidad de un país para desarrollarse en dicho campo en forma autónoma. No supone independencia absoluta, sino capacidad decisional sobre su uso y desarrollo.

**Software Privativo:** programas o aplicaciones en el que el usuario no puede acceder al código fuente o tiene un acceso restringido y, por tanto, se ve limitado en sus posibilidades de uso, modificación y redistribución.

**Solaris:** es un sistema operativo de tipo Unix desarrollado desde 1992.

**Solución:** es la propuesta de la solución definitiva al problema, es analizada con respecto a su: posible impacto, viabilidad y conveniencia.

**SXP:** metodología compuesta por las metodologías SCRUM y XP, especialmente indicada para proyectos pequeños, rápido cambio de requisitos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad.

**SWL:** Software Libre, se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

**TCP/IP:** Transfer Control Protocol / Internet Protocol. Protocolos que se utiliza en Internet para transmitir datos. El TCP está orientado a la conexión que establece una línea de diálogo entre el emisor y el receptor antes de que se transfieran los datos.

**TIC:** Tecnologías de la Información y las Comunicaciones.

**UML:** Lenguaje Unificado de Modelado.

**Usuarios:** las personas que utilizan el servicio.