

**Universidad de las Ciencias Informáticas**

Facultad 2



**Título:** Descubrimiento automático de la red de computadoras para el sistema GRHS.

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

Autores: Arlety de la Caridad Sebastian Martínez

René Juan Pérez de Corcho Hernández

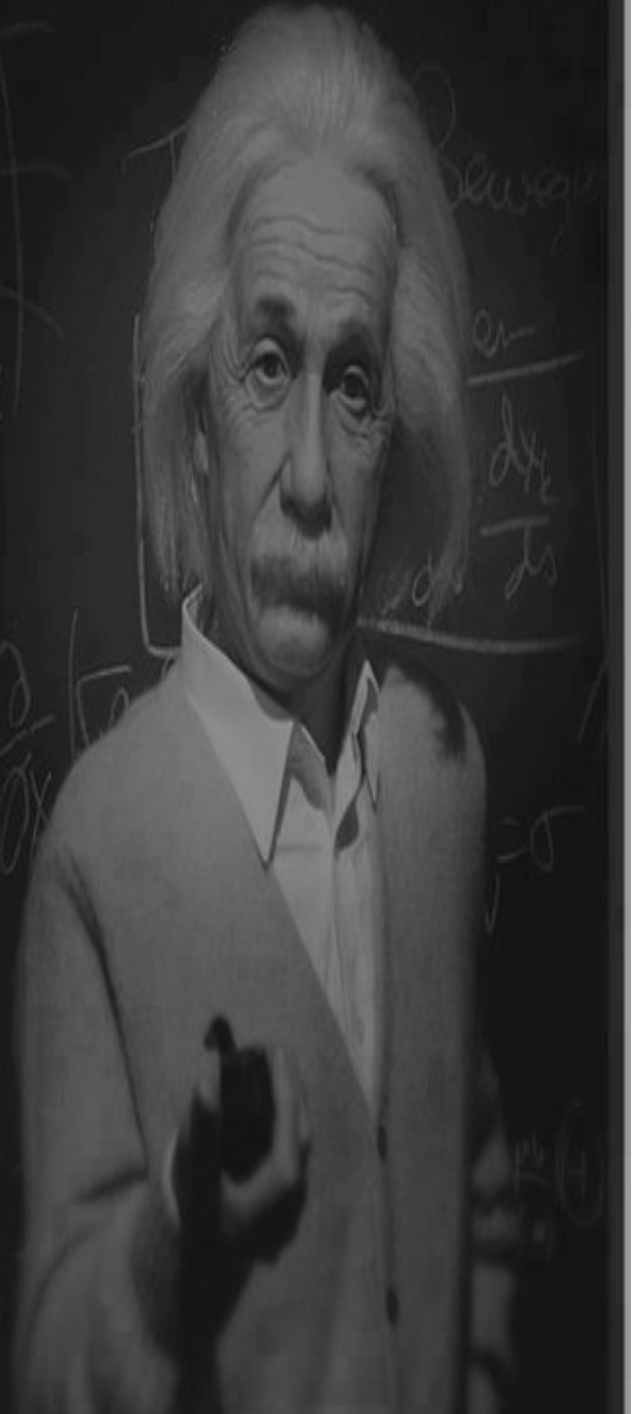
Tutor: MSc. Ramón Alexander Anglada Martínez

La Habana, Junio del 2015

Descubrimiento automático de la red de computadoras para el sistema GRHS.

"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."

Albert Einstein



# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Declaración de Autoría

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

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

---

Arlety de la Caridad Sebastian Martínez  
(Autor)

---

René Juan Pérez de Corcho Hernández  
(Autor)

---

MSc. Ramón Alexander Anglada Martínez  
(Tutor)

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Datos de Contacto

Arlety de la Caridad Sebastian Martínez

Dirección: Avenida 63 #2018 entre 20B y 20C. Cotorro. La Habana.

Correo electrónico: [acsebastian@estudiantes.uci.cu](mailto:acsebastian@estudiantes.uci.cu)

René Juan Pérez de Corcho Hernández

Dirección: Calle Independencia edificio 3 apto 18. Majagua. Ciego de Ávila.

Correo electrónico: [rjperezdecorcho@estudiantes.uci.cu](mailto:rjperezdecorcho@estudiantes.uci.cu)

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Agradecimientos

(...)

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

Dedicatoria

*Dedico esta tesis a la persona más importante de mi vida, a mi mamá por estar siempre a mi lado, por su apoyo incondicional y porque gracias a ella estoy hoy aquí. A mi abuela Marta por guiarme en cada una de las cosas que he hecho aun cuando no se encuentra físicamente. También a mi familia por brindarme tanto amor y a mis amistades por estar siempre cuando las he necesitado.*

*Arlety.*

*Dedico esta tesis a mis abuelos por estar siempre a mi lado, a mi hermano Iván y su madre por ser más que mi familia. También a mis familiares en general por transmitirme tanto cariño y a mis amistades por haber tenido el privilegio de conocerlas.*

*A mi padre por haber formado lo que soy, en especial a mi madre que aunque no nos acompaña en existencia, siempre confió en mí, esperó lo mejor de mí y fue cada segundo mi motor impulsor.*

*René*

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Resumen

El inventario de los recursos de hardware y software en una red de computadoras es cada vez más una tarea necesaria con la creciente evolución tecnológica. Con el fin de automatizar este difícil y complejo proceso se han creado innumerables herramientas informáticas en todo el mundo. El Centro de Telemática (TLM) de la Universidad de las Ciencias Informáticas (UCI) desarrolló una herramienta similar, denominada Gestor de Recursos de Hardware y Software (GRHS), con el objetivo de inventariar los recursos de hardware y software en una red de computadoras. Sin embargo, GRHS no permite descubrir automáticamente todas las máquinas de la red de computadoras y cuales tienen instalada o no la herramienta, lo cual constituye una deficiencia del sistema que influye negativamente en el inventario del hardware y el software de la red de computadoras donde esté desplegado GRHS.

Se realizó un estudio del estado del arte de las herramientas, componentes y sistemas similares existentes para el descubrimiento de la red, tomando la tecnología nmap como solución base para resolver el problema planteado. La solución se desarrolló utilizando la metodología XP, utilizando el framework de desarrollo web Django 1.4.5, integrándose satisfactoriamente al sistema GRHS.

**Palabras Claves:** descubrimiento automático, red de computadoras, nmap, GRHS.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Índice General

Capítulo 1: Fundamentación Teórica .....	11
1.1 Introducción .....	11
1.2 Principales conceptos asociados a la Investigación .....	11
1.2.1 Red de computadoras .....	11
1.2.2 Biblioteca .....	11
1.2.3 Componente .....	11
1.2.4 Protocolo.....	11
1.3 Sistemas similares que realizan descubrimiento de la red.....	12
1.3.1 SolarWinds LANsurveyor.....	12
1.3.2 Spectrum .....	12
1.3.3 PRTG Network Monitor.....	13
1.4 Principales bibliotecas y/o componentes para realizar el descubrimiento de la red de computadoras .....	13
1.4.1 Nmap .....	14
1.4.2 Scapy.....	14
1.4.3 Zenmap .....	14
1.4.4 Masscan .....	15
1.4.5 Python-nmap .....	15
1.5 Protocolos de referencia en la investigación.....	16
1.5 Metodología, lenguaje de programación, herramientas y tecnologías a utilizar...	17
1.5.1 Metodología de desarrollo .....	17



# Descubrimiento automático de la red de computadoras para el sistema GRHS.

1.5.2 Lenguaje de programación .....	19
1.5.3 Entorno de Desarrollo Integrado.....	20
1.5.4 Framework de Desarrollo.....	20
1.5.4 Sistema Gestor de Bases de Datos.....	21
1.5.5 Herramienta CASE .....	22
1.5.6 Lenguaje de Modelado BPMN .....	22
1.6 Conclusiones parciales .....	23
Capítulo 2: Exploración y Planificación .....	24
2.1 Introducción .....	24
2.2 Propuesta de solución.....	24
2.3 Modelado del proceso de negocio .....	26
2.4 Características del sistema .....	28
2.5 Exploración .....	29
2.5.1 Historias de Usuarios.....	30
2.6 Planificación .....	38
2.6.1 Estimación del esfuerzo por HU .....	39
2.6.2 Plan de Iteraciones .....	39
2.6.3 Plan de Entregas .....	41
2.7 Conclusiones parciales .....	41
Capítulo 3: Diseño e Implementación.....	42
3.1 Introducción .....	42
3.2 Arquitectura de software .....	42

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

3.2.1 Arquitectura Cliente-Servidor.....	42
3.3 Patrón Arquitectónico.....	43
3.3.1 Modelo Plantilla Vista .....	44
3.4 Representación de las Capas de la Arquitectura .....	45
3.4.1 Capa Vista .....	46
3.4.2 Capa Modelo .....	47
3.4.3 Capa Plantilla.....	48
3.5 Patrones de Diseño .....	49
3.5.1 Patrón GRASP.....	49
3.6 Tarjetas CRC .....	51
3.7 Modelo Físico de la Base de Datos .....	53
3.8 Tareas de Ingeniería.....	54
3.9 Pruebas.....	58
3.9.1 Pruebas unitarias.....	58
3.9.2 Pruebas de aceptación .....	60
3.10 Conclusiones parciales .....	67
Conclusiones Generales .....	69
Recomendaciones.....	70
Bibliografía .....	71

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Índice de Tablas

Tabla 1. HU1. Adicionar automáticamente una subred.....	31
Tabla 2. HU2. Descubrir computadoras encendidas por subred.....	32
Tabla 3. HU3. Verificar si el cliente está instalado en cada computadora.....	33
Tabla 4. HU4. Mostrar características esenciales de las computadoras.....	34
Tabla 5. HU5. Configurar automáticamente el escaneo de la red.....	35
Tabla 6. HU6. Exportar a formato (Excel). .....	37
Tabla 7. HU7. Mostrar estadísticas generales. ....	38
Tabla 8. Estimación del esfuerzo por HU.....	39
Tabla 9. Plan de iteraciones.....	41
Tabla 10. Plan de entregas. ....	41
Tabla 11. Tarjeta CRC: Rango.....	52
Tabla 12. Tarjeta CRC: Rango2.....	52
Tabla 13. Tarjeta CRC: NetworkInterface .....	52
Tabla 14. Tarjeta CRC: viewEscanear .....	53
Tabla 15. Tarea de Ingeniería 1- Almacenar subredes. ....	54
Tabla 16. Tarea de Ingeniería 2- Mostrar computadoras encendidas por subred.....	55
Tabla 17. Tarea de Ingeniería 3- Descubrir clientes. ....	55
Tabla 18. Tarea de Ingeniería 4- Obtener características de las computadoras.....	56
Tabla 19. Tarea de Ingeniería 5- Configurar escaneo.....	57
Tabla 20. Prueba de Aceptación #1: HU: Adicionar automáticamente una subred.....	61
Tabla 21. Prueba de Aceptación #2: HU: Adicionar automáticamente una subred.....	62

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Tabla 22. Prueba de Aceptación #3: HU: Descubrir computadoras encendidas por subred. ....	62
Tabla 23. Prueba de Aceptación #4: HU: Descubrir computadoras encendidas por subred. ....	63
Tabla 24. Prueba de Aceptación #5: HU: Verificar si el cliente está instalado en cada computadora. ....	63
Tabla 25. Prueba de Aceptación #6: HU: Verificar si el cliente está instalado en cada computadora. ....	64
Tabla 26. Prueba de Aceptación #7: HU: Mostrar características esenciales de las computadoras. ....	65
Tabla 27. Prueba de Aceptación #8: HU: Mostrar características esenciales de las computadoras. ....	65
Tabla 28. Prueba de Aceptación #9: HU: Configurar automáticamente el escaneo de la red. ....	66
Tabla 29. Prueba de Aceptación #10: HU: Exportar a formato (Excel). ....	66
Tabla 30. Prueba de Aceptación #11: HU: Mostrar estadísticas generales. ....	66

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Índice de Ilustraciones

Ilustración 1. Propuesta de solución del módulo de descubrimiento automático de la red de computadoras. ....	25
Ilustración 2. Modelado del proceso de negocio del módulo de descubrimiento automático de la red de computadoras. ....	28
Ilustración 3. Arquitectura Cliente - Servidor. ....	43
Ilustración 4. Patrón de arquitectura MTV. ....	45
Ilustración 5. Capa Vista. ....	46
Ilustración 6. Capa Modelo. ....	47
Ilustración 7. Capa Plantilla. ....	49
Ilustración 8. Modelo Físico de la Base de Datos. ....	53
Ilustración 9. Prueba unitaria a la iteración 1 ....	59
Ilustración 10. Prueba unitaria del Modelo ....	59
Ilustración 11. Prueba unitaria a la iteración 2 ....	59
Ilustración 12. Prueba unitaria a la iteración 3 ....	60
Ilustración 13. Resultado de las pruebas de aceptación. ....	67

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) se encuentran actualmente en constante desarrollo. Dada la necesidad de la sociedad de estar al nivel de dicho avance, tanto profesionales como estudiantes están motivados a buscar métodos y soluciones para automatizar sus tareas, en el menor tiempo y costo posible.

(1)

En la actualidad resulta casi imposible predecir hasta qué punto pueda influir en el futuro las tecnologías informáticas. Aunque se vive en un mundo controlado por redes de computadoras, se encuentran una variedad de recursos que no son controlados y hasta desconocidos por sus propietarios o administradores de redes. Así surge la necesidad de herramientas que proporcionen información de hardware y software de las redes de computadoras e informan al administrador de red sobre los cambios que son realizados en la red de computadoras que administra. Todo este proceso se conoce como inventario del hardware y el software de la red de computadoras.

Cuba no está exenta de esta necesidad y apuesta actualmente al desarrollo de la informática como variante para el avance tecnológico. Con este objetivo se creó la Universidad de la Ciencias Informáticas (UCI), que tiene dentro de sus misiones servir de soporte a la industria cubana del software. (2) En la actualidad, la UCI cuenta con un sistema de redes interconectadas. Dentro de las que se encuentran recursos informáticos, los cuales no son inventariados como es debido. Esta tarea de realizar el inventario de hardware y software se dificulta porque la mayoría se realiza de forma manual y en algunos casos la información sobre el hardware y el software es desconocida por los administradores de esta tecnología.

Con el fin de disminuir estas dificultades, el Centro de Telemática (TLM) de la Facultad 2 desarrolló una herramienta informática, la cual está desplegándose en la universidad, la misma permite realizar el inventario de hardware y software en una red de computadoras, denominada Gestor de Recursos de Hardware y Software (GRHS). Este sistema permite mantener un control sobre los recursos de hardware presentes en las

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

computadoras de la red que han sido inventariadas, así como conocer el software que se utiliza en las estaciones de trabajo.

Este sistema para realizar la tarea de inventario de hardware y software de la red de computadoras se apoya en un cliente (Gclient, Cliente de GRHS) que se instala en cada computadora. Esta instalación se realiza de forma directa por los administradores de tecnología de cada docente.

Sin embargo a pesar de todas las ventajas antes expuestas que brinda el sistema GRHS, la herramienta posee un grupo de dificultades relacionadas con el conocimiento o descubrimiento de las computadoras presentes en la red donde se va a instalar Gclient. Primeramente el sistema no cuenta con un mecanismo automático que le permita conocer el total de computadoras a inventariar y su ubicación en la red, lo cual contribuye a la instalación de manera manual de Gclient aumentando la probabilidad de cometer errores por parte de los administradores de tecnología que están designados a esta tarea, los cuales en algunos casos obvian estaciones de trabajo que deben ser inventariadas. El sistema no es capaz de identificar después de realizado el inventario inicial, las computadoras nuevas presentes en la red que adquieren una dirección IP y consumen servicios, las cuales también deben ser inventariadas. Además no permite comprobar de manera automática si fue instalado Gclient en cada una de las estaciones de trabajo de la red, lo cual influye de manera negativa en el control de los recursos de hardware y software que trata de garantizar la herramienta. Por último el sistema no reconoce de forma constante las computadoras que ya no poseen Gclient instalado lo cual pudo haber sido producto de formatear el sistema operativo sin autorización o por violar alguna política o medida de seguridad informática.

Teniendo en cuenta la situación problemática planteada se define como **problema a resolver**: La incapacidad del sistema GRHS para descubrir todas las computadoras presentes en la red, lo cual afecta negativamente la realización del inventario de hardware y software en todas las estaciones de trabajo de la red. Teniéndose como **objeto de estudio** de la investigación las redes de computadoras. El **campo de acción** se enmarca en la red de computadoras de la UCI donde está desplegado el sistema

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

GRHS. El **objetivo general** se dirige a desarrollar un módulo para el sistema GRHS que permita descubrir automáticamente todas las estaciones de trabajo de la red de computadoras.

Para alcanzar el objetivo trazado y dar solución al problema planteado se elaboraron las siguientes **tareas de investigación**:

- ✓ Realización de un estudio de los componentes sobre los que está desplegado el sistema GRHS.
- ✓ Realización de un estudio sobre los sistemas similares que realizan el descubrimiento automático en una red de computadoras.
- ✓ Utilización de las herramientas definidas por el proyecto GRHS y selección de la metodología para el desarrollo del módulo.
- ✓ Diseño de una propuesta de solución que permita realizar el descubrimiento automático de la red de computadoras en el sistema GRHS.
- ✓ Desarrollo de un módulo para el sistema GRHS que permita el descubrimiento de la red.
- ✓ Realización de las pruebas unitarias y las pruebas de aceptación para validar la solución propuesta.

Para el desarrollo del trabajo se tuvieron en cuenta diferentes métodos, los cuales fueron de suma importancia durante la investigación:

## **Métodos teóricos:**

**Analítico-sintético:** Este método fue utilizado en todo el proceso investigativo, permitiendo descomponer el problema en varias partes y resaltar los elementos esenciales en cada una. Posibilitando un mejor entendimiento y comprensión de la situación en cuestión, para luego con los resultados obtenidos realizar un análisis.



# Descubrimiento automático de la red de computadoras para el sistema GRHS.

**Histórico Lógico:** Se realizó un estudio del sistema GRHS desde sus inicios hasta la actualidad, permitiendo obtener conocimiento del mismo referente a cómo ha evolucionado, cómo se comporta y las necesidades que tiene en estos momentos.

## **Métodos empíricos:**

**Entrevista:** Se realizaron entrevistas al personal calificado en el tema de descubrimiento de la red de computadoras, pertenecientes al proyecto GRHS, que permitió contar con la información necesaria, convirtiéndose en una técnica de recopilación.

**Observación:** Se utilizó durante el proceso de obtención de información, en el seguimiento del trabajo del personal que labora en el proyecto GRHS.

En aras de estructurar el proceso de desarrollo del presente trabajo de una manera coherente y organizada, se dividió el mismo en un total de 3 capítulos:

## **Capítulo 1**

“Fundamentación Teórica”, aborda el estudio del arte del tema relacionado con el trabajo que se desarrolla, los conceptos y definiciones investigados durante el análisis de las tendencias actuales. Incluye además un estudio de las herramientas utilizadas para el desarrollo del módulo, así como las tecnologías y lenguajes existentes útiles para el proceso de implementación.

## **Capítulo 2**

“Exploración y Planificación”, en este capítulo se elabora la propuesta de solución, se explican las diferentes características que va a presentar el módulo a implementar y sus funcionalidades, de acuerdo con las fases de Exploración y Planificación que plantea la metodología XP.

## **Capítulo 3**

“Diseño e Implementación”, en este capítulo se define la arquitectura y los patrones de diseño utilizados en el módulo a desarrollar. Se confeccionan las tarjetas CRC para

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

lograr una mejor organización de las clases y se elaboraron las tareas de ingeniería correspondientes a cada HU. Además se evalúa la calidad de la solución final mediante la realización de pruebas, con el objetivo de solucionar errores y no conformidades presentes en la implementación.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Capítulo 1: Fundamentación Teórica

### **1.1 Introducción**

En el siguiente capítulo se expone la fundamentación teórica del módulo a desarrollar y se brinda un análisis de los principales conceptos asociados al descubrimiento de la red de computadoras. Como aspectos esenciales se profundiza en el estudio de los sistemas similares, así como la selección de la metodología y el lenguaje a emplear para dar solución al objetivo general planteado.

### **1.2 Principales conceptos asociados a la Investigación**

#### 1.2.1 Red de computadoras

Una red de computadoras es una interconexión de computadoras para compartir información, recursos y servicios. Esta interconexión puede ser a través de un enlace físico o inalámbrico. (3)

#### 1.2.2 Biblioteca

En el mundo de la informática, una biblioteca se usa para referirse a un programa que contiene varias funciones para lograr un propósito bien definido y específico. Estas bibliotecas están diseñadas de tal forma que son fácilmente integradas a otros programas que requieren usar las funcionalidades que ella ofrece. (4)

#### 1.2.3 Componente

Componente es aquello que forma parte de la composición de un todo. Se trata de elementos que, a través de algún tipo de asociación o contigüidad, dan lugar a un conjunto uniforme. (5)

#### 1.2.4 Protocolo

Conjunto estándar de reglas y requisitos que permite que dos elementos electrónicos se conecten e intercambien información el uno con el otro. Los protocolos regulan la transmisión de datos entre dispositivos y entre una red de trabajo con varios

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

dispositivos mediante un control de errores y una especificación del método de compresión de datos que utilicen. (6)

## **1.3 Sistemas similares que realizan descubrimiento de la red**

A continuación se describen varios sistemas similares que se utilizan para el descubrimiento de la red.

### 1.3.1 SolarWinds LANsurveyor

Permite descubrir y mapear de manera automática cada uno de los dispositivos de la red. LANsurveyor ofrece una sofisticada tecnología de descubrimiento multinivel para generar mapas fáciles de ver y entender, además pueden ser exportados a Microsoft Visio, eliminando la necesidad de dibujar manualmente un mapa de red.

Las principales características incluyen:

- Generación de mapas de red en Microsoft Visio.
  - Detección automática de nuevos dispositivos y cambios en la topología de red.
  - Inventario de recursos de red.
  - Creación de una base de datos de la topología de red.
  - Descubrimiento multinivel (OSI capa 2 y 3) incluyendo: Dirección IP, Mac Address, Nombre del DNS, Nombre del Node, Switch Port connection y Último usuario logeado.
- (7)

### 1.3.2 Spectrum

Spectrum proporciona la supervisión en tiempo real de toda la infraestructura de red del usuario, recopilando de forma proactiva datos procedentes de los diferentes dispositivos de red, sistemas, aplicaciones y elementos de seguridad que la conforman. Asimismo, la herramienta permite identificar cualquier problema de rendimiento o de disponibilidad de la red. Dentro de las principales características de Spectrum se destacan:

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

- ✓ Inteligente y fácil de instalar.
- ✓ Arquitectura sólida, con alta disponibilidad y escalable.
- ✓ Gestión multifabricante, además de detectar fallos, los pueda prever.
- ✓ Genera informes de planificación para operaciones y ejecutivos. (8)

### 1.3.3 PRTG Network Monitor

Es un software de monitorización de red de la empresa Paessler AG. PRTG se ejecuta en Windows y utiliza los protocolos SNMP, sniffing de paquetes, WMI, IP SLA y NetFlow para obtener los datos necesarios para las estadísticas y sensores.

Los principales beneficios de PRTG Network Monitor son:

- ✓ Se evitan las pérdidas causadas por fallos en la red sin detectar.
- ✓ Reducción de costes ya que se puede comprar el ancho de banda y hardware según las necesidades reales. (9)

Luego de realizarse un estudio de los sistemas similares que realizan un descubrimiento de la red, se llegó a la conclusión de que dichos sistemas no pueden integrarse a la plataforma GRHS, pues no cubren las necesidades del sistema, porque no son adaptables al mismo, dado que sus funcionalidades no son aprovechables, son costosos y tienen licencia propietaria.

## **1.4 Principales bibliotecas y/o componentes para realizar el descubrimiento de la red de computadoras**

A continuación se describen las principales bibliotecas y/o componentes que realizan descubrimiento de la red.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## 1.4.1 Nmap

Nmap (Network Mapper) es una biblioteca libre y de código abierto, de gran utilidad para la detección de redes y auditorías de seguridad. A muchos sistemas y administradores de red también les resulta útil para tareas como inventario de la red, los horarios de actualización de servicio de gestión y monitoreo de servicios. Nmap utiliza paquetes IP (Internet Protocol) para determinar qué servicios están disponibles en la red de computadoras, muestra de cada computadora el servicio, el sistema operativo y su versión. Fue diseñado para escanear rápidamente grandes redes. Nmap se ejecuta en sistemas operativos Linux, Windows y Mac OS X. (10)

## 1.4.2 Scapy

Scapy es una potente biblioteca escrita y soportada en Python la cual permite la creación, manipulación e inyección de paquetes de un entorno de red. Con Scapy se pueden crear, capturar, manipular y analizar una gran cantidad de paquetes con varios protocolos de red existentes. Ejecuta ataques de reconocimiento y escaneo de puertos de forma automática, también se pueden manipular los campos de los paquetes que se envían a los objetivos, con Scapy se tienen las respuestas de los objetivos en estado “crudo” lo que quiere decir que el trabajo interpretativo se deja plenamente al programador. Scapy hace principalmente dos tareas: el envío de paquetes y el recibimiento de respuestas. También define un conjunto de paquetes, que los envía, los recibe y coincide con cada petición de respuesta recibida. (11)

## 1.4.3 Zenmap

Zenmap es una biblioteca multiplataforma (Linux, Windows, Mac OS X, BSD) es gratuita y de código abierto, tiene como objetivo hacer que Nmap resulte más fácil de usar, proporciona características avanzadas para los usuarios de Nmap experimentados. Realiza frecuentes exploraciones utilizadas que se pueden guardar como perfiles para que sean fáciles de ejecutar repetidamente. Cuenta con un creador de comandos que permite la interacción de líneas de códigos con la biblioteca Nmap.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Los resultados del análisis pueden guardarse para ser consultados en cualquier ocasión. (12)

## 1.4.4 Masscan

Masscan es un escáner rápido de puerto TCP. Produce resultados similares a nmap. Internamente, funciona más como scanrand, unicornscan y Zenmap, utilizando la transmisión asíncrona. La principal diferencia es que es más rápido que estos otros escáneres. Además, es más flexible, permitiendo rangos de direcciones y rangos de puertos arbitrarios. (13)

## 1.4.5 Python-nmap

Con esta biblioteca es posible ejecutar escaneos de forma síncrona o asíncrona. Un escaneo síncrono, consiste simplemente en que python-nmap bloqueará el hilo de ejecución principal hasta que el escaneo finalice su ejecución. Un escaneo asíncrono ejecuta el escaneo nmap en un nuevo hilo de ejecución y permite que el hilo principal siga ejecutándose. En un escaneo asíncrono, se permite la definición de una función de callback que será invocada de forma automática cuando se obtenga un resultado sobre alguno de los puertos analizados. Para el desarrollo del módulo se decidió utilizar esta biblioteca pues puede integrarse al sistema GRHS satisfactoriamente.

De esta manera, de las diferentes bibliotecas estudiadas que permiten el descubrimiento de la red, se llegó a la conclusión de seleccionar python-nmap por ser la más completa entre todas las herramientas antes expuestas. Es una biblioteca libre que brinda seguridad al servidor, detecta los hosts activos en una subred, pero sin dudas su función más importante es escanear los puertos de la computadora o computadoras en cuestión y establecer si un puerto está abierto o cerrado. No se decidió utilizar Scapy porque las pruebas de integración realizadas con el sistema GRHS no fueron satisfactorias, pues no devolvía correctamente la cantidad de hosts activos de una subred. De la biblioteca Zenmap se estudió la forma en que realiza el escaneo de la red, además es una aplicación para utilizar de forma gráfica Nmap, por este motivo resultó más fácil integrar python-nmap por ser una biblioteca de python

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

que usa Nmap, además utiliza el mismo lenguaje de programación definido en el proyecto GRHS.

## **1.5 Protocolos de referencia en la investigación**

A continuación se describen los protocolos de descubrimiento estudiados en la investigación.

### 1.5.1 Protocolo ARP

El protocolo ARP (Protocolo de resolución de direcciones) es un estándar TCP/IP necesario para resolver direcciones IP, que utiliza direcciones de control de acceso a medios empleadas por el hardware de LAN. ARP proporciona diferentes servicios de protocolo a hosts que se encuentran en la misma red física. Primeramente las direcciones de control de acceso a medios se obtienen mediante una solicitud de difusión de red en forma de la pregunta. Por otra parte, cuando se responde a una solicitud ARP, el remitente de la respuesta y el solicitante registran sus direcciones IP y de control de acceso a medios respectivos como una entrada en una tabla local, llamada la caché de ARP, para su uso posterior como referencia. (14)

### 1.5.2 Protocolo ICMP

El protocolo ICMP (Protocolo de mensajes de control de Internet) es un estándar TCP/IP. Con ICMP, los hosts y los enrutadores que utilizan la comunicación IP pueden informar de errores e intercambiar información de control y estado limitada. Los mensajes ICMP se suelen enviar automáticamente cuando un datagrama IP no es capaz de llegar a su destino, un enrutador IP no puede reenviar datagramas a la velocidad actual de transmisión o un enrutador IP redirige el host que realiza el envío para que utilice una ruta mejor para llegar al destino. (15) Este protocolo es el que utiliza nmap para el descubrimiento de la red de computadoras.



# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## 1.5.3 Protocolo RIP

El protocolo RIP (Routing Protocol Information o Protocolo de encaminamiento de información), es uno de los protocolos de encaminamiento interior más sencillos y con mayor uso. Es un protocolo de puerta de enlace interna utilizado por los routers, aunque también puede actuar en equipos para intercambiar información acerca de redes IP. (16) Utiliza el protocolo UDP (Protocolo de Datagrama de Usuario) para enviar sus mensajes a través del puerto 520. Es una implementación directa del encaminamiento vector-distancia, calcula el camino más corto hacia la red de destino utilizando dicho encaminamiento.

## 1.5.4 Protocolo OSPF

El protocolo OSPF (Open Shortest Path First o El camino más corto primero) es un protocolo de encaminamiento jerárquico de pasarela interior o IGP (Interior Gateway Protocol), que usa el algoritmo Dijkstra para calcular la ruta más corta posible. Además, construye una base de datos enlace-estado idéntica en todos los encaminadores de la zona. OSPF es probablemente el tipo de protocolo IGP más utilizado en redes grandes. Puede operar con seguridad usando MD5 (Message-Digest Algorithm 5) para autenticar a sus puntos antes de realizar nuevas rutas y antes de aceptar avisos de enlace-estado. (17)

El estudio de los protocolos de descubrimiento de equipos de interconexión utilizados mundialmente, confirmó el uso de python-nmap, que utiliza internamente ICMP para el descubrimiento.

## **1.5 Metodología, lenguaje de programación, herramientas y tecnologías a utilizar**

### 1.5.1 Metodología de desarrollo

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería. (18)

## 1.5.1.1 Programación Extrema (eXtremeProgramming, XP)

La metodología de desarrollo XP está centrada en potenciar las relaciones entre personas como clave para obtener éxito en el desarrollo del software, la misma promueve el trabajo en equipo, la comunicación con el cliente, proporcionando a cada uno de estos un buen clima de trabajo y un aspecto muy importante es que se preocupa por el aprendizaje de los programadores.

Es una metodología ágil que surge como una nueva forma o guía para realizar proyectos de software, esencialmente está basada en la simplicidad y agilidad, también en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. La misma se ha definido para proyectos con requisitos muy cambiantes e imprecisos y que exista un alto riesgo técnico.

El ciclo de vida de XP consiste en 4 fases fundamentales:

- ✓ **Exploración:** Los clientes definen las historias de usuario, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.
- ✓ **Planificación de la Entrega:** Se establece la prioridad de cada historia de usuario, se estiman los esfuerzos que requieren cada una de las tareas al igual que el tiempo de duración de cada una de ellas.
- ✓ **Iteraciones:** Se incluyen varias iteraciones sobre el sistema antes de ser entregado, las mismas no deben exceder las 3 semanas.
- ✓ **Producción:** En esta fase se realizan pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. De igual forma se toman decisiones sobre añadir nuevas características al sistema.

(19)

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Se decidió utilizar XP porque es un módulo pequeño y de poca duración, desarrollado por un dúo de programadores, el cliente forma parte del equipo de desarrollo, genera poca documentación, la implementación se divide en iteraciones cortas, los requisitos pueden cambiar en cualquier momento y en cada iteración se le entrega al cliente una versión funcional del módulo.

## 1.5.2 Lenguaje de programación

Un lenguaje de programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. (20) Para el desarrollo del módulo de descubrimiento automático de una red de computadoras serán utilizadas las mismas tecnologías que utiliza el sistema GRHS, empleando como lenguaje del lado del servidor Python 2.7 y como lenguajes del lado del cliente HTML5 y JavaScript. A continuación se muestra una breve descripción de cada uno de ellos.

### 1.5.2.1 Python 2.7

Python es un lenguaje de programación multipropósito de alto nivel. Su filosofía de diseño enfatiza la productividad del programador y la legibilidad del código. Tiene un núcleo sintáctico minimalista con unos pocos comandos básicos y simple semántica, pero además tiene una enorme y variada librería estándar, que incluye una Interfaz de Programación de Aplicaciones (API) para muchas de las funciones en el nivel del sistema operativo (SO). (21) Se decidió utilizar python porque GRHS está implementado sobre este lenguaje, además el módulo a implementar se va a integrar a este sistema y debe ser desarrollado bajo sus mismas tecnologías. En el centro de TLM se definió este lenguaje de programación para sus productos, lo que beneficia que cualquier persona pueda darle mantenimiento al sistema o programar nuevas funcionalidades cuando sea necesario.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## 1.5.2.2 HTML5

HTML5 es la última evolución de la norma que define el lenguaje de marcas de hipertexto (HTML por sus siglas en inglés). Se trata de una nueva versión del lenguaje HTML, con nuevos elementos, atributos y comportamientos. Diseñado para ser utilizado por todo tipo de desarrolladores. Describe la estructura y el contenido semántico de un documento web. (22) Se utilizará el lenguaje HTML5 para el diseño de las interfaces del módulo, siendo este el utilizado en el sistema GRHS actualmente.

## 1.5.2.3 JavaScript

JavaScript es un lenguaje ligero e interpretado, orientado a objetos, más conocido como el lenguaje de script para páginas web. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional e imperativa. (23) Este lenguaje es utilizado para realizar acciones o eventos de manera que permita visualizar el descubrimiento de cada subred.

## 1.5.3 Entorno de Desarrollo Integrado

Un IDE (Entorno de Desarrollo Integrado o Entorno de Desarrollo Interactivo) es una aplicación de software que proporciona servicios integrales para los programadores informáticos para el desarrollo de software. Una IDE normalmente consiste en un editor de código fuente que permite construir herramientas de automatización. (24)

### 1.5.3.1 Eclipse Helios 3.6

Es un IDE de desarrollo que cuenta con varias Interfaces de Programación de Aplicaciones (API). Permite programar en diferentes lenguajes, utilizando python para el desarrollo del módulo.

## 1.5.4 Framework de Desarrollo

Un framework se puede definir como un conjunto de componentes que estructuran un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. (25) Se

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

definieron como framework a utilizar en el desarrollo del módulo de descubrimiento automático de una red de computadoras a Django 1.4.5 y JQuery 1.10. A continuación se describen cada uno de ellos.

## **1.5.4.1 Django 1.4.5**

Django es un framework de desarrollo web sobre Python que permite desarrollar rápidamente aplicaciones web, incluye el paradigma Modelo Vista Template. Se centra en automatizar todo lo posible y se adhiere al principio DRY (Don't Repeat Yourself). (26) Es el framework de desarrollo que se utilizará para la implementación del módulo por ser actualmente el que se emplea en el sistema GRHS.

## **1.5.4.2 JQuery 1.10**

Jquery es una biblioteca JavaScript rápida, pequeña y rica en funciones. Hace las cosas como documento HTML recorrido y la manipulación, el control de eventos, animación y Ajax mucho más simple con un API fácil de usar que funciona a través de una multitud de navegadores. Con una combinación de flexibilidad y extensibilidad. (27)

## **1.5.4 Sistema Gestor de Bases de Datos**

Un Sistema Gestor de Bases de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. (28) Tiene como objetivo brindar una interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de tres lenguajes: lenguaje de definición de datos, lenguaje de manipulación de datos y lenguaje de consulta.

## **1.5.4.1 PostgreSQL 9.1**

PostgreSQL es la base de datos relacional de código abierto más avanzada del mundo. Distribuida bajo licencia BSD (Berkeley Software Distribution por sus siglas en inglés) y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (29) Sus características técnicas la hacen una de las bases de datos más potentes del mercado. Además de todas las características antes mencionadas, es la que utiliza el sistema GRHS en la actualidad.

## 1.5.5 Herramienta CASE

La herramienta CASE (Computer Aided Software Engineering por sus siglas en inglés) permite Modelar los Procesos de Negocios de las organizaciones. Permite mejorar el diseño de los sistemas, elaborar los requerimientos de los usuarios y garantiza la eficiencia.

### **1.5.5.1 Visual Paradigm 8.0**

Visual Paradigm for UML es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Tiene como ventajas que facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta. Permite controlar que el modelado con UML sea correcto. (30) Esta herramienta es utilizada para modelar las tablas de la BD generadas por el módulo y para el modelado del proceso de negocio.

## 1.5.6 Lenguaje de Modelado BPMN

Business Process Modeling Notation (BPMN por sus siglas en inglés) es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. (31) Esta notación se utiliza en el modelado del proceso de negocio del módulo a desarrollar.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## 1.6 Conclusiones parciales

Con la realización de este capítulo se dejaron fundamentadas las definiciones relacionadas con la investigación que ayudaron a un mejor entendimiento de la situación planteada. De los sistemas similares estudiados se determinó que no pueden ser integrados a la plataforma GRHS porque sus funcionalidades no cumplen con las necesidades actuales del sistema. Luego de realizado un estudio de las bibliotecas que realizan descubrimiento de la red se decidió utilizar para el desarrollo del módulo a python-nmap por las características que posee, pues permite solucionar de forma satisfactoria las deficiencias del sistema. Se definieron las herramientas fundamentales para el descubrimiento de la red y se seleccionó XP como la metodología de desarrollo a utilizar. Se decidió utilizar python 2.7 como lenguaje de programación, Eclipse Helios 3.6 como IDE de desarrollo y el framework de desarrollo web Django 1.4.5, por ser las tecnologías en las que se desarrolló el sistema GRHS. Además como sistema gestor de bases de datos se empleó PostgreSQL 9.1 y la notación BPMN para el modelado del proceso de negocio del módulo a implementar.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Capítulo 2: Exploración y Planificación

### 2.1 Introducción

En el presente capítulo se presenta la propuesta de solución del módulo a desarrollar y se enfatizan en las características esenciales del mismo. Además, se hace referencia a las fases de Exploración y Planificación propias de la metodología de desarrollo utilizada y se plasman las Historias de Usuarios (HU) para lograr un mejor entendimiento de la solución a implementar.

### 2.2 Propuesta de solución

Se propone para dar solución al objetivo general planteado, primeramente la obtención de un listado de las subredes, partiendo de la dirección IP y la máscara de red enviados por el cliente de GRHS. Facilitando de esta forma dos opciones para realizar el escaneo de la subred.

1. El primer tipo de escaneo permitirá la selección de una sola subred y el descubrimiento de la misma, informando al usuario cuando el sistema está escaneando o no esa subred y una vez finalizado dicho escaneo.
2. El otro tipo de escaneo será implementado de manera automática y programada, escaneando todas las subredes almacenadas en la base de datos, contará con un servicio que podrá ser activado y desactivado y se le podrá configurar la cantidad de horas en que se desea que se realice el escaneo de todas las subredes.

A continuación se muestra una imagen donde podrá observarse el funcionamiento del módulo.



## Descubrimiento automático de la red de computadoras para el sistema GRHS.

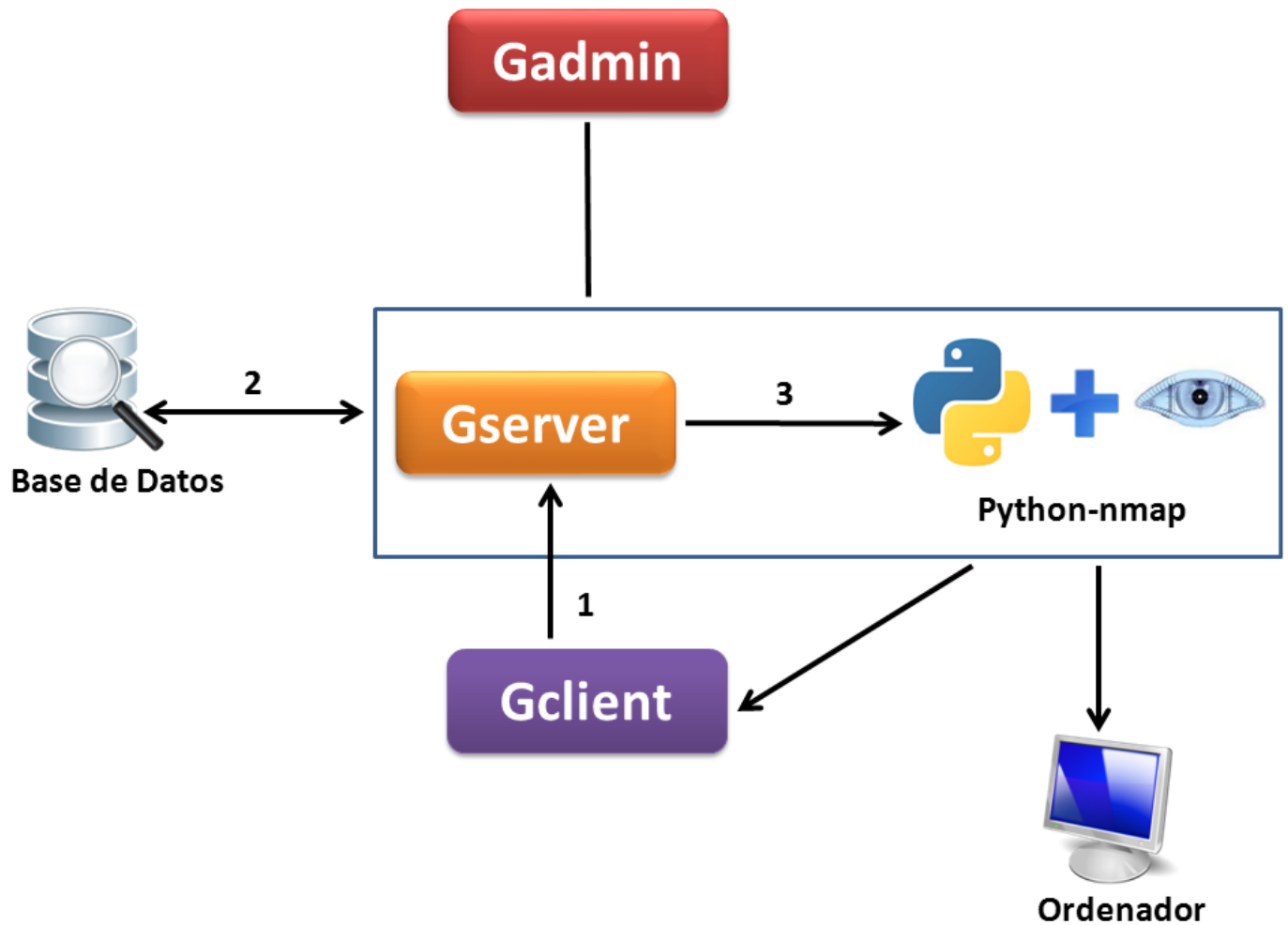


Ilustración 1. Propuesta de solución del módulo de descubrimiento automático de la red de computadoras.

Primeramente el cliente (Gclient) le envía los datos al servidor (Gserver), entre los que se encuentran la dirección IP y su máscara. El servidor determina la subred partiendo de estos dos parámetros recibidos y la almacena en base de datos. El servidor consulta la base de datos para obtener el listado de todas las subredes y a través de python-nmap realizar el escaneo de la misma, interactuando con todos los ordenadores de dicha subred, descubriendo si tienen o no el cliente de GRHS instalado.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Gadmin se encargará de mostrar los resultados del escaneo para que sean visualizados por el administrador. La aplicación contará con un registro de trazas, donde se almacenarán los datos provenientes del escaneo de las subredes, así como cualquier error ocurrido. Toda esta información se mostrará al usuario en una tabla paginada.

Como se explicó anteriormente, el módulo tiene dos formas de realizar el escaneo, la automática será utilizando programación multihilo para que el mismo pueda ejecutarse como un proceso independiente de la aplicación, el cual detectará las estaciones de trabajo que están encendidas en ese momento a través de la biblioteca python-nmap. Para comprobar si está instalado o no el cliente se reutilizará la biblioteca python-request que permite realizar peticiones http entre cliente-servidor.

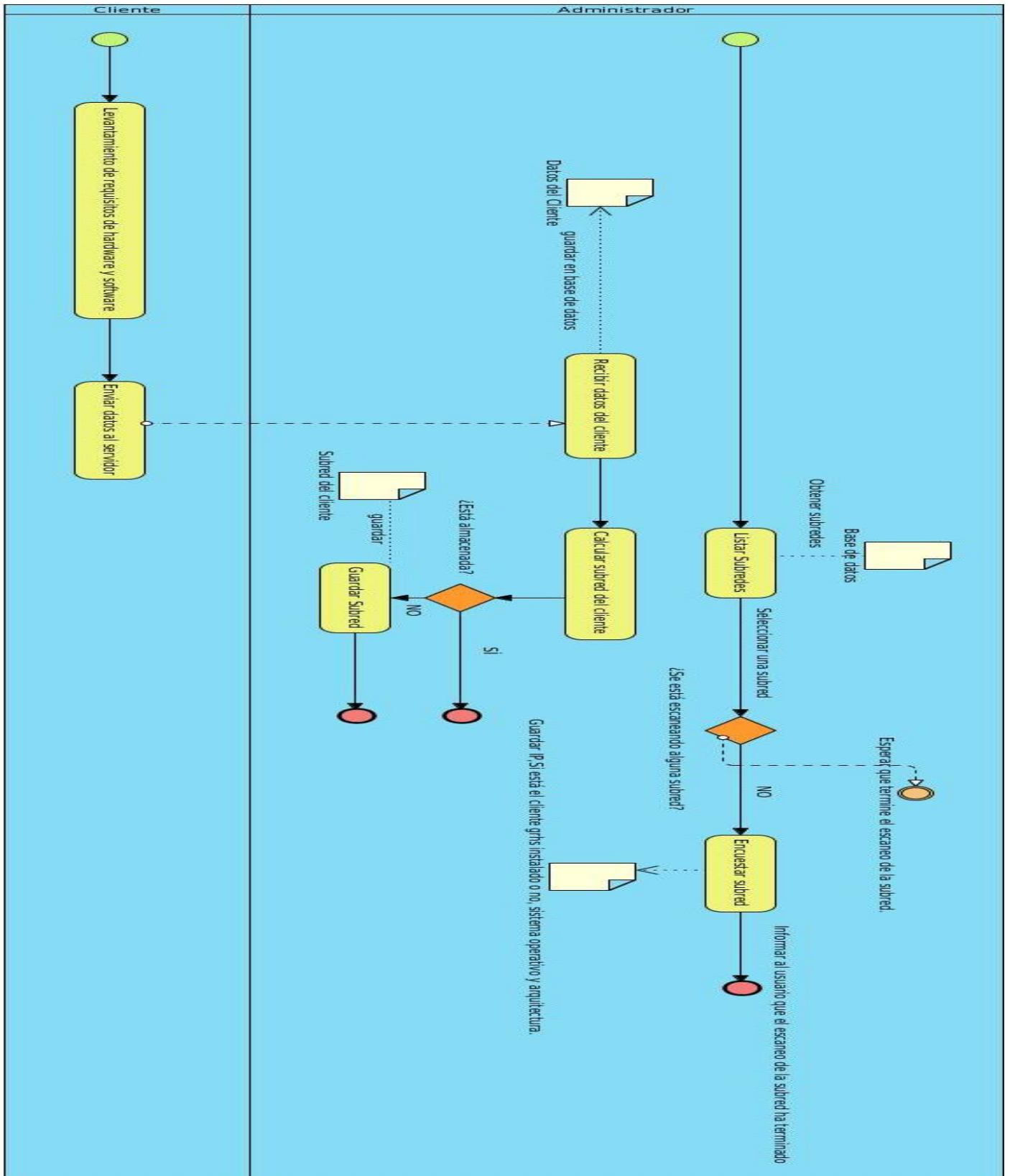
De las computadoras encendidas por cada subred se registrarán su identificador, su dirección IP, si tiene el cliente de GRHS instalado o no, el puerto por el que está ejecutándose, la subred a la que pertenece, la fecha en que fue encuestada la estación de trabajo y su localización. Los datos obtenidos se almacenarán en base de datos, permitiendo la visualización de los mismos, el filtrado de los resultados por cada uno de los datos almacenados y generar un documento en formato excel. Estos resultados se actualizarán siempre que se realice un escaneo, al mostrarlos se visualizarán los que se escanean en ese momento y los anteriores serán almacenados en un registro de trazas.

También serán mostradas las estadísticas generales del escaneo, la cantidad de subredes que están registradas en el sistema, la cantidad de computadoras encendidas que fueron encontradas y de estas las que no poseen el cliente de GRHS instalado.

## **2.3 Modelado del proceso de negocio**

A continuación se muestra el modelado del proceso de negocio del módulo de descubrimiento automático de la red de computadoras.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.



# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Ilustración 2. Modelado del proceso de negocio del módulo de descubrimiento automático de la red de computadoras.

Los usuarios deben estar autenticados previamente en el sistema y de esta manera los clientes de GRHS envían los datos de cada uno de ellos y su identificador al servidor. A partir de la información obtenida se genera un listado de subredes, donde se verifica si la subred enviada ya está almacenada y en caso contrario se agrega a la lista. En este proceso el administrador es el encargado de seleccionar cada que tiempo desea realizar el inventario y así conocer que estaciones de trabajo tienen o no el cliente instalado.

## 2.4 Características del sistema

Representan los requisitos no funcionales del módulo a implementar, en las categorías de usabilidad, seguridad, disponibilidad, legales, diseño e implementación, hardware y software. A continuación se listan los requisitos identificados para el desarrollo de la solución:

### **Requisitos de usabilidad**

- ✓ Los usuarios deben poseer conocimientos básicos referentes al uso de una computadora. Además deben tener conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.

### **Requisitos de seguridad**

- ✓ El módulo está desarrollado en el marco de trabajo Django, el cual brinda una arquitectura MTV y cumple con los estándares de seguridad establecidos en el sistema GRHS para de esta manera garantizar el acceso únicamente a las personas autorizadas.

### **Requisitos de disponibilidad**

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

- ✓ El módulo a desarrollar que será integrado al sistema GRHS debe estar disponible en todo momento para las personas autorizadas que necesiten acceder y manejar la información que contiene dicho sistema.

## **Requisitos legales:**

- ✓ La Universidad de las Ciencias Informáticas poseerá la propiedad intelectual sobre el producto final.

## **Restricciones del diseño y la implementación:**

- ✓ El módulo debe ser implementado utilizando el lenguaje de programación Python 2.7. Además las herramientas a utilizar en la codificación serán Eclipse Helios 3.6, el framework Django 1.4.5 y la biblioteca python-nmap para el desarrollo del módulo.

## **Requisitos de hardware**

- ✓ La computadora donde va a estar el servidor de GRHS debe tener un microprocesador de 2 GHz, memoria 1 GB de RAM y de disco duro 500 GB.

## **Requisitos de software**

- ✓ Las computadoras donde sea ejecutado el sistema, tanto cliente como servidor, deben tener instalado Firefox 3.X o superior. El servidor de base de datos debe tener instalado PostgreSQL en su versión 9.1.

## **2.5 Exploración**

La exploración es la primera fase definida por la metodología XP. Es aquí donde el cliente plantea a grandes rasgos lo que necesita a través de las HU, lo que facilita a los programadores estimar el tiempo de desarrollo. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## 2.5.1 Historias de Usuarios

Las historias de usuarios son la técnica utilizada en la metodología XP, donde se representa una breve descripción de los requisitos del software. El cliente debe usar una terminología sencilla y sin lenguaje técnico en el momento de detallar cada una de las actividades que realizará el sistema.

**HU1.** Adicionar automáticamente una subred.

**HU2.** Descubrir computadoras encendidas por subred.

**HU3.** Verificar si el cliente está instalado en cada computadora.

**HU4.** Mostrar características esenciales de las computadoras.

**HU5.** Configurar automáticamente el escaneo de la red.

**HU6.** Exportar a formato (Excel).

**HU7.** Mostrar estadísticas generales.

Se definen como puntos reales el tiempo de duración real de cada HU, estos puntos se definieron luego de implementada cada una de las funcionalidades del módulo.

Historia de Usuario	
<b>Número:</b> HU1	<b>Nombre Historia de Usuario:</b> Adicionar automáticamente una subred.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1.8
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez René Juan Pérez de Corcho Hernández	

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

<b>Descripción:</b> Primeramente el cliente se comunica con el servidor, enviándole su dirección IP y la máscara de red. El servidor verifica si la subred a la que pertenece ese IP está o no almacenada, en caso de no estarlo la almacena automáticamente en el sistema.
<b>Observaciones.</b>
<b>Prototipo de Interfaz:</b> No aplica

Tabla 1. HU1. Adicionar automáticamente una subred.

Historia de Usuario	
<b>Número:</b> HU2	<b>Nombre de Historia de Usuario:</b> Descubrir computadoras encendidas por subred.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez René Juan Pérez de Corcho Hernández	
<b>Descripción:</b> El administrador primeramente selecciona la subred que necesita inventariar y después da clic en la imagen que realiza el escaneo. De cada subred se muestra su localización.	
<b>Observaciones.</b> El administrador debe seleccionar solo una subred para escanear, en caso de seleccionar varias el sistema muestra un mensaje.	
<b>Prototipo de Interfaz:</b>	

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

The screenshot shows the XILEMA GRHS web interface. The main content area displays a table titled 'Escanear' with two columns: 'Subred' and 'Localización'. The table lists 10 subnets and their corresponding locations. Below the table, it indicates 'Mostrando 1 - 10 de 166' and 'Página 1 de 17'.

Subred	Localización
10.51.16	Doc1-Fac2-TLM-Lab301
10.51.18	Doc1-Fac2-TLM-Lab303
10.51.19	Doc1-Fac2-TLM-Lab304
10.51.20	Doc1-Fac2-TLM-Lab305
10.51.60	Doc1-Fac2-TLM-Subdir
10.51.30	Doc1-Fac2-TLM-Dir
10.31.30	Doc1-Fac2-Oficinas
10.51.17	Doc1-Fac2-TLM-Lab302
10.31.62	Doc1-Fac2-TLM-SIAI
10.53.9	Doc3-VRT-DIN-Lab205

Tabla 2. HU2. Descubrir computadoras encendidas por subred.

Historia de Usuario	
<b>Número:</b> HU3	<b>Nombre Historia de Usuario:</b> Verificar si el cliente está instalado en cada computadora.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 0.8
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez	



## Descubrimiento automático de la red de computadoras para el sistema GRHS.

René Juan Pérez de Corcho Hernández
<b>Descripción:</b> El servidor encuesta a cada una de las estaciones de trabajo, recibiendo una respuesta para saber si tienen el cliente de GRHS instalado o no.
<b>Observaciones.</b>
<b>Prototipo de Interfaz:</b> No aplica

Tabla 3. HU3. Verificar si el cliente está instalado en cada computadora.

Historia de Usuario	
<b>Número:</b> HU4	<b>Nombre Historia de Usuario:</b> Mostrar características esenciales de las computadoras.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1.5
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez  René Juan Pérez de Corcho Hernández	
<b>Descripción:</b> De las estaciones de trabajo que se encuentren encendidas, se muestra de cada una de ellas el identificador, su dirección IP, si tiene o no el cliente de GRHS instalado, el puerto, la subred a la que pertenece, la fecha en que fue realizado el escaneo y la localización.	
<b>Observaciones:</b> El administrador debe estar autenticado en el sistema para poder acceder a la información obtenida de cada uno de los escaneos.	

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

**Prototipo de Interfaz:**

The screenshot shows the GRHS web interface. At the top, there is a navigation menu with options: Sistema, Agentes, Inventario, Incidencias, Acción, Trazas, Configuración, and Descubrir. The main content area displays a table of results under the heading 'Resultados'. The table has the following columns: Identificador, Dirección IP, Gclient, Puerto, Subred, Fecha, and Localización. The data rows are as follows:

Identificador	Dirección IP	Gclient	Puerto	Subred	Fecha	Localización
MB0092803	10.51.30.242	NO	8418	10.51.30	21/05/2015	Doc1-Fac2-TLM-Dir
AF001665	10.51.30.244	SI	8418	10.51.30	21/05/2015	Doc1-Fac2-TLM-Dir
	10.51.30.249	SI	8418	10.51.30	21/05/2015	Doc1-Fac2-TLM-Dir
	10.51.30.250	SI	8418	10.51.30	21/05/2015	Doc1-Fac2-TLM-Dir
MB0101456	10.51.30.252	NO	8418	10.51.30	21/05/2015	Doc1-Fac2-TLM-Dir
MB0194910	10.51.30.253	NO	8418	10.51.30	21/05/2015	Doc1-Fac2-TLM-Dir
	10.51.30.254	NO	8418	10.51.30	21/05/2015	Doc1-Fac2-TLM-Dir
	10.31.30.31	NO	8418	10.31.30	21/05/2015	Doc1-Fac2-Oficinas
	10.31.30.69	SI	8418	10.31.30	21/05/2015	Doc1-Fac2-Oficinas
	10.31.30.73	NO	8418	10.31.30	21/05/2015	Doc1-Fac2-Oficinas

Mostrando 61 - 70 de 393

Unidad de las Ciencias Informáticas. XILEMA, Gestor de recursos de hardware y software (GRHS). 2014 Versión 1.0.0.0

Tabla 4. HU4. Mostrar características esenciales de las computadoras.

Historia de Usuario	
<b>Número:</b> HU5	<b>Nombre Historia de Usuario:</b> Configurar automáticamente el escaneo de la red.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Baja (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 0.6

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

**Programador(es) responsable(s):** Arlety de la Caridad Sebastian Martínez

René Juan Pérez de Corcho Hernández

**Descripción:** El administrador selecciona la opción de configurar de forma automática, adicionando cada que tiempo en horas debe realizarse el escaneo. El sistema para escanear automáticamente debe verificar si está activo o no el escaneo y en caso de estarlo verificar si está escaneando o no.

**Observaciones:** Para realizar esta configuración primeramente el administrador debe estar autenticado en el sistema y de esta forma poder decidir cada que tiempo desea que el sistema comience a escanear.

## Prototipo de Interfaz:

The screenshot shows the GRHS web interface. The top navigation bar includes 'Sistema', 'Agentes', 'Inventario', 'Incidencias', 'Acción', 'Trazas', 'Configuración', and 'Descubrir'. The user is logged in as 'root'. The main content area is titled 'Configuración' and displays a table with the following data:

Horas	Activo	Escaneando
12	✘	✘

At the bottom of the table, it says 'Mostrando 1 - 1 de 1'. The footer of the page reads: 'Universidad de las Ciencias Informáticas, XILEMA, Gestor de recursos de hardware y software (GRHS), 2014 Versión 1.0.0.0'.

Tabla 5. HU5. Configurar automáticamente el escaneo de la red.

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

Historia de Usuario	
<b>Número:</b> HU6	<b>Nombre Historia de Usuario:</b> Exportar a formato (Excel).
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Baja (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez René Juan Pérez de Corcho Hernández	
<b>Descripción:</b> El administrador puede seleccionar la opción de exportar a formato Excel la información obtenida después de realizado el escaneo, agrupándolos por localización.	
<b>Observaciones:</b> El administrador debe seleccionar la opción de exportar a formato Excel una vez terminado el escaneo.	
<b>Prototipo de Interfaz:</b>	

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

	A	B	C	D	E	F	G	H	I
1	IP Address	Gclient	Port	Subnet	Date				
2	10.31.30.31	NO	8418	10.31.30	21/05/2015				
3	10.31.30.69	SI	8418	10.31.30	21/05/2015				
4	10.31.30.73	NO	8418	10.31.30	21/05/2015				
5	10.31.30.79	NO	8418	10.31.30	21/05/2015				
6	10.31.30.80	NO	8418	10.31.30	21/05/2015				
7	10.31.30.81	NO	8418	10.31.30	21/05/2015				
8	10.31.30.85	NO	8418	10.31.30	21/05/2015				
9	10.31.30.97	NO	8418	10.31.30	21/05/2015				
10	10.31.30.98	NO	8418	10.31.30	21/05/2015				
11	10.31.30.100	NO	8418	10.31.30	21/05/2015				
12	10.31.30.112	SI	8418	10.31.30	21/05/2015				
13	10.31.30.115	NO	8418	10.31.30	21/05/2015				
14	10.31.30.125	SI	8418	10.31.30	21/05/2015				
15	10.31.30.150	NO	8418	10.31.30	21/05/2015				
16	10.31.30.160	NO	8418	10.31.30	21/05/2015				
17	10.31.30.162	NO	8418	10.31.30	21/05/2015				
18	10.31.30.164	NO	8418	10.31.30	21/05/2015				
19	10.31.30.170	NO	8418	10.31.30	21/05/2015				
20	10.31.30.171	SI	8418	10.31.30	21/05/2015				
21	10.31.30.180	NO	8418	10.31.30	21/05/2015				
22	10.31.30.181	NO	8418	10.31.30	21/05/2015				
23	10.31.30.183	NO	8418	10.31.30	21/05/2015				
24	10.31.30.184	NO	8418	10.31.30	21/05/2015				
25	10.31.30.187	NO	8418	10.31.30	21/05/2015				
26	10.31.30.188	NO	8418	10.31.30	21/05/2015				
27	10.31.30.190	NO	8418	10.31.30	21/05/2015				
28	10.31.30.193	NO	8418	10.31.30	21/05/2015				
29	10.31.30.196	SI	8418	10.31.30	21/05/2015				
30	10.31.30.197	SI	8418	10.31.30	21/05/2015				
31	10.31.30.199	SI	8418	10.31.30	21/05/2015				
32	10.31.30.200	NO	8418	10.31.30	21/05/2015				
33	10.31.30.201	SI	8418	10.31.30	21/05/2015				
34	10.31.30.202	NO	8418	10.31.30	21/05/2015				

Tabla 6. HU6. Exportar a formato (Excel).

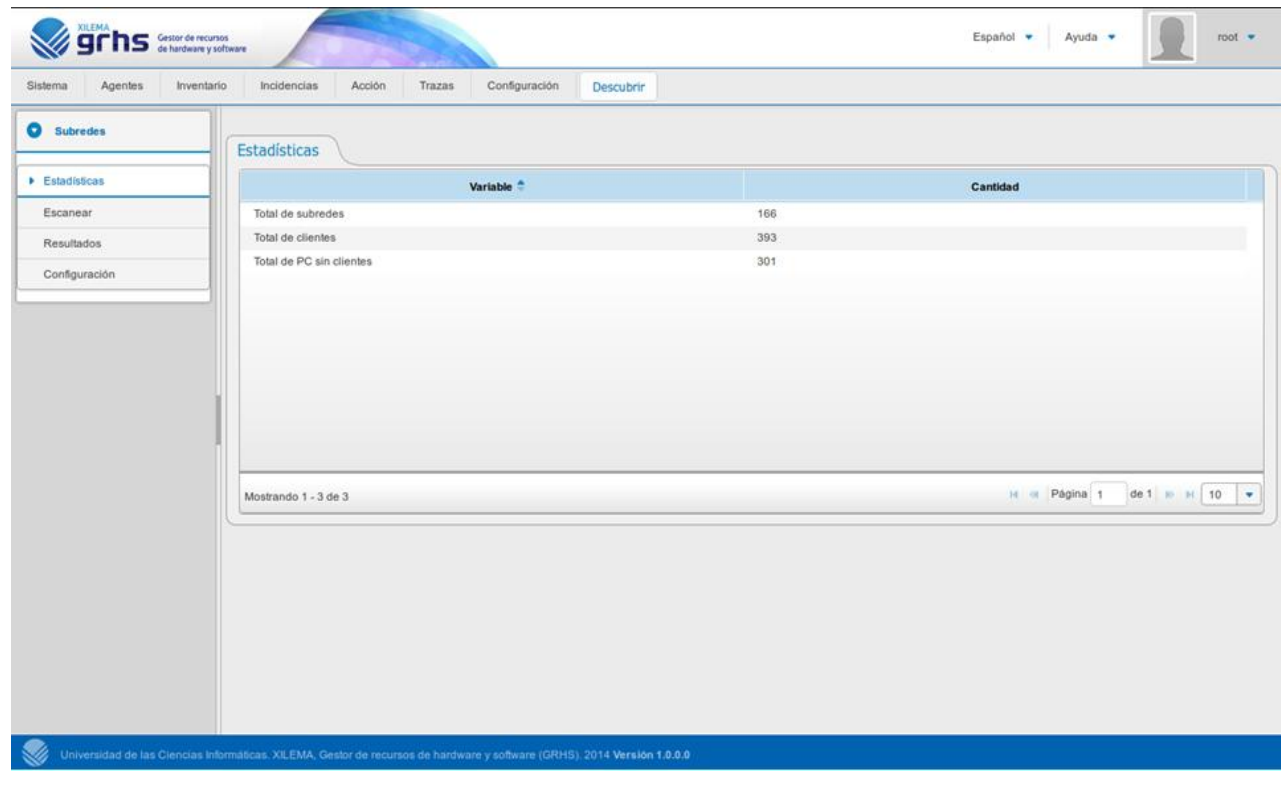
Historia de Usuario	
<b>Número:</b> HU6	<b>Nombre Historia de Usuario:</b> Mostrar estadísticas generales.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Baja (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 0,5
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez René Juárez Pérez de Corcho Hernández	

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

**Descripción:** Se muestra la cantidad total de subredes, la cantidad de computadoras encuestadas y el total de PC que no tienen el cliente GRHS instalado.

**Observaciones:** Primeramente para poder visualizar las estadísticas generales el administrador debe estar autenticado en el sistema y de esta forma acceder al menú Descubrir para poder acceder a cada una de las funcionalidades que brinda el modulo implementado.

## Prototipo de Interfaz:



The screenshot displays the GRHS web application interface. At the top, there is a header with the XILEMA logo and the text 'Gestor de recursos de hardware y software'. The user is logged in as 'root'. A navigation menu includes 'Sistema', 'Agentes', 'Inventario', 'Incidentes', 'Acción', 'Trazas', 'Configuración', and 'Descubrir'. The main content area is titled 'Estadísticas' and contains a table with the following data:

Variable	Cantidad
Total de subredes	166
Total de clientes	393
Total de PC sin clientes	301

At the bottom of the table, it indicates 'Mostrando 1 - 3 de 3' and 'Página 1 de 10'.

Tabla 7. HU7. Mostrar estadísticas generales.

## 2.6 Planificación

En la fase de la planificación de la metodología XP, el cliente prioriza cada una de las HU y los programadores realizan una estimación del esfuerzo por cada una de ellas.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## 2.6.1 Estimación del esfuerzo por HU

La siguiente tabla contiene la estimación del esfuerzo por cada HU según el orden en que serán realizadas. En esta estimación se incluye todo el esfuerzo asociado a la implementación de cada HU y se expresa utilizando como medida los puntos de estimación (punto máximo). Cada punto se considera una semana ideal de trabajo, donde se trabaja el tiempo que ha sido planificado sin interrupciones. En la tabla se muestran los puntos estimados por cada HU, donde al final se concluyó que las mismas se realizarán en un período máximo de 9 semanas de trabajo.

Historia de Usuario	Puntos estimados
Adicionar automáticamente una subred.	2
Descubrir computadoras encendidas por subred.	1
Verificar si el cliente está instalado en cada computadora.	1
Mostrar características esenciales de las computadoras.	2
Configurar automáticamente el escaneo de la red.	1
Exportar a formato (Excel).	1
Mostrar estadísticas generales.	1

Tabla 8. Estimación del esfuerzo por HU.

## 2.6.2 Plan de Iteraciones

Luego de identificar las HU y la estimación del esfuerzo por cada una de ellas, se procede a realizar el plan de iteraciones en el cual estarán comprendidas las HU según

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

el orden en que sean realizadas por cada iteración, además del total de semanas que durará cada una de estas. Teniendo en cuenta el riesgo de cada una de las HU, así como otros factores subjetivos, se decidió dividir el proyecto en tres iteraciones, detalladas a continuación:

**Iteración 1:** En esta iteración se implementarán las HU de prioridad Alta, funcionalidades que son de vital importancia para cubrir las necesidades con el cliente.

**Iteración 2:** En esta iteración se implementarán las HU de prioridad Media, funcionalidades que no son menos importantes que las mencionadas anteriormente pero por decisión del equipo se decidió realizarlas en otra iteración.

**Iteración 3:** En esta iteración se implementarán las HU de prioridad Baja.

La tabla que se muestra a continuación se encarga de reflejar el orden de las HU por cada iteración, así como la duración total en semanas de las mismas.

Iteración	Orden de las HU por iteraciones	Duración total en semanas
1	<ul style="list-style-type: none"><li>✓ Adicionar automáticamente una subred.</li><li>✓ Descubrir computadoras encendidas por subred.</li></ul>	3
2	<ul style="list-style-type: none"><li>✓ Verificar si el cliente está instalado en cada computadora.</li><li>✓ Mostrar características esenciales de las computadoras.</li></ul>	3
3	<ul style="list-style-type: none"><li>✓ Configurar automáticamente el escaneo de la red.</li></ul>	3



# Descubrimiento automático de la red de computadoras para el sistema GRHS.

	<ul style="list-style-type: none"><li>✓ Exportar a formato (Excel).</li><li>✓ Mostrar estadísticas generales.</li></ul>	
--	---	--

Tabla 9. Plan de iteraciones.

## 2.6.3 Plan de Entregas

El plan de entregas define la fecha fin de cada iteración, este plan se obtiene de cada una de las reuniones realizadas entre el equipo de desarrollo y el cliente.

A continuación se muestra una tabla con las fechas aproximadas de entregas por cada iteración.

<b>Módulo</b>	<b>Final 1ra Iteración 03/04/2015</b>	<b>Final 2da Iteración 24/04/2015</b>	<b>Final 3ra Iteración 15/05/2015</b>
Descubrimiento automático de la red de computadoras para el sistema GRHS.	Módulo v0.1	Módulo v0.5	Módulo v1.0

Tabla 10. Plan de entregas.

## 2.7 Conclusiones parciales

En este capítulo se describió la propuesta del módulo a desarrollar, integrando la biblioteca python-nmap por ser la más adaptable al sistema GRHS y por la precisión de los resultados obtenidos. Además se identificaron las características que el sistema debe cumplir así como la descripción de cada una de las HU y la estimación del esfuerzo dedicado a la realización de cada una de ellas según el orden establecido de acuerdo a las necesidades del cliente. Con la obtención del plan de entregas se logró delimitar el ciclo de desarrollo del sistema.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Capítulo 3: Diseño e Implementación

### 3.1 Introducción

En este capítulo se describe la fase de implementación, propia de la metodología de desarrollo XP. Se detalla la arquitectura de software, así como los patrones de diseño utilizados en el desarrollo del mismo. Se definen las tarjetas Clase – Responsabilidad – Colaborador (CRC) como artefacto para el enfoque orientado a objetos. Además se exponen las tareas de ingeniería asociadas a cada HU, las cuales han sido utilizadas como base para la implementación del módulo. Se evalúa la calidad de la solución a través de las pruebas de software realizadas.

### 3.2 Arquitectura de software

La arquitectura de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño y seguridad) y servir como guía en el desarrollo. (32)

#### 3.2.1 Arquitectura Cliente-Servidor

La arquitectura Cliente/Servidor tiene como principio fundamental la agrupación de todo tipo de aplicaciones en dos grupos fundamentales, que sería básicamente dividir aquellas aplicaciones en las que se ofrecen distintos servicios específicos y funcionalidades acordes, mientras que por otro lado están aquellos programas que hacen uso de los servicios mencionados. (33)

A continuación se muestra dicha arquitectura para entender mejor su funcionamiento.

Descubrimiento automático de la red de computadoras para el sistema GRHS.

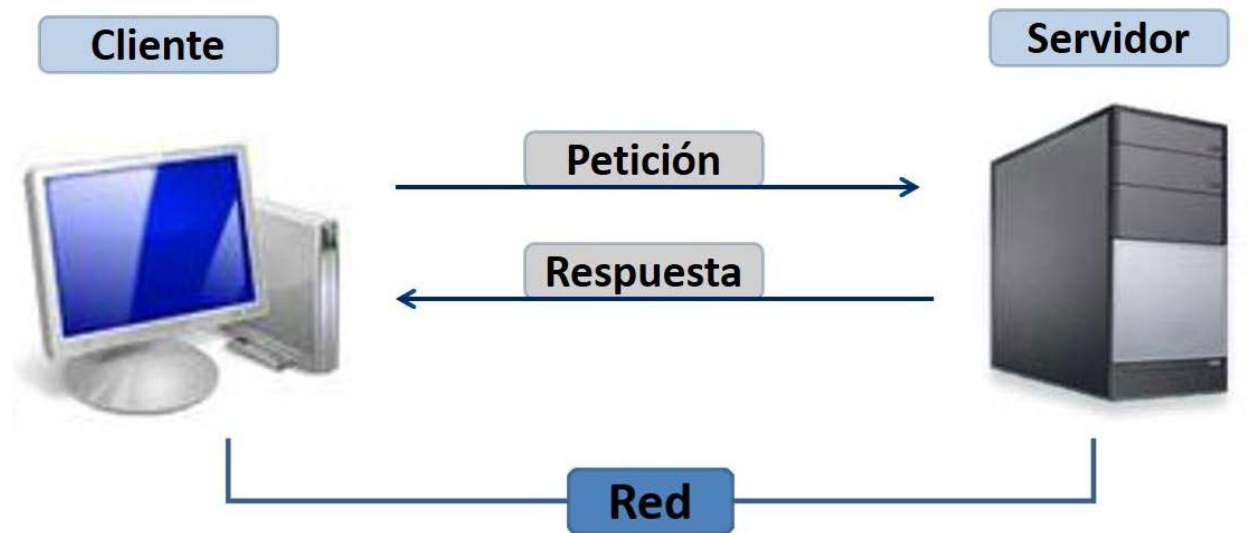


Ilustración 3. Arquitectura Cliente - Servidor.

Esta arquitectura ofrece ventajas de tipo organizativo debido a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita el diseño del sistema. La separación entre cliente y servidor en el proyecto GRHS es de tipo lógico, porque el servidor no se ejecuta necesariamente sobre una sola computadora ni es necesariamente un sólo programa.

En el servidor se encuentra instalado el Gserver y en el cliente Gclient, cumpliendo con la arquitectura antes mencionada.

### 3.3 Patrón Arquitectónico

Los patrones arquitectónicos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos. (34)

Representan un diseño organizativo para guiar el desarrollo del software. Para el desarrollo del módulo el patrón a seguir es el Modelo-Plantilla-Vista (MTV según sus siglas en inglés).

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## 3.3.1 Modelo Plantilla Vista

El framework Django es una de las tecnologías empleadas en el desarrollo del módulo, el cual usa una modificación de la arquitectura Modelo – Vista - Controlador (MVC), llamada Modelo – Plantilla – Vista (MTV) empleado en la implementación de la solución. En Django el Modelo continua siendo modelo, la vista se nombra Plantilla y el controlador se llama Vista.

**La Capa Modelo:** Se refiere a la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tienen y las relaciones entre ellos.

**La Capa Plantilla:** Se refiere a la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: cómo deberían ser mostradas sobre una página web u otro tipo de documento.

**La Capa Vista:** Se refiere a la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega en la plantilla apropiada. (35)

A continuación se muestra dicha relación para entender mejor su funcionamiento.

Descubrimiento automático de la red de computadoras para el sistema GRHS.

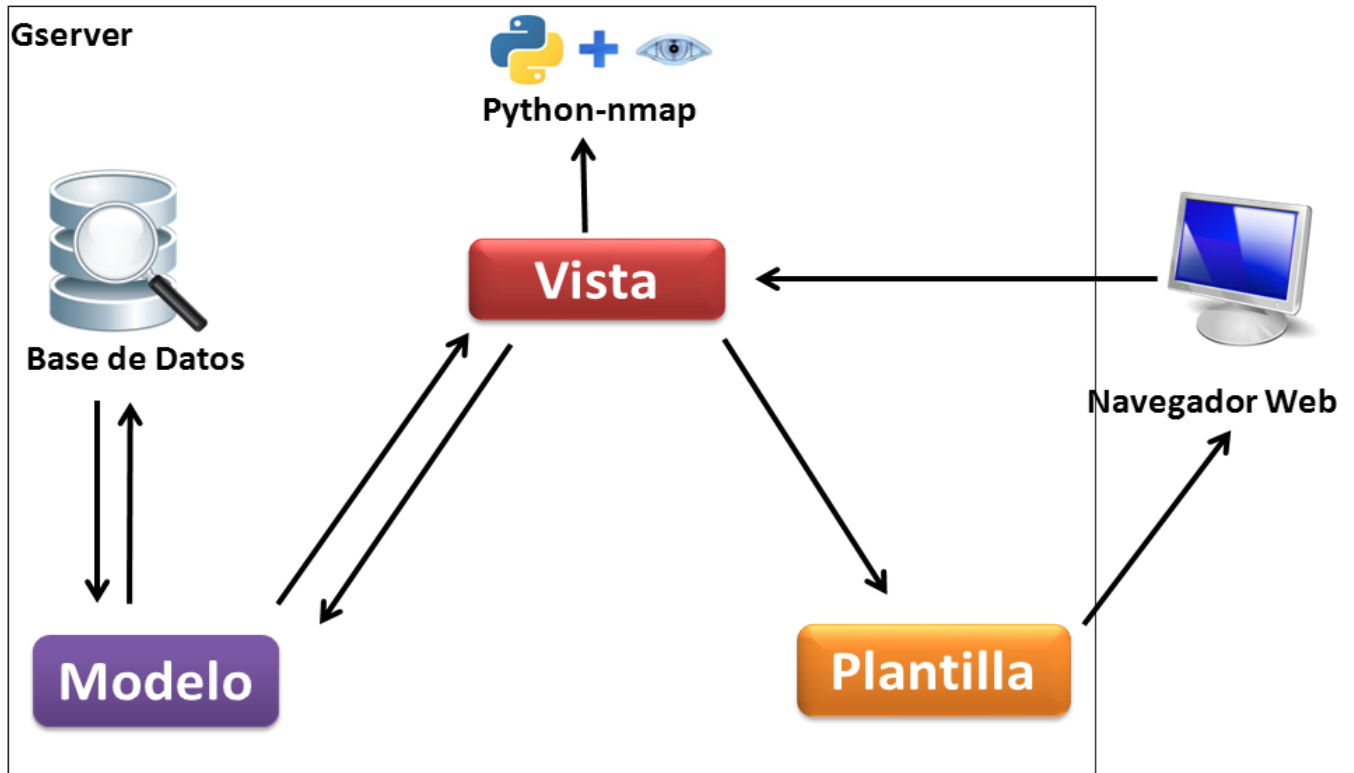


Ilustración 4. Patrón de arquitectura MTV.

Como parte del funcionamiento de este patrón, primeramente el navegador manda una solicitud a la vista, luego la vista interactúa con el modelo para obtener los datos, después hace una llamada a la plantilla y la plantilla se encarga de renderizar la respuesta a la solicitud del navegador. Este patrón se encuentra dentro de Gserver y en la capa Vista se implementó la biblioteca python-nmap, se reutilizó la biblioteca python-request y se desarrolló el escaneo automático y programando utilizando programación multihilo.

### 3.4 Representación de las Capas de la Arquitectura

A continuación se describen las clases situadas en cada una de las capas de la arquitectura del patrón MTV. Esta arquitectura se encuentra estructurada por 3 capas: Vista, Modelo y Plantilla.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## 3.4.1 Capa Vista

La vista se presenta mediante funciones python, tiene como objetivo determinar qué datos serán visualizados. A continuación se muestra una representación de las clases de la capa Vista.

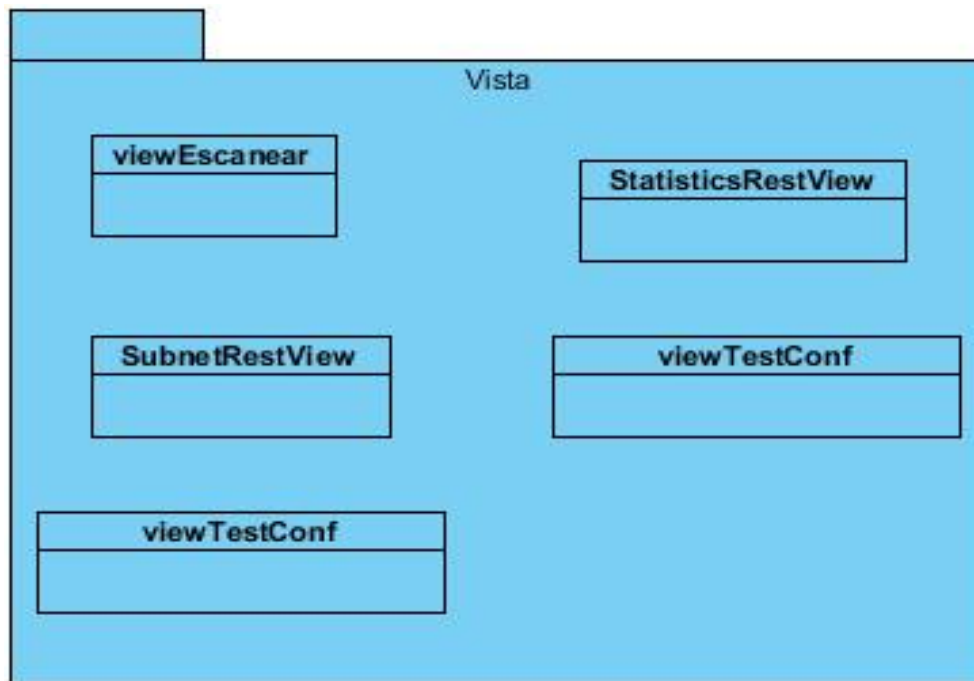


Ilustración 5. Capa Vista.

**viewEscanear:** Esta vista se encarga del escaneo de una subred, controlando el inicio y el fin de la misma y no permitiendo escaneos simultáneos de más de una subred.

**SubnetRestView:** Esta vista se encarga de buscar la cantidad de subredes y su localización.

**StatisticsRestView:** Esta vista se encarga de buscar la cantidad de subredes, la cantidad de computadoras inventariadas y la cantidad de estas últimas que no tienen el cliente de GRHS instalado.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

**ConfigurationRestView:** Esta vista se encarga de controlar la configuración del escaneo programado de todas las subredes, permitiendo activar y desactivar el servicio, facilitando al administrador poder introducir el tiempo en horas que él determina para realizar un nuevo escaneo e informando si el servidor está o no escaneando.

**viewTestConf:** Esta vista de encarga del escaneo de todas las subredes.

## 3.4.2 Capa Modelo

El modelo es el encargado de definir los datos almacenados, se encuentra en forma de clases. Cada dato almacenado se encuentra en una variable con determinados parámetros. Las clases modelo del módulo se encuentran declaradas en el archivo models.py. A continuación se muestra una representación de las clases de dicha capa.

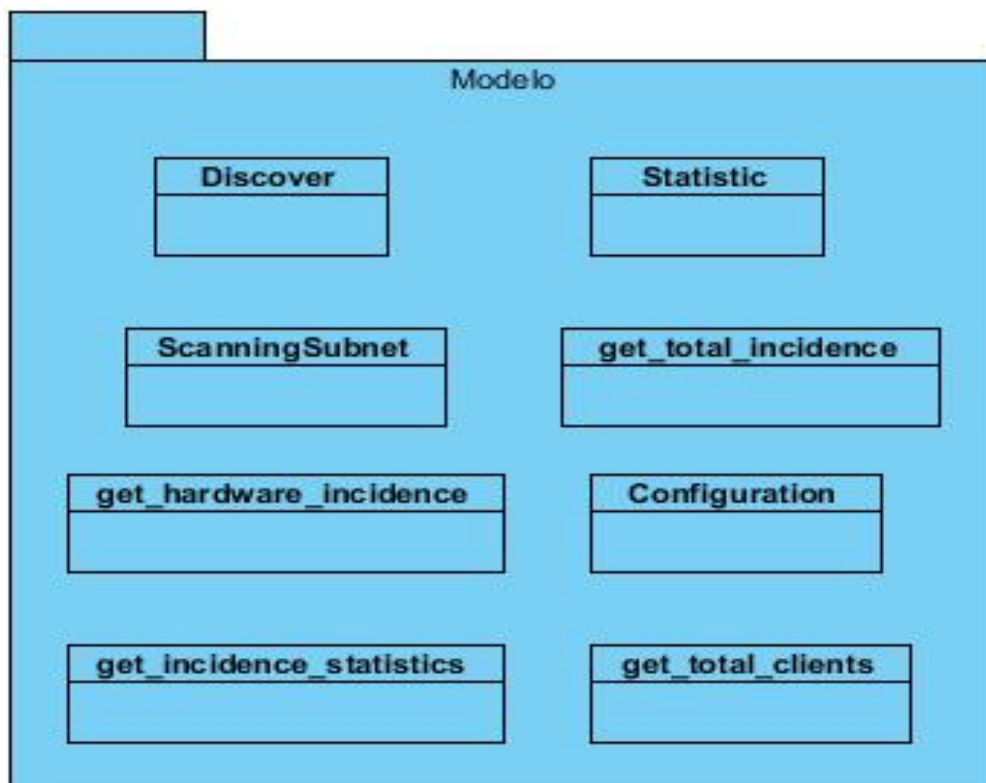


Ilustración 6. Capa Modelo.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

**Discover:** Modelo responsable de registrar, modificar, eliminar, buscar y obtener detalles de los resultados obtenidos por los escaneos de las subredes.

**ScanningSubnet:** Modelo responsable de registrar, modificar, eliminar, buscar y obtener detalles del escaneo de una subred.

**Statistic:** Modelo responsable de registrar, modificar, eliminar, buscar y obtener detalles de las estadísticas.

**Configuration:** Modelo responsable de registrar, modificar, eliminar, buscar y obtener detalles de la configuración del escaneo programado de todas las subredes.

**get\_incidence\_statistics:** Modelo responsable de buscar y obtener detalles de las estadísticas.

**get\_total\_incidence:** Modelo responsable de buscar y obtener detalles de las subredes.

**get\_total\_clients:** Modelo responsable de buscar y obtener detalles del total de computadoras escaneadas.

**get\_hardware\_incidence:** Modelo responsable de buscar y obtener detalles del total de computadoras escaneadas que no tienen el cliente instalado.

## 3.4.3 Capa Plantilla

La capa plantilla recibe los datos de la vista y luego los organiza para mostrarlos en el navegador web. Dichas plantillas están situadas en la carpeta Templates del módulo. A continuación se muestra una representación de la capa Plantilla.



Descubrimiento automático de la red de computadoras para el sistema GRHS.

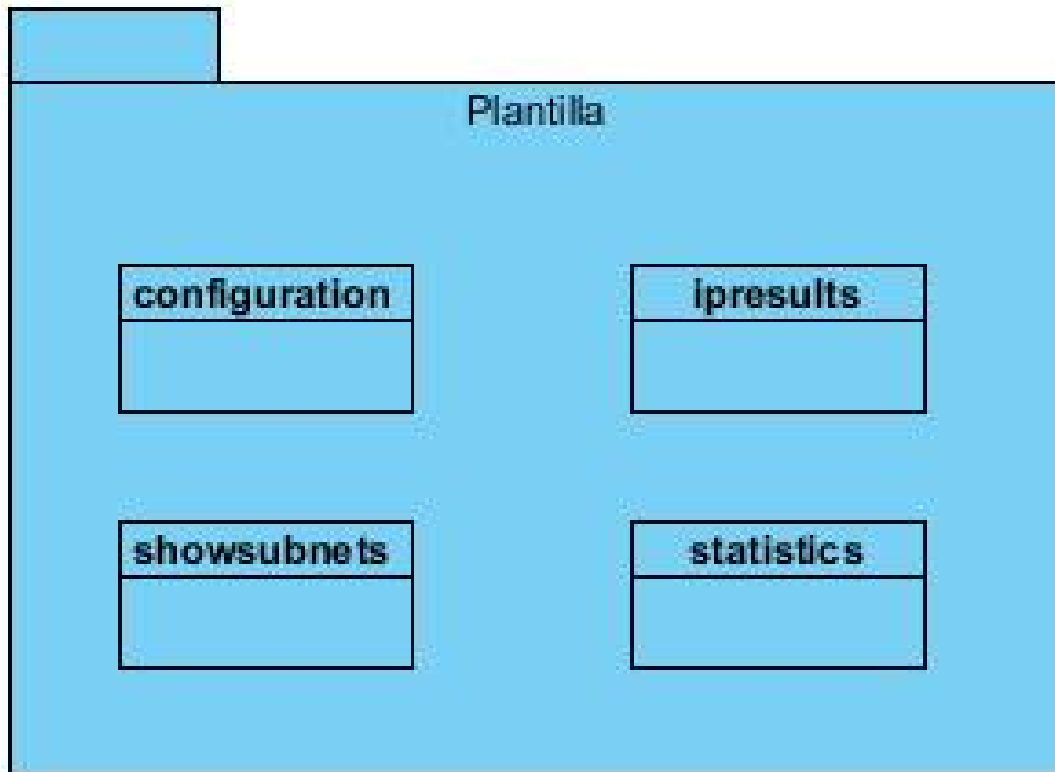


Ilustración 7. Capa Plantilla.

Cada uno de estos archivos define las plantillas HTML, empleando estilos CSS y lenguajes JavaScript y JQuery.

### 3.5 Patrones de Diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. (36)

#### 3.5.1 Patrón GRASP

Dentro de los patrones de diseño se encuentran los Patrones Generales de Asignación de Responsabilidades de Software (GRASP por sus siglas en inglés). Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

a objetos (37). A continuación se explican los patrones GRASP que fueron utilizados para el desarrollo del módulo.

## ✓ Patrón Experto:

Un experto es una clase que tiene toda la información necesaria para implementar una responsabilidad. Dentro de sus ventajas se encuentran:

- Encapsulamiento de la información.
- Distribución del comportamiento del manejo de la información. (38)

Con este patrón se desea que los objetos realicen tareas relacionadas con la información que poseen.

**Ejemplo:** Las clases Discover, ScanningSubnet, Statistic, Configuration y Network cuentan con la información necesaria para cumplir con cada una de las responsabilidades que le corresponden. En el caso de la clase Discover tiene la responsabilidad de poseer la información necesaria para obtener los resultados de los escaneos.

## ✓ Patrón Creador:

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación. (39) Este patrón se basa en asignarle a una clase la responsabilidad de crear una instancia en otra clase.

**Ejemplo:** Las clases DiscoverSerializer, StatisticSerializer, ConfigurationSerializer y SubnetSerializer son las responsables de crear una nueva instancia de las clases Discover, Statistic, Configuration y Network para la visualización por el usuario.

## ✓ Patrón Bajo Acoplamiento

El patrón Bajo Acoplamiento consiste en asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

conectada a otras clases, con las que conoce y con que recurre a ellas. Propone el diseño de clases más independientes, lo que facilita la reutilización en otros sistemas.

**Ejemplo:** A la clase ExportExcel se le asigna la responsabilidad de exportar los resultados obtenidos de la clase DiscoverSerializer.

## ✓ Patrón Alta Cohesión

El patrón Alta Cohesión se basa en asignar una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas donde todos sus elementos trabajan juntos para proporcionar algún comportamiento bien delimitado.

**Ejemplo:** A las clases busquedaCompleta y busquedaSubred se le asignan responsabilidades con el objetivo que todos sus elementos colaboren y trabajen juntos.

## 3.6 Tarjetas CRC

La técnica CRC proporciona una forma de trabajo basado en grupos, para encontrar los objetos del dominio de la aplicación, sus responsabilidades y como colaboran con otros para realizar tareas. En esta técnica se utilizan las llamadas tarjetas CRC, las cuales son confeccionadas durante la etapa de diseño de la metodología de desarrollo XP y dentro de las cuales se registran el nombre de la clase, sus responsabilidades y las otras clases con las que colaboran.

Las principales características de las tarjetas son:

- ✓ Identificación de clases y asociaciones que participan del diseño del sistema.
- ✓ Obtención de las responsabilidades que debe cumplir cada clase.
- ✓ Establecimiento de cómo una clase colabora con otras clases para cumplir con sus responsabilidades.

A continuación se describen las tarjetas CRC correspondientes al módulo.

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

Clase: Rango	
Responsabilidad	Colaborador
Proveer las subredes para el escaneo y devolver las computadoras encendidas en dicha subred y las características esenciales.	NetworkInterface  Escanear

Tabla 11. Tarjeta CRC: Rango

Clase: Rango2	
Responsabilidad	Colaborador
Proveer las subredes para el escaneo automático y programado de todas ellas.	Escanear

Tabla 12. Tarjeta CRC: Rango2

Clase: NetworkInterface	
Responsabilidad	Colaborador
Se encarga de almacenar en base de datos las subredes en un listado único.	Save

Tabla 13. Tarjeta CRC: NetworkInterface

Clase: viewEscanear	
Responsabilidad	Colaborador

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Facilita los datos de las subredes y de las computadoras pertenecientes a dichas subredes para mostrarlos visualmente.	SubnetRestView  StatisticsRestView  viewShowSubnets
--	---

Tabla 14. Tarjeta CRC: viewEscanear

## 3.7 Modelo Físico de la Base de Datos

A continuación se presenta el modelo físico de la base de datos del módulo de descubrimiento automático de la red de computadoras del sistema GRHS.

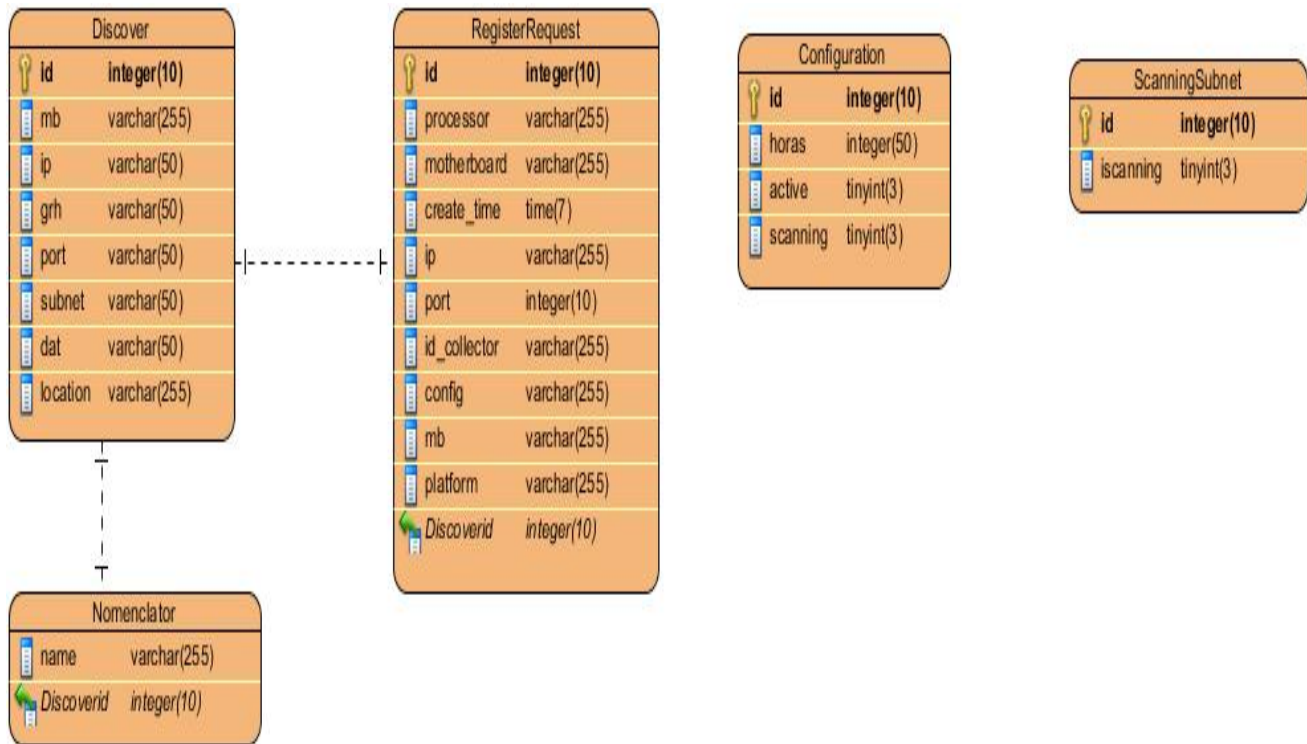


Ilustración 8. Modelo Físico de la Base de Datos.

De las entidades mostradas anteriormente, las más significativas son:

- ✓ **Discover:** La cual almacena los resultados de los escaneos de las subredes.

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

- ✓ **Configuration:** La cual almacena la configuración del escaneo programado de todas las subredes.
- ✓ **ScanningSubnet:** La cual almacena la información correspondiente a si se está haciendo un escaneo o no de una subred.

### 3.8 Tareas de Ingeniería

Las tareas de ingeniería son escritas por los desarrolladores a partir de las HU elaboradas por el cliente y de esta forma brindar un detalle más profundo para la correcta implementación de las mismas y estimar un tiempo más próximo a la realidad para cada una de ellas.

Tarea de Ingeniería	
<b>Número de Tarea:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre de la Tarea:</b> Almacenar subredes.	
<b>Tipo de Tarea:</b> Desarrollo (Desarrollo/Corrección/Mejora)	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 16/03/2015	<b>Fecha Fin:</b> 27/03/2015
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez René Juan Pérez de Corcho Hernández	
<b>Descripción:</b> Obteniendo la dirección IP y la máscara de red enviadas por el cliente, se calcula la subred a la que pertenece el mismo y se almacena en base de datos.	

Tabla 15. Tarea de Ingeniería 1- Almacenar subredes.

Tarea de Ingeniería	
<b>Número de Tarea:</b> 2	<b>Número Historia de Usuario:</b> 2

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

<b>Nombre de la Tarea:</b> Mostrar computadoras encendidas por subred.	
<b>Tipo de Tarea:</b> Desarrollo (Desarrollo/Corrección/Mejora)	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 30/03/2015	<b>Fecha Fin:</b> 03/04/2015
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez René Juárez Pérez de Corcho Hernández	
<b>Descripción:</b> El administrador selecciona la subred que se va a inventariar, inicia el escaneo y se guardan en base de datos las computadoras que están encendidas en ese momento en dicha subred.	

Tabla 16. Tarea de Ingeniería 2- Mostrar computadoras encendidas por subred.

Tarea de Ingeniería	
<b>Número de la Tarea:</b> 3	<b>Número de Historia de Usuario:</b> 3
<b>Nombre de la Tarea:</b> Descubrir clientes.	
<b>Tipo de Tarea:</b> Desarrollo (Desarrollo/Corrección/Mejora)	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 06/04/2015	<b>Fecha Fin:</b> 10/04/2015
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez René Juárez Pérez de Corcho Hernández	
<b>Descripción:</b> Esta tarea facilitará conocer si una estación de trabajo tiene instalado o no el cliente de GRHS.	

Tabla 17. Tarea de Ingeniería 3- Descubrir clientes.

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

Tarea de Ingeniería	
<b>Número de la Tarea:</b> 4	<b>Número de Historia de Usuario:</b> 4
<b>Nombre de la Tarea:</b> Obtener características de las computadoras.	
<b>Tipo de Tarea:</b> Desarrollo (Desarrollo/Corrección/Mejora)	<b>Puntos Estimados:</b> 2
<b>Fecha Inicio:</b> 13/04/2015	<b>Fecha Fin:</b> 24/04/2015
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez  René Juan Pérez de Corcho Hernández	
<b>Descripción:</b> Se realizarán consultas a la base de datos para obtener la información de los clientes registrados en la misma, obteniendo su identificador, su dirección IP, si tiene o no el cliente de GRHS instalado, el puerto por el que está ejecutándose, la subred a la que pertenece, la fecha en la que se realizó el escaneo y su localización.	

Tabla 18. Tarea de Ingeniería 4- Obtener características de las computadoras.

Tarea de Ingeniería	
<b>Número de Tarea:</b> 5	<b>Número Historia de Usuario:</b> 5
<b>Nombre de la Tarea:</b> Configurar escaneo.	
<b>Tipo de Tarea:</b> Desarrollo (Desarrollo/Corrección/Mejora)	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 27/04/2015	<b>Fecha Fin:</b> 01/05/2015
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez  René Juan Pérez de Corcho Hernández	



## Descubrimiento automático de la red de computadoras para el sistema GRHS.

**Descripción:** El administrador al seleccionar la opción configurar se mostrará un formulario donde él será capaz de introducirle el tiempo en horas en que desea se realice el escaneo y un activador para comenzar a escanear.

Tabla 19. Tarea de Ingeniería 5- Configurar escaneo.

Tarea de Ingeniería	
<b>Número de Tarea:</b> 6	<b>Número Historia de Usuario:</b> 6
<b>Nombre de la Tarea:</b> Exportar a formato (Excel).	
<b>Tipo de Tarea:</b> Desarrollo (Desarrollo/Corrección/Mejora)	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 04/05/2015	<b>Fecha Fin:</b> 08/05/2015
<b>Programador(es) responsable(s):</b> Arlety de la Caridad Sebastian Martínez René Juan Pérez de Corcho Hernández	
<b>Descripción:</b> El administrador es capaz de seleccionar si desea exportar la información obtenida del escaneo en formato Excel.	

Tabla 18. Tarea de Ingeniería 6- Exportar a formato (Excel).

Tarea de Ingeniería	
<b>Número de Tarea:</b> 7	<b>Número Historia de Usuario:</b> 7
<b>Nombre de la Tarea:</b> Mostrar estadísticas.	
<b>Tipo de Tarea:</b> Desarrollo (Desarrollo/Corrección/Mejora)	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 11/05/2015	<b>Fecha Fin:</b> 15/05/2015

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

**Programador(es) responsable(s):** Arlety de la Caridad Sebastian Martínez

René Juan Pérez de Corcho Hernández

**Descripción:** El administrador recibe la información general de los escaneos realizados, obteniendo el total de subredes, el total de computadoras escaneadas y el total de estaciones de trabajo que no tienen el cliente instalado.

Tabla 9. Tarea de Ingeniería 7- Mostrar estadísticas.

## 3.9 Pruebas

La metodología XP dentro de sus fases propone la constante realización de pruebas al sistema. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Estas pruebas no son más que una actividad en la cual el módulo desarrollado es probado bajo ciertas condiciones asegurando la eficacia del mismo.

### 3.9.1 Pruebas unitarias

Son las encargadas de verificar el código y son diseñadas por los programadores. Tienen como objetivo asegurar el correcto funcionamiento de las interfaces o flujo de datos entre clases. Las pruebas unitarias siempre están orientadas a caja blanca y permiten al programador conocer si determinada funcionalidad se puede agregar al sistema sin afectar su funcionamiento. (40) Para la realización de esta prueba se utilizó la herramienta PyUnit, utilizándose en las 3 iteraciones que comprenden el desarrollo del módulo.

#### 3.9.1.1 Iteración 1

A continuación se muestran dos fragmentos de código en los cuales no se detectaron errores.

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

```
...
-----
Ran 3 tests in 0.005s

OK
Destroying test database for alias 'default'...
```

Ilustración 9. Prueba unitaria a la iteración 1

```
Validating models...
0 errors found
Django version 1.4.5, using settings 'xilema.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Ilustración 10. Prueba unitaria del Modelo

### 3.9.1.2 Iteración 2

A continuación se muestra un fragmento de código en el cual no se detectó errores.

```
..
-----
Ran 2 tests in 0.003s

OK
Destroying test database for alias 'default'...
```

Ilustración 11. Prueba unitaria a la iteración 2

### 3.9.1.3 Iteración 3

A continuación se muestra un fragmento de código en el cual no se detectaron errores.

```
-----
Ran 6 tests in 0.006s

OK
Destroying test database for alias 'default'...
```

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Ilustración 12. Prueba unitaria a la iteración 3

## 3.9.2 Pruebas de aceptación

Las pruebas de aceptación que propone la metodología XP, también llamadas pruebas de caja negra, se derivan de las historias de usuarios que se han implementado como parte del desarrollo del módulo. Se centran en las características y funcionalidades generales de la solución elaboradas por parte del cliente. La representación de las pruebas de aceptación correspondientes a cada historia de usuario se representará mediante tablas, divididas en las siguientes secciones:

- Clases Válidas: se describe una entrada válida que hace el usuario con el objetivo de verificar si se obtiene el resultado esperado.
- Clases Inválidas: se describe una entrada inválida que hace el usuario con el objetivo de verificar si se obtiene el resultado esperado y cómo responde el módulo.
- Resultado Esperado: se describe el resultado que se espera ya sea para entradas válidas o inválidas.
- Resultado de la Prueba: se describe el resultado que se obtiene.
- Observaciones: algún señalamiento en la prueba. (41)

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
----------------	------------------	--------------------	------------------------	---------------

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

El sistema obtiene la dirección IP y la máscara de cada estación de trabajo, almacenándolos en base de datos.		El sistema muestra el listado de subredes, incluyendo la subred del cliente que ha sido añadida automáticamente.	Satisfactorio.	
---	--	--	----------------	--

Tabla 20. Prueba de Aceptación #1: HU: Adicionar automáticamente una subred.

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
	Si las computadoras no están encendidas el sistema no obtiene la dirección IP y la máscara de red de cada una de ellas.	El sistema muestra un listado de todas las subredes donde no se encuentra la subred registrada por el cliente.	Satisfactorio.	

Tabla 21. Prueba de Aceptación #2: HU: Adicionar automáticamente una subred.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El administrador selecciona la subred que desea inventariar.		El sistema muestra el resultado del escaneo realizado.	Satisfactorio.	

Tabla 22. Prueba de Aceptación #3: HU: Descubrir computadoras encendidas por subred.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

	El administrador selecciona más de una subred a inventariar.	El sistema muestra un mensaje informando que solo pueden seleccionar una subred.	Satisfactorio.	
--	--	--	----------------	--

Tabla 23. Prueba de Aceptación #4: HU: Descubrir computadoras encendidas por subred.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El módulo encuesta a cada una de las estaciones de trabajo, recibiendo una respuesta.		El sistema muestra si el cliente está instalado o no en cada una de las computadoras.	Satisfactorio.	

Tabla 24. Prueba de Aceptación #5: HU: Verificar si el cliente está instalado en cada computadora.

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
	Si alguna de las estaciones de trabajo está apagada no se recibe respuesta.	De las estaciones de trabajo que están apagadas y no han podido ser encuestadas no se muestra ninguna información.	Satisfactorio.	

Tabla 25. Prueba de Aceptación #6: HU: Verificar si el cliente está instalado en cada computadora.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
----------------	------------------	--------------------	------------------------	---------------



## Descubrimiento automático de la red de computadoras para el sistema GRHS.

<p>El sistema muestra de cada una de las computadoras escaneadas su identificador, su dirección IP, si tienen o no el cliente de GRHS instalado, el puerto, la subred a la que pertenece, la fecha en la que fue realizado el escaneo y su localización.</p>		<p>Se obtiene un listado con el identificador, la dirección IP, la subred a la que pertenece, el puerto por el que está ejecutándose, si tiene o no el cliente de GRHS instalado, la fecha en que fue encuestado y su localización.</p>	<p>Satisfactorio.</p>	
--	--	---	-----------------------	--

Tabla 26. Prueba de Aceptación #7: HU: Mostrar características esenciales de las computadoras.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
	<p>Si alguna de las estaciones de trabajo está apagada no se reciben los datos.</p>	<p>Se obtiene un listado donde no se encuentra la información correspondiente a la estación de trabajo apagada.</p>	<p>Satisfactorio.</p>	

Tabla 27. Prueba de Aceptación #8: HU: Mostrar características esenciales de las computadoras.

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El administrador selecciona la opción de configurar el escaneo de forma automática.		El sistema realiza el escaneo según el tiempo definido por el administrador.	Satisfactorio.	

Tabla 28. Prueba de Aceptación #9: HU: Configurar automáticamente el escaneo de la red.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El administrador selecciona la opción de exportar a formato Excel la información obtenida del escaneo.		El sistema genera un documento Excel.	Satisfactorio.	

Tabla 29. Prueba de Aceptación #10: HU: Exportar a formato (Excel).

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El administrador selecciona la opción de exportar a formato Excel la información obtenida del escaneo.		El sistema genera un documento Excel.	Satisfactorio.	

Tabla 30. Prueba de Aceptación #11: HU: Mostrar estadísticas generales.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

Al terminar las pruebas de aceptación se obtuvo un total de 25 no conformidades. En la primera iteración se encontraron 12, las cuales fueron resueltas. En la segunda iteración se obtuvieron 9 no conformidades y todas fueron resueltas. En la tercera iteración se encontraron 4 no conformidades las cuales fueron resueltas, durante las tres iteraciones no quedó ninguna no conformidad pendiente.

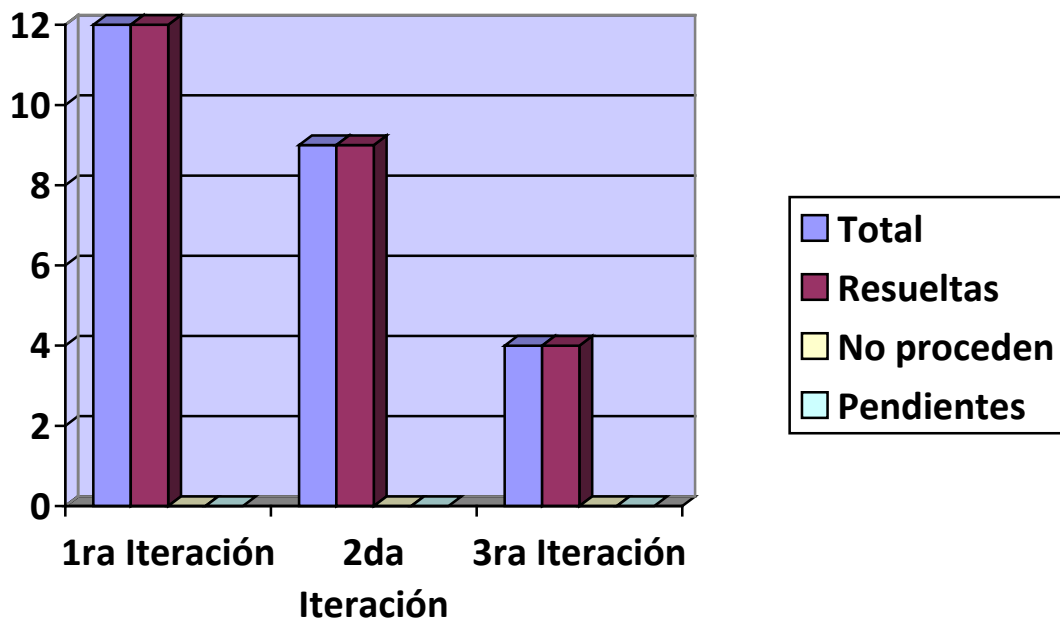


Ilustración 13. Resultado de las pruebas de aceptación.

## 3.10 Conclusiones parciales

En este capítulo se definió la arquitectura de software, para de esta forma diseñar y guiar el proceso de desarrollo del módulo implementado. Además se plantearon los patrones de diseños que se utilizaron en la elaboración de la solución. Se elaboraron las tarjetas CRC permitiendo identificar y organizar las clases orientadas a objetos. Se obtuvo el diagrama de diseño de la base de datos y se realizaron las tareas de ingeniería asociadas a cada HU. Se realizaron las pruebas unitarias y de aceptación para cada iteración del proceso de desarrollo del módulo, con el objetivo de garantizar

## Descubrimiento automático de la red de computadoras para el sistema GRHS.

un buen funcionamiento de la solución implementada y de esta manera cumplir con las necesidades del cliente.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Conclusiones Generales

En la presente investigación se realiza un descubrimiento automático de todas las estaciones de trabajo de la red de computadoras del sistema GRHS, arribando a las siguientes conclusiones:

- El estudio realizado demostró que los sistemas similares que realizan descubrimiento de la red no pueden ser integrados al sistema GRHS. De esta forma de las principales bibliotecas que realizan descubrimiento se seleccionó python-nmap como herramienta principal, por las funcionalidades que brinda.
- El estudio de las herramientas y tecnologías permitieron el desarrollo de un sistema robusto y flexible.
- El proceso de desarrollo fue guiado por la metodología XP, documentando toda la información del módulo, permitiendo una mejor comprensión del mismo al ser redactadas las historias de usuario, tarjetas CRC y las tareas de ingeniería, artefactos que se generan en esta metodología.
- Se realizaron las pruebas unitarias y de aceptación, asegurando de esta forma la buena calidad del módulo implementado.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Recomendaciones

Se recomienda a los interesados en mejorar o continuar esta investigación:

- ✓ Incorporar al sistema la posibilidad de escanear cualquier tipo de subred.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

## Bibliografía

1. **Belloch, Consuelo.** UV. [Online] 2014. [Cited: Noviembre 14, 2014.] <http://www.uv.es/bellochc/pedagogia/EVA1.pdf>.
2. **Universidad de las Ciencias Informáticas.** UCI. [Online] 2012. [Cited: Noviembre 25, 2014.] <http://www.uci.cu/?q=mision>.
3. **Comunidad de Programadores.** lwp. *lwp*. [Online] 2015. [Cited: Abril 17, 2015.] <http://www.lawebdelprogramador.com/diccionario/mostrar.php?letra=R&page=2>.
4. **Castro, Luis.** about. *about*. [Online] 2015. [Cited: Abril 17, 2015.] <http://aprenderinternet.about.com/od/Glosario/fl/Que-es-una-libreria.htm>.
5. Definicion. [Online] [Cited: Mayo 29, 2015.] <http://definicion.de/componentes/>.
6. AllAboutCookies. [Online] [Cited: Mayo 29, 2015.] <http://www.allaboutcookies.org/es/faqs/protocolo.html>.
7. iSoftland. *iSoftland*. [Online] [Cited: Abril 17, 2015.] <http://www.isoftland.com/software/solarwinds/lansurveyor?showall=1>.
8. **Redes Telecom.** redestelecom. *redestelecom*. [Online] Noviembre 22, 2002. [Cited: Abril 17, 2015.] <http://www.redestelecom.es/infraestructuras/reportajes/1024145001803/spectrum-gestion-mas-eficaz-red.1.html>.
9. **Paessler AG.** paessler. *paessler*. [Online] [Cited: Abril 17, 2015.] <http://www.es.paessler.com/prtg>.
10. **Lyon, Gordon.** nmap. *nmap*. [Online] Mayo 28, 2014. [Cited: Abril 17, 2015.] <http://www.nmap.org>.
11. **Biondi, Philippe.** Scapy. *Scapy*. [Online] Abril 19, 2010. [Cited: Abril 17, 2015.] <http://www.secdev.org/projects/scapy/doc/introduction.html#about-scapy>.
12. **Lyon, Gordon.** zenmap. *nmap*. [Online] Mayo 28, 2014. [Cited: Abril 17, 2015.] <http://www.nmap.org/zenmap>.
13. **Darknet.** *darknet*. [Online] 2015. [Cited: Abril 17, 2015.] <http://www.darknet.org.uk/2014/09/masscan-fastest-tcp-port-scanner/>.
14. [Online] [Cited: Mayo 29, 2015.] <https://msdn.microsoft.com/es-es/library/cc758357%28v=ws.10%29.aspx>.
15. [Online] [Cited: Mayo 29, 2015.] <https://msdn.microsoft.com/es-es/library/cc758065%28v=ws.10%29.aspx>.

# Descubrimiento automático de la red de computadoras para el sistema GRHS.

16. **Xuletas.** [Online] [Cited: Mayo 28, 2015.] <http://www.xuletas.es/ficha/protocolos-de-encaminamiento-rip-y-ospf/>.
17. **Xuletas.** [Online] [Cited: Mayo 28, 2015.] <http://www.xuletas.es/ficha/protocolos-de-encaminamiento-rip-y-ospf/>.
18. **InnovalProcess.** *InnovalProcess.* [Online] 2014. [Cited: Abril 18, 2015.] <http://www.innovalprocess.com/detalle-ingenieria-software.html>.
19. **Letelier, Patricio.** *cyta.* [Online] Enero 15, 2006. [Cited: Abril 18, 2015.] <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
20. Instituto Tecnológico de Celaya. [Online] [Cited: Abril 18, 2015.] [www.iqcelaya.itc.mx/~vicente/Programacion/Lenguajes.doc](http://www.iqcelaya.itc.mx/~vicente/Programacion/Lenguajes.doc).
21. **Di Pierro, Massimo.** *web2py.* [Online] [Cited: Abril 18, 2015.] <http://web2py.com/books/default/chapter/36/02/el-lenguaje-python>.
22. **Comunidad MDN.** *MDN.* [Online] [Cited: Abril 18, 2015.] <https://developer.mozilla.org/es/docs/HTML/HTML5>.
23. —. JavaScript. *MDN.* [Online] [Cited: Abril 18, 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript>.
24. **Docsetools.com.** *Docsetools.* [Online] [Cited: Abril 18, 2015.] [http://docsetools.com/articulos-para-saber-mas/article\\_56561.html](http://docsetools.com/articulos-para-saber-mas/article_56561.html).
25. **Gutiérrez, Javier.** [Online] [Cited: Abril 18, 2015.] [www.lsi.us.es/~javier/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javier/investigacion_ficheros/Framework.pdf).
26. **Django project.** *django.* [Online] [Cited: Abril 18, 2015.] <http://www.django.es>.
27. **JQuery Foundation.** *JQuery.* [Online] [Cited: Abril 18, 2015.] <https://jquery.com/>.
28. **Sphinx.** Curso SQL. [Online] [Cited: Abril 18, 2015.] [http://csrg.inf.utfsm.cl/~jfuentes/\\_build/html/lectures/week1/lecture1.html](http://csrg.inf.utfsm.cl/~jfuentes/_build/html/lectures/week1/lecture1.html).
29. **Martinez, Rafael.** *PostgreSQL-es.* [Online] Julio 17, 2009. [Cited: Abril 18, 2015.] <http://www.postgresql.org.es/node/313#sobre-postgresql>.
30. **Guión Visual Paradigm.** [Online] 2014. [Cited: Abril 18, 2015.] <http://www.ie.inf.uc3m.es/grupo/docencia/reglada/ls1y2/PracticaVP.pdf>.
31. **Corporate BizAgi.** *BizAgi Process Modeler.* [Online] [Cited: Abril 18, 2015.] <http://www.bizagi.com/esp/descargas/BPMNbyExample.pdf>.
32. **Buzz.** *BuzzSG.* [Online] [Cited: Abril 18, 2015.] <http://sg.com.mx/revista/27/arquitectura-software>.



# Descubrimiento automático de la red de computadoras para el sistema GRHS.

33. **Magazine.** *MasterMagazine.* [Online] [Cited: Abril 18, 2015.] <http://www.mastermagazine.info/termino/4294.php>.
34. **IngenioDS.** *IngenioDS.* [Online] [Cited: Abril 18, 2015.] <https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/>.
35. **Holovaty, Adrian and Kaplan-Moss, Jacob.** LibrosWeb. *LibrosWeb.* [Online] 2015. [Cited: Mayo 6, 2015.] [https://librosweb.es/libro/django\\_1\\_0/capitulo\\_5/el\\_patron\\_de\\_diseno\\_mtv.html](https://librosweb.es/libro/django_1_0/capitulo_5/el_patron_de_diseno_mtv.html).
36. **Tedeschi, Nicolás.** *Microsoft Developer Network.* [Online] [Cited: Abril 18, 2015.] <https://msdn.microsoft.com/es-es/library/bb972240.aspx>.
37. *Prácticas de Software.* [Online] [Cited: Abril 18, 2015.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
38. **Universidad Simón Bolívar.** [Online] [Cited: Abril 18, 2015.] [ldc.usb.ve/~ci3715/teoria/patrones.pdf](http://ldc.usb.ve/~ci3715/teoria/patrones.pdf).
39. **Universidad Simón Bolívar.** [Online] [Cited: Abril 18, 2015.] [ldc.usb.ve/~ci3715/teoria/patrones.pdf](http://ldc.usb.ve/~ci3715/teoria/patrones.pdf).
40. **Oré, Alexander.** *CalidadySoftware.* [Online] [Cited: Mayo 29, 2015.] [http://www.calidadysoftware.com/testing/pruebas\\_unitarias1.php](http://www.calidadysoftware.com/testing/pruebas_unitarias1.php).
41. **Vaisala.** [Online] [Cited: Mayo 29, 2015.] <http://es.vaisala.com/sp/services/projectservices/Pages/acceptancetesting.aspx>.