

Universidad de las Ciencias Informáticas

Facultad 10



“Sistema de Clonación y Distribución de Imágenes de Sistemas Operativos”

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

Autor : Dayron Pérez Roldán

Tutor : Ing. Abel Meneses Abad

Consultante : Ing. Alejandro Valdés Villarubia

Ciudad de la Habana 23 de junio del 2008

Año 50 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter no exclusivo.

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

Dayron Pérez Roldán

Ing. Abel Meneses Abad.

El éxito de los hombres no se mide por su éxito inmediato, sino por su éxito definitivo, no se mide por el dinero que acumularon, sino por el resultado de sus obras.

José Martí.

Agradecimientos

A mis padres **Sandra Roldán Carballoza** y **Jesús Angel Pérez Castillo**, por todo su amor, por el apoyo, el ejemplo y por saberme guiar siempre por el camino correcto.

A mi hermanito **Daynel Pérez Roldán** por ser el mayor tesoro que tengo en la vida y por ser mi fuerza en los momentos difíciles.

A mi abuela **Dulce Maria Pérez Castillo** por adorarme, por tantos años de preocupación, dedicación y entrega.

A mis abuelos **Gina** y **Jesús** por sus buenos consejos y por haberme apoyado y darme fuerzas para continuar.

A mi tía **Mary** por alentarme siempre y ayudarme en los momentos que más lo necesite.

A mis tios **Jorge** y **Iliana** y a mis primos por sus consejos, por apoyarme y alentarme para acabar mi carrera.

A mis amigos y hermanos **Alejandro Valdés Villarruvia**, **Aramis Rescalla Pupo** y **Reynaldo Heredia Rodríguez** que le debo mucho en mi carrera y en la vida.

A mi amiga y novia **Yener Uyoa del Toro** por haber dado su vida a mi lado, por entregarme lo mejor de ella y por enseñarme a ser una mejor persona.

A mi tutor **Abel Meneses Abad**, por ayudarme siempre que lo necesitaba.

Jorge Mijail Vazquez Paredes un amigo más al cual le debo muchas de las cosas que se desarrollaron en la aplicación y por dedicar su tiempo al desarrollo de la misma.

A mis **compañeros de estudio** por toda la ayuda que me han prestado.

A todo el que de una forma u otra ha hecho posible la realización de este sueño.

Dedicatoria

A mi hermanito Daynel.

A mi familia y amigos en general.

Resumen

Debido al alto número de laboratorios docentes con los que cuenta la Universidad de las Ciencias Informáticas (UCI) y al gran número de computadoras de diversos tipos que componen cada uno, se hace necesario una elevada cantidad de personal para poder mantener estos laboratorios en óptimas condiciones para su uso. Teniendo en cuenta esto y con el objetivo de reducir el tiempo y la cantidad de personal necesaria para realizar esta tarea, es que se desarrolla el presente trabajo.

En este trabajo se hace un estudio de los sistemas de clonación y distribución de software existentes en el mundo y los métodos de clonación empleados, también se hace un análisis del funcionamiento y características fundamentales de cada uno de estos sistemas y las herramientas empleadas para su desarrollo. Además se lleva a cabo la implementación de un sistema capaz de clonar, particionar e instalar imágenes de sistemas operativos en las computadoras de una subred de forma simultánea al igual que todas aquellas pruebas llevadas a cabo para lograr el perfecto funcionamiento del software. Un sistema que para su funcionamiento utiliza la tecnología de Clientes Ligeros, repartiendo una meta-distribución del sistema operativo GNU/Linux a todas las computadoras de la subred y convirtiéndolas en clientes y tomando el control total del hardware de cada una de las computadoras.

Índice de contenido

INTRODUCCIÓN.....	1
1. CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	4
1.1. Introducción.....	4
1.2. Sistemas de clonación y distribución de software.....	4
1.2.1. Funcionamiento de los sistemas de clonación y distribución de software.....	5
1.2.2. Métodos de clonación.....	7
1.3. Sistemas de clonación y distribución de software de uso actual.....	9
1.3.1. Brutalix 1.1.....	11
1.3.2. Clonezilla 1.0.10-6.....	11
1.3.3. Norton Ghost 14.0.....	12
1.3.4. Rembo Toolkit 4.0.....	13
1.3.5. ADS (Automated Deployment Services) 1.1.....	15
1.4. Tendencias de los sistemas de clonación y distribución de software.....	16
1.4.1. Reseña histórica.	16
1.4.2. Principales tendencias.	18
1.5. Herramientas, lenguajes y tecnologías a utilizar.....	18
1.5.1. Code::blocks.....	19
1.5.2. Anjuta.....	20
1.5.3. wxWidgets.....	21
1.5.4. Sfdisk.....	21
1.5.5. Partimage.....	22
1.5.6. NFS.....	22
1.5.7. Bash.....	23
1.5.8. Perl.....	23
1.5.9. Lenguaje C/C++	24
1.5.9.1. Ncurses.....	25
1.5.9.2. Socket.....	25
1.5.9.3. Hilos.....	26
1.5.9.4. Semáforos.....	26
1.5.10. LTSP.....	26
1.5.11. DHCP.....	26
1.5.12. TFTP.....	27
1.5.13. UML.....	27
1.5.14. Postgresql.....	27

1.6. Metodologías ágiles utilizadas.....	29
1.6.1. Scrum.....	30
1.6.2. XP.....	31
1.7. Conclusiones.....	32
2. CAPÍTULO II. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.....	33
2.1. Introducción.....	33
2.2. Valoración crítica del diseño propuesto por el analista.....	33
2.3. Análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados. Estrategia de integración.....	35
2.4. Conclusiones.....	37
3. CAPÍTULO III. DESARROLLO ÁGIL DEL SISTEMA DE CLONACIÓN Y DISTRIBUCIÓN DE IMÁGENES DE SISTEMA OPERATIVOS.....	38
3.1. Introducción.....	38
3.2. Planificación del proyecto por roles.....	38
3.3. Historias de usuarios y prototipos de interfaz de usuario.....	38
3.4. Base de datos.....	75
3.5. Diagrama de componentes.....	76
3.6. Plan de releases.....	77
3.7. Conclusiones.....	77
4. CAPÍTULO IV. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	78
4.1. Introducción.....	78
4.2. Casos de Prueba.....	78
4.2.1. Caso de Prueba Historia de Usuario: U-SCDSW-01.....	79
4.2.2. Caso de Prueba Historia de Usuario: U-SCDSW-02.....	81
4.2.3. Caso de Prueba Historia de Usuario: U-SCDSW-03.....	82
4.2.4. Caso de Prueba Historia de Usuario: U-SCDSW-06.....	84
4.2.5. Caso de Prueba Historia de Usuario: U-SCDSW-07.....	85
4.2.6. Caso de Prueba Historia de Usuario: U-SCDSW-09.....	86
4.2.7. Caso de Prueba Historia de Usuario: U-SCDSW-13.....	87
4.2.8. Caso de Prueba Historia de Usuario: U-SCDSW-15.....	88
4.2.9. Caso de Prueba Historia de Usuario: U-SCDSW-16.....	89
4.2.10. Caso de Prueba Historia de Usuario: U-SCDSW-17.....	90
4.2.11. Caso de Prueba Historia de Usuario: U-SCDSW-18.....	92
4.2.12. Caso de Prueba Historia de Usuario: U-SCDSW-19.....	93
4.3. Resultados obtenidos.....	94
4.3.1. Acerca del tiempo de desarrollo.....	94
4.3.2. Acerca de las funcionalidades obtenidas.....	94

4.4. Conclusiones.....	96
Conclusiones Generales.....	97
Recomendaciones.....	98
Referencias Bibliográficas.....	99
Bibliografía.....	102
Anexos.....	103
Anexo No. 1 Diseño operacional de la aplicación RESC.....	103
Glosario de Términos.....	104

INTRODUCCIÓN

Las tecnologías de la información y las comunicaciones (TIC) se hacen imprescindibles para el desarrollo mundial. Cuba con el objetivo de alcanzar un mayor desarrollo, se ha introducido aun más en la esfera de la producción de software, con la creación de la más joven de las universidades cubanas, la Universidad de las Ciencias Informáticas (UCI). Esta universidad está estructurada en diferentes facultades, donde cada facultad se orienta a un perfil determinado, pero que tienen como principal objetivo la formación de especialistas en informática, teniendo como principio del proceso docente educativo "La formación desde la producción".

De acuerdo con este principio, la formación docente está vinculada a la producción, donde el estudiante adquiere las habilidades prácticas para el desarrollo y producción de software, el cual se desarrolla en los laboratorios docentes.

Estos laboratorios docentes se encuentran equipados con diferentes tipos de computadoras, que son atendidas por el grupo de técnicos de la universidad, quienes les dan el mantenimiento e instalan el software necesario de acuerdo con los requerimientos del proceso docente educativo. Teniendo en cuenta que para cada facultad las necesidades en cuanto a software que utilizan son diferentes y que las imágenes a instalar en las computadoras tienen que ser personalizadas para cada tipo de computadora y teniendo en cuenta el perfil de cada facultad, lo que provoca que se tengan que hacer varias imágenes personalizadas con las necesidades de cada una de las facultades.

El **problema** que motiva la realización de este trabajo, es que actualmente en la UCI el proceso docente está soportado por importantes servicios digitales, entre ellos las imágenes para la docencia. Este servicio se efectúa en más de 2500 computadoras en la universidad. Para instalar y mantener las imágenes de los diferentes sistemas operativos se emplean a nivel UCI un equipo de técnicos, y aplicaciones informáticas conocidas como sistemas de clonación. Los sistemas de clonación son eficientes en tres niveles en dependencia de su característica. El proceso se realiza con software propietario, sin la calidad requerida. En software libre las aplicaciones para estos fines no sirven de manera adecuada al diseño del servicio en la UCI. No es posible entonces clonar imágenes o distribuir software de manera automática en la UCI con software libre, con las mismas prestaciones que las aplicaciones propietarias existentes para el servicio de imágenes a la docencia. Producto a esto surge la **situación problemática**, que consiste en que se tiene que pasar imagen a una gran cantidad de computadoras a la vez y no existe un software que automatice este proceso. Por lo que el

problema científico consiste en ¿Cómo implementar un software que automatice el proceso de clonación y distribución de imágenes en la UCI?

Con este trabajo se pretende implementar y probar un sistema de clonación de segundo nivel para la distribución de software y actualizaciones propio para la UCI.

Por tanto el **objeto de investigación** de este trabajo son los sistemas de clonación y distribución de software existentes y su funcionamiento. El **campo de acción** serán todos los laboratorios docentes de la facultad 10.

El **objetivo** de este trabajo es implementar un sistema cliente/servidor que permita clonar imágenes de sistemas operativos GNU/Linux de forma remota. Implementar dicho sistema de manera que se pueda distribuir software y actualizaciones para todos los laboratorios docentes de la facultad 10; con perspectivas de poder implantar el uso de dicho software en toda la UCI.

Para cumplir con el objetivo propuesto se han definido los siguientes **objetivos específicos**:

- ◆ Investigar los diferentes sistemas de clonación y distribución de software existentes en Cuba y el mundo.
- ◆ Comprender el funcionamiento de los sistemas de clonación y distribución de software.
- ◆ Implementar y probar un sistema cliente/servidor que permita clonar y distribuir imágenes de sistemas operativos de manera remota.

El desarrollo de las distintas tareas se ordena teniendo como punto central el Sistema. Se pretende que este y todas las aplicaciones que conllevarán su funcionamiento puedan ser usadas por otros programas para beneficios de todos. Las herramientas a utilizar son todas herramientas libres, punto este que permitirá su redistribución, modificación y la propia utilización fuera de cualquier conflicto legal que pudiera pensarse.

Las principales **tareas** que se proponen para concretar los objetivos son las siguientes:

- ◆ Investigar los sistemas de clonación existentes en cuba y el mundo, sus características, ventajas y desventajas.
- ◆ Comprender el funcionamiento de forma general de los sistemas de clonación y distribución de software para lograr una mejor implementación del sistema propuesto.
- ◆ Crear las bases para que este producto pueda ser usado por diferentes tipos de

arquitecturas de hardware.

- ◆ Desarrollar un software con robustez y fiabilidad para el usuario final.
- ◆ Diseñar y realizar todas las pruebas necesarias para limar los posibles errores que se puedan presentar en el ciclo de vida del producto.

Este trabajo está estructurado en 4 capítulos y anexos, que incluye todo lo relacionado con el trabajo investigativo sobre los sistemas de clonación y distribución de software, así como la implementación y pruebas del sistema en cuestión. A continuación se muestra una breve descripción de cada uno de los capítulos.

Capítulo 1: Fundamentación teórica. En este capítulo se hace un análisis del tema a tratar, de las tecnologías actuales relacionadas con él y los métodos de clonación empleados. Además se hace una breve descripción de las técnicas y tecnologías utilizadas para la implementación del sistema.

Capítulo 2: Descripción y análisis de la solución propuesta. Se realiza una valoración crítica de la solución propuesta por el analista. Se analizan posibles componentes ya existentes los cuales pueden ser utilizados nuevamente; y la estrategia de integración al sistema.

Capítulo 3: Desarrollo ágil del sistema de clonación y distribución de imágenes de sistemas operativos. En este capítulo se realiza el desarrollo ágil del **SistClon**. Se explica toda la dinámica del proyecto en forma de historias de usuarios, prototipos de interfaz de usuario y algunos modelos auxiliares además del plan de releases para las entregas intermedias.

Capítulo 4: Validación de la solución propuesta. En este capítulo se plasman los casos de pruebas o test de aceptación a las que fue sometida la aplicación en cada una de las iteraciones. Se exponen los resultados obtenidos y se muestran las funcionalidades alcanzadas en el período de desarrollo.

1. CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.

1.1. Introducción

En el presente capítulo se analiza el funcionamiento de los sistemas de clonación, se exponen también los fundamentos de los sistemas de clonación y distribución de software existentes. Se muestran las diferentes herramientas, tecnologías y lenguajes que fueron usados para realizar la implementación del sistema.

1.2. Sistemas de clonación y distribución de software.

En la actualidad uno de los mayores problemas al que se enfrenta cualquier institución o empresa que presente al menos un número considerable de computadoras es el de dar soporte y distribuir la misma imagen de sistemas operativos y software a una serie de computadoras por igual; el proceso después de ser repetido una y otra vez a cada computadora de forma individual se convierte en tedioso. Por esta razón los desarrolladores se entregaron a la tarea de automatizar este proceso lo mejor posible para brindar rapidez y eficiencia a los administradores y técnicos que se encargan de mantener y dar soporte a dichos grupos de computadoras.

El termino Clonación en el vocablo informático viene asociado exactamente a replicar una información que se encuentra en una zona de memoria a otra zona de memoria, a esto también se le llama "copia exacta". En la clonación de sistemas operativos existe una variante y es la copia exacta o clonación de una zona de memoria de una computadora a otra zona de memoria en el mismo disco duro o en otro disco duro remoto ya sea por medio de una salva de el sistema operativo en algún dispositivo de almacenamiento (CD/DVD/USB,etc) o la manera más usada actualmente, vía red usando el Pre-Boot Execution Environment (PXE)(Peter Anvin) de la tarjeta de red y a estos software se les denomina sistemas de instalación remota.

Los sistemas de instalación remota se definen como los software que permiten que un usuario instale y configure nuevos equipos clientes de forma remota, sin necesidad de trabajar directamente en cada equipo cliente. Estos sistemas en la actualidad se han desarrollado en dos ramas fundamentales: los sistemas de instalación remota básica, y los sistemas de clonación y distribución de software. Los sistemas de instalación remota básica se limitan a la instalación de un sistema operativo base en los equipos clientes, haciendo uso de la técnica de los llamados archivos de respuestas. Esta técnica consiste en crear un archivo donde se especifican las respuestas a las preguntas que hace el instalador por

defecto de los sistemas operativos, estos archivos son creados por el administrador del sistema y una vez concluida su configuración son distribuidos en los diferentes equipos clientes con el objetivo de automatizar el proceso de instalación de los diferentes equipos.

A diferencia de los sistemas de instalación remota básica, los sistemas de clonación y distribución de software son capaces de realizar funciones como particionamiento de discos duros, clonación de sistemas operativos y distribución de imágenes de sistemas operativos. Para poder realizar estas funciones, hacen uso de diferentes tecnologías y utilidades como: Linux Terminal Server Project (LTSP)(Morales A, Torres J, y Quesada A), Partimage(Francois Dupoux y Franck Ladurelle), PXE(Intel Corporation 1999), entre otras.

1.2.1. Funcionamiento de los sistemas de clonación y distribución de software.

Los sistemas de Clonación y Distribución de Software se basan en un clásico modelo cliente/servidor, donde los clientes son los equipos que se van a instalar, y el servidor es la computadora que le proveerá a los clientes las imágenes que se les instalará. Por lo tanto en estos sistemas se tienen dos tipos de aplicaciones diferentes: una para el cliente y otra para el servidor. La aplicación servidor utiliza diferentes servicios (DHCP(Bucknell University), TFTP(Massachusetts Institute of Technology), NFS(Sun Microsystems)) y tecnologías (Cluster) para poder comunicarse con los diferentes clientes y realizar el proceso de clonación y distribución.

Este proceso de clonación y distribución de software se divide en dos procesos fundamentales: El procesos de arranque y conexión del cliente con el servidor y el proceso de transferencia de ficheros y ejecución de comandos.

Proceso de arranque y conexión del cliente con el servidor.

En este proceso el servidor comienza una comunicación con el cliente, suministrándole una configuración de la interfaz TCP/IP de la tarjeta de red, para establecer la conexión por la que realizará la transferencia de ficheros. Este proceso inicia una vez que se enciende la computadora cliente.

- Cuando se enciende la computadora cliente, este irá a través de su Power On Self Test - POST (autorevisión de encendido).
- Durante la autorevisión, el BIOS buscará roms de expansión. Cada tarjeta de red posee una bootrom Etherboot(Marty Connor y Michael Brown), la cual es una rom de expansión. Por lo tanto el BIOS detectará la rom en la placa de red.

- Una vez que el POST es completado, la ejecución saltará dentro del código Etherboot.
- El código Etherboot buscará una placa de red. Una vez que la misma es detectada, será inicializada.
- El código Etherboot realizará luego una petición DHCP mediante un broadcast por la red. La petición incluirá la dirección MAC de la tarjeta de red.
- El servidor DHCP verá el broadcast y responderá el pedido leyendo su archivo de configuración y localizando la entrada que coincide con la dirección MAC que fue enviada. El servidor DHCP construirá un paquete de respuesta, conteniendo varias piezas de información. Este paquete será enviado de vuelta a la estación de trabajo. Esta respuesta incluye:
 - ◆ Dirección IP asignada al terminal
 - ◆ Máscara de red configurada para la red local.
 - ◆ Directorio del cual extraer el kernel.
 - ◆ Localización del sistema de archivos raíz a montar.
 - ◆ Parámetros opcionales que deben ser pasados al kernel, a través de la línea de comandos del kernel.
- El código Etherboot recibirá la respuesta desde el servidor y configurará la interfase TCP/IP en la tarjeta de red con los parámetros que le fueron suministrados.

Una vez concluido este paso automáticamente comienza el proceso de transferencia de ficheros y ejecución de comandos.

Proceso de transferencia de ficheros y ejecución de comandos.

Al concluir el proceso de configuración de la interfaz TCP/IP de la tarjeta de red se comienza la configuración de los diferentes servicios de transferencia de ficheros.

- Utilizando Trivial File Transfer Protocol (TFTP), el código Etherboot se contactará con el servidor y comenzará a descargar el núcleo.
- Una vez que el núcleo fue completamente descargado en la estación de trabajo, el código Etherboot colocará al mismo en la correcta localización de memoria.
- El control es pasado luego al núcleo, el cual inicializará al sistema completo y a todos los periféricos que reconozca.
- Luego hará otro pedido al servidor DHCP, el cual responderá enviando la información que necesita para continuar, entre ellas:

- ◆ Dirección IP asignada a la computadora.
 - ◆ Máscara de red configurada para la red local.
 - ◆ El directorio raíz a ser montado por NFS.
 - ◆ El gateway.
 - ◆ El servidor DNS.
 - ◆ El hostname de la computadora.
- Una vez que estos parámetros llegados a la computadora, la nueva interfaz de red será configurada.
 - El sistema de archivo raíz se montará vía NFS. Este se montará como solo lectura como medida de seguridad debido a que será montado simultáneamente en cada uno de los clientes a instalar.

Una vez montado el sistema de archivos en el cliente, el control es tomado por el kernel pasando a hacer un reconocimiento del hardware del cliente. Después de terminado este reconocimiento, el kernel pasa a ejecutar los diferentes scripts de configuración y ejecución de comandos determinado por el administrador del sistema para realizar la instalación, clonación o particionamiento del cliente.

Pero no todos los sistemas de clonación utilizan los mismos métodos para restaurar un sistema operativo. En la próxima sección se hará referencia a los diferentes métodos de clonación de sistemas operativos, sus ventajas y desventajas, y cual puede ser el método más óptimo.

1.2.2. Métodos de clonación.

Como se explicó anteriormente la clonación y distribución de software es el proceso de duplicar o instalar sistemas operativos desde una computadora servidor a una o más computadoras clientes. Esto se puede hacer por medio de una salva o backup(copia de seguridad) de datos con toda la información de una partición o el disco duro entero de la computadora servidor para su posterior restitución en las computadoras clientes. Hay dos métodos conocidos para la clonación de particiones de discos que contengan el sistema operativo que se desea duplicar, **clonación basada en sectores** y **clonación basada en la estructura**. El método de clonación basada en sectores además puede ser dividido en dos categorías :(Rembo Technology y IBM Company 2005)

Clonación a ciegas o sin conocimiento del sistema de archivos.

Este es el método más simple de la clonación. Cada uno de los sectores (unidades secuenciales de datos del disco) son almacenados en un backup de datos y a continuación restaurados en las computadoras clientes. Este método no tiene conocimiento de la estructura del sistema de archivos subyacente. Este tipo de método tiene la ventaja de apoyar a prácticamente todos los sistemas operativos, ya que no se preocupan por el contenido del disco duro. El primer inconveniente de este método es que la partición de destino debe tener el mismo tamaño que la partición de referencia pues las estructuras de archivos se copian tal y como son, con toda la información sobre el tamaño de los archivos transferidos sin modificar. La segunda desventaja es que el espacio libre también es clonado (este método no sabe si una determinada unidad de datos está en uso o no).

Si la partición de referencia usa solo el 10 % del total del espacio en disco, entonces solo el 10 % de la imagen del disco es relevante. Pequeñas imágenes de disco representan una transferencia más rápida y un rápido restablecimiento, es evidente que para particiones muy grandes – con solo un 10 % de uso – mediante este método se estaría enviando información innecesaria y sobrecargando el uso de la red, esto representa una seria desventaja.

Clonación semi-consciente o con medio conocimiento del sistema de archivos.

Este método es una evolución de los métodos de clonación a ciegas descrito en el punto anterior, y ha sido diseñado para eliminar las dos principales desventajas del mismo. El método de clonación con medio conocimiento del sistema de archivos como ya lo dice su nombre tiene un conocimiento limitado de los mecanismos de asignación utilizados por la estructura del sistema de archivos subyacente. Este conocimiento permite omitir los datos no utilizados en las unidades y almacenar sólo la información relevante en la imagen del disco.

Además existen herramientas que usan este método y son capaces de restaurar una imagen de disco en una partición que difiere en tamaño a la partición original. Esto se hace modificando limitadas partes de la estructura del sistema de archivos subyacente. Este método resuelve las dos principales desventajas de la clonación a ciegas, pero también introduce una nueva desventaja. El método de clonación semi-consciente debe ser capaz de comprender y modificar las estructuras internas utilizadas por el sistema de archivos instalado en la partición de referencia. En consecuencia, este método solo funcionará para la lista de sistemas de archivos soportados por la herramienta de clonación.

Teniendo en cuenta las deficiencias o desventajas de los métodos de clonación a ciegas o

semi-consciente es que se realiza el estudio del método de clonación basada en la estructura, ya que resuelve las mayores desventajas de los métodos tratados anteriormente.

La clonación basada en la estructura es lo contrario del método de clonación basado en sectores, ya que requiere un amplio conocimiento del sistema de ficheros instalado en el ordenador de referencia y la capacidad de reconstruir un completo sistema de ficheros en el ordenador cliente. Este método trabaja con archivos y directorios en lugar de los sectores al igual que el proceso de copia común. Las principales ventajas de este método son:

- Es más flexible que el método de clonación basado en sectores. La herramienta de clonación puede seleccionar que ficheros va a salvar o a restaurar (o archivos que no va a salvar como por ejemplo los ficheros de paginación de memoria virtual).
- Puede restaurar los archivos existentes en un sistema de ficheros. Los archivos existentes no se descartan.
- Las imágenes de discos pueden ser editadas para eliminar los archivos o directorios no deseados en la etapa de la restauración (las imágenes de discos son como un sistema de archivos).
- Sincronización de archivos. Archivos ya instalados en el equipo cliente e idénticos a los archivos en la imagen de referencia no son restaurados.
- Reduce los requerimientos de almacenamiento en el servidor debido a que solo procesa la información existente.
- Acelera el proceso de creación y descarga de las imágenes de sistemas operativos.

Estos son los principales métodos de clonación existentes en la actualidad (marzo/2008), son los más usados por los programas destinados a realizar este proceso de clonación y distribución de software.

1.3. Sistemas de clonación y distribución de software de uso actual.

A nivel mundial se han desarrollado diferentes sistemas para la instalación y administración de grupos de equipos en una red. Algunos de estos sistemas se han orientado específicamente a la clonación de sistemas operativos y otros han evolucionado para dar lugar a aplicaciones que son capaces de clonar un sistema operativo y distribuirlo e instalarlo de manera remota en computadoras en masa. En la actualidad existen 29 sistemas de clonación y distribución de software conocidos, de ellos 15 son privativos y 14 son libres.

En la siguiente tabla se ilustran estos sistemas :

Nombre del Sistema	Sistemas Operativos	Licencia
PC INSPECTOR clone maxx 1.0 (CONVAR)	Win98/98SE/Me/2000/XP	Privativo
Partimage Is Not Ghost (PING) 2.01.12 (PING Team)	Linux/Unix-like OSes	GPL
XXClone 0.58.0 (XXClone Team)	Win98/98SE/Me/2000/XP/Vista	Privativo
Norton Ghost 14.0 (Symantec)	WinXP/Vista	Privativo
G4u 2.3 (G4u Team)	Linux/BSD/Solaris	BSD
HDClone 3.2.9 (Miray)	Win98/98SE/Me/2000/XP	Privativo
RIS (Remote Instalation Service) (Microsoft Windows)	Win2003/XP	Privativo
G4L 0.24 (G4L Team)	Linux	GPL
CloneHDD 1.3 (CloneHDD Team)	FreeBSD	GPL
Clonezilla 1.0.10-6 (Clonezilla Team)	Linux/BSD/Unix-like OSes	GPL
Acronis True Image 9.0.3 (Acronis Team)	Linux/WinXP/Vista	Privativo
Sprite Clone 1.7.1 (Sprite Clone Team)	Win2000/XP	Privativo
Rembo Toolkit 4.0 (Rembo Technology y IBM Company 2005)	Linux/Win2003/XP	Privativo
Carbon Copy Cloner 3.1 (Carbon Copy Cloner Team)	Mac OS X	Privativo
Brutalix 1.1 (Brutalix Team)	Linux	GPL
CloneSys 1.3.0 (CloneSys Team)	Linux	GPL
NetRestore 3.4.3 (NetRestore Team)	Mac OS X	GPL
DriveClone 3.5 (Interbel)	Win2000/XP/2003/Vista	Privativo
DrvClonerXP 2.1 (DrvClonerXP Team)	Win2000/XP	Privativo
Image for Windows 1.70 (Image for Windows Team)	Win98/98SE/Me/2000/XP/Vista	Privativo
ADS (Automated Deployment Services) 1.1 (Microsoft Windows)	Win2000/2003/XP/SE	Privativo
Udpcast (Udpcast Team)	Linux	GPL
M23 (M23 Team)	Linux	GPL
Apple Software Restore (ASR) (Apple)	Mac OS X	Privativo
Drive SnapShot 1.3 (Drive SnapShot Team)	WinNT/2000/XP/2003/PE/x64/Vista	Privativo
FAI (Alejandro Valdés Villarrubia 2007)	Linux/Solaris	GPL
LUI (Alejandro Valdés Villarrubia 2007)	Linux	GPL
VA SystemInstaller (Alejandro Valdés	Linux/BSD/Unix-like OSes	GPL

Villarrubia 2007)			
SystemImager Team)	4.1.6 (SystemImager	Linux/BSD/Unix-like OSes	GPL

Debido a que son desarrollados para utilizarlos sobre sistemas operativos específicos, trae consigo que presenten diferentes características y que utilicen tecnologías, servicios y métodos de clonación diferentes para realizar sus funciones. A continuación se describen algunos de estos sistemas.

1.3.1. Brutalix 1.1.

Brutalix es una mini distribución GNU/Linux basada en DSL y Knoppix. A pesar de ocupar menos de 50 MB, incluye una amplia colección de software útil para la administración de aulas informáticas con gran cantidad de equipos. Es una alternativa, basada en herramientas libres, a las funciones de copia de imágenes y mantenimiento de computadoras de IBM Tivoli - Rembo. Brutalix permite la clonación automatizada de aulas mediante arranque PXE, soportando distribución de imágenes mediante unicast, multicast y p2p, además Brutalix usa el método de clonación **semi-consciente** para clonar computadoras.

El hecho de pesar tan poco permite cargar la imagen en memoria RAM, con lo que gana en velocidad. Además Brutalix incluye herramientas de clonación, reconocimiento de hardware, recuperación de datos dañados en discos, posee compatibilidad con diversos sistemas de ficheros de diversos sistemas operativos como Window, Linux y Mac OS, posee herramientas de particionamiento, recuperación de contraseñas, herramientas de testeo y configuración de red (escáners, esnifer, firewall, routers, etc), control remoto de computadoras y servidores FTP, SSH(SSH Communications Security), WEB, NFS, etc.

Además de todas estas herramientas Brutalix es una distribución como otra cualquiera. Tiene su escritorio y herramientas útiles para el uso diario. Tiene lector de correo electrónico, navegador web, herramientas ofimáticas, herramientas multimedia, impresión, pero su principal función es la clonación de sistemas operativos para mantener aulas informáticas.

1.3.2. Clonezilla 1.0.10-6.

Es una herramienta, con licencia GNU/GPL, para gestión y clonado de particiones. Se basa en DRBL, partimage, ntfsclose y udpcast. Clonezilla puede crear copias y recuperarlas desde múltiples sistemas operativos, ya sean particiones o discos duros completos, y se

basa en el método de clonación **semi-consciente** para clonar computadoras. Posee dos versiones en la actualidad Clonezilla-live y Clonezilla Server Edition. Clonezilla-live es adecuado para crear copias de seguridad y restaurarlas en un solo ordenador; mientras que Clonezilla Server Edition es para clonar masivamente un grupo de ordenadores simultáneamente. Clonezilla solo salva y restaura los bloques usados del disco duro. Durante una prueba que se realizo recientemente a la versión beta del producto se demostró que Clonezilla Server Edition fue capaz de clonar 41 computadoras simultáneamente; le tomo 10 min. clonar 5.6 Gbytes usando multicast.

Características de Clonezilla:

- ◆ Es software libre (GPL).
- ◆ Soporta los sistemas de ficheros : ext2, ext3, reiserfs, xfs, jfs de GNU / Linux, FAT y NTFS de MS Windows. Para sistemas de ficheros no soportados se realizan una copia byte a byte del disco duro usando la aplicación dd de Linux.
- ◆ LVM2 (LVM no es la versión 1) bajo GNU / Linux es compatible.
- ◆ Usa multicast para clonar masivamente ordenadores de la red.
- ◆ Se basa en partimage, ntfsclose y dd para guardar y restaurar no solo particiones sino también un conjunto de discos.

1.3.3. Norton Ghost 14.0.

Es una herramienta para crear copias de respaldo de sistemas y archivos completos, mas específicamente es un programa para la clonación de sistemas operativos y el método que usa para clonar es la **clonación basada en la estructura**. Norton Ghost producido originalmente por Binary Research, pero adquirido por Symantec el 24 de junio del 1998. El nombre Ghost se originó como un acrónimo de "General Hardware-Oriented System Transfer".

Murray Haszard escribió Ghost en 1995, aprovechando la experiencia y copiando archivos producidos previamente por Binay Research. Inicialmente Ghost solo soportaba los sistemas de archivos FAT, aunque también podía copiar otros sistemas de archivos basándose en la copia sector a sector. Ghost añadió soporte para el sistema NTFS después en el año 1996 y en 1999 añadió el soporte para ext2 y posteriormente para ext3.

Características de Norton Ghost 14.0:

- ◆ Copia de respaldo del sistema completo (imagen de disco): realiza copias de respaldo

de todo el contenido de un disco duro o una partición.

- ◆ Copia de respaldo de archivos y carpetas: realiza copias de respaldo sólo de los archivos y las carpetas específicos que elija, en lugar de guardar un disco completo.
- ◆ Copia de respaldo en FTP: realiza copias de puntos de recuperación en sitios FTP para facilitar la administración de las copias de respaldo en ubicaciones externas.
- ◆ Copia de respaldo externa: realiza copias de respaldo de los archivos en dispositivos de almacenamiento conectados a la red (NAS).
- ◆ Integración con Symantec ThreatCon: activa la creación de copias de respaldo incrementales cada vez que ThreatCon alcanza el nivel de amenaza especificado. Ideal para equipos sin protección contra virus.
- ◆ Copia de respaldo incremental y diferencial: realiza copias de respaldo sólo de los archivos modificados.
- ◆ Integración con Google Desktop: hace que la recuperación de datos sea incluso más rápida mediante índices de copias de respaldo con los que se pueden realizar búsquedas.
- ◆ LightsOut Restore: restaura el sistema con un entorno de recuperación de software en disco, sin que sea necesario un CD de arranque.
- ◆ Administración remota: controla otras instalaciones de Norton Ghost (12.0 o superior) de la red de área local.
- ◆ Compresión y cifrado avanzados: minimiza el espacio de almacenamiento y protege los documentos confidenciales.
- ◆ Distribuye e instala sistemas operativos de manera remota.

Norton Ghost es una excelente herramienta de clonación, pero tiene un gran inconveniente: no es libre. Por tanto para poder gozar de este magnifico servicio el usuario debe comprar el producto y solo puede adaptarlo a sus necesidades pagando a la empresa para que desarrolle un producto nuevo.

1.3.4. Rembo Toolkit 4.0.

Es una potente y versátil herramienta para la clonación y distribución de sistemas operativos y se basa en el método de la **clonación basada en la estructura**. Rembo Toolkit permite personalizar la clonación, el despliegue y los procesos de recuperación; ha sido diseñado prestando especial interés en el rendimiento y como resultado se obtuvo una herramienta que posee un control total del hardware y de los recursos de red.

Características de Rembo Toolkit 4.0:

- ◆ Clona y despliega sistemas operativos como Windows y UNIX.
- ◆ Particiona, navega y limpia disco duros.
- ◆ Crea paquetes de software y es capaz de reconocer las diferencias.
- ◆ Incorpora imágenes de discos con paquetes de software sobre la marcha.
- ◆ Es capaz de arrancar DOS, Windows 2000/XP/PE y UNIX en la Memoria de Acceso Aleatorio (RAM).
- ◆ Busca y modifica archivos en particiones NTFS, FAT, EXT2/3.
- ◆ Soporta las últimas tecnologías disponibles como PXE, Wake-On-Lan, multicast y DVD-Rom.
- ◆ Posee una aplicación cliente y un servidor. El lado del cliente se divide en un agente de arranque y un agente de sistema operativo (SO).
- ◆ El lado del cliente se encuentra embebido en la aplicación servidor y se envía a las computadoras a través de una conexión de red (red de arranque, o de arranque PXE) o mediante CD-ROM, disquete o una partición de disco duro.
- ◆ Incluye una interfaz gráfica que permite a los administradores hacer el trabajo de forma más amigable y fácil.
- ◆ Tiene implementados sus propios protocolos encriptados para transferir información desde el servidor Rembo y protocolos multicast que permiten arrancar múltiples ordenadores clientes a la vez sin que esto produzca ninguna sobrecarga en la red.

Es importante hacer notar que Rembo ofrece la posibilidad de restaurar solamente los ficheros que han cambiado (sincronización), crear imágenes de disco diferenciales y "mezclarlas" en tiempo de restauración.

En mayo/2006 IBM adquiere la empresa Rembo Technology, una operación que le permitió reforzar sus ofertas de herramientas de gestión Tivoli y de virtualización. La intención de IBM era incorporar las operaciones de Rembo en la división de software Tivoli, e integrar su tecnología en la plataforma Tivoli Provisioning Manager y en IBM Director software. Además, ofrecerá los productos Rembo como parte del software, hardware y servicios Virtualization Engine para ayudar a las empresas a gestionar entornos virtualizados, reforzando así su posición en el mercado de rápido crecimiento.

1.3.5. ADS (Automated Deployment Services) 1.1.

Los Servicios de Implantación Automatizados 1.0 (Automated Deployment Services 1.0, ADS) es un nuevo conjunto de herramientas desarrolladas para Windows Server 2003 y facilita la creación y administración de entornos de servidores Windows muy grandes, a escalas considerables. ADS incluye un nuevo conjunto de herramientas de imagen desarrolladas por Microsoft y una infraestructura con seguridad mejorada para una rápida implantación tanto de Windows 2000 Server como de Windows Server 2003 de forma remota en servidores "limpios". Los Servicios de Implantación Automatizados también ofrecen un marco de ejecución remota de scripts fiable que permite al administrador realizar administración basada en scripts con seguridad sobre cientos de servidores, con la misma facilidad que se hacía sobre un servidor único.

Los Servicios de Implantación Automatizados de Windows Server 2003 combinan la velocidad de las tradicionales herramientas de imagen con la flexibilidad de una instalación basada en scripts, para ofrecer los siguientes beneficios clave:

Beneficio	Funcionalidad
Reducción drástica del coste de implantación del servidor.	Un servidor de Entorno de Ejecución Pre-arranque (<i>Preboot Execution Environment, PXE</i>) inteligente y un agente de implantación creado dinámicamente permiten a los servidores remotos crear a partir de equipos "limpios" compatibles con PXE, reduciendo el coste asociado a la implantación de servidores.
Consistencia en el aprovisionamiento del servidor, menos errores humanos.	A través de una potente automatización de tareas dirigida por pasos de ejecución, la secuencia de tareas de ejemplo se pueden ampliar y adaptar para automatizar la configuración del hardware, la implantación del sistema operativo y la instalación de las aplicaciones, permitiendo codificar las prácticas operacionales de una organización y eliminar errores humanos.
Flexibilidad y agilidad a través de nuevas herramientas de imagen.	Nuevas y potentes herramientas creadas por Microsoft utilizan el conocimiento de la estructura de sistema de ficheros NTFS para crear imágenes más pequeñas que se pueden actualizar y editar sin necesidad de haber sido previamente implantadas en un servidor.
Potente administración masiva de servidores.	A través del entorno de ejecución remota fiable, los Servicios de Implantación Automatizados mejoran las inversiones en

	desarrollo de scripts ya existentes y extienden su capacidad de administrar cientos de servidores.
Configuración más sencilla del hardware.	Utilizando Virtual Floppy, los Servicios de Implantación Automatizados incorporan herramientas de MS-DOS de fabricantes de servidores estándar dentro del proceso de implantación para automatizar la configuración del hardware.
Fácil integración gracias a la posibilidad de elección entre diferentes interfaces de usuario.	ADS ofrece una interfaz gráfica de usuario sencilla de usar, un conjunto de herramientas de línea de comandos y una rica interfaz de programa para la Instrumentación de Administración de Windows (Windows Management Instrumentation, WMI). Los Servicios de Implantación Automatizados permiten el funcionamiento "point-and-click" o la integración con otras soluciones del proceso de implantación.
Registro consistente de la historia administrativa.	Los Servicios de Implantación Automatizados ofrecen un almacenamiento de datos centralizado para mantener una completa historia de todas las tareas administrativas realizadas utilizando la infraestructura de los Servicios de Implantación Automatizados.

1.4. Tendencias de los sistemas de clonación y distribución de software.

Las tecnologías de la información están en constante desarrollo, las cuales están determinadas por una combinación de influencias tecnológicas y de nuevos requerimientos por parte de los usuarios finales. Estas influencias actúan sobre todos los componentes de las tecnologías existentes y establecen el comportamiento futuro de cada una de ellas.

1.4.1. Reseña histórica.

En el mundo de la informática se avanza con gran rapidez y es muy difícil medir sus cambios y determinar su significado. Lo que resulta evidente es que en la medida en que se desarrolla el software surgen nuevas aplicaciones mucho más extensas y complejas. Los sistemas de clonación y distribución de software no quedan por supuesto fuera de esta evolución tecnológica los cuales están en constante perfección y desarrollo atendiendo siempre las necesidades del cliente.

Hace muchos años atrás no se manejaba el concepto de clonación de sistemas operativos, los ordenadores se encontraban aislados y ni tan siquiera se manejaba el concepto de red de ordenadores. El 29 de octubre del 1969 surge el primer prototipo de internet con el envío del primer mensaje a través de Arpanet (DOD) lo cual significó la posibilidad de comunicar un grupo de computadoras, aunque no existía aún la idea de clonación de múltiples clientes.

Por estos años, algunos fabricantes de computadoras utilizaban máquinas para copiar el contenido de un disco duro a otro. Esto tenía la desventaja de copiar no solo la información usada en el disco duro sino también los sectores no utilizados ya que el hardware usado no tenía conocimiento de las estructuras de los discos. Un disco duro de mayor tamaño no podía ser copiado a uno de menor tamaño y tampoco se podía copiar uno de menor tamaño a otro de mayor tamaño, los discos debían ser idénticos y de iguales geometrías.

Los sistemas operativos UNIX fueron los primeros en dar pasos de avances en cuanto a clonación de discos duros. Durante la creación de UNIX en el año 1973 sus desarrolladores necesitaron un comando de bajo nivel para copiar datos entre dispositivos y crearon dd (duplicador de datos), aunque aún no se podía duplicar información para un grupo de computadoras conectadas a una red local, al menos con dd se podía crear salvadas del sistema operativo y duplicarlo en la misma computadora. Dd era capaz de realizar una copia exacta de la información que existía en un disco duro y la replicaba en el otro disco; así surge la clonación basada en sectores, "dd" usaba el **método de clonación a ciegas**.

Para mediados de los 80's comienza el auge de las redes de computadoras y la necesidad de sistemas operativos en red y sistemas operativos distribuidos y con ello surge también la necesidad de poder replicar un sistema operativo en múltiples computadoras. La red mundial Internet se va haciendo accesible a toda clase de instituciones y se comienzan a dar múltiples soluciones entre ellas está el surgimiento de un nuevo protocolo a nivel de aplicación, según el Modelo OSI; NFS (Network File System) el cual posibilita que distintos ordenadores conectados a una misma red puedan acceder a ficheros remotos como si se trataran de locales, fue desarrollado en 1984 por Sun Microsystems. NFS vino a ser una importante herramienta que se comenzó a usar en muchos sistemas de clonación y distribución de software, posibilitando clonar de manera simultánea un grupo de computadoras de una misma red basándose en un modelo Cliente/Servidor.

En el 1995 Murray Hazzard escribió Ghost el cual vendría a ser el pionero de los sistemas de clonación y distribución de software; fue publicado por Binary Research en 1995 y vendido a Symantec el 24 de Junio del 1998 donde pasó a tener el nombre de Norton Ghost. Ghost fue

todo un éxito justo cuando Windows 95 creaba una gran demanda para la clonación de discos. Ghost fue uno de los primeros sistemas de clonación que mejoró la clonación a ciegas para dar lugar a un clonado con **semi-conocimiento del sistema de archivos**. Inicialmente soportaba solamente los sistemas de ficheros FAT. En 1996 añadió soporte para NTFS y en 1999 incluye el soporte para ext3.

El surgimiento de Ghost trajo consigo competencia y por ende el surgimiento de diferentes sistemas de clonación que realizaban funciones similares a Ghost y soportaban diferentes sistemas de ficheros soportados sobre diferentes sistemas operativos.

El desarrollo continuo y el conocimiento aportado de los sistemas de archivos trajo consigo el surgimiento de un nuevo paradigma de clonación el cual es muy usado en la actualidad por diferentes software de clonación. La clonación basada en la estructura eliminó los mayores problemas de los métodos tradicionales y apporto mayor facilidad en el manejo de la imagen del sistema operativo a clonar.

1.4.2. Principales tendencias.

Después de un profundo estudio realizado se puede llegar a la conclusión de que los sistemas de clonación de manera general en la actualidad tienen como principales tendencias las mencionadas a continuación :

- ◆ Tienen a estar integrados en muchos paquetes para la gestión empresarial como herramienta principal para el mantenimiento de software. Ejemplo: Rembo Toolkit -> IBM Tivoli.
- ◆ Además tienen a usar el método de la clonación basada en la estructura, permitiendo esto un mejor control de los elementos del sistema operativo (véase sección 1.2.2). Ejemplo : Norton Ghost, Rembo Toolkit, Acronis True Image, etc.
- ◆ Tienen a incluir entre sus funciones el del procesamiento estadístico de información asociada a hardware y software de cada computadora cliente, reconocida después de un proceso de instalación o clonación. Lo cual sirve para la realización de auditorías de hardware y software. Ejemplo: Rembo Toolkit, IBM Tivoli, etc.
- ◆ Tienen a estar incluidas en mini distribuciones de sistema operativos. Ejemplo: Brutalix.

1.5. Herramientas, lenguajes y tecnologías a utilizar.

En el desarrollo de todo sistema informático es de vital importancia la selección de las

herramientas, lenguajes y tecnologías a utilizar, paso que garantizará, de realizarse correctamente, un óptimo desempeño del sistema. Para el desarrollo del sistema al cual se refiere este trabajo, la selección se realizó teniendo en cuenta la infraestructura tecnológica de la UCI y valorando que el sistema a desarrollar, estaría orientado a funcionar sobre el sistema operativo GNU/Linux.

1.5.1. Code::blocks

Code::Blocks (Code::Blocks Team), es una herramienta de entorno de desarrollo integrado (IDE en inglés) para el desarrollo de programas en lenguaje C++. Está basado en la plataforma de interfaces gráficas WxWidgets, lo que le permite correr libremente en diversos sistemas operativos, y es de licencia GPL.

Debido a que Dev-C++ es un IDE para programar en C y C++ y está creado en Delphi, surgió la idea y necesidad de crear un IDE hecho en los lenguajes adecuados: C y C++. Con esta motivación se creó el IDE Code::Blocks.

Características.

Soporte de Compiladores:

Debido a que en sí Code::Blocks es sólo la interface del entorno de desarrollo, puede enlazarse a una variedad de compiladores para poder desarrollar su trabajo. Por defecto, Code::Blocks buscará una serie de compiladores y configurará los que halle. En particular, usa MinGW.

Otros compiladores soportados:

- ◆ Microsoft Visual Studio Toolkit (una extensión de compilador de C++ de Microsoft).
- ◆ GCC, en sus versiones para Microsoft (ya sea MinGW o Cygwin) y Linux.
- ◆ Digital Mars Compiler.
- ◆ Intel C++ Compiler.

Todos estos compiladores pueden ser detectados automáticamente si están ya instalados al iniciar Code::Blocks.

Aunque no está oficialmente soportado (producto de su bajo nivel de adhesión a la norma de C++), Microsoft Visual Studio 6 puede ser configurado y utilizado, aunque no con muy buenos resultados.

Características de Edición :

Entre otras, Code::Blocks soporta varias de las características ya consideradas "clásicas" y que sirven de apoyo a la programación, como el coloreo de sintaxis, el autocódigo (generar plantillas de código para proyectos), el autocompletado, etc.

Compatibilidad con Bibliotecas:

Code::Blocks trae integradas plantillas para generar varias clases de programas, ya sea la clásica aplicación de consola, bibliotecas estáticas o dinámicas, o proyectos completos enlazados con populares bibliotecas como OpenGL y SDL; sin embargo, Code::Blocks integra sólo las plantillas, las bibliotecas deben instalarse por separado.

Más Características:

Otras de las características de Code::Blocks:

- ◆ Importación de proyectos Visual C++ y Dev-C++.
- ◆ Soporte para packs del Dev-C++.
- ◆ Inclusión y generación de plug-ins.
- ◆ Vista de Árbol de Proyectos.
- ◆ Estadísticas y Resumen de código.
- ◆ Generación de XML para proyectos.
- ◆ Exportación a formatos XML, RTF y de OpenOffice.org.

1.5.2. Anjuta.

Anjuta(Anjuta Team) es un entorno integrado de desarrollo (IDE) desarrollado por Naba Kumbar en 1999. Fue desarrollado para programar en C y C++ en sistemas GNU/Linux aunque actualmente admite lenguajes como C#, Perl y Python. Su principal objetivo es trabajar con GTK y en el escritorio GNOME, además ofrece un gran número de características avanzadas de programación. Anjuta es un software libre, liberado bajo la licencia GPL.

Anjuta incluye:

- ◆ Administrador de proyectos.
- ◆ Asistentes.

- ◆ Plantillas.
- ◆ Depurador interactivo.
- ◆ Editor de texto; que verifica y resalta las sintaxis escritas.
- ◆ Extensión para Subversion (control de versiones).
- ◆ Interprete de comandos propio.

A pesar de incluir abundantes utilidades y tener una amplia variedad de funcionalidades es un sistema bastante estable y muy ligero en el consumo de recursos.

1.5.3. wxWidgets.

Fue diseñado en 1992 en el Instituto de Aplicaciones de Inteligencia Artificial (Artificial Intelligence Applications Institute), en la Universidad de Edinburg, por Julian Smart. Inicialmente se llamaba wxWindows pero tuvo que ser cambiado al nombre wxWidgets debido a que Microsoft interpuso una demanda a finales del 2003 por una posible confusión con el nombre de su Sistema Operativo.

Las wxWidgets (wxWidgets Team) son unas bibliotecas multiplataforma y libres, para el desarrollo de interfaces gráficas programadas en lenguaje C++. Están publicadas bajo licencia Open Source y GNU LGPL, similar a la GPL con la excepción de que el código binario producido por el usuario a partir de ellas, puede ser propietario, permitiendo desarrollar aplicaciones empresariales sin coste.

Las WxWidgets proporcionan una interfaz gráfica basada en las bibliotecas ya existentes en el sistema (nativas), con lo que se integran de forma óptima y resultan muy portables entre distintos sistemas operativos. Están disponibles para Windows, MacOS, GTK+, Motif, OpenVMS ,OS/2, también en plataformas móviles como Microsoft Pocket PC y Palm OS. También pueden ser utilizadas desde otros lenguajes de programación, aparte del C++: Java, Javascript, Perl, Python, Smalltalk, Ruby .

Cuenta con una parte denominada wxBase que incluye clases como wxString, clases para el manejo de archivos y directorios de manera independiente del sistema, funcionalidades como: gráficos 2D, 3D con OpenGL, Bases de Datos (ODBC), Redes, Impresión, Hilos, visión e impresión del HTML, un sistema de archivos virtual y cuenta con algunos IDEs.

1.5.4. Sfdisk.

Es una aplicación que se encarga de manipular las tablas de particiones, suministra

información sobre las particiones, comprueba las particiones en un dispositivo y con él es posible crear o eliminar particiones y unidades lógicas y definir la partición activa, si es que no lo está. Presenta la ventaja de que se puede automatizar el proceso, creando un script y pasándole todos los parámetros necesarios para realizar el proceso de una forma simple y rápida.(sfdisk Team)

1.5.5. Partimage.

Partimage(Francois Dupoux y Franck Ladurelle) es una aplicación desarrollada para el trabajar sobre el sistema operativo GNU/LINUX. Esta aplicación fue desarrollada para guardar particiones de cualquier sistema de ficheros en un archivo de imagen. La imagen puede ser comprimida en formato GZIP/BZIP2, con tal de ocupar el mínimo espacio en el disco duro, también brinda la posibilidad de dividir este archivo imagen en múltiples ficheros para poder ser copiados en diferentes discos de almacenamiento. Partimage también tiene la funcionalidad de realizar el proceso contrario; teniendo un archivo imagen, instalarlo en una partición vacía.

Partimage soporta los siguientes sistemas de ficheros:

- ◆ ReiserFS: Un sistema de archivos utilizado por Linux y Unix.
- ◆ ext2fs/ext3fs: El sistema de archivos estándar de Linux.
- ◆ FAT16/32: El sistema de archivos de DOS & Windows 9x.
- ◆ NTFS: El sistema de archivos de Windows NT, 2000 y XP.
- ◆ HPSF: El sistema de archivos de IBM OS/2.
- ◆ JFS: Creado por IBM y usado en Aix (soporte beta).

1.5.6. NFS.

Network File System (Sistema de archivos de red), o NFS, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales. Originalmente fue desarrollado en 1984 por Sun Microsystems, con el objetivo de que sea independiente de la máquina, el sistema operativo y el protocolo de transporte, esto fue posible gracias a que está implementado sobre los protocolos XDR (presentación) y ONC RPC (sesión) . El protocolo NFS está incluido por defecto en los Sistemas Operativos UNIX y las distribuciones

Linux.

Características:

- ◆ El sistema NFS está dividido al menos en dos partes principales: un servidor y uno o más clientes. Los clientes acceden de forma remota a los datos que se encuentran almacenados en el servidor.
- ◆ Las estaciones de trabajo locales utilizan menos espacio de disco debido a que los datos se encuentran centralizados en un único lugar pero pueden ser accedidos y modificados por varios usuarios, de tal forma que no es necesario replicar la información.
- ◆ Los usuarios no necesitan disponer de un directorio “home” en cada una de las máquinas de la organización. Los directorios “home” pueden crearse en el servidor de NFS para posteriormente poder acceder a ellos desde cualquier máquina a través de la infraestructura de red.
- ◆ También se pueden compartir a través de la red dispositivos de almacenamiento como disqueteras, CD-ROM y unidades ZIP. Esto puede reducir la inversión en dichos dispositivos y mejorar el aprovechamiento del hardware existente en la organización.
- ◆ Todas las operaciones sobre ficheros son síncronas. Esto significa que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esa operación. En caso de una solicitud de escritura, el servidor escribirá físicamente los datos en el disco, y si es necesario, actualizará la estructura de directorios, antes de devolver una respuesta al cliente. Esto garantiza la integridad de los ficheros.

1.5.7. Bash.

Bash (GNU Project) es un intérprete de comandos de tipo Unix (shell) escrito para el proyecto GNU. Su nombre es un acrónimo de Bourne-Again Shell. Es el shell por defecto en la mayoría de las distribuciones de GNU/Linux de hoy en día. Se encarga de interpretar las órdenes que le damos para su proceso por el kernel. Es el encargado de leer los caracteres tecleados por los usuarios, los interpreta y los ejecuta. Al escribir un comando, es el Shell y no el S.O., quien se encarga de interpretarlo y ordenar que se ejecute.

1.5.8. Perl.

Perl(Larry Wall) (Lenguaje Práctico para la Extracción e Informe) es un lenguaje de programación diseñado por Larry Wall y creado en 1987. Perl fue originalmente desarrollado para la manipulación de texto y en la actualidad es ampliamente utilizado en la

administración de sistemas y en el desarrollo Web. Perl toma características del C y del lenguaje interpretado bash. La estructura completa de perl deriva de lenguaje C, perl es un lenguaje imperativo, con variables, expresiones, asignaciones y bloques de códigos delimitados por llaves. Perl tiene una gran potencia en la manipulación de textos debido a que incluye expresiones regulares que facilitan el trabajo con textos.

1.5.9. Lenguaje C/C++.

C es un lenguaje de programación creado en 1969 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Se trata de un lenguaje tipado de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

C es un lenguaje muy eficiente puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas. A pesar de su bajo nivel es el lenguaje más portado en existencia, habiendo compiladores para casi todos los sistemas conocidos. Proporciona facilidades para realizar programas modulares y/o utilizar código o bibliotecas existentes.

Entre las características de C las más importantes son:

- ◆ Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de ficheros, proporcionadas por bibliotecas.
- ◆ Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado no llevado al extremo (permitiendo ciertas licencias rupturistas).
- ◆ Un sistema de tipos que impide operaciones sin sentido.
- ◆ Usa un lenguaje de preprocesado, el preprocesador de C, para tareas como definir macros e incluir múltiples ficheros de código fuente.

- ◆ Acceso a memoria de bajo nivel mediante el uso de punteros.
- ◆ Un conjunto reducido de palabras clave.
- ◆ Los parámetros se pasan por valor. El paso por referencia se puede simular pasando explícitamente el valor de los punteros.
- ◆ Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.
- ◆ Tipos de datos agregados (struct) que permiten que datos relacionados se combinen y se manipulen como un todo.

El **C++** es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C. Las principales características del C++ son el soporte para programación orientada a objetos y el soporte de plantillas o programación genérica (templates). Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos. Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel:

- ◆ Posibilidad de redefinir los operadores (sobrecarga de operadores)
- ◆ Identificación de tipos en tiempo de ejecución (RTTI)

C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatismos trae (obliga a hacerlo casi todo manualmente al igual que C) lo que "dificulta" mucho su aprendizaje. (Ken Thompson y Dennis M. Ritchie)

1.5.9.1. Ncurses.

Ncurses(GNU Project) es una librería de programación que provee una API que permite al programador escribir interfaces basadas en texto, posibilitando la construcción de una Interfaz para el usuario, para aplicaciones ejecutadas en consola.

1.5.9.2. Socket.

Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada. Un socket queda definido por una dirección IP, un protocolo y un número de puerto.

Los sockets permiten implementar una arquitectura cliente-servidor. La comunicación ha de ser iniciada por uno de los programas que se denomina programa cliente. El segundo programa espera a que otro inicie la comunicación, por este motivo se denomina programa servidor.

Un socket es un fichero existente en la computadora cliente y en la computadora servidor, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

1.5.9.3. Hilos.

Un hilo de ejecución, en sistemas operativos, es una característica que permite a una aplicación realizar varias tareas concurrentemente. Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación, etc. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

1.5.9.4. Semáforos.

Un semáforo es una variable especial protegida (o tipo abstracto de datos) que constituye el método clásico para restringir o permitir el acceso a recursos compartidos (por ejemplo, un recurso de almacenamiento del sistema o variables del código fuente) en un entorno de multiprocesamiento (en el que se ejecutarán varios procesos concurrentemente). Fueron inventados por Edsger Dijkstra y se usaron por primera vez en el sistema operativo THEOS.

1.5.10. LTSP.

Linux Terminal Server Project o **LTSP** son un conjunto de aplicaciones servidores que proporcionan la capacidad de ejecutar Linux en computadoras remotas. El sistema de funcionamiento del LTSP consiste en repartir por medio de la red el núcleo de Linux que es ejecutado por los clientes y que posteriormente ejecutaran secuencias de scripts típicos de una mini distribución. Los clientes podrán acceder a las aplicaciones por medio de una consola textual o por un servidor gráfico que se comparte utilizando el protocolo XDMCP.

1.5.11. DHCP.

DHCP (sigla en inglés de Dynamic Host Configuration Protocol) es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes

conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

1.5.12. TFTP.

TFTP son las siglas de Trivial File Transfer Protocol (Protocolo de transferencia de archivos trivial). Es un protocolo de transferencia muy simple semejante a una versión básica de FTP. TFTP a menudo se utiliza para transferir pequeños archivos entre computadoras en una red, como cuando un terminal X Window o cualquier otro cliente ligero arranca desde un servidor de red.

Características:

- ◆ Utiliza UDP (puerto 69) como protocolo de transporte (a diferencia de FTP que utiliza el puerto 21 TCP).
- ◆ No puede listar el contenido de los directorios.
- ◆ No existen mecanismos de autenticación o cifrado.
- ◆ Se utiliza para leer o escribir archivos de un servidor remoto.
- ◆ Soporta tres modos diferentes de transferencia, "netascii", "octet" y "mail", de los que los dos primeros corresponden a los modos "ascii" e "imagen" (binario) del protocolo FTP.

1.5.13. UML.

La Tecnología de Orientación a Objetos constituye la base de la reutilización de código por medio de componentes. UML (Lenguaje de Modelación Unificado o Unified Modeling Language, en inglés) es el lenguaje estándar adoptado por el OMG (Object Management Group) y mundialmente aceptado para la descripción de los "planos" de software. UML es un lenguaje gráfico para visualizar, especificar, construir y documentar cada una de las partes que comprende el desarrollo de software. UML posee formas de modelar conceptos como lo son procesos de negocio y funciones de sistema, además de aspectos concretos como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

1.5.14. Postgresql.

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado

Ingres en la Universidad Berkeley de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation.

En 1986 otro equipo dirigido por Michael Stonebraker de Berkeley continuó el desarrollo del código de Ingres para crear un sistema de bases de datos objeto-relacionales llamado Postgres. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, Postgres fue renombrado a PostgreSQL, tras un breve periplo como Postgres95. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

Características Principales :

- ◆ DBMS Objeto-Relacional : PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transactions, optimización de consultas, herencia, y arrays.
- ◆ Altamente extensible : PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.
- ◆ SQL comprensible : PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- ◆ Integridad referencial : PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- ◆ API flexible : La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.
- ◆ Lenguajes procedurales : PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- ◆ MVCC (Multi-Version Concurrency) : MVCC, o Control de Concurrencia Multi-Versión, es

la tecnología que PostgreSQL usa para evitar bloqueos innecesarios.

- ◆ Cliente/Servidor : PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- ◆ WAL (Write Ahead Logging) : Incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos.

1.6. Metodologías ágiles utilizadas.

Todo desarrollo de software(Sanchez, M. A. M 2002) es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo. Cuando los proyectos que se van a desarrollar son de mayor envergadura, ahí si toma sentido el basarse en una metodología de desarrollo, y empezar a buscar cual sería la más apropiada para el desarrollo del software.

Las metodologías ágiles(José H. Canós, Patricio Letelier, y Msc Carmen Penadés) forman parte del movimiento de desarrollo ágil de software conocidos anteriormente como metodologías livianas, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil como la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Las Metodologías Ágiles se basan en los siguientes principios :

- ◆ Realizar entregas cortas en el tiempo y continuas.
- ◆ Dar la bienvenida a los cambios.
- ◆ Entregas periódicas y frecuentes que funcionen.
- ◆ Los clientes forman parte del equipo de desarrollo.
- ◆ Equipo con individuos motivados. Darles para ello el ambiente, apoyo y confianza.

- ◆ La comunicación directa es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo. Intenta evitar el teléfono, correos electrónicos, fax, etc.
- ◆ La medida principal de progreso es el software que funciona.
- ◆ Desarrollo sostenible. Es indispensable que exista paz y armonía en el equipo para que el proyecto tenga éxito.
- ◆ Buen diseño y calidad técnica.
- ◆ La simplicidad es algo básico.
- ◆ Equipos autorganizados.
- ◆ El equipo debe realizar reflexiones periódicamente para plantearse cómo llegar a ser más efectivo.

Entre las más conocidas están : (Scott W. Ambler)

1. XP (Extreme Programming).
2. SCRUM.
3. Metodología Crystal.
4. Dynamic Systems Development Method (DSDM).
5. Adaptive Software Development (ASD).
6. Feature -Driven Development (FDD).
7. Lean Development (LD).
8. BUP.
9. AUP.

Para el desarrollo de este trabajo se propone el uso de dos de las muchas metodologías ágiles enunciadas anteriormente, SCRUM para la parte de planificación, y XP para la parte de desarrollo.

1.6.1. Scrum.

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La

segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. Scrum no esta definida como metodología independiente, sino como un complemento a otras metodologías como lo son XP y MSF. Como método, Scrum enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas; de allí su deliberada insuficiencia y su complementariedad. En Scrum se fomentan valores como:

- ◆ Equipos auto-dirigidos y auto-organizados. No hay Director que decida; la excepción es el Scrum Master que debe ser 50% programador y que resuelve problemas, pero no manda.
- ◆ Una vez elegida una tarea, no se agrega trabajo extra. En caso que se agregue algo, se recomienda quitar alguna otra cosa.
- ◆ Encuentros diarios con las tres preguntas siguientes:
 - ✓ ¿Qué es lo que se hizo el día anterior?
 - ✓ ¿Qué es lo que se va a hacer hoy?
 - ✓ ¿Qué impedimentos tengo para realizar mi trabajo?
- ◆ Iteraciones de treinta días; se admite que sean más frecuentes.
- ◆ Demostración a participantes externos al fin de cada iteración.
- ◆ Al principio de cada iteración, planeamiento adaptativo guiado por el cliente.

1.6.2. XP.

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo y corto equipo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Esta metodología se basa en:

- ◆ **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantando en algo hacia el futuro, se pueden hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- ◆ **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ◆ **Programación en pares:** una particularidad de esta metodología es que propone la

programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- ◆ Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- ◆ El manejo del cambio se convierte en parte sustantiva del proceso.
- ◆ El costo del cambio no depende de la fase o etapa.
- ◆ No introduce funcionalidades antes que sean necesarias.

1.7. Conclusiones.

En este capítulo se ha cumplido su objetivo principal, que fue mostrar al lector de forma general los aspectos teóricos a tener en cuenta para el desarrollo de un sistema de clonación y distribución de imágenes de sistemas operativos. Con la intención de ubicarlo y sumergirlo en el mundo de los servicios de soporte e instalación remota, así como mostrarle los diferentes sistemas existentes actualmente, sus características, funcionamiento y los métodos que emplean para la clonación.

A la vez se ha hecho un estudio de las tendencias futuras de los sistemas de clonación y se realizó una valoración de las herramientas, tecnologías y lenguajes propuestos a utilizar para el futuro desarrollo del sistema a proponer en este trabajo así como se escogió también la metodología de software que se empleará.

2. CAPÍTULO II. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

2.1. Introducción

En este capítulo se realizará el análisis propuesto por el analista para la solución del problema. Se analizarán posibles componentes ya existentes los cuales pueden ser utilizados nuevamente así como la estrategia de integración al sistema.

2.2. Valoración crítica del diseño propuesto por el analista.

El diseño propuesto por el analista fue valorado como bueno, fácil de entender y muy específico pues permitió adquirir una comprensión de los aspectos relacionados con los requisitos funcionales y no funcionales del sistema, que sirvieron de base para el comienzo del desarrollo del mismo, así como también permitió acotar el desarrollo a los lenguajes de programación más óptimos y adecuados para programar el sistema.

Se obtuvieron además, diagramas como los de clases de diseño que mostraron la parte estática del sistema, la representación de las clases y sus relaciones cosa que facilita en gran medida la implementación de las mismas dada sus definiciones.

Unido a esto se encuentran los diagramas de interacción, que explican la colaboración que existe entre las clases y cómo son llamados los métodos y sentencias dentro de cada una, de los cuales se obtiene la información necesaria para conocer el orden de las acciones a implementar. Mientras más legible sea el diseño propuesto más fácil es la adaptación de los programadores al mismo y por ende se agiliza el proceso de traducción de clases del diseño a clases de implementación.

No obstante tras un análisis realizado en el presente año (2008) a los aplicaciones propuestas por el analista para llevar a cabo el desarrollo del **SistClon** (Sistema de Clonación y Distribución de Imágenes de Sistemas Operativos) se decidió sustituir el uso de algunas de ellas por otras encontradas recientemente que realizan un trabajo más completo y más óptimo que las propuestas.

Se decidió reemplazar el uso del fdisk como aplicación propuesta para realizar el particionado de cada PC cliente por el sfdisk debido a cuestiones técnicas como son:

- ♦ La información que sfdisk brinda de las particiones es más completa.
- ♦ El sfdisk permite pasar como parámetro un fichero que contenga la estructura de la

tabla de particiones, mientras que el fdisk no posee esta opción.

- ◆ También, en la misma ayuda del fdisk se dice que el sfdisk es más poderoso y más correcto que el fdisk e incluso que el cfdisk.
- ◆ La forma de construir el fichero para particionar una PC cliente de manera adecuada y según las solicitudes del administrador se hace mucho más fácil y versátil con el sfdisk que usando el fdisk.

Se decidió también no usar el SSH para el envío de comandos y scripts simples a cada cliente debido a que :

- ◆ El proceso de comunicar dos puntos o más en una subred interna se hace un poco complejo usando SSH ya que se debe configurar cada cliente de forma manual para que acepte comandos y scripts del servidor y el objetivo principal es lograr interactuar lo menos posible con cada cliente de manera manual.
- ◆ El envío de comandos y scripts a una o dos computadoras de una subred con ssh es relativamente rápido, sin embargo cuando tenemos 30 clientes conectados al servidor el envío de comandos y scripts a través del ssh, pierde velocidad.
- ◆ Además el sistema en si en su totalidad necesitaba un control de cada cliente conectado al servidor para poder mostrarlo en la interfaz gráfica y poder permitirle al usuario una forma fácil de interactuar con cada cliente de manera individual o total y con el SSH se hacía un tanto difícil integrarlo a una interfaz amigable al usuario final.

Debido a estos inconvenientes presentados con el SSH se decide implementar una aplicación que resolviera no solo los problemas explicados anteriormente sino también que se integrara en su totalidad como la base principal del **SistClon** para la comunicación con todos los clientes de una manera dinámica, fácil y configurable para los fines del presente desarrollo. Surge entonces el RESC (Remote Execute Scripts and Command) como una aplicación que estaría integrada completamente en la interfaz gráfica del producto final y que permitiera controlar cada cliente de una manera óptima. Con el RESC el envío de información a cada cliente se hace mucho más rápido y puede soportar hasta 200 clientes o más, conectados al mismo tiempo al servidor, además el cliente detecta de manera automática la ubicación del servidor y no necesita una configuración manual para conectarse al mismo.

Características del RESC:

- ◆ Posee una cola de mensajes que permite enviar múltiples mensajes al mismo tiempo.
- ◆ Esta dividido en dos programas uno del lado del servidor (RESC-Server) y el otro del

lado del cliente (RESC-Client).

- ◆ Posee un programa en los clientes llamado rumsg que es el encargado de ejecutar todas las órdenes del Servidor y el mismo es independiente del RESC-Client, lo cual permite mayor velocidad y evita cargar al Cliente con múltiples ejecuciones.
- ◆ Posee un protocolo de comunicación que permite ahorrar ancho de banda.
- ◆ Bloquea Conexiones con la misma dirección IP.
- ◆ Posee funciones que permiten mostrar los errores al usuario.
- ◆ Detecta cambios en la red de tal manera que si se pierde la conexión con el servidor, el cliente se bloquea hasta que se inicia el servidor otra vez y entonces se restablece la conexión.

(Ver Anexo No. 1)

Se decidió también eliminar el envío de la imagen a clonar haciendo uso del servidor partimaged debido a los siguientes problemas presentados en el mismo :

- ◆ Solo permite 15 clientes conectados al mismo tiempo al servidor y como mínimo un laboratorio docente posee 30 computadoras lo cual supera las posibilidades del servidor.
- ◆ No soporta más de una conexión haciendo peticiones de lectura o escritura debido a que posee problemas en el manejo de los hilos para manejar las operaciones de cada cliente.

Por tanto se decidió compartir la imagen a clonar por NFS y usar el partimage del lado del cliente para clonar el ordenador.

Además también se incorporó como herramientas para el desarrollo del **SistClon** el uso del Codeblocks como IDE (Entorno de Desarrollo Integrado) para programar la interfaz gráfica del sistema haciendo uso de wxwidgets como framework para desarrollo multiplataforma.

2.3. Análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados. Estrategia de integración.

La reutilización resulta sumamente importante pues brinda ventajas como el ser clave a la hora de reducir los tiempos y costes del desarrollo además permite generar aplicaciones eficientes y de gran fiabilidad.

Para dar solución al problema propuesto en el presente trabajo es necesario hacer uso de otros componentes ya existentes que vienen a ser de vital importancia para dar cumplimiento a los requerimientos del sistema.

Un componente importante es el LTSP (*ver sección 1.5.10*) el cual es usado para iniciar todos los clientes por la red enviándoles un mini sistema operativo con los scripts y aplicaciones necesarias para gestionar, configurar y clonar el ordenador cliente. La aplicación RESC-Cliente se encuentra entre las aplicaciones que posee el sistema de archivos exportado y se encarga de mantener al cliente conectado constantemente al servidor para recibir órdenes del mismo.

Internamente el LTSP hace uso de aplicaciones como son el DHCP (*ver sección 1.5.11*) para obtener un ip de manera dinámica, el TFTP (*ver sección 1.5.12*) para poder exportar el kernel del mini sistema operativo y finalmente el NFS (*ver sección 1.5.6*) para exportar el sistema de archivo que cada cliente obtendrá en el inicio por la red (*ver sección 1.2.1*) y el cual lleva consigo todas las aplicaciones que servirán para realizar el clonado total del sistema.

Otro componente importante es el sfdisk (*ver sección 1.5.4*) el cual se usa para gestionar el particionado de cada computadora cliente. El sfdisk recibe como parámetro de entrada un fichero que contiene la estructura de la tabla de particiones que el administrador del sistema configuró previamente de manera remota.

Después de particionado el cliente, se usa cualquiera de las opciones del mkfs para darle formato a las particiones de acuerdo al tipo de sistema de archivo especificado para cada partición.

También se hace uso del componente partimage (*ver sección 1.5.5*) como la aplicación encargada de clonar el sistema operativo que el administrador sirve por NFS. El partimage obtiene la dirección de la imagen que se desea clonar en una partición y se encarga de restablecerla en la partición especificada a clonar.

Estrategia de Integración.

Todos los componentes cumplen una tarea importante dentro del sistema en general; cada uno de ellos se interrelaciona entre sí de acuerdo a su propósito final en el sistema.

El LTSP brinda el mini sistema operativo que incluye un grupo de aplicaciones entre ellas el sfdisk y el partimage, además de otras como el RESC-Cliente. El RESC-Cliente es el encargado de obtener todas las órdenes del administrador y ejecutarlas. Cuando el

administrador envíe la estructura de la tabla de particiones que el desea para cada cliente el RESC-Client se lo comunica al sfdisk haciendo una llamada al mismo y pasándole por parámetro el fichero con la tabla de particiones. Después de particionar se pasa a formatear la partición atendiendo al sistema de ficheros seleccionado por el administrador haciendo uso de la opción correspondiente del mkfs.

Después de haber realizado el particionado del disco duro de cada cliente el administrador procede a enviar la ubicación de la imagen a clonar y en que partición desea clonarla, esta orden llega al RESC-Client y este a su vez ejecuta el partimage pasándole estos datos mediante comandos y el mismo procede a realizar el restablecimiento del sistema operativo en la partición seleccionada.

2.4. Conclusiones.

Con la realización de este capítulo, se arribó a la conclusión, de que la obtención del diseño propuesto por el analista resultó muy importante, pues permitió adquirir una comprensión de todo lo relacionado con los requisitos funcionales y no funcionales del sistema, que sirvieron de base para el comienzo del desarrollo del mismo; permitió acotar los lenguajes de programación óptimos y adecuados para el desarrollo del sistema. Además se remplazaron algunas aplicaciones propuestas por el analista con el objetivo de realizar un trabajo más completo y óptimo. También se analizaron los componentes que el sistema reutiliza y la forma en que los integra.

3. CAPÍTULO III. DESARROLLO ÁGIL DEL SISTEMA DE CLONACIÓN Y DISTRIBUCIÓN DE IMÁGENES DE SISTEMA OPERATIVOS.

3.1. Introducción

En este capítulo se realizará el desarrollo ágil del **SistClon** utilizando la propuesta de una metodología ágil para el desarrollo de software que se realizó en el año 2008 por la ingeniera Malay Rodriguez Villar (MALAY, R. V. 2007). Se explicará toda la dinámica del proyecto en forma de historias de usuarios, prototipos de interfaz de usuario y algunos modelos auxiliares.

3.2. Planificación del proyecto por roles.

Rol	Miembro
Gerente (Management)	Dayron Pérez Roldán
Cliente (Customer)	Robmay García Fuentes. Dirección de los laboratorios.
Programadores (Programmers)	Dayron Pérez Roldán Jorge Mijail Vazquez Paredes José Ernesto Torres Sánchez
Analista (Analyst)	Dayron Pérez Roldán
Diseñadores (Designers):	Dayron Pérez Roldán Jorge Mijail Vazquez Paredes
Encargado de Pruebas (Tester)	Dayron Pérez Roldán
Arquitecto (Architect)	Irina Gonzalez Oduardo

3.3. Historias de usuarios y prototipos de interfaz de usuario.

A continuación se relacionan las historias de usuario, los prototipos de interfaz de usuarios y las tareas asociadas a ellas; se relacionan con la prioridad que tienen y los usuarios que se encargan de desarrollarlas. Esta es solo una planificación inicial, el proceso es cambiante para ir adecuándolo a las necesidades y nuevas propuestas. Todas las decisiones se toman de conjunto con el cliente que es parte del equipo de desarrollo.

Historia de Usuario	
Número: U-SCDSW-01	Nombre Historia de Usuario: Programar RESC-Client.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 2
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Debe estar incluido en el sistema de archivos que se envía al cliente y su función es conectar al cliente con el servidor y brindarle a la aplicación rumsg todas las órdenes que el servidor le envíe ya sean comandos o scripts.	
Observaciones: El RESC-Client se auxilia de la aplicación rumsg para ejecutar todas las órdenes del servidor creando una cola de mensajes entre ellos para poder comunicarse.	

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-01
Nombre Tarea: Crear Socket para establecer la conexión con el servidor.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 20/02/2008	Fecha Fin: 20/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Crea el socket para poder establecer la conexión por TCP/IP con el servidor y inicializa todas las variables del entorno.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-01
Nombre Tarea: Crear cola de mensajes.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 20/02/2008	Fecha Fin: 20/02/2008

Programador Responsable: Dayron Pérez Roldán
Descripción: Crea la cola de mensajes para poder comunicarse con la aplicación rumsg.

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-SCDSW-01
Nombre Tarea: Escribir Fichero	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.1
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 20/02/2008	Fecha Fin: 20/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Dado una dirección y el contenido que se desea escribir, se crea un fichero.	

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: U-SCDSW-01
Nombre Tarea: Tomar Control	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.1
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 21/02/2008	Fecha Fin: 21/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: En caso de que se pierda la conexión con el servidor se toma el control de la aplicación y se pone en espera de que el servidor se conecte nuevamente.	

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: U-SCDSW-01
Nombre Tarea: Reiniciar sistema.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.1
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 22/02/2008	Fecha Fin: 22/02/2008

Programador Responsable: Dayron Pérez Roldán
Descripción: Reinicia la ejecución del Cliente.

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: U-SCDSW-01
Nombre Tarea: Recibir Datos del Servidor	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 23/02/2008	Fecha Fin: 23/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se reciben todos los datos del servidor y se procesan de acuerdo al tipo de dato enviado, escribiendo los mismos finalmente en la cola de mensajes para su próxima ejecución por parte de la aplicación rumsg. En caso de error se toma el control del cliente.	

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: U-SCDSW-01
Nombre Tarea: Verificar conexiones.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.2
Fecha Inicio: 23/02/2008	Fecha Fin: 24/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Verificar que el servidor se encuentre activo y en caso de desconexión o error en la red se toma el control del cliente. Si el servidor se encuentra activo se envía una señal al servidor de que el cliente también está activo.	

Historia de Usuario	
Número: U-SCDSW-02	Nombre Historia de Usuario: Programar rumsg.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Se encarga de obtener de la cola de mensajes todas las órdenes que el servidor envía y que el RESC-Client le comunica para ejecutarlas una a una.	
Observaciones:	

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-02
Nombre Tarea: Procesar Datagramas.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.2
Fecha Inicio: 25/02/2008	Fecha Fin: 26/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se encarga de procesar todos los datagramas recibidos en la cola de mensajes, y ejecutarlos según su tipo, ya sea de tipo fichero o comando. El fichero se salvará en /tmp/nombre.ext para ser ejecutado luego. No se ejecutará el comando o el fichero hasta que no se reciban todos los datagramas que lo componen.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-02
Nombre Tarea: Obtener Datos.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.2

Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 25/02/2008	Fecha Fin: 26/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se encarga de leer de la cola de mensajes todos los datos para procesarlos correctamente.	

Historia de Usuario	
Número: U-SCDSW-03	Nombre Historia de Usuario: Programar RESC-Server.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 2
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 2
Descripción: Gestiona todas las conexiones de los clientes y es el encargado de enviar las órdenes de ejecución de comandos o scripts a cada cliente individualmente o a todos al mismo tiempo.	
Observaciones: El RESC-Server se ejecuta como un demonio y recibe las órdenes de la interfaz gráfica que luego se las comunica a los clientes. La comunicación entre el RESC-Server y la interfaz gráfica se realiza usando una cola de mensajes.	

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-03
Nombre Tarea: Leer Fichero.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.1
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 26/02/2008	Fecha Fin: 26/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Lee un fichero en una dirección especificada y lo carga en una lista para	

su posterior envío a los clientes.

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-03
Nombre Tarea: Crear Socket para recibir conexiones de los clientes.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 26/02/2008	Fecha Fin: 26/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Crea el Socket para poder recibir la conexión por TCP/IP de todos los clientes y inicializa todas las variables del entorno.	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-SCDSW-03
Nombre Tarea: Programar clase Familia de Clientes.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 27/02/2008	Fecha Fin: 27/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se encarga de Almacenar todas las conexiones que existen en el servidor. Posee una lista de clientes, con toda su información relevante.	

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: U-SCDSW-03
Nombre Tarea: Programar clase cliente.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 27/02/2008	Fecha Fin: 27/02/2008
Programador Responsable: Dayron Pérez Roldán	

Descripción: Se encarga de Encapsular la Información de cada Cliente.

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: U-SCDSW-03
Nombre Tarea: Aceptar Conexiones de cada Cliente.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.2
Fecha Inicio: 27/02/2008	Fecha Fin: 28/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Gestiona todas las conexiones entrantes y las añade a la lista de clientes conectados al servidor, además gestiona la información que se envía de cada cliente al servidor.	

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: U-SCDSW-03
Nombre Tarea: Verificar conexiones de cada cliente.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.2
Fecha Inicio: 28/02/2008	Fecha Fin: 29/02/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se deben verificar que cada cliente este conectado al sistema y sino no lo está, eliminarlo de la lista de conexiones del servidor.	

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: U-SCDSW-03
Nombre Tarea: Programar protocolo de comunicación.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.3

Fecha Inicio: 1/03/2008	Fecha Fin: 3/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Si los ficheros que se envía son muy grandes se deben tratar en bloques de 256 bytes y enviar cada uno de estos a cada cliente donde el mismo los reconstruye nuevamente y pasa a ejecutarlo. De igual forma si es muy pequeño solo se construye un bloque del mismo tamaño del fichero o comando y se envía para su posterior ejecución.	

Tarea de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: U-SCDSW-03
Nombre Tarea: Programar la clase entorno.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.4
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 4/03/2008	Fecha Fin: 8/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Gestiona todo el envío de comandos y scripts a cada cliente, así como la comunicación con la interfaz gráfica del SistClon por medio de la cola de mensajes.	

Historia de Usuario	
Número: U-SCDSW-04	Nombre Historia de Usuario: Diseñar y Programar base de datos de SistClon.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán Jorge Mijaíl Vazquez Paredes	Iteración Asignada: 1
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Se debe diseñar la base de datos de SistClon la cual se usará para almacenar toda la información del hardware de cada cliente. Además se deben programar todas los procedimientos almacenados que permitan interactuar con la base de datos.	
Observaciones: Se usará PostgreSQL como gestor de base de datos.	

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-04
Nombre Tarea: Diseñar Base de Datos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.2
Fecha Inicio: 9/03/2008	Fecha Fin: 10/03/2008
Programador Responsable: Dayron Pérez Roldán, Jorge Mijail Vazquez Paredes	
Descripción: Diseñar el modelo de datos que se usará para almacenar la información del hardware de cada cliente.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-04
Nombre Tarea: Programar procedimientos almacenados para insertar información en la base de datos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.2
Fecha Inicio: 9/03/2008	Fecha Fin: 10/03/2008
Programador Responsable: Dayron Pérez Roldán, Jorge Mijail Vazquez Paredes	
Descripción: Se debe programar todos los procedimientos almacenados para insertar información en la base de datos por cada tabla existente.	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-SCDSW-04
Nombre Tarea: Programar consultas para obtener información de la base de datos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.2
Fecha Inicio: 9/03/2008	Fecha Fin: 10/03/2008
Programador Responsable: Dayron Pérez Roldán, Jorge Mijail Vazquez Paredes	
Descripción: Se programarán todas las consultas que sean necesarias para obtener la información de la base de datos.	

Historia de Usuario	
Número: U-SCDSW-05	Nombre Historia de Usuario: Gestionar Listado de Direcciones IP
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 1
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Se realiza al iniciar el sistema y consiste en gestionar y crear un listado de las direcciones IP de los PC clientes que están conectados con el sistema, este listado se mantendrá en constante actualización de acuerdo al estado del cliente.	
Observaciones: El cliente debe tener al iniciar el RESC-Client ejecutándose y del lado del servidor debe existir el RESC-Server ejecutándose también.	



Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-05
Nombre Tarea: Verificar conexiones entrantes.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 9/03/2008	Fecha Fin: 9/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Verificar que no exista una doble conexión desde el mismo ip.	

Historia de Usuario	
Número: U-SCDSW-06	Nombre Historia de Usuario: Gestionar información del PC cliente.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 1
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Se realiza automáticamente una vez que se inicia el sistema y consiste en gestionar toda la información referente a cada PC cliente; tipo de disco duro, tamaño, dirección IP, cantidad de particiones existentes, etc. Información que será necesaria para realizar las operaciones definidas por el usuario.	
Observaciones:	

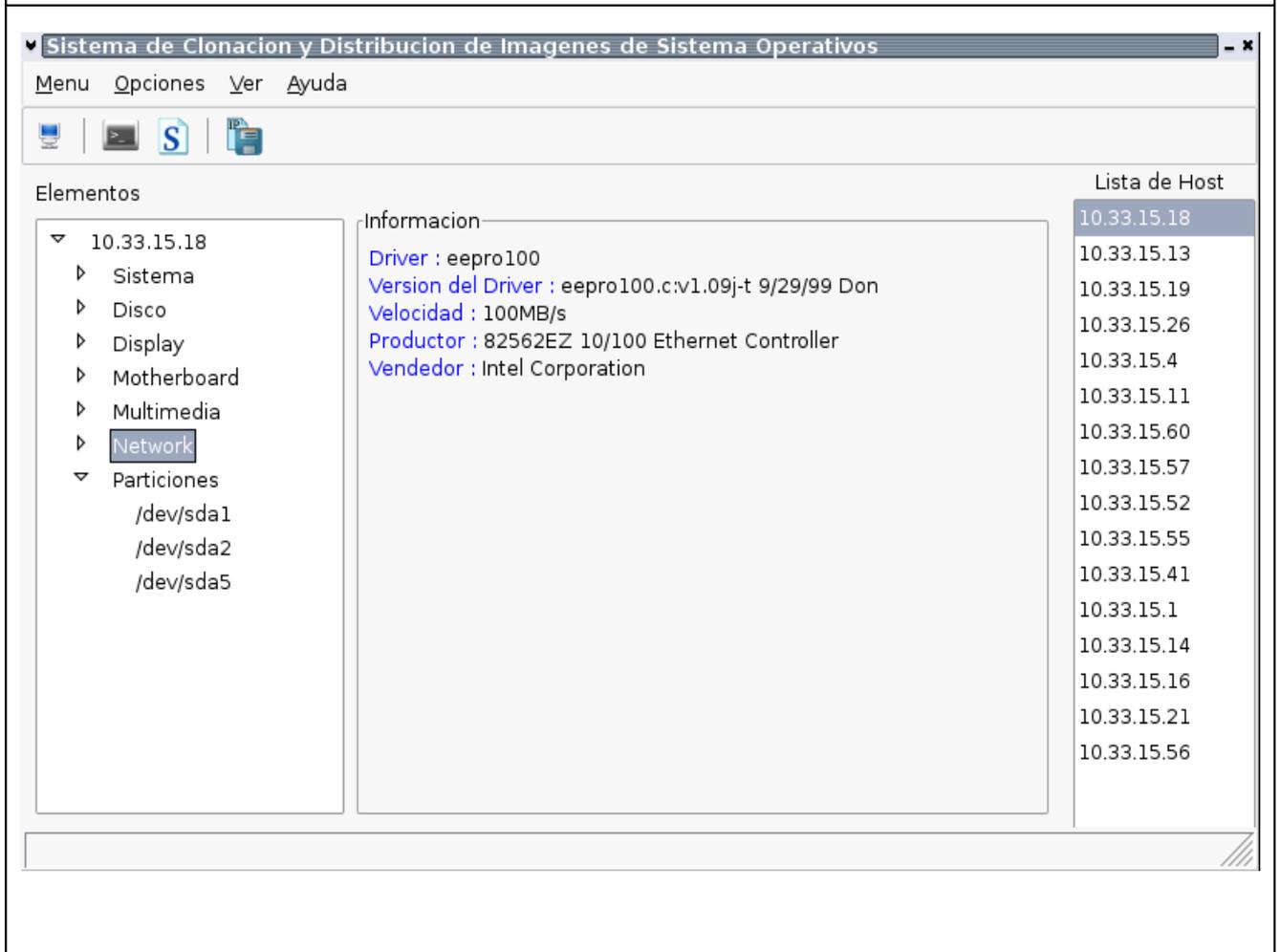
Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-06
Nombre Tarea: Generar XML con información del Hardware.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.3
Fecha Inicio: 9/03/2008	Fecha Fin: 11/03/2008

Programador Responsable: Dayron Pérez Roldán
Descripción: Usando el comando (lshw -xml) se genera un xml que contiene toda la información del hardware del cliente.

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-06
Nombre Tarea: Programar parser para leer XML.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.3
Fecha Inicio: 9/03/2008	Fecha Fin: 11/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se encarga de interpretar y obtener la información que se encuentra en el XML y almacenarla para su posterior uso.	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-SCDSW-06
Nombre Tarea: Enviar información del PC cliente a la Base de datos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.3
Fecha Inicio: 9/03/2008	Fecha Fin: 11/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Consiste en establecer la conexión con la base de datos y enviar toda la información que se gestionó del PC cliente. La base de datos usada es PostgreSQL y la aplicación es un scripts en perl.	

Historia de Usuario	
Número: U-SCDSW-07	Nombre Historia de Usuario: Mostrar información del cliente.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 2
Prioridad en Negocio: Baja (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Se realiza automáticamente una vez que el usuario selecciona un cliente en el listado de direcciones IP de los PC clientes conectados al sistema.	
Observaciones: Las conexiones deben estar abiertas para poder acceder a cada cliente.	

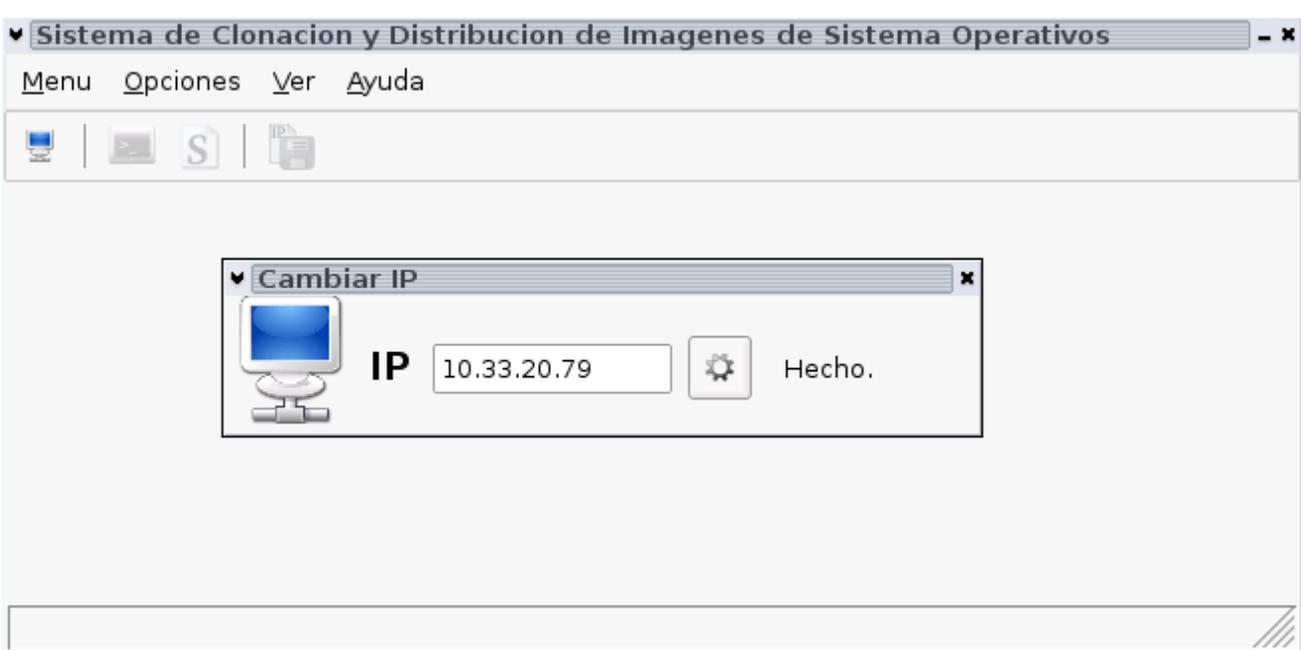


Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-07
Nombre Tarea: Verificar conexión con la base de datos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 12/03/2008	Fecha Fin: 12/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se verifica la conexión a la base de datos y se gestionan los errores que se puedan dar.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-07
Nombre Tarea: Obtener información de la base de datos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 12/03/2008	Fecha Fin: 12/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Dado el cliente seleccionado se busca en la base de datos y se obtiene, para mostrar toda la información del hardware de dicho cliente.	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-SCDSW-07
Nombre Tarea: Diseño de interfaz Mostrar Información del Cliente.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 13/03/2008	Fecha Fin: 13/03/2008
Programador Responsable: Dayron Pérez Roldán	

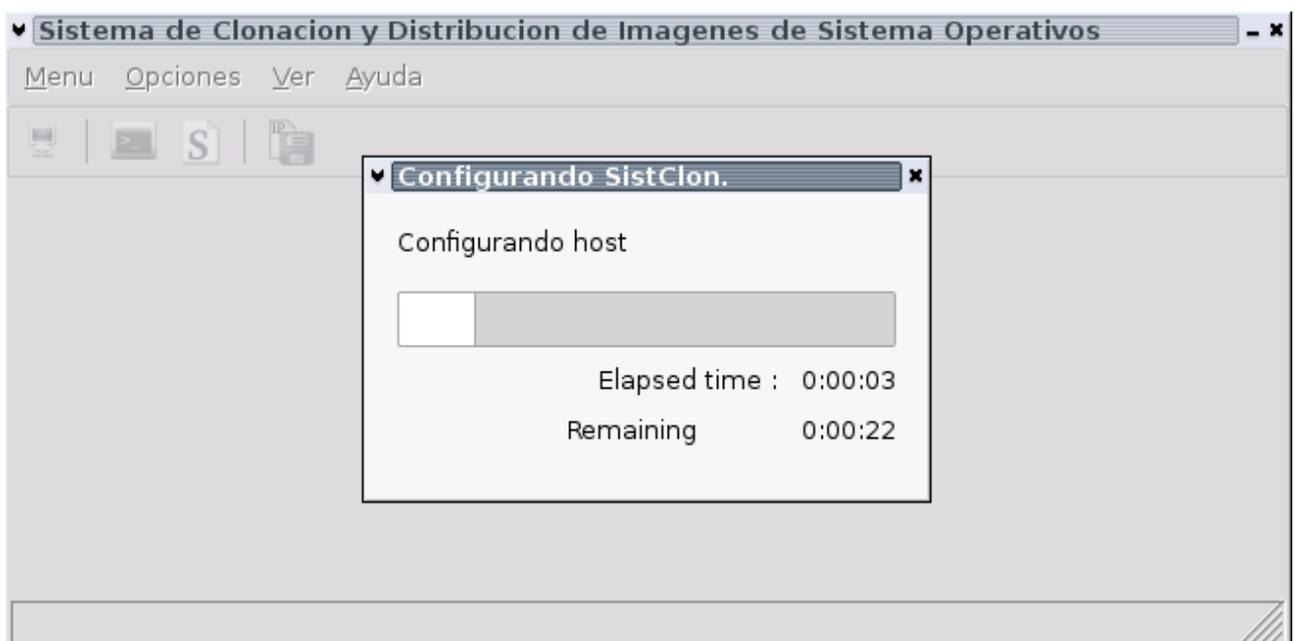
Descripción: Se diseñará una ventana para mostrar toda la información del cliente seleccionado. Esta información se mostrará de manera general o parcial, usando para esto el componente TreeView y un TextCtrl para mostrarla y poder resaltarla con colores.

Historia de Usuario	
Número: U-SCDSW-08	Nombre Historia de Usuario: Cambiar IP
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 2
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Permite al usuario de una manera fácil configurar la interfaz de red.	
Observaciones: Debe estar logueado como root para realizar esta operación.	
	

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-08
Nombre Tarea: Realizar verificaciones de errores.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 14/03/2008	Fecha Fin: 14/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se verifica que el usuario haya escrito la dirección correctamente el numero ip. También se verifica que tenga permisos de administración (root) para realizar esta operación.	

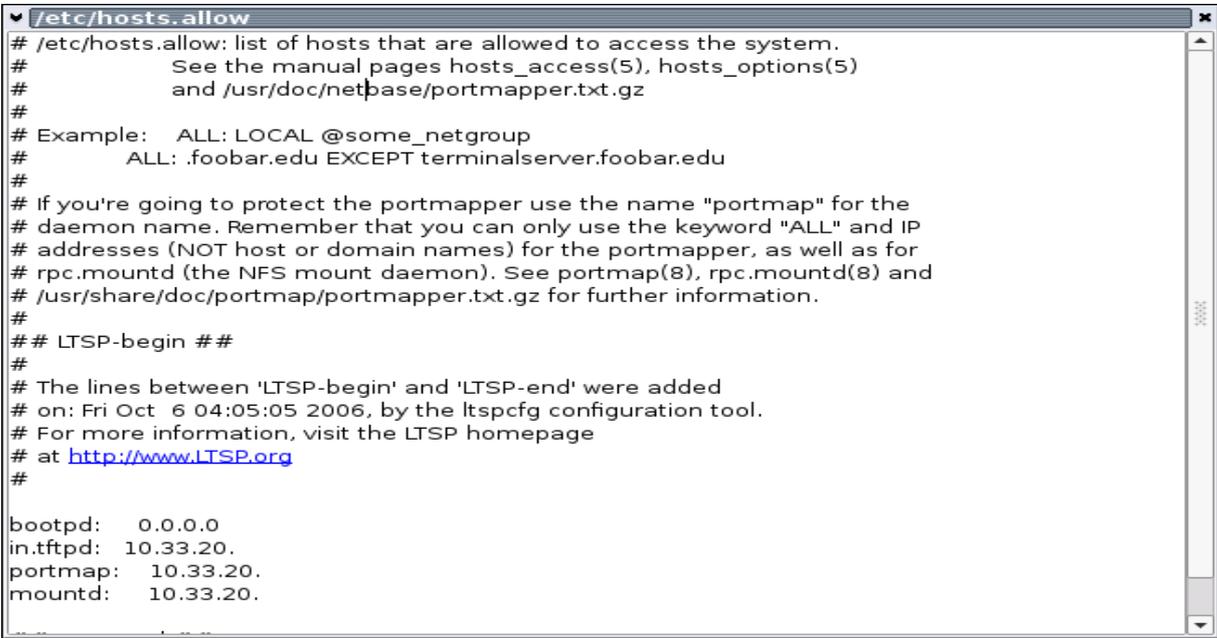
Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-08
Nombre Tarea: Diseño de interfaz cambiar IP.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 15/03/2008	Fecha Fin: 15/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se diseñará una ventana que permita entrar la nueva dirección ip y tendrán un botón para aplicar la nueva configuración de la interfaz de red.	

Historia de Usuario	
Número: U-SCDSW-09	Nombre Historia de Usuario: Configurar SistClon.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 2
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Se encarga de configurar todos los servicios, scripts y aplicaciones que componen al sistema basándose en la nueva interfaz de red proporcionada por el administrador del sistema.	
Observaciones: Debe estar logueado como root para realizar esta operación.	



Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-09
Nombre Tarea: Crear Script para configurar los servicios de SistClon.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 16/03/2008	Fecha Fin: 16/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se encarga de dado un número de ip configurar los principales servicios y scripts que componen la ejecución de SistClon.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-09
Nombre Tarea: Crear Interfaz gráfica para mostrar el progreso de la configuración.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 16/03/2008	Fecha Fin: 16/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Muestra el progreso de la configuración de los servicios y scripts que componen SistClon usando el componente wxProgressDialog.	

Historia de Usuario	
Número: U-SCDSW-10	Nombre Historia de Usuario: Visualizar Archivo.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 2
Prioridad en Negocio: Baja (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Permite buscar y mostrar el archivo seleccionado.	
Observaciones:	
 <pre> /etc/hosts.allow # /etc/hosts.allow: list of hosts that are allowed to access the system. # See the manual pages hosts_access(5), hosts_options(5) # and /usr/doc/netbase/portmapper.txt.gz # # Example: ALL: LOCAL @some_netgroup # ALL: .foobar.edu EXCEPT terminalserver.foobar.edu # # If you're going to protect the portmapper use the name "portmap" for the # daemon name. Remember that you can only use the keyword "ALL" and IP # addresses (NOT host or domain names) for the portmapper, as well as for # rpc.mountd (the NFS mount daemon). See portmap(8), rpc.mountd(8) and # /usr/share/doc/portmap/portmapper.txt.gz for further information. # ## LTSP-begin ## # # The lines between 'LTSP-begin' and 'LTSP-end' were added # on: Fri Oct 6 04:05:05 2006, by the ltspcfg configuration tool. # For more information, visit the LTSP homepage # at http://www.LTSP.org # bootpd: 0.0.0.0 in.tftpd: 10.33.20. portmap: 10.33.20. mountd: 10.33.20. </pre>	

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-10
Nombre Tarea: Diseñar y crear Interfaz gráfica para visualizar archivo.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 17/03/2008	Fecha Fin: 17/03/2008

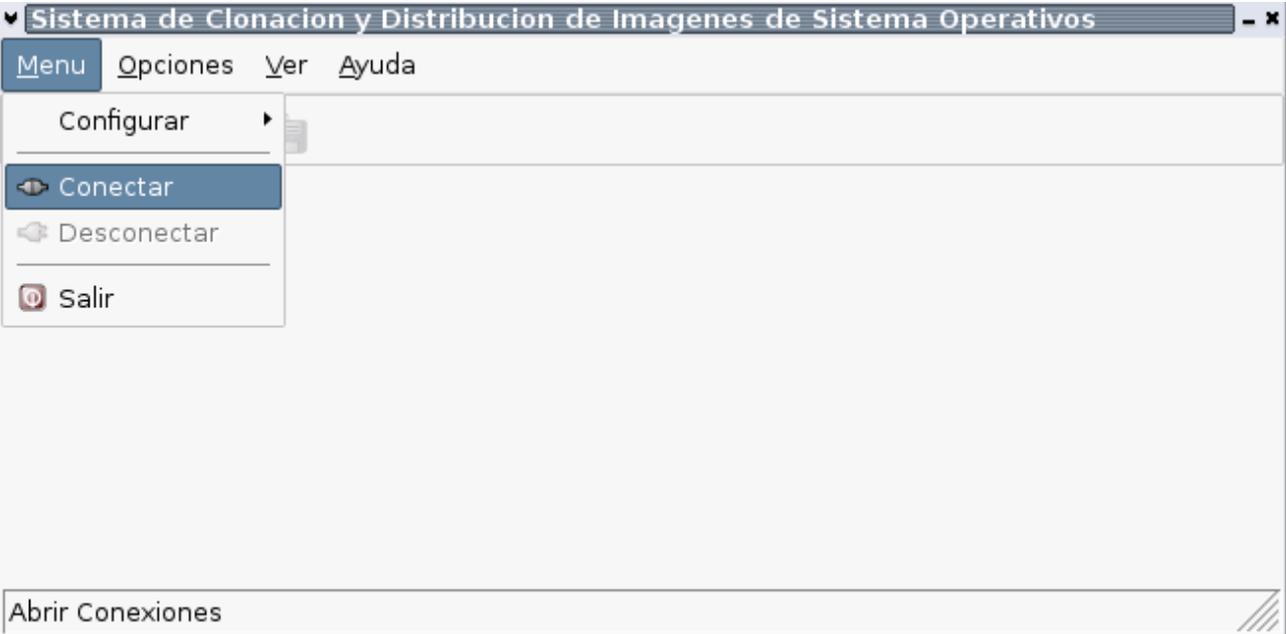
Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-11
Nombre Tarea: Crear interfaz gráfica para salvar listado de ip.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 17/03/2008	Fecha Fin: 17/03/2008
Programador Responsable: Dayron Pérez Roldán	
<p>Descripción: Se gestiona usando el componente wxFileDialog el lugar donde se desea escribir el fichero que contiene el listado de clientes conectados al sistema.</p>	

Historia de Usuario	
Número: U-SCDSW-12	Nombre Historia de Usuario: Configurar Acceso a la Base de Datos.
Modificación de Historia de Usuario Número:	
Usuario: Jose Ernesto Torres Sanchez	Iteración Asignada: 2
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
<p>Descripción: Permite configurar el acceso a la base de datos, facilitando el nombre de la base de datos a consultar, la dirección del host donde se encuentra, el usuario y la contraseña que le permitan acceder a la misma.</p>	
<p>Observaciones: El servidor debe tener la conexión a la base de datos por defecto.</p>	



Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-12
Nombre Tarea: Diseñar interfaz gráfica para configurar el acceso a la base de datos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.2
Fecha Inicio: 18/03/2008	Fecha Fin: 19/03/2008
Programador Responsable: José Ernesto Torres Sánchez	
Descripción: Se debe diseñar una interfaz gráfica que permita configurar el acceso a la base de datos, esta interfaz gráfica debe recibir el nombre de la base de datos, el host, el usuario y la contraseña para establecer la conexión. Además debe tener un botón que permita establecer la conexión.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-12
Nombre Tarea: Programar clase para acceder a la base de datos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 18/03/2008	Fecha Fin: 18/03/2008
Programador Responsable: José Ernesto Torres Sánchez	
Descripción: Esta clase se encarga de gestionar el trabajo con la base de datos, brinda la posibilidad de establecer una conexión con una base de datos dada, también brinda la posibilidad de realizar consultas y obtener los resultados. Además maneja todas las excepciones o errores que se puedan dar tanto en la sentencia SQL con a la hora de establecer la conexión con la base de datos especificada.	

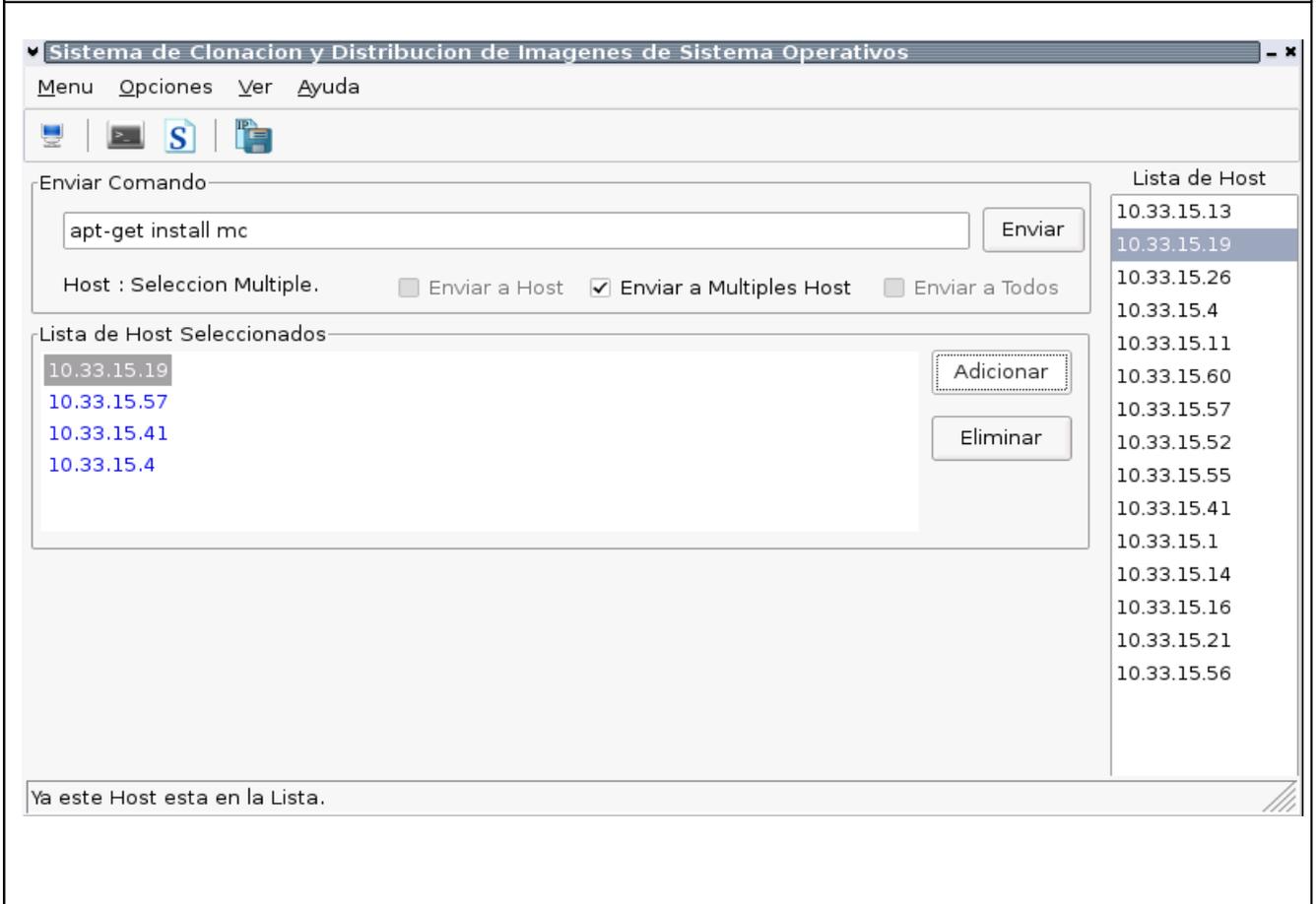
Historia de Usuario	
Número: U-SCDSW-13	Nombre Historia de Usuario: Abrir conexiones.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 2
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
<p>Descripción: Se crea un nuevo hilo de ejecución que debe permitir acceder a la cola de mensajes para obtener el listado de cliente que se encuentran conectados al servidor (RESC-Server) y posteriormente mostrar en la interfaz gráfica todos los clientes.</p> <p>Además brinda la posibilidad de mantener la comunicación entre la interfaz gráfica y el demonio RESC-Server.</p>	
<p>Observaciones: El RESC-Server escribe en la cola cada cliente que se conecta y cuando se abren las conexiones tan solo se leen los cliente conectados. Pero quien maneja todo el flujo de información es el RESC-Server.</p>	
	

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-13
Nombre Tarea: Crear hilo para gestionar clientes.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.3
Fecha Inicio: 20/03/2008	Fecha Fin: 22/03/2008
Programador Responsable: Dayron Pérez Roldán	
<p>Descripción: Este hilo es creado en forma de clase usando las posibilidades que brinda la programación con wxWidgets. Esta clase hilo tienen funciones que permiten acceder a la cola de mensajes que existe entre el RESC-Server y la interfaz gráfica para obtener la lista de clientes conectados al sistema, además posee funciones para procesar los datos enviados del servidor de acuerdo con su clasificación. También tiene métodos implementados que brindan la posibilidad de enviar ya sean comandos o scripts al RESC-Server y además recibir información del mismo. También posee un método que es el encargado de mantener actualizada la interfaz gráfica de acuerdo a los clientes que se van conectando o desconectando.</p>	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-13
Nombre Tarea: Diseñar Interfaz gráfica para visualizar los clientes conectados al sistema.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.3
Fecha Inicio: 20/03/2008	Fecha Fin: 23/03/2008
Programador Responsable: Dayron Pérez Roldán	
<p>Descripción: Se debe diseñar una interfaz gráfica que muestre los clientes que se encuentran actualmente conectados al sistema. Debe mostrarlos como una lista de clientes.</p>	

Historia de Usuario	
Número: U-SCDSW-14	Nombre Historia de Usuario: Cerrar conexiones.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 2
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: Debe poner en el estado de pause la ejecución del hilo abrir conexiones para no permitir la entrada de más conexiones y bloquear el envío de información del servidor hacia los clientes. También cierra la ventana con la lista de clientes.	
Observaciones:	

Historia de Usuario	
Número: U-SCDSW-15	Nombre Historia de Usuario: Enviar Comandos.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 2
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 1
<p>Descripción: Permite al usuario escribir el comando que desea enviar a los clientes y además permite seleccionar si desea enviarlo a todos los clientes, a un solo cliente o a múltiples clientes, permitiendo escoger a los que desea enviar el comando.</p> <p>Posteriormente escribe el comando en la cola de mensajes y el RESC-Server obtiene el comando y lo envía a los clientes atendiendo a la cantidad de clientes que se le deseaba enviar el comando.</p>	
<p>Observaciones: Tiene una interfaz amigable al usuario e intuitiva.</p>	



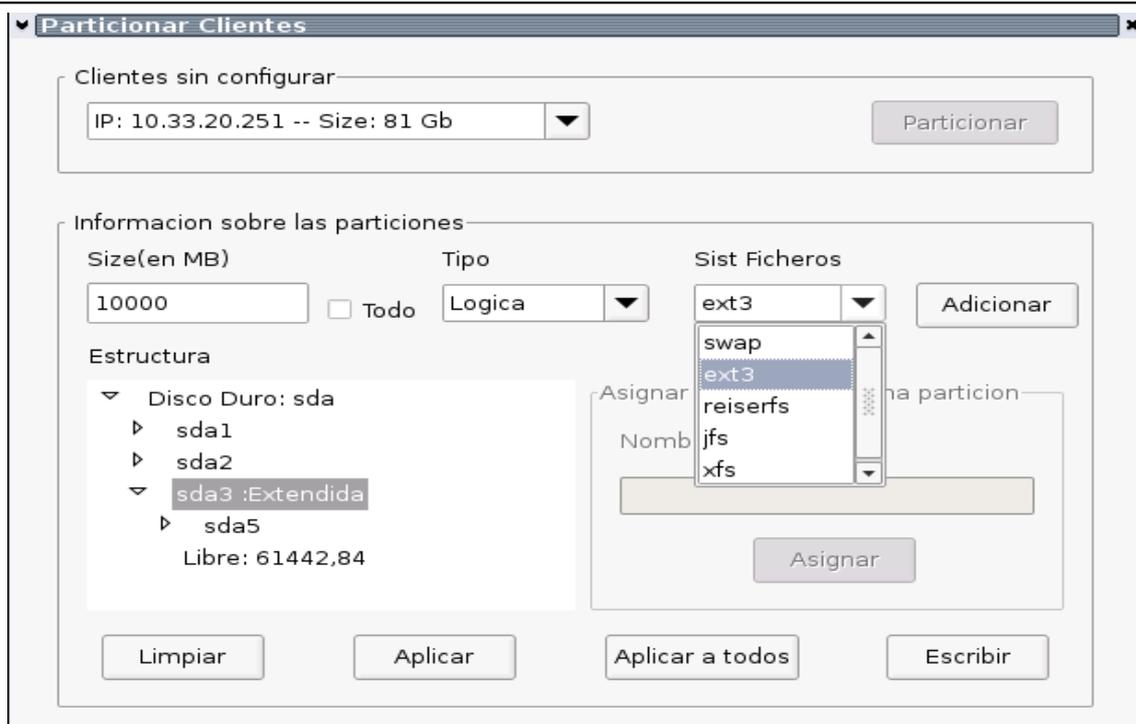
Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-15
Nombre Tarea: Diseñar interfaz gráfica para enviar comandos.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 24/03/2008	Fecha Fin: 24/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Se diseñará una interfaz gráfica que le permita al usuario escribir el comando que desea enviar a los clientes, además de permitirle escoger a cuantos clientes desea enviar el comando (a todos, a múltiples o a un solo cliente). Debe permitir seleccionar el/los clientes que se les pasará el comando.	

Historia de Usuario	
Número: U-SCDSW-16	Nombre Historia de Usuario: Enviar Scripts.
Modificación de Historia de Usuario Número:	
Usuario: Dayron Pérez Roldán	Iteración Asignada: 2
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 1
<p>Descripción: Permite al usuario buscar y seleccionar la dirección donde se encuentra el scripts que desea enviar a los clientes y además permite seleccionar si desea enviarlo a todos los clientes, a un solo cliente o a múltiples clientes, permitiendo escoger a los que desea enviar el script.</p> <p>Posteriormente escribe la dirección donde se encuentra el scripts a enviar en la cola de mensajes y el RESC-Server obtiene la dirección y carga el scripts bloque a bloque y lo envía a los clientes atendiendo a la cantidad de clientes que se le deseaba enviar el scripts.</p>	
Observaciones: Tiene una interfaz amigable al usuario e intuitiva.	



Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-16
Nombre Tarea: Diseñar interfaz gráfica para enviar scripts.	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 25/03/2008	Fecha Fin: 25/03/2008
Programador Responsable: Dayron Pérez Roldán	
Descripción: Diseñar una interfaz gráfica que le permita al usuario seleccionar el script que desea enviar a cada cliente y que le permita escoger a cuantos clientes desea enviar el script (a todos, a múltiples o a un solo cliente). Debe permitir seleccionar el/los clientes que se les enviará el script.	

Historia de Usuario	
Número: U-SCDSW-17	Nombre Historia de Usuario: Configurar particionamiento.
Modificación de Historia de Usuario Número:	
Usuario: Jorge Mijail Vazquez Paredes	Iteración Asignada: 3
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 3
Descripción: Configura la estructura de la tabla de particiones para un cliente seleccionado. Se aplica esta configuración a este único cliente o a todos aquellos clientes que posean el disco duro con las mismas características del cliente seleccionado.	
Observaciones: Tiene una interfaz amigable al usuario e intuitiva.	



Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-17
Nombre Tarea: Diseñar y crear la interfaz gráfica	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 01/05/2008	Fecha Fin: 01/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Crear y diseñar la interfaz gráfica para poder seleccionar el cliente y gestionar la configuración de su tabla de particiones.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-17
Nombre Tarea: Diseñar y crear las clases	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 02/05/2008	Fecha Fin: 02/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Diseñar y crear las clases necesarias para gestionar los datos de los clientes y la configuración de la tabla de particiones.	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-SCDSW-17
Nombre Tarea: Implementar las consultas a la base de datos	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 03/05/2008	Fecha Fin: 03/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Implementar las consultas para obtener la información necesaria del cliente seleccionado para configurar la tabla de particiones.	

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: U-SCDSW-17
Nombre Tarea: Implementar funciones	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 04/05/2008	Fecha Fin: 09/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Implementar las funciones necesarias para convertir la configuración de la tabla de particiones elaborada, al formato entendible por el "sfdisk". En el caso del tamaño de las particiones sería convertir de megas a sectores.	

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: U-SCDSW-17
Nombre Tarea: Configurar la tabla de particiones	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 10/05/2008	Fecha Fin: 15/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: El usuario configura la estructura de la tabla de particiones, esto se realiza adicionando particiones especificando el tamaño, tipo y sistema de ficheros de la misma. La información dada se gestiona haciendo uso de las funciones y clases implementadas.	

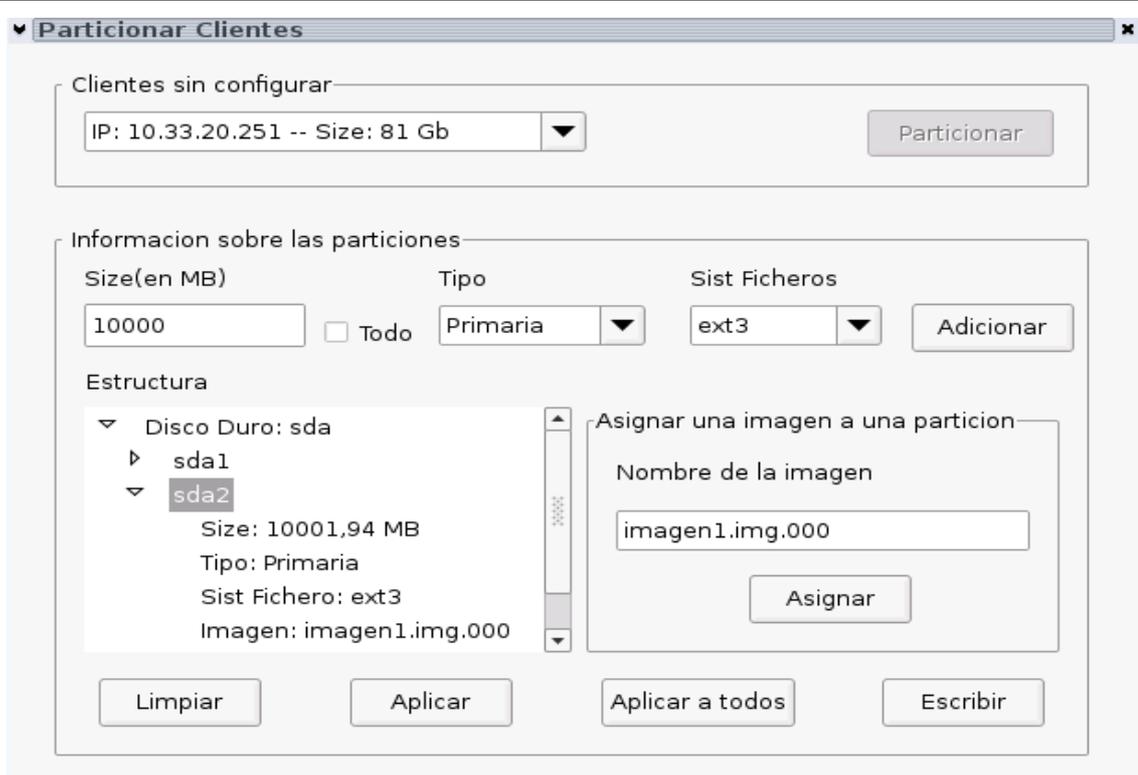
Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: U-SCDSW-17
Nombre Tarea: Aplicar la configuración	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.1
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 16/05/2008	Fecha Fin: 16/05/2008

Programador Responsable: Jorge Mijail Vazquez Paredes

Descripción: Se le aplica al cliente seleccionado la configuración de la tabla de particiones confeccionada, pasa a la lista de clientes configurados y se elimina del listado de clientes sin configurar.

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: U-SCDSW-17
Nombre Tarea: Aplicar a todos la configuración	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 17/05/2008	Fecha Fin: 17/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Se le aplica la configuración de la tabla de particiones confeccionada a todos los clientes que posean el disco duro con las mismas características del cliente seleccionado, pasan a la lista de clientes configurados y se eliminan del listado de clientes sin configurar.	

Historia de Usuario	
Número: U-SCDSW-18	Nombre Historia de Usuario: Asignar imagen.
Modificación de Historia de Usuario Número:	
Usuario: Jorge Mijail Vazquez Paredes	Iteración Asignada: 3
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: El usuario asigna la imagen a clonar en una o varias particiones de un PC cliente determinado.	
Observaciones:	



Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-18
Nombre Tarea: Diseñar la interfaz gráfica	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 15/05/2008	Fecha Fin: 15/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Diseñar la interfaz gráfica para asignar una imagen a una partición determinada.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-SCDSW-18
Nombre Tarea: Implementar funciones	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.3
Fecha Inicio: 16/05/2008	Fecha Fin: 19/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Implementar las funciones necesarias para conocer cual es la partición seleccionada a la que se le quiere asignar la imagen.	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-SCDSW-18
Nombre Tarea: Asignar imagen	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.1
Fecha Inicio: 20/05/2008	Fecha Fin: 20/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Se le asigna a la partición seleccionada la imagen especificada. Se gestiona que la partición no sea un área de intercambio.	

Historia de Usuario	
Número: U-SCDSW-19	Nombre Historia de Usuario: Escribir configuraciones.
Modificación de Historia de Usuario Número:	
Usuario: Jorge Mijail Vazquez Paredes	Iteración Asignada: 3
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 1
Descripción: El usuario escribe las configuraciones previamente hechas para aquellos clientes que se hayan configurado.	
Observaciones:	

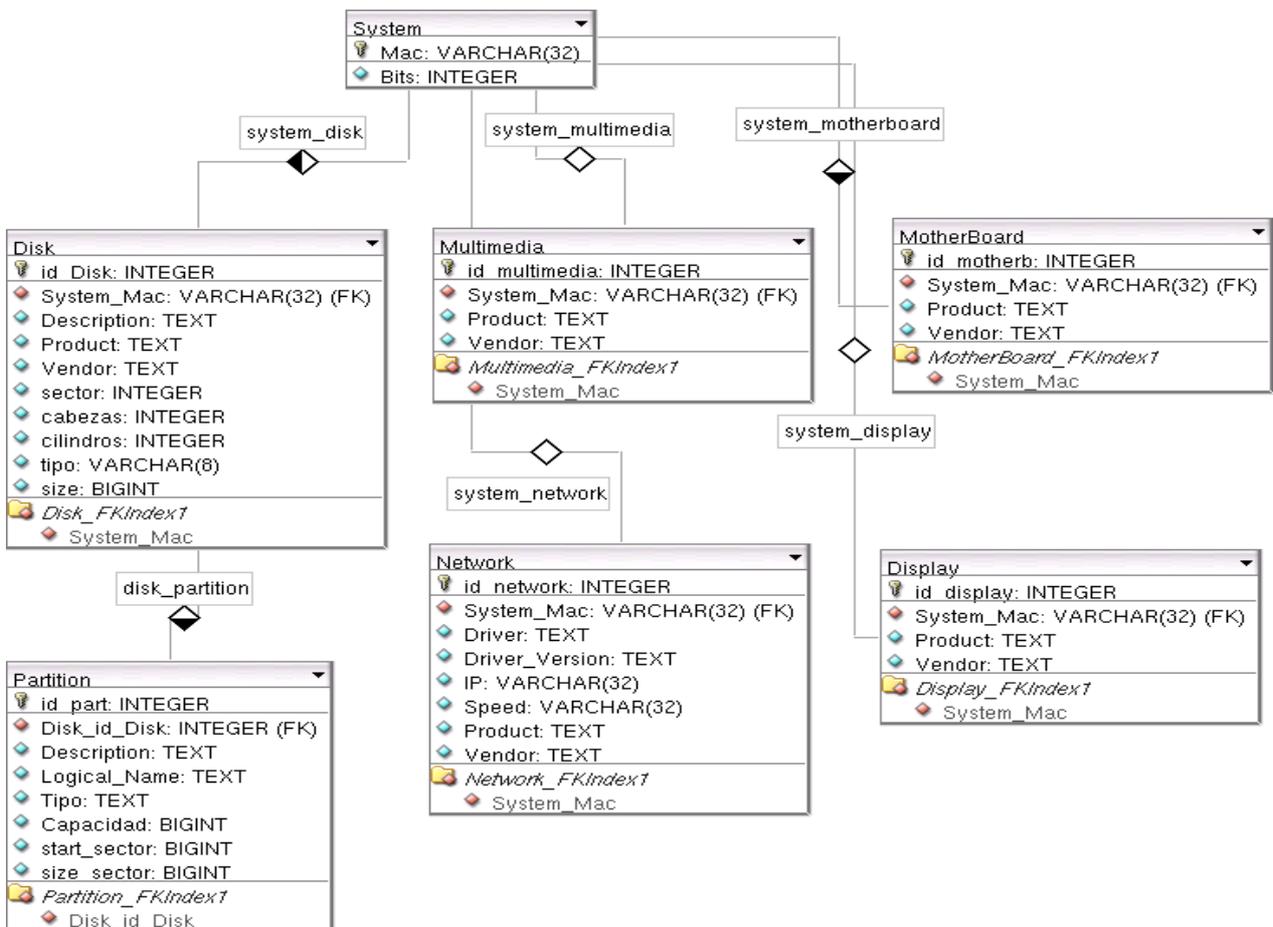
Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-SCDSW-19
Nombre Tarea: Implementar funciones	
Tipo de Tarea : Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados: 0.4
Fecha Inicio: 21/05/2008	Fecha Fin: 25/05/2008
Programador Responsable: Jorge Mijail Vazquez Paredes	
Descripción: Implementar las funciones necesarias para escribir la configuración de un cliente dado. Obtener del mismo, la configuración de la tabla de particiones, la imagen asignada a cada una de ellas y la partición que gestiona el gestor arranque.	

3.4. Base de datos.

A continuación se presenta el modelo de datos empleado para la aplicación final. Se utilizó el DBDesigner 4.0.5.4 para realizar el diseño de la base de datos. Se usó PostgreSQL como gestor de base de datos por sus potentes características. (ver sección 1.5.14)

Al iniciar el sistema en cada cliente se recoge información importante del hardware, esta información es de vital importancia para el administrador del sistema ya que la misma es empleada para poder realizar una correcta selección de la imagen del sistema operativo que se desea clonar en el ordenador cliente de acuerdo a las características de cada uno de los componentes y dispositivos del hardware.

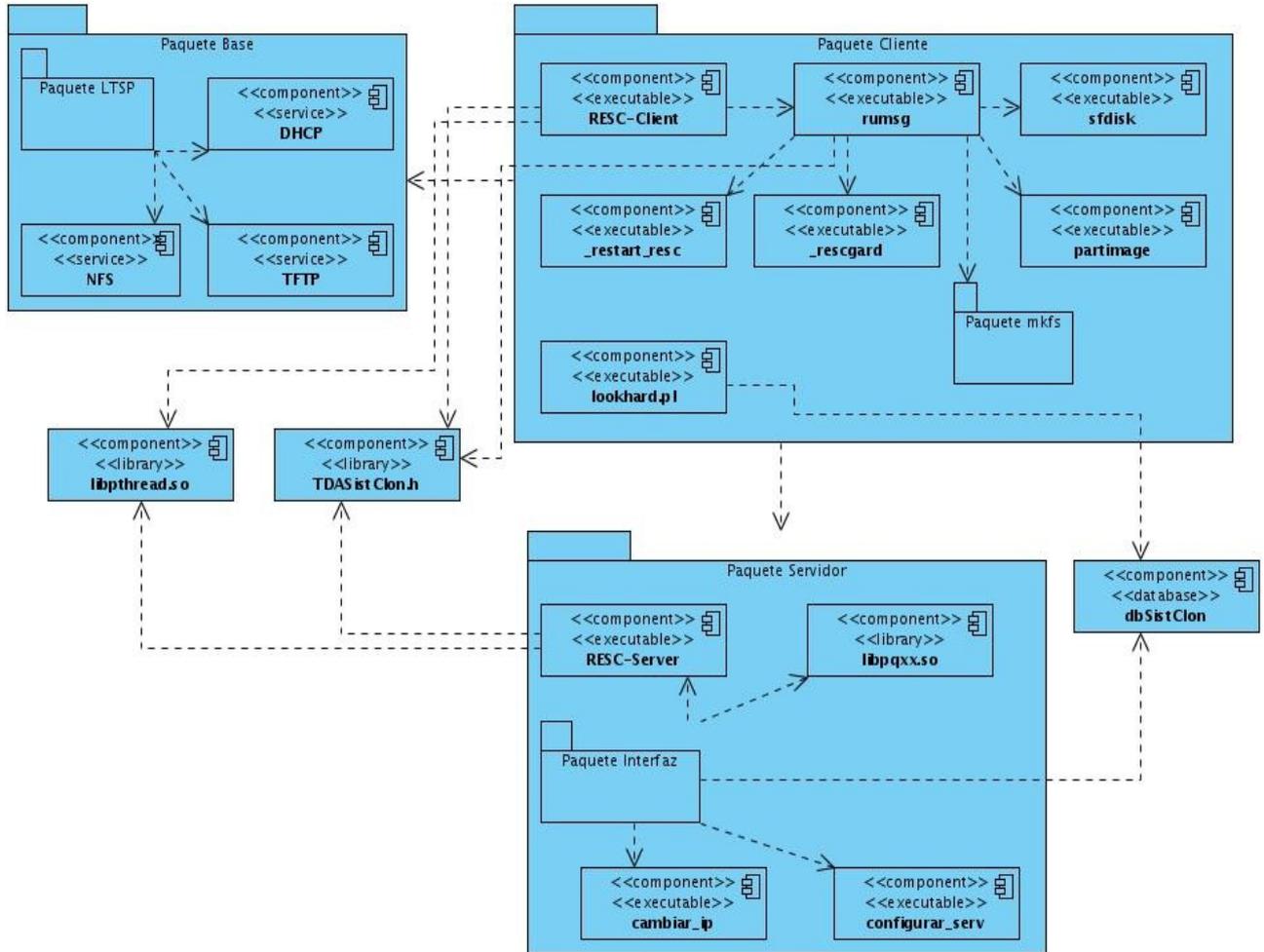
En la tabla "Partition" se almacena la estructura de la tabla de particiones del cliente en su estado actual antes de la clonación, esta tabla se actualizará si se realizan cambios en la estructura de la tabla de particiones.



3.5. Diagrama de componentes.

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

A continuación se presente el diagrama de componentes para el sistema que se propone.



3.6. Plan de releases.

En este paso se define el plan de releases e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha Historia. Como resultado de la priorización de Historias se llegó a la siguiente planificación:

Release	Historias de Usuarios	Tiempo estimado (Semanas)
1	3, 1, 2, 5, 4, 6	8
2	8, 9, 12, 13, 14, 7, 11, 10	8
3	15, 16,17,18,19	7

3.7. Conclusiones.

Se realizó la planificación del proyecto, se establecieron las historias de usuario y las principales tareas asociadas a las mismas para la primera planificación y se priorizaron. La planificación de iteraciones se hizo teniendo en cuenta la prioridad para el negocio de cada historia de usuario. Se realizó el diseño de la base de datos del sistema, así como el diagrama de componentes el cual permitió dar una mejor visión del proyecto.

4. CAPÍTULO IV. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

4.1. Introducción

Para lograr un producto con calidad es necesario trazarse un plan de pruebas desde el principio. Darle seguimiento a los cambios y desarrollar iterativamente. En este capítulo se plasman los casos de pruebas o test de aceptación a las que fue sometida la aplicación en cada una de las iteraciones, el cumplimiento de estos casos de pruebas fue el hito para avanzar hacia la próxima iteración. Se expone además una relación de las funcionalidades con las que cuenta el sistema hasta la fecha.

4.2. Casos de Prueba.

La programación extrema define entre iteración e iteración un conjunto de casos de pruebas o tests de aceptación para poder avanzar a una iteración superior. Durante el desarrollo del Sistema de Clonación y Distribución de Imágenes de Sistemas Operativos se diseñaron un conjunto de casos de prueba a las que fue sometido el sistema para comprobar el funcionamiento de acuerdo a las Historias de Usuario.

Se definieron casos de prueba para todas las historias de usuario, a continuación se relacionan las pruebas más significativas realizadas a aquellas historias de usuarios que se consideraban la espina dorsal del proyecto.

4.2.1. Caso de Prueba Historia de Usuario: U-SCDSW-01

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Programar RESC-Client*.

En esta historia se intenta probar que el RESC-Client es capaz de establecer una conexión con el servidor para aceptar órdenes del mismo en modo de comandos y scripts.

Además se verificará que sea capaz de detectar variaciones en la red que puedan causar perder la conexión con el servidor.

También se comprobará que las órdenes del servidor sean correctamente interpretadas y pasadas al rumsg para su posterior ejecución.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-01
Descripción de la Prueba: Al iniciar cada cliente se ejecuta el RESC-Client, donde el mismo comienza buscando al servidor para poder establecer la conexión. Si encuentra al servidor se establece la conexión y con ella se inicializan dos procesos: 1 – Verificar conexión con el servidor. 2 – Recibir Datos. Sino se establece la conexión, el cliente se mantiene en espera para poder conectarse al servidor.	
Condiciones de Ejecución: Se necesita el ip del servidor.	
Entrada / Pasos de ejecución: El cliente inicia y ejecuta el script encargado de poner a correr al RESC-Client suministrándole el IP del servidor.	
Resultado Esperado: Se establece la conexión con el servidor y se comienza a recibir datos del mismo.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 2	Número Historia de Usuario: U-SCDSW-01
<p>Descripción de la Prueba: Una vez que el RESC-Client haya establecido la conexión con el servidor, se procederá a comprobar que se detecte variaciones en la red realizando las siguientes pruebas :</p> <p>1- Se quitará el cable de Red en el servidor, para simular la caída de la red y poder verificar que los clientes detecten este cambio.</p> <p>2- Se terminará de manera inesperada la ejecución del servidor.</p>	
<p>Condiciones de Ejecución: Se necesita que existan clientes conectados.</p>	
<p>Entrada / Pasos de ejecución: Se detecta que se perdió la conexión e internamente se toma el control de la aplicación, quedando en espera para poder restablecerla.</p>	
<p>Resultado Esperado: Los clientes detectan que se perdió la conexión con el servidor y reaccionan tomando el control y notificando que se perdió la conexión; en este instante los clientes se quedan en espera de que se restablezca la conexión con el servidor para volver a conectarse.</p>	
<p>Evaluación de la Prueba: Prueba Satisfactoria.</p>	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 3	Número Historia de Usuario: U-SCDSW-01
<p>Descripción de la Prueba: Se comenzará a enviar comandos y scripts desde el servidor para poder verificar que se reciban de forma correcta y los mismos sean notificados y ejecutados por parte de la aplicación rumsg.</p>	
<p>Condiciones de Ejecución: Se necesita que existan clientes conectados. Se necesita además que la aplicación rumsg este corriendo.</p>	
<p>Entrada / Pasos de ejecución: Se introducirán los comandos y scripts que se desean enviar a los clientes. En esa ventana se mostrará un botón que permita enviar a cada cliente el texto introducido.</p>	
<p>Resultado Esperado: Se ejecuta en cada cliente cada comando y script enviado desde el servidor de manera correcta.</p>	
<p>Evaluación de la Prueba: Prueba Satisfactoria.</p>	

4.2.2. Caso de Prueba Historia de Usuario: U-SCDSW-02

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Programar rumsg*.

Para esta historia se comprueba que la aplicación rumsg recibe y ejecuta de manera correcta todos las órdenes que son enviadas al RESC-Client y que el mismo le comunica.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-02
Descripción de la Prueba: Se leerán los comandos y scripts de la cola de mensajes y se procesarán según su tipo para finalmente dar paso a su ejecución.	
Condiciones de Ejecución: Se necesita que la aplicación RESC-Client se encuentre corriendo.	
Entrada / Pasos de ejecución: Se lee la cola de mensajes para obtener los datos. Se procesan los datos para saber si son de tipo comando o de tipo script y entonces se pasa a ejecutar si es un comando y sino se salva el script y posteriormente se ejecuta.	
Resultado Esperado: Se obtienen los mensajes de la cola. Se procesan correctamente y finalmente se ejecutan.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.3. Caso de Prueba Historia de Usuario: U-SCDSW-03

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Programar RESC-Server*.

En esta historia se comprueba que la aplicación RESC-Server sea capaz de aceptar la conexión de aquellos ordenadores que tienen el RESC-Client corriendo.

Además se verificará que sea capaz de detectar variaciones en la red que puedan causar perder la conexión con los clientes de manera general o por cada cliente de forma individual.

También se comprobará que las órdenes del servidor sean correctamente enviadas a todos los clientes o a cada uno de forma individual.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-03
Descripción de la Prueba: Se inicia el RESC-Server y se espera a que se conecten todos los clientes. Para la prueba se están usando 5 laboratorios (150 pc) que poseen la aplicación cliente corriendo y se espera que el servidor soporte la conexión de 150 clientes. Además se intenta probar que la comunicación cliente/servidor se realice de manera correcta.	
Condiciones de Ejecución: Que exista conexión y que todos los clientes se encuentren ejecutándose.	
Entrada / Pasos de ejecución: Se inicia el servidor y el mismo se queda en espera de aceptar las conexiones entrantes para luego ir las almacenando en una lista de direcciones ip que representan los clientes conectados al servidor. Por cada cliente que se conecta el servidor comienza a responderle con mensajes de vida para hacerle saber al cliente que la conexión se mantiene.	
Resultado Esperado: Se espera que los clientes se mantengan conectados al servidor y que la comunicación entre ellos fluya de forma correcta.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 2	Número Historia de Usuario: U-SCDSW-03
<p>Descripción de la Prueba: Una vez que se haya iniciado el servidor y los clientes se encuentren conectados se le realizaran dos pruebas para verificar que el servidor es capaz de detectar cuando se pierde la conexión con los clientes.</p> <p>1 – Se quitará la red del servidor para simular una caída en la red local.</p> <p>2 – Se le quitará la red a algunos clientes, así como se terminará de manera inesperada su ejecución.</p>	
<p>Condiciones de Ejecución: Que el servidor y los clientes se encuentren conectados.</p>	
<p>Entrada / Pasos de ejecución: Se detecta que el/los cliente(s) no confirman su conexión al servidor y se espera 15 segundos para recibir información. Sino se recibe información alguna se toma al cliente o a los clientes como que se perdió la conexión con ellos y se eliminan de la lista de clientes conectados y se cierra el socket.</p>	
<p>Resultado Esperado: Que el servidor detecte cuando los clientes o el cliente pierden la conexión.</p>	
<p>Evaluación de la Prueba: Prueba Satisfactoria.</p>	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 3	Número Historia de Usuario: U-SCDSW-03
<p>Descripción de la Prueba: Se comenzará a enviar una serie de comandos y scripts a los clientes para verificar que el servidor sea capaz de enviarlos a cada cliente. Estas comandos y scripts son pasados por la interfaz gráfica de SistClon y obtenidos de la cola de mensajes.</p>	
<p>Condiciones de Ejecución: Que existan clientes conectados al servidor.</p>	
<p>Entrada / Pasos de ejecución: Se escribirá en caso de querer enviar un comando, el comando deseado y se presionará el botón enviar. Este comando es añadido a la cola de mensajes para ser obtenido luego por el RESC-Server.</p> <p>En caso de querer enviar un script, el mismo se seleccionará y posteriormente se cargará en memoria y será enviado a cada cliente repitiendo el proceso anterior.</p>	
<p>Resultado Esperado: Se espera que el servidor sea capaz de obtener de la cola de mensajes las órdenes y enviarlas a cada cliente y que las mismas se ejecuten correctamente.</p>	
<p>Evaluación de la Prueba: Prueba Satisfactoria.</p>	

4.2.4. Caso de Prueba Historia de Usuario: U-SCDSW-06

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Gestionar información del PC cliente*.

Para esta historia se intenta probar que la gestión de información del hardware de los ordenares clientes se realiza de manera satisfactoria y su envío a la base de datos es correcto.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-06
Descripción de la Prueba: Se iniciará un cliente con el SistClon y al iniciar se ejecutará un comando para generar la información del hardware en un XML y el cual será parseado con el script (lookhard.pl) para obtener la información más relevante del hardware y finalmente se enviará a la base de datos para ser verificada.	
Condiciones de Ejecución: Debe existir en el cliente el comando lshw el cual genera la información del hardware.	
Entrada / Pasos de ejecución: Se inicia el cliente y se genera el XML con el lshw para después ejecutar el script lookhard.pl para parsear el XML y obtener la información la cual finalmente es enviada a la base de datos.	
Resultado Esperado: Que se reconozca correctamente el hardware del ordenador cliente y que esta información sea enviada de manera correcta a la base de datos.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.5. Caso de Prueba Historia de Usuario: U-SCDSW-07

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Mostrar información del cliente*.

En esta historia se intenta probar que al seleccionar un cliente de la lista de clientes conectados la información que se encuentra en la base de datos del mismo, se muestre de manera correcta.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-07
Descripción de la Prueba: El administrador selecciona un ip de la lista de ip conectados, se debe mostrar la información del hardware del ordenador seleccionado, si dicho ordenador está entre los ip conectados pero no existe información del mismo en la base de datos se genera un error.	
Condiciones de Ejecución: Que existan clientes conectados. Debe existir la información en la base de datos.	
Entrada / Pasos de ejecución: Se selecciona el ip al cual se desea ver la información del hardware.	
Resultado Esperado: Se muestra de manera correcta la información del hardware.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.6. Caso de Prueba Historia de Usuario: U-SCDSW-09

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Configurar SistClon*.

Para esta historia se comprueba que se hayan configurado de manera correcta todos los servicios, scripts y aplicaciones que componen al SistClon.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-09
Descripción de la Prueba: Se escoge la opción <i>Configurar SistClon</i> del menú principal la cual se encarga de configurar todos los scripts, servicios y aplicaciones que componen al SistClon y luego se inicia algunos clientes para probar que la reciente configuración haya resultado correcta.	
Condiciones de Ejecución: Se necesita un ip para configurar todo.	
Entrada / Pasos de ejecución: Se cambia el ip del servidor y luego se ejecuta la opción configurar SistClon.	
Resultado Esperado: Se espera que se hayan configurado correctamente todos los servicios, scripts y aplicaciones lo cual permitiría que los clientes inicien de forma correcta.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.7. Caso de Prueba Historia de Usuario: U-SCDSW-13

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Abrir conexiones*.

Para esta historia se comprueba que se puedan enviar y recibir mensajes del RESC-Server y los mismos sean procesados de forma correcta, además de verificar que se visualice en la interfaz gráfica la conexión de todos los clientes conectados al RESC-Server.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-13
Descripción de la Prueba: Se inicia el RESC-Server como demonio para que se puedan conectar todos los clientes y luego se levanta la interfaz gráfica y se escoge la opción <i>Abrir Conexiones</i> del menú principal para verificar que todos los clientes se visualicen en la lista de clientes conectados. Luego se enviarán algunos comandos y scripts para probar que la conexión esta estable.	
Condiciones de Ejecución: Debe tener el RESC-Server corriendo como demonio.	
Entrada / Pasos de ejecución: Se escoge la opción <i>Abrir conexiones</i> y se genera un hilo de ejecución que se encarga de leer de la cola de mensajes la lista de ip conectados al servidor.	
Resultado Esperado: Que lo clientes se muestren en la interfaz gráfica. Además se espera que reciban las órdenes del servidor de manera correcta.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.8. Caso de Prueba Historia de Usuario: U-SCDSW-15

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Enviar Comandos*.

En esta historia se intenta probar que el envío de comandos haciendo uso de la interfaz gráfica se realiza correctamente, además de gestionar todos los posibles errores que puedan ocurrir en la interfaz gráfica.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-15
Descripción de la Prueba: Se selecciona la opción <i>Enviar Comandos</i> . Se intenta enviar un comando sin seleccionar ningún cliente. Se intenta también presionar el botón <i>Enviar</i> sin escribir ningún comando. Se comprueba que el comando se haya ejecutado en el/los cliente(s).	
Condiciones de Ejecución: Que existan clientes conectados. Además que se escriba un comando y que se haya escogido una de las opciones de envío.	
Entrada / Pasos de ejecución: Se escribe el comando, luego se selecciona una de las opciones de envío y finalmente se selecciona(n) el/los cliente(s) para enviar el comando. Después se pulsa el botón <i>Enviar</i> .	
Resultado Esperado: Que el comando se envíe de forma correcta en caso de no existir ningún error en la interfaz gráfica. Si existen errores en la interfaz gráfica se espera que se informen al usuario de manera agradable y precisa.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.9. Caso de Prueba Historia de Usuario: U-SCDSW-16

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Enviar Scripts*.

En esta historia se intenta probar que el envío de scripts haciendo uso de la interfaz gráfica es correcto, además de gestionar todos los posibles errores que puedan ocurrir en la interfaz gráfica.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-16
Descripción de la Prueba: Se selecciona la opción <i>Enviar Script</i> y se procede a buscar y seleccionar el script que se desea enviar. Se intenta enviar un script sin seleccionar ningún cliente. Se intenta también presionar el botón <i>Enviar</i> sin seleccionar ningún script. Se comprueba que el script se haya ejecutado en el/los cliente(s).	
Condiciones de Ejecución: Que existan clientes conectados. Además que se haya seleccionado un script y se escoja una de las opciones de envío.	
Entrada / Pasos de ejecución: Se selecciona el script, luego se selecciona una de las opciones de envío y finalmente se selecciona(n) el/los cliente(s) para enviar el script. Después se pulsa el botón <i>Enviar</i> .	
Resultado Esperado: Que el script se envíe de forma correcta en caso de no existir ningún error en la interfaz gráfica. Si existen errores en la interfaz gráfica se espera que se informen al usuario de manera agradable y precisa.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.10. Caso de Prueba Historia de Usuario: U-SCDSW-17

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Configurar particionamiento*.

Para esta historia se comprueba que se pueda configurar y mostrar la tabla de particiones para el cliente seleccionado de forma rápida y agradable al usuario final. Además se comprueba que esta configuración se aplique correctamente a todos aquellos clientes que posean iguales características de disco duro en caso de seleccionar el botón *Aplicar a todos*.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-17
Descripción de la Prueba: Se escoge un cliente de la lista de clientes sin configurar y se activa el área en la interfaz gráfica que permite configurar la estructura de la tabla de particiones. Se procede a configurar la tabla de particiones de la siguiente forma : una partición primaria que será la swap con un tamaño de 512 Mb, tres particiones lógicas de 17000, 12000 y 33000 Mb con un sistema de ficheros ext3 y finalmente se adiciona otra partición primaria de 10500 Mb con un sistema ficheros reiserfs.	
Condiciones de Ejecución: Que existan cliente conectados al sistema y que la información del hardware del cliente se encuentre en la base de datos.	
Entrada / Pasos de ejecución: Se selecciona el cliente a configurar y se presiona el botón <i>Particionar</i> . Se escribe el tamaño de la partición que se desea crear en Mb, luego se escoge el tipo (Primaria/Lógica) para después escoger el sistema de ficheros (swap, ext3, reiserfs, jfs, xfs) y se presiona finalmente el botón <i>Adicionar</i> , mostrándose la estructura que se le va dando a la tabla de particiones y el espacio libre que va quedando repitiéndose este proceso para cada partición que se desee adicionar.	
Resultado Esperado: Se espera que la configuración de la tabla de particiones para el cliente seleccionado se realice correctamente. También se espera que la estructura de la tabla de particiones se muestre en el orden correcto y le de una visión al usuario de como va quedando.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 2	Número Historia de Usuario: U-SCDSW-17
Descripción de la Prueba: Se presiona el botón <i>Aplicar a todos</i> una vez que se haya configurado el cliente seleccionado.	
Condiciones de Ejecución: Que se haya configurado la estructura de la tabla de particiones de un cliente antes seleccionado.	
Entrada / Pasos de ejecución: Presionar botón <i>Aplicar a todos</i> .	
Resultado Esperado: Se espera que la configuración creada para el cliente seleccionado se aplique a todos los clientes que tengan el disco duro con las mismas características y estos sean eliminados de la lista de clientes sin configurar.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.11. Caso de Prueba Historia de Usuario: U-SCDSW-18

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Asignar imagen*.

Para esta historia se comprueba que la asignación de la imagen que se desea clonar se asigne a una partición válida.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-18
Descripción de la Prueba: Se asigna una imagen a una partición de tipo swap y además se le asigna una imagen a una partición de tipo ext3.	
Condiciones de Ejecución: Que existan particiones adicionales.	
Entrada / Pasos de ejecución: Se selecciona una partición y se activa el área para asignar una imagen, se escribe el nombre de la imagen y se presiona el botón <i>Asignar</i> .	
Resultado Esperado: En caso de asignar una imagen a una partición no válida (swap) se espera que se muestre un mensaje al usuario informándole que no es posible asignar la imagen porque no es una partición válida. En caso de ser una partición válida se espera que se realice la asignación de forma correcta.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.2.12. Caso de Prueba Historia de Usuario: U-SCDSW-19

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: *Escribir configuraciones*.

Para esta historia se intenta probar que se escriba en los clientes las configuraciones previamente realizadas de forma correcta.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Número Historia de Usuario: U-SCDSW-19
Descripción de la Prueba: Se realiza a un grupo de clientes(30) las configuraciones necesarias que incluyen particionamiento y asignación de imágenes a clonar. Posteriormente se escribe dicha información en los clientes configurados.	
Condiciones de Ejecución: Que existan clientes conectados y configurados.	
Entrada / Pasos de ejecución: Se presiona el botón <i>Escribir</i> .	
Resultado Esperado: Se espera que las configuraciones realizadas a los clientes hayan sido escritas correctamente. Además en caso de que no existan configuraciones para escribir se muestre un mensaje de error.	
Evaluación de la Prueba: Prueba Satisfactoria.	

4.3. Resultados obtenidos.

En este apartado se relacionan los resultados obtenidos hasta el momento por el equipo de desarrollo de **SistClon**. Resaltar que como resultado de este trabajo **SistClon** está disponible en su versión 0.3, lo que también significa que se pueden esperar muchos más resultados en versiones superiores.

4.3.1. Acerca del tiempo de desarrollo.

El problema del mantenimiento de la imagen de la docencia en la facultad 10 era una situación problemática que se venía dando desde su concepción. No fue hasta septiembre del 2006 que se comienzan a dar los principales avances para encontrar la manera de darle solución al problema presentado. Se comenzó realizando labores investigativas.

A mediados del mes de noviembre del 2006 se realizan las planificaciones iniciales. Se descarga e instala el LTSP el cual sería la base del futuro proyecto que se gestaba. Aprender a configurar y a trabajar con el LTSP tomó aproximadamente 1 mes y medio. Después se comenzó a realizar el análisis y el diseño del sistema, tarea que demoró unos 3 meses.

De todas las tareas, la que tomó más tiempo cumplimentar fue la implementación del RESC el cual serviría para poder comunicar al servidor con todos los clientes (más de 2 meses y medio). En septiembre del 2007 se comienzan a implementar un grupo de tareas con el objetivo de poder obtener para mediados del año 2008 una versión completamente funcional del **SistClon**. El 30 de mayo del 2008 se realizó el lanzamiento de la versión 0.3 inestable.

Si se tiene en cuenta que el equipo de desarrollo no tenía experiencia previa de los elementos y conceptos más importantes para desarrollar un sistema de clonación de imágenes de sistema operativos, el período de tiempo empleado fue relativamente corto y se superaron las expectativas de la planificación inicial.

4.3.2. Acerca de las funcionalidades obtenidas.

Entre las principales funcionales que posee **SistClon** hasta su versión 0.3 se pueden mencionar :

- ◆ Es capaz de iniciar un grupo de ordenadores clientes por la red usando para ello la tecnología de clientes ligeros.
- ◆ Permite obtener toda la información del hardware de los clientes conectados al sistema.

- ◆ Controla cada cliente de forma remota, permitiendo esto la no interacción del administrador del sistema con cada cliente en particular para realizar tareas como :
 - Particionamiento de disco duro.
 - Formateado de particiones.
 - Restablecimiento de una imagen de un sistema operativo GNU/Linux.
 - Ejecución de comandos y scripts.
 - Configuración de ficheros del sistema, servicios, entre otras utilidades.
- ◆ Posee interfaz gráfica en el servidor que permite gestionar y administrar de forma remota todas las configuraciones de los clientes de manera sencilla y amigable.
- ◆ Tiempos de clonación superiores a UDPCAST y otros sistemas libres, podrían establecerse como parámetros de servicio las siguientes :
- ◆ Clonación de 50 Gb en 30 computadoras en un tiempo aproximado de 2 horas y 30 minutos. En la siguiente tabla se resumen algunos tiempos de clonación para diferentes cantidades de clientes :

Volumen de Información a transferir	Cantidad de clientes	Tiempo aproximado de transferencia (minutos)
10 Gb	30	55 - 65
15 Gb	30	70 - 80
30 Gb	30	95 - 105
3 Gb	1	3 - 5
5 Gb	1	6 - 8
10 Gb	1	10 - 15
15 Gb	1	25 - 30
30 Gb	1	45 - 55

Velocidad máxima de la red en el docente = 100 Mbits/s = 12 Mbytes/s

- ◆ Instalación, configuración y actualización de software en los clientes.

4.4. Conclusiones.

Se presentaron algunos casos de pruebas que guiaron la calidad del sistema, y determinaron en cada momento si se estaba o no en condiciones de continuar avanzando. En el análisis de los resultados obtenidos se muestran las funcionalidades alcanzadas por el sistema en el período que se ha estado trabajando en su desarrollo.

Conclusiones Generales

Se ha realizado un estudio de los sistemas de clonación y distribución de software más usados a nivel mundial, se ha analizado su funcionamiento y características fundamentales. Se implementó y probó un sistema cliente/servidor que permite clonar y distribuir imágenes de sistemas operativos GNU/Linux de manera remota lo cual satisface las necesidades existentes en la facultad 10 en cuanto a la instalación de imágenes de sistemas operativos en las computadoras de los laboratorios docentes. Por todo lo anteriormente expuesto se concluye que los objetivos trazados para el presente trabajo se han cumplido satisfactoriamente.

Recomendaciones

- ◆ Que el presente trabajo se siga desarrollando para implementar nuevas funcionalidades y mejorar su calidad y robustez.
- ◆ Que se realice un estudio de la complejidad de los algoritmos empleados para el desarrollo del producto con el objetivo de mejorar su rendimiento.
- ◆ Que el producto se comience a emplear para el mantenimiento de la imagen de los laboratorios docentes en la facultad 10.
- ◆ Que el producto sea probado a gran escala a fin de validar su funcionamiento.
- ◆ Que el trabajo sea inscrito como un producto propio de la UCI.

Referencias Bibliográficas

Acronis Team. Acronis True Image. Available from world wide web: <<http://es.acronis.com/>>.

Anjuta Team. Anjuta. Available from world wide web: <<http://anjuta.org/>>.

Apple. ASR. *Apple Software Restore*. Available from world wide web: <<http://www.madriver.k12.oh.us/technology/whitepapers/asr/>>.

Brutalix Team. Brutalix. Available from world wide web: <<http://osluz.unizar.es/node/109>>.

Bucknell University. DHCP. *Dynamic Host Configuration Protocol*. Available from world wide web: <<http://www.ietf.org/rfc/rfc2131.txt>>.

Carbon Copy Cloner Team. Carbon Copy Cloner. Available from world wide web: <<http://www.bombich.com/software/cccl.html/>>.

CloneHDD Team. CloneHDD. Available from world wide web: <<http://sourceforge.net/projects/clonehdd/>>.

CloneSys Team. CloneSys. Available from world wide web: <<http://sourceforge.net/projects/clonesys/>>.

Clonezilla Team. Clonezilla. Available from world wide web: <<http://www.clonezilla.org/>>.

Code::Blocks Team. Code::Blocks. Available from world wide web: <<http://www.codeblocks.org/>>.

DOD. Arpanet. *Advanced Research Projects Agency Network*. Available from world wide web: <<http://es.wikipedia.org/wiki/Arpanet>>.

Drive SnapShot Team. Drive SnapShot. Available from world wide web: <<http://www.drivesnapshot.de/en/index.htm>>.

DrvClonerXP Team. DrvClonerXP. Available from world wide web: <<http://drvclonerxp.pro.softonic.com/>>.

G4L Team. G4L. Available from world wide web: <<http://sourceforge.net/projects/g4l>>.

G4u Team. G4u. *Harddisk Image Cloning for PCs*. Available from world wide web: <<http://www.feyrer.de/g4u/>>.

GNU Project. Bash. Available from world wide web: <<http://www.gnu.org/software/bash/>>.

Image for Windows Team. Image for Windows. Available from world wide web: <<http://image-for-windows.softonic.com/>>.

Interbel. DriveClone. Available from world wide web: <<http://www.interbel.es/productos/driveClone/>>.

Ken Thompson, y Dennis M. Ritchie. C/C++. Available from world wide web: <<http://es.wikipedia.org/wiki/C%2B%2B>>.

- M23 Team. M23. Available from world wide web: <<http://m23.sourceforge.net/PostNuke-0.750/html/modules.php?name=News&file=article&sid=1000000007>>.
- MALAY, R. V. Introducción de procedimientos Ágiles en la producción de software en la Facultad 7 de la Universidad de las Ciencias Informáticas. 2007. [cited 15 Mayo 2008]. Available from world wide web: <http://bibliodoc.uci.cu/TD/TD_0693_07.pdf>.
- Marty Connor , y Michael Brown. Etherboot. Available from world wide web: <<http://sourceforge.net/projects/etherboot>>.
- Microsoft Windows. ADS. Available from world wide web: <<http://www.microsoft.com/spain/windowsserver2003/technologies/management/ads/adsbenefits.aspx>>.
- Miray. HDClone. Available from world wide web: <<http://www.miray.de/products/sat.hdclone.html>>.
- Morales A, Torres J, y Quesada A. LTSP. *Linux Terminal Server Project*. Available from world wide web: <<http://wiki.ltsp.org/twiki/bin/view/Ltsp/Documentation>>.
- Sanchez, M. A. M. Metodologías de Desarrollo de Software. 2002. [cited 20 Mayo 2008]. Available from world wide web: <http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf>.
- Scott W. Ambler. AM. *Agile Modeling*. Available from world wide web: <<http://www.agilemodeling.com/>>.
- Thomas Lange. FAI. *Fully Automatic Installation*. Available from world wide web: <<http://www.informatik.uni-koeln.de/fai/index.html>>.
- José H. Canós, Patricio Letelier, y Msc Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software. 2005. [cited 17 Mayo 2008]. Available from world wide web: <<http://www.willydev.net/descargas/prev/ToDoAgil.Pdf>>.
- Alejandro Valdés Villarrubia. Sistema de Clonación y Distribución de Imágenes de Sistema Operativos. 2007. [cited 27 Marzo 2008]. Available from world wide web: <http://bibliodoc.uci.cu/TD/TD_0874_07.pdf>.
- CONVAR. PC Inspector Clone maxx. Available from world wide web: <http://www.pcinspector.de/Sites/clone_maxx/info.htm?Language=5>.
- Francois Dupoux, y Franck Ladurelle. Partimage. Available from world wide web: <<http://www.partimage.org/>>.
- GNU Project. NCurses. Available from world wide web: <<http://www.gnu.org/software/ncurses/>>.
- Intel Corporation. *Preboot Execution Environment (PXE) Specification*. September 1999 [cited 18 Febrero 2008]. Available from world wide web: <<http://www.pix.net/software/pxeboot/archive/pxespec.pdf>>.
- Larry Wall. Perl. Available from world wide web: <<http://www.perl.org/>>.

Massachusetts Institute of Technology. TFTP. *Trivial File Transfer Protocol*. Available from world wide web: <<http://tools.ietf.org/html/rfc1350>>.

Microsoft Windows. RIS. *Remote Installation Services*. Available from world wide web: <http://en.wikipedia.org/wiki/Remote_Installation_Services>.

NetRestore Team. NetRestore. Available from world wide web: <<http://netrestore.softonic.com/mac>>.

ntfsclone Team. ntfsclone. Available from world wide web: <<http://man.linux-ntfs.org/ntfsclone.8.html>>.

Peter Anvin. PXE. *Preboot Execution Environment*. Available from world wide web: <<http://syslinux.zytor.com/pxe.php>>.

PING Team. PING. *Partimage Is Not Ghost*. Available from world wide web: <<http://ping.windowsdream.com/ping.html>>.

Rembo Technology, y IBM Company. *Rembo Toolkit 4.0 User Guide*. December 2005 [cited 26 Mayo 2008]. Available from world wide web: .

sfdisk Team. sfdisk. Available from world wide web: <<http://linux.die.net/man/8/sfdisk>>.

Sprite Clone Team. Sprite Clone. Available from world wide web: <<http://sprite-clone.softonic.com/palm>>.

SSH Communications Security. SSH. *Secure Shell*. Available from world wide web: <<http://www.openssh.org/>>.

Sun Microsystems. NFS. *Network File System Protocol Specification*. Available from world wide web: <<http://tools.ietf.org/html/rfc3530>>.

Symantec. Norton Ghost. Available from world wide web: <<http://www.symantec.com/es/mx/norton/products/overview.jsp?pcid=br&pvid=ghost14>>.

SystemImager Team. SystemImager. Available from world wide web: <<http://sourceforge.net/projects/systemimager/>>.

Udpcast Team. Udpcast. Available from world wide web: <<http://udpcast.linux.lu/>>.

wxWidgets Team. wxWidgets. Available from world wide web: <<http://www.wxwidgets.org/>>.

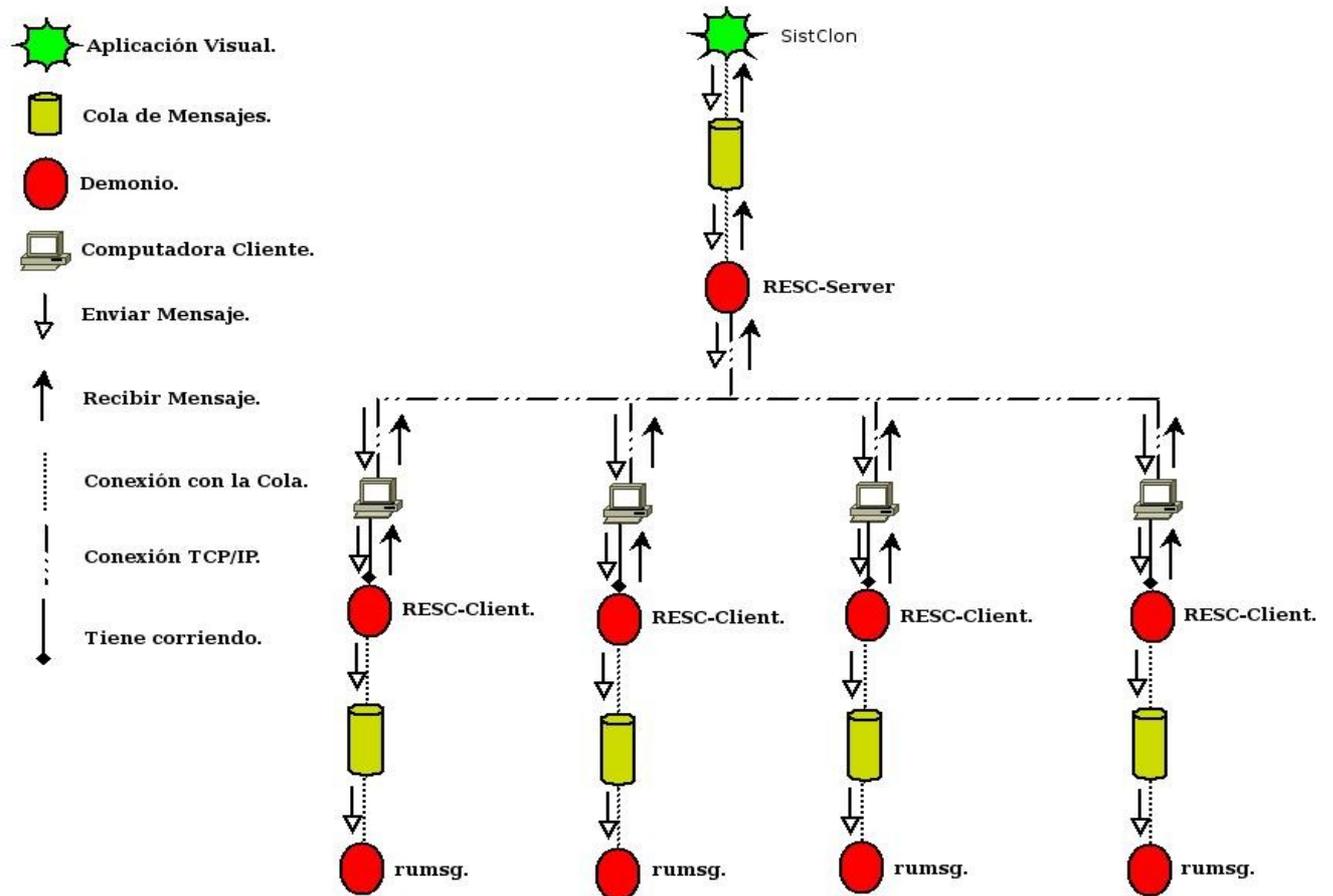
XXClone Team. XXClone. Available from world wide web: <<http://xxclone.softonic.com/>>.

Bibliografía

- Roselló, V. J. A. *Clustering de Alta Disponibilidad bajo GNU/Linux*. Septiembre 2001 Available from world wide web: <<http://beta.redes-linux.com/manuales/cluster/memoria.pdf>>.
- Rembo Technology, y IBM Company. *Rembo Toolkit 4.0 User Guide*. December 2005 [cited 26 Mayo 2008]. Available from world wide web: .
- MALAY, R. V. Introducción de procedimientos Ágiles en la producción de software en la Facultad 7 de la Universidad de las Ciencias Informáticas. 2007. [cited 15 Mayo 2008]. Available from world wide web: <http://bibliodoc.uci.cu/TD/TD_0693_07.pdf>.
- Morales A, Torres J, y Quesada A. LTSP. *Linux Terminal Server Project*. Available from world wide web: <<http://wiki.ltsp.org/twiki/bin/view/Ltsp/Documentation>>.
- Sanchez, M. A. M. Metodologías de Desarrollo de Software. 2002. [cited 20 Mayo 2008]. Available from world wide web: <http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf>.
- José H. Canós, Patricio Letelier, y Msc Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software. 2005. [cited 17 Mayo 2008]. Available from world wide web: <<http://www.willydev.net/descargas/prev/ToDoAgil.Pdf>>.
- Alejandro Valdés Villarrubia. Sistema de Clonación y Distribución de Imágenes de Sistema Operativos. 2007. [cited 27 Marzo 2008]. Available from world wide web: <http://bibliodoc.uci.cu/TD/TD_0874_07.pdf>.
- Intel Corporation. *Preboot Execution Environment (PXE) Specification*. September 1999 [cited 18 Febrero 2008]. Available from world wide web: <<http://www.pix.net/software/pxeboot/archive/pxespec.pdf>>.

Anexos

Anexo No. 1 Diseño operacional de la aplicación RESC.



Glosario de Términos

Cluster: Un cluster es un grupo de equipos independientes que ejecutan una serie de aplicaciones de forma conjunta y aparecen ante clientes y aplicaciones como un solo sistema.

Power_On_Self_Test: Un test automático después del encendido de la computadora que comprueba que todo esté conectado correctamente y que no haya ningún problema en los dispositivos.

BroadCast: Es un modo de transmisión de información por difusión donde la computadora emisora trasmite los paquetes simultáneamente para todos los receptores.

MAC: Es un identificador hexadecimal de 48 bits que se corresponde de forma única con una tarjeta o interfaz de red.

Kernel: Núcleo. Parte esencial de un sistema operativo que provee los servicios más básicos del sistema. Se encarga de gestionar los recursos como el acceso seguro al hardware de la computadora. Se encarga también del multiplexado, determinando qué programa accederá a un determinado hardware si dos o más quieren usarlo al mismo tiempo.

Gateway: Dispositivo dedicado a intercomunicar sistemas con protocolos incompatibles. Puerta de enlace, acceso, pasarela. Nodo en una red informática que sirve de punto de acceso a otra red.

PXE: Pre-Boot Execution Environment es una herramienta que permite a una computadora bootear de un servidor en una red antes de bootear el sistema operativo del propio disco.

RAM: Memoria de acceso aleatorio. Tipo de memoria donde la computadora guarda información para que pueda ser procesada más rápidamente. En la memoria RAM se almacena toda información que está siendo usada en el momento.

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

Particionar: Creación de divisiones en un disco duro para aplicarles un formato lógico (sistema de archivos).

Partición: División lógica en un disco duro.

API: Interfaz de Aplicación del Programa. Grupo de rutinas del sistema operativo o de una aplicación que definen cómo invocar cualquier servicio desde un programa.

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

Protocolo: Conjunto de normas que rigen un determinado proceso de comunicación.

Imagen: Grupo de configuraciones específicas que se extraen de una PC para distribuirlas en un grupo de PC.

Linux Terminal Server Project (LTSP): son un conjunto de aplicaciones servidores que proporcionan la capacidad de ejecutar Linux en computadoras de pocas prestaciones de

velocidad o computadoras de bajo costo.

Partimage: es un programa que crea una imagen de una partición, haciendo muy fácil recuperar dicha partición.

Dynamic Host Configuration Protocol (DHCP): Protocolo de Configuración de Hosts Dinámicos. Es un protocolo para asignar direcciones de IP dinámicas en una red.

Trivial File Transfer Protocol (TFTP): Protocolo de transferencia de archivos trivial.

Network: es una tarjeta de expansión que se inserta en una computadora para que pueda conectarse a una red. La mayoría de estas NIC se diseñan para un tipo particular de red, protocolo y medio, aunque algunas pueden servir para varias diferentes redes.

Network File System (NFS): Sistema de Archivos de Red. Desarrollado por Sun Microsystems, es un protocolo que permite establecer sistemas de archivos distribuidos entre múltiples máquinas.

Backup: copia de seguridad, copia de respaldo. Copia de archivos o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de los originales.

BIOS: Basic Input/Output System: Sistema básico de ingreso/salida de datos. Conjunto de procedimientos que controla el flujo de datos entre el sistema operativo y dispositivos tales como el disco rígido, la placa de vídeo, el teclado, el mouse y la impresora.

SSH : es el nombre de un protocolo y del programa que lo implementa. Este protocolo sirve para acceder a máquinas a través de una red.

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

Cliente: Sistema que establece un intercambio de datos con un servidor.

TCP/IP: Transfer Control Protocol / Internet Protocol. Son los protocolos que se utilizan en Internet para transmitir datos. El TCP está orientado a la conexión que establece una línea de diálogo entre el emisor y el receptor antes de que se transfieran los datos. El IP trata cada paquete de forma independiente e incluye en la cabecera información adicional para así controlar la información. Estos protocolos garantizan que la comunicación entre dos aplicaciones es precisa.

Etherboot: es código Open Source para la creación de boot ROM para el arranque por la red para plataformas x86, AMD64, Itanium, Hammer, Hyperstone y ARM (noMMU).

Máscara de red : es una combinación de bits que sirve para delimitar el ámbito de una red de computadoras. Su función es indicar a los dispositivos qué parte de la dirección IP es el número de la red, incluyendo la subred, y qué parte es la correspondiente al host.

Domain Name System (DNS): es un servicio de búsqueda de datos de uso general, distribuido y multiplicado. Su utilidad principal es la búsqueda de direcciones IP de sistemas

centrales (Hosts) basándose en los nombres de éstos. El estilo de los nombres de hosts utilizado actualmente en Internet es llamado nombre de dominio (Com, Net, Org, Es, etc.).

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

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

GPL: La GNU General Public License (inglés: Licencia Pública General) es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es Software Libre.

DRBL: Diskless Remote Boot in Linux (inglés : Disco de arranque remoto en Linux).

LVM (LVM2): es un administrador de volúmenes lógicos para el kernel de Linux. Entre sus funciones está la posibilidad de administrar espacios de usuarios, redimensionar los grupos y volúmenes lógicos, instantáneas de lectura y escritura y RAID0 de volúmenes lógicos, no soporta RAID1 o RAID5.

Multicast: Modo de difusión de información en vivo que permite que ésta pueda ser recibida por múltiples nodos de la red y por lo tanto por múltiples usuarios.

XP : Es una disciplina de desarrollo de software que persigue simplificar los procesos de desarrollo. Fue diseñada para ser usada con equipos de desarrollo pequeños que necesiten desarrollos ágiles y con requerimientos cambiantes.

RUP: Metodología de desarrollo de software basada en UML. Organiza el desarrollo de software en 4 fases.

MSF: El Marco de Soluciones Microsoft abarca temas tales como la estructura del equipo de trabajo que llevará adelante el proyecto (Modelo de Equipos) tanto con el personal de la Consultora como del cliente, y sus roles en el mismo. Cubre el análisis y la mitigación de los riesgos inherentes al proyecto (Modelo de Riesgos), y la forma en que se analiza y establece el alcance del propio proyecto (Modelo de Procesos).

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

Framework: es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a

desarrollar y unir los diferentes componentes de un proyecto.

P2P: Peer to Peer. Sistema de red basado en el protocolo TCP/IP utilizado para el intercambio de ficheros entre usuarios de la red. Un mecanismo popular de intercambio de información, en el cual un cliente identifica a otros clientes e intercambia información con ellos sin pasar por un servidor central.

Unicast: Comunicación establecida entre un solo emisor y un solo receptor en una red.

DSL: son las siglas en inglés de Digital Subscriber Line que significa Línea de abonado digital. Este término es utilizado para hacer referencia de forma global a todas las tecnologías que proveen una conexión digital sobre línea de abonado de la red telefónica local.

Wake On Lan: es un estándar de redes de computadoras Ethernet que permite encender remotamente computadoras apagadas.