



**Lira: Solución para la configuración de
Servidores de Clientes Ligeros sin Disco Duro
usando la distribución de GNU/Linux cubana
Nova**

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

Presentado en la Facultad 10
de la Universidad de las Ciencias Informáticas por

Viviana Alvarez Ferriol
Alberto José Aragón Alvarez

en junio del 2009

Tutor: Ing. Abel Meneses Abad
Co-Tutor: Ing. Mijail Hurtado Fedorovich

Versión del documento 0.9.7

Observaciones introductorias

Versión 0.2 publicada en junio del 2009

Copyright © 2009 Universidad de las Ciencias Informáticas (UCI)

Licencia. Esta obra se publica bajo la licencia Creative Commons 3.0 que establece las siguientes condiciones:



Atribución: Debes reconocer y citar la obra de la forma especificada por el autor y el licenciante.



No Comercial: No se puede utilizar esta obra para fines comerciales.



Licenciar igual: Si se altera o transforma esta obra, o se genera una obra derivada, sólo se puede distribuir la obra generada bajo una licencia idéntica a esta.



Para ver una copia de esta licencia visitar la siguiente dirección en Internet:
<http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>.

Marcas comerciales y marcas de servicios. Todas las marcas comerciales, marcas de servicios, logotipos y nombres de compañías mencionadas en esta obra son propiedad de sus respectivos dueños. Las mismas están protegidas bajo la ley de marcas comerciales y la ley de competencia desleal.

La importancia del glosario. Los autores recomiendan leer el glosario completo antes de comenzar con el primer capítulo de la presente obra.

Indicaciones para el uso de la obra. Este trabajo está optimizado tanto para su uso en una pantalla de una computadora como en papel. Se recomienda el uso de la versión digital, cada vez que sea aplicable. Es un archivo en Formato de Documento Portátil (PDF) con hipervínculos para facilitar la navegación. Los hipervínculos aparecen resaltados con fuente de color rojo en el documento.

Agradecimientos

Dedicatorias

Resumen

En estos momentos Cuba se encuentra invirtiendo en computadoras *clientes ligeros* que tienen como característica la carencia de *disco duro* (en lo adelante *diskless*) lo que da lugar a que en el mercado mundial se vendan más baratas, constituyendo esto un ahorro de divisas para el país. Estas máquinas clientes se enlazan a otras llamadas *Servidores* que si poseen *disco duro* a través de la red para hacerles solicitudes (Figura 1.2). En la actualidad en Cuba no existe una aplicación que facilite el trabajo de los administradores de servidores de *diskless* en empresas y centros donde ya se están utilizando. Este trabajo es bastante complejo y nada funcional, puesto que actualmente hay que configurar los ficheros de los servicios necesarios como: DHCP, NFS, PXE y TFTP entre otros a puras líneas de comandos cuando con una solución como la propuesta en este trabajo se ahorraría tiempo y esfuerzo. Por esta razón los autores del presente trabajo se trazaron como objetivo principal desarrollar una solución con software libre para sistemas operativos GNU/Linux, que permitiera el despliegue de la solución de servidores de *diskless* en Cuba. Dicha solución es implementada y probada con éxito en este trabajo para la distribución cubana de GNU/Linux de nombre Nova.

En aras de lograr dicho objetivo los autores realizaron una búsqueda (sistematizada en el presente trabajo) consultando diferentes bibliografías con el fin de encontrar una aplicación libre que pudiera ser modificada o a la que se le pudieran añadir funcionalidades, para ajustarla a las necesidades específicas de esta investigación.

Luego de haber realizado esta búsqueda los autores determinaron que no existe ninguna aplicación que cumpla con los requisitos de este trabajo por lo que se propusieron desarrollar desde cero un software fácil de usar y libre que permitiera configurar los servidores de *diskless*. Fue desarrollado en lenguaje Python, con librerías gráficas GTK.

En el trabajo se relacionan cada uno de los pasos de la metodología de desarrollo ágil SXP, así como sus artefactos, que permitieron lograr una calidad aceptable en el producto final en dos meses. También se utilizó un estándar de código a la hora de programar la solución que permitiera, en un futuro, a otros desarrolladores implementar nuevas funcionalidades o refinar las ya existentes. El documento consta de un glosario bastante amplio que

posibilitará la comprensión completa de la obra así como de las referencias bibliográficas de los artículos, libros y sitios web consultados.

Como resultado final los autores lograron liberar la versión 0.2 del software Lira que permitirá configurar adecuadamente los servidores de diskless para distribuciones de Nova (por el momento), además se provee de una imagen¹ optimizada para el uso en diskless y un kernel² parcheado con AUFS³. Los procedimientos para lograrlo desde cero son anexados al presente trabajo.

¹ Hace referencia a una imagen ISO de un sistema operativo, que es un archivo donde se almacena la imagen exacta de un sistema de ficheros.

² Es el núcleo o parte fundamental de un sistema operativo que gestiona la interacción de los diferentes programas con el hardware de la computadora.

³ Hace referencia a la acción de poner un parche en este caso al kernel con AUFS que es un servicio de sistema de archivos que permite que se monten diferentes sistemas de archivos al mismo tiempo simulando ser un solo para sistemas de archivos en GNU/Linux.

Índice

| | |
|--|----|
| Introducción..... | 1 |
| 1 Fundamentación Teórica..... | 6 |
| 1.1 Diskless..... | 6 |
| 1.2 Ventajas e Inconvenientes..... | 7 |
| Ventajas..... | 7 |
| Inconvenientes..... | 7 |
| 1.3 Comparación de hardware..... | 8 |
| 1.4 Fundamento de los Diskless..... | 8 |
| Procesos para la configuración de un diskless..... | 13 |
| 1.5 Aplicaciones para servidores de diskless..... | 20 |
| Aplicaciones Libres..... | 21 |
| Aplicaciones No Libres..... | 27 |
| 1.6 Características que identifican a la herramienta Lira..... | 29 |
| 1.7 Herramientas, lenguajes y metodologías a utilizar..... | 29 |
| Herramientas..... | 30 |
| Lenguajes..... | 33 |
| Bibliotecas..... | 34 |
| Metodología de Desarrollo..... | 37 |
| 1.8 Conclusiones..... | 40 |
| 2 Planificación y Definición de la aplicación..... | 41 |
| 2.1 Información que se maneja..... | 41 |
| 2.2 Propuesta del sistema..... | 42 |
| Proceso de Configuración de un Servidor de Diskless..... | 42 |
| 2.3 Arquitectura de la Herramienta Lira..... | 46 |
| 2.4 Fase Planificación-Definición del Sistema de Lira..... | 46 |
| Plantilla de Concepción del Sistema..... | 47 |
| Modelo Historias de Usuarios del Negocio..... | 47 |
| Captura de requisitos..... | 49 |
| Definir las Historias de Usuarios..... | 51 |
| Plantilla de Historias de Usuarios..... | 51 |
| Planificación de las Historias de Usuarios..... | 57 |
| 2.5 Diseño con las Metáforas..... | 58 |
| Metáfora del Sistema..... | 58 |
| Plantilla de Modelo de Diseño..... | 58 |
| 2.6 Conclusiones..... | 58 |
| 3 Implementación y Pruebas..... | 59 |
| 3.1 Fase “Desarrollo” de Lira..... | 59 |
| 3.2 Tareas para lograr la implementación de Lira..... | 59 |
| Plantilla de Tareas de Ingeniería..... | 59 |
| Plantilla de Plan de Releases..... | 63 |
| 3.3 Implementación de Lira..... | 64 |
| Estándares de Codificación..... | 64 |
| 3.4 Fase “Pruebas” de Lira..... | 65 |
| Casos de prueba de Aceptación para Lira..... | 65 |
| 3.5 Resultados Obtenidos:..... | 67 |
| 3.6 Acerca de las funcionalidades obtenidas..... | 68 |
| Conclusiones Generales..... | 69 |
| Recomendaciones..... | 71 |
| Referencias Bibliográficas..... | 72 |
| Anexos..... | 75 |
| Glosario de Términos..... | 86 |

Introducción

Antecedentes

Cuba en los últimos años ha apostado por el desarrollo de la informática como ciencia productora de bienes y servicios para nuestra sociedad. Esta política favorece el desarrollo de múltiples esferas económico-sociales como la salud y la educación, donde el apoyo de las Tecnologías de la Informática y las Comunicaciones (en lo adelante TIC) juegan un papel muy importante, logrando de esta manera un paso de avance en el proceso de informatización de la sociedad cubana.

Como parte de este proceso surge la Universidad de las Ciencias Informáticas (en lo adelante UCI) como fuente de profesionales de la informática que serán los encargados de llevar a cabo el mismo.

“La Universidad de las Ciencias Informáticas nació para socializar y multiplicar como nunca antes, los estudios superiores de la informática en Cuba.

En cinco años, el país pasó de 2 483 a 16 395 estudiantes en carreras universitarias de computación, multiplicando la matrícula de la especialidad casi 7 veces que, unido a los 39 000 estudiantes de politécnicos de informática, significa una fuerza de más de 50 000 jóvenes comprometidos con la idea fundamental de convertir la informática en la rama más productiva y aportadora de recursos para la nación, empleando a fondo la inteligencia.” **[ITOO]**

Un desarrollo informático depende de múltiples recursos, dígame tecnología informática y personal capacitado para manipularlas, por lo que la nación cubana ha importado, durante más de una década, equipos de varias denominaciones y características.

Pero un grave problema o traba para lograr que las ciencias informáticas sean una de las fuentes de ingreso más importantes de la economía cubana es que el software imprescindible, tanto para desarrollar nuevas aplicaciones como para el trabajo de los usuarios en las oficinas, es propiedad de grandes transnacionales de la industria del software mundial como Microsoft, Sun Microsystems y otras.

La mayoría de los sistemas operativos usados en Cuba son de la familia Microsoft, que además constituyen copias ilegales de software, pues por el bloqueo del gobierno de los Estados Unidos esta compañía no puede vender ni contratar servicios a un país embargado.

Esta situación impide que en Cuba se creen empresas de desarrollo de software usando tecnologías privativas. Quedando como única alternativa desarrollar bajo las pautas del Software Libre y el Código Abierto.

Por esta razón el gobierno cubano se ha trazado como estrategia lograr la independencia tecnológica de la nación cubana, jugando un papel fundamental el proceso de migración hacia el software libre. Para lograr el éxito de este proceso es esencial la creación, basado en una distribución GNU/Linux, de un nuevo sistema operativo hecho en Cuba basado en códigos fuentes y distribuido bajo licencias libres que constituya la plataforma que soporte el proceso de transición.

Según Ángel Goñi Oramas, líder del proyecto Nova: “En estos momentos el Proyecto Nova tiene como objetivo la creación de un sistema operativo, no la mera personalización de una distribución. Se aspira a proveer un producto orientado a usuarios inexpertos que hayan tenido que migrar de Microsoft Windows a entornos *GNU/Linux* o cuya experiencia con computadoras sea nula. Se pretende automatizar la mayor cantidad de procesos posible de forma que la interacción de la persona con el sistema sea fácil e intuitiva facilitando el proceso de transferencia de conocimientos y aprendizaje, algo tan difícil cuando se trata de asimilar nuevas tecnologías.” **[IT01]**

El acelerado desarrollo de nuevas tecnologías, hace que las mismas se hagan obsoletas más rápidamente y paralelamente a esto se suma el desarrollo del software, que por sus nuevas funcionalidades también requiere mayor disponibilidad de recursos de hardware. Pero en algunas ocasiones este fenómeno provoca que se desechen tecnologías todavía útiles en otros entornos donde el consumo de recursos sea menor, dígase en bibliotecas, centros de consulta, hospitales, centros de navegación por Internet, etcétera.

“Hace muchos años (desde finales de los 90) que se viene hablando en el mundo de los *clientes ligeros*: máquinas poco potentes, sin disco duro (también conocidas por su término en inglés *diskless*), que sustituirían a los ordenadores de sobremesa normales. Todos los programas y los documentos residirían en un servidor central, que contendría la configuración. De esta forma toda la configuración y administración podría hacerse en remoto, reduciendo el coste de mantenimiento al mínimo” **[IT02]**.

El proyecto Nova dedica una de sus líneas productivas al soporte de máquinas sin disco duro o *diskless*. El proceso de configuración de un servidor para *diskless* actualmente se realiza de forma manual y es bastante complicado por lo que se hace necesaria la creación de una solución que permita la configuración y administración, de manera más rápida y sencilla, de los servidores involucrados por parte de usuarios

con pocos conocimientos de los procesos y protocolos internos de diferentes servicios de red, que pudieran estar controlados bajo la interfaz gráfica de una aplicación más amigable.

El precio de un diskless es mucho menor que el de una computadora estándar, esto permite que a un menor costo se puedan desplegar o poner en marcha mayor cantidad de terminales en centros de trabajo, bibliotecas y demás; a las cuales se sumarían aquellas máquinas supuestamente obsoletas tecnológicamente. Para el funcionamiento de los diskless es necesaria la configuración de un grupo de servicios orientados a un servidor. Producto de esto se nos plantea la **situación problemática** que en el momento de realizar esta investigación la *configuración de los servidores* de diskless se hace manualmente y solo puede ser realizada por personas altamente preparadas en el tema de servidores de GNU/Linux.

Por lo antes expuesto el **problema científico** consiste en: ¿Como diseñar e implementar una solución que permita, de forma más rápida y sencilla, la *configuración de los protocolos* necesarios para el correcto funcionamiento de los diskless con servidores que posean sistema operativo GNU/Linux?

Por tanto el **objeto de estudio** de este trabajo abarca los diskless así como los diferentes protocolos y sus respectivas configuraciones para que los mismos funcionen adecuadamente enmarcándose en el **campo de acción** la *configuración de los protocolos* necesarios para el correcto funcionamiento de los servidores de diskless en el sistema operativo Nova; planteándonos como **idea a defender** que el desarrollo de una solución, que permita la configuración automática de los proveedores de servicios necesarios para el funcionamiento adecuado de los diskless con el sistema operativo Nova, permitirá que este proceso se haga de manera más rápida, eficiente y sencilla.

El **objetivo general** del presente trabajo es: Desarrollar una solución con software libre para sistemas operativos GNU/Linux, que permita el despliegue de los servidores de diskless en Cuba.

Desglosándose en los siguientes **objetivos específicos**:

- Caracterizar y sistematizar las aplicaciones que permiten la configuración de servidores de diskless existentes en el mundo, con sus principales rasgos.
- Modelar el funcionamiento de una propuesta de solución para la configuración de servidores de diskless.

- Implementar y probar la aplicación con la plataforma de desarrollo y la metodología seleccionada.

Para dar cumplimiento a los objetivos específicos nos trazamos las siguientes **tareas de investigación**:

- Realizar un levantamiento acerca de la presencia de sistemas similares al propuesto por medio de una búsqueda en Internet y en algún otro recurso bibliográfico.
- Comparar críticamente, en caso de que exista, una aplicación con la óptica de identificar posibles funcionalidades no implementadas; o su adecuación al entorno específico cubano.
- Modelar el funcionamiento de la aplicación propuesta haciendo uso de un lenguaje de modelado.
- Programar haciendo uso del lenguaje de programación seleccionado la aplicación propuesta en este trabajo.
- Realizar pruebas de calidad a la aplicación.

Los autores pretenden obtener como **resultado** una solución que incluya una aplicación de escritorio que constituya una “Herramienta para la configuración de servidores de diskless con sistemas operativos Nova (denominado en lo adelante Lira)”, así como una imagen optimizada de Nova para el uso en diskless y una kernel parcheado con AUFS.

Con la utilización de esta herramienta los administradores de servidores de diskless podrán contar con una solución basada en entorno gráfico para configurar adecuadamente los protocolos de red, ficheros y otros recursos necesarios para lograr que el usuario final de los diskless quede satisfecho con el producto. El mismo posibilitará que cualquier usuario con conocimientos básicos del tema, redes y diskless, pueda utilizar Lira y montar un servidor que suministre imágenes a máquinas clientes.

Para organizar el presente documento el contenido se ha dividido en cuatro capítulos estructurados de la siguiente manera:

Capítulo 1. Fundamentación Teórica. En este capítulo se realiza un estudio de las aplicaciones gráficas para la configuración de servidores de terminales ligeras a nivel nacional e internacional; así como la metodología, herramientas y lenguajes necesarios para el desarrollo del sistema propuesto. Se muestran los conceptos

básicos necesarios para la comprensión del funcionamiento de una aplicación de este tipo, además de todas las facilidades que debe brindarle al usuario.

Capítulo 2. Planificación y Definición de la aplicación. Se describe como se realiza actualmente la configuración manual de los clientes ligeros sin una herramienta basada en un entorno gráfico y se describe la propuesta del sistema. Se realiza el modelado del negocio y del sistema incluyendo las plantillas que se generan según la metodología utilizada, que permiten una mayor comprensión del proceso de desarrollo.

Capítulo 3. Desarrollo Ágil de Lira. Este capítulo se centra en la aplicación de la metodología escogida para desarrollar la aplicación propuesta; se hace un recorrido por las fases Implementación y Pruebas de dicha metodología y se realiza la planificación de las iteraciones del proyecto.

Este documento contiene además, Conclusiones, Recomendaciones, Bibliografía, Anexos y Glosario de Términos.

1 Fundamentación Teórica

*¿Por qué esta magnífica tecnología científica, que ahorra trabajo
y nos hace la vida mas fácil, nos aporta tan poca felicidad?
La repuesta es está, simplemente: porque aún
no hemos aprendido a usarla con tino.*

Albert Einstein

Resumen. En el presente capítulo se definirá qué es un diskless, en qué se basa su funcionamiento y los diferentes procesos que se llevan a cabo para la configuración de un servidor de diskless. También se describirán las herramientas libres o apropiativas que permiten la configuración de los servidores necesarios para servir correctamente un grupo de diskless conectados en red así como las herramientas, lenguajes y metodologías de desarrollo que se van a utilizar en la elaboración del sistema propuesto.

1.1 Diskless

Son máquinas pequeñas sin disco duro que arrancan por la red tomando de un servidor la imagen que le permite al usuario disponer de un Sistema Operativo y un paquete de aplicaciones y herramientas en su puesto de trabajo. Estos diskless dependen de un servidor o estación de trabajo principal que es la encargada de distribuir las imágenes por cada una de las terminales que se conecten. Los diskless basan su eficiencia en la utilización de los recursos mínimos para su funcionamiento (poco procesamiento de datos) por lo que no necesitan de una máquina potente, sino que dejan este trabajo al servidor.

A diferencia de otras máquinas de escritorio, en los diskless solo necesitaremos que el equipo tenga soporte para *arranque por red (etherboot)*, Preboot Execution Environment o Entorno de Ejecución Prearranque (en lo adelante PXE) **[PXE00]** o en su defecto una disquetera para etherboot.

1.2 Ventajas e Inconvenientes

Ventajas

- **Mejora del rendimiento de la red.** Puede incrementar el rendimiento de la red virtual que utiliza para la relación cliente-servidor, reduciendo el tráfico de datos. Esto le permite sustituir un flujo elevado de datos por información correspondiente al escritorio de trabajo.
- **Reducción de costes.** Como no se necesita procesar gran cantidad de datos no se requiere de un equipo potente lo cual hace que el precio de los diskless sea más barato que el de una computadora convencional. Además reduce el costo de instalación y mantenimiento de la red corporativa por la que se realiza el flujo de datos cliente-servidor. Aprovecha los equipos en desuso, lo que minimiza los gastos en hardware nuevo. Gestiona los datos de forma centralizada, esto permite ahorrar en software y mantenimiento del mismo.
- **Incremento de la seguridad.** Al estar toda la información concentrada en el servidor se incrementa el nivel de seguridad de la red con restricciones de acceso físico.
- **Máxima fiabilidad.** La utilización de un menor número de componentes hardware le permite incrementar la fiabilidad y calidad de los mismos.
- **Disminución del consumo energético.** En estas computadoras el consumo energético disminuye grandemente puesto que carecen de disco duro, el procesador es bastante pequeño y poco potente al igual que la tarjeta de video.
(Ver Anexo 1 - ¿Cuánto consume una computadora?)

Inconvenientes

La utilización de diskless no es adecuada cuando se requiere una gran potencia desde los terminales, como puede ser el procesamiento de vídeo y aplicaciones de Diseño Asistido por Ordenador (CAD). Además los clientes ligeros no permiten el almacenamiento de gran cantidad de información personal por parte del usuario. Esta tecnología tampoco es viable cuando se desea utilizar el ordenador para correr juegos que requieran una potencia gráfica bastante avanzada.

En resumen los clientes ligeros no son máquinas personales más pequeñas, sino que son terminales para el uso en entornos específicos donde se quiera divulgar mucha información a un gran número de usuarios o donde se requiera de una labor

simultánea por parte de un grupo de trabajo. Pueden ser utilizados en diferentes escenarios como son: escuelas, hospitales, centros de navegación Internet entre otros.

1.3 Comparación de hardware

Para montar un grupo de máquinas diskless es necesario tener un servidor que les provea de los servicios necesarios para su correcto funcionamiento.

La configuración de hardware de los servidores está en dependencia principalmente del sistema operativo con el cual se quieran montar los diskless. A continuación los autores ilustran con una tabla comparativa la necesidad de hardware requerida para sistemas de la familia GNU/Linux y Microsoft Windows.

| Servidor con GNU/Linux | Servidor con Windows |
|---|---|
| Procesador 3.0 Ghz o mayor | Procesador 3.0 Ghz o mayor. |
| 1 GB de Memoria RAM | 2 GB (3 GB recomendado) de Memoria RAM |
| Placa de red de 1Gbps | Placa de red de 1Gbps |
| Disco Rígido 80 GB | Disco Rígido 100 GB |
| Sistema Operativo GNU/Linux. | Sistema Operativo Windows |
| Cantidad de Terminales que conecta: 18 a 21 | Cantidad de Terminales que conecta: 18 a 21 |
| Terminales con GNU/Linux | Terminales con Windows |
| Procesador Pentium 66 MHz o mayor | Procesador 366MHz o mayor |
| 16 MB RAM (32 MB recomendado) | 128 MB RAM |
| Placa de red 10 Mbps (PCI recomendado) | Placa de red 10 Mbps (PCI recomendado) |
| Monitor | Monitor |
| Teclado | Teclado |
| Mouse | Mouse |

Tabla 1.1 Comparación de hardware necesario para Diskless en dos sistemas operativos diferentes.¹

1.4 Fundamento de los Diskless

Cuando se habla de diskless es imposible dejar de mencionar dos partes importantes en esta tecnología: el servidor y el cliente. El servidor contiene los protocolos, bien configurados, para lograr que los clientes cumplan su función de hacer que el usuario final se sienta a gusto con el servicio que se le brinda a través de una máquina que no tiene un disco duro. El servidor además sirve la imagen a los clientes para realizar tra-

¹ Información extraída de los sitios de las herramientas para configurar diskless Linux Terminal Server Project (GNU/Linux) y Winconnect (Windows), explicadas y referenciadas en el presente documento más adelante.

bajos como revisar el correo, o una información en la Web, utilizar los paquetes ofimáticos, etc.

Los diskless se pueden implementar en dos modalidades diferentes: con *procesamiento del lado del servidor* o con *procesamiento del lado del cliente*. Cada una de estas modalidades ofrece ventajas y desventajas que la hacen apropiada para un entorno específico.

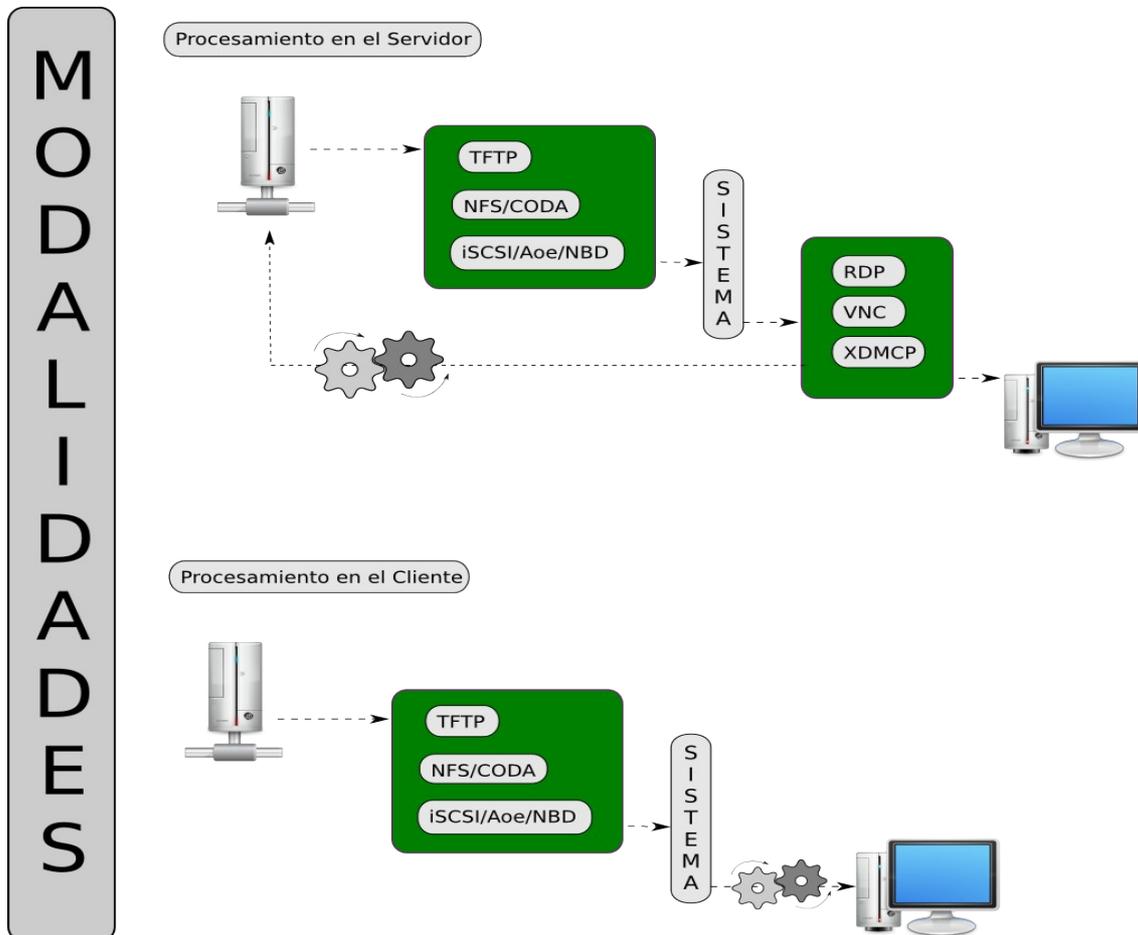


Figura 1.1 Protocolos para compartir la imagen en dependencia de la modalidad utilizada.

Procesamiento del lado del servidor

Características

- El servidor exporta el sistema operativo mediante NFS de sólo lectura.
- El sistema operativo puede estar comprimido usando squashfs¹ para mejorar el rendimiento del servidor.

¹ Formato de compresión de archivos.

- El objetivo final del cliente es conectarse al servidor mediante los protocolos XDMCP¹ (X Display Manager Control Protocol o Protocolo de Control de Administrador de la Pantalla X) [XDMCP00] o VNC² [VNC00] (Virtual Network Computing o Computación en Red Virtual).
- Permite conectarse mediante el protocolo RDP³ [RDP00] (Remote Desktop Protocol o Protocolo de Escritorio Remoto) para Windows.
- El cliente no procesa prácticamente nada ya que todo se procesa del lado del servidor.

Ventajas

- Los ordenadores de clientes puede ser de bajas prestaciones ya que el procesamiento se hace en el servidor.
- La administración de los clientes es menos complicada puesto que todos trabajan directamente en el servidor.
- Posibilita una tasa alta de procesamiento en dependencia de las características del hardware del servidor y la cantidad de clientes conectados.
- El tráfico por la red es menor en comparación con la modalidad del *procesamiento del lado del cliente* lo que lo hace factible en redes de poco ancho de banda.

Desventajas

- Soporta un límite de conexiones bastante bajo directamente proporcional a las capacidades de hardware del servidor puesto que cada cliente tiene que conectarse remotamente al servidor para poder funcionar.
- Si una aplicación congela⁴ el sistema del servidor se ven afectados todos los clientes.

¹ Es un protocolo utilizado en redes de computadoras para comunicar un ordenador servidor con un gestor de ventanas basado en X-Window con el resto de los clientes que se conectarán a este con propósitos interactivos.

² Es un software libre basado en la arquitectura cliente-servidor que nos posibilita controlar el ordenador servidor a través de un cliente remoto. Permite que el sistema operativo en cada cliente sea diferente.

³ Es un protocolo creado por Microsoft que permite la comunicación en la ejecución de una aplicación entre un terminal y un servidor Windows.

⁴ Hace referencia a cuando el sistema por algún motivo se paraliza impidiendo su funcionamiento.

- Demanda mayor responsabilidad del administrador con respecto a configuraciones internas del servidor u otros datos sensibles como pueden ser las versiones del software instalado.
- Es más susceptible a ataques con el uso de exploits¹.

Procesamiento del lado del cliente

Características

- El sistema comparte el sistema operativo mediante NFS.
- Procesa todo en el cliente.

Ventajas

- Solamente es necesario un kernel con soporte para NFS del lado del servidor.
- Es capaz de soportar gran número de conexiones en dependencia de la capacidad de la red.
- No se requiere que el servidor posea un hardware muy potente, cualquier máquina decente puede funcionar como un servidor.
- Permite aprovechar el hardware de los clientes para el procesamiento de datos.
- Permite home compartido².

Desventajas

- Hace que la carga de la red sea mayor puesto que por cada cliente es necesario establecer dos conexiones con el servidor.
- El kernel del cliente debe estar parcheado con AUFS que soporte NFS.

El uso de una modalidad u otra está en dependencia de los objetivos con los que se quieran usar y con el hardware tanto del lado del servidor como del lado del cliente que se tenga. Por ejemplo si se cuentan con diskless con buenas prestaciones (como los que está comprando nuestro país) y de una red con capacidad aceptable y servidores de bajas prestaciones es más adecuado utilizar la modalidad del *procesamiento de lado del cliente*. En otro caso donde la red y las computadoras

¹ Es un software, una secuencia de comandos o un fragmento de código que aprovecha una vulnerabilidad del sistema con la finalidad de causar un comportamiento anómalo en un programa informático.

² Es una técnica que permite que los usuarios en una subred tengan disponible su home (similar al Documents and Settings de Windows) en cualquier ordenador conectándose a un servidor que se los provee por NFS.

clientes no sean muy potentes y los servidores consten con un hardware potente es más factible aplicar la modalidad del *procesamiento del lado del servidor*.

De forma general en cualquiera de las dos modalidades, en el servidor existe el protocolo:

■ **DHCP(Dinamic Host Control Protocol o Protocolo de Control Dinámico de Anfitrión)[DHCP00]:** Para lograr asignar el ip, la subred, la máscara, el directorio raíz a ser montado por NFS y el o los servidores DNS, para la salida a Internet o para que tengan al menos red interna las terminales clientes. Además incluye la dirección de la imagen que debe cargar este cliente.

También se incluyen otros protocolos a través de los cuales se va a transmitir la información y los datos para el funcionamiento de los diskless como por ejemplo la imagen a servir, el kernel, el sistema de archivos y otros; que estarán en dependencia de la modalidad que se utilice y que se explicarán en detalle más adelante.

Del lado del cliente, solo existen demonios¹ que serán capaces de estar a la escucha en caso de que el servidor se reinicie o algún otro inconveniente.

A continuación mostramos una figura donde se ilustran cada uno de los procesos que ocurren en la modalidad de procesamiento del lado del servidor con XDMCP.

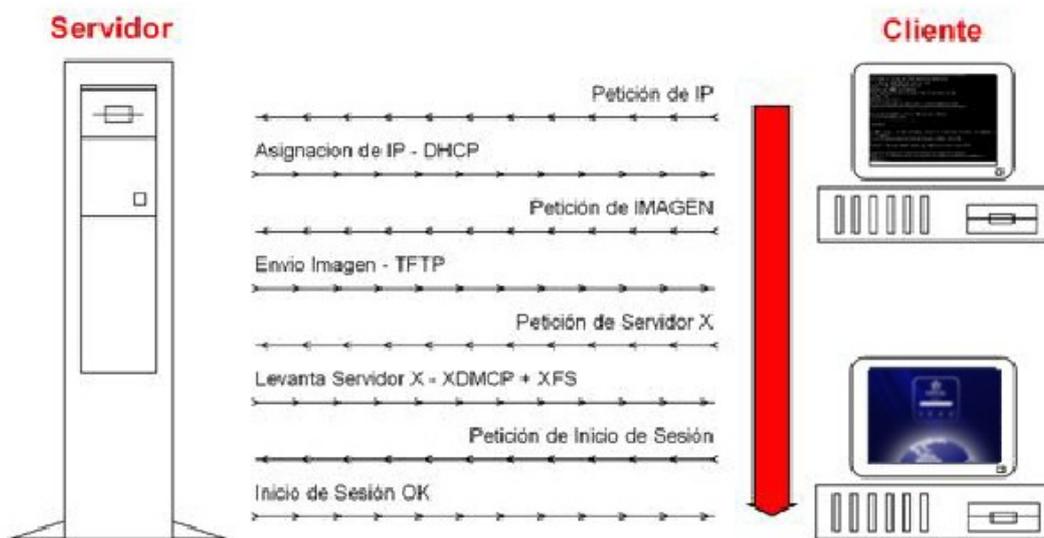


Figura 1.2 Procesamiento en el Servidor con XDMCP

¹ De las siglas en inglés D.A.E.M.O.N (Disk And Execution Monitor o Monitor de Ejecución y Disco), es un tipo especial de proceso informático que se ejecuta en segundo plano en lugar de ser controlado directamente por el usuario(es un proceso no interactivo). No tienen una interfaz de interacción con el usuario y mientras se activen no dejan de trabajar.

Procesos para la configuración de un diskless

Proceso de petición y entrega de IP¹ (DHCP)

Este proceso comienza una vez que una PC cliente se enciende. Esta buscará en la red algún servidor a quien mandarle una solicitud de IP, una vez que lo encuentra le envía la petición. El servidor por su parte lo que registra de esta solicitud es la MAC² del cliente o sea el número de la tarjeta de red, le asigna un IP de los que tiene en el rango de asignación del DHCP que está en la máquina servidora permitiendo que a través de un gateway³ (puerta de enlace) y un DNS (Domain Name System)[DNS00] los clientes puedan navegar.

El servidor DHCP verá el broadcast⁴ (difusión) y responderá el pedido, leerá su archivo de configuración y localizará la entrada que coincide con la dirección MAC que fue enviada. Entre la información que envía el servidor a los diskless podemos encontrar:

- Dirección IP asignada al terminal.
- Máscara de red configurada para la red local.
- Directorio del cual extraer el kernel.
- Nombre del kernel a descargar.

El bootcode recibirá la respuesta del servidor DHCP y configurará la interfaz TCP/IP de la placa de red con los parámetros que fueron suministrados.

Proceso de petición y entrega de inicio (PXE)

A través de esta herramienta los ordenadores pueden iniciar desde un servidor de una red antes que iniciar por su propio disco.

El cliente intenta levantar una imagen por la red y el PXE le envía información de:

- La imagen que debe levantar.
- El kernel.

¹ Número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (generalmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol).

² En redes de computadoras la dirección MAC (Media Access Control Access o dirección de control de acceso al medio) es un identificador de 48 bits (6 bytes) que corresponde de forma única a una interfaz o tarjeta de red.

³ Es un dispositivo que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación. Su propósito es traducir la información del protocolo utilizado en una red al protocolo utilizado en la red de destino.

⁴ Es un modo de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea sin necesidad de reproducir la misma transmisión nodo por nodo.

- Le asigna los permisos sobre esa imagen.

Además de otras cosas que necesita conocer la máquina cliente para lograr mostrar una imagen. Una vez que el cliente tiene esta información está listo para extraer la imagen del servidor.

Proceso de transferencia y recepción de datos:

Existen varios protocolos mediante los cuales se puede implementar este proceso que están en dependencia de la modalidad que se ponga en práctica ya sea *procesamiento del lado del servidor* o *procesamiento del lado del cliente*. Estos ayudarán a enviar las imágenes de arranque o cualquier tipo de datos a los clientes que las soliciten.

El bootcode¹ enviará un pedido a través de uno de estos protocolos al servidor para comenzar a descargar el kernel, según la información adquirida del PXE. Una vez que el kernel esté completamente descargado en el terminal, el bootcode hará un salto hacia el código de inicio del mismo. El kernel comenzará a ejecutarse, inicializando todo el sistema y todos los periféricos. Luego hará otro pedido al servidor DHCP, el cual responderá enviando la información que necesita para continuar, por ejemplo:

- Dirección IP asignada al terminal.
- Máscara de red configurada para la red local.
- El gateway.
- El servidor DNS.
- El hostname (nombre del anfitrión)² del terminal.

A continuación se explican cada uno de los posibles protocolos a utilizar para el proceso de transferencia y recepción de datos.

TFTP

Este es un protocolo bastante sencillo utilizado para la transferencia de archivos por la red. Está diseñado para ser pequeño y fácil de implementar por lo que no cuenta con muchas de las características de FTP (File Transfer Protocol o Protocolo de Transferencia de Archivos). Lo único que puede hacer es leer y escribir de un servidor

¹ Hace referencia al master boot code (código de arranque maestro) que le indica a una computadora como debe arrancar el sistema, en este caso por la red.

² Programa que se utiliza para mostrar o establecer el nombre actual del sistema (nombre del equipo).

remoto. No puede listar directorios y no posee mecanismo de autenticación de usuarios.

Básicamente cualquier transferencia utilizando este protocolo comienza con una solicitud de lectura o escritura de un archivo, que también sirve como solicitud de conexión. Si el servidor concede el pedido, se abre una conexión y se envía el archivo en pequeños paquetes de 512 bytes. Cada vez que un paquete es enviado debe ser confirmada su recepción por parte del receptor para continuar con el envío del próximo paquete. En caso de que se pierda un paquete en la red, el receptor, luego de un tiempo prefijado, enviará una señal de retransmisión para que el servidor envíe nuevamente el paquete enviado. La conexión termina cuando el servidor recibe un mensaje de error desde el receptor o cuando la transmisión ha concluido.

NFS/CODA

Son sistemas de archivos distribuidos que permiten el almacenamiento de archivos en uno o más servidores y los hacen accesibles a los clientes como si se trataran de archivos locales. Los sistemas distribuidos ofrecen muchas ventajas:

- Los archivos están más accesibles.
- Compartir archivos de una sola localización es más sencillo que hacer copias de los mismos en todos los clientes.
- Las copias de respaldo y de seguridad son más fáciles de hacer.

Los dos sistemas de ficheros distribuidos más conocidos son NFS y CODA.

NFS

Este protocolo provee acceso remoto transparente a archivos compartidos en la red. Está diseñado para ser portable entre diferentes máquinas, sistemas operativos, arquitecturas de red y protocolos de transporte. Este protocolo permite compartir el sistema de archivos completo de un sistema como si fuera el archivo local de un cliente. Todas las operaciones sobre ficheros son síncronas, es decir que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esta 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 logrando de esta manera la integridad de los ficheros.

CODA

Este sistema de ficheros en red avanzado fue desarrollado en la Universidad Carnegie Mellon. Es bastante prometedor y potencialmente importante porque consta de algunas características que no se encuentran en ningún otro sistema como son:

- Altas prestaciones a través de una cache¹ en el lado del cliente.
- Replicación en varios servidores.
- Buena escalabilidad².
- Adaptación al ancho de banda.
- Funcionamiento continuado durante fallos producidos en la red del servidor.
- Posibilidad de funcionamiento desconectado para procesadores móviles.
- Modelo de seguridad para autenticación, encriptación y control de acceso.
- Semántica bien definida de compartición, incluso con fallos en la red.
- Software libre disponible de manera gratuita. **[CODA00]**

iSCSI/Aoe/NBD

Son sistemas de transferencia de archivos por bloque que permiten compartir datos en redes locales.

iSCSI

Es un estándar oficial que permite el uso de del protocolo SCSI³ (Small Computers Interface o Sistema de Interfaz para Pequeñas Computadoras) sobre redes TCP/IP. Este protocolo utiliza TCP/IP para la transferencia de datos y solamente requiere de una simple y sencilla interfaz Ethernet⁴ para funcionar. Permite una solución de almacenamiento centralizada de bajo costo.

En el contexto de almacenamiento, iSCSI posibilita a un ordenador utilizar un iniciador iSCSI (initiator) para conectar a un dispositivo SCSI (target) como puede ser un disco duro o una cabina de cintas en una red IP para acceder a los mismos a nivel de

¹ Es un conjunto de datos duplicados de otros originales, con el objetivo de disminuir el tiempo de acceso a los datos originales al tener una copia local.

² Hace referencia al proceso mediante el cual en una red se aumentan la cantidad de terminales o usuario afectando al mínimo el rendimiento de la misma.

³ Es una interfaz estándar para la transferencia de datos entre distintos dispositivos de una computadora, se utiliza principalmente en discos duros aunque también interconecta una amplia gama de dispositivos como scanners, unidades CD/DVD, impresoras, etc.

⁴ Es un estándar de redes de computadoras de área local.

bloque. Desde el punto de vista de los drivers¹ y las aplicaciones de software, los dispositivos parecen estar conectados realmente como dispositivos SCSI locales. Los entornos más complejos, consistentes en múltiples hosts y/o dispositivos son llamados redes de área de almacenamiento. **[iSCS00]**

AoE

Es un protocolo de red diseñado para acceder a dispositivos de almacenamiento ATA² (Advanced Technology Attachment o Tecnología Avanzada de Archivos Adjuntos) mediante redes Ethernet.

En esencia la idea de este protocolo es reemplazar el cable físico que va desde el puerto ATA al disco duro por un interfaz Ethernet permitiendo que los discos duros sean accesibles desde la red. Los datos son escritos al disco duro y leídos desde el dispositivo mediante bloques de datos que viajan por la red.

El objetivo de AoE es manejar un disco duro o un dispositivo de almacenamiento masivo a través de la red permitiendo compartir de esta forma datos y archivos en una red local para varios clientes. **[AoE00]**

NBD

Network Block Device o Dispositivo de Bloque de Red. Típicamente es utilizado para acceder a un dispositivo de almacenamiento que no reside físicamente en la máquina local sino en una remota. En la máquina cliente el dispositivo parece una partición de un disco duro pero es en realidad un direccionamiento al disco duro físico que se encuentra en el servidor. Cada vez que la computadora cliente quiera leer o escribir en la partición se enviará un pedido vía TCP al servidor que responderá con la información solicitada. **[NBD00]**

Los procesos mencionados anteriormente son similares para cualquiera de las dos modalidades, la diferencia radica principalmente en el proceso de compartir los archivos de la imagen, como se puede observar en la Figura 1.1, que puede hacerse a través de una conexión de escritorio remoto o utilizando cualquiera de los protocolos de transporte de archivos explicados con anterioridad. En el caso de la modalidad de *procesamiento del lado del servidor* es necesario una conexión de tipo escritorio remoto mediante XDMCP, RDP o VNC para lograr que el cliente funcione correctamente.

¹ Es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz para usarlo.

² Es el puerto que controla los dispositivos de almacenamiento masivo de datos como los discos duros.

Proceso de Compartir archivos de la imagen por la red:

Los protocolos que intervienen en este proceso también están en dependencia de la *modalidad* que se implemente ya sea *procesamiento del lado del servidor* o del *lado del cliente*. A continuación se explicaran los protocolos que intervienen en la modalidad de *procesamiento del lado del servidor* puesto que en la otra modalidad se utilizan los mismos protocolos explicados en el proceso anterior.

XDMCP-XFS

Es un protocolo para compartir archivos en una red. Funciona como una conexión de escritorio remoto a la máquina servidora.

- Luego de bajar el sistema base (que pesa unos pocos MB) es necesario utilizar una serie de aplicaciones que por su tamaño no son práctico copiarlas en la RAM.
- El servidor publica con XFS un sistema más completo.
- El cliente utiliza los archivos (y programas) que están en el servidor (cargando en memoria lo indispensable).

RDP

Es un protocolo desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre un terminal y un servidor Windows.

El modo de funcionamiento es bastante simple, la información gráfica generada en el servidor es convertida a un formato RDP para ser enviada a la computadora cliente, la cual interpretará la información y mostrará la imagen en la pantalla del terminal. Cualquier acción que realice el usuario será redirigida hacia el servidor que la completará y enviará la respuesta al terminal. Sus características más notables son:

- El redireccionamiento del audio permite al usuario ejecutar un programa de audio en una ventana remota y escuchar el sonido en el ordenador local.
- El redireccionamiento del sistema de ficheros permite a los usuarios utilizar sus ficheros locales en una ventana remota.
- El redireccionamiento de puertos permite utilizar los puertos serie y paralelo directamente.
- El portapapeles puede compartirse entre los ordenadores local y remoto.
- Programas remotos: aplicaciones con ficheros del lado del cliente.

- Ancho de banda ajustado para clientes RDP.
- Soporte de varios monitores. La sesión puede mostrarse en dos monitores.

VNC

Es un software libre basado en una arquitectura cliente-servidor que nos permite establecer un escritorio remoto hacia un servidor desde una computadora cliente. Da la posibilidad que el sistema operativo en cada computadora sea distinto y además permite compartir cualquier sistema operativo. Las características fundamentales de VNC son:

- El estado del sistema no se almacena en la máquina local. Es decir uno puede estar trabajando en una computadora, sentarse en cualquier otra y cuando establezca la conexión las aplicaciones estarán en el mismo estado que se dejaron en la otra máquina. Además si se reinicia o se apaga la máquina inesperadamente se conservan en el estado que se dejaron antes de terminar la conexión los programas que se estaban ejecutando.
- Es pequeño y sencillo. Por ejemplo el visor de Win32 pesa unos 150 kilobytes y puede ser ejecutado directamente desde un floppy ya que no es necesario instalarlo.
- Es independiente de la plataforma.
- Es compartible, un mismo escritorio puede ser utilizado por varios usuarios a la vez.
- Es software libre y gratis.

En resumen:

Los diskless requieren un servidor de DHCP para determinar, de una MAC suministrada su ubicación en la red y darle una IP fija. Además necesitan una placa base que permita arranque a través de PXE por la red. Del lado del servidor es imprescindible configurar el protocolo que se encargará de la transmisión de los datos para poder transmitir la ROM (Read Only Memory o Memoria de Solo Lectura)¹ de arranque y posteriormente proveer el kernel del sistema que se configurará según los argumentos especificados en la configuración del PXE, para que posteriormente el kernel sepa donde buscar los ficheros del sistema en el servidor, Los cuales se transmitirán por NFS u otro protocolo en dependencia de la modalidad que se utilice.

¹ Es una tipo de medio de almacenamiento utilizado en las computadoras y otros dispositivos electrónicos. Los datos almacenados en el mismo no se pueden modificar.

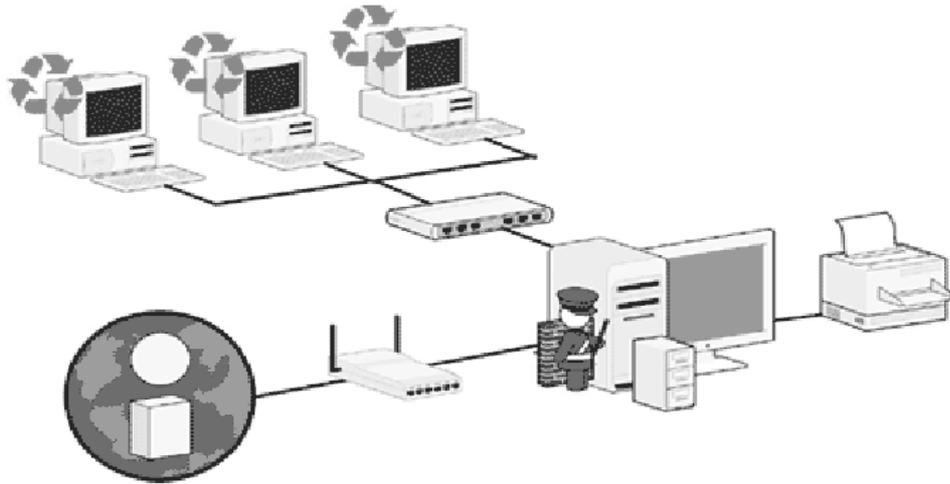


Figura 1.3 Ejemplo de diskless conectados a un servidor.

1.5 Aplicaciones para servidores de diskless

A continuación haremos una descripción de diversas aplicaciones que se utilizan a nivel mundial para la configuración de los servidores de diskless.

Se tendrán en cuenta los siguientes aspectos para la comparación entre estas aplicaciones:

- Funcionamiento de la aplicación.
- Protocolos que se configuran con la aplicación.
- Requerimientos de Hardware en el servidor.
- Sistemas operativos que brinda para el arranque.
- Licencia.
- Otras características propias.
- Comparación con la aplicación propuesta en este documento.

Aplicaciones Libres

Lan Core[LC00]



Figura 1.4 Aplicación Lan Core.

■ **Funcionamiento de la aplicación.** Es una herramienta para la gestión de diskless. Proporciona información actualizada de los terminales conectados, acceso a su configuración y control sobre la ejecución de los terminales.

■ **Protocolos que se configuran con la aplicación.** XDMCP y RDP.

■ **Requerimientos de Hardware en el servidor.**

- Memoria: 512MB.
- Disco Duro: 100MB libres.

■ **Sistemas operativos que brinda para el arranque.** Se puede elegir el sistema de arranque de las diskless a través de un menú de inicio:

- Windows Server 2000.
- Windows Server 2003.
- Windows XP Profesional.
- Linux: Debian, Red Hat, Suse.

■ **Licencia.** En el sitio de la aplicación no viene reflejada la licencia bajo la cual se distribuye.

■ **Otras características propias.**

- Permite la configuración de los terminales conectados:
 - ◆ Dirección del servidor y protocolo utilizado.
 - ◆ Nombre de usuario y clave de acceso.
 - ◆ Resolución de escritorio: ancho, alto y número de colores.

- ◆ Resolución del monitor: refresco horizontal y vertical.
- ◆ Idioma del teclado.
- Tendrá un control absoluto sobre sus terminales.
 - ◆ Podrá comprobar cuando están conectadas.
 - ◆ Apagar o reiniciar remotamente los terminales.
- Soporte para múltiples idiomas: Español, Inglés, Portugués y Catalán.
- En su versión para los sistemas operativos de Microsoft Windows, puede ser configurada para su uso con servidores DHCP externos.
- Soporta los protocolos:
 - ◆ RDP. Conexión con servidores Microsoft Windows por escritorio remoto.
 - ◆ XDMCP. Sesiones X remotas con servidores Linux.

■ **Comparación con la aplicación propuesta en este documento.** Este sistema es muy útil para la administración de los diskless, una vez que ya se tiene configurado cada uno de los protocolos que van del lado del servidor, por lo cual esta aplicación no cumple con la característica de ser capaz de configurar desde cero un servidor de diskless en la distribución Nova de GNU/Linux. Además solo permite implantar la modalidad de *procesamiento del lado del servidor* mediante la conexión remota de los clientes.

OpenThinClient[OTC00]



Figura 1.5 Logo de Open Thin Client.

■ **Funcionamiento de la aplicación.** Es una solución de código abierto y libre que consiste en un sistema operativo junto con un componente administrativo basado en una interfaz gráfica. Está pensado para ambientes donde se debe montar y administrar una cantidad media o grande de diskless de manera eficiente.

Tiene una filosofía de centralizar los servicios en un servidor determinado lo cual minimiza el trabajo administrativo del lado del servidor, donde se controlarán las terminales.

El inicio y la configuración de los diskless son implementados usando tecnologías estándares como LDAP (Ligthweight Directory Access Protocol o Protocolo Ligero de Acceso a Directorios)¹, DHCP, TFTP y NFS.

El diskless solo necesita una interfaz PXE para iniciar la imagen. Todos los datos de configuración se guardan en una base de datos LDAP que viene junto con el servidor Open Thin Client.

■ **Protocolos que se configuran con la aplicación.** DHCP, TFTP y NFS.

■ **Requerimientos de Hardware en el servidor.** Estos requisitos de hardware son para el servidor de los diskless.

- Mínimo 500 MB, recomendado 1 GB de RAM para Microsoft Windows XP, 2000 Server and Server 2003.
- Mínimo 1 GB, recomendado 1,5 GB RAM para Microsoft Windows Vista and Server 2008
- Mínimo 500 MB, recomendado 1 GB de RAM para GNU/Linux.
- Mínimo 1 GB, recomendado 2 GB de espacio libre en el disco duro.

■ **Sistemas operativos que brinda para el arranque.** Ubuntu optimizado para uso en diskless.

■ **Licencia.** Está a la disposición del que la quiera descargar bajo la licencia de GPL.

■ **Otras características propias.** El sistema de administración OpenThinClient así como el servidor de OpenThinClient están escritos en Java lo que lo hace multiplataforma.

■ **Comparación con la aplicación propuesta en este documento.** Esta aplicación no permite la configuración de los servidores para los diskless desde cero sino que viene con una configuración preestablecida para la imagen que va a servir, en este caso Ubuntu. Además como viene con una imagen preconfigurada para ser servida no permite que el usuario configure la imagen a servir a sus necesidades, aspecto que se encuentra entre los resultados esperados de la solución propuesta en este trabajo.

¹ Es un protocolo a nivel de aplicación (según el modelo OSI) que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. También es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden hacerse consultas.

LTSP (Linux Terminal Server Project) [LTSP00]



Figura 1.6 Logo de LSTP.

■ **Funcionamiento de la aplicación.** No es una aplicación sino un conjunto de ellas para servidores que permiten ejecutar GNU/Linux en computadoras de bajas prestaciones o de bajo costo.

Su funcionamiento consiste en repartir a través de la red un kernel de GNU/Linux que es ejecutado por los clientes que posteriormente ejecutarán secuencias de scripts¹ típicos de una minidistribución. 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.

■ **Protocolos que se configuran con la aplicación.** Este conjunto de aplicaciones configurarán los protocolos DHCP, TFTP, NFS y XDMCP para lograr el correcto funcionamiento de los servidores de diskless.

■ **Requerimientos de Hardware en el servidor.**

- Procesador: Pentium 4 a 3.0 Ghz.
- Memoria: mínimo 512 MB RAM, recomendado 1 GB.

■ **Sistemas operativos que brinda para el arranque.** Brinda soporte para varias distribuciones de GNU/Linux como Ubuntu, openSUSE, Fedora y Debian

■ **Licencia.** Se distribuye bajo licencia GPL.

■ **Otras características propias.** Está orientada solo a sistemas operativos GNU/Linux y el consumo de red es elevado.

■ **Comparación con la aplicación propuesta en este documento.** Esta aplicación es muy usada a nivel mundial, sobre todo en el sector educativo, para la implantación de la tecnología de diskless. El principal inconveniente que tiene

¹ En informática es un guión o conjunto de instrucciones que permiten la automatización de tareas creando pequeñas utilidades.

es que solo implementa la modalidad de *procesamiento en el lado del servidor* a diferencia del sistema propuesto que pretende incluir cualquiera de las dos modalidades de procesamiento tanto en el servidor como en el cliente.

PXES Universal Linux Thin Clients [PXE00]



Figura 1.7 Logo de PXES.

- **Funcionamiento de la aplicación.** Es una distribución optimizada de GNU/Linux que convierte cualquier computadora en un diskless capaz de conectarse a un servidor de XDMCP, VNC o RDP. Básicamente lo que hace es configurar un cliente para su conexión a un servidor de escritorio remoto.
- **Protocolos que se configuran con la aplicación.** Configura los protocolos XDMCP, RDP y VNC
- **Requerimientos de Hardware para el servidor.**
 - Procesador: Pentium 4 a 2.0 Ghz mínimo.
 - RAM: 512 MB mínimo.
- **Sistemas operativos que brinda para el arranque.** En general cualquier sistema operativo que soporte conexión a escritorio remoto como: GNU/Linux, Solaris, Windows, BSD y otros.
- **Licencia.** Se distribuye bajo la licencia GPL.
- **Otras características propias.** Es una aplicación bastante ligera que requiere 32 MB de RAM y un procesador con arquitectura x86 del lado del cliente.
- **Comparación con la aplicación propuesta en este documento.** Esta aplicación no configura nada del lado del servidor sino que configura un diskless para una conexión a escritorio remoto con un servidor que esté brindando este servicio lo cual no cumple con los objetivos del sistema propuesto en esta investigación.

TCOS Thin Client Operating System [TCOS00]



Figura 1.8 Logo de LSTP.

■ **Funcionamiento de la aplicación.** Es una herramienta para implementar la tecnología diskless. El proceso comienza cuando el diskless realiza una petición tipo DHCPREQUEST a la red. El servidor DHCP en ese segmento de red, le responderá y luego de algunos intercambios protocolares, le entregará la imagen del cargador de arranque *pxelinux.0*. Ésta imagen se encargará de pedirle al mismo servidor, el kernel del sistema que correrá en memoria del cliente, la que conocemos como *initramfs* para la primera parte de la carga y *vmlinuz* para quedar una vez finalizada la carga. Luego le solicitará la imagen que se copiará para la memoria RAM del cliente y se ejecutará.

■ **Protocolos que se configuran con la aplicación.** DHCP, TFTP y NFS.

■ **Requerimientos de Hardware en el servidor.**

□ Procesador: Pentium 4 o AMD.

□ Memoria: 300 MB además de 80 MB extras por cada cliente conectada (2-4 GB para 25 clientes).

■ **Sistemas operativos que brinda para el arranque.** Distribuciones de Debian y Ubuntu

■ **Licencia.** Distribuido bajo licencia GPL.

■ **Otras características propias.** Cuenta con una herramienta gencos para construir la imagen que se va a servir basada en Debian o Ubuntu.

■ **Comparación con la aplicación propuesta en este documento.** Esta aplicación en comparación con la propuesta en el presente trabajo tiene el inconveniente que solo está disponible para distribuciones de Debian y Ubuntu. Además solo realiza el procesamiento del lado del cliente.

Aplicaciones No Libres

WinConnect^[WC00]



Figura 1.9 Funcionamiento de WinConnect en una red de computadoras

■ **Funcionamiento de la aplicación.** WinConnect, de Thinsoft (c)¹ es un software que, instalado en cualquier computadora con Windows-XP² le permite actuar como servidor, admitiendo hasta 21 sesiones remotas con un procesador Pentium 4 a 3.0 Ghz y un mínimo de 2 GB de memoria RAM (3 GB recomendado). Permite la conexión mediante Wifi³, LAN⁴ (Local Area Network o Red de Área Local) o Internet, VPN⁵ (Virtual Private Network o Red Privada Virtual), de computadoras, PDAs⁶ (Personal Digital Assistant o Asistente Digital

¹ Compañía desarrolladora de software para clientes ligeros en general.

² Es una línea de sistemas operativos desarrollada por Microsoft. Se considera que se encuentran en el mercado 400 millones de copias funcionando. Las letras "XP" provienen de la palabra 'eXPeriencia' en inglés 'eXPerience'.

³ Es un sistema de envío de datos sobre redes de computadoras que utiliza ondas de radio en lugar de cables.

⁴ Es la interconexión de varios ordenadores y periféricos. Su extensión está limitada físicamente a un edificio o un entorno de 200 metros., con repetidores se podría abarcar un área de un kilómetro.

⁵ Es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet.

⁶ Es un computador de mano originalmente diseñado como agenda electrónica con un sistema de reconocimiento de escritura.

Personal), etcétera, compartiendo dispositivos y perifera del ordenador servidor.

Una conexión de banda ancha mejora considerablemente el rendimiento de esta aplicación aunque no es indispensable puesto que el RDP (Remote Desktop Protocol o Protocolo de Escritorio Remoto) transfiere solo el mínimo de datos posibles para controlar remotamente su computadora.

Los usuarios de diskless pueden ejecutar las diferentes aplicaciones simultánea e independientemente para crear o editar documentos y compartir periféricos como impresoras u otro hardware instalado en la computadora host (anfitrión). Todos los usuarios pueden también navegar en Internet y acceder al correo al mismo tiempo.

■ **Protocolos que se configuran con la aplicación.** DHCP y RDP.

■ **Requerimientos de Hardware.** Este sistema requiere una computadora con conexión a una red o Internet que tenga previamente instalado Windows XP o Windows Small Server 2003¹. Para Windows XP el mínimo de memoria recomendado para dos usuarios es de 256 MB mientras que para Windows Small Server 2003 se necesitan 384 MB. Además se necesitarán 64 MB adicionales por cada usuario extra que se pretende conectar al servidor.

■ **Sistemas operativos que brinda para el arranque.** Windows, en varias de sus versiones.

■ **Licencia.** La licencia con la que es distribuido es WinConnect Server XP License.

[WC01]

■ **Otras características propias.**

- Soporta RDP 5.1 para la conexión a servidores Windows NT (Nueva Tecnología) 4 Terminal Server², Windows 2000 Server, Windows XP Server y Windows XP Profesional.
- Soporta impresión local.
- Soporta sonido para lo cual es requerido un host XP.

¹ Es una suite integrada de servidor de Microsoft diseñado para el funcionamiento de la infraestructura diseñado para el funcionamiento de la infraestructura de la red de las pequeñas y medianas empresas que no tengan más de 75 estaciones de trabajo o usuarios.

² Es la cuarta versión de Windows NT. Es un sistema Windows de 32 bit disponible para estaciones de trabajo y versiones para servidores con una interfaz gráfica similar a Windows 95.

- Soporta color de pantalla de hasta 24 bit para lo cual se necesita un host XP.
- Interfaz fácil de usar que tiene configuración de menús.

■ **Comparación con la aplicación propuesta en este documento.** El principal problema de este sistema es que es propietario y tiene un costo de 99.95 dólares americanos lo cual la hace inaccesible e impropia para los objetivos de nuestra sociedad y nuestro proyecto de software libre. Además solo sirve sistemas de la familia de Windows.

1.6 Características que identifican a la herramienta Lira

Lira es una solución que incluye una herramienta de configuración para los servidores de diskless así como una imagen optimizada del sistema operativo para su uso en diskless y un kernel parcheado a AUFS. La herramienta Lira deberá ser capaz de preguntarle al administrador del servidor algunas características, que identifican la configuración de los protocolos del lado del servidor de una manera gráfica sin tener que acceder a los *archivos de configuración*¹. Con estas características básicas, que recibe del administrador, Lira será capaz de configurar dicho servidor para que sirva cierta cantidad de diskless. Lira tiene como requerimiento no funcional de software que estará orientado a la distribución Nova de GNU/Linux. En el arranque Lira será capaz de permitir que el administrador escoja la imagen que quiere para los diskless, especificar que siempre será alguna de las distribuciones de Nova: eduNova, serverNova y Nova (por el momento). Además Lira posibilitará configurar un servidor de diskless para cualquiera de las dos modalidades de procesamiento tanto del lado del servidor como del lado del cliente.

Lira usará, como producto final, la licencia libre GPL (Licencia Pública General).

1.7 Herramientas, lenguajes y metodologías a utilizar

En el desarrollo de todo sistema informático es de vital importancia la selección de las herramientas, lenguaje y metodologías a utilizar, paso que garantizará, de realizarse correctamente, un óptimo desempeño del sistema. En en caso específico de la

¹ Son archivos en en sistemas operativos GNU/Linux donde se encuentra la configuración de un servicio o una aplicación determinada.

presente investigación la mayoría de estos aspectos están definidos por políticas del proyecto al que pertenece el sistema que se pretende desarrollar.

Herramientas

Herramienta Case¹: BoUML.

Es una herramienta CASE gratuita (licencia GPL) para el modelado de diagramas con con UML² (Unified Modeling Language o Lenguaje Unificado de Modelado) 2.0 que permite especificar y generar código en los lenguajes de programación C++, Java, IDL, PHP y Python. Es multiplataforma ya que corre sobre los sistemas operativos Solaris, Unix, MacOS X y Windows. Es muy rápido y no requiere gran cantidad de memoria RAM para manejar unas cuantas miles de clases

Soporta gran cantidad de diagramas. Es rápida y apenas consume memoria y además es sencilla de utilizar.

Permite generar código para Java³, C++ e IDL y permite hacer reingeniería inversa. También es capaz de generar documentación en varios formatos (HTML⁴, XML⁵ y otros) y es una herramienta multiplataforma. **[BU00]**

Herramientas para la implementación

Eclipse

La Fundación Eclipse es una comunidad de código abierto, cuyos proyectos están encaminados a crear un entorno de desarrollo integrado (en lo adelante IDE⁶), de código abierto y multiplataforma compuesto por frameworks⁷ extensibles,

¹ Herramientas CASE (Ingeniería de Software Asistida por Ordenador): Diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

² Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

³ Lenguaje de Programación. Ídem para C++, IDL y Python en esta misma página.

⁴ De las siglas Hypertext Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

⁵ De las siglas en inglés eXtensible Markup Language (Lenguaje de Marcas Extensible) es un metalenguaje de etiquetas que permite definir la gramática de lenguajes específicos.

⁶ Responde a las siglas en inglés de Integrated Development Environment o Entorno de Desarrollo Integrado que un programa compuesto por un grupo de herramientas para un programador que le permiten desarrollar aplicaciones de manera más rápida y ágil.

⁷ Responde a las siglas en inglés de Integrated Development Environment o Entorno de Desarrollo Integrado que un programa compuesto por un grupo de herramientas para un programador que le permiten desarrollar aplicaciones de manera más rápida y ágil.

herramientas y rutinas para construir, desplegar y administrar software a lo largo de su ciclo de vida. El proyecto que más renombre ha alcanzado es el IDE del mismo nombre. Eclipse fue desarrollado inicialmente por IBM¹ (International Business Machines o Máquinas de Negocios Internacionales) pero ahora su desarrollo está en manos de la Fundación Eclipse.

Eclipse emplea módulos o complementos para proporcionar toda su funcionalidad a la plataforma, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software que le permiten a Eclipse extenderse usando otros lenguajes de programación como el C++ o Python. La arquitectura plugin (complemento) permite escribir cualquier extensión deseada en el ambiente, como sería el complemento de Eclipse para Qt², el cual integra el diseñador de interfaces, además de que completa y corrige las funciones de una forma muy óptima. Eclipse provee al programador frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc.

Principales Características:

- Editor de texto.
- Resaltador de sintaxis.
- Compilación en tiempo real.
- Prueba unitarias con JUnit³.
- Control de versiones con CVS⁴.
- Asistentes: para creación de proyectos, clases, test (prueba), etcétera.

Eclipse se distribuye actualmente bajo la licencia Eclipse Public License que es considerada por la Free Software Foundation⁵ (Fundación del Software Libre) como una licencia de software libre. La última versión de este IDE es la 3.4.2 vio la luz el 11 de febrero del 2009 que consta con paquetes específicos en dependencia del lenguaje en el que se programe.

¹ Es una empresa (conocida como el gigante azul) que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

² Es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario.

³ Es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

⁴ Sistemas de Versiones Concurrentes (del inglés Concurrent Versions Systems).

⁵ Es una asociación creada por Richard Stallman con el propósito de promover el software libre.

Geany

Es un pequeño IDE para GNU/Linux basado en GTK 2³ y con muy pocas dependencias. Es un editor de códigos muy estable, sencillo y rápido. Tiene resalte de sintaxis, autocompletado, atajos de teclado, muestra información de las funciones, tabulación automática, anidado de funciones según la apertura y cierre de paréntesis que se correspondan.

Las funciones son agrupadas en un árbol ordenado alfabéticamente que a la vez actúa como un acceso directo a las mismas. Dispone además de una consola de comandos integrada, con el intérprete que se quiera, bash⁴ (bourne again shell u otro shell bourne) por defecto.

Herramienta para la interfaz gráfica

Glade

Es una herramienta de desarrollo visual de interfaces gráficas mediante GTK/GNOME⁵; es independiente del lenguaje de programación y predeterminadamente no genera código fuente sino un archivo XML.

Lenguajes

Lenguaje de Programación

Python

Python es un lenguaje de programación que se compara habitualmente con otros lenguajes como: TCL, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de código abierto.

Python permite dividir el programa en módulos reutilizables desde otros programas Python. Viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas. También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema, sockets y hasta interfaces a GUI (interfaz gráfica con el usuario) como GTK, Qt entre otros. Ahorra un tiempo considerable en el desarrollo del

³ Es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario.

⁴ Es un Intérprete de órdenes de UNIX

⁵ GNOME es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix/Linux, compuesto enteramente de software libre.

programa, pues no es necesario compilar ni enlazar. Algunas de sus variables son similares a la de otros lenguajes como if, float y bool.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación estructurada y programación funcional. Otros muchos paradigmas más están soportados mediante el uso de extensiones. Python usa tipos de datos dinámicos y reference counting¹ (conteo de referencias) para el manejo de memoria. Una característica importante de Python es la resolución dinámica de nombres, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado ligadura dinámica de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Nuevos módulos se pueden escribir fácilmente en C o C++. Python puede utilizarse como un lenguaje de extensión para módulos y aplicaciones que necesitan de una interfaz programable. Aunque el diseño de Python es de alguna manera hostil a la programación funcional tradicional del lenguaje de programación Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia del Lisp como puede ser Scheme.

[PY00]

Python posee una licencia de código abierto, denominada *Python Software Foundation License*, que es compatible con la licencia GPL. Esta licencia no obliga a liberar el código fuente al distribuir los archivos binarios. **[PY01]**

Lenguaje de Modelado

UML

UML, es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema o un software orientado a objeto. Uno de los objetivos principales de la creación de UML es posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello es necesario definir una notación y semántica común. Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación. Constituye el lenguaje de la Ingeniería de Software y es utilizado no solo para la especificación de un sistema sino también para propósitos de comunicación entre las

¹ Es una técnica para contabilizar las veces que un determinado recurso está siendo referido. Por lo general ese recurso son bloques de memoria y la técnica permite establecer cuando no existe ninguna referencia a ese bloque y éste puede ser liberado.

personas involucradas en el desarrollo del mismo (ingenieros, administradores, líderes y otros), o para la documentación de software existente. **[UML00]**

Bibliotecas

GTK+

GTK+ o The GIMP Toolkit (conjunto de rutinas para GIMP) es un grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI) para los entornos gráficos, principalmente GNOME, XFCE y ROX de sistemas UNIX. Inicialmente fue creado para desarrollar el programa de manejo de imágenes GIMP, sin embargo en la actualidad es muy usada por muchos otros programas en los sistemas GNU/Linux. Este está escrito en C, pero puede ser usado desde otros lenguajes de programación como C++, C#, Java, Perl, PHP o Python 2.5.

Características

- **Estabilidad.** Brinda un excelente rendimiento para el desarrollo de aplicaciones. Además está soportado por una amplia comunidad de desarrolladores y los mantenedores principales de las compañías como son Red Hat¹, Novell² y Opened Hand³.
- **Multiplataforma.** Actualmente es usado en GNU/Linux, Unix, Windows y Mac OS X⁴ (Macintosh Operating System o Sistema Operativo de Macintosh).
- **Cómodo.** Abastece una serie de características que hoy en día los desarrolladores desean:
 - Apariencia agradable.
 - Soporte de temas.
 - Soporte para multihilos⁵.
 - Enfoque orientado a Objetos.
 - Internacionalización.
 - Localización.

¹ Distribución de GNU/Linux

² Es una compañía dedicada al software, específicamente en el área de sistemas operativos de redes, entre otras ramas de la tecnología. Es la empresa dueña de los derechos de la distribución SuSE Linux.

³ Es una compañía desarrolladora de software open source y dispositivos embebidos.

⁴ Es el nombre del primer sistema operativo de Apple (compañía productora de software y hardware) para los ordenadores Macintosh.

⁵ Es una tecnología que permite ejecutar múltiples hilos de ejecución a la vez.

- Accesibilidad.
- Soporte bidireccional de texto¹ (LTR/RTL, Left To Right/Right To Left).
- Soporte UTF8²
- Documentación.

■ **Interfaces.** Tiene una amplia colección de componentes básicos que incluyen:

- Ventanas (ventanas normales, de diálogos, asistentes y de información).
- Componentes visuales (etiquetas, imágenes, barras de progreso, barras de estados).
- Botones con múltiples funcionalidades.
- Textos de entrada numéricos (con o sin completamiento).
- Editor de textos multilíneas.
- Árboles, listas y vistas cuadrículadas o iconos.
- Cajas de diálogos.
- Menús (con imágenes y botones de elección).
- Barras de herramientas.
- Manejadores de Interfaces de usuarios (UI).
- Selectores (selectores de colores, archivos y fuentes).
- Iconos de estado (área de notificación en Linux)-Soportes de impresión.

Glib³

Es una biblioteca de propósito general que se usa para implementar muchas funciones no gráficas. Uno de los mayores beneficios de usar Glib es que provee una interfaz de plataforma independiente que permite que el código pueda ser usado en muchas plataformas como UNIX, Windows, OS/2⁴ y BeOS⁵. Otro aspecto de este es la amplia gama de tipo de datos que deja disponible al desarrollador:

¹ Esto significa que se puede escribir en cualquiera de los dos sentidos de izquierda a derecha como en el idioma español o de derecha a izquierda como en el idioma árabe.

² UTF8 (8-bit Unicode Transformation Format): Es una norma de transmisión de longitud variable para

³ Es una biblioteca de propósito general que se utiliza para implementar muchas funciones no gráficas.

⁴ Es un sistema operativo de IBM.

⁵ Es un sistema operativo de Be Incorporated en 1990 orientado principalmente a proveer alto rendimiento en aplicaciones multimedia.

- Incluye la gestión de estructuras de datos como listas enlazadas, arrays dinámicos, tablas de búsqueda, árboles binarios, o la gestión de Entrada/Salida asíncrona. Provee estructuras y funciones para el manejo de listas simple y doblemente enlazadas.
- Provee independencia con el sistema operativo, mejorando la portabilidad.
- Los segmentos de memoria son una forma eficiente de crear secciones de memoria que tienen todos los mismos tamaños. Pueden ser usados para crear arreglos aún de elementos de tamaño conocido.
- Manejo de plugins.
- Manejo dinámico de memoria.
- Soporte para multihilos.

Además de proporcionar varios tipos de datos, también dispone de numerosos tipos de funciones. Entre estas se encuentran funciones de manipulación de archivos, soporte de internacionalización, cadenas de caracteres, advertencias, banderas de depuración, carga dinámica de módulos, entre otras.

GDK¹ (GIMP Drawing Kit o Paquete de Dibujo de GIMP)

Es una biblioteca de gráficos que actúa como un intermediario entre gráficos de bajo nivel y gráficos de alto nivel. Las siglas GDK corresponden a GIMP Drawing Kit (Herramientas de Dibujo de GIMP), este originalmente se desarrolló para el sistema de ventanas X. Actúa entre el servidor X y la biblioteca GTK+, manejando las prestaciones básicas como primitivas de gráficos, mapas de bits, cursores, fuentes, eventos de ventanas y funcionalidades de arrastrar y soltar.

GDK es una parte importante de la portabilidad de GTK+. Como Glib provee funcionalidades independientemente de la arquitectura a las demás bibliotecas, todo lo que se necesita para hacer que GTK+ corra en cualquier plataforma es definir la capa de gráficos de un determinado sistema operativo en GDK.

¹ Es una biblioteca de gráficos que actúa como un intermediario entre gráficos de bajo nivel y gráficos de alto nivel.

Metodología de Desarrollo

SXP²

La metodología SXP hace uso de dos metodologías ágiles, que se describirán más adelante, muy utilizadas actualmente en el desarrollo del software. La misma, por lo tanto, constituye un híbrido entre SCRUM y XP incorporando de ambas las mejores prácticas para lograr el objetivo final de desarrollar un software de calidad.

Esta metodología divide el Proceso de Software en dos grandes ramas: Gestión de Proyectos e Ingeniería de Software como se muestra en la siguiente figura:

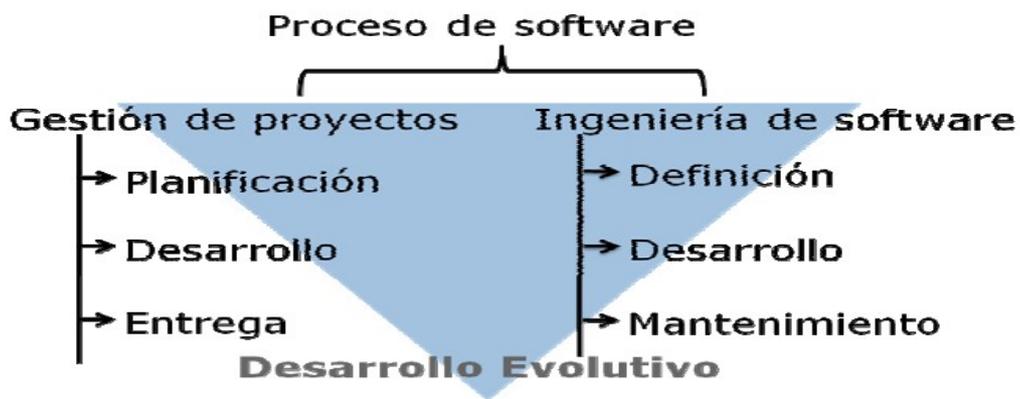


Figura 1.10 Estructura de la Metodología de Desarrollo SXP.

Para el proceso de Gestión de Proyectos se utilizan las técnicas más significativas de SCRUM y para la Ingeniería de Software se usan las características más destacadas de XP. A continuación se ofrece una breve descripción de ambas metodologías.

SCRUM

Más que una metodología es una herramienta de autogestión del equipo de desarrollo. Los propios programadores deciden qué hacer (tareas) y en que tiempo lo van a completar. Permite ver claramente el progreso de las tareas por lo que los jefes pueden percibir diariamente que se ha hecho y que falta, es decir tener una idea general del estado del proyecto acorde a las tareas a cumplir. Se basa en iteraciones o Sprints. Su principio básico es que desde un inicio es muy difícil tener una lista de funcionalidades, que no se vayan a modificar, bien definidas del sistema ya que los usuarios y el propietario de la aplicación van haciendo sucesivas aportaciones. Por esto se plantea el desarrollo de versiones ampliadas, todas ellas usables y puestas a

² Es una metodología ágil de desarrollo creada en la Universidad de las Ciencias Informáticas para proyectos de software libre.

prueba por el usuario final. Además está orientada a pequeños equipos de desarrollo y plazos de entrega muy cortos

Entre los beneficios que brinda SCRUM[SCRUM00] están:

- Incluye la gestión de estructuras de datos como listas enlazadas, arrays dinámicos, tablas de búsqueda, árboles binarios, o la gestión de Entrada/Salida asíncrona.
- Aumenta la conformidad del cliente mediante la entrega periódica de resultados tangibles además de integrarlo en el ciclo de desarrollo, permitiendo que el producto final se adecue más a sus necesidades.
- Potencia la formación de equipos autosuficientes y multidisciplinarios, reduciendo la carga de gestión y creando un ambiente propicio para que los programadores desarrollen sus potencialidades al máximo.
- Se centra en el producto y las personas y hace énfasis en la eliminación pre-activa de las trabas e impedimentos que puedan influir en el correcto desarrollo del software. Permitiendo un aumento en la productividad con mayor calidad.
- SCRUM es iterativo e incremental lo cual significa que liberamos código constantemente que representa una mejora ampliada y mejor de la versión anterior del producto.
- La medida del avance del proyecto es un producto que funciona correctamente no las horas o recursos dedicados a una tarea o grupo de tareas.
- Todo en SCRUM tiene un límite de tiempo. Al cabo de ese límite se entrega lo que esté hecho, no se alargan los plazos de entrega para cumplir con los planes. Se cambia la concepción: “recursos” + “funcionalidades deseadas” = “fecha de entrega” y lo convierte en “recursos” + “fecha de entrega” = “funcionalidades que podremos entregar”. Si hay una funcionalidad de la lista que no pudimos implementar no importa pero entregamos.
- El SCRUM Máster no es un jefe tradicional sino un siervo-líder cuya función no es distribuir trabajo ni determinar que funcionalidades se implementan y en que orden sino cerciorarse de que el proceso de SCRUM siga en marcha y eliminar los impedimentos que pueda tener el equipo de desarrollo en su camino. En SCRUM el equipo decide y el cliente evalúa.

XP: Programación Extrema

Es utilizada en proyectos a corto plazo por equipos pequeños de desarrollo. Es un proceso ligero de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software. Su utilidad se mide en valores como la simplicidad, comunicación, retroalimentación y coraje.

La metodología consiste en una programación rápida o extrema que incluye al usuario final como parte del equipo de desarrollo ya que es uno de los requisitos para llevar a buen término el proyecto. Tiene entregas frecuentes y una refactorización continua lo que nos permite mejorar el diseño cada vez que se le haga una nueva funcionalidad. Sustenta la propiedad colectiva, la programación en parejas y pequeños encuentros diarios donde cada uno de los integrantes del equipo cuenta que ha hecho, que problemas tiene y que hará más adelante. **[MA00]**

Lo que propone XP es:

- Empieza en pequeño y añade funcionalidades con retroalimentación continua.
- La gestión de los cambios se convierte en pieza clave del proceso de desarrollo.
- El costo del cambio no depende de la fase o la etapa.
- No se programan funcionalidades que no sean necesarias antes de tiempo.
- El cliente o el usuario forma parte del equipo.

Esta metodología ágil centra su éxito en fomentar las buenas relaciones interpersonales en el equipo de desarrollo logrando un mayor trabajo en conjunto y preocupándose por la superación continua de los integrantes del proyecto. Es clave la participación del cliente como parte del proceso de desarrollo, la simplicidad de las soluciones implementadas así como el coraje para enfrentar los cambios.

1.8 Conclusiones

Como resultado de la investigación realizada en este capítulo los autores determinaron que es necesaria la creación de una solución que incluya una herramienta que permita configurar los protocolos requeridos y por ende los servidores de diskless en sistemas operativos GNU/Linux, además de que sea fácil de usar y lo más ligera posible; así como una imagen del sistema operativo Nova optimizada para su uso en diskless y un kernel parcheado con AUFS.

Para lograr la correcta implementación del software los autores acordaron utilizar tecnologías libres y ágiles determinadas en su mayoría por políticas del proyecto Nova

para el cual se va a desarrollar la aplicación: Python como lenguaje de programación; Eclipse y el Geany se utilizarán como IDEs de programación así como Glade para el diseño de la interfaz gráfica; SXP como metodología de desarrollo de software, una metodología desarrollada en la Universidad de las Ciencias Informáticas orientada al desarrollo de software libre. Y como herramienta para realizar los diferentes artefactos de ingeniería del software se seleccionó BoUML con el potente y versátil lenguaje de modelado UML.

2 Planificación y Definición de la aplicación.

El sabio no dice todo lo que piensa, pero siempre piensa todo lo que dice.

Aristóteles

Resumen. En este capítulo se describe el entorno de trabajo sobre el que se desarrolla la aplicación. Además, se hace la propuesta del sistema y se determinan los requerimientos funcionales y no funcionales del mismo. Se lleva a cabo el modelado del negocio y se definen las historias de usuarios, los actores y trabajadores que intervienen en cada una de ellas.

2.1 Información que se maneja

La información con la cual se trabaja son los archivos de configuración de un servidor GNU/Linux (con sistema operativo Nova) como son el *dhcp.conf*, el *pxe.cfg*, el *exports* y el *in.tftp*. Además se manipularán datos referente a las imágenes que se van a servir como son los permisos sobre la misma entre otros.

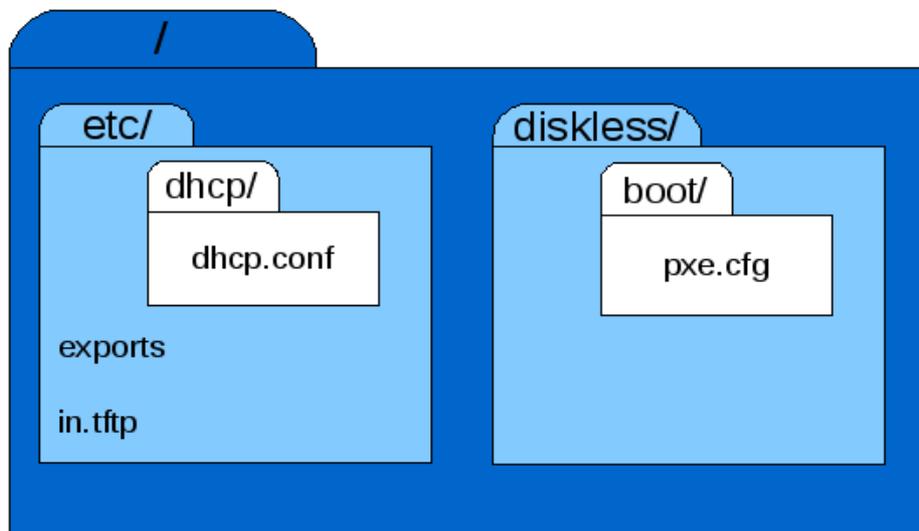


Figura 2.1 Ubicación de los archivos de configuración en el sistema de archivos del servidor con sistema operativo Nova

2.2 Propuesta del sistema.

Proceso de Configuración de un Servidor de Diskless

Actualmente la configuración de un servidor de diskless con sistemas operativos GNU/Linux es muy compleja debido al gran conocimiento que debe poseer un administrador en redes de telecomunicaciones y servidores. Esta labor es, para un administrador de redes de computadoras en Cuba, algo más que conocimiento básico y si una tarea bastante complicada.

Imagen y kernel

En primer lugar el administrador deberá saber parchear un kernel con AUFS (**Ver Anexo 6 - Parchear kernel con AUFS**) y crear una imagen (**Ver Anexo 8 - Crear una imagen**) apta para diskless de forma que estos puedan interactuar con ella perfectamente. Téngase en mente que esta imagen debe ser lo más óptima posible, debido a que el diskless no es una computadora tradicional que pueda realizar gran cantidad de procesamiento, por lo que las aplicaciones que se incluyan deberán estar previamente seleccionadas y configuradas para lograr que no sobrecarguen el sistema y que el flujo de trabajo sea aceptablemente rápido. Ambos procesos son altamente complicados para alguien que no está acostumbrado a trabajar con servidores GNU/Linux, que es el caso de la mayoría de los administradores de red de los centros de trabajo de nuestro país.

Por esta razón uno de los resultados tangibles de nuestra investigación es un kernel parcheado con AUFS así como una imagen del sistema operativo Nova optimizada para el trabajo en diskless.

Archivos de configuración

En segundo lugar el administrador debe saber manipular archivos de configuración de servidores en GNU/Linux, labor que en el momento de realizada esta investigación es prácticamente un tema tabú para los administradores de red, que están acostumbrados a las facilidades que brinda el sistema operativo Microsoft Windows a la hora de administrar servidores de redes.

Un archivo de configuración como se puede ver en la siguiente figura no es más que texto plano que caracteriza ciertas opciones que pueden estar habilitadas o no en un servidor.

```

get-lease-hostnames true;
allow booting;
allow bootp;
option broadcast-address 10.33.4.255 ;

# Declaracion del space pxenova
option space pxenova;
option pxenova.magic code 208 = string;
option pxenova.configfile code 209 = text;
option pxenova.pathprefix code 210 = text;
option pxenova.reboottime code 211 = unsigned integer 32;

include "/var/lib/dhcp/macperhosts";

subnet 10.33.4.0 netmask 255.255.255.0
{
  pool {
    option domain-name "domain.example.org";
    option domain-name-servers 10.0.0.3, 10.0.0.4;
    option routers 10.33.4.254;
    default-lease-time 604800;
    max-lease-time 604800;
    site-option-space "pxenova";
    if exists dhcp-parameter-request-list {
      option dhcp-parameter-request-list = concat(option dhcp-parameter-request-list,d0,d1,d2,d3);
    }
    option pxenova.configfile "pxe.cfg";

    #option pxenova.pathprefix "/tftpboot";
    option pxenova.reboottime 5;

    #especifica el servidor de tftp
    next-server 10.33.4.156;
    filename "/pxelinux.0";
    range 10.33.4.9 10.33.4.19;
    allow unknown-clients;
  }
}

```

Figura 2.2 Archivo de configuración *dhcpcd.conf*

Para un administrador experto en este tipo de trabajo con servidores GNU/Linux configurar un archivo como este no debe ser de gran dificultad pero para uno que está acostumbrado a trabajar con el entorno gráfico de Windows Server 2003 como es el caso de la mayoría de los administradores en Cuba se vuelve un galimatías inteligible en en cual no saben por donde comenzar.

Básicamente un archivo de configuración está dividido en secciones en las cuales se encuentran un grupo de variables a las cuales se le asignan valores que son los que hacen que el sistema intérprete las opciones o características que queremos activar y como.

Como consecuencia de lo anteriormente expresado otro resultado tangible de la presente investigación son los archivos de configuración *dhcpcd.conf*, *pxe.cfg*, *exports* y *in.tftpd* correctamente configurados para el correcto funcionamiento de un servidor de diskless en una distribución Nova. **(Ver Anexo 7 - Archivos de Configuración)**

Lira

En este trabajo se propone una herramienta que permite configurar los protocolos necesarios del lado del servidor para que este logre establecer una buena comunicación con las terminales diskless. Además es capaz de preguntar al usuario donde se encuentra la imagen que se desea servir. También es capaz de implementar ambas mo-

dalidades explicadas en el capítulo anterior para el funcionamiento de los diskless (procesamiento del lado del servidor y del cliente) lo cual la hace única en su clase ya que las estudiadas en esta investigación solo implementan una u otra.

Esta aplicación evita que el usuario tenga que configurar cada uno de los protocolos manualmente, lo que redundaría en ahorro de tiempo para el administrador del servidor.

Finalmente, Lira reiniciará cada uno de los protocolos antes mencionados a través de sus demonios. Permitiendo una buena comunicación entre las terminales y el servidor.

Para determinar las características funcionales de Lira y basados en el estudio de aplicaciones semejantes, hecho en el capítulo anterior; los autores elaboraron la siguiente tabla¹:

¹ En la tabla se utilizan las abreviaturas PLS y PLC para hacer referencia a Procesamiento del Lado del Servidor y Procesamiento del Lado del Cliente Respectivamente.

| Nombre | Modalidad que implementa | | Necesario Instalar | | Licencia | | |
|-------------------------|--------------------------|-----------|--|----|---|-------------------|--|
| | Protocolos que configura | | Sistema Operativo de la Imagen | | Hardware requerido | Otros | |
| LanCore | XDMCP, VNC y RDP | PLS | Windows: XP, Server 2000 y 2003 Linux: Debian, Red Hat y Suse | SI | 64 MB RAM 100 MB HDD | - | Monitoreo de terminales |
| Open Thin Client | DHCP, TFTP y NFS | PLS y PLC | Ubuntu optimizado para el uso en diskless | SI | 500-1GB RAM 1-2GB HDD | GPL | Programado en Java |
| LTSP | Todos. | PLS | Suse, Fedora, Debian, Ubuntu y otros | NO | Pentium 4 a 3.0 GHz y 512 MB RAM | GPL | Solo para sistemas GNU/Linux |
| PXES | XDMCP, VNC y RDP | PLS | Cualquiera | SI | Pentium 4 a 2.0 Ghz y 412 MB RAM | GPL | Ligera, necesita 32 MB RAM y un procesador x86 |
| TCOS | DHCP, TFTP y NFS | PLS | Debian y Ubuntu | SI | Pentium 4 o AMD y 300 MB RAM y 80 MB extras por cada diskless conectado | GPL | Herramienta gentcos para crear imagen a compartir |
| Win Connect | DHCP y RDP. | PLS | Windows XP o Small Server 2003 | SI | 256-384 MB RAM 1-2GB RAM | WCSL ¹ | Soporta impresión local y sonido |
| Lira | Todos. | PLS | Nova (por el momento). | NO | 512 MB RAM 1-2GB HDD | GPL | Multiplataforma y arquitectura para diferentes distribuciones de Linux |

Tabla 2.1 Comparación entre diferentes aplicaciones y Lira²

¹ WinConnect Server XP License.

² Los datos incluidos en la tabla son extraídos de los sitios oficiales de cada una de las aplicaciones referenciados en el capítulo anterior.

2.3 Arquitectura de la Herramienta Lira

Para la implementación del software se escogió una arquitectura por capas. Las arquitecturas en capas constituyen uno de los estilos de programación que aparecen con mayor frecuencia mencionados en las literaturas informáticas. El estilo de programación en capas constituye una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior, y se sirve de las prestaciones que le brinda la inmediata inferior. En la práctica, las capas suelen ser entidades complejas, compuestas por varios paquetes o subsistemas. **[AI00] [AI01]**

En la capa de *presentación* se encuentra el componente GTK 2.14.7, que constituye el framework¹ gráfico que se utiliza para implementar la interfaz visual de la aplicación, mientras que en la capa *lógica de negocio* se encuentran las clases implementadas para la creación de objetos y la manipulación de los mismos, que constituirán el núcleo de la aplicación.

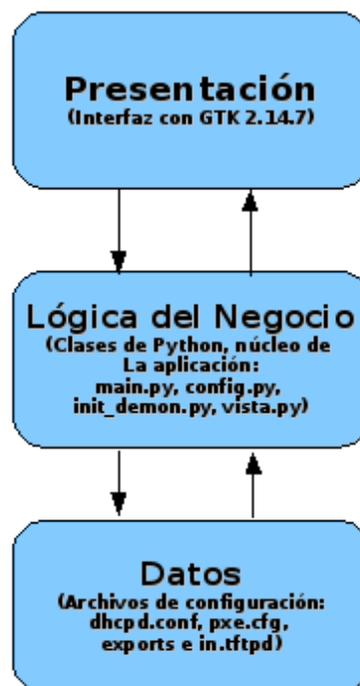


Figura 2.3 Arquitectura tres capas de Lira.

2.4 Fase Planificación-Definición del Sistema de Lira

La Fase *Planificación-Definición* es la primera de las cuatro fases que describe la metodología SXP. En esta fase es donde se generan todos los documentos que se relacionan con la concepción inicial del sistema. También se incluyen algunos que

¹ Es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

forman parte de la primera etapa de la Ingeniería de Software, como son los relacionados con el *Negocio*, *Requisitos del Sistema* y *Diseño*.

A continuación se procede a explicar todo el diseño de Lira en forma de historias de usuarios, prototipos de interfaz de usuario y algunos modelos auxiliares.

Plantilla de Concepción del Sistema

Este documento se escribe luego de sostener una entrevista con el cliente. En el documento aparece una visión general del producto a implementar así como los roles que intervendrán en el desarrollo y sus respectivas responsabilidades en este proceso.

Aquí se podrá encontrar además el tipo de proyecto a que pertenece así como la especificación del Polo Productivo y su clasificación. Se recogen también en esta plantilla las herramientas que se utilizarán para desarrollar la aplicación, el alcance que va a tener, una descripción de los involucrados en el negocio, cuales son los motivos por los cuales se lleva a cabo el desarrollo del software y la propuesta de solución.

Esta plantilla es de gran importancia en la documentación del proyecto puesto que es la base para generar los demás artefactos durante el ciclo de desarrollo del software.

“Alcance de Lira. Se pretende poner en práctica la aplicación en todas las instituciones u organismos que necesiten clientes ligeros.

Herramientas utilizadas: Para la realización de este sistema se utilizarán diferentes herramientas como: BoUML, para los diagramas de la metodología; Python, como lenguaje de programación y las librerías PyQt y QT para implementar algunas funcionalidades con el lenguaje antes mencionado.

Solución propuesta: Se propone desarrollar una herramienta de configuración para servidores de diskless con una implementación sencilla y ligera que pueda ser ejecutada en los más diversos ámbitos.

(Ver Anexo 2 - Plantilla de Concepción del Sistema)

Modelo Historias de Usuarios del Negocio

Este artefacto se genera del juego de la planificación, una vez que está bien definida la concepción del sistema es más fácil comprender el negocio.

Esta plantilla es una forma de interactuar con los clientes y viceversa. Una vez que se modela el negocio con metodologías ágiles se definen historias de usuarios en vez de casos de uso.

Para comprender mejor el negocio hay que definir además los actores y trabajadores del mismo.

| Actor | Descripción |
|------------------------------|---|
| Administrador de Terminales. | Es la entidad que se beneficia del sistema. Es a quien esta dirigida la aplicación. |

Tabla 2.2 Actores del Negocio

| Trabajador | Descripción |
|------------------------------|--|
| Administrador de Terminales. | Es el entidad encargada de configurar cada una de los protocolos para lograr una buena conexión cliente-servidor |

Tabla 2.3 Trabajadores del Negocio

Una vez que se tienen definidos estos importantes roles para el negocio, se representa el *Diagrama de Historias de Usuarios del Negocio* correspondiente, ya sea el conocido Diagrama de Negocio o el Diagrama de Domino.

Diagrama de Historias de Usuarios del Negocio

El objetivo del modelado del negocio es comprender y describir las clases más importantes dentro del contexto del sistema. Este detalla las principales entidades presentes y sus relaciones; se puede decir que provee una primera vista estructural del sistema.

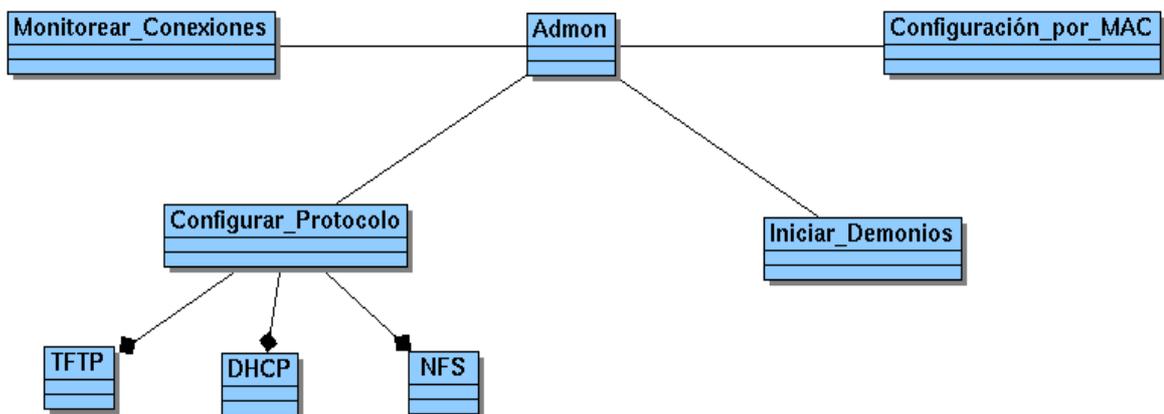


Figura 2.4 Diagrama del Negocio

Captura de requisitos

La plantilla de *Lista de Reserva del Producto*, es el primer artefacto generado en la etapa de captura de requisitos y está conformada por una lista priorizada que define el trabajo que se va a realizar en el proyecto.

Esta plantilla permite la organización de los requisitos, tanto funcionales como no funcionales, en dependencia de la prioridad que tengan para el desarrollo del sistema.

| Requisitos Funcionales | | | | |
|------------------------|------|---|-------------------------------|-------------------|
| Asignado a | Ítem | Descripción | Estimación | Estimado por |
| Prioridad | | Muy Alta | | |
| Viviana | 1 | Configuración gráfica de los protocolos: <ul style="list-style-type: none"> •DHCP •TFTP •NFS | 1 semana para cada protocolo. | Alberto y Viviana |
| Prioridad | | Alta | | |
| Alberto | 2 | Integración con el sistema operativo Nova-Gentoo y que permita levantar los servicios correspondientes. | 1 semana | Alberto y Viviana |
| Viviana | 3 | Iniciar los demonios correspondientes a los protocolos que se van a configurar. | 0.5 semana | Alberto y Viviana |

Tabla 2.4 Requisitos Funcionales solo con Prioridad Muy Alta y Alta

| Requisitos No Funcionales | | | | |
|----------------------------------|-------------|---|-------------------|---------------------|
| Asignado a | Ítem | Descripción | Estimación | Estimado por |
| Prioridad | | Muy Alta | | |
| Alberto | 1 | El software debe contar con una interfaz gráfica de administración. | 1 semana | Alberto y Viviana |
| | 2 | El software debe ser fácil de manejar y de comprender. | | Alberto y Viviana |
| | 3 | El servidor donde se alojara la aplicación debe tener como mínimo: <ul style="list-style-type: none"> •Procesador: Pentium 4 a 2.0 GHZ o superior. •Memoria: 1GB de RAM o superior •Tarjeta de Red: Ethernet LAN 10/100 Mbps •Capacidad de HDD: 1-2 GB mínimo | | |
| Prioridad | | Alta | | |
| Viviana | 4 | Identificar el usuario, según la MAC y dale solo los permisos que le corresponden. | 0.5 semana | Alberto y Viviana |
| Viviana | 5 | Proteger la información de accesos no autorizados. | 0.5 semana | Alberto y Viviana |
| Alberto | 6 | Debe contar con un sistema de salvallas externas en caso de desastre. | 0.1 semana | Alberto y Viviana |

Tabla 2.5 – Requisitos No Funcionales solo con Prioridad Muy Alta y Alta.

(Ver Anexo 3 – Lista de Reserva del Producto)

Definir las Historias de Usuarios.

Las *historias de usuario* son la técnica utilizada en XP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el *Proceso Unificado de Rational* (más conocido por sus siglas en inglés RUP). Las mismas son escritas por los analistas como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el analista y los desarrolladores para definir las.

Plantilla de Historias de Usuarios.

Esta plantilla es fácil de comprender dado que esta escrita en lenguaje del cliente, en ella se especifica cada uno de los requisitos del sistema, sin necesidad de documentaciones externas. Con esta se reflejan todas las características del sistema. Sirve de guía para el proceso de implementación del sistema.

| Historia de Usuario | |
|---|--|
| Numero: HU-1 | Nombre de Historia de Usuario: Configurar Protocolo TFTP. |
| Modificación de Historia de Usuario Numero: 1.0 | |
| Usuario: Alberto y Viviana | Iteración Asignada: 1 |
| Prioridad en el Negocio: Muy Alta | Puntos Estimados: 1 semana |
| Riesgo en Desarrollo: | Puntos Reales: 1 semana |
| Descripción: Esta historia permite que el servidor tenga el servicio TFTP trabajando correctamente para lograr una transferencia de datos exitosa. | |
| Observaciones: | |
| Prototipo de interface: | |
|  | |

Tabla 2.6 Historia de Usuario:Configurar Protocolo TFTP

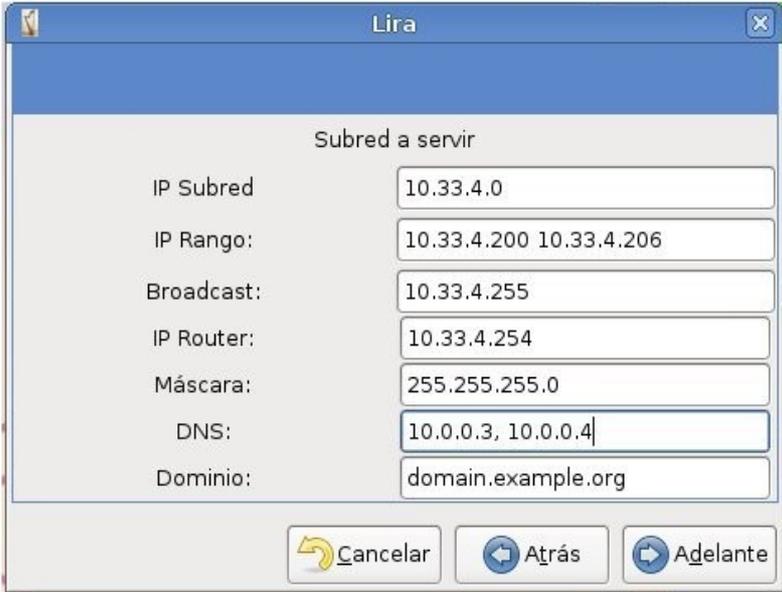
| Historia de Usuario | |
|---|---|
| Numero: HU-2 | Nombre de Historia de Usuario: Configurar Protocolo DHCP |
| Modificación de Historia de Usuario Numero: 1.0 | |
| Usuario: Viviana y Alberto | Iteración Asignada: 2 |
| Prioridad en el Negocio: Muy Alta | Puntos Estimados: 1 semana |
| Riesgo en Desarrollo: | Puntos Reales: 1 semana |
| Descripción: Esta historia de usuario es la que garantiza que el protocolo DHCP este funcionando correctamente para la petición de las maquinas cliente. | |
| Observaciones: Dentro de la configuración se incluye servir a las máquinas que no carecen de disco duro. Además sirve de guía para los diskless. | |
| Prototipo de interface: | |
|  <p>The screenshot shows a window titled "Lira" with a subtitle "Subred a servir". It contains several input fields: "IP Subred" (10.33.4.0), "IP Rango:" (10.33.4.200 10.33.4.206), "Broadcast:" (10.33.4.255), "IP Router:" (10.33.4.254), "Máscara:" (255.255.255.0), "DNS:" (10.0.0.3, 10.0.0.4), and "Dominio:" (domain.example.org). At the bottom, there are three buttons: "Cancelar", "Atrás", and "Adelante".</p> | |

Tabla 2.7 Historia de Usuario: Configurar protocolo DHCP

| Historia de Usuario | |
|---|--|
| Numero: HU-3 | Nombre de Historia de Usuario: Configurar Protocolo NFS |
| Modificación de Historia de Usuario Numero: 1.0 | |
| Usuario: Viviana y Alberto | Iteración Asignada: 3 |
| Prioridad en el Negocio: Muy Alta | Puntos Estimados: 1 semana |
| Riesgo en Desarrollo: | Puntos Reales: 1 semana |
| Descripción: Esta historia de usuario es la que garantiza que el cliente sepa las características de la imagen que cargar. | |
| Observaciones: | |
| Prototipo de interface: | |
|  | |

Tabla 2.8 Historia de Usuario: Configura protocolo NTFS

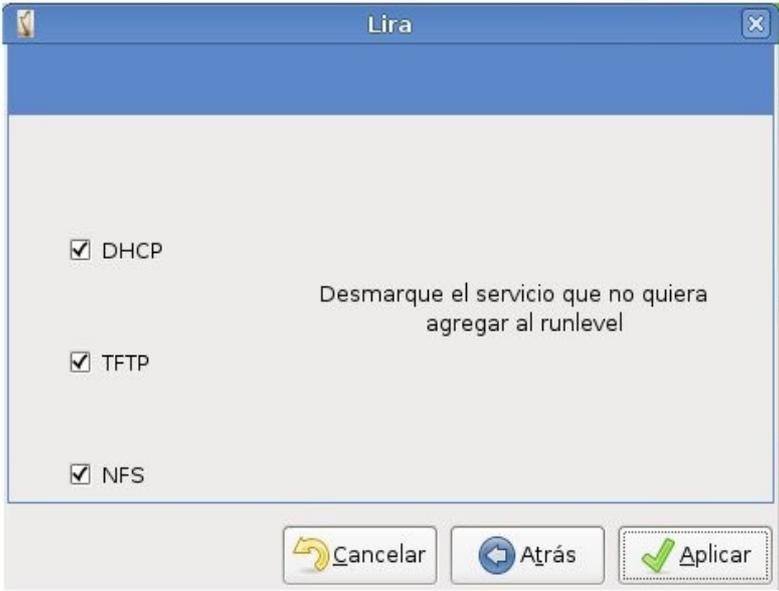
| Historia de Usuario | |
|--|--|
| Numero: HU-4 | Nombre de Historia de Usuario: Iniciar Demonios |
| Modificación de Historia de Usuario Numero: 1.0 | |
| Usuario: Viviana y Alberto | Iteración Asignada: 3 |
| Prioridad en el Negocio: Alta | Puntos Estimados: 0.6 semana. |
| Riesgo en Desarrollo: | Puntos Reales: 0.6 semana |
| Descripción: Esta historia de usuario garantiza que una vez que el administrador haga un cambio se inicien nuevamente los demonios para que cargue estos cambios. Además permite añadir los procesos que aparecerán en el runlevel default ¹ . | |
| Observaciones: | |
| Prototipo de interface: | |
|  | |

Tabla 2.9 Historia de Usuario: Iniciar Demonios

¹ En español nivel de ejecución se refiere al estado en que está corriendo un sistema operativo.

| Historia de Usuario | |
|--|---|
| Numero: HU-5 | Nombre de Historia de Usuario: Configuración por MAC |
| Modificación de Historia de Usuario Numero: 1.0 | |
| Usuario: Viviana y Alberto | Iteración Asignada: 4 |
| Prioridad en el Negocio: Media | Puntos Estimados: 2 semanas |
| Riesgo en Desarrollo: | Puntos Reales: 2 semanas. |
| Descripción: Esta historia de usuario permitirá al administrador especificar los servicios acorde a la MAC del cliente. | |
| Observaciones: | |
| Prototipo de interface: | |
|  | |

Tabla 2.10 Historia de Usuario: Configuración por MAC

| Historia de Usuario | |
|--|---|
| Numero: HU-6 | Nombre de Historia de Usuario: Monitorear Conexiones |
| Modificación de Historia de Usuario Numero: 1.0 | |
| Usuario: Viviana y Alberto | Iteración Asignada: 5 |
| Prioridad en el Negocio: Media | Puntos Estimados: 2 semanas |
| Riesgo en Desarrollo: | Puntos Reales: 2 semanas. |
| Descripción: Esta historia de usuario permitirá al administrador monitorear las conexiones hechas al servidor por cada diskless que tenga en su subred. | |
| Observaciones: | |
| Prototipo de interface: | |
|  | |

Tabla 2.11 Historia de Usuario: Monitorear Conexiones

Planificación de las Historias de Usuarios.

En esta sección se planifica acertadamente el proyecto según su prioridad, riesgos y esfuerzo. Esta planificación es buscando condiciones ideales para escribir el código sin distracciones. **(Ver Anexo 4-Planificación de Historias de Usuario)**

2.5 Diseño con las Metáforas.

Metáfora del Sistema.

El Sistema Lira es una herramienta que permitirá la configuración de los protocolos necesarios para lograr el correcto funcionamiento de un *servidor de diskless*.

Plantilla de Modelo de Diseño

Esta plantilla permite confeccionar un diseño inicial y sencillo del sistema y es la base para la definición de una futura arquitectura. Por lo que en ella se realiza un diagrama de componentes que describe 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. **(Ver Anexo 5 - Diagrama de componentes)**

2.6 Conclusiones

Se realizó la planificación del proyecto llegando a un acuerdo entre el cliente y el grupo de desarrollo y se definieron las *historias de usuario*¹. Se hizo un *levantamiento de requisitos* con el fin de determinar las necesidades del cliente. Se realizó el *diseño de metáforas*, donde se mostró el *diagrama de componentes* del sistema el cual permitió dar una mejor visión del proyecto.

¹ En este trabajo se implementan las cuatro primeras historias de usuarios, el resto se concibieron para versiones posteriores.

3 Implementación y Pruebas

No hay cosas imposibles sino hombre incapaces

José Martí

Resumen. En el presente capítulo se definen las *tareas de ingeniería* a desarrollar por los programadores para lograr dar cumplimiento a cada historia de usuario respectivamente. Además se elabora el *plan de releases* donde queda plasmado el tiempo en el cual se pretende liberar cada versión del producto. Se menciona el estándar de codificación utilizado por el equipo para programar la aplicación. Por último se documentan las pruebas de aceptación realizadas al software.

3.1 Fase “Desarrollo” de Lira

Desarrollo, es la segunda fase definida por la metodología SXP. En la primera parte de esta fase se generan todos los documentos relacionados con la planificación de las iteraciones, y además se recogen las principales definiciones que se manejan en la metodología y otros términos de difícil entendimiento para los clientes, así como las tareas a realizar durante la implementación. Además se genera el código fuente en la etapa de *implementación* y los documentos relacionados con las pruebas.

3.2 Tareas para lograr la implementación de Lira

Plantilla de Tareas de Ingeniería

La plantilla de *Tareas de Ingeniería* posee una gran importancia, pues permite definir cada una de las actividades que estarán asociadas a las historias de usuario y que permitirán su implementación. También posibilita conocer que programador está asignado a cada tarea, así como el tiempo que se necesita para su realización, lo que facilita la estimación del tiempo que llevará cada historia de usuario en implementarse, de acuerdo a su complejidad¹.

¹ No se incluyen las tareas de ingeniería de las iteraciones cuatro y cinco porque no están contempladas en la versión 0.2 de Lira que es la que abarca esta investigación.

Iteración 1

| Tarea de Ingeniería | |
|--|---|
| Numero Tarea: TI-1 | Nombre de la Historia de Usuario: Configurar Protocolo TFTP. |
| Nombre de la Tarea: Estudiar la configuración del protocolo TFTP. | |
| Tipo de Tarea: Estudio | Puntos Estimados: 0.2 semana |
| Fecha de Inicio: 30/3/2009 | Fecha Fin: 1/4/2009 |
| Programador Responsable: Viviana y Alberto | |
| Descripción: Se estudia la configuración del protocolo TFTP. | |

Tabla 3.1 Tarea de Ingeniería: Estudiar la configuración del protocolo TFTP.

| Tarea de Ingeniería | |
|--|--|
| Numero Tarea: TI-2 | Nombre de la Historia de Usuario: Configurar Protocolo TFTP |
| Nombre de la Tarea: Instalación y configuración del protocolo TFTP manualmente. | |
| Tipo de Tarea: Desarrollo. | Puntos Estimados: 0.2 semana |
| Fecha de Inicio: 1/4/2009 | Fecha Fin: 3/4/2009 |
| Programador Responsable: Viviana | |
| Descripción: Se configura manualmente el fichero para determinar los campos que deberán estar presentes en la aplicación final. | |

Tabla 3.2 Tarea de Ingeniería: Instalación y configuración del protocolo TFTP manualmente.

| Tarea de Ingeniería | |
|--|--|
| Numero Tarea: TI-3 | Nombre de la Historia de Usuario: Configurar Protocolo TFTP |
| Nombre de la Tarea: Implementación del método que permitirá configurar el protocolo TFTP y de la interfaz gráfica del mismo. | |
| Tipo de Tarea: Desarrollo. | Puntos Estimados: 0.6 semanas |
| Fecha de Inicio: 6/4/2009 | Fecha Fin: 9/4/2009 |
| Programador Responsable: Viviana | |
| Descripción: Se implementa el método en la clase correspondiente que permitirá a la aplicación final configurar el protocolo DHCP correctamente así como la interfaz gráfica para introducir los datos requeridos al administrador para lograr configurar el protocolo adecuadamente. | |

Tabla 3.3 Tarea de Ingeniería: Implementación del método que permitirá configurar el protocolo TFTP y de la interfaz gráfica del mismo.

Iteración 2

| Tarea de Ingeniería | |
|--|---|
| Numero Tarea: TI-4 | Nombre de la Historia de Usuario: Configurar Protocolo DHCP. |
| Nombre de la Tarea: Estudiar la configuración del protocolo DHCP. | |
| Tipo de Tarea: Estudio | Puntos Estimados: 0.2 semana |
| Fecha de Inicio: 13/4/2009 | Fecha Fin: 15/4/2009 |
| Programador Responsable: Viviana y Alberto | |
| Descripción: Se estudia la configuración del protocolo DHCP. | |

Tabla 3.4 Tarea de Ingeniería: Estudiar la configuración del protocolo DHCP.

| Tarea de Ingeniería | |
|--|---|
| Numero Tarea: TI-5 | Nombre de la Historia de Usuario: Configurar Protocolo DHCP. |
| Nombre de la Tarea: Instalación y configuración del protocolo DHCP manualmente. | |
| Tipo de Tarea: Estudio | Puntos Estimados: 0.2 semana |
| Fecha de Inicio: 15/4/2009 | Fecha Fin: 17/4/2009 |
| Programador Responsable: Viviana y Alberto | |
| Descripción: Se configura manualmente el fichero para determinar los campos que deberán estar presentes en la aplicación final. | |

Tabla 3.5 Tarea de Ingeniería: Instalación y configuración del protocolo DHCP manualmente.

| Tarea de Ingeniería | |
|--|--|
| Numero Tarea: TI-6 | Nombre de la Historia de Usuario: Configurar Protocolo DHCP |
| Nombre de la Tarea: Implementación del método que permitirá configurar el protocolo DHCP y de la interfaz gráfica del mismo. | |
| Tipo de Tarea: Estudio | Puntos Estimados: 0.6 semana |
| Fecha de Inicio: 20/4/2009 | Fecha Fin: 23/4/2009 |
| Programador Responsable: Viviana y Alberto | |
| Descripción: Se implementa el método en la clase correspondiente que permitirá a la aplicación final configurar el protocolo DHCP correctamente así como la interfaz gráfica para introducir los datos requeridos al administrador para lograr configurar el protocolo adecuadamente. | |

Tabla 3.6 Tarea de Ingeniería: Implementación del método que permitirá configurar el protocolo DHCP y de la interfaz gráfica del mismo.

Iteración 3

| Tarea de Ingeniería | |
|--|--|
| Numero Tarea: TI-7 | Nombre de la Historia de Usuario: Configurar Protocolo NFS. |
| Nombre de la Tarea: Estudiar la configuración del protocolo NFS. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.2 semana |
| Fecha de Inicio: 27/4/2009 | Fecha Fin: 29/4/2009 |
| Programador Responsable: Viviana | |
| Descripción: Se configura manualmente el fichero para determinar los campos que deberán estar presentes en la aplicación final. | |

Tabla 3.7 Tarea de Ingeniería: Estudiar la configuración del protocolo NFS.

| Tarea de Ingeniería | |
|--|---|
| Numero Tarea: TI-8 | Nombre de la Historia de Usuario: Configurar Protocolo NFS |
| Nombre de la Tarea: Instalación y configuración del protocolo NFS manualmente. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.2 semana |
| Fecha de Inicio: 29/4/2009 | Fecha Fin: 1/5/2009 |
| Programador Responsable: Viviana | |
| Descripción: Se configura manualmente el fichero para determinar los campos que deberán estar presentes en la aplicación final. | |

Tabla 3.8 Tarea de Ingeniería: Instalación y configuración del protocolo NTFS manualmente.

| Tarea de Ingeniería | |
|--|--|
| Numero Tarea: TI-9 | Nombre de la Historia de Usuario: Configurar Protocolo NFS. |
| Nombre de la Tarea: Implementación del método que permitirá configurar el protocolo NFS y de la interfaz gráfica del mismo. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.6 semana |
| Fecha de Inicio: 4/5/2009 | Fecha Fin: 7/5/2009 |
| Programador Responsable: Alberto. | |
| Descripción: Se implementa el método en la clase correspondiente que permitirá a la aplicación final configurar el protocolo DHCP correctamente así como la interfaz gráfica para introducir los datos requeridos al administrador para lograr configurar el protocolo adecuadamente. | |

Tabla 3.9 Tarea de Ingeniería: Implementación del método que permitirá configurar el protocolo NFS y de la interfaz gráfico del mismo.

| Tarea de Ingeniería | |
|---|---|
| Numero Tarea: TI-10 | Nombre de la Historia de Usuario: Iniciar Demonios |
| Nombre de la Tarea: Implementación del método que permitirá reiniciar los demonios. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.6 semana |
| Fecha de Inicio: 11/5/2009 | Fecha Fin: 14/5/2009 |
| Programador Responsable: Alberto. | |
| Descripción: Se implementa el método en la clase correspondiente que permitirá a la aplicación final reiniciar los demonios. Se implementa el método que permitirá incluir en el run level default los protocolos deseados así como la interfaz gráfica del mismo. | |

Tabla 3.9 Tarea de Ingeniería: Implementación del método que permitirá reiniciar los demonios.

Plantilla de Plan de Releases

En esta plantilla se define el *Plan de Releases* 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 release se tiene en cuenta la prioridad que definió el cliente para dicha historia. Como resultado de la priorización de las historias se llegó a la siguiente planificación:

- **Release 1.** Se propone codificar las historias de usuario más importantes, que proveen las funcionalidades más críticas del sistema:
 - Configurar protocolo TFTP.
 - Configurar protocolo DHCP.
 - Configurar protocolo NFS.
- **Release 2.** Se codificará una historia de prioridad alta:
 - Iniciar Demonios.
- **Release 3.** Entrenamiento:
 - Probar la funcionalidad con servidores reales.
 - Capacitar personal que utilizará la aplicación.
- **Release 4.** Se codificarán dos historias de usuario de prioridad media y se diseñará una nueva interfaz para Lira.

- ☐ Monitorear Conexiones.
- ☐ Configuración por MAC.
- ☐ Diseño de nueva interfaz de Lira.

Se destaca que todas las historias de usuario definidas llevan implementación visual.

| Releases | Historias de Usuarios | Tiempo Estimado (semanas) |
|----------|-----------------------|---------------------------|
| 1 | 1,2,3 | 3 |
| 2 | 4 | 0,6 |
| 3 | Entrenamiento | 2 |
| 4 | 5,6 | 2 |

Tabla 3.10 Plan de Releases

3.3 Implementación de Lira.

Estándares de Codificación

Para el desarrollo de Lira se utilizó el estándar de codificación **PEP8**, que es el estándar de codificación, del lenguaje de programación Python, utilizado en el proyecto Nova y definido a nivel mundial como guía para los programadores en este lenguaje. Las pautas proporcionadas en éste mejoran la legibilidad y el entendimiento del código.

Ejemplo de algunas pautas de este estándar son:

- El nombre de los métodos está separado por guión bajo y mayúsculas en la primera letra de cada palabra.

```
def Guardar_P1(self):
```

- El nombre de las variables comienza con mayúsculas.

```
global Isubred
```

- Las definiciones de métodos dentro de una misma clase se separan con una línea en blanco.

Este estándar controla gran cantidad de aspectos referentes al estilo de programación para el lenguaje de programación Python **[EP00]**.

3.4 Fase “Pruebas” de Lira

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

Casos de prueba de Aceptación para Lira.

Caso de Prueba Historia de Usuario: HI-1.

Esta sección cubre un conjunto de pruebas funcionales relacionadas con la historia de usuario: **Configurar protocolo TFTP.**

| Caso de Prueba de Aceptación | |
|--|---|
| Código Caso de Prueba: 1 | Nombre Historia de Usuario: Configurar el protocolo TFTP |
| Nombre de la persona que realiza la prueba: Mijail Hurtado Fedorovich. | |
| Descripción de la Prueba: Se ejecuta el método conf_pxe de Lira, y se le pasa por parámetro el valor que necesita y se verifica que escribe el parámetro sin problemas. | |
| Condiciones de Ejecución: conf_pxe.py debe estar en correcto estado y debe ser ejecutado con permisos de root con el fin de poder guardar los cambios que se realicen al fichero situado en /diskless/boot/tftpboot con el nombre de pxe.cfg. | |
| Entrada / Pasos de ejecución: Ejecutar Lira y una vez que solicita la Dirección IP y la subred a servir, el administrador deberá proveerlos para configurar el protocolo en cuestión. | |
| Resultado Esperado: Los valores insertados deberán aparecer en el archivo de configuración donde les corresponda. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 3.11 Caso de Prueba realizado a la Historia de Usuario: Configurar protocolo TFTP.

Caso de Prueba Historia de Usuario: HI-2

Esta sección cubre un conjunto de pruebas funcionales relacionadas con la historia de usuario: **Configurar protocolo DHCP.**

| Caso de Prueba de Aceptación | |
|---|---|
| Código Caso de Prueba: 1 | Nombre Historia de Usuario: Configurar el protocolo DHCP |
| Nombre de la persona que realiza la prueba: Mijail Hurtado Fedorovich. | |
| Descripción de la Prueba: Se ejecuta el método <code>config_dhcp</code> de Lira, y se le pasa por parámetro el valor que necesita y se verifica que escribe el parámetro sin problemas. | |
| Condiciones de Ejecución: <code>config_dhcp</code> debe estar en correcto estado y debe ser ejecutado con permisos de root con el fin de poder guardar los cambios que se realicen al fichero situado en <code>/etc/dhcp</code> con el nombre de <code>dhcpd.conf</code> | |
| Entrada / Pasos de ejecución: Se introducen los datos en la pantalla "Configuración de DHCP" que permitirán un correcto funcionamiento del protocolo en cuestión. | |
| Resultado Esperado: Los valores insertados deberán aparecer en el archivo de configuración donde les corresponda. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 3.12 Caso de Prueba realizado a la Historia de Usuario: Configurar el protocolo DHCP.

Caso de Prueba Historia de Usuario: HI-3

Esta sección cubre un conjunto de pruebas funcionales relacionadas con la historia de usuario: **Configurar protocolo NFS.**

| Caso de Prueba de Aceptación | |
|--|--|
| Código Caso de Prueba: 1 | Nombre Historia de Usuario: Configurar el protocolo NFS |
| Nombre de la persona que realiza la prueba: Mijail Hurtado Fedorovich. | |
| Descripción de la Prueba: Se ejecuta el método <code>conf_export</code> de Lira, y se le pasa por parámetro el valor que necesita verificándose que escribe el parámetro sin problemas. | |
| Condiciones de Ejecución: <code>conf_export</code> debe estar en correcto estado y debe ser ejecutado con permisos de root con el fin de poder guardar los cambios que se realicen al fichero situado en <code>/diskless</code> con el nombre de <code>exports</code> | |
| Entrada / Pasos de ejecución: Se solicita al administrador, en la pantalla correspondiente, que entre la subred (es válida tanto para el TFTP como para el NFS). | |
| Resultado Esperado: Los valores insertados deberán aparecer en el archivo de configuración donde les corresponda. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 3.13 Caso de Prueba realizado a la Historia de Usuario: Configurar el protocolo NFS.

Caso de Prueba Historia de Usuario: HI-4

Esta sección cubre un conjunto de pruebas funcionales relacionadas con la historia de usuario: **Iniciar Demonios**.

| Caso de Prueba de Aceptación | |
|---|---|
| Código Caso de Prueba: 1 | Nombre Historia de Usuario: Iniciar Demonios |
| Nombre de la persona que realiza la prueba: Mijail Hurtado Fedorovich. | |
| Descripción de la Prueba: Se ejecuta el método Iniciar_Demonio1 de Lira | |
| Condiciones de Ejecución: Iniciar_Demonio1 debe estar en correcto estado y debe ser ejecutado con permisos de root con el fin de poder reiniciar los demonios recién configurados. | |
| Entrada / Pasos de ejecución: Una vez que se aplican los cambios efectuados a cada fichero se procede a reiniciar los demonios para lo cual se le solicita al usuario que seleccione aquellos que deberán incluirse en el runlevel default del sistema | |
| Resultado Esperado: Se inician los demonios DHCPD, NFS Y TFTP y se incluyen los procesos seleccionados en el runlevel default del sistema. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 3.14 Caso de Prueba realizado a la Historia de Usuario: Iniciar Demonios

3.5 Resultados Obtenidos:

Como fruto del trabajo llevado a cabo por los autores del presente trabajo se obtuvo la solución Lira que incluye la herramienta de configuración del mismo nombre en su versión 0.2, un kernel parcheado con AUFS y una imagen del sistema operativo Nova; que está siendo utilizada por el proyecto Sistclon de la Universidad de las Ciencias Informáticas, así como por el proyecto Nova al cual pertenece, y se hacen gestiones para que se utilice por la Fiscalía General de la República y el Ministerio de Cultura, por lo que se pueden esperar mejores resultados en versiones superiores.

Es importante destacar que la calidad de la solución posibilitó lograr conectar una mayor cantidad de diskless en una subred con menos recursos con respecto a las herramientas similares analizadas en el capítulo uno como se muestra en la siguiente tabla cuyos datos son resultado de las pruebas prácticas realizadas en el laboratorio de desarrollo de la aplicación utilizando la modalidad de procesamiento en el lado del cliente.

| Servidor con GNU/Linux (LTSP: Linux Terminal Server Project) | Servidor con Windows (WinConnect) | Servidor con GNU/Linux (Lira) |
|---|---|---|
| Procesador 3.0 Ghz o mayor | Procesador 3.0 Ghz o mayor. | Procesador 3.0 Ghz. |
| 1 GB de Memoria RAM | 2 GB (3 GB recomendado) de Memoria RAM | 1 GB de Memoria RAM |
| Placa de red de 1Gbps | Placa de red de 1Gbps | Placa de red de 1Gbps |
| Disco Rígido 80 GB | Disco Rígido 100 GB | Disco Rígido 160 GB |
| Sistema Operativo GNU/Linux. | Sistema Operativo Windows | Sistema Operativo Nova |
| Cantidad de Terminales que conecta: 18 a 21 | Cantidad de Terminales que conecta: 18 a 21 | Cantidad de Terminales que conecta: 25 a 35 |
| Terminales con GNU/Linux | Terminales con Windows | Terminales con imagen Nova de Lira |
| Procesador Pentium 66 MHz o mayor | Procesador 366MHz o mayor | Procesador Pentium 4 a 1.3 Ghz |
| 16 MB RAM (32 MB recomendado) | 128 MB RAM | 256 MB RAM |
| Placa de red 10 Mbps (PCI recomendado) | Placa de red 10 Mbps (PCI recomendado) | Placa de red 10 Mbps (PCI) |
| Monitor | Monitor | Monitor |
| Teclado | Teclado | Teclado |
| Mouse | Mouse | Mouse |

Tabla 3.15 Comparación de hardware entre LTSP, WinConnect y Lira¹

3.6 Acerca de las funcionalidades obtenidas.

Entre las principales funcionalidades que posee la herramienta Lira v0.2 se pueden mencionar:

- Es capaz de configurar los protocolos DHCP, TFTP y NFS en el servidor de diskless.
- Es capaz de reiniciar los demonios dhcpd, in.tftpd y nfs en el servidor de diskless e incluir aquellos que se seleccionen en el runlevel default del sistema.

¹ Los datos de Lira son extraídos de las pruebas hechas en el laboratorio con los clientes Hanel que se están importando al país.

Conclusiones Generales

Al finalizar este capítulo los desarrolladores concluyeron la implementación de la herramienta Lira en su versión 0.2 con una planificación adecuada de su tiempo basada en el cumplimiento de cada una de las tareas de ingeniería definidas en el presente capítulo, así como en el plan de releases que define cuanto tiempo tomará el lanzamiento de cada versión del producto final. Además contaron con un estándar de codificación apropiado para lograr que el código fuera lo más entendible posible y permitir así su reutilización en el futuro o que otro desarrollador aporte nuevas funcionalidades a la aplicación. También se documentaron algunas pruebas realizadas al software para validar la calidad del mismo. Derivado del resultado de estas se llega a la conclusión de que el software está listo para su uso y distribución en su versión 0.2.

Con el presente trabajo se han logrado los objetivos trazados, se ha realizado una investigación de las aplicaciones que permiten configurar servidores a nivel mundial de las que se detalló el funcionamiento y características principales. Se plasmó en este documento todo el proceso de ingeniería de software imprescindible para el desarrollo del sistema propuesto. Se elaboraron una serie de diagramas y tablas que facilitan al lector el entendimiento de todo el proceso de concepción del software propuesto.

A lo largo de la investigación, los autores tuvieron que afrontar riesgos que pusieron en peligro el cumplimiento de las metas propuestas entre los cuales estuvo la poca experiencia con el lenguaje de programación, que redundó en mayor tiempo de estudio del mismo. En los inicios, tenían poco conocimiento de la configuración de los protocolos en los servidores de diskless, lo que significó dedicar tiempo extra en el aprendizaje de estos. El riesgo más grande afrontado por el equipo de trabajo fue la situación en que se encontró uno de los desarrolladores que tuvo que duplicar esfuerzos en otras tareas que le fueron asignadas, recayendo en el otro el peso de todo el trabajo, lo cual retrasó en gran medida las fechas de entrega planificadas en un inicio y a la postre determinó que no se pudieran implementar todas las historias de usuario adecuadamente.

Al término de la presente investigación se encuentra lista la solución Lira que incluye la herramienta del mismo nombre en su versión 0.2, un kernel parchado con AUFS y una imagen optimizada del sistema operativo Nova para el uso en clientes ligeros que permitirá configurar los protocolos necesarios para la puesta en marcha de un servidor

de diskless de una manera más sencilla y automática, lo cual redundará en ahorro de tiempo y esfuerzo por parte de los administradores de dichos servidores. La herramienta Lira por la arquitectura con la que fue programada permite una fácil implantación en otras plataformas y el uso de otras imágenes y distribuciones de GNU/Linux.

Recomendaciones

Teniendo en cuenta que esta solución se piensa utilizar donde quiera que se desee montar una red de diskless es necesario tener presente las siguientes recomendaciones para lograr un producto de mayor calidad y funcionalidad:

- Realizar las iteraciones cuatro y cinco que permitirán implementar las Historias de Usuario *Configurar por MAC* y *Monitorear Conexiones* así como cumplir con el cuarto release del Plan de Releases de la herramienta Lira.
- Diseñar una nueva interfaz gráfica bajo las normas de la *Guía para las Interfaces Gráficas de GNOME versión 2.0* que incluya las Historias de Usuario *Monitorear Conexiones* y *Configuración por MAC*.
- Implementar nuevas funcionalidades que aporten mayor versatilidad a la herramienta Lira.
- Crear imágenes optimizadas para otras distribuciones de GNU/Linux e integrarlas a la solución Lira.
- Utilizar Lira para la configuración de diferentes redes de diskless para obtener inconformidades que constituyan posibles mejoras a la aplicación.

Referencias Bibliográficas

[IT00] Discurso pronunciado por el miembro del Consejo de Estado, Carlos Valenciaga Díaz en el acto en ocasión de la Segunda Graduación | Portal UCI - Universidad de las Ciencias Informáticas. [visitado el 13 Febrero 2009]. Disponible en Internet: <<http://www.uci.cu/?q=node/340>>.

[IT01] Angel Goñi Oramas. Proyecto Nova, base para crear un sistema operativo con repercusiones sociales. 2007.

[IT02] León Geeko. La era pervasiva: Clientes ligeros: Wyse S10. [visitado el 8 Febrero 2009]. Disponible en Internet: <<http://pervasiva.blogspot.com/2007/11/clientes-ligeros-wyse-s10.html>>.

[PXE00] Intel Corporation. Preboot Execution Environment(PXE) Specification. Version 2.1. Septiembre 1999. [citado el 20 Mayo 2009]. Disponible en Internet: <<http://www.pix.net/software/pxeboot/archive/pxespec.pdf>>.

[CL02] Clientes Ligeros | Ultimas. [visitado el 12 Febrero 2009]. Disponible en Internet: <<http://rubikpc.com/shop/index.php/Ultimas/Clientes-Ligeros.html>>.

[XDMCP00] Linux XDMCP HOWTO. [visitado el 20 Mayo 2009]. Disponible desde internet: <<http://tldp.org/HOWTO/XDMCP-HOWTO/index.html>>.

[VNC00] VNC - Virtual Network Computing from AT&T Laboratories Cambridge. [visitado el 9 Junio 2009]. Disponible en Internet: <http://www.hep.phy.cam.ac.uk/vnc_docs/index.html>.

[RDP00] Remote Desktop Protocol. [visitado el 9 Junio 2009]. Disponible en Internet: <http://es.wikipedia.org/wiki/Remote_Desktop_Protocol>.

[DHCP00] RFC 2131 - Dynamic Host Configuration Protocol. [visitado el 21 Mayo 2009]. Disponible en internet: <<http://tools.ietf.org/html/rfc2131>>.

[TFTP00] RFC 1350 - The TFTP Protocol (Revisión 2). [visitado el 20 Mayo 2009]. Disponible en Internet: <<http://www.faqs.org/rfcs/rfc1350.html>>.

[NFS00] RFC 1094 - NFS: Network File System Protocol specification. [visitado el 20 Mayo 2009]. Disponible en internet: <<http://www.faqs.org/rfcs/rfc1094.html>>.

[DNS00] RFC 1035 - Domain names - implementation and specification. [visitado el 20 Mayo 2009]. Disponible en Internet: <<http://www.faqs.org/rfcs/rfc1035.html>>.

[CODA00] Coda File System. [visitado el 9 Junio 2009]. Disponible en Internet: <<http://www.coda.cs.cmu.edu/>>.

[iSCSI00] Internet Small Computer Systems Interface (iSCSI). [visitado el 9 Junio 2009]. Disponible en Internet: <<http://www.ietf.org/rfc/rfc3720.txt>>.

[Aoe00] Ata over ethernet. [visitado el 9 Junio 2009]. Disponible en Internet: <http://es.wikipedia.org/wiki/Ata_over_ethernet>.

[NBD00] The Enhanced Network Block Device. [visitado el 9 Junio 2009]. Disponible en Internet: <<http://www.it.uc3m.es/ptb/nbd/>>.

[TE00] Ernesto Díaz Vázquez, Antonio Ferral Sainz. Servidor de Terminales Ligeras. Junio 2008, 83.

[VP00] Visual Paradigm for UML - Wikipedia, the free encyclopedia. [visitado el 10 Febrero 2009]. Disponible en Internet: <http://en.wikipedia.org/wiki/Visual_Paradigm_for_UML>.

[BU00] Alejandro Pérez García. BOUML, una herramienta CASE de UML gratuita. [visitado el 2 Marzo 2009]. Disponible en Internet: <<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=bouml>>.

[PY00] Python - Wikipedia, la enciclopedia libre. [visitado el 6 Febrero 2009]. Disponible en Internet: <<http://es.wikipedia.org/wiki/Python>>.

[PY01] Python Programming Language -- Official Website. [visitado el 6 Febrero 2009]. Disponible en Internet: <<http://www.python.org/>>.

[SCRUM00] Beneficios de Scrum | proyectos Ágiles. [visitado el 20 Mayo 2009]. Disponible en Internet: <<http://www.proyectosagiles.org/beneficios-de-scrum>>.

[QT00] Qt (biblioteca) - Wikipedia, la enciclopedia libre. [visitado el 6 Febrero 2009]. Disponible en Internet: <[http://es.wikipedia.org/wiki/Qt_\(biblioteca\)](http://es.wikipedia.org/wiki/Qt_(biblioteca))>.

[LC00] Clientes Ligeros. [visitado el 13 Febrero 2009]. Disponible en Internet: <<http://thinclients.es/lancore.php>>.

[OTC00] Free Open Source Thin Client Solution - OpenThinClient : home. [visitado el 13 Febrero 2009]. Disponible en Internet: <<http://openthinclient.org/home>>.

[LTSP00] LTSP - Linux Terminal Server Project. [visitado el 9 Junio 2009]. Disponible en Internet: <<http://www.ltsp.org/>>.

[PXES00] PXES Universal Linux Thin Client. [visitado el 8 Junio 2009]. Disponible en Internet: <<http://dtmilano.googlepages.com/pxesuniversallinuxthinclient>>.

[TCOS00] TCOS Project (thin client operating system). [visitado el 8 Junio 2009]. Disponible en Internet: <<http://www.tcosproject.org/>>.

[WC00] Thin Client Software, Linux Thin Client, RDP Client, RDP Software, Linux RDP, Terminal Services. [visitado el 13 Febrero 2009]. Disponible en Internet: <http://www.thinsoftinc.com/product_rdp_client_winconnect.aspx>.

[WC01] Infogal Servicios Informáticos. [visitado el 18 February 2009]. Disponible en Internet: <<http://www.infogal.net/pdf/pyme.pdf>>.

[MA00] Gregorio R. M. , Jorge F. Z. *MA00-Programación Extrema y Software Libre*. 2002.

[AI00] Dayron Pérez Roldán. Sistema de Clonación y Distribución de Imágenes de Sistemas Operativos. Junio 2008.

[AI01] Programación por capas - Wikipedia, la enciclopedia libre. [visitado el 19 Marzo 2009]. Disponible en Internet: <http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles>.

[EP00] Guía de estilo del código Python. [visitado el 5 Mayo 2009]. Disponible en Internet: <<http://mundogeek.net/traducciones/guia-estilo-python.htm>>.

[CPC00] ¿Cuanta electricidad consume un PC? [visitado el 21 Mayo 2009]. Disponible en Internet: <<http://www.taringa.net/posts/info/1146726/%C2%BFCuanta-electricidad-consume-un-PC.html>>.

[UML00] Paul Kimmel. *UML: Demystified*. Mc Graw Hill, 2005 [citado el 25 Mayo 2009].

Anexos

Anexo 1 - ¿Cuánto consume una computadora?

El consumo total de una computadora está dado por la suma del consumo de todos sus componentes:

CPU = 40W - 125W

Placa Madre = 10W

Ventilador = 5W

Tarjeta gráfica incorporada a la placa = 20W - 50W

Fuente de poder = 10W

Disco duro = 5W - 10W

CD-ROM = 3W - 6W

Tarjeta de video independiente = 59 W como promedio

En total son unos 245 W. Mientras más potentes sean los componentes mayor cantidad de energía consumirá la computadora en su conjunto. **[CPC00]**

Anexo 2 – Plantilla de Concepción del Sistema

Nombre del Proyecto: Nova.

Nombre del Producto: Lira: Asistente Gráfico para la configuración de servidores de terminales ligeras con Sistemas Nova.

Versión: 0.2

Polo Productivo: Software Libre.

Clasificación del proyecto: Desarrollo de aplicación.

Tipo de Proyecto: Nacional.

Resumen: Este producto pretende implementar una herramienta que le facilite el trabajo de configuración de servidores de diskless con sistemas Nova (por el momento) a los administradores de los mismos. Entre los protocolos que se pretende que configure dicho asistente se encuentran: DHCP, NFS y TFTP.

Palabras claves: Herramienta, Diskless, Nova, servidores.

Surgimiento: La idea de desarrollar una aplicación como está surge a partir de la puesta en práctica de la tecnología de clientes ligeros con sistemas Nova en algunas terminales. Ante la dificultad de configuración que se observó se propuso elaborar un software que permitiera que estas tareas se realicen de una manera más sencilla y más ágil ya que a la hora de configurar gran cantidad de terminales sería un trabajo bastante tedioso.

Qué es: La aplicación es un asistente gráfico que a través de una serie de pantallas y pasos permita la configuración sencilla de los protocolos necesarios para que un servidor de clientes ligeros funcione adecuadamente.

Metodología a utilizar: Se utilizará la metodología ágil SXP que a grosso modo es una compilación de las mejores prácticas de dos metodologías ágiles muy bien conocidas como son Scrum y XP. Esta metodología permitirá a los autores de esta investigación implementar en menor tiempo la aplicación con un equipo reducido de desarrolladores, en este caso solo dos personas.

Involucrados: Ingeniero Abel Meneses Abad, experto en desarrollo de aplicaciones de software libre. Ingeniero Mijail Hurtado Fedorovich, experto en servidores de terminales ligeras. Viviana Álvarez Ferriol y Alberto Aragón Álvarez, estudiantes universitarios de la Universidad de las Ciencias Informáticas.

Roles:

| Rol | Responsabilidad | Nombre |
|-------------|---|---|
| SCRUM | Su principal función va a ser la de asegurarse que se cumplan los objetivos trazados con el producto y que las tareas se cumplan en tiempo y con la calidad requerida. | Abel Meneses Abad |
| Gerente | Tomar decisiones acerca de estándares y convenciones a seguir durante el proyecto. Participa en la definición de objetivos y requerimientos. | Mijail Hurtado Fedorovich |
| Cliente | Ayudará a definir la Historias de Usuario y los casos de prueba de aceptación, para validar su implementación. Asignará la prioridad de las Historias de Usuario y en que iteración se van a realizar | Proyecto Nova |
| Programador | Define las tareas de ingeniería y produce el código del sistema. Selecciona el estándar de programación a utilizar. | Viviana Álvarez Ferriol Alberto Aragón Alvarez |
| Analista | Escribe la concepción del sistema y las historias de usuario. Crea el modelo de Historia de Usuario del Negocio así como la Lista de Reserva del Producto. Trabaja en colaboración con el cliente. | Viviana Alvarez Ferriol |
| Diseñador | Encargado del diseño del sistema, así como de los prototipos de interface, máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción. | Alberto Aragón Alvarez |
| Tester | Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente. Responsable de soporte de pruebas. | Mijail Hurtado Fedorovich |

Misión: La misión de este sistema es facilitar el trabajo de los administradores de servidores de terminales ligeras a la hora de configurar los protocolos necesarios para que estos funcionen adecuadamente.

Visión: Las aspiraciones de los autores del producto propuesto son que esta aplicación contribuya a la implementación de la tecnología de clientes ligeros con sistemas Nova en las diversas áreas de la economía y de la sociedad de nuestro país donde su aplicación reporte beneficios para el pueblo.

Alcance: Se pretende poner en práctica la aplicación en todas las instituciones u organismos que necesiten clientes ligeros.

Herramientas utilizadas: Para la realización de este sistema se utilizarán diferentes herramientas como: BoUML, para los diagramas de la metodología; Python, como lenguaje de programación y las librerías PyQT y QT para implementar algunas funcionalidades con el lenguaje antes mencionado.

Solución propuesta: Se propone desarrollar una herramienta de configuración para servidores de diskless con una implementación sencilla y ligera que pueda ser ejecutada en los más diversos ámbitos.

Anexo 3 – Lista de Reserva del Producto

| Requisitos Funcionales | | | | |
|------------------------|------|--|-------------------------------|-------------------|
| Asignado a | Item | Descripción | Estimación | Estimado por |
| Prioridad | | Muy Alta | | |
| Viviana | 1 | Configuración gráfica de los protocolos: <ul style="list-style-type: none"> •DHCP •TFTP •NFS | 1 semana para cada protocolo. | Alberto y Viviana |
| Prioridad | | Alta | | |
| Alberto | 2 | Integración con el sistema operativo Nova-Gentoo y que permita levantar los servicios correspondientes. | 1 semana | Alberto y Viviana |
| Viviana | 3 | Iniciar los demonios correspondientes a los protocolos que se van a configurar. | 0.5 semana | Alberto y Viviana |
| Prioridad | | Media | | |
| Alberto | 4 | Configuraciones diferentes y restringidas en dependencia de la MAC de la computadora. | 1 semana | Alberto y Viviana |
| Alberto | 5 | Monitorear las conexiones realizadas al servidor (quién está conectado y quien no). | 1 semana | Alberto y Viviana |
| Prioridad | | Baja | | |
| | | | | |

| Requisitos No Funcionales | | | | |
|----------------------------------|-------------|---|-------------------|---------------------|
| Asignado a | Item | Descripción | Estimación | Estimado por |
| Prioridad | | Muy Alta | | |
| Alberto | 1 | El software debe contar con una interfaz gráfica de administración. | 1 semana | Alberto y Viviana |
| | 2 | El software debe ser fácil de manejar y de comprender. | | Alberto y Viviana |
| | 3 | El servidor donde se alojara la aplicación debe tener como mínimo: <ul style="list-style-type: none"> •Procesador: Pentium 4 a 2.0 GHZ o superior. •Memoria: 1GB de RAM o superior •Tarjeta de Red: Ethernet LAN 10/100 Mbps •Capacidad de HDD: 1-2 GB mínimo | | |
| Prioridad | | Alta | | |
| Viviana | 4 | Identificar el usuario, según la MAC y darle solo los permisos que le corresponden. | 0.5 semana | Alberto y Viviana |
| Viviana | 5 | Proteger la información de accesos no autorizados. | 0.5 semana | Alberto y Viviana |
| Alberto | 6 | Debe contar con un sistema de salvallas externas en caso de desastre. | 0.1 semana | Alberto y Viviana |
| Prioridad | | Media | | |
| Alberto | 8 | El sistema debe correr en GNU/Linux para cualquier distribución y en Windows. | | Alberto y Viviana |

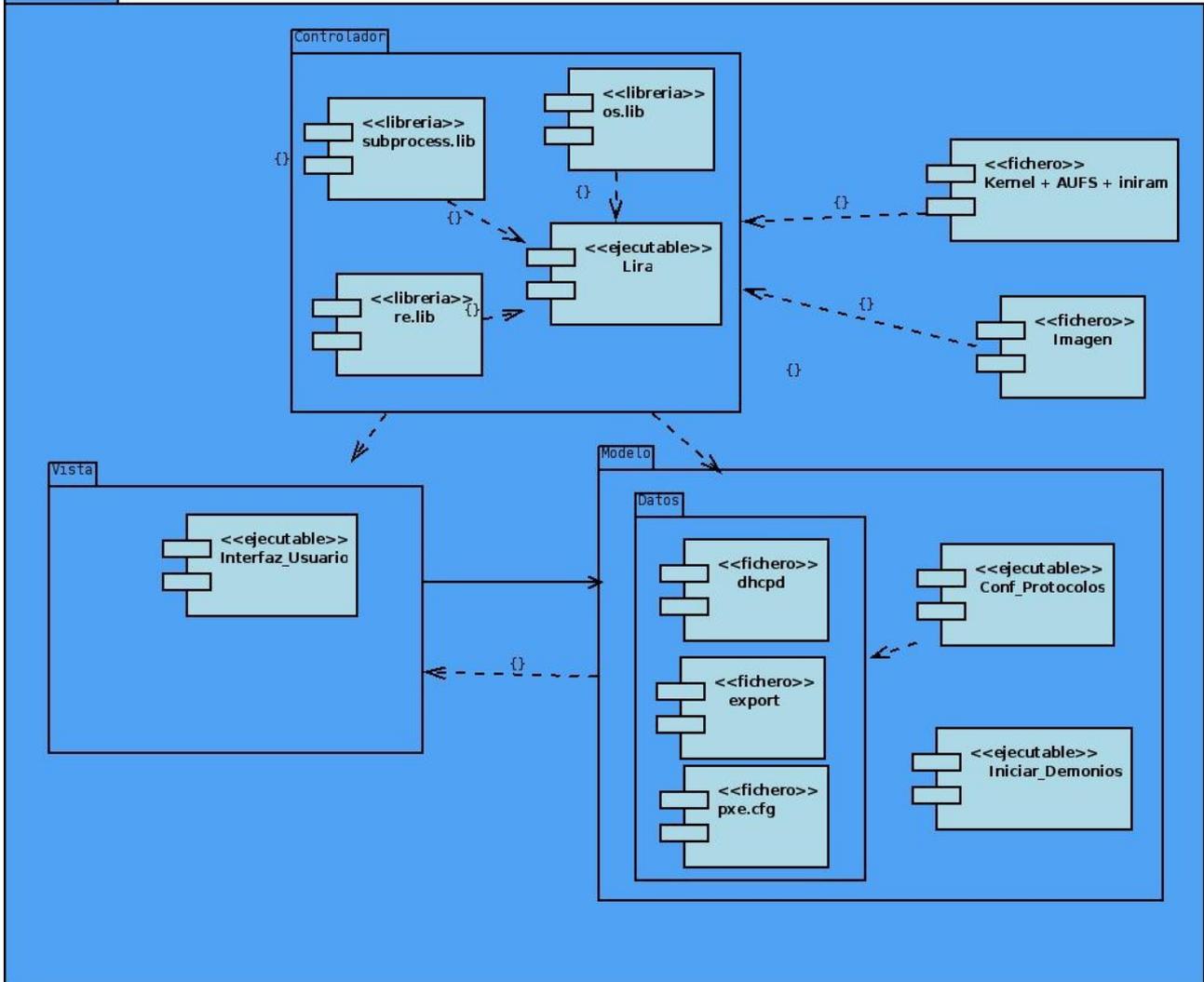
| Prioridad | | Baja | | |
|-----------|-----|--|------------|-------------------|
| Viviana | 1,2 | Interfaz gráfica con GTK. | 1 semana | Alberto y Viviana |
| Alberto | 9 | Hacer un Manual de instalación y un Manual de Usuario. | 0.4 semana | Alberto y Viviana |

Anexo 4 - Plantilla de Planificación de Historias de Usuarios.

| No | Nombre HU | Prioridad | Riesgo | Esfuerzo (Días) | Iteración |
|----|---------------------------|-----------|--------|-----------------|-----------|
| 1 | Configurar Protocolo TFTP | Muy Alta | Alto | 7 | 1 |
| 2 | Configurar Protocolo DHCP | Muy Alta | Alto | 7 | 2 |
| 3 | Configurar Protocolo NFS | Alta | Alto | 7 | 3 |
| 4 | Iniciar Demonios | Alta | Alto | 5 | 3 |
| 5 | Configuración por MAC | Media | Alto | 14 | 4 |
| 6 | Monitorear Conexiones | Media | Alto | 14 | 5 |

Anexo 5 - Diagrama de Componentes

Solucion_Lira



Anexo 6 - Parchear kernel con AUFS

Lo primero que hay que tener es el paquete AUFS copiado en la carpeta destino donde se va a parchear el kernel además del kernel que se desea parchear. AUFS está siendo desarrollado y probado en las versiones linux-2.6.16 del kernel y posteriores. Los archivos fuentes de AUFS están en `./fs/aufs` o en `./fs/aufs25`. La configuración de AUFS está escrita en `./Kconfig.in`. Antes de construir `./fs/aufs/Kconfig` se necesita configurar el kernel para lo cual, suponiendo que la carpeta donde está el kernel sea `/kernel`, tendremos que seguir los siguientes pasos en modo consola:

- `cd /kernel/source`

- `make menuconfig`

- Ir en `/kernel/local` al `local.mk` y poner la dirección donde se encuentran los sources del kernel en la variable `KDIR`.

- `make include/linux/version.h /include/linux/utsrelease.h` y luego ejecute

- `cd aufs.wcvs/aufs`

- `rm fs/aufs/Kconfig`

- `make -f local.mk kconfig`

- Copiar la carpeta de `/aufs/fs/aufs25` para `/kernel/fs`.

- Copiar `/aufs/include/linux/aufs_type.h` para `/kernel/include/linux`.

- Adicionar en `/kernel/fs/Makefile` "`obj-$(CONFIG_AUFS) += aufs/`"

- Adicionar en `/kernel/fs/Kconfig` en la sección AUFS `source "fs/aufs/Kconfig"`

- Luego parcheamos el kernel con:

- `patch -p0 < ../aufs/patch/put_filp.patch`

- `patch -p1 < ../aufs/patch/sec_perm-2.6.24.patch`

Anexo 7 - Archivos de Configuración

```
get-lease-hostnames true;
allow booting;
allow bootp;
option broadcast-address 10.33.4.255 ;

# Declaracion del space pxenova
option space pxenova;
option pxenova.magic      code 208 = string;
option pxenova.configfile code 209 = text;
option pxenova.pathprefix code 210 = text;
option pxenova.reboottime code 211 = unsigned integer 32;

include "/var/lib/dhcp/macperhosts";

subnet 10.33.4.0 netmask 255.255.255.0
{ pool {
    option domain-name "domain.example.org";
    option domain-name-servers 10.0.0.3, 10.0.0.4;
    option routers 10.33.4.254;
    default-lease-time 604800;
    max-lease-time 604800;
    site-option-space "pxenova";
    if exists dhcp-parameter-request-list {
        option dhcp-parameter-request-list = concat(option dhcp-parameter-request-list,d0,d1,d2,d3);
    }
    option pxenova.configfile "pxe.cfg";

    #option pxenova.pathprefix "/tftpboot";
    option pxenova.reboottime 5;

    #especifica el servidor de tftp
    next-server 10.33.4.156;
    filename "/pxelinux.0";
    range 10.33.4.9 10.33.4.19;
    allow unknown-clients;
}
}
```

Archivo de configuración dhcpd.conf

```
/diskless/nova 10.33.4.156/24(ro,async,no_root_squash,no_subtree_check)
```

Archivo de configuración exports

```
#label memtest
#kernel ../memtest.bin

label linux
say Iniciando PXE
kernel ../vmlinuz
append initrd=../initramfs.cpio.gz ip=dhcp nfsroot= 10.33.4.5:/diskless/nova root_type=nfs nfsparms=ro,hard,nolo
```

Archivo de configuración pxe.cfg

```
INTFTPD_PATH="/diskless/boot/tftpboot"
INTFTPD_OPTS="-R 4096:32767 -l -s ${INTFTPD_PATH}"
```

Archivo de configuración in.tftpd

Anexo 8 – Crear una imagen

El proceso de crear una imagen es algo complicado a la hora de optimizarla para que consuma pocos recursos y que sea bastante liviana. Para lo anterior se utilizará una partición del disco duro donde se instalará la distribución de GNU/Linux deseada. Luego de realizar esto se procederá a instalar cada una de las aplicaciones que deseemos se encuentren disponibles en la imagen final. Se deben escoger, si existen varias alternativas, entre las aplicaciones que tienen un mismo fin aquellas que consuman menor cantidad de memoria RAM y que posean una interfaz gráfica sencilla que posibiliten que el diskless las ejecute más rápidamente. También se pueden configurar cada una de estas aplicaciones desabilitando ciertas funciones que permitan incrementar su rendimiento y disminuir el uso de memoria de las mismas. Luego de tener instalada la distribución de GNU/Linux y todas las aplicaciones (correctamente optimizadas y configuradas) se procederá al proceso de comprimir dicha imagen para lo cual se utilizará el comando “*tar -cjpf pepe.tar.bz2 bin/ boot/ dev/ etc/ home/*” donde “*pepe*” es el nombre que le vamos a asignar a la imagen y a continuación el formato del comprimido que se desee, los autores recomiendan *tar.bz2*; el resto del comando son todas las carpetas que queremos que se incluyan en la imagen.

Glosario de Términos

Diskless: es una computadora que carece de disco duro y que realiza todas sus funcionalidades por la red a través de un servidor. También una computadora de bajas prestaciones se puede habilitar como un diskless lo cual permite rescatar computadoras que ya no se usan por considerarse obsoletas.

GNU/Linux: es el término para referirse al sistema operativo similar a Unix que utiliza como base las herramientas del sistema GNU y el núcleo Linux. Es un software libre distribuido bajo la licencia GPL (Licencia Pública General de GNU) y otras. Las variantes de este sistema se denominan distribuciones y su objetivo es ofrecer un sistema ajustado a las necesidades de determinado grupo de usuarios.

SWAP: en informática es el espacio en un disco duro reservado para *intercambiar* (de aquí su nombre) con la memoria RAM aquellos procesos poco activos y de esta forma liberar la memoria principal para otros procesos. El proceso podrá estar guardado en el disco hasta que sea necesario y de esta forma no gastar memoria física. Generalmente se reserva una partición de un tamaño determinado para que funcione como swap.

DHCP: siglas en inglés de Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Anfitrión). Es un protocolo que permite a los nodos de una red obtener los parámetro de red dinámicamente (Máscara de red, puerta de enlace broadcast y otros). Es un protocolo de tipo cliente servidor en el que generalmente el servidor posee una lista de direcciones IP y las va asignando a los clientes conforme están van estando disponibles, manteniendo siempre un registro de quien ha tenido esa IP, cuánto tiempo y a quién se le ha asignado después.

NFS: siglas en inglés de Network Fyle System (Sistemas de Archivos de Red). Es un protocolo de la capa de aplicación según el modelo OSI. Es utilizado para sistemas de archivo distribuidos en una red y permite ver archivos remotos como si fueran locales. El protocolo NFS está por defecto incluido en los sistemas operativos UNIX y las distribuciones de GNU/Linux.

Modelo OSI: el modelo de referencia de interconexión de sistemas abiertos (OSI Open System Interconnection) lanzado en 1984 y creado por ISO, constituye el marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

PXE: siglas en inglés de Preboot eXecution Environment (Entorno de Ejecución de Prearranque). Es un entorno para arrancar e instalar el sistema operativo en computadoras desde una red. Está relacionado con varios protocolos de red como IP, DHCP, TFTP y UDP.

TFTP: siglas en inglés de Trivial File Transfer Protocol (Protocolo de Transferencia de Archivos Trivial). Es un protocolo de transferencia de archivos bastante sencillo, algo así como una versión básica de FTP. 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. Su característica distintiva es que no necesita de autenticación para establecer la transferencia y tampoco tiene mecanismos de cifrado.

X Window: el sistema de ventanas X (X Window System) fue desarrollado a mediados de los años '80 en el MIT (instituto Tecnológico de Massachusetts) para dotar de una interfaz gráfica a los sistemas Unix. Este protocolo permite la interacción gráfica en red entre un usuario y una o más computadoras haciendo transparente la red para éste. Generalmente se refiere a la versión 11 de este protocolo, X11, el que está en uso actualmente.

DNS: siglas en inglés de Domain Name System (Sistema de Nombres del Dominio). Es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Su uso más común es la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio. Así el sitio de Google en vez de ser `http://82.100.200.33` (ejemplo) quedaría `www.google.com`.

eduNova: distribución de Nova orientada a fines educativos.

serverNova: distribución de Nova orientada a servidores.

GPL: La Licencia Pública General de GNU es una licencia creada por la Free Software Foundation (Fundación del Software Libre) que está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Internet: es un conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolo TCP/IP garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

TCP/IP: La familia de protocolos de Internet es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

GTK: GTK+ o The GIMP Toolkit es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (GUI).

GNOME: es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix/Linux, compuesto enteramente de software libre.

Red Hat: es la compañía responsable de la creación y mantenimiento de una distribución del sistema operativo GNU/Linux que lleva el mismo nombre: Red Hat Enterprise Linux, y de otra más, Fedora.

Mac OS: abreviatura de Macintosh Operating System (Sistema Operativo de Macintosh), es el nombre del primer sistema operativo de Apple para los ordenadores Macintosh. El Mac OS original fue el primer sistema operativo con una interfaz gráfica de usuario en tener éxito.

RAM: de las siglas en inglés Random Acces Memory o Memoria de Acceso Aleatorio. Es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados. Es el área de trabajo para la mayor parte del software de un computador.

HDD: de las siglas en inglés Hard Disc Drive o Unidad de Disco Duro. Es un dispositivo de almacenamiento no volátil que conserva la información aún con la pérdida de energía empleando un sistema de grabación magnética digital.

Runlevel: en español nivel de ejecución se refiere a un estado en el cual su sistema está corriendo y contiene guiones que serán ejecutados al ingresar o salir del nivel de ejecución. En Gentoo hay siete niveles definidos, tres internos y cuatro definidos por el usuario. Los definidos por el usuario son: *boot* que inicia los servicios necesarios para los otros niveles de ejecución, *default* para uso diario, *nonetwork* en caso de no requerirse la red y *single* en caso de necesitar arreglar el sistema.

Demonio: es un programa(s) informático(s) que se ejecuta(n) en segundo plano en Unix/Linux para brindar algún servicio como puede ser correo, servicio de impresión y otros. Estos programas se mantienen en ejecución mientras el sistema esté activo o hasta que el usuario no lo detenga. Un servicio puedes estar soportado por más de un demonio.