

Máster de Software Libre

Trabajo de Fin de Máster

Título:

Servidor de Escritorios Remotos

Especialización:

Administración de Redes y Sistemas Operativos

Autor: **Edwin Boza Gaibor**
Tutor UOC: **Jordi Massaguer Pla**
Tutor Externo: **Ricard Iravedra (OpenTrends)**

Junio 2012



*Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual 3.0 España de Creative Commons.
La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-sa/3.0/es/>*

AGRADECIMIENTO

Quiero empezar agradeciendo a mis tutores, Jordi y Ricard, por su invaluable guía y apoyo durante este semestre. Sin duda han sido fundamentales para haber llevado a buen término este proyecto. Gracias también por su paciencia.

A Antonio Rodil, que nos ha acompañado como tutor a lo largo del máster, y a todos los profesores y consultores de las diferentes materias, que además de compartir sus conocimientos y experiencia han puesto el toque humano tan necesario en programas de estudios de tipo virtual como este.

A Gladys, mi compañera de vida, por animarme a seguir estudiando y luego soportar la falta de tiempo, el mal humor y las frustraciones que inevitablemente han sido parte de este proceso. Gracias por estar ahí, por tus enseñanzas, tus consejos y sobre todo por tu amor incondicional.

Para Andrés, espero legarte un buen ejemplo.

RESUMEN

El presente proyecto ha tenido como objetivo desarrollar una prueba de concepto de un appliance en formato LiveCD que permita demostrar de manera fácil, sin necesidad de realizar tareas previas de instalación y configuración, el funcionamiento de la herramienta FreeNX para realizar conexiones de escritorios remotos.

Se ha desarrollado además una serie de interfaces web que facilitan la interacción con el sistema, facilitando las tareas de creación de usuarios y conexión remotas, así como la descarga de programas cliente y su respectiva configuración.

Durante el desarrollo del proyecto también se ha experimentado las bondades de la utilización del portal SuseStudio para la creación de distribuciones linux personalizadas, con grupos de programas y configuraciones particulares que permitan resolver problemas o necesidades puntuales.

INDICE DE CONTENIDO

1. INTRODUCCION.....	7
2. OBJETIVOS.....	8
3. ANALISIS DE REQUISITOS Y VIABILIDAD.....	10
3.1 Análisis de los requisitos.....	10
3.2 Análisis de la Viabilidad.....	11
3.2.1 Viabilidad para el Appliance LiveCD.....	11
3.2.2 Viabilidad para el Servicio de Escritorios Remotos.....	12
3.2.3 Viabilidad para la Interface Web de Administración.....	12
4. DISEÑO DE LA SOLUCION PROPUESTA.....	14
4.1 Arquitectura de la solución.....	14
4.2 Arquitectura lógica de la Interfaz Web.....	15
4.2.1 Identificación de los subsistemas.....	16
4.3 Casos de uso.....	17
4.4 Especificaciones de Desarrollo y Pruebas.....	20
5. IMPLEMENTACION.....	21
5.1 Creación del appliance en modalidad kiosko.....	21
5.2 Instalación y configuración del servicio FreeNX.....	22
5.3 Desarrollo de las interfaces web e integración en el appliance.....	22
6. RESULTADOS.....	25
7. VALORACION ECONOMICA.....	29
8. ANALISIS DE LICENCIAS.....	30
9. CONCLUSIONES.....	32

1. INTRODUCCION

Debido a diversos factores como el teletrabajo, el soporte técnico remoto, los entornos corporativos globales, entre otros, el uso de escritorios remotos es ya una realidad cotidiana que permite a las personas interactuar con aplicaciones instaladas en una determinada computadora sin necesidad de tener acceso físico a la misma o incluso estando a miles de kilómetros de distancia.

Aunque inicialmente las conexiones remotas para trabajar en computadoras o servidores ubicados en otros lugares permitían únicamente el acceso a interfaces basadas en texto debido principalmente a las restricciones de velocidad de las redes, el constante desarrollo de las comunicaciones y dispositivos de procesamiento ha permitido que en el presente se pueda acceder a entornos gráficos ejecutándose en el computador remoto, teniendo al alcance ya sea una simple aplicación o incluso todo el escritorio de trabajo de un usuario.

Este aumento de capacidades y posibilidades genera a su vez la necesidad de disponer de sistemas más elaborados que permitan no solo los ya mencionados accesos remotos, sino que también engloben herramientas administrativas que faciliten las tareas de publicación, así como la organización de los permisos de acceso a los recursos disponibles.

Existen en la actualidad algunas soluciones privativas completas, como aquellas de la reconocida marca Citrix, que ofrecen éstas y otras características avanzadas, sin embargo, las particularidades del software privativo hace que sea una necesidad buscar y desarrollar herramientas libres que permitan suplir las mismas necesidades, ofreciendo a los usuarios la posibilidad de elegir.

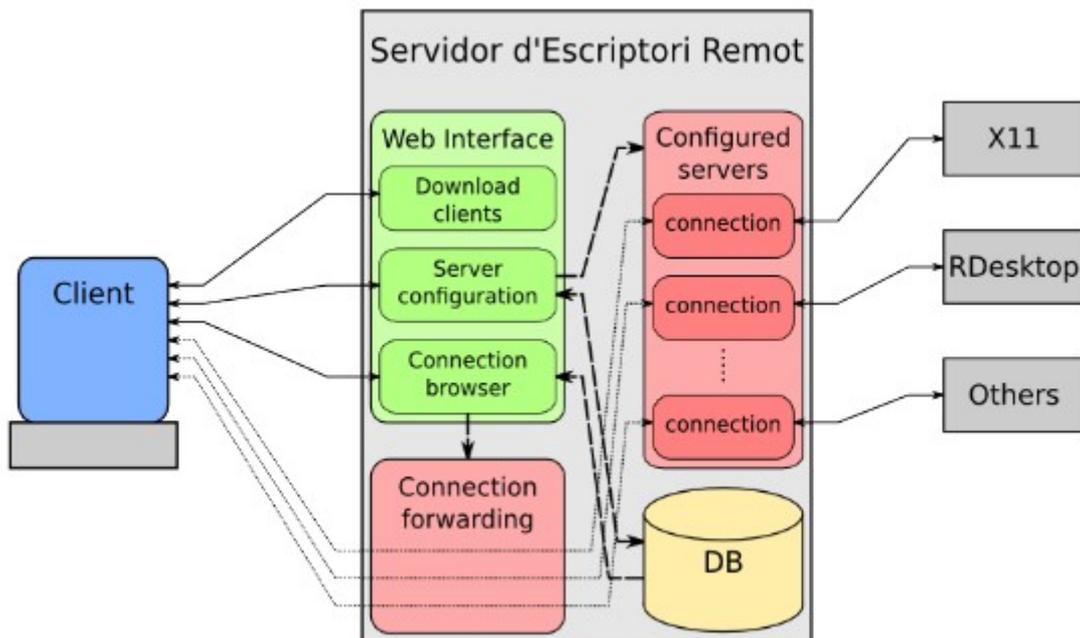
Dentro de las opciones libres para este tipo de aplicaciones se encuentra FreeNX, que es una implementación libre de la tecnología NX, la misma que mediante compresión busca mejorar el rendimiento del protocolo gráfico X de manera que se pueda transmitir incluso por enlaces de comunicación lentos.

El punto débil de las herramientas libres disponibles es quizás la falta integración en una solución global que facilite tanto la instalación y configuración iniciales, así como las posteriores tareas administrativas, además de la posibilidad de comprobar el funcionamiento y características disponibles sin necesidad de repetir todo el proceso en cada demostración.

2. OBJETIVOS

El objetivo general del presente proyecto será la creación de una prueba de concepto en formato LiveCD para la gestión de las conexiones de escritorios remotos utilizando como base FreeNX y que deberá servir para hacer demostraciones del funcionamiento general del software implicado sin tener que hacer preparaciones, instalaciones ni configuraciones de sistema previas.

El siguiente gráfico es un esquema conceptual de las funcionalidades que deberán implementarse en el *appliance*:



Como características principales tendremos las siguientes:

- Permitir la conexión mediante escritorio remoto al propio appliance utilizando el servidor FreeNX (prueba de concepto básica).
- Habilidad para hacer de proxy de los servicios de escritorio remoto soportados así como de hacer forwarding de sus conexiones.
- Habilidad para configurar el sistema vía WEB.
- Habilidad para publicar las conexiones disponibles vía WEB.
- Página de descargas de los clientes necesarios.
- Gestión de usuarios de las interfaces WEB.

Los objetivos específicos del proyecto serán:

- Desarrollo de un *appliance* en formato LiveCD, utilizando la herramienta SUSEStudio, con una instalación de FreeNX preconfigurada, soportando al menos los protocolos X11 y RDP.

- Desarrollar de una aplicación con interfaz WEB que incluya:
 - Gestión básica de usuarios y accesos
 - Configuración de servidores gráficos remotos y sus conexiones.
 - Publicación de las conexiones configuradas en forma de enlace o de instrucciones de configuración para los clientes
 - Descargar via WEB los diferentes clientes necesarios para la conexión a los escritorios remotos soportados.
 - Demostración para guiar a los usuarios en la configuración y utilización de los productos instalados. Debe servir como página de bienvenida, menú principal y/o demostración y debe llevar los logos de openTrends y de la UOC.
- Integrar la aplicación web en el appliance preparado con FreeNX, de tal manera que al arrancarlo ponga en marcha Firefox por defecto a pantalla completa, sin pestañas ni mensajes de inicio y con la interfaz WEB a modo de demostración como página de inicio.
- Realizar un estudio legal básico sobre las licencias del software incluido en el LiveCD con el objetivo de detectar incompatibilidades entre las licencias de los productos instalados.

3. ANALISIS DE REQUISITOS Y VIABILIDAD

Debido a que el presente es un proyecto principalmente de implantación y con requisitos claramente definidos incluso en cuanto a los programas a ser utilizados, el análisis de viabilidad de enfoca más en una revisión de las elecciones realizadas y su relación con los requerimientos a ser cumplidos.

3.1 Análisis de los requisitos

Los requisitos específicos del presente proyecto son los siguientes:

Desarrollo de un appliance en formato LiveCD. La utilización de un LiveCD para la presentación de las capacidades y características de una determinada solución es una tendencia entre los proveedores de software puesto que facilita el proceso al no requerir un equipo (servidor o PC) que sea reinstalado para la demostración, ni tiempo y esfuerzo para instalaciones.

Instalación y preconfiguración de los programas requeridos. Este requisito va de la mano del desarrollo del LiveCD y se enmarca en el deseo de minimizar los recursos y tiempo requeridos para la demostración, permitiendo inclusive su uso a personal con poco conocimiento técnico que no estaría en capacidad de realizar una instalación y configuración completa por su cuenta.

Permitir la conexión mediante escritorio remoto al propio appliance. La que se considera como la prueba de concepto básica puesto que bastará con arrancar iniciar el LiveCD e instalar el programa cliente en una PC cualquiera para que se pueda apreciar las funcionalidades del sistema de escritorio remoto.

Habilidad para hacer de proxy y forwarding de conexiones. El objetivo es poder utilizar el appliance para conectarse a sesiones gráficas remotas de otros servidores. Por ejemplo, si desde la pc de la casa se desea acceder remotamente a 2 servidores internos de la oficina, uno ejecutando Windows y otro ejecutando Linux, el appliance servirá de intermediario entre el cliente remoto y dichos servidores.

Interfaz administrativa vía WEB. La necesidad de una interfaz web para la administración del appliance es fundamental para conseguir el objetivo de tener demostraciones de la solución que no requieran procesos previos de instalación, por lo cual es necesario también que la interfaz web sea accesible tanto desde otros equipos, como desde la consola local del mismo appliance. Otro requisito de esta interfaz es que permita una gestión básica de control de acceso por usuarios.

Habilidad para configurar los servidores gráficos remotos y sus conexiones. La interfaz administrativa debe ser capaz de permitir la configuración de los servidores gráficos remotos, de manera que se facilite la demostración de la capacidad de *proxy* y *forwarding* hacia servidores propios del cliente que esté probando el sistema.

Habilidad para publicar las conexiones disponibles vía WEB. Otra de las funciones que

debe tener la interfaz administrativa es la posibilidad de visualizar las conexiones remotas configuradas o disponibles, así como proveer las facilidades en forma de enlace o de instrucciones de configuración para los clientes que las accederán.

Página de descargas de los clientes necesarios. Una página que permita descargar los diferentes clientes necesarios para la conexión a los escritorios remotos soportados es necesaria para evitar que durante el proceso de prueba el usuario deba navegar en Internet en busca del agente requerido para que su estación de trabajo pueda realizar las conexiones remotas.

Demostración para guiar a los usuarios en la configuración y utilización de los productos instalados. Como se ha mencionado en los objetivos del proyecto, este módulo de la interfaz web debe servir como página de bienvenida y permitiendo una rápida familiarización con el sistema, sus capacidades y modo de operación. Debe además dar acceso al menú principal de la aplicación.

Utilización de Software Libre. El objetivo principal del proyecto es permitir demostrar las capacidades de herramientas libres para escritorios remotos, por lo que la utilización de Software Libre es una necesidad. Este requerimiento será transversal para todos los productos usados o desarrollados. Habrá que tomar en cuenta también que haya compatibilidad entre las diferentes licencias utilizadas por lo que será necesario un apartado dedicado al análisis de las mismas.

3.2 Análisis de la Viabilidad

A efectos de estudiar la viabilidad para la implementación del presente proyecto, se han agrupado los requerimientos en 3 áreas que simplificarán el análisis.

3.2.1 Viabilidad para el Appliance LiveCD

En la actualidad existen varios métodos de generación de distribuciones Linux personalizadas, en general cada una permite modificar una distribución específica, por ejemplo *Reconstructor*¹ que permite personalizar distribuciones Debian y derivadas como Ubuntu, o *Revisor*² que a su vez se utiliza para crear distribuciones personalizadas basadas en Fedora. Sin embargo, para el presente proyecto se ha elegido *SUSE Studio*³, que como es de suponer, permite crear distribuciones de Linux basadas en diferentes versiones de OpenSuSE y SuSE Linux Enterprise.

Las principales características por las que se elige a SUSE Studio son:

- Es una herramienta web en la que todo el proceso se puede realizar sin necesidad de instalar aplicaciones adicionales en nuestro computador local.
- Facilita la creación de los appliances en varios formatos incluyendo LiveCDs y máquinas virtuales en formatos populares como Vmware y VirtualBox.

1 <http://build.reconstructor.org/>

2 <http://revisor.fedoraunity.org/>

3 <http://susestudio.com/>

- Permite control de versiones, con *changelog* incluido.
- Posee plantillas a partir de las cuales empezar nuevos proyectos.
- Permite la ejecución en línea del appliance creado (*TestDrive*), lo que permite verificar su funcionamiento antes de descargar el archivo generado.

3.2.2 Viabilidad para el Servicio de Escritorios Remotos.

La utilización de NX como tecnología de escritorio remoto está basada en su alto rendimiento para transmitir escritorios remotos completos a través incluso de enlaces lentos, lo que lo hace ideal para casi cualquier escenario de teletrabajo, soporte remoto, e-learning, etc.

La tecnología NX fue desarrollada por la compañía *NoMachine* y es parte fundamental en su producto propietario NX Server⁴, sin embargo, liberaron el protocolo NX bajo licencia GPL lo que permitió la aparición de FreeNX⁵, que es una implementación libre (GPL) de la misma tecnología. FreeNX es compatible con los productos de NoMachine al punto de que los programas clientes de esta última (que son gratuitos pero no abiertos) pueden conectarse con servidores FreeNX sin realizar ningún cambio.

En un servidor NX, las sesiones X11 son comprimidas y enviadas de manera segura utilizando el protocolo SSH hacia el cliente, donde éstas son descomprimidas y mostradas. También puede actuar de *proxy* (a través de un componente opcional) con el objetivo de comprimir otros protocolos como RDP y VNC, para que alcancen niveles de rendimiento similares a los que logra de manera nativa con X11.

3.2.3 Viabilidad para la Interface Web de Administración

Para la interface web, se debe utilizar herramientas libres que sean reconocidas, fáciles de instalar (de preferencia incluidas en los repositorios oficiales de la misma distribución) y cuyas características permitan el desarrollo de los requisitos especificados.

Para el caso del servidor web, se ha seleccionado *Apache2*⁶ por ser el aplicativo por defecto en la mayoría de las distribuciones debido que su amplia adopción en servidores de Internet, así como su expansibilidad de características a través de módulos.

En cuanto al sistema de Bases de Datos, al igual que en el caso anterior se utilizará la opción por defecto y más utilizada que es MySQL⁷L, que además se caracteriza por ser rápida y fácil de utilizar y administrar. Otro punto a tomar en cuenta es que el sistema SUSE Studio posee una opción para importar una base de datos MySQL, lo que facilita el desarrollo del appliance.

Para el desarrollo de la interfaz Web las opciones son PHP Y JSP, la primera es probablemente la más viable, sencilla y rápida de implementar, sin embargo, JSP se caracteriza por tener una mayor cantidad de librerías que le permiten realizar casi

4 <http://www.nomachine.com/products.php>

5 <http://freenx.berlios.de/>

6 http://projects.apache.org/projects/http_server.html

7 <http://www.mysql.com/>

cualquier tarea imaginable. La decisión final se tomará luego del análisis y diseño de la aplicación web, lo cual debe resultar en requisitos mucho más específicos a nivel de programación.

En caso de utilizar JSP, será necesaria la implantación de Tomcat⁸ como servidor de aplicaciones, el cual también es ampliamente utilizado en muchas distribuciones Linux y se integra con el Apache para proveer interfaces limpias y transparentes para el usuario final.

8 <http://tomcat.apache.org/>

4. DISEÑO DE LA SOLUCION PROPUESTA

4.1 Arquitectura de la solución

La arquitectura general de la solución propuesta se puede representar mediante 3 bloques principales: los clientes, el *appliance* y los escritorios remotos, tal como se ilustra en la siguiente figura:

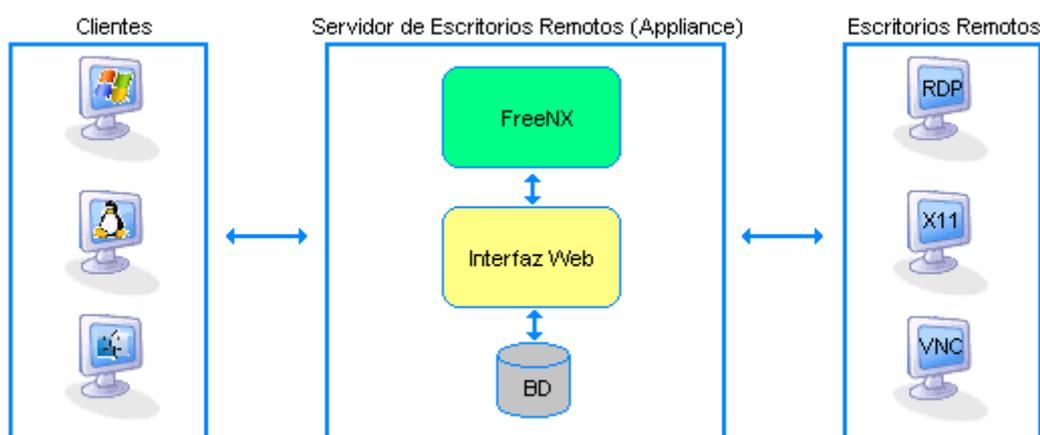


Figura 1. Arquitectura General de la solución propuesta.

El primer bloque representa a las computadoras o máquinas virtuales que puedan ejecutar el programa cliente del protocolo NX. Se recomienda usar el cliente gratuito (no libre) provisto por la compañía NoMachine, debido a su mayor desarrollo, sin embargo debe ser posible conectarse utilizando otros clientes incluyendo implementaciones libres.

El cliente de NoMachine está disponible para sistemas operativos Microsoft Windows, Linux, Solaris y Mac OSX, lo que facilita la utilización de la solución, ampliando la base de posibles usuarios.

Para efectos del desarrollo y pruebas del proyecto, se utilizarán al menos 2 máquinas virtuales ejecutando alguna versión de Microsoft Windows y alguna distribución de Linux.

El segundo bloque bosqueja los componentes del Servidores de Escritorios Remotos, que es el objetivo del presente proyecto. Se observa el bloque correspondiente a FreeNX, que es el corazón del sistema por ser el que implementa y provee la tecnología de compresión y transmisión de las sesiones gráficas. Este programa estará preconfigurado con opciones básicas genéricas de manera que los usuarios lo podrán utilizar tan pronto como el appliance termine de iniciarse, pudiendo ser posible en primera instancia conectarse a una sesión gráfica del mismo appliance.

En este mismo bloque también aparece la interfaz web y la base de datos, las cuales operarán en conjunto con el objetivo de proveer al usuario una interfaz amigable que le permitirá obtener información acerca del funcionamiento del sistema, así como configurar de manera gráfica y asistida conexiones temporales hacia otros servicios de escritorios remotos, con el fin de comprobar el funcionamiento con servidores reales propios del

usuario. Otras opciones que deberán estar disponibles en la interfaz web son la posibilidad de configurar el direccionamiento IP del appliance, así como tener un control básico de accesos por usuarios, donde cada usuario tendrá restricciones respecto a qué servicios remotos podrá acceder. También incluirá una página de descarga de los programas cliente.

Cabe recordar que el appliance será implementado en modalidad de Live-CD, basado en una distribución de Linux de 32 bits, lo que permitirá que sea ejecutado en casi cualquier computador disponible.

Por último, el tercer bloque ilustrado corresponde a los escritorios remotos que podrían ser accesibles a través del sistema. En primera instancia se dispondrá de soporte para conexiones remotas a servicios RDP (Rdesktop), VNC y X11 (sesiones gráficas en Linux/Unix). Dado que el alcance del proyecto abarca el desarrollo de un appliance con todo incluido, el escritorio remoto disponible por defecto será una sesión gráfica hacia él mismo. Se espera que los clientes deseen además comprobar el reenvío de las conexiones hacia sus propios servidores, para lo cual harán uso de las opciones de configuración incluidas en la interfaz web administrativa.

4.2 Arquitectura lógica de la Interfaz Web

A diferencia de los otros componentes, cuyas tareas son básicamente de instalación y configuración, la interfaz web administrativa es el único módulo que se desarrolla desde cero, por lo que requiere un análisis y diseño más detallado.

Partiendo de los requerimientos especificados en la sección de los objetivos, se puede diagramar la interfaz web de la siguiente manera:

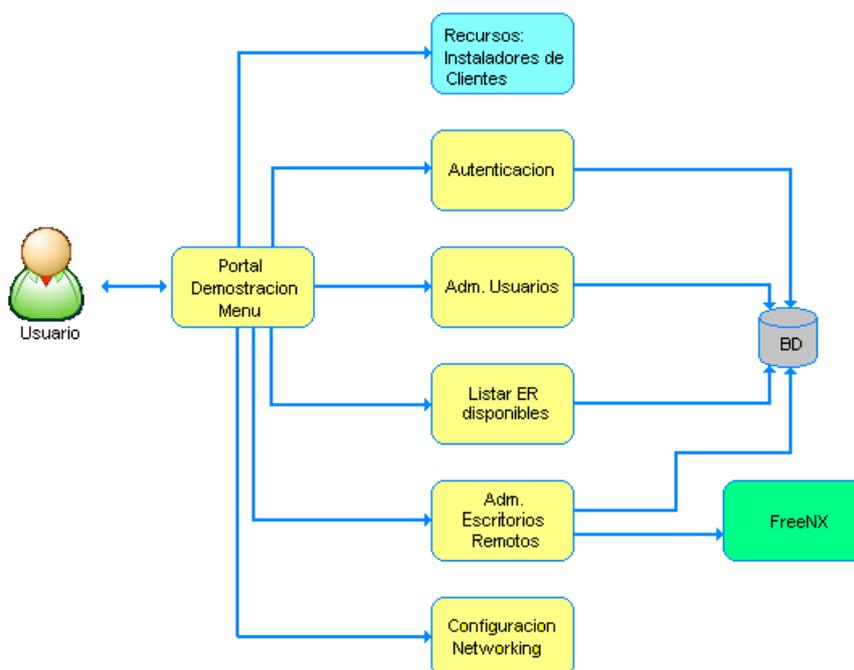


Figura 2: Arquitectura lógica de la interfaz web

4.2.1 Identificación de los subsistemas

Portal, será la página inicial del appliance e incluirá:

- Identificación del sistema y logos de los auspiciantes (UOC y OpenTrends)
- Demostración de funcionamiento del sistema. Un tutorial para permitir a los usuarios nuevos utilizar el aplicativo.
- Información de direccionamiento IP del appliance, para permitir conocer la dirección IP a la que se deben conectar los clientes, la misma que por defecto será obtenida mediante DHCP.
- Menú con las opciones habilitadas dependiendo del tipo de usuario activo: Si no ha iniciado sesión, estará disponible únicamente la página de las descargas. Si ha iniciado sesión un usuario regular, se habilitará la opción de listar los escritorios remotos disponibles para dicho usuario. Si el usuario es administrador, se habilitarán todas las opciones que incluyen además cambio de configuración IP, administración de usuarios y administración de escritorios remotos.
- Formulario de inicio de sesión.

Instaladores de Clientes, será una página web estática con enlaces para descargar los clientes de NX para los diferentes sistemas operativos.

Autenticación, será el módulo que reciba los requerimientos del formulario de inicio de sesión y realice las respectivas validaciones con la información almacenada en la base de datos.

Administración de Usuarios, módulo al que solo puede acceder el usuario administrador y que permitirá la creación, modificación y eliminación de los usuarios del aplicativo.

Listado de Escritorios Remotos Disponibles, permitirá a cada usuario visualizar la información de los escritorios remotos a los que el Administrador le ha dado acceso. Incluirá un enlace para descargar el archivo de configuración, así como instrucciones respecto a la configuración del programa cliente, de manera que el usuario pueda elegir entre las dos opciones.

Administración de Escritorios Remotos, módulo al que solo puede acceder el usuario administrador y que permitirá la creación, modificación y eliminación de los accesos a los diferentes escritorios remotos disponibles, así como la asignación de permisos de acceso para los usuarios. En los casos necesarios, este módulo deberá interactuar directamente con los archivos de configuración del *FreeNX*, para evitar que los usuarios deban editarlos manualmente.

Configuración de Networking, este subsistema, al que solo podrá acceder el usuario administrador, facilitará la configuración manual de la dirección IP del appliance en caso de que no haya sido posible obtener una dirección válida mediante DHCP.

Base de Datos. Como repositorio de datos se usará una base de datos sencilla, que ayudará a mantener la persistencia tanto de los datos de los usuarios, como de los escritorios remotos configurados y las relaciones entre ambas entidades, tal como se detalla en la siguiente figura:

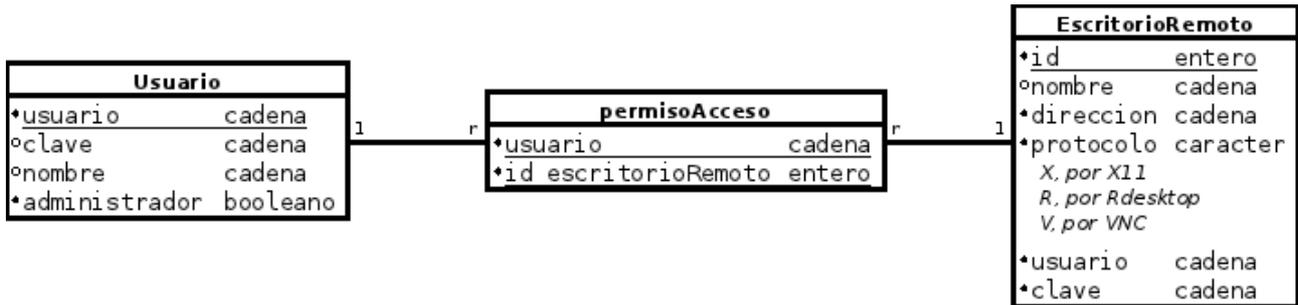


Figura 3: Diagrama de la Base de Datos

4.3 Casos de uso

Se han identificado 5 casos de uso que representan la mayoría de los escenarios posibles de uso de la interfaz web. Dado que el acceso a la interfaz web de manera local en la consola del mismo appliance o de manera remota desde una PC en la red debe dar el mismo resultado, no se ha considerado dicho factor para la elaboración de los casos de uso.

Caso de uso 1:

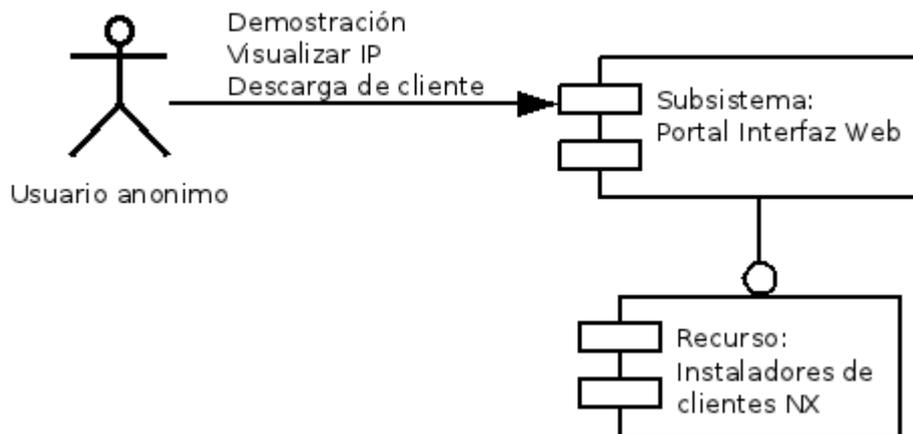


Figura 4: Caso de Uso 1 – Acceso a interfaz web sin identificación de usuario.

Caso de uso 2:

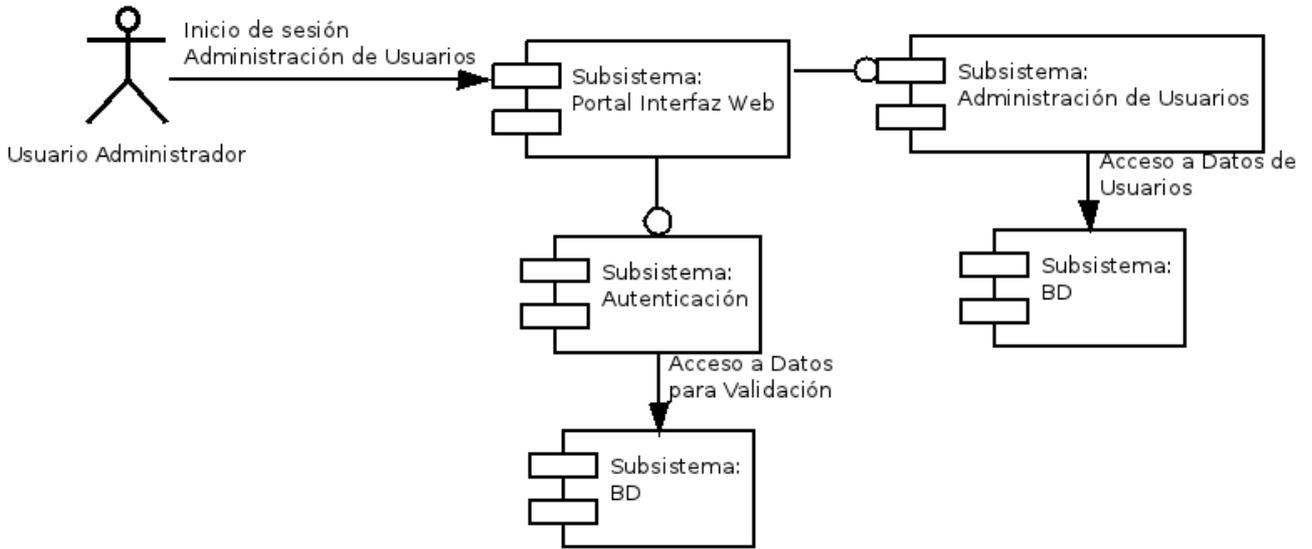


Figura 5: Caso de Uso 2 – Usuario Administrador accede a interfaz de administración de usuarios.

Caso de uso 3:

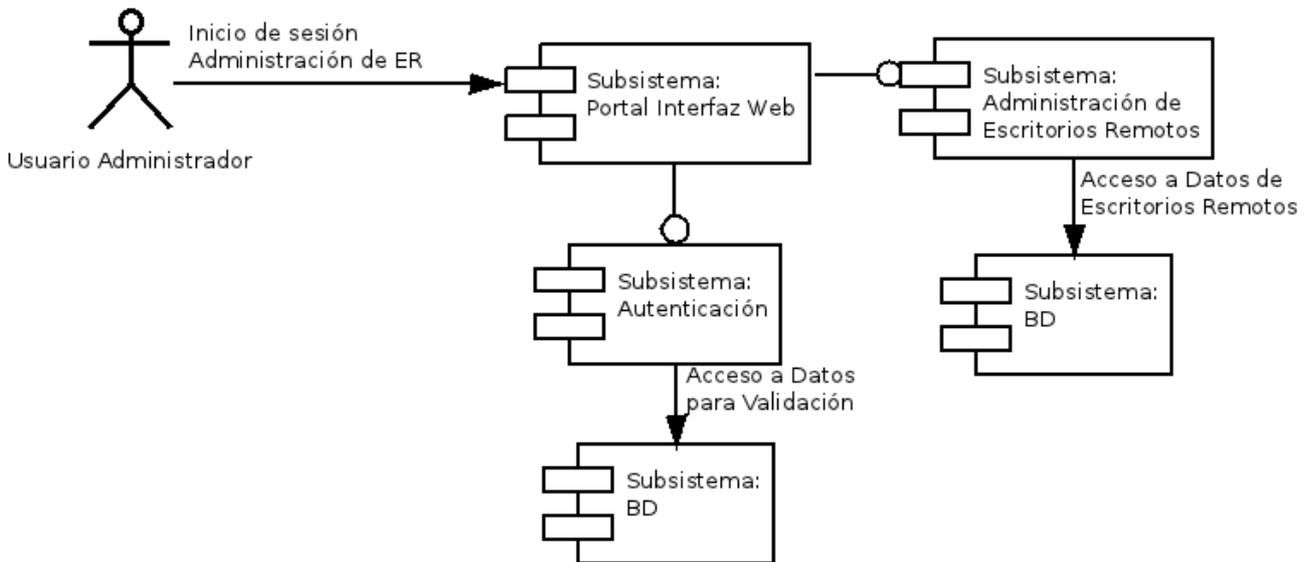


Figura 6: Caso de Uso 3 – Usuario Administrador accede a interfaz de administración de escritorios remotos.

Caso de uso 4:

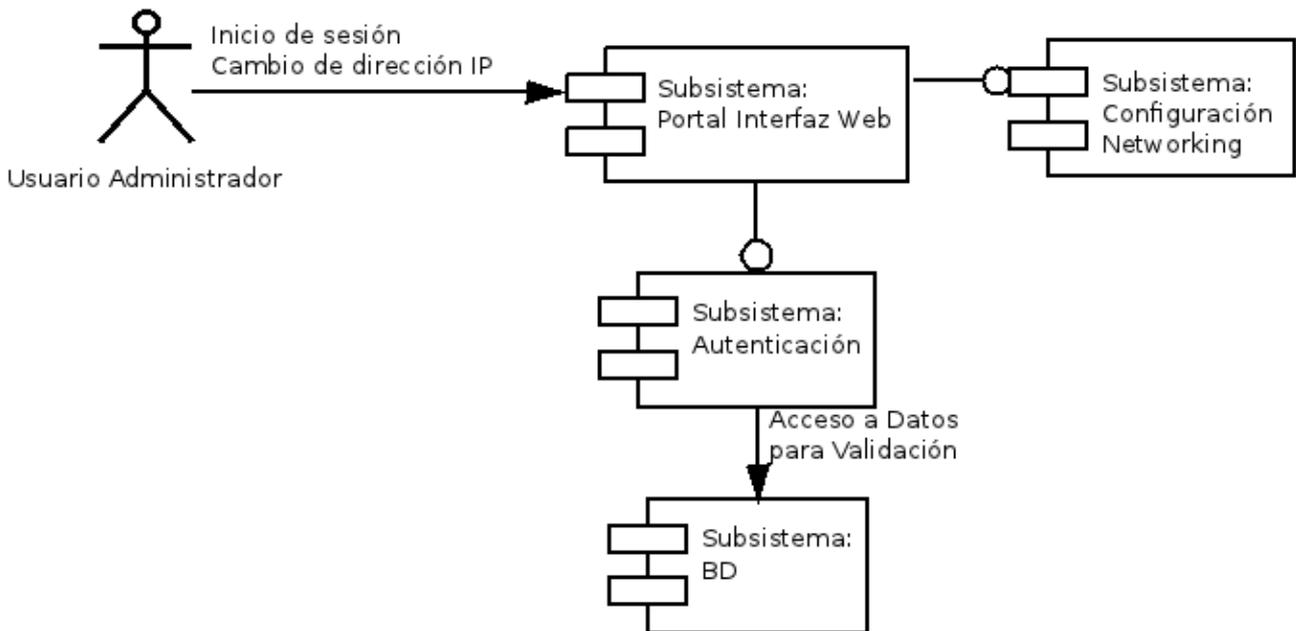


Figura 7: Caso de Uso 4 – Usuario Administrador accede a interfaz de configuración de dirección IP.

Caso de uso 5:

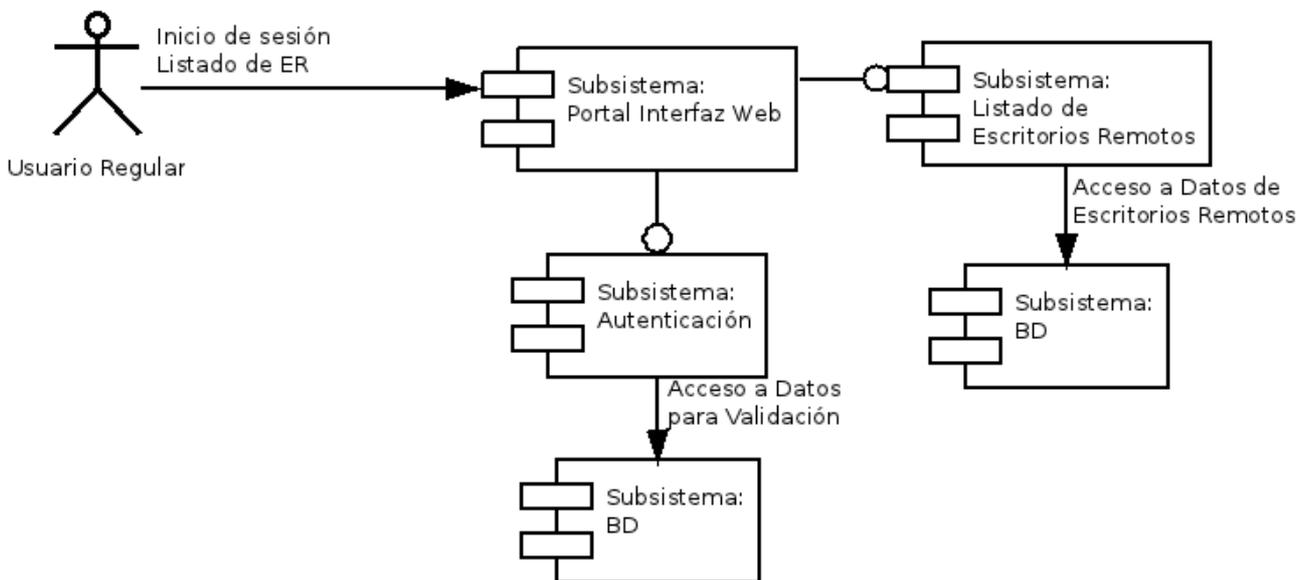


Figura 8: Caso de Uso 5 – Usuario Regular accede a listado de Escritorios remotos disponibles.

4.4 Especificaciones de Desarrollo y Pruebas

Las interfaces web serán desarrolladas utilizando el lenguaje JSP, debido a su robustez y versatilidad, mientras que para el despliegue de las mismas se utilizara Apache Tomcat 7 como servidor de aplicaciones. Como motor de Base de Datos se utilizará MySQL 5.5. En ambos casos son las versiones estándares disponibles en la distribución utilizada (OpenSUSE 12.1)

El entorno de desarrollo estará conformado por una PC corriendo la distribución de Linux Fedora 16 (la más reciente al momento del desarrollo, además de *Eclipse Indigo Service Release 2* como IDE. Así mismo se emplearán máquinas virtuales basadas en VirtualBox para las ejecuciones de prueba del appliance a medida que avanza el desarrollo.

Con respecto al conjunto de pruebas que se deben evaluar para confirmar el correcto funcionamiento del sistema tenemos las siguientes:

- Que el appliance inicie en modo kiosko, ejecutando el aplicativo web desarrollado, teniendo como página principal el portal con la demostración de uso del sistema.
- Que se pueda acceder remotamente (vía HTTP) al aplicativo web desde otra PC.
- Que los enlaces de descarga de los clientes NX sean válido y estén disponibles incluso si el usuario no ha iniciado sesión.
- Que se pueda conectar con un cliente NX a la sesión gráfica del appliance utilizando credenciales predefinidas.
- Que el usuario administrador pueda iniciar sesión con credenciales predefinidas y acceder al menú completo de opciones.
- Que el usuario administrador, con una sesión válida, pueda crear, modificar y eliminar otros usuarios.
- Que el usuario administrador, con una sesión válida, pueda crear, modificar y eliminar conexiones a otros servidores remotos.
- Que el usuario administrador, con una sesión válida, pueda establecer permisos de acceso entre los usuarios del sistema y los servidores remotos establecidos.
- Que el usuario administrador, con una sesión válida, pueda modificar la dirección IP del appliance y que éste sea accesible con la nueva dirección IP después del cambio.
- Que un usuario regular pueda iniciar sesión con las credenciales especificadas por el administrador y acceder al menú restringido, teniendo acceso únicamente al listado de escritorios remotos establecidos por el mismo administrador.

5. IMPLEMENTACION

Las tareas de implementación se dividen en 3 grupos correspondientes a la creación del appliance en modalidad Kiosko, la instalación y configuración del servicio FreeNX para los escritorios remotos y finalmente el desarrollo de las interfaces web y su integración en el appliance.

5.1 Creación del appliance en modalidad kiosko

Para esta tarea se ha utilizado el *SUSE Studio*, que es un portal web que permite construir appliances o distribuciones de Linux con configuraciones específicas para servir a un propósito particular. Este es un servicio provisto por Novell, Inc., por lo que el sistema operativo base para todos los appliances es alguna versión de openSUSE o SuSE Enterprise Linux.

El proceso de creación de un appliance es en realidad muy simple e intuitivo y ha requerido de los siguientes pasos básicos:

- 1) Creación del nuevo appliance, para lo cual se debe elegir el sistema operativo, una plantilla que define el software instalado por defecto, la arquitectura y el nombre del mismo. Para el presente proyecto se ha utilizado openSUSE 12.1 de 32 bits, con la plantilla de "Minimal X" y se lo ha nombrado "FreeNX-Kiosk"
- 2) El segundo paso consiste en la elección del software a ser instalado. Existe la opción de agregar repositorios adicionales para tener mayor flexibilidad respecto al software disponible. En la primera etapa del proyecto, además de la lista de programas preseleccionados de acuerdo a la plantilla, se agregó el navegador Mozilla Firefox, que está disponible en uno de los repositorios estándares (openSUSE 12.1 Updates)
- 3) Posteriormente existe una etapa de configuración del appliance, con variadas opciones que incluyen la configuración del idioma, puertos disponibles a través del firewall, configuración de la red, creación de usuarios del sistema, personalización de fondos de pantalla, selección de nivel de inicio, scripts para ser ejecutados en cada inicio del sistema, etc. Para la primera versión de nuestro appliance los cambios más relevantes han sido la creación de un usuario *kiosk*, que el sistema inicie en modo gráfico (level 5), que se inicie la sesión del usuario creado y se ejecute el navegador Mozilla Firefox.

Una vez creada esta versión del appliance, se procedió a descargarla y ejecutarla en una máquina virtual con VirtualBox. Lo que fue aprovechado para manualmente descargar e instalar el Add-on "R-kiosk"⁹, que permite ejecutar el Firefox en modo kiosko (sin menú ni barras de herramientas) Como página de inicio se utilizó de manera temporal un archivo HTML local que simplemente muestra un mensaje informativo. Luego se procedió a crear un archivo comprimido con la carpeta `/home/kiosk/.mozilla/firefox` y se lo subió junto con el archivo html hacia el OpenSuse, donde existe una opción para incluir archivos al

⁹ <https://addons.mozilla.org/en-US/firefox/addon/r-kiosk/>

appliance, permitiendo configurar la ubicación de los mismos así como sus permisos y propietarios.

Luego de estos cambios se procedió a clonar el appliance con el nuevo nombre "FreeNX-Kiosk-PrimeraEntrega", el cual fue compartido en el "Gallery" del SUSE Studio para poder ser evaluado como la primera entrega del proyecto.

5.2 Instalación y configuración del servicio FreeNX

La siguiente tarea consistió en modificar el anterior LiveCD de forma que queden instalados y pre-configurados los componentes del FreeNX, el software utilizado para la conexión a los servidores remotos. El mayor tiempo invertido fue en encontrar y configurar adecuadamente los repositorios que contengan el programa FreeNX. Dicho paquete fue encontrado en 2 repositorios: "X11:/RemoteDesktops" que es un repositorio oficial y el "homes:/please_try_again". Aunque el segundo ofrece una versión ligeramente más reciente, se optó por utilizar el primero por ser un repositorio más confiable.

Cabe resaltar que al momento de agregar el nuevo repositorio se presentaron inconvenientes como que el modo gráfico ya no se iniciaba. Después de investigar se encontró que algunos paquetes no estaban siendo incluidos y debieron ser seleccionados manualmente, entre ellos están: xorg-x11-libXext, xorg-x11, xorg-x11-libXrender. En una etapa posterior del proyecto se presentó un problema similar con el paquete "tar".

Además de la instalación del RPM de FreeNX, hace falta ejecutar un script que lo configura. Dicho script (nxsetup) presenta un error respecto a una carpeta inexistente ("/usr/lib/cups/backend/ipp") y aunque inicialmente la creaba manualmente, ahora se optó por instalar el paquete "cups" que es el que la incluye. El comando que debe ejecutarse para configurar el FreeNX es:

```
nxsetup --install --setup-nomachine-key
```

Con la instalación y configuración del FreeNX se procedió a generar una nueva versión del appliance, la misma que se utilizó para realizar las pruebas de funcionamiento del mismo, conectando desde una PC (virtual) Windows el cliente de NoMachine hacia la sesión gráfica del usuario "kiosk", que es el que por defecto ejecuta firefox en modo kiosko. También se realizó pruebas de redireccionamiento utilizando el cliente de NoMachine instalado en una PC Linux, para conectar a una sesión RDP en una máquina virtual Windows XP, utilizando el servidor FreeNX como proxy.

5.3 Desarrollo de las interfaces web e integración en el appliance

El siguiente paso en la implementación del proyecto consistió en la preparación del entorno de ejecución para la aplicación web, comenzando con la instalación de MySQL y Tomcat, tarea que no presentó mayores inconvenientes puesto que se tratan de aplicaciones comunes disponibles en los repositorios estándares, por lo que solo hubo necesidad de seleccionarlos.

A partir de esta instalación se realizó una prueba de publicación de una página JSP de ejemplo que verifica la conectividad a una Base de datos MySQL creada previamente, la cual también resultó satisfactoria de manera inmediata.

Con respecto al “*desarrollo del aplicativo web*”, de acuerdo con el diseño establecido, incluye las siguientes páginas JSP:

- “cabecera.jsp”: Se encarga de mostrar el nombre del proyecto y los logos tanto de la UOC como de OpenTrends. También incluye el formulario de “login” o el nombre del usuario que ya ha iniciado sesión. Finalmente mostrará una línea con la dirección IP actual del servidor (que será útil tomando en cuenta que el appliance se configurará vía DHCP y mostrará el aplicativo en modo pantalla completa), además de un botón para “Apagar” el appliance. Cabe resaltar que el formulario de login a su vez llamará a la clase “login” que se encargará de validar las credenciales y en caso de autenticación positiva guardará los datos del usuario como atributos de la sesión, lo que permite luego la activación de las opciones del menú y el acceso a las interfaces de acuerdo al tipo de usuario logueado.

- “menu.jsp”: Muestra un menú horizontal con las opciones disponibles de acuerdo con el nivel de usuario:

“Inicio” -> Disponible siempre.

“Descargas” -> Disponible siempre.

“Escritorios Remotos” -> Disponible para cualquier usuario que inicie sesión.

“Usuarios” -> Disponible únicamente para el administrador.

“Configuración” -> Disponible únicamente para el administrador.

Las páginas JSP para cada interfaz son:

- “index.jsp”: Es la página inicial y mostrará instrucciones demostrativas sobre el uso del appliance.

- “descargas.jsp”: Muestra una tabla con los enlaces para descargar el instalador del cliente NX para distintas plataformas.

- “escritorios.jsp”: Para el caso de usuarios regulares, esta página muestra el listado de escritorios remotos asignados al usuario logueado, visualizando los datos de conexión, así como permitiendo la descarga del archivo de configuración del cliente NX para facilitar las pruebas. En el caso del usuario administrador, se muestran todos los escritorios remotos configurados, además de la posibilidad de editarlos, eliminarlos o agregar escritorios nuevos.

- “usuarios.jsp”: Esta interfaz que solo está disponible para el usuario administrador, permite la creación, modificación y eliminación de usuarios regulares para la aplicación web, así como la asignación de los escritorios remotos disponibles para cada uno de ellos.

- “configuracion.jsp”: Esta interfaz que solo está disponible para el usuario administrador, permite la configuración manual de la interfaz de red. Es útil en casos en que el appliance no haya podido conseguir dirección IP mediante DHCP. Se hace uso de scripts de shell,

ejecutados mediante “sudo” para realizar éstas configuraciones.

Finalmente, el proceso de integrar las interfaces WEB al appliance,“ básicamente consistió de 3 subtareas:

- La generación de un respaldo de la base de datos que fue subida al SuseStudio, que tiene una opción específica para importar los datos en MySQL.
- La generación de un archivo WAR con el proyecto desarrollado, el mismo que se lo incluyó en la carpeta “/src/tomcat7/webapps” para que sea desplegado en el momento de iniciar el servicio Tomcat
- La inclusión de archivos adicionales que soportan la aplicación web, como los scripts para obtener o modificar la configuración de la red, las plantillas para los archivos de configuración del cliente NX, el cambio en el sudo para permitir al Tomcat la ejecución de ciertos comando privilegiados y el cambio en el script “post-inicio” para ejecutar el servicio del Tomcat.

6. RESULTADOS

Los resultados tangibles de las pruebas durante las diferentes etapas del desarrollo del proyecto han sido:

- Se cumplió con el primer entregable (LiveCD con Firefox a pantalla completa al inicio), lo que a su vez permitió la familiarización con el entorno de trabajo de SuseStudio.
- Se confirmó la viabilidad del proyecto al comprobarse el funcionamiento del software FreeNX en la plataforma operativa elegida.
- Se configuró la base del sistema requerido para proceder con la etapa de desarrollo de una interfaz web para la administración de los escritorios remotos.
- Se diseñó e implementó la base de datos requerida y se desarrolló el aplicativo con las interfaces especificadas.
- Se integró el aplicativo en el appliance y se comprobó el funcionamiento de las opciones desarrolladas hasta el momento.



Figura 9: Pantalla inicial del appliance ejecutado en máquina virtual.

- Se pueda acceder al aplicativo tanto desde la consola local como remotamente (vía HTTP) desde otra PC.
- Los enlaces de descarga de los clientes NX son válidos y pueden ser descargados desde el appliance incluso si el usuario no ha iniciado sesión.
- Se comprobó el acceso del usuario con privilegios de administrador (admin/admin), del usuario regular (test/test) y la variación del menú según el tipo de sesión activa.



Figura 10: Menú con sesión no iniciada.



Figura 11: Menú para un usuario regular.



Figura 12: Menú para el usuario administrador.

- Se comprobó que el usuario administrador, con una sesión válida, puede crear/modificar/eliminar otros usuarios, crear/modificar/eliminar conexiones a servidores remotos y modificar la dirección IP del appliance y que éste sea accesible con la nueva dirección IP después del cambio.

FreeNX Kiosk
Patrocinado por: openTrends UOC Universitat Oberta de Catalunya

Administrador
Usuario activo
Cerrar sesión

Dirección IP Actual: 192.168.168.3
Apagar

INICIO DESCARGAS ESCRITORIOS REMOTOS USUARIOS CONFIGURACIÓN

Lista de Escritorios Remotos

Agregar Escritorio

Nombre	Dirección	Protocolo	Usuario	Clave	Options
WindowsXP	192.168.168.6	RDesktop	Usuario	usuario	✎ ✖ Archivo
FedoraEdwin	192.168.168.2	VNC	eboza	eboza	✎ ✖ Archivo

Figura 13: Pantalla de administración de escritorios remotos.

INICIO DESCARGAS ESCRITORIOS REMOTOS USUARIOS CONFIGURACIÓN

Lista de Usuarios

Agregar Usuario

Nombre	Usuario	Options
Administrador	admin	✎
Test	test	✎ ✖

Figura 14: Pantalla de administración de usuarios.

INICIO DESCARGAS ESCRITORIOS REMOTOS USUARIOS CONFIGURACIÓN

Configuración IP Actual:

Dirección IP / Máscara de red: 192.168.168.3/ 24
Puerta de Enlace: 192.168.168.254

Nueva Configuración IP:

Dirección IP / Máscara de red: /
Puerta de Enlace:

Guardar

Figura 15: Pantalla de administración de configuración de red.

- Se comprobó que un usuario regular puede iniciar sesión con las credenciales especificadas por el administrador y acceder al menú restringido, teniendo acceso únicamente al listado de escritorios remotos establecidos por el mismo administrador y descargarse el respectivo archivo de configuración para el cliente NX en cada caso.

- Se pueda conectar con un cliente NX a la sesión gráfica (X) del appliance utilizando credenciales predefinidas (test1/test1)

Durante la etapa de pruebas se recibieron comentarios de parte del cliente, respecto a detalles a pulir, los cuales fueron acogidos:

- En la cabecera de la aplicación web, debajo del logo, no solo se muestra la dirección IP del appliance sino que se especifica la URL de acceso para que pueda ser abierta desde otra PC.

- A la hora de editar las configuraciones de conexiones remotas, se ha agregado una validación en el nombre de la conexión para no permitir caracteres especiales que no sean luego aceptados por el cliente NX.

- Se corrigió la lista de conexiones remotas para evitar que un usuario no autorizado se descargue el archivo de configuración al que no tenga acceso.

7. VALORACION ECONOMICA

El presente proyecto, por tratarse exclusivamente de compilación y desarrollo de un producto de software, ha tenido una estructura de costos simplificada, limitada principalmente a la valoración del tiempo requerido para su implementación.

Se debe resaltar que las actuales tecnologías de virtualización ayudan a reducir los costos en equipos, pues de lo contrario hubiese sido necesario contar con al menos 1 equipo de computación adicional para la realización de las pruebas de funcionamiento.

Para la valoración del tiempo dedicado al desarrollo de proyecto se ha tomado como referencia el precio promedio ofertado por empresas locales (Guayaquil, Ecuador) para servicios de desarrollo de sistemas que es de USD\$30,00

A partir de los datos anteriores, el costo estimado del proyecto sería:

Concepto	Cantidad	Valor Unitario	Valor Total
Horas de trabajo	91	30,00	2,730,00
Varios (estimado)		100,00	100,00
<i>TOTAL</i>		USD\$	2,830,00

8. ANALISIS DE LICENCIAS

Un punto importante a ser tomado en consideración al momento de realizar un proyecto de software libre es el tema de las licencias. La licencia del software es el contrato entre el autor o apoderado de los derechos del programa y en usuario de mismo, que determina las facultades y restricciones que tiene éste último respecto al producto licenciado.

Existen ciertas licencias que restringen los trabajos derivados que se pueden realizar, así como la integración en productos nuevos, llegando incluso a causar incompatibilidades entre diferentes componentes de un proyecto mayor, por lo que es necesario realizar un análisis para determinar si existe algún inconveniente antes de publicar el proyecto.

El proyecto del *FreeNX-Kiosk*, es un proyecto de tipo híbrido puesto que consta de una parte de desarrollo de software nuevo (la interfaz web) y una parte de integración de programas ya desarrollados. Por este motivo se ha realizado el análisis básico de compatibilidad de las licencias de los diferentes componentes.

De acuerdo a la información provista en los paquetes RPM instalados, se ha podido establecer que las licencias de los principales componentes integrados en el proyecto son:

Componente	Licencia
OpenSuse 12.1	GNU GPL v2
FreeNX	GPL v2 or later
MySQL	GPLv2
Firefox	MPL-1.1 o GPL-2.0+ o LGPL-2.1+
Tomcat	Apache License, Version 2.0
R-Kiosk (plugin firefox)	<p>EULA: “<i>R-Kiosk extension is in public domain. It may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise used by anyone for commercial or non-commercial purposes.</i></p> <p><i>R-Kiosk extension is supplied as is. You use this extension at your own risk.</i>”</p>

Como se puede observar, tanto el sistema operativo OpenSuse, como 3 de los principales componentes están protegidos por la licencia *GNU GPL v2*¹⁰, por lo que se establece la compatibilidad entre ellos. De la misma manera, el plugin R-Kiosk viene con un acuerdo de licencia (EULA) indicando que es de *dominio* público y que específicamente señala que puede ser libremente distribuido para fines comerciales o no, por lo que no existe problema en ser incorporado en el presente proyecto.

Resta únicamente verificar la compatibilidad de la licencia *Apache License, Version 2.0*¹¹ del Tomcat. Dado que el tema de la compatibilidad de las licencias es frecuente y puede

10 <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>

11 <http://www.apache.org/licenses/LICENSE-2.0>

ser difícil de interpretar, la Free Software Foundation, creadora de la *Licencia Pública General de GNU* (GNU GPL) que es a su vez la licencia de software libre más utilizada, ha creado un documento público donde se listan otras licencias populares y se indica su compatibilidad con la licencia GNU GPL, dicho documento se encuentra en la dirección: <http://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses>

De acuerdo con éste documento, la licencia "*Apache License, Version 2.0*" es incompatible con la licencia GNU GPL versión 2, sin embargo, esta incompatibilidad en las licencias se hace efectiva únicamente cuando se desea enlazar o combinar el código de programas diferentes en uno solo, lo cual no es el caso del presente proyecto, donde todos los componentes básicamente se ejecutan lado a lado pero sin ser modificados o peor mezclado su código.

9. CONCLUSIONES

La conclusión obvia del proyecto es que la disponibilidad de software libre facilita no solo el mejoramiento de los productos disponibles, sino también la compilación de los mismos para la creación de nuevos productos. En el caso del presente proyecto, se ha utilizado programas libres como el FreeNX, MySQL, Tomcat, Firefox, entre otros y se ha desarrollado una interfaz web nueva para crear un appliance que facilita la demostración de las capacidades de conexiones a escritorios remotos disponibles en entornos libres.

También ha sido posible constatar dichas capacidades al momento de configurar el servicio FreeNX para integrarlo en el producto terminado, quedando demostradas las posibilidades de conexión remota no solo hacia el servidor Linux, sino redireccionando dichas conexiones hacia otros servidores con los servicios conocidos RDP y VNC.

La utilización de la tecnología NX, en este caso mediante FreeNX, no solo que mejora el rendimiento de las conexiones remotas a sesiones de trabajo gráficas, sino que además permite la creación de un punto de entrada en la red para dichas conexiones, asegurándolas mediante el protocolo SSH. De esta manera, se evita tener que configurar el firewall perimetral de la red para abrir una conexión, potencialmente peligrosa, por cada servicio disponible en la red interna que se quiera poner a disposición de los usuarios remotos.

El uso de un aplicativo web para centralizar la administración del appliance ayuda en el sentido de que facilita la comprensión de su funcionamiento y se puede simplificar el proceso de configuración y pruebas del sistema. Además, que se agrega un nivel de seguridad al limitar el acceso a la información de conexión a escritorios remotos dependiendo del usuario que ha iniciado sesión.

Sin embargo, un paso adicional que se puede implementar es integrar aún más el aplicativo web con el sistema FreeNX para que las restricciones de acceso no solo sean a los datos de configuración del cliente, sino que exista la restricción al momento de establecer la conexión remota, de manera que una tercera persona, aún conociendo los datos de conexión, no pueda conectarse si no conoce la clave del usuario creada en la interfaz web.

Con respecto a las herramientas de desarrollo, sin lugar a dudas, el uso del portal SuseStudio para la generación de la distribución es una ayuda invaluable, no solo que evita la necesidad de aprender acerca del proceso interno de creación de una distribución de Linux, también evita tener que invertir en la infraestructura para dichas tareas y todo esto lo hace mediante una interfaz amigable, intuitiva y flexible con una variedad de opciones suficientes para que pueda ser utilizada tanto por usuarios novatos como por desarrolladores más avanzados con mayores requerimientos.

Otra de las tecnologías utilizadas en este proyecto ha sido la virtualización mediante VirtualBox, que ha facilitado las labores de prueba y desarrollo del producto, eliminando la necesidad de tener equipos de computación adicionales para estas tareas. La posibilidad de ejecutar, comparar y modificar varias instancias simultáneas de la misma o diferentes versiones del producto, limitado únicamente por los recursos disponibles en la estación de trabajo real, es en definitiva un avance enorme para entornos de desarrollo disponible

desde hace pocos años.