



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 10

“Personalización de distribuciones basadas en la familia SUSE Linux.”

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autor: Roilán Zaballa Coca.

Tutores: Israel Gonzales Delgado.

Yosley Bello Rodríguez.

Asesor: Pedro Carlos Pérez Martinto.

Ciudad de la Habana, junio, 2009

“Año 50 de la Revolución”

Datos de Contacto

Lic. Israel González Delgado: Investigador Auxiliar, graduado de licenciatura en Ciencias de la Computación en el año 1996 en la Universidad de la Habana. Ha trabajado en el tema de redes y Sistemas Operativos de la familia UNIX y en el desarrollo de DRIVERS y aplicaciones, además cuenta con una experiencia de 12 años en el tema.

Ing. Yosley Bello Rodríguez

Oficial de Automatización. Graduado de Ingeniería en Ciencias Informáticas (2008) en la Universidad de las Ciencias Informáticas. Ha trabajado en el tema de los sistemas operativos Linux con 3 años de experiencia.

Agradecimientos

En primer lugar quiero agradecerles a todas las personas que me ayudaran a lo largo de la carrera, a mis compañeros y amigos, a Yenner, Annis y el profesor Martinto que me ayudaran cantidad para poder realizar este trabajo, a mis tutores que siempre me auxiliaron frente a cualquier duda u obstáculo.

A toda mi familia por el apoyo que siempre me han brindado, a Fefa y Raúl que han luchado conmigo estos cinco años aunque todavía le queda más tiempo, a mis abuelos, a mi tía Marina, y muy especial a mi mamá que es la persona que más se ha sacrificado por mí, a mi hermana que le sirva de ejemplo, a mi novia por su paciencia, amor y confianza y a Jorge que es mi padre y mi amigo, que sin dudas es la persona que más me ha enseñado y guiado en el camino de la vida para ser lo que soy hoy y poder hacer este sueño realidad.

Dedicatoria

Este trabajo se lo dedico a mi familia, a mi mamá, a mi hermana, a mi novia, a Jorge y a todas esas personas que me han ayudado a lo largo de la carrera.

Resumen

En la actualidad el Ministerio del Interior (MININT) se encuentra enfrascado en el desarrollo de disímiles aplicaciones sobre plataformas de sistemas operativos libres como GNU/Linux, para ello utiliza las distribuciones de la familia SUSE Linux debido a su estabilidad para desarrollar y funcionamiento como servidor de base de datos con el trabajo de gestores como Oracle.

Sucede que estas distribuciones traen incorporados software que no son necesarios para el funcionamiento y rendimiento de las aplicaciones que se están creando, y por este motivo es que se requiere de una “Personalización de distribuciones basadas en la familia SUSE Linux”.

Es por ello que en este trabajo se desarrolló un sistema web para crear imágenes personalizadas de las distribuciones basadas en SUSE Linux la cual permitirá que estas tengan los servicios, aplicaciones, funcionalidades y software que realmente sean imperiosos para el trabajo de las herramientas y sistemas que el MININT despliega.

La aplicación se integra con la herramienta llamada KIWI que se encuentra instalada en el servidor mediante un archivo XML (config.xml), en el cual es guardada toda la información referente a lo que el usuario desea que tenga su sistema personalizado, a partir de dicha información es que se realiza el proceso de personalización, el cual consta de dos pasos: preparar y crear. Una vez terminado estos pasos se podrá obtener un sistema personalizado listo para usar.

Índice

Introducción	1
Capítulo I. Fundamentación Teórica.	4
1.1 Introducción.	4
1.2 Software Libre y GNU/Linux.	4
1.2.1 ¿Qué es GNU/Linux?	5
1.2.2 Distribuciones de GNU/Linux.	5
1.2.3 ¿Qué distribución utilizar?.....	6
1.3 ¿Por qué utilizar SUSE como punto de partida?.....	7
1.4 Principales Características de SUSE.	8
1.4.1 YaST.	8
1.4.2 Formato de Paquetes utilizados en SUSE.	9
1.4.3 Repositorios de SUSE.....	10
1.5 Herramientas que permiten personalizar distribuciones de la familia SUSE.	10
1.5.1 KIWI.	10
1.5.2 Creador de Imágenes.....	11
1.5.3 SUSE Studio.	12
1.6 Metodologías de desarrollo de software.	12
1.6.1 Extreme Programming (XP).....	13
1.7 Herramientas y lenguajes a utilizar.....	13

1.7.1 Servidores Web.	14
1.7.1 Apache.....	14
1.7.2 Intérpretes de comando.	15
1.7.3 Gestores de base de datos (SGBD).	16
1.7.4 Sistemas para administrar SGBD.	17
1.7.5 Lenguajes de programación Web.	18
1.7.6 Herramientas para el desarrollo web.	22
1.7.7 Framework de Desarrollo.	23
1.8 Conclusiones.	25
Capítulo II Planificación.....	26
2.1 Introducción.....	26
2.2 Fase de planificación.	26
2.2.1 Alcance.	26
2.2.2 Prioridad.	26
2.3 Historias de Usuarios.	27
2.4 Plan de Iteraciones.	32
2.4.1 Plan de duración de las iteraciones.	33
2.5 Plan de entregas.	34
2.6 Conclusiones.	34
Capítulo III Fase de Diseño.	36

3.1 Introducción.	36
3.2 Metáfora para el sistema.	36
3.2.1 Definición de términos.	36
3.2.2 Tarjetas CRC (Cargo o clase, Responsabilidad y Colaboración).	36
3.3 Soluciones Puntuales.	38
3.4 Fase Implementación.	38
3.5 Tareas de Ingeniería.....	39
3.5.1 Iteración 1.	39
3.5 Conclusiones.	44
Capítulo IV Pruebas de la Herramienta.	46
4.1 Introducción.	46
4.2 Disponibilidad del cliente.	46
4.3 Pruebas.	47
4.3.1 Pruebas de Aceptación.	47
4.4 Conclusiones.	57
Conclusiones Generales	58
Recomendaciones.....	59
Referencias Bibliográficas	60
Bibliografía	61

Índice de Tablas

Tabla 1. Historia de Usuario - Gestionar Usuario.	27
Tabla 2. Historia de Usuario - Autenticar Usuario.....	27
Tabla 3. Historia de Usuario - Registrar Usuario.	28
Tabla 4. Historia de Usuario - Gestionar Contraseña.....	28
Tabla 5. Historia de Usuario - Preparar Personalización.	29
Tabla 6. Historia de Usuario - Agregar o Quitar programas.....	29
Tabla 7. Historia de Usuario - Crear Personalización.	30
Tabla 8. Historia de Usuario - Descargar ISO.....	30
Tabla 9. Historia de Usuario - Subir Paquete.....	31
Tabla 10. Historia de Usuario - Listar mis proyectos.	31
Tabla 11. Historia de Usuario - Gestionar Repositorios.	32
Tabla 12. Plan de duración de iteraciones.	33
Tabla 13. Plan de entregas.	34
Tabla 14. Tarjetas CRC - Clase Autenticación.....	37
Tabla 15. Tarjetas CRC - Clase Administración.	37
Tabla 16. Tarjetas CRC - Clase Preparar Personalización.....	38
Tabla 17. Distribución de Historias de Usuario para la Iteración 1.....	39

Tabla 18. Tarea de Ingeniería Preparar Personalización.....	40
Tabla 19. Tarea de Ingeniería Agregar o Quitar Programas.	40
Tabla 20. Tarea de Ingeniería Listar repositorios disponibles.	40
Tabla 21. Tarea de Ingeniería Crear personalización.	41
Tabla 22. Tarea de Ingeniería Descargar ISO.	41
Tabla 23. Tarea de Ingeniería Subir paquete.....	42
Tabla 24. Distribución de Historias de Usuario para la Iteración 2.....	42
Tabla 25. Tarea de Ingeniería Estudiar la librería FreakAuth_light.	42
Tabla 26. Tarea de Ingeniería Instalar y configurar la librería FreakAuth_light.	43
Tabla 27. Distribución de Historias de Usuario para la Iteración 3.....	43
Tabla 28. Tarea de Ingeniería Listar mis proyectos.	43
Tabla 29. Tarea de Ingeniería Gestionar Repositorios.....	44
Tabla 30. HU1_P1 Prueba de Aceptación Gestionar usuario.....	48
Tabla 31. HU1_P2 Prueba de Aceptación Gestionar usuario.....	48
Tabla 32. HU1_P3 Prueba de Aceptación Gestionar usuario.....	49
Tabla 33. HU2_P1 Prueba de Aceptación Autenticar usuario.	49
Tabla 34. HU2_P2 Prueba de Aceptación Autenticar usuario.	50
Tabla 35. HU3_P1 Prueba de Aceptación Registrar usuario.....	50
Tabla 36. HU3_P2 Prueba de Aceptación Registrar usuario.....	51
Tabla 37. HU4_P1 Prueba de Aceptación Gestionar contraseña.	51

Tabla 38. HU4_P2 Prueba de Aceptación Gestionar contraseña.	52
Tabla 39. HU5_P1 Prueba de Aceptación Agregar o Quitar programas.	52
Tabla 40. HU5_P2 Prueba de Aceptación Agregar o Quitar programas.	53
Tabla 41. HU5_P3 Prueba de Aceptación Agregar o Quitar programas.	53
Tabla 42. HU6_P1 Prueba de Aceptación Crear Personalización.	54
Tabla 43. HU7_P1 Prueba de Aceptación Descargar ISO.	54
Tabla 44. HU8_P1 Prueba de Aceptación Subir paquete.	55
Tabla 45. HU9_P1 Prueba de Aceptación Listar mis plantillas.	55
Tabla 46. HU10_P1 Prueba de Aceptación Gestionar repositorios.	56
Tabla 47. HU10_P2 Prueba de Aceptación Gestionar repositorios.	56

Introducción

Con el inicio del siglo XXI comenzó un desarrollo acelerado de las tecnologías, con lo cual se han logrado grandes avances para la evolución y beneficio de la humanidad, uno de estos adelantos es la producción ascendente de computadoras, donde cada día se incorporan más funcionalidades, con el objetivo de mejorar la relación entre el usuario y las mismas, lo que ha provocado que la elaboración de hardware y software se expandan vertiginosamente.

Dentro de la creación de software ha tenido gran impacto en los últimos años los de código abierto gracias a las libertades que nos brindan, a causa de eso ha surgido una gran diversidad en los sistemas operativos basados en GNU/Linux¹. Los beneficios derivados de estos sistemas son numerosos pero se caracterizan fundamentalmente por la potencialidad y flexibilidad para ejecutar multitarea y multiusuario, Shell² programables y detección de hardware.

En el mundo existen varias distribuciones basadas en GNU/Linux, una de ellas es la de origen alemán "Software und Systementwicklung" (SUSE) o, Desarrollo de Sistemas y de Software, en español. De esta ha nacido una familia debido a que han surgido varios productos destinados a un uso específico como son: SUSE Linux Enterprise Server (SLES), SUSE Linux Enterprise Desktop y openSUSE.

Estas distribuciones no solo se destacan por ser una de las más sencillas de instalar y administrar, debido a que cuenta con varios asistentes gráficos para completar diversas tareas, en especial por su gran herramienta de instalación y configuración YaST (Yet another Setup Tool), sino también que presentan gran estabilidad y rendimiento en el uso de servidores de base de datos como es el caso de el gestor de base de datos Oracle.

En Cuba existen pocas empresas u órganos que utilicen sistemas operativos de la familia SUSE, uno de ellos es el Ministerio del Interior (MININT), el cual ha venido desarrollando una serie de aplicaciones sobre

¹ Acrónimo recursivo que significa GNU No es Unix (*GNU is Not Unix*).

² Cualquier programa que el usuario utilice para ejecutar comandos.

sistemas operativos basados en SUSE Linux, pero sucede que muchas de estas herramientas creadas por el MININT no necesitan todos los servicios o funcionalidades que vienen incorporadas en los sistemas operativos de la familia SUSE Linux, por lo que no se cuenta con una herramienta que permita la personalización de distribuciones de la familia SUSE que contenga solo los servicios necesarios.

Por tal motivo se plantea como **problema a resolver**: ¿Cómo personalizar distribuciones de la familia SUSE para contribuir a un elevado nivel y ajuste a los intereses del MININT?

Como **Objeto de Estudio**: Proceso de desarrollo de software orientado a los sistemas base para ser aplicado en las instituciones del MININT.

Objetivo General: Desarrollo de una herramienta que permita la personalización de los sistemas bases de la familia SUSE.

Campo de Acción: Proceso de desarrollo de herramientas WEB para la personalización de sistemas base basados en SUSE dirigidos al MININT.

Tareas Investigativas:

- Realizar un estudio de las principales características de los sistemas operativos de la familia SUSE Linux.
- Analizar las aplicaciones existentes que permitan la creación de distribuciones personalizadas basadas en SUSE Linux.
- Seleccionar una metodología adecuada que permita guiar el proceso de desarrollo.
- Estudio de las herramientas y tecnologías a utilizar para el desarrollo del sistema informático.

Entre los **métodos científicos** a utilizar, de acuerdo a las diferentes etapas de las tareas a desarrollar se encuentran:

Analítico-Sintético: para identificar los conceptos y definiciones más importantes relacionadas con la creación de personalización, su interrelación y posteriormente poder arribar a una propuesta adecuada a la situación planteada.

Análisis Histórico-Lógico: para estudiar la evolución y desarrollo hasta la actualidad de los sistemas operativos de la familia SUSE y de las diferentes herramientas que permiten crear una personalización de estos sistemas.

Inductivo-deductivo: proceso de generalización sobre las características, nivel de desarrollo y ventajas de las herramientas existentes e implementadas.

Capítulo 1: Fundamentación teórica. Se hace un análisis del tema a tratar, de las tecnologías actuales relacionadas con las herramientas de personalización de distribuciones basadas en SUSE Linux a nivel mundial. Además una breve descripción de las técnicas utilizadas para la implementación del sistema, así como una reseña histórica, principales tendencias y metodología a utilizar.

Capítulo 2: Características del sistema. Se detallan los artefactos generados durante las fases de planificación, propias de la metodología de desarrollo utilizada para la implementación de la aplicación web.

Capítulo 3: Diseño e Implementación. Se precisan con mayor especificidad los artefactos, con vista a la implementación del sistema, de acuerdo a lo definido por la metodología de desarrollo de software.

Capítulo 4: Pruebas de la herramienta. Se expone todo lo relacionado a la fase de pruebas, propias de la metodología de desarrollo utilizada para la implementación de la aplicación web.

Capítulo I. Fundamentación Teórica.

1.1 Introducción.

Con el uso de un sistema operativo completamente libre tenemos la posibilidad de emplearlo o modificarlo para que sea utilizado para un determinado uso u objetivo específico, para ello habría que hacer una personalización del mismo. Se define personalización como las modificaciones, cambios y transformaciones que se le hace a la distribución usada para lograr el objetivo por el cual lo hacemos.

En este capítulo se realiza un estudio de los términos que más se utilizan para crear distribuciones personalizadas de SUSE. Para ello es necesario analizar conceptos, características, así como las tendencias en la actualidad de las herramientas que permiten hacer personalizaciones a esta distribución y de los sistemas operativos libres en particular la familia SUSE.

1.2 Software Libre y GNU/Linux.

El término "Software Libre" se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo exacto y preciso, se refiere a cuatro libertades que poseen los usuarios del software. De acuerdo con tal definición, el software es "libre" si garantiza las siguientes libertades:

- "Libertad 0", ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.)
- "Libertad 1", estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente)
- "Libertad 2", copiar el programa de manera que se pueda ayudar al vecino o a cualquiera persona.
- "Libertad 3", mejorar el programa, y hacer públicas las mejoras, de forma que se beneficie toda la comunidad. (1)

El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo que no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en algunas ocasiones el código fuente; no obstante, este tipo de software no es libre en

el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa. Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquél que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de éste, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público.

1.2.1 ¿Qué es GNU/Linux?

GNU/Linux es el primer sistema operativo basado en UNIX (otro sistema operativo legendario) que es 100% software libre. Si bien anteriormente había otros sistemas operativos de libre distribución (como MINIX), estos no eran totalmente software libre ya que eran regidos por licencias más restrictivas. GNU/Linux es un proyecto que ya lleva 20 años en desarrollo y lo estará por muchos más, ya que se asienta sobre una base de miles de programadores de todas partes del mundo. De manera análoga se utiliza popularmente el término "Linux" a la hora de referirse a este sistema operativo, aunque este término no está correctamente empleado ya que el término "Linux" estrictamente se refiere al núcleo o Kernel Linux.

Técnicamente, GNU/Linux es un sistema operativo de software libre que cumple las normas POSIX (Portable Operating System Interface), su base es un núcleo o Kernel monolítico llamado Linux combinado con un grupo de librerías y herramientas. Su estructura general es la típica de cualquier sistema UNIX (núcleo, "interprete de comandos", aplicaciones). GNU/Linux tiene todas las características que se pueden esperar de un moderno y flexible sistema operativo. Incluye multitarea real, memoria virtual, librerías compartidas, dirección y manejo propio de memoria. Es sin lugar a dudas uno de los ejemplos más prominentes del software libre y del desarrollo del código abierto.

1.2.2 Distribuciones de GNU/Linux.

Una distribución de GNU/Linux es una variante de ese sistema operativo que incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones hogareñas, empresariales y para servidores. Pueden ser exclusivamente de software libre o también incorporar aplicaciones o controladores privativos.

La base de cada distribución incluye el núcleo Linux, con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos o grupos de software, como Apache, MySQL, PostgreSQL, Perl, Python, PHP, Gnome y KDE.

En la actualidad hay muchas distribuciones libres tales como:

- GNewSense, una distribución GNU/Linux completamente libre basada en Debian y Ubuntu, patrocinada por la FSF
- UTUTO-e, una distribución GNU/Linux basada en Gentoo. Fue la primera distribución de GNU/Linux completamente libre reconocida por el Proyecto GNU.
- Dynebolic, una distribución de GNU/Linux que pone un énfasis especial en la edición de audio y vídeo.
- Musix GNU+Linux, una distribución GNU+Linux basada en Knoppix, con un énfasis especial en la producción de audio.
- BLAG, BLAG Linux y GNU, una distribución GNU/Linux basada en Fedora.
- Trisquel, una distribución GNU/Linux destinada a pequeñas empresas, usuarios domésticos y centros educativos. (2)

Estas rechazan las aplicaciones, las plataformas de programación, los controladores y el software no libre. Existen muchas otras pero que no cumplen con esto ya que usan software que no son libres y por tal motivo no son consideradas libres, aunque muchas están trabajando en este sentido.

1.2.3 ¿Qué distribución utilizar?

Hoy en día se cuenta con miles de distribuciones y al ver esto para muchas personas que inician o pretenden comenzar a usar un sistema operativo con el cuál puedan hacer las maravillas que deseen, es realmente complejo, porque surgen interrogantes como: ¿Por qué tantas?, ¿Qué sentido tienen?, ¿Cuál escojo?, darles respuestas a estas es difícil, ya que cada distribución es desarrollada con un fin específico, un ejemplo de ello es UTUTO, la cual en un solo CD instalable comprime 4.7 GB con casi todos los programas realmente necesarios, solucionándonos dos problemas graves, el de la capacidad de disco duro y de conexión.

Otros prefieren usar una con gran estabilidad como Ubuntu y openSUSE, las cuales fueron las dos mejores distribuciones en el año 2008, o simplemente otra como Debian por su popularidad, y así sucede con el uso de todas las distribuciones, ya que las mismas son creadas y utilizadas para satisfacer las necesidades de un grupo determinado de personas, por lo que recomendar una porque sea la mejor o sea la más linda o cualquier otro tipo de razón es del gusto de cada quién, ya que la esencia del problema para utilizar una radica en los intereses a los cuales dicha distribución debe satisfacer.

1.3 ¿Por qué utilizar SUSE como punto de partida?

El origen de SUSE se remonta al año 1992, cuando cuatro entusiastas usuarios de Linux, los alemanes Roland Dyroff, Thomas Fehr, Hubert Mantel y Burchard Steinbild, lanzaron el proyecto bajo el nombre de SuSE (Desarrollo de sistemas y software). Al comienzo, la naciente empresa solía vender discos informáticos los cuales contenían la versión alemana de Linux-Slackware, pero no pasó mucho tiempo hasta Linux-SUSE se transformara en una distribución independiente a través de su versión lanzada en mayo de 1996.

SUSE fue comprado por Novell, Inc.³ a finales del 2003 desde entonces hasta la fecha amplias han sido las modificaciones en el desarrollo. Los cambios en la licencia y disponibilidad de Linux-SUSE fueron aplicados casi de inmediato, las imágenes ISO fueron distribuidas libremente desde servidores públicos, y lo más significativo fue que el desarrollo de la distribución fue por primera vez abierto para todo el público.

El código de openSUSE ha llegado a ser el sistema base de los productos comerciales de NOVELL, primeramente llamado Novell Linux, pero renombrado más adelante como Empresas de Escritorios SUSE Linux y Empresas de Servidores SUSE Linux desde el lanzamiento del proyecto openSUSE y la versión 10.0 en octubre del 2005, la distribución a llegado a ser completamente libre en todo sentido.

Todas estas transformaciones de la familia de productos basados en SUSE Linux han ocurrido de manera sorprendente, debido a que ha pasado poco tiempo desde que fuera comprada por Novell hasta la fecha y este considerada una de las mejores, no son solo las que satisfacen nuestros intereses sino también por

³ **Novell**, Inc. es una compañía de origen estadounidense dedicada al software, específicamente en el área de sistemas operativos de redes, como Novell Netware y Linux.

la estabilidad que le presenta al MININT y a muchas empresas de gran prestigio a nivel mundial, que no solo desarrollan software sino también se desempeñan en otras áreas.

El rendimiento y funcionamiento que reportan estas distribuciones utilizadas en servidores de base de datos con el uso de un gestor tan potente como Oracle, es otra de las características que hace que sea SUSE la que se utilice como punto de partida, no obstante en un futuro habría que pensar en NOVA, una distribución que se desarrolla en nuestro país pero que apenas fue lanzada su primera versión y por tanto no tiene una madurez y estabilidad que responda a las necesidades del MININT.

1.4 Principales Características de SUSE.

Las distribuciones de la familia SUSE incluyen varios escritorios, entre ellos los más conocidos que son KDE y GNOME, siendo el primero el escritorio por omisión, incorporan las herramientas necesarias para redistribuir el espacio de tu disco duro permitiendo así la coexistencia con otros sistemas operativos existentes en el mismo. Usa sistemas de paquetes RPM (Red Hat Package Manager) aunque no guarda relación con esta distribución.

También es posible utilizar el sistema de instalación CNR (Click 'N Run) originalmente creado por la empresa que distribuía Lindows OS (actualmente, Linespire y Freespire en su versión gratuita). Este sistema sincroniza la computadora al servidor CNR y al darle clic en la página de navegación a alguno de los programas, este se instala de manera automática en el ordenador. Incluye además un programa único de instalación y administración llamado YaST que permite realizar actualizaciones, configurar la red y el cortafuego, administrar a los usuarios, y muchas más opciones, todas ellas integradas en una sola interfaz amigable.

1.4.1 YaST.

YaST es la herramienta de instalación y configuración de openSUSE, SUSE Linux Enterprise y las distribuciones SUSE Linux. Es popular por su facilidad de uso y atractiva interfaz gráfica, así como la capacidad de configurar su sistema rápidamente durante y luego de la instalación. (3)

Actualmente existen dos versiones de YaST disponibles, ambas con las mismas capacidades, una es la versión que utiliza el framework QT y la otra la versión realizada sobre GTK (por defecto desde openSUSE

10.3 para el entorno GNOME). Aunque independientemente del entorno de escritorio utilizado puede optar por una u otra. (3)

YaST no solo le guía durante la instalación de openSUSE sino también para configurar sus dispositivos, el entorno de red, los servicios del sistema, afinar su configuración de seguridad y muchas otras cosas. Todas estas tareas pueden realizarse desde el Centro de Control de YaST donde se encuentran todas las opciones, una de ellas es instalar y desinstalar software en tu máquina, para esto cuenta con un sistema de gestión de paquetes o conocido también como gestor de paquetes, que no es más que una colección de herramientas que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software basta con introducir en el buscador el programa que quieres instalar para que te aparezca y con un clic instalarlo, pero todo esto es posible gracias a los repositorios de SUSE.

1.4.2 Formato de Paquetes utilizados en SUSE.

SUSE Linux utiliza RPM con los programas principales RPM y rpmbuild para la administración de los paquetes de software. La gran base de datos de RPM facilita la gestión de los paquetes para todos los implicados: los usuarios, los administradores de sistema y los que generan los paquetes; RPM ofrece una gran cantidad de información sobre el software instalado.

Básicamente, RPM puede actuar de cinco maneras distintas: instalar, desinstalar o actualizar paquetes de software, volver a crear la base de datos RPM, enviar consultas a la base de datos RPM o a archivos RPM individuales, comprobar la integridad de los paquetes y firmar paquetes. El rpmbuild se utiliza para generar paquetes listos para instalar a partir de las fuentes originales.

Los archivos RPM instalables tienen un formato binario especial que incluye los archivos con los programas e información adicional usada por RPM. Esta información adicional se usa para configurar el software del paquete o para la documentación en la base de datos RPM, estos archivos tienen la extensión .rpm.

Un paquete RPM en realidad no es más que un fichero con terminación .rpm, que contiene una aplicación (o su código fuente) y cierta información sobre la misma. Su nombre tiene la siguiente estructura:

`{nombre}-{versión de la aplicación}-{versión del paquete}.{arquitectura}.rpm`

1.4.3 Repositorios de SUSE.

Un repositorio, depósito o archivo es un sitio centralizado donde se almacena y mantiene determinada información que en este caso son paquetes, podemos interpretar el concepto como “almacén de paquetes”, que un paquete no es más que programas, librerías, plugins o simplemente una pieza de software que cumple una determinada funcionalidad.

Los repositorios pueden estar físicamente ubicados en un disco duro, en una memoria flash, en un CD-ROM, en internet o simplemente en cualquier dispositivo al cual se pueda acceder; los repositorios oficiales públicos de SUSE contienen únicamente software libre. A pesar de las comodidades y ventajas que tiene un repositorio a muchos usuarios no cuentan con internet o red, para acceder a repositorios oficiales y estar actualizado es esta una de las razones y de aquí la importancia de hacer una personalización de sistemas base de SUSE que tenga solamente lo que deseamos.

1.5 Herramientas que permiten personalizar distribuciones de la familia SUSE.

Actualmente en el mundo existen muy pocas herramientas que permitan una personalización de distribuciones de sistemas basados en SUSE, y las pocas que existen realmente se tornan difíciles o complejas para usuarios con un mínimo de experiencia, además de las deficiencias de las mismas. Se puede mencionar y hacer referencias solo a tres de ellas:

1.5.1 KIWI.

La creación de imágenes con la herramienta KIWI consta de dos pasos básicos: preparar y crear; para ejecutar ambos pasos se busca la configuración de un archivo XML que contiene la información del tipo de imagen que se requiere y de los software que se necesitan, el primer paso, o sea, preparar requiere la siguiente estructura: (4)

`Kiwi -prepare /ruta donde se encuentra el archivo de configuración.xml/ --root /ruta del directorio donde se va a almacenar la imagen/`

Figura 1. Sintaxis para la preparación de una imagen con KIWI.

Ejemplo:

- `kiwi -prepare /usr/share/kiwi/imagen/livecd/ --root /mnt/suse/.`

Concluido el proceso de preparación le sigue la creación, paso en el cual se obtendrá un sistema operativo que será utilizado según el tipo que se desee adquirir ya sea un LiveCD, para su instalación a través de cualquier dispositivo de almacenamiento USB o bien para ser utilizado en una máquina virtual.

`Kiwi -create /ruta donde se almacena la imagen/ -d /destino donde se generara el iso o tar.gz/ -type iso, usb, xen`

Figura 2. Sintaxis para la creación de una imagen con KIWI.

Ejemplos:

- `kiwi -create /mnt/suse/ -d /mnt/suseresult/ -type iso.`
- `kiwi -create /mnt/suse/ -d /mnt/suseresult/ -type usb.`

Si bien esta herramienta realiza una personalización deseada por el usuario presenta un inconveniente pues requiere de conocimientos avanzados de comandos a ejecutar así como la engorrosa configuración del archivo XML en el cual Kiwi se basa para llevar a cabo dicho proceso.

1.5.2 Creador de Imágenes.

Este no es más que una parte del paquete `yast2-product-creator` o sea una interfaz de usuario para KIWI. `Yast2-add-on-creator` crea un producto personalizado (imagen ISO) a partir de archivos RPM. El `yast2-cd-creator` crea una imagen ISO a partir de productos existentes. Se puede utilizar para personalizar un producto existente. Por ejemplo, se puede crear una imagen ISO más pequeña con un subconjunto de todos los paquetes (por ejemplo, paquetes básicos para un sistema mínimo) o, por el contrario, una imagen ISO con más productos como SLES + SDK. Por otro lado el `yast2-product-creator` es un sucesor de `cd-creator`. Incluye una GUI para el sistema de imagen kiwi. Esto permite además crear una imagen de Live-CD, XEN, etc. a partir de la misma configuración utilizada en el creador de CD. El creador de productos se distribuye junto con el SDK.

1.5.3 SUSE Studio.

De esta no podemos hablar mucho aún ya que se encuentra en periodo de prueba, pero lo novedoso de ella es que es una aplicación web, lo que la pone en ventaja de las dos anteriores porque al igual que las anteriores mencionadas realiza las mismas operaciones y de manera más sencilla y rápida. Todo esto está muy bueno pero, ¿Los que no contamos con internet?, y si contara con ella ¿Cuanto demoraría descargar una imagen personalizada para un DVD que tiene 4.2 GB si mi ancho de banda es de menos de 128 MB? Además de estas inconvenientes esta herramienta se desconoce si va a ser liberada.

Luego de haber realizado un estudio de estas herramientas que son las más utilizadas se puede arribar a la conclusión de que ninguna puede ser propuesta para utilizar ya que no cumplen los requisitos que satisfagan las necesidades del MININT a pesar de que fueron creadas bajo las libertades de software libre, pero si reutilizar lo que hace el KIWI uniéndolo a una aplicación web que reúna las ventajas que poseen otras como SUSE Studio con lo cual si se lograría una herramienta que realmente cumpla las expectativas esperadas.

1.6 Metodologías de desarrollo de software.

Para la creación de un software es necesario basarse en una metodología de desarrollo de software que ayude a planificar y organizar todo el proceso para poder obtener un producto de óptima calidad y así los clientes queden satisfechos con los resultados. En muchas ocasiones no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de tres o cuatro meses, pero esto también sucede con proyectos de mayor envergadura donde muchas veces el software es diseñado de manera rígida, con los requerimientos que el cliente nos solicitó, de tal manera que cuando el cliente en la etapa final solicita un cambio se torna muy difícil realizarlo, pues si lo hacemos altera muchas cosas que no se habían previsto, y es justo éste, uno de los factores que atrasan el proyecto y por tanto la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido. Obviamente para evitar estos incidentes se debe haber llegado a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio o modificación no perjudique al desarrollo del mismo.

Actualmente se cuenta con una buena cantidad de propuestas que definen este marco de trabajo, es común escuchar hablar de metodologías tradicionales o robustas como Rational Unified Process (RUP), Microsoft Solutions Framework (MSF) y Métrica surgida en el año 1989 o metodologías ágiles como

Extreme Programming (XP), Scrum, Cristal Methods y Feature Driven Development, pero las experiencias indican que no existe una receta mágica y no se trata de solo seleccionar una de las propuestas y seguirla sino los problemas mencionados anteriormente no desaparecerán.

1.6.1 Extreme Programming (XP).

En este caso se usará la metodología ágil: XP, ya que XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

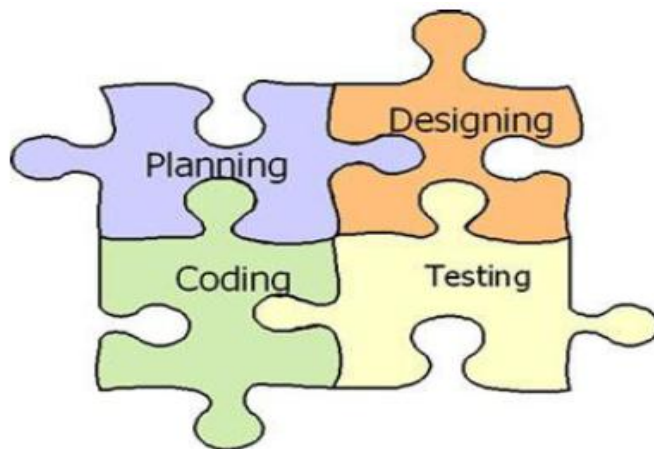


Figura 1: Fases de Extreme Programming.

1.7 Herramientas y lenguajes a utilizar.

El acelerado desarrollo tecnológico de los últimos años propició el paso de la sociedad industrial a la sociedad de la información y del conocimiento. Las tecnologías web, desde esta perspectiva, se presentan como una variable determinante del éxito de las empresas y organizaciones. Para el desarrollo de una herramienta que permita crear distribuciones personalizadas de sistemas bases basados en SUSE, capaz de satisfacer todas las demandas requeridas, que estas se realicen con la mayor agilidad posible, y que además sea una herramienta de software libre, basándonos en los principios, libertades y las ventajas que

nos ofrece el mismo se ha decidido utilizar como tecnología un servidor web Apache, como lenguaje de programación Hypertext Pre-Processor (PHP).

1.7.1 Servidores Web.

El hombre, a lo largo de la historia ha necesitado almacenar información de manera temporal o permanente, conforme a su dominio de la tecnología, ha ido desarrollando procedimientos eficaces y estables. A partir de esto ha desplegado diferentes variantes, como constituyen en la actualidad los servidores.

Un servidor no es más que una computadora incluida en una red, que le ofrece ya sea servicios o información a otras PCs, llamadas clientes. Existen varios tipos de servidores como: los servidores de correo, de chat, de fax, de Web.

Los servidores Web son ordenadores que a través del protocolo HTTP (Hypertext Transfer Protocol), los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. A modo conceptual, *“un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (Hypertext Transfer Protocol Secure)”* (Cibernetia, 2009)

Como servidor Web destacado se puede señalar a Internet Information Services (IIS), que ofrece una serie de servicios para los ordenadores como: FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), NNTP (Network News Transport Protocol) y HTTP/HTTPS. Se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl, aunque este servidor solo funciona con sistema operativo Windows, por lo que es propiedad de Microsoft Corporation.

1.7.1 Apache.

Otro de los servidores Web más utilizados a nivel mundial es el servidor HTTP Apache. El servidor HTTP Apache es un software (libre) de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. La

arquitectura del servidor Apache es muy modular. Dicho servidor consta de diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web. Algunos de estos módulos son:

- `mod_ssl` - Comunicaciones Seguras vía TLS.
- `mod_rewrite` - reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como PHP en páginas estáticas HTML para así engañar a los navegantes o a los motores de búsqueda en cuanto a cómo fueron desarrolladas estas páginas).
- `mod_dav` - Soporte del protocolo WebDAV.
- `mod_deflate` - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- `mod_php` - Páginas dinámicas en PHP.
- `mod_security` - Filtrado a nivel de aplicación, para seguridad.(5)

Ventajas.

Es multiplataforma, extensible y de código abierto, puede soportar un sin número de sistemas operativos. HTTP Apache está conformado para ser un servidor Web flexible, atributo que le permite personalizar las necesidades de cada sitio.

Dados los argumentos expuestos con anterioridad se tomó la decisión de utilizar el servidor HTTP Apache.

1.7.2 Intérpretes de comando.

Un intérprete de comandos es la interfaz de comunicación entre el usuario y el sistema operativo. Existen numerosos *shell*: `csh` (C Shell), `sh` (Bourne Shell), pero, quizás, el más extendido entre los usuarios e implementaciones de Linux es el `bash` (Bourne Again Shell). La principal diferencia entre los *shell* es en las funciones que aportan, instrucciones, funcionalidad en general. Y esta es la razón dominante que determinará que intérprete de comandos utilizar.

Tiene como ventaja las siguientes:

- Permiten administrar remoto los servidores vía `ssh`.
- Muchos servidores de hoy en día no tienen interfaz gráfica porque dicha interfaz consume recursos

indispensables para otros servicios, por lo que lo mejor sería ejecutar comandos en la consola (Interprete de comando).

1.7.3 Gestores de base de datos (SGBD).

Un Sistema Gestor de Base de Datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida. Permite a los usuarios definir, crear y mantener la base de datos (BD) proporcionando acceso controlado a la misma. Dentro de los sistemas gestores más distribuidos a nivel mundial se encuentran: Oracle, PostgreSQL, MySQL.

Oracle (Relational Data Base Management System)

Oracle se destaca por su estabilidad y escalabilidad, además de ser multiplataforma. En el desarrollo de páginas Web pasa lo mismo al ser un sistema de alto costo no está extendido como otras bases de datos como SQL Server, Access o MySQL.

PostgreSQL

Es un poderoso manejador de bases de datos de código abierto sin costos de licencia, diseñado para administrar grandes cantidades de datos. Es robusto, confiable y mantiene la integridad de los datos. Se ejecuta en la mayoría de los Sistemas Operativos más utilizados en el mundo incluyendo, Linux, varias versiones de UNIX y en Windows.

PostgreSQL soporta funciones que donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés). Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos ofrece, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario, donde ha alcanzado más de seis millones de usuarios, considerado el más utilizado en el mundo actualmente., se ofrece bajo la licencia GNU GPL (General Public License) y es desarrollado por MySQL AB⁴ como software libre en un esquema de licenciamiento dual. , es robusto, posee triggers, vistas además de cursores. Dispone de

⁴ Compañía de software fundada en 1995, creadora del sistema administrador de bases de datos relacionales MySQL

diferentes funciones, las cuales exigen los desarrolladores profesionales, como funciones SSL(**Secure Socket Layer**) e integración con la mayor parte de los entornos de programación como C, C++, Eiffel, Java, Perl, PHP, Python etc. Al ejecutarse en la inmensa mayoría de sistemas operativos es muy común que los datos se puedan transferir de un sistema a otro sin dificultad. (6)

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C⁵. Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Se resolvió disponer de este sistema gestor de bases de datos en la realización de la Herramienta MinPSUSE puesto que la base de datos de la aplicación es pequeña no obstante se podría utilizar otro gestor como PostgreSQL sin problema alguno para la aplicación puesto que se utiliza un drivers que es el encargado de conectar el sistema con el gestor de base de datos.

1.7.4 Sistemas para administrar SGBD.

Muchas son las aplicaciones destinadas para la administración de SGBD, ejemplo de ellos son, EMS MySQL Manager, MySQL Query Browser, phpMyAdmin, etc.

⁵ La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C.

1.7.4.1 PhpMyAdmin.

Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de una interfaz web muy intuitiva. Actualmente puede crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL y hacer un backup de la base de datos, administrar claves en campos, administrar privilegios, exportar datos en varios formatos además de que está disponible en 55 idiomas. Se encuentra disponible bajo la licencia GPL. (7)

1.7.5 Lenguajes de programación Web.

Los lenguajes de programación web han ido surgiendo debido a las tendencias y necesidades de las plataformas, con el objetivo de permitir la interacción con los usuarios y la utilización de los sistemas de Base de Datos, se clasifican en lenguajes del lado del cliente y lenguajes del lado del servidor. Entre los lenguajes que trabajan del lado del cliente podemos citar algunos de ellos como son HTML, JAVASCRIPT, APPLETS de JAVA, VISUAL BASIC SCRIPT (VBSCRIPT), CSS, XSL. En los lenguajes que trabajan del lado del servidor se pueden mencionar algunos como PERL, ASP, PHP, JSP, JAVA. En esta investigación se profundizará en los lenguajes de interés para el desarrollo de la aplicación.

1.7.5.1 Hypertext Markup Language (HTML).

Hypertext Markup Language (HTML) por sus siglas en inglés, en español (*Lenguaje de Marcas de Hipertexto*), es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también le indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos -ya sea en su computadora o en otras así como otros recursos de Internet, como FTP y Gopher. (Chuck Musciano y Bill Kennedy, pp. 7, 1999).

HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. Es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

1.7.5.2 Extensible Markup Language (XML).

Extensible Markup Language (XML) por sus siglas en inglés, en español (Lenguaje de Marcas), es un lenguaje de etiquetado extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML, además tiene un formato que permite la lectura de datos a través de diferentes aplicaciones. (8)

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

Entre las tecnologías XML disponibles se pueden destacar:

- ✓ XSL: Lenguaje Extensible de Hojas de Estilo, cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación como puede ser una ventana de un navegador Web o un dispositivo móvil, o un conjunto de páginas de un catálogo, informe o libro.
- ✓ XPath: Lenguaje de Rutas XML, es un lenguaje para acceder a partes de un documento XML.
- ✓ XLink: Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos XML.
- ✓ XPointer: Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.
- ✓ XQL: Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web. (8)

1.7.5.3 Javascript.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript se puede crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el

propio navegador. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. (9)

Entre las acciones típicas que se pueden realizar en Java Script se tienen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambio de color o cualquier otro dinamismo. Por el otro, este lenguaje permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se puede crear páginas interactivas con programas como calculadoras, agendas o tablas de cálculo. (9)

Brinda la posibilidad de programar pequeños scripts, pero también programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

1.7.5.4 Cascading Style Sheets (CSS).

Las hojas de estilo en cascada (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea). (10)

1.7.5.5 PHP.

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.(11)

PHP es un acrónimo recursivo que significa PHP (Hypertext Pre-Processor) inicialmente PHP Tools, o, Personal Home Page Tools). Está publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre. Es ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. (11)

Entre las características más importantes podemos mencionar:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

1.7.6 Herramientas para el desarrollo web.

Existe una gran variedad de IDE (Integrated Development Environment) para PHP donde los más habituales son: Eclipse, NetBeans además de Zend Studio. A continuación se realizará una descripción de cada uno de los mismos, donde a partir de ésta se seleccionará la herramienta a utilizar en el desarrollo de MinPSUSE.

Eclipse.

Eclipse es un entorno de desarrollo integrado, de código abierto, multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar IDEs, como el IDE de Java llamado JDT (Java Development Toolkit) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

NetBeans.

NetBeans se refiere a una plataforma para el desarrollo de aplicaciones de escritorio usando Java y a un entorno de desarrollo integrado (IDE) desarrollado usando la Plataforma NetBeans. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo.

Zend Studio.

Zend Studio (Zend Development Environment) es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Características.

- No requiere la instalación previa de PHP ni del entorno de ejecución de Java.

- Soporte para PHP 4 y PHP 5.
- Permite resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- phpDoc integrado.
- Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).
- Inserción automática de paréntesis y corchetes de cierre.
- Sangrado automático y otras ayudas de formato de código.
- Emparejamiento (*matching*) de paréntesis y corchetes.
- Detección de errores de sintaxis en tiempo real.
- Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada.

(*Breakpoints*), seguimiento de variables y mensajes de error del intérprete de PHP. Permite también la depuración en servidores remotos (requiere Zend Platform(es la propuesta de Zend Technologies para el desarrollo de aplicaciones Web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre).

- Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox (opcional)
- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado.
- Soporte para control de versiones usando CVS o Subversion (a elección del desarrollador).
- Cliente FTP integrado.
- Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Java script y XML.

Debido a todas las ventajas que ofrece el Zend Studio se decidió utilizarlo en el desarrollo de la Herramienta MinPSuse.

1.7.7 Framework de Desarrollo.

A partir del desarrollo alcanzado en el mundo de las tecnologías ha sido necesaria la creación de marcos de trabajo (framework) con el objetivo de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel para proveer un sistema funcional.

Aunque los frameworks existen desde hace décadas, el lanzamiento de Ruby On Rails⁶ en 2004 supuso una revolución en el desarrollo de las aplicaciones web que aún hoy continúa, dando origen a la gran diversidad de frameworks que en la actualidad existen como:

- **Zend Framework**

Zend Framework se centra en el desarrollo de aplicaciones más robustas, confiables, modernas para la Web 2.0 y los servicios web, consumiendo una amplia gama de APIs disponibles de los principales proveedores Google, Amazon, Yahoo!, Flickr, Strikelron and ProgrammableWebç

- **Akelos**

El Framework de PHP Akelos, es una plataforma de desarrollo de aplicaciones web basada en el patrón de diseño MVC (Modelo – Vista - Controlador), que se enfoca en buenas prácticas que permiten programar Vistas que emplean AJAX de forma fácil, controlar las peticiones y las respuestas a través de las Controladoras, manejar aplicaciones estandarizadas internacionalmente, modelos de comunicación y de base de datos utilizando convenciones sencillas.

- **Symfony**

Primera versión publicada en Octubre de 2005 por Fabien Potencier. Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Para empezar, emplea el tradicional patrón de diseño Modelo Vista Controlador (MVC)⁷ para separar las distintas partes que forman una aplicación web, la lógica de negocio, la lógica de servidor y la presentación. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además,

⁶ Entorno de programación (Rails) que se apoya en el lenguaje Ruby. Goza de gran popularidad para el desarrollo de aplicaciones de tipo Web 2.0.

⁷ Patrón arquitectural muy difundido en uso en aplicaciones de entorno Web.

automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

- **CodeIgniter**

Poderoso Framework para PHP que facilita la escritura de código repetitivo, y a comparación de otros Frameworks como CakePHP, Symfony o Zend Framework. CodeIgniter es adecuado para desarrollos que no requieran un entorno de trabajo que marque mucho la aplicación, para cuando sea necesario mucho rendimiento. Pensado para aquellas aplicaciones que se ejecutan en hosting compartido que ejecutan muchas versiones de PHP con diferentes configuraciones. También te puede ser útil si no quieres usar un framework con configuraciones iniciales, o que sea necesaria la línea de comandos. Este entorno de trabajo trae consigo gran cantidad de documentación, ofreciendo de esta forma los elementos necesarios para que el desarrollador no le sea tan compleja la tarea de desarrollo.

El mismo incluye varias librerías para gestionar el acceso a datos, sesiones de usuarios, formularios, la seguridad, etc. Además la comunidad de usuarios ha creado una serie de plugins, clases y librerías para extenderlo que lo hacen aún más interesante.

Por estas razones se selecciona CodeIgniter para la adecuada implementación del sistema informático “MinPSUSE”.

1.8 Conclusiones.

En este capítulo se ha realizado un estudio de las características de la distribución a personalizar, de las herramientas que permiten crear una personalización de la distribución lo cual son de vital importancia para la realización de este trabajo, dando lugar a efectuar un estudio de las principales tecnologías para el desarrollo del software, como son: los lenguajes de programación web, tipo de servidores web etc., además de la metodología de desarrollo.

Capítulo II Planificación.

2.1 Introducción

Para el desarrollo de este capítulo y en general para complementar la solución propuesta de la herramienta se utilizó el desarrollo ágil utilizando la metodología ágil XP, por tal motivo es que hacemos alusión a las prioridades de las historias de usuarios, así como sus descripciones ofreciendo elementos necesarios para realizar estimaciones de tiempo con vista al plan de entrega del sistema, planificando además la pruebas para comprobar las funcionalidades desarrolladas.

2.2 Fase de planificación.

La fase de planificación da inicio a la metodología XP. En esta fase se identifican las historias de usuarios (HU), Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en la construcción de la aplicación Web.

2.2.1 Alcance.

Se presenta una aplicación Web, soportada sobre bases de datos en MySQL, la cual se desarrollará con lenguaje PHP 5.0 debido a sus potencialidades como software libre montado sobre un servidor Apache. Para la guía del desarrollo después de consumir un estudio de varias metodologías de desarrollo de software se decide recurrir a XP, puesto que es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. La herramienta será capaz de realizar de manera amigable la creación personalizada de un sistema operativo con soporte en SUSE Linux.

Ofrecidos los elementos generales del trabajo en la presente investigación, se considera necesario abordar los aspectos técnicos de la misma.

2.2.2 Prioridad.

En la implementación de la herramienta se consideran como historias de usuarios críticas a las pertenecientes a las funcionalidades: *“Preparar personalización”*, *“Agregar o quitar programas”* y *“Crear Personalización”*.

2.3 Historias de Usuarios.

Las Historias de Usuarios representan una breve descripción del comportamiento del sistema, emplean terminología del cliente sin lenguaje técnico, se realiza una por cada característica principal del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación, a continuación se describen cada una de las Historias de Usuarios que aparecen en el sistema.

HU_01_Gestionar Usuario.

Historia de Usuario	
Número:1	Nombre Historia de Usuario: Gestionar Usuario.
Usuario: superadmin, administrador.	Iteración Asignada: 2.
Prioridad en Negocio: Media.	Puntos Estimados: 1.
Riesgo en Desarrollo: Medio.	Puntos Reales: 1.
Descripción: El superadmin accede al sistema para crear, editar o eliminar una cuenta de usuario.	
Observaciones: La aplicación solo cuenta con un usuario con el rol de (superadmin) que es el que tiene el control total del sitio.	

Tabla 1. Historia de Usuario - Gestionar Usuario.

HU_02_Autenticar Usuario.

Historia de Usuario	
Número:2	Nombre Historia de Usuario: Autenticar Usuario.
Usuario: superadmin, administrador, usuario.	Iteración Asignada: 2.
Prioridad en Negocio: Media.	Puntos Estimados: 1.
Riesgo en Desarrollo: Medio.	Puntos Reales: 1.
Descripción: Se brinda la posibilidad a cualquiera de los roles definidos de acceder al sistema, para ello la persona introduce el nombre de usuario y la contraseña del mismo, se verifican que los datos sean correctos para otorgarle el permiso de entrada.	

Tabla 2. Historia de Usuario - Autenticar Usuario.

HU_03_Registrar Usuario.

Historia de Usuario	
Número:3	Nombre Historia de Usuario: Registrar Usuario.
Usuario: invitado.	Iteración Asignada: 2.
Prioridad en Negocio: Media.	Puntos Estimados: 1.
Riesgo en Desarrollo: Medio.	Puntos Reales: 1.
Descripción: Se brinda la posibilidad a cualquiera usuario que visite por primera vez la aplicación que se registre en la misma ofreciéndole así las posibilidades de utilizar el sistema, cuando termina el de registrarse se le envía un correo con los datos entrados por el visitante y con un link para que entre y se active la cuenta.	

Tabla 3. Historia de Usuario - Registrar Usuario.

HU_04_Gestionar Contraseña.

Historia de Usuario	
Número:4	Nombre Historia de Usuario: Gestionar contraseña.
Usuario: superadmin, administrador, usuario.	Iteración Asignada: 2.
Prioridad en Negocio: Baja.	Puntos Estimados: 1.
Riesgo en Desarrollo: Medio.	Puntos Reales: 1.
Descripción: Se brinda la posibilidad a cualquiera de los roles de cambiar su contraseña para acceder al sistema, o en caso de olvido de la misma, la aplicación admite el envío de esta a través del correo del propio usuario.	
Observaciones: El superadmin contará con los privilegios para cambiar la contraseña a cualquier usuario.	

Tabla 4. Historia de Usuario - Gestionar Contraseña.

HU_05_ Preparar Personalización.

Historia de Usuario	
Número:5	Nombre Historia de Usuario: Preparar Personalización.
Usuario: superadmin, administrador, usuario.	Iteración Asignada: 1.
Prioridad en Negocio: Alta.	Puntos Estimados: 1.
Riesgo en Desarrollo: Medio.	Puntos Reales: 1.
Descripción: El sistema le ofrece al usuario unas configuraciones predefinidas con lo básico y necesario para obtener un sistema personalizado, dichas configuraciones pueden ser modificadas o ajustadas de acuerdo a las exigencias o necesidades por la cual es objeto de creación.	

Tabla 5. Historia de Usuario - Preparar Personalización.

HU_06_Agregar o Quitar programas.

Historia de Usuario	
Número:6	Nombre Historia de Usuario: Agregar o Quitar programas.
Usuario: superadmin, administrador, usuario.	Iteración Asignada: 1.
Prioridad en Negocio: Alta.	Puntos Estimados: 1.
Riesgo en Desarrollo: Alto.	Puntos Reales: 1.
Descripción: Se visualizan los software disponibles en el repositorio organizados por categorías, con vista a su selección para conformar la imagen a personalizar.	

Tabla 6. Historia de Usuario - Agregar o Quitar programas.

HU_07_Crear Personalización.

Historia de Usuario	
Número:7	Nombre Historia de Usuario: Crear Personalización.
Usuario: superadmin, administrador, usuario.	Iteración Asignada: 1.
Prioridad en Negocio: Alta.	Puntos Estimados: 1.
Riesgo en Desarrollo: Alto.	Puntos Reales: 0.9
Descripción: Después de conformar la imagen deseada por el usuario autenticado en el sistema, la aplicación permitirá la opción de crear personalización, donde se crea un archivo con extensión ISO listo para descargarlo.	

Tabla 7. Historia de Usuario - Crear Personalización.

HU_08_Descargar ISO.

Historia de Usuario	
Número:8	Nombre Historia de Usuario: Descargar ISO.
Usuario: superadmin, administrador, usuario.	Iteración Asignada: 1.
Prioridad en Negocio: Alta.	Puntos Estimados: 1.
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.7
Descripción: Tras crear la imagen personalizada el sistema ofrece la posibilidad de almacenar el archivo ISO en la PC (Personal Computer) del usuario autenticado.	

Tabla 8. Historia de Usuario - Descargar ISO.

HU_09_Subir paquete.

Historia de Usuario	
Número:9	Nombre Historia de Usuario: Subir paquete.
Usuario: superadmin, administrador, usuario.	Iteración Asignada: 1.
Prioridad en Negocio: Alta.	Puntos Estimados: 1.
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.8
Descripción: Tras la selección de la plantilla base de la imagen el usuario tendrá la posibilidad de incorporar paquetes de funcionalidades para dicha imagen.	
Observaciones: El usuario deberá asignar el nombre propio del paquete importado.	

Tabla 9. Historia de Usuario - Subir Paquete.

HU_10_Listar mis proyectos.

Historia de Usuario	
Número:10	Nombre Historia de Usuario: Listar mis proyectos.
Usuario: superadmin, administrador, usuario.	Iteración Asignada: 3.
Prioridad en Negocio: Baja.	Puntos Estimados: 1.
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.8.
Descripción: Luego de haberse autenticado se le muestra a el usuario todos los proyectos seleccionadas que no son más que un archivo XML donde está toda la configuración de la personalización creada anteriormente por el mismo, mostrándole además las opciones de eliminar o modificar dándole así la posibilidad de crear una nueva personalización .	
Observaciones: Un proyecto no es más que una plantilla básicas que estas es un archivo de configuración XML pero con los paquetes básicos necesarios para un determinado entorno de escritorio o sin este, Ejemplo de plantillas creadas: Solo texto(es sin entorno gráfico), GNOME, KDE3 KDE4.	

Tabla 10. Historia de Usuario - Listar mis proyectos.

HU_11_Gestionar Repositorios.

Historia de Usuario	
Número:11	Nombre Historia de Usuario: Gestionar Repositorios
Usuario: superadmin, administrador	Iteración Asignada: 3.
Prioridad en Negocio: Baja.	Puntos Estimados: 1.
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.9.
Descripción: Se les muestra un listado de todos los repositorios por cada plantilla básica de configuración, brindado así la posibilidad de modificar, agregar o eliminar los mismos.	
Observaciones: Solo los usuarios con privilegios de administración podrán realizar estas modificaciones	

Tabla 11. Historia de Usuario - Gestionar Repositorios.

2.4 Plan de Iteraciones.

Una vez identificadas las Historias de Usuario del sistema se procede a la planificación de la etapa de implementación del proyecto (12). En base a lo antes mencionado se decide realizar esta en tres iteraciones, las cuales se detallan a continuación.

- **Iteración 1** El objetivo de esta es la implementación de las Historias de Usuario de prioridad alta. Al finalizar se contará con una versión de prueba con las funcionalidades concernientes a preparar personalización, agregar y quitar programas, subir paquete, crear personalización y descargar ISO.
- **Iteración 2** Esta iteración tiene como objetivo la implementación de las Historias de Usuario de prioridad media como: la gestión de usuarios, la gestión de contraseñas, registro y autenticación de usuario. Al final de esta se contará con una primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.
- **Iteración 3** Durante el transcurso de esta se implementaron las Historias de Usuario de baja prioridad. Al finalizar la misma se constará con la versión 1.0 del producto final, adicionando todo lo concerniente a la gestión de repositorios y el listado de los proyectos creados perteneciente a cada

usuario. Como resultado de esta, el sistema será puesto en funcionamiento durante un periodo de tiempo para evaluar su desempeño.

2.4.1 Plan de duración de las iteraciones.

En el ciclo de vida de un proyecto regido por la Metodología XP se crea el plan de duración de cada una de las iteraciones, en este caso se hace para el único equipo de desarrollo con el cual se cuenta. Este plan tiene como finalidad mostrar el tiempo de duración de cada iteración, así como el orden en que serán implementadas las Historias de Usuario en cada una de las mismas.

Iteraciones	Orden de las Historias de Usuario a implementar.	Duración total de las iteraciones.
Iteración 1	• Preparar Personalización. • Agregar y/o quitar programas. • Subir paquete. • Crear Personalización • Descargar ISO.	• 5 semanas.
Iteración 2	• Gestionar usuario. • Autenticar Usuario. • Registrar Usuario. • Gestionar Contraseña.	• 4 semanas.
Iteración 3	• Listar mis proyectos. • Gestionar Repositorios	• 3 semanas.

Tabla 12. Plan de duración de iteraciones.

2.5 Plan de entregas.

A continuación se presenta el plan de entregas elaborado para la fase de implementación.

Historias de Usuario.	Fin 1ra iteración 1ra semana de abril.	Fin 2da iteración 4ta semana de abril.	Fin 3ra iteración 3ra semana de mayo.
Gestionar Usuario.		1.0	
Autenticar Usuario.		1.0	
Registrar Usuario.		1.0	
Gestionar Contraseña.		1.0	
Preparar personalización	1.0		
Agregar y quitar programas.	1.0		
Crear personalización.	1.0		
Subir paquete.	1.0		
Descargar ISO.	1.0		
Listar mis proyectos.			1.0
Gestionar repositorios			1.0

Tabla 13. Plan de entregas.

2.6 Conclusiones.

Durante el desarrollo del presente capítulo se referencia a todo lo concerniente a la fase de Planificación propia de la metodología de desarrollo utilizada para la implementación de la aplicación Web, concibiendo una descripción de cada uno de los artefactos generados en el transcurso del mismo. Se asume una implementación por etapas. Esta fase se considera como un permanente diálogo entre las partes: lo deseable y lo posible, determinándose:

- **Ámbito:** ¿Qué es lo que el software debe de resolver para que genere valor?
- **Prioridad:** ¿Qué debe ser hecho en primer lugar?
- **Composición de versiones:** ¿Cuánto es necesario hacer para saber si el negocio va mejor con software que sin él?
- **Fecha de versiones:** ¿Cuáles son las fechas en la presencia del software o parte del mismo para marcar la diferencia?
- **Estimaciones:** ¿Cuánto tiempo lleva implementar una característica?

- **Procesos:** ¿Cómo se organiza el trabajo y el equipo?
- **Programación detallada:** Dentro de una iteración ¿Qué problemas se resolverán primero?

Capítulo III Fase de Diseño.

3.1 Introducción.

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada historia de usuario. Este se expresa utilizado como medida el punto. Un punto se considera como una semana ideal de trabajo donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción (12). Esta estimación incluye todo el esfuerzo asociado a la implementación de la Historia de Usuario, por ejemplo: las pruebas unitarias, la integración y refactorización del código y la preparación y ejecución de las pruebas de aceptación. (12)

3.2 Metáfora para el sistema.

Una metáfora para el sistema es una historia que todo el mundo puede contar a cerca de cómo el sistema funciona. (BECK. 2002) La metáfora en XP permite generar una explicación común por parte de los desarrolladores, analistas e incluso el personal que trabaje en la aplicación de generar un concepto común acerca de los términos que se manejarán en cuanto a los nombres y definiciones de la aplicación a implementar. La importancia de la asignación de un nombre determinado a cualquier objeto o parte de la herramienta radica en la comprensión por parte de los usuarios, quedando explícito en el nombre asignado la intención de la herramienta.

3.2.1 Definición de términos.

- ✓ **MinPSUSE:** Nombre que adopta la herramienta en desarrollo, llevada a cabo por la investigación.

3.2.2 Tarjetas CRC (Cargo o clase, Responsabilidad y Colaboración).

Para el desarrollo exitoso de la aplicación poder diseñar el sistema como un equipo se debe cumplir con tres principios: Cargo o Clase, Responsabilidad y Colaboración (CRC). Las tarjetas CRC permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos permitiéndole que el equipo completo contribuya en la tarea del diseño.

Una tarjeta CRC representa un objeto. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

En la aplicación MinPSUSE cada pantalla se comporta como un objeto independiente, de esta forma la herramienta está formada por las siguientes clases:

- 1- Autenticación.
- 2- Administración.
- 3- Preparar personalización.

Clase: Autenticación.

Responsabilidades	Clases relacionadas
Mostrar el formulario para la autenticación de cada usuario.	Administración.
Mostrar formulario para cambio de contraseña.	
Mostrar formulario para envío de contraseña por e-mail.	
Visualiza la opción para salir del sistema.	

Tabla 14. Tarjetas CRC - Clase Autenticación.

Clase: Administración.

Responsabilidades	Clases relacionadas
Listar usuarios y administradores del sistema.	
Gestionar determinado usuario o administrador del sistema.	
Gestionar repositorio.	

Tabla 15. Tarjetas CRC - Clase Administración.

Clase: Preparar personalización.

Responsabilidades	Clases relacionadas
Listar plantillas base de la personalización.	
Mostrar un formulario para el cambio de nombre de dicha plantilla.	
Agregar o quitar programas a la imagen.	
Importar paquetes.	
Listar paquetes importados con anterioridad por ese usuario.	
Listar los repositorios.	
Mostrar los software desde el repositorio, ordenados por	

grupos de paquetes.	
Mostrar listado de los software que contiene un determinado grupo de paquetes.	
Crear ISO.	
Descargar ISO personalizado.	

Tabla 16. Tarjetas CRC - Clase Preparar Personalización.

3.3 Soluciones Puntuales.

Una personalización de determinada distribución permite realizar una selección de una serie de software que el usuario desea incluir en su imagen SUSE Linux a instalar. El sistema se desarrolla por primera vez, liberándose a partir de la investigación la primera versión de MinPSUSE.

- **No se añadirá funcionalidad en las primeras etapas.**

Se debe evitar añadir funcionalidades que en ese momento no se necesiten, aun incluso cuando se sabe exactamente cómo implementar, es decir, hay que centrarse en la tarea que se ha fijado para ese momento y hacerla de la mejor forma posible. Hay que programar lo que se ha fijado, y no perder el tiempo en desarrollar código que no se sabe si será utilizado.

- **Reaprovechar cuando sea posible**

Cuando se elimina redundancia, se excluye funcionalidad inútil, y se fortalecen antiguos diseños, de esta forma se está reciclando código. El reciclaje, dentro del ciclo de vida de un proyecto, ahorra tiempo e incrementa la calidad. El reciclaje implica mantener el código limpio y fácil de comprender, modificar y ampliar. Esto puede resultar un poco costoso al principio, pero resulta fundamental a la hora de realizar diseños futuros.

3.4 Fase Implementación.

XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para retroalimentar a los desarrolladores con la opinión de este. En el presente capítulo se detallan las tres

iteraciones llevadas a cabo durante la etapa de construcción del sistema web, exponiendo las tareas generadas por cada Historia de Usuario.

Ajustándose a la planificación realizada, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose al finalizar, un producto listo para su puesta en producción. A continuación se detallan cada una de las iteraciones.

Durante el transcurso de las iteraciones se realiza la implementación de las Historias de Usuario seleccionadas para cada una de estas. Al inicio de las mismas, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando posteriormente cada una de estas a un equipo (o una persona) responsable de su implementación. Estas tareas son para el uso de los programadores, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente (12).

3.5 Tareas de Ingeniería.

3.5.1 Iteración 1.

Durante esta iteración se abordaron las Historias de Usuario de mayor prioridad, con el fin de obtener una versión del producto con algunas de las funcionalidades críticas para ser mostrado al cliente y obtener una rápida retroalimentación de este.

Historias de usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Preparar personalización.	1	1
Agregar o Quitar programas.	1	1
Subir paquete.	1	0.8
Crear personalización.	1	0.9
Descargar ISO.	1	0.7

Tabla 17. Distribución de Historias de Usuario para la Iteración 1.

Tarea de Ingeniería	Número Tarea: 03
Número Historia de Usuario: HU_04_Preparar Personalización.	
Nombre Tarea: Preparar Personalización	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 1
Fecha Inicio: 02/03/2009	Fecha Fin: 05/03/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Diseñar una interfaz e implementar la funcionalidad de Preparar personalización que es el primer paso para el proceso de personalización.	

Tabla 18. Tarea de Ingeniería Preparar Personalización.

Tarea de Ingeniería	Número Tarea: 04
Número Historia de Usuario: HU_05_Agregar o Quitar Programa	
Nombre Tarea: Agregar o Quitar Programas	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 0.8
Fecha Inicio: 06/03/2009	Fecha Fin: 14/03/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Diseñar e implementar la interfaz para agregar o quitar programas, donde se muestran los paquetes ordenados por grupo de paquetes con las opciones de ver los detalles de cada paquete y de instalar o desinstalar.	

Tabla 19. Tarea de Ingeniería Agregar o Quitar Programas.

Tarea de Ingeniería	Número Tarea: 05
Número Historia de Usuario: HU_05_Agregar o Quitar Programa	
Nombre Tarea: Listar repositorios disponibles.	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 0.2
Fecha Inicio: 16/03/2009	Fecha Fin: 16/03/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Se muestran los repositorios disponibles, su tipo y la dirección donde se encuentran los mismos	

Tabla 20. Tarea de Ingeniería Listar repositorios disponibles.

Tarea de Ingeniería	Número Tarea: 06
Número Historia de Usuario: HU_06_Crear personalización	
Nombre Tarea: Crear personalización	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 1
Fecha Inicio: 17/03/2009	Fecha Fin: 22/03/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Luego de haber terminado de realizarse la imagen deseada por el usuario se activa la opción de crear personalización donde se inicia la creación del archivo ISO que contendrá la imagen personalizada.	

Tabla 21. Tarea de Ingeniería Crear personalización.

Tarea de Ingeniería	Número Tarea: 07
Número Historia de Usuario: HU_07_Descargar ISO	
Nombre Tarea: Descargar ISO	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 1
Fecha Inicio: 23/03/2009	Fecha Fin: 27/03/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Después de haber terminado la creación de el ISO se le muestra al usuario para que pueda descargar dicho archivo.	

Tabla 22. Tarea de Ingeniería Descargar ISO.

Tarea de Ingeniería	Número Tarea: 08
Número Historia de Usuario: HU_08_Subir paquete	
Nombre Tarea: Subir paquete.	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 1
Fecha Inicio: 28/03/2009	Fecha Fin: 05/04/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Diseñar e implementar una interfaz para brindarle la posibilidad a los usuarios de que suban paquetes creados por ellos mismo o simplemente que no se encuentren en el repositorio para ser agregados a la personalización.	

Tabla 23. Tarea de Ingeniería Subir paquete.

Historias de usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Gestionar usuario.	1	1
Autenticar Usuario.	1	1
Registrar Usuario.	1	1
Gestionar Contraseña.	1	1

Tabla 24. Distribución de Historias de Usuario para la Iteración 2.

Tarea de Ingeniería	Número Tarea: 01
Número Historia de Usuario: HU_01_Gestionar Usuario.	
Nombre Tarea: Estudiar la librería FreakAuth_light.	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 1
Fecha Inicio: 06/04/2009	Fecha Fin: 11/04/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Realizar un estudio de la documentación de la librería FreakAuth_light para el marco de trabajo CodeIgniter para comprender y valorar el funcionamiento de la misma.	

Tabla 25. Tarea de Ingeniería Estudiar la librería FreakAuth_light.

Tarea de Ingeniería	Número Tarea: 02
Número Historia de Usuario: HU_01_Gestionar Usuario.	
Nombre Tarea: Instalar y configurar la librería FreakAuth_light.	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 1
Fecha Inicio: 12/04/2009	Fecha Fin: 15/04/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Luego de la instalación de esta librería hacemos las configuraciones necesarias para un correcto funcionamiento de las acciones que realiza la misma como son: registrarse, autenticarse, editar, eliminar y definir roles de los usuarios existentes.	

Tabla 26. Tarea de Ingeniería Instalar y configurar la librería FreakAuth_light.

Historias de usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Listar mis proyectos.	1	0.8
Gestionar repositorios	1	0.9

Tabla 27. Distribución de Historias de Usuario para la Iteración 3.

Tarea de Ingeniería	Número Tarea: 09
Número Historia de Usuario: HU_09_Listar mis proyectos.	
Nombre Tarea: Listar mis proyectos.	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 1
Fecha Inicio: 05/05/2009	Fecha Fin: 08/05/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Diseñar e implementar una interfaz donde se le muestre al usuario las plantillas	

Tabla 28. Tarea de Ingeniería Listar mis proyectos.

Tarea de Ingeniería	Número Tarea: 10
Número Historia de Usuario: HU_10_Gestionar Repositorios.	
Nombre Tarea: Gestionar Repositorios.	
Tipo de Tarea: (Desarrollo / Estudio)	Puntos Estimados: 1
Fecha Inicio: 10/05/2009	Fecha Fin: 17/05/2009
Programador Responsable: Roilán Zaballa Coca.	
Descripción: Diseñar e implementar una interfaz donde se muestren los repositorios de cada plantilla de configuración, con las opciones de editar y eliminar los mismos.	

Tabla 29. Tarea de Ingeniería Gestionar Repositorios.

3.5 Conclusiones.

En este capítulo se representaron las clases del diseño para mostrar la relación y responsabilidades de estas, además se desarrollaron las tareas correspondientes por cada Historia de Usuario para darle solución a las mismas.

Capítulo IV Pruebas de la Herramienta.

4.1 Introducción.

En el transcurso de las iteraciones se conforma la implementación de las Historias de Usuario (HU) seleccionadas para cada una de estas funcionalidades. Al comenzar dicho proceso se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando posteriormente cada una de estas un plazo de tiempo determinado para su implementación. Estas tareas son para el uso del programador, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente.

4.2 Disponibilidad del cliente.

Una de las pocas condiciones que impone la metodología XP es tener al usuario siempre disponible. No sólo para ayudar al equipo de desarrollo, sino formando parte de él. Todas las fases que se realizan en un proyecto XP requieren de comunicación con el usuario, preferiblemente cara a cara, en persona, sin intermediarios.

Durante la reunión del plan de entregas, el usuario propondrá qué historia de usuario se incluye en cada plan. También se negociarán los plazos de entrega. El usuario o cliente tomará las decisiones que le afecten para alcanzar los objetivos de su negocio.

También es necesario que el cliente colabore en la realización de los test. Estos test comprobarán que el sistema está listo para pasar a la fase de producción. El usuario comprobará los resultados obtenidos y tomará decisiones en cuanto a la utilización o no del sistema realizado.

- **Estándares de implementación.**

El código ha de ser desarrollado siguiendo los estándares de desarrollo para facilitar su lectura y modificación por cualquier miembro del equipo de desarrollo. Es decisiva, para poder plantear con éxito la propiedad colectiva del código. Ésta sería impensable sin una codificación basada en estándares que haga que todo el mundo se sienta cómodo con el código escrito por cualquier otro miembro del equipo.

- **Desarrollar unidad de pruebas.**

Cuando los test son creados antes que el código, la implementación del código será mucho más rápida. El tiempo empleado en desarrollar un test y algo de código para probarlo es aproximadamente el mismo tiempo que se emplea en crear exclusivamente dicho código. La creación de las unidades de test ayuda al programador a tener una visión a cerca del cómo, en definitiva, del comportamiento del programa. Además, aún se está a tiempo de dar marcha atrás, ya que el programador no ha concluido la implementación. Esta manera de trabajar resulta especialmente beneficiosa en el diseño de complicados sistemas software.

Cualquier característica de un programa para la que no haya un test automatizado, simplemente no existe.

4.3 Pruebas.

Uno de los pilares fundamentales de XP es el proceso de pruebas (12), el cual anima a los desarrolladores a probar constantemente y tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados así como el tiempo entre la introducción de este en el sistema y su detección (13). Todo esto contribuye a elevar la calidad de los productos desarrollados y la seguridad de los programadores a la hora de introducir cambios y modificaciones. XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores y encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida además de comprobar que dicha funcionalidad sea la esperada por el cliente (13).

4.3.1 Pruebas de Aceptación.

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las Historias de Usuario (13). Durante las iteraciones las HU seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable.

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Adicionar usuario.	
Descripción: Prueba para añadir un nuevo usuario al sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado como administrador.	
Entrada/ Pasos de ejecución: ✓ El administrador accede al módulo de usuarios. ✓ Llena todos los campos para adicionar el nuevo usuario. ✓ Hace clic en el botón Adicionar usuario.	
Resultado Esperado: Si la información es correcta y no existe el usuario registrado previamente entonces el registro es correcto.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 30. HU1_P1 Prueba de Aceptación Gestionar usuario.

Caso de Prueba de Aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Eliminar usuario.	
Descripción: Prueba para eliminar un usuario del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado como administrador.	
Entrada/ Pasos de ejecución: ✓ El administrador accede al módulo de usuarios. ✓ Hace clic en la imagen  Eliminar usuario. ✓ Se le muestra una alerta para verificar que realmente desea eliminar el usuario. ✓ Confirma o no la eliminación del usuario.	
Resultado Esperado: El usuario es eliminado de la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 31. HU1_P2 Prueba de Aceptación Gestionar usuario.

Caso de Prueba de Aceptación	
Código: HU1_P3	Historia de Usuario: 1
Nombre: Modificar usuario.	
Descripción: Prueba para modificar un usuario del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado como administrador.	
Entrada/ Pasos de ejecución: ✓ El administrador accede al módulo de usuarios. ✓ Hace clic en la imagen  Modificar usuario. ✓ Se busca el usuario en la base de datos y se muestran los datos de cada campo. ✓ Se cambia la información en el campo o en los campos que se desean modificar. ✓ Hace clic en el botón Guardar.	
Resultado Esperado: Los datos son modificados y guardados en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 32. HU1_P3 Prueba de Aceptación Gestionar usuario.

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Autenticar usuario con datos correctos.	
Descripción: Prueba para autenticarse en el sistema cuando el usuario, la contraseña y el código de seguridad son entrados correctamente.	
Condiciones de Ejecución:	
Entrada/ Pasos de ejecución: ✓ El usuario entra los datos y hace clic en el botón entrar. ✓ El sistema verifica los datos enviados. ✓ Se le muestra el mensaje de inicio de sesión.	
Resultado Esperado: El usuario ha iniciado correctamente sesión.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 33. HU2_P1 Prueba de Aceptación Autenticar usuario.

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Autenticar usuario con datos incorrectos.	
Descripción: Prueba para autenticarse en el sistema cuando el usuario entra incorrectamente unos de los datos para poder iniciar sesión.	
Condiciones de Ejecución:	
Entrada/ Pasos de ejecución: ✓ El usuario entra los datos y hace clic en el botón entrar. ✓ El sistema verifica los datos enviados. ✓ Se le muestra un mensaje indicándole el error cometido.	
Resultado Esperado: Mientras que el usuario no entra los datos correctamente se le muestra un mensaje indicándole el error y además se le señala el campo donde está el error.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 34. HU2_P2 Prueba de Aceptación Autenticar usuario.

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Registrar usuario con datos correctos.	
Descripción: Prueba para registrarse en el sistema cuando los datos entrados por el usuario son de forma correcta.	
Condiciones de Ejecución:	
Entrada/ Pasos de ejecución: ✓ El usuario entra los datos y hace clic en el botón estoy de acuerdo.. ✓ El sistema verifica los datos enviados. ✓ Se le muestra el mensaje. ✓ Se le envía un e-mail con los datos del registro.	
Resultado Esperado: El usuario reciba el correo con los datos de su cuenta para poder activarla.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 35. HU3_P1 Prueba de Aceptación Registrar usuario.

Caso de Prueba de Aceptación	
Código: HU3_P2	Historia de Usuario: 3
Nombre: Autenticar usuario con datos incorrectos.	
Descripción: Prueba para registrarse en el sistema cuando los datos entrados por el usuario son de forma incorrecta.	
Condiciones de Ejecución:	
Entrada/ Pasos de ejecución: ✓ El usuario entra los datos y hace clic en el botón estoy de acuerdo. ✓ El sistema verifica los datos enviados. ✓ Se le muestra un mensaje indicándole el error cometido.	
Resultado Esperado: Mientras que el usuario no entra los datos correctamente se le muestra un mensaje indicándole el error y además se le señala el campo donde está el error.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 36. HU3_P2 Prueba de Aceptación Registrar usuario.

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4
Nombre: Cambiar contraseña con datos correctos.	
Descripción: Prueba para que un usuario cambie su contraseña cuando los datos entrados están correctos.	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ El usuario entra los datos y hace clic en el botón enviar. ✓ El sistema verifica los datos enviados. ✓ Se le muestra un mensaje.	
Resultado Esperado: La contraseña ha sido cambiada correctamente.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 37. HU4_P1 Prueba de Aceptación Gestionar contraseña.

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: 4
Nombre: Cambiar contraseña con datos incorrectos.	
Descripción: Prueba para verificar que las contraseñas de los usuarios sean modificadas correctamente previéndoles de los errores que puedan cometer.	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ El usuario entra los datos y hace clic en el botón enviar. ✓ El sistema verifica los datos enviados. ✓ Se le muestra un mensaje con los errores cometidos.	
Resultado Esperado: La contraseña no es modificada hasta que los datos entrados por el usuario no son entrados correctamente.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 38. HU4_P2 Prueba de Aceptación Gestionar contraseña.

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Agregar paquetes.	
Descripción: Con esta prueba se puede ver que sean agregados los paquetes que deseamos en la imagen personalizada al archivo XML de configuración para que sean instalados posteriormente.	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ El usuario hace clic en el botón instalar. ✓ El paquete es agregado al XML.	
Resultado Esperado: El paquete es agregado correctamente al XML para ser instalado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 39. HU5_P1 Prueba de Aceptación Agregar o Quitar programas.

Caso de Prueba de Aceptación	
Código: HU5_P2	Historia de Usuario: 5
Nombre: Quitar paquetes.	
Descripción: Con esta prueba se puede ver que los paquetes que están en el archivo XML de configuración sean quitados para que no se instalen.	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ El usuario hace clic en el botón desinstalar. ✓ El paquete es quitado del XML.	
Resultado Esperado: El paquete es quitado correctamente del archivo XML.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 40. HU5_P2 Prueba de Aceptación Agregar o Quitar programas.

Caso de Prueba de Aceptación	
Código: HU5_P3	Historia de Usuario: 5
Nombre: Ver detalles del paquete.	
Descripción: Prueba para comprobar que las descripciones detalladas de cada paquete sea mostrada correctamente.	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ El usuario hace clic en el botón detalles. ✓ El sistema busca los detalles del paquete. ✓ Se le muestra una descripción detallada del paquete.	
Resultado Esperado: Qué el usuario vea la descripciones de los paquetes.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 41. HU5_P3 Prueba de Aceptación Agregar o Quitar programas.

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Crear personalización.	
Descripción:	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ El usuario hace clic en el menú en Crear Personalización. ✓ El sistema comienza a crear el ISO personalizado. ✓ Se le muestra al usuario el ISO con la opción para descargar.	
Resultado Esperado: El ISO es creado correctamente	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 42. HU6_P1 Prueba de Aceptación Crear Personalización.

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Descargar ISO.	
Descripción: Luego de realizar cualquier personalización se le muestran los ISO que el usuario tiene creados con la opción de descargar.	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ Se le muestra a los usuarios los ISO creados. ✓ El usuario hace clic en el botón descargar. ✓ Selecciona el lugar donde desea guardar el ISO y comienza la descarga.	
Resultado Esperado: El usuario pueda descargar los ISO creados.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 43. HU7_P1 Prueba de Aceptación Descargar ISO.

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: 8
Nombre: Subir paquete.	
Descripción: Prueba para la subida de paquetes de los usuarios	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ El usuario hace clic en el botón examinar y busca el paquete. ✓ El sistema sube el paquete para la carpeta del usuario. ✓ Se le muestra un mensaje.	
Resultado Esperado: El paquete es subido correctamente al servidor para poder agregarlo al archivo XML de configuración.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 44. HU8_P1 Prueba de Aceptación Subir paquete.

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Listar mis proyectos.	
Descripción: Prueba para comprobar que cuando un usuario inicia sesión se le muestren todos los proyectos creados por el.	
Condiciones de Ejecución: Estar autenticado.	
Entrada/ Pasos de ejecución: ✓ Luego de autenticarse se le muestran los proyectos creados.	
Resultado Esperado: Que se le muestren al usuario todos los proyectos que el ha creado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 45. HU9_P1 Prueba de Aceptación Listar mis plantillas.

Caso de Prueba de Aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Modificar repositorio.	
Descripción: Prueba para verificar los datos de un repositorio sean modificados correctamente	
Condiciones de Ejecución: Estar autenticado como administrador.	
Entrada/ Pasos de ejecución: ✓ El administrador accede al módulo de repositorios. ✓ Hace clic en la imagen  Modificar repositorio. ✓ Se cambia la información en el campo o en los campos que se desean modificar. ✓ Hace clic en el botón Guardar.	
Resultado Esperado: Se ha modificado el repositorio en la plantilla básica de configuración.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 46. HU10_P1 Prueba de Aceptación Gestionar repositorios.

Caso de Prueba de Aceptación	
Código: HU10_P2	Historia de Usuario: 10
Nombre: Eliminar repositorio.	
Descripción: Prueba para eliminar un repositorio de una plantilla básica.	
Condiciones de Ejecución: Estar autenticado como administrador.	
Entrada/ Pasos de ejecución: ✓ El administrador accede al módulo de repositorios. ✓ Hace clic en la imagen  Eliminar repositorio. ✓ Se le muestra una alerta para verificar que realmente desea eliminar el repositorio. ✓ Confirma o no la eliminación del repositorio.	
Resultado Esperado: El repositorio es eliminado de la plantilla básica.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 47. HU10_P2 Prueba de Aceptación Gestionar repositorios.

4.4 Conclusiones.

En este capítulo se desarrollaron las pruebas de aceptación para verificar que el sistema implementado funcione correctamente, para comprobar la seguridad e integridad de la aplicación ya que estas pruebas le proporcionan al cliente la conformidad con el producto.

Conclusiones Generales

Luego de la investigación realizada se puede afirmar que se cumplieron los objetivos propuestos, arribando a las siguientes conclusiones:

- Al realizar un análisis detallado de los aspectos teóricos que rigen la investigación se hizo evidente la necesidad de desarrollar una herramienta web que permita a los usuarios del MININT llevar a cabo la personalización del sistema operativo SUSE.
- Ante la necesidad de desarrollar dicha herramienta se precisó entonces de la implementación de una metodología que guió el proceso de desarrollo de la aplicación.
- La planificación y diseño de las funcionalidades expuestas propició una vía de solución que guió el proceso de desarrollo de la herramienta *“MinPSUSE”*.
- Con la realización de este trabajo se logró desarrollar una herramienta para el MININT que permite la personalización de distribuciones de la familia SUSE.

Recomendaciones

A lo largo del desarrollo del presente trabajo, han ido surgiendo ideas que podrían implementarse en un futuro, de forma que se logre una aplicación más útil y efectiva, para lo cual se recomienda:

- ✓ Elaborar un instalador que permita de forma automática la instalación de la aplicación.
- ✓ Desarrollar el módulo de ayuda que guíe a los usuarios en todo el proceso de personalización.
- ✓ Desarrollar un módulo de configuración que le permita a los usuarios en todo el proceso de personalización crear configuraciones como:
 1. Adicionar usuarios del sistema personalizado.
 2. Configurar la red.
 3. Cambiar el fondo de escritorio.

Referencias Bibliográficas

1. "La Definición de Software Libre - Proyecto GNU - Free Software Foundation (FSF)," [En línea] 17 de Enero de 2009. [Citado el: 21 de Enero de 2009.] <http://www.gnu.org/philosophy/free-sw.es.html>
2. "Enlaces a otros sitios sobre software libre - Proyecto GNU - Free Software Foundation (FSF)," [En línea] 11 de Diciembre de 2008. [Citado el: 25 de Enero de 2009.] <http://www.gnu.org/links/links.es.html#FreeGNULinuxDistributions>.
3. "YaST/About - openSUSE," [En línea] 03 de Octubre de 2007. [Citado el: 3 de Febrero de 2009.] <http://es.opensuse.org/YaST/About>
4. Schaefer, Marcus. January 27, 2009. *openSUSE - KIWI Image System*. January 27, 2009.
5. "Apache httpd Modules - The Apache HTTP Server Project," [Citado el: 12 de Febrero de 2009.] <http://httpd.apache.org/modules/>
6. "MySQL - Wikipedia, la enciclopedia libre," [Citado el: 18 de Febrero de 2009.] <http://es.wikipedia.org/wiki/MySQL>
7. "phpMyAdmin - Wikipedia, la enciclopedia libre," [Citado el: 19 de Febrero de 2009.] <http://es.wikipedia.org/wiki/PhpMyAdmin>
8. "Guía Breve de Tecnologías XML," [Citado el: 19 de Febrero de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>
9. "Qué es Javascript," [Citado el: 21 de Febrero de 2009.] <http://www.desarrolloweb.com/articulos/25.php>
10. "Hojas de Estilo Web," [Citado el: 27 de Febrero de 2009.] <http://www.spanish-translator-services.com/espanol/t/Style/>
11. "PHP: News Archive - 2007," [Citado el: 28 de Febrero de 2009.] <http://www.php.net/archive/2007.php>
12. Beck, Kent. 2000. *Extreme Programming Explained*. Addison : Wesley, 2000.
13. Crispin, Lisa y House. 2002. *Testing Extreme Programming*. Addison : Wesley, 2002.

Bibliografía

Beck, Kent. 2000. *Extreme Programming Explained*. Addison : Wesley, 2000.

2007. Creating Operating System Images. [En línea] 13 de Febrero de 2007. [Citado el: 12 de Marzo de 2009.] <https://build.opensuse.org/documentation/obs/sec.kiwi.create.html>.

Crispin, Lisa y House. 2002. *Testing Extreme Programming*. Addison : Wesley, 2002.

2009. Live USB stick - openSUSE. [En línea] Novell, Marzo de 29 de 2009. [Citado el: 1 de Abril de 2009.] http://en.opensuse.org/Live_USB_stick.

2005. proyecto de desarrollo del Software. [En línea] 9 de Enero de 2005. [Citado el: 10 de Febrero de 2009.] <http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemploxp/index.html>.

Schaefer, Marcus. January 27, 2009. *openSUSE - KIWI Image System*. January 27, 2009.

Whittaker, Roger y Davies, Justin. 2008. *openSUSE® 11.0 and SUSE® Linux® Enterprise Server Bible*. 2008.

2008. Zypper/Uso/11.0 - openSUSE. [En línea] Novell, 29 de Junio de 2008. [Citado el: 5 de Marzo de 2009.] <http://es.opensuse.org/Zypper/Uso/11.0>.