

Universidad de las Ciencias Informáticas

Facultad 10



Sistema para la Factoría de CDs de instalación del
sistema operativo NovaFAR.

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

Autores:

Elisa Gamez López.
Carlos Alberto García Rodríguez.

Tutor:

Ing. Keiver Hernández Fernández.

Co-Tutor:

Ing. Yordan Vélez Rodríguez.

Ciudad de la Habana, mayo 2010.

"Año 52 de la Revolución"

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los _ días del mes de mayo del año 2010.

Elisa Gamez López.

Carlos Alberto García Rodríguez.

Firma de los Autores

Firma de los Autores

Ing. Keiver Hernández Fernández.

Firma del Tutor.

El mundo es grande para los valientes, solo los cobardes lloran creyendo en la mala suerte!...

DATOS DE CONTACTO

Autora: Elisa Gamez López.

Correo electrónico: egamez@estudiantes.uci.cu

Autor: Carlos Alberto García Rodríguez.

Correo electrónico: cgarcia@estudiantes.uci.cu

Tutor: Keiver Hernández Fernández.

Especialidad de graduación: Ingeniería en Ciencias Informáticas.

Categoría docente: Profesor Recién Graduado.

Años de experiencia: 1 año.

Años de graduado: 1 año.

Correo electrónico: khfernandez@uci.cu

Co_Tutor: Yordan Vélez.

Especialidad de graduación: Ingeniería en Ciencias Informáticas.

Años de experiencia: 3 años.

Años de graduado: 3 años.

Correo electrónico: yvelez@uci.cu

Largo y espinoso nos parece el camino cuando vislumbramos tan lejos la meta: se resquebraja la voluntad y flaquean las piernas.

Pero cuando la noche se nos hace muy oscura, la mano amiga, el consejo oportuno, el hombre que se nos ofrece solidario y firme, hacen que renazcan las fuerzas y se aligere la marcha.

A esos que durante estos años estuvieron a mi lado, que me apoyaron e hicieron que la noche se volviera día, mi profundo agradecimiento...

Elisa y Carlos...

Elisa Gamez López

Agradezco:

*A un gran amigo que hizo posible que hoy estuviese cumpliendo el sueño de mis padres y que estuviera enfrentando esta nueva etapa de la vida, **Elvis** te debo gran parte de mí...*

*Al negro que diseñó las fotos de mi herramienta, y que yo lo quiero mucho, **Mendosasa**...*

*A mi gran amigo, el rey de la consola, mi tutor **Keiver**, te adoro!...*

*A la persona que siempre supo darme su mano, correr conmigo y demostrarme que hay verdaderos amigos que no solo están en las buenas y que de casualidad es mi oponente, **Yosbel Ernesto** gracias por todo...*

*A todos aquellos **profes** que de una manera hicieron que me cayera y a otros que me ayudaron a levantar, de ellos aprendí a vivir...*

*A mi segunda mamá, **mi suegra**, por enseñarme tanto de la vida, quererme y hacer el flan más rico del mundo...*

*A mi segundo papá, **mi suegro**, por alimentarnos tanto, por ser el viejito mas peleón de trinidad, y por demostrarme su confianza y amor...*

*A mis tíos **Toní y Tato**, por ser personas claves en mi formación...*

*A los **vecinos** q siempre estuvieron al tanto de mí...*

En especial:

*A mi Ángel de la guarda, mi **abuela** y mi **bisabuela** Clara, ya que sin ellas no podría vivir ni tener firmeza en esta vida...*

*A la **madre** más linda y maravillosa del mundo, por soportarme, malcriarme, hacerme su hermana y aguantarme siempre con una sonrisa...te amo!...*

*A mi **papito** lindo, por dejarme nacer, y porque nunca va a saber lo mucho que yo lo amo...*

*Y sin tener palabras, a la persona más maravillosa del mundo, a la que me enseñó a creer en mí, a amar, a pelear, a ser yo sin importarme nada, al nene más lindo que pueda existir en el mundo, a mi novío **Redecto**...te amaré por siempre...!!!*

Carlos Alberto García Rodríguez

*Les agradezco a todos mis **amigos** que han estado cerca cuando los he necesitado en especial a **Jorge Luis, Miguel A, Alexander**. A mis **profesores** y **compañeros** que me han ayudado a la largo de la carrera.*

*A mi **familia** que siempre se ocupó de mí y me ayudó a salir adelante. Especialmente a mi **madre** que siempre ha estado presente en los momentos que la he necesitado.*

A los tutores por la gran ayuda y apoyo que nos han brindado, sin ellos nada de esto hubiera sido posible.

A los vecinos del barrio que estuvieron pendiente siempre de mis resultados.

A mi novia por apoyarme siempre y darme aliento para seguir adelante.

A todas las personas que me han aportado su granito de arena en mi formación.

A mi tía Anita, y primos Pura y Pedrí que me apoyaron en esta temporada por la capital.

Con todo el amor del mundo dedicamos este trabajo:

A la persona que quisiera tener mucho más a mi lado, Mi Papá...

A mi gran tesoro, a la persona que más amo, que siempre confió en mí, me apoyó sin condición y que sin ella yo no viviría, Mi mamita...

Al amor de mi vida, Redecto Rodríguez Castillo...

Y a la viejita más linda del mundo, Mi Abuela Clara...

Elisa Gamez López...

A mi madre por confiar siempre en mí y apoyarme incondicionalmente. Sin ella no hubiese logrado llegar tan lejos; es la mejor madre del planeta.

A mi hermana por lograr hacerme la vida menos aburrida con nuestras peleas de hermanos.

En fin se las dedico a todas las personas que han querido siempre lo mejor para mí.

Carlos Alberto García Rodríguez

En la actualidad es muy común que una determinada empresa o institución necesite probar una aplicación o distribución de Linux; una de las formas más usadas es la creación de un LiveCD/DVD que contenga dicha aplicación. Luego el LiveCD/DVD obtenido puede usarse en cualquier ordenador sin necesidad de instalar en los discos duros absolutamente nada. Otro de los usos ampliamente extendidos de los LiveCD/DVDs es que se pueden usar como herramientas de rescate en caso de que ocurra algún error en el sistema de un ordenador determinado.

En el presente trabajo se estudian las diferentes herramientas para la creación de LiveCD/DVDs que existen. Además se realiza un análisis del funcionamiento y características fundamentales de cada uno de estos sistemas y las herramientas empleadas para su desarrollo. Como parte del trabajo se lleva a cabo el diseño y la implementación de una aplicación gráfica para la generación de LiveCD/DVDs a partir de sistemas Nova GNU/Linux que pudiera facilitar el trabajo al usuario. La aplicación desarrollada copia el sistema original y lo comprime usando un sistema de ficheros squashfs el cual permite grandes ratios de compresión. El resultado final es una imagen ISO lista para ser quemada en un CD o un DVD.

PALABRAS CLAVES

Kernel, Linux, LiveCD/DVD.

Índice

ÍNDICE DE TABLAS.	IXI
INTRODUCCIÓN.	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL SOFTWARE NOVAFAR LIVE.	5
SISTEMAS DE CREACIÓN DE LIVECD/DVDS DE DISTRIBUCIONES GNU/LINUX.	5
LiveCD.	5
Proceso de arranque de un LiveCD/DVD.....	6
Funcionamiento de los sistemas de creación de LiveCD/DVD.	7
1.2. SISTEMAS DE CREACIÓN DE LIVECD/DVD.	8
1.2.1. <i>Dfsbuild</i>	8
1.2.2. <i>Bootcd</i>	9
1.2.3. <i>LiveCD-Tools</i>	9
1.2.4. <i>Live-magic</i>	10
1.2.5. <i>Live-Helper</i>	11
1.2.6. <i>Remastersys</i>	11
1.2.7. <i>Linux Live Scripts</i>	12
1.2.8. <i>UCK</i>	12
1.2.9. <i>Reconstructor</i>	13
1.2.10. <i>Nova LiveCD Kit</i>	13
1.2.11. <i>Calculate-Linux Scratch (CLS)</i>	14
1.2.12. <i>Comparación entre las herramientas utilizadas en el centro (NovaLiveCD Kit y Calculate Linux)</i>	14
Herramienta NovaLiveCD Kit.....	14
Herramienta Scratch Calculate-Linux (CLS).	15
1.3. HERRAMIENTAS, LENGUAJES Y METODOLOGÍAS A USAR.	15
1.3.1. <i>Framework Qt</i>	16
Potencialidades de Qt 4.	16
Qt Creator.	17
Qt Designer.	18
Signal y Slot.	18
1.3.2. <i>Mksquashfs</i>	19
1.3.3. <i>Aufs</i>	19
1.3.4. <i>Squashfs</i>	20
1.3.5. <i>Rsync</i>	21
1.3.6. <i>Lenguaje para el modelado (UML)</i>	21
1.3.7. <i>Intérprete de Comandos (Bash)</i>	22
1.3.8. <i>Lenguaje de Programación (C/ C++)</i>	22
1.3.9. <i>Metodología de desarrollo (SXP)</i>	25
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SOFTWARE NOVAFAR LIVE.	27
2.1. ROLES DEL PROYECTO.	27
2.2. MODELO DE DOMINIO.	28
Figura 1. Modelo de Dominio.	28

2.3. HISTORIAS DE USUARIO Y PROTOTIPO DE INTERFAZ.....	28
2.4. DIAGRAMA DE CLASES.....	42
Figura 2: Diagrama de Clases.....	42
2.5. DIAGRAMA DE COMPONENTES.....	43
Figura 3: Diagrama de Componentes.....	43
2.6. PLAN DE RELEASES.....	43
CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DEL SOFTWARE NOVAFAR LIVE.....	45
3.1. CASOS DE PRUEBAS.....	45
3.2. PROPUESTA DEL SISTEMA A IMPLEMENTAR.....	45
3.3. ARQUITECTURA DE LA APLICACIÓN.....	46
3.4. LISTADO DE HISTORIAS DE USUARIOS A PROBAR.....	47
3.4.1. Caso de Prueba Historias de Usuarios: HU-NFL-1.....	47
3.4.2. Caso de Prueba Historias de Usuarios: HU-NFL-2.....	48
3.4.3. Caso de Prueba Historias de Usuarios: HU-NFL-3.....	50
3.4.4. Caso de Prueba Historias de Usuarios: HU-NFL-4.....	53
3.4.5. Caso de Prueba Historias de Usuarios: HU-NFL-5.....	55
3.4.6. Caso de Prueba Historias de Usuarios: HU-NFL-6.....	55
3.5. Resultados Obtenidos.....	58
3.5.1. Acerca del tiempo de desarrollo.....	58
3.5.2. Acerca de las Funcionalidades Obtenidas.....	58
CONCLUSIONES.....	60
RECOMENDACIONES.....	61
REFERENCIAS BIBLIOGRÁFICAS.....	62
BIBLIOGRAFÍA.....	64
GLOSARIO.....	65
ANEXOS.....	69
ANEXO # 1: LISTA DE RESERVA DEL PRODUCTO.....	69
ANEXO #2: PLANTILLA DE PLAN DE PRUEBA PARA NOVAFAR LIVE.....	72

Índice de Tablas.

Tabla 1. Roles del proyecto.....	27
Tabla 2. HU Programar Scripts Auxiliares.....	29
Tabla 3. Tarea de Ingeniería #1 de la HU Programar Scripts Auxiliares.....	30
Tabla 4. Tarea de Ingeniería #2 de la HU Programar Scripts Auxiliares.....	30
Tabla 5. Tarea de Ingeniería #3 de la HU Programar Scripts Auxiliares.....	31
Tabla 6. HU Copia del sistema origen.....	32
Tabla 7. Tarea de Ingeniería #1 de la HU Copia del sistema origen.....	32
Tabla 8. Tarea de Ingeniería #2 de la HU Copia del sistema origen.....	33
Tabla 9. Tarea de Ingeniería #3 de la HU Copia del sistema origen.....	34
Tabla 10. HU Compactar Imagen.....	34
Tabla 11. Tarea de Ingeniería #1 de la HU Compactar Imagen.....	35
Tabla 12. Tarea de Ingeniería #2 de la HU Compactar Imagen.....	35
Tabla 13. HU Crear imagen ISO.....	36
Tabla 14. Tarea de Ingeniería #1 de la HU Crear imagen ISO.....	37
Tabla 15. Tarea de Ingeniería #2 de la HU Crear imagen ISO.....	37
Tabla 16. HU Adicionar Perfil.....	38
Tabla 16. Tarea de Ingeniería #1 de la HU Adicionar Perfil.....	39
Tabla 17. Tarea de Ingeniería #2 de la HU Adicionar Perfil.....	39
Tabla 18. HU Configuración.....	40
Tabla 19. Tarea de Ingeniería #1 de la HU Configuración.....	40
Tabla 20. Tarea de Ingeniería #2 de la HU Configuración.....	41
Tabla 21. Tarea de Ingeniería #3 de la HU Configuración.....	41
Tabla 22. Plan de releases.....	44
Tabla 23: Caso de Prueba de Aceptación #1 de la HU: NFL-1.....	48
Tabla 24: Caso de Prueba de Aceptación #2 de la HU: NFL-1.....	48
Tabla 25: Caso de Prueba de Aceptación #1 de la HU: NFL-2.....	49
Tabla 26: Caso de Prueba de Aceptación #2 de la HU: NFL-2.....	50
Tabla 27: Caso de Prueba de Aceptación #3 de la HU: NFL-2.....	50
Tabla 28: Caso de Prueba de Aceptación #1 de la HU: NFL-3.....	51
Tabla 29: Caso de Prueba de Aceptación #2 de la HU: NFL-3.....	52
Tabla 30: Caso de Prueba de Aceptación #3 de la HU: NFL-3.....	52
Tabla 31: Caso de Prueba de Aceptación #1 de la HU: NFL-4.....	53
Tabla 32: Caso de Prueba de Aceptación #2 de la HU: NFL-4.....	54
Tabla 33: Caso de Prueba de Aceptación #3 de la HU: NFL-4.....	55
Tabla 34: Caso de Prueba de Aceptación #1 de la HU: NFL-5.....	55
Tabla 35: Caso de Prueba de Aceptación #1 de la HU: NFL-6.....	56
Tabla 36: Caso de Prueba de Aceptación #2 de la HU: NFL-6.....	57
Tabla 37: Caso de Prueba de Aceptación #3 de la HU: NFL-6.....	58

Introducción.

Uno de los aspectos más importantes del mundo contemporáneo es el acelerado desarrollo de la ciencia y la tecnología; dentro de esta, la producción de software que se ha convertido en una importante vía de ingresos económicos. En dicha actividad han surgido un gran número de modelos, procesos, tecnologías, teorías y normas a seguir, en busca de lograr mejoras en la productividad del proceso de desarrollo. Este desarrollo de software viene liderado por dos corrientes fundamentales; el desarrollo cerrado o privativo y el desarrollo libre.

El movimiento de software libre surgió cuando Richard Stallman anunció el proyecto GNU (GNU is Not Unix). La meta de este movimiento es dar libertades a los usuarios sobre los programas que usan. De acuerdo con la *Free Software Foundation(FSF)*, el *software libre* se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, cambiar y mejorar el software; para un mejor entendimiento, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa con cualquier propósito; estudiar el funcionamiento del mismo y adaptarlo a las necesidades propias, además de distribuir copias de dichas modificaciones, lo que puede ayudar a otros usuarios y finalmente la libertad de optimizar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie. [1]

Con el objetivo de convertir al país en una potencia en el desarrollo de software se crea al calor de la Batalla de Ideas, una nueva universidad, la Universidad de Ciencias Informáticas (UCI), como dijera el Comandante en jefe Fidel: *“Los objetivos de esta institución son altamente estratégicos. Al enemigo le va a preocupar diez veces más que cuando era un centro de exploración radioelectrónica, porque este es un centro estratégico del futuro y para el desarrollo del país”*. [2]

La UCI, con el objetivo de lograr un mejor desarrollo productivo, y cumplir con la meta fundamental por la que fue creada, se ha dividido en 10 facultades, cada una de estas con un perfil de trabajo diferente, pero orientadas todas al desarrollo informático.

La Facultad 10 se dedica principalmente al desarrollo de aplicaciones libres y a brindar soporte a sistemas de esta índole. Además entre sus tareas se destacan: la formación continua de profesionales integrales y revolucionarios especializados en el campo del software libre, al mismo tiempo de desarrollar investigaciones en esta área para impulsar el uso del software libre en toda la comunidad universitaria así como ser pionera en la migración de muchas instituciones a lo largo de

todo el país. Es precisamente en el seno de esta facultad que nace la Comunidad de Software Libre de la UCI.

Las Fuerzas Armadas Revolucionarias (FAR), es una de las instituciones que ha venido dando pasos en el uso, desarrollo y explotación de herramientas libres.

- 1- Los directivos ordenan la utilización de herramientas en todos sus organismos y precisan que tanto a directivos como a los propios implicados asuman correctamente la tarea.
- 2- Lleva muy de cerca el proceso de desarrollo de la distribución cubana de Software Libre (Nova).
- 3- Hereda las ventajas del software libre, tiene más seguridad y agilidad en el desarrollo de trabajos.

Nova GNU/Linux es un proyecto perteneciente a la UCI cuya misión y objetivo fundamental es la creación y el mantenimiento de una distribución basada en la tecnología GNU/Linux. Dicha distribución ha devenido en los últimos meses como uno de los productos informáticos más populares y controversiales por su contribución a la independencia tecnológica, a la migración a tecnologías de software libre y código abierto así como a la sustitución de importaciones por concepto de licencias de software y reutilización de hardware obsoleto.

Con el sistema operativo Nova GNU/Linux, se pretende un impacto positivo en la sociedad contribuyendo a obtener un estatus de independencia y soberanía tecnológica; enfocarse en el proceso de informatización de la sociedad y responder a las necesidades reales de las empresas y organizaciones cubanas; contribuir a crear una comunidad de desarrollo que fomente la socialización del conocimiento y los resultados alcanzados, entre muchas otras tareas que a lo largo del camino irán apareciendo con buenas intenciones.[3]

Las distribuciones de software libre son un elemento fundamental en este movimiento de software libre, muchas son las herramientas y sistemas que se usan para darle funcionalidades a las mismas, así como para desarrollarlas. Es muy común que una determinada empresa o institución necesite una distribución con características específicas para su uso, como probar aplicaciones o sistemas sin necesidad de instalar dicha distribución en la computadora.

Para lograr este comportamiento, una de las formas más usadas es la creación de un *LiveCD/DVD* que contenga dicha aplicación o sistema. Un *LiveCD/DVD* puede usarse en cualquier ordenador sin necesidad de instalar en los discos duros absolutamente nada, además los *LiveCD/DVD* se pueden usar como herramientas de rescate en caso de que ocurra algún error en el sistema de un ordenador determinado.

La Unidad de Compatibilización, Integración y Desarrollo de Productos para la Defensa (UCID), de conjunto con la Facultad 10 decidió migrar a un sistema operativo libre, debido a que se adquieren nuevas facilidades que no permite el software privativo, por ejemplo, la libertad de tener el código fuente para poder modificarlo y adaptarlo a las necesidades propias de la institución, así como el porcentaje de seguridad que brinda tener un software al cual se le puede auditar el código y añadirle conocimiento propio.

Es por ello que la migración a un sistema operativo creado por cubanos (Nova GNU/Linux), y compatibilizado por la UM-1722 (Unidad Militar-1722), fomenta los términos de seguridad informática que se requieren en las FAR. El desarrollo y compatibilización de este sistema operativo, permite adaptarlo a las necesidades y requisitos propios de entidades militares y entornos seguros de desarrollo. Es de interés, además, promover estos pasos en la línea de servidores, para así mejorar los procesos de gestión en los centros de datos. El centro UCID fue el lugar designado por las FAR para compatibilizar Nova con el objetivo de producir el sistema operativo NovaFAR.

Actualmente el proceso de creación de los *LiveCD/DVDs* de NovaFAR es muy engorroso y la herramienta que se usa no cumple con todas las funcionalidades que se necesita, ya que no posee una interfaz gráfica agradable que permita desarrollar este trabajo y además, es extremadamente complejo su uso, ya que está hecha en Bash y requiere de conocimientos propios del lenguaje; es por esto que se requiere de otra herramienta que sea más factible para el equipo de desarrollo en el proceso de generación de *LiveCD/DVDs* del sistema operativo NovaFAR. Teniendo en cuenta estos aspectos, el centro UCID decide llevar a cabo el desarrollo de una aplicación para la creación de *LiveCD/DVDs* que permita mejorar el proceso de instalación y despliegue del mismo.

El **problema a resolver** está basado en que la herramienta empleada para la creación de *LiveCD/DVDs* de instalación del sistema operativo NovaFAR no satisface los requerimientos.

El **objeto de estudio** está enmarcado en los procesos de creación de *LiveCD/DVDs*, por lo que el **campo de acción** es el proceso de automatización de la generación de *LiveCD/DVDs* para NovaFAR.

Surge entonces la **idea a defender** fundamentada en que el desarrollo de la aplicación propuesta permitirá generar *LiveCD/DVDs* de instalación para el sistema operativo NovaFAR de una forma más factible cumpliendo con las necesidades existentes en el centro UCID.

El **objetivo general** del presente trabajo es desarrollar una aplicación que permita la creación de *CD/DVDs* de instalación para el sistema operativo NovaFAR.

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

- ✓ Elaborar el marco teórico sobre las diferentes herramientas para generar LiveCD/DVDs de NovaFAR GNU/Linux y otras distribuciones importantes en Cuba y el mundo.
- ✓ Diseñar la aplicación de forma tal que sea robusta y extensible para generar LiveCD/DVDs.
- ✓ Implementar la aplicación propuesta con todas sus funcionalidades en correspondencia con las características diseñadas para la utilización de una herramienta que genere LiveCD/DVDs.
- ✓ Probar las principales funcionalidades de la aplicación.

La presente investigación está estructurada en 3 capítulos, que incluye todo lo relacionado con el trabajo investigativo sobre las herramientas para la creación de *LiveCD/DVDs*, así como el diseño, implementación y pruebas de la aplicación propuesta. A continuación se muestra una breve descripción de cada uno de los capítulos.

Capítulo 1: *Fundamentación Teórica*: En este capítulo se introducirán conceptos como la creación de un LiveCD/DVD, también se incluirá los tipos de aplicaciones usadas en el mundo y en la UCI y el proceso general que siguen la mayoría de estos sistemas así como sus características. Además se explican las diferencias de su funcionamiento entre distintas distribuciones de GNU/Linux.

Capítulo 2: *Análisis y Diseño*: En este capítulo se realiza el desarrollo ágil de la aplicación propuesta. Se explica la dinámica del proyecto en forma de historias de usuarios, prototipos de interfaz de usuario y algunos modelos auxiliares además del plan de releases para las entregas.

Capítulo 3: *Validación de la solución propuesta*: Finalmente en este capítulo se explica el proceso de implementación y se modelan los casos de pruebas a las que fue sometida la aplicación en cada una de las iteraciones. Se exponen los resultados obtenidos y se muestran las funcionalidades alcanzadas en el período de desarrollo.

Capítulo 1: Fundamentación Teórica.

En el presente capítulo se llevará a cabo el estudio de las herramientas que se analizarán como parte del marco teórico, así como los lenguajes y tecnologías escogidas para darle solución al problema planteado.

Sistemas de creación de LiveCD/DVDs de distribuciones GNU/Linux.

LiveCD.

En la actualidad las organizaciones e instituciones necesitan de distribuciones de sistemas operativos libres que satisfagan sus necesidades, ante la ocurrencia de fallos en el sistema. Es usual que se creen LiveCD cuando surge una nueva distribución ya que es un método para difundir y distribuir dicho sistema, de esta manera el usuario puede probar dicha distribución sin necesidad de instalarla en el disco duro.

Un LiveCD o LiveDVD, es un sistema operativo (normalmente acompañado de un conjunto de aplicaciones) almacenado en un medio extraíble, tradicionalmente un CD o un DVD (de ahí sus nombres), que puede ejecutarse sin necesidad de instalarlo en el disco duro de una computadora, el cual usa la memoria de acceso aleatorio (RAM) para simular un medio de escritura.

De esta forma solamente hace falta introducir el CD o DVD en el ordenador y encenderlo, al cabo de unos minutos se tendrá un sistema Linux funcionando en el mismo. Este tipo de distribuciones solamente sirve para demostraciones y pruebas, ya que una vez que se apague el ordenador, todo lo que se ha hecho desaparece.

Algunas distribuciones del tipo LiveCD vienen también con la opción de instalación una vez que se ha probado. Existen muchas distribuciones de este tipo, algunas solamente en versión LiveCD, otras como demostraciones de distribuciones que se pueden instalar de la manera tradicional. [4]

Proceso de arranque de un LiveCD/DVD.

Todos los LiveCDs aunque sean plenamente funcionales y sean útiles tienen una gran desventaja, respecto a los sistemas operativos instalados en el disco duro, una que afecta directamente al usuario, que es la rapidez.

El acceso al disco CD/DVD, siempre será más lento que el acceso que se pueda tener a un disco duro, por tanto, como usuario que evalúe el uso de la distribución con el LiveCD; se tiene que tener en cuenta, que la velocidad a la que se trabajará si se instala en el disco duro, será mucho mayor.

Un LiveCD/DVD contiene una distribución de Linux como la que se instala en un disco duro. Sin embargo, no basta con copiar la instalación desde el disco duro a un CD/DVD para obtener un sistema autoarrancable, ya que existen pequeñas diferencias entre un LiveCD/DVD y un disco duro normal.

Un CD o un DVD es un medio de solo lectura y el kernel de Linux necesita tener acceso de escritura a algunas partes del sistema para poder operar apropiadamente, como por ejemplo los directorios temporales de procesos y dispositivos. Hay variadas formas para solucionar este problema, todos ellos son usando la memoria RAM del sistema. Algunos de estos métodos permiten el acceso de escritura sólo a los directorios y archivos básicos, y por lo tanto, no permiten al usuario modificar el sistema o instalar nuevos paquetes en el Live CD/DVD.

Uno de los métodos antes mencionados es **unionfs**, el cual permite tener acceso de escritura en todo el sistema, haciendo posible la instalación de paquetes y la modificación de archivos del sistema. Esto se realiza mediante la fusión de parte de la memoria RAM con el sistema de archivos de sólo lectura del LiveCD/DVD, que lo hace parecer un sistema de archivos que tienen permisos de lectura y escritura.

El sistema debe tener un tamaño determinado para que quepa en el medio y el mismo debe estar comprimido; generalmente se usa **squashfs**. Por lo que de la misma forma el tipo de compresión usado debe ser detectado al inicio del proceso de arranque. Así el kernel usado para el arranque del LiveCD/DVD debe tener los módulos para detectar el sistema de compresión usado. Estas consideraciones exigen una preparación especial en el momento de arranque, algunas de las cuales debe realizarse incluso antes de montar el sistema de ficheros.

Linux presenta un mecanismo que permite que estos pasos se hagan en el momento de arranque incluso antes de que el sistema de archivos raíz esté montado. Se llama el sistema de ficheros inicial o **initramfs**. Este mecanismo se utiliza también en el proceso de arranque de instalaciones en discos duros, ya que añade flexibilidad al proceso.

El gestor de arranque carga el kernel y el **initramfs** en la memoria y se inicia el kernel. Este entonces desempaqueta el **initramfs** y lo monta como sistema de ficheros inicial, a continuación busca el programa **init** en este sistema, una vez encontrado, lo ejecuta y comienza el proceso de arranque. Este script de inicio se encarga de encontrar el verdadero sistema de ficheros y montarlo. También es responsable por cualquier operación especial que se quiera hacer en el momento del arranque.

Funcionamiento de los sistemas de creación de un LiveCD/DVD.

La mayor parte de estas aplicaciones parten de un sistema pre instalado en una partición de la computadora, la cual debe ocupar menos de 2GB de espacio pues la imagen comprimida que se crea posteriormente debe ser menor de 700 MB, en caso que se quiera crear un LiveDVD (4.7 GB) el tamaño de la partición puede ser mucho mayor.

Una vez hecho los cambios necesarios en los ficheros de configuración y haber personalizado todo lo que se estime, se instalan algunos paquetes necesarios para que el sistema sea autoarrancable, como los módulos necesarios del kernel para reconocer el sistema de ficheros usado para comprimir el sistema del LiveCD/DVD y los scripts del **initramfs**.

Al estar listos los requisitos anteriormente mencionado se crea una imagen comprimida de dicho sistema, aunque se puede hacer un LiveCD/DVD sin comprimir el sistema de ficheros usando cualquier otro que no lleve compresión, como por ejemplo ext2; siempre teniendo en cuenta el tamaño del mismo. Siguiendo a esto se prepara el gestor de arranque que tendrá el LiveCD/DVD, que pueden ser Syslinux, Grub u otro y por último se crea la imagen con sistema de ficheros ISO 9660, lista para quemarla en un CD o DVD.

Otros sistemas para generar LiveCD/DVDs lo que hacen es conectarse a los repositorios oficiales de las distribuciones como Debian o Ubuntu y descargan paquetes que pueden ser previamente seleccionados por el usuario. Este método no es muy útil si se quiere tener un sistema con una configuración personal. Una de las aplicaciones más usadas para este fin es **debootstrap** (aclarar que este programa es solo para Debian y sistemas basados en él).

1.2. Sistemas de creación de LiveCD/DVDs.

A nivel internacional y nacional, se han creado herramientas con el objetivo de automatizar el proceso de creación de LiveCD/DVDs. Muchos sistemas se han adaptado a las necesidades de las distribuciones para las que fueron creadas, pero otras son de uso más genérico.

A continuación se hace un recuento de algunos de estos sistemas:

1.2.1. Dfsbuild.

Debian desde cero (DFS en inglés) crea un LiveCD/DVD que está diseñado para proporcionar un núcleo y un ambiente de rescate completo. El ambiente de rescate contiene herramientas para trabajar con el sistema de ficheros, editores, un ambiente de desarrollo para el lenguaje C, entre otras. Es muy configurable y se pueden crear LiveCD/DVDs con un núcleo personalizado y un conjunto de paquetes totalmente diferente.

Dfsbuild trabaja obteniendo las aplicaciones desde el repositorio de Debian GNU/Linux más cercano y genera una imagen ISO que contendrá un sistema Debian GNU/Linux autoarrancable. También permite adicionar en la imagen todos los paquetes de Debian GNU/Linux y ficheros que necesite debootstrap. Además, la imagen generada brinda la posibilidad de instalar el sistema base de Debian GNU/Linux en el ordenador.

La aplicación tiene un archivo de configuración que permite modificar las principales directivas de la misma, entre ellas el repositorio a usar, desde donde dfsbuild descargará las aplicaciones incluidas en el LiveCD/DVD, así como el Kernel a usar y la lista de paquetes a incluir en el LiveCD/DVD. [5]

Ventajas:

- ✓ Liberado bajo licencia GPL.
- ✓ Permite elegir la sección en el repositorio de Debian GNU/Linux para hacer el LiveCD/DVD.
- ✓ Soporta la creación de LiveCD/DVDs para varias arquitecturas (amd64, i386, alpha, powerpc).
- ✓ Facilita guardar el contenido de los archivos de configuración.

Desventajas:

- ✓ Posibilita la inclusión de terceros paquetes, a pesar de ser complicada.
- ✓ Cuando ocurre un error es posible continuar desde este punto, pero no es funcional la mayoría de las veces.
- ✓ Solo funciona en el sistema operativo Debian.

1.2.2. Bootcd.

Es una herramienta que construye un LiveCD funcional de un sistema Debian GNU/Linux con el comando `bootcdwrite`. Además puede crear una imagen ISO a través de NFS en un sistema remoto. Cuando se ejecuta el sistema desde el CD no es necesario ningún disco. Todos los cambios se harán en la memoria RAM de la computadora. Para la reutilización de los cambios en el próximo arranque puede guardarse la configuración personal en un disquete con el comando `bootcdflopcp`. Si el arranque de la unidad de CD no es compatible, arrancar desde disquetes es también posible.

Si `bootcd` encuentra algún problema avisará y permitirá ignorarlo, continuar o cancelar. El problema más habitual es que la imagen creada no quepa en un CD o un DVD, para que el sistema quepa en un CD tendrá que ser realmente minimalista. [6]

Principales Características:

- ✓ Soporta como gestores de arranque: Lilo, Grub, Syslinux.
- ✓ Permite la realización de copias de seguridad con esta herramienta.
- ✓ No comprime con sistemas eficientes de compresión como `squashfs`.
- ✓ El fichero de configuración es muy simple.
- ✓ El proceso es totalmente cerrado, la interacción con el usuario es mínima.

1.2.3. LiveCD-Tools.

Es una herramienta para la generación de LiveCDs en Fedora y sus derivados, incluidos los sistemas basados en las distribuciones como RHEL, CentOS y otros. Para crear LiveCD/DVDs con la misma, se requiere de un conjunto de archivos de configuración, además una lista de paquetes y la descripción del tipo de configuración del sistema de arranque. [7]

Existen tres paquetes encargados de crear distintos tipos de LiveCD/DVDs. Los paquetes siguen un régimen de sucesión que ayuda en la creación de las versiones derivadas de Fedora, a continuación se enumeran los mismos:

1. Fedora-livecd es un LiveCD mínimo sin interfaz de usuario.
2. Fedora-livecd-gnome se basa en fedora-base e incluye un entorno de escritorio GNOME.
3. Fedora-livecd-tools se basa en la parte superior del paquete fedora-gnome, con un entorno GNOME con otras aplicaciones y funcionalidades.

Principales Características:

- ✓ Posibilidad de instalar software mientras esté ejecutándose el LiveCD/DVD. Usando UnionFS.
- ✓ Facilita la creación de imágenes para CD-ROM y DVDs.
- ✓ Genera LiveCD/DVDs para unidades USB y discos duros.
- ✓ Utiliza SELinux para el cumplimiento de las funciones de Seguridad.
- ✓ Permite la persistencia de los datos de la sesión de trabajo del usuario y todas las configuraciones en el momento de la creación del LiveCD/DVD.

1.2.4. Live-magic.

Es una herramienta que crea LiveCD/DVDs de Debian GNU/Linux así como imágenes netboot, entre otras. Todo esto lo hace con la ayuda de live-helper. Esta herramienta tiene una interfaz desarrollada con python y GTK con características simples para la construcción de los LiveCD/DVDs. [8]

Principales Características:

- ✓ Genera Live-helper ISOs para CD, DVD de arranque, imágenes para dispositivos USB, USB Thumb e imágenes netboot.
- ✓ La interfaz permite la creación de un LiveCD/DVD de forma fácil al usuario.
- ✓ Es difícil realizar la inclusión de paquetes personalizados.

Live-magic no es más que una interfaz gráfica que permite generar un LiveCD/DVD usando live-helper como aplicación base. Muchas veces en las pruebas que se realizaron, la aplicación fallaba o se interrumpía el proceso de creación, además de no tener gran cantidad de opciones de configuración.

1.2.5. Live-Helper.

Es una herramienta de creación de LiveCD/DVDs en Debian GNU/Linux. Se trata de un CD estándar similar a los instaladores que permite arrancar un ordenador pero a diferencia de los instaladores su finalidad es dar una utilización totalmente funcional al ordenador sin necesidad de tocar en absoluto la configuración original del sistema instalado, si lo hubiera. [9]

Principales Características:

- ✓ Es una aplicación que trabaja a nivel de consola.
- ✓ Crea por defecto una imagen ISO a partir de la rama estable. Aunque esta opción es configurable.
- ✓ Permite la inclusión de un repositorio local.
- ✓ Funciona en equipos de bajo rendimiento permitiendo la generación de un LiveCD/DVD personalizado con entorno de escritorio en modo texto, funcionando con 48 MB de RAM y Swap.
- ✓ Permite agregar o eliminar paquetes mediante la ejecución del LiveCD/DVD.
- ✓ Permite la modificación del sistema de arranque.
- ✓ Permite la edición de la estructura de directorios del ISO.
- ✓ Es posible agregar el instalador oficial de Debian GNU/Linux al LiveCD/DVD, ofreciéndole al usuario la opción de poder seleccionar si arranca el LiveCD/DVD o el instalador del mismo.

1.2.6. Remastersys.

Es un programa para la distribución Ubuntu Linux usado para modificar los LiveCD/DVDs oficiales de la misma o de sus derivados, y permite además hacer una copia de seguridad de todo un sistema, incluyendo los datos de usuario en un LiveCD/DVD instalable. [10]

Permite crear dos tipos de imágenes:

1. Una imagen completa del disco duro incluyendo los datos de la carpeta /home
2. Una imagen completa del disco duro pero sin incluir los datos de la carpeta /home.

Inicialmente fue creado para ser capaz de hacer copias de seguridad o crear una copia de una distribución Ubuntu o derivada de ésta. La inspiración para hacer esta aplicación procedía de mklivecd, script que usa Mandriva, además del script "remasterme" que se encuentra en PCLinuxOS. La imagen ISO creada también puede instalarse en una memoria USB para la creación de un LiveUSB. Dispone de una versión de línea de comandos y una versión en interfaz gráfica. En la actualidad se trabaja con Ubuntu, Linux Mint, Klikit-Linux y más distribuciones basadas en Ubuntu.

1.2.7. Linux Live Scripts.

Linux Live Scripts no es más que un conjunto de scripts de consola desarrollados en Bash que permiten generar un LiveCD de una distribución de Linux ya previamente instalada. También puede crear un sistema autoarrancable desde dispositivos USB Flash Drive, USB Pen Drive, cámara conectada al puerto USB o un iPod. [11]

Principales Características:

- ✓ Utiliza los cambios en los parámetros de arranque para especificar persistencia en los ficheros y directorios.
- ✓ Permite adicionar nuevas aplicaciones y módulos al sistema mientras se está ejecutando.
- ✓ Permite configurar el sector de arranque para ejecutar Linux desde el LiveCD/DVD. Si se combinan con la persistencia de los cambios realizados, el LiveCD/DVD se comportará como instalado, todos los cambios son almacenados transparente para el medio de arranque (incluso en un USB).

1.2.8. UCK.

UCK (Ubuntu Customization Kit) es un kit de herramientas para Ubuntu, Kubuntu, Xubuntu y Edubuntu que sirve para crear LiveCD "mejorados" a partir de las imágenes ISO de los CD oficiales de instalación de Ubuntu.

Permite crear un LiveCD mediante un script a nivel de consola o utilizando una interfaz gráfica. Es posible personalizar el sistema instalando y eliminando paquetes. De la misma forma es posible añadir y eliminar módulos al Kernel de Linux. [12]

1.2.9. Reconstructor.

Reconstructor utiliza los LiveCD/DVD oficiales de Ubuntu Linux como base y permite la personalización de las pantallas de inicio (usplash), y el software que se encuentra en los mismos. También puede usar el comando chroot para hacer otros cambios antes de crear el LiveCD/DVD. Mantiene la sólida base de Ubuntu y sólo permite la personalización. Reconstructor está escrito en Python y está licenciado bajo la licencia GPL. [13]

Principales Características:

- ✓ Permite añadir nuevos programas al LiveCD/DVD.
- ✓ Los LiveCD/DVDs generados facilitan añadir repositorios personalizados.
- ✓ Capacidad para utilizar programas gráficos dentro del chroot (experimental).

1.2.10. Nova LiveCD Kit.

La herramienta LiveCD Kit de Nova GNU/Linux está desarrollada en el lenguaje de programación Bash brindando la facilidad de poder ser mantenida por cualquier desarrollador de forma muy sencilla. Esta es altamente flexible, y sirve para generar un LiveCD/DVD al gusto del usuario, la parte más difícil es a la hora de configurar las variables de entorno ya que se debe poseer un mínimo de conocimiento del sistema operativo que se vaya a usar.

Permite mantener un control de versiones de cada LiveCD/DVD utilizando un nombre de perfil, una fecha y una versión así como mantener varios perfiles de forma que el usuario puede mantener varios proyectos a la vez. Además posibilita hacer backups completos y restaurarlos en caso de que se vaya a hacer algo de lo que no se esté seguro.

Brinda una estimación del tamaño máximo que puede ocupar su sistema para no exceder el tamaño estándar de un ISO de 700 MB. La aplicación posee un menú intuitivo que permite realizar la operación solicitada. Esta es multiplataforma, es decir que puede usarse para generar un LiveCD/DVD desde cualquier plataforma GNU/Linux. Otra característica importante de esta aplicación es que permite hacer chroot al sistema original o al sistema objeto para eliminar o adicionar paquetes así como para modificar archivos de configuración. [14]

1.2.11. Calculate-Linux Scratch (CLS).

Calculate-Linux Scratch está basada en Gentoo y está destinada principalmente a los administradores y todos los usuarios que quieren tener su propia distribución de Linux, que estará optimizado para ciertas tareas. Esta distribución tiene como objetivo crear un LiveCD/DVD para cualquier tarea. [15]

1.2.12. Comparación entre las herramientas utilizadas en la UCID (NovaLiveCD Kit y Calculate Linux).

Luego de un estudio realizado de todas las herramientas existentes para la creación de LiveCD/DVDs, se decide hacer una comparación entre las dos herramientas usadas actualmente en la UCID, con el objetivo de basar su funcionamiento en la más óptima para la realización de su interfaz visual.

Herramienta NovaLiveCD Kit.

Ventajas:

- ✓ Puede ser mantenida por cualquier desarrollador de forma muy sencilla.
- ✓ Es altamente flexible.
- ✓ Posibilita la realización y restauración de backups completos.
- ✓ Brinda una estimación del tamaño máximo que puede ocupar su sistema para no exceder el tamaño estándar de un ISO de 700 MB.
- ✓ Es multiplataforma.

Desventajas:

- ✓ Requiere mínimos conocimientos del sistema operativo para configurar las variables de entorno.
- ✓ No está diseñada para usuarios comunes.
- ✓ Impide la creación de initramfs automático y sistemas desde cero.
- ✓ No es oficial el paquete.
- ✓ Deficiencias en cuanto a términos usados; insuficiente documentación ya que no posee ayuda.
- ✓ El código no está bien distribuido.
- ✓ No posee una interfaz visual.
- ✓ No posee la opción de quemar.

Herramienta Scratch Calculate-Linux (CLS).

Ventajas:

- ✓ El LiveCD/DVD contiene una copia exacta de la imagen de instalación.
- ✓ CLS incluye utilidades para configurar, compilar e instalar el sistema.
- ✓ El proceso de creación del ISO es interactivo.
- ✓ Es cien por ciento compatible con Gentoo.
- ✓ Posee el máximo de optimización para sistemas más antiguos.
- ✓ Permite una fácil actualización a través del repositorio Portage.

Desventajas:

- ✓ El Scratch no se encuentra separado de la distribución.
- ✓ No posee una interfaz visual.
- ✓ Se requiere de un conocimiento de la distribución para poder utilizar la herramienta.
- ✓ Es necesario que el CD/DVD esté en la unidad de lectura.

Se llega a la conclusión de que el funcionamiento de la herramienta que se desarrollará en el presente trabajo estará basado en la herramienta Nova LiveCD Kit, ya que el Scratch de Calculate-Linux provee muy buenas condiciones pero tiene como limitante que hay que hacerlo desde el LiveCD/DVD, por lo que la velocidad de creación del LiveCD/DVD, estará en dependencia de la RAM de la PC donde se esté construyendo el LiveCD/DVD, mientras que la herramienta Nova LiveCD Kit, la filosofía que sigue, permite hacérselo a cualquier distribución, esencialmente basada en Gentoo aunque ya se ha probado en otras distribuciones como Ubuntu y Suse.

1.3. Herramientas, Lenguajes y Metodologías a usar.

En el desarrollo de todo sistema informático es de vital importancia la selección de las herramientas, lenguajes y tecnologías a utilizar, paso que garantizará, un óptimo desempeño de la aplicación. Para el desarrollo del sistema al cual se refiere este trabajo, la selección se realizó teniendo en cuenta la infraestructura tecnológica de la UCI y valorando que la herramienta a desarrollar, estaría orientado a funcionar sobre un sistema pre-instalado Nova GNU/Linux.

1.3.1. Framework Qt.

Inicialmente Qt apareció como biblioteca desarrollada por Trolltech (en aquel momento "Quasar Technologies") en 1992 siguiendo un desarrollo basado en el código abierto, pero no libre.

Se usó activamente en el desarrollo del escritorio KDE (entre 1996 y 1998), con un notable éxito y rápida expansión. Esto fomentó el uso de Qt en programas cerrados para el escritorio, situación vista por el proyecto GNU como amenaza para el software libre. Para contrarrestar la situación se plantearon dos ambiciosas iniciativas: por un lado el equipo de GNU en 1997 inició el desarrollo del entorno de escritorio GNOME con GTK+ para GNU/Linux. Por otro lado intentaron hacer una biblioteca compatible con Qt pero totalmente libre, llamada Harmony.

En noviembre de 1998, se anunció el cambio de licencia de Qt que, a pesar de todo, no contaba con el beneplácito de la Free Software Foundation. El 4 de septiembre de 2000, Trolltech comenzó a ofrecer las bibliotecas Qt bajo la licencia GNU versión 2.1 y el 18 de enero de 2008, Trolltech anunció que también ofrecerá Qt bajo la licencia GNU v3.

Qt cuenta actualmente con un sistema de triple licencia: GNU v2 y GNU v3 para el desarrollo de software de código abierto (open source) y software libre, y otra de pago para el desarrollo de aplicaciones con cualquier licencia. Hoy la última versión de las bibliotecas es la 4.6, y además de las múltiples mejoras, son también liberadas bajo licencia GNU para Windows y Mac. [16]

Potencialidades de Qt 4.

Las librerías de Qt están implementadas usando el lenguaje de programación C++ de forma nativa y además existen bindings para C, Python (PyQt), Java (Qt Jambi), Perl (PerlQt), Gambas (gb.qt), Ruby (QtRuby), PHP (PHP-Qt) y Mono (Qyoto) entre otros.

El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML y una multitud de otros para el manejo de ficheros, además de estructuras de datos tradicionales. Entre las principales funcionalidades de Qt se encuentran la posibilidad de hacer aplicaciones en modo línea de comando sin interfaz gráfica y además se añaden las siguientes nuevas tecnologías:

- ✓ **Tulip:** Es un nuevo conjunto de plantillas contenedoras de clases más ligeras, seguras y fáciles de usar que los contenedores anteriores.

- ✓ **Interview:** Es una arquitectura de representación de elementos tanto en forma de árbol, como de tablas y lista.
- ✓ **Arthur:** Es el nuevo motor de representación gráfica se integrará con dibujado por pixel y vectores, como lo son QuickDraw bajo Mac OS X, Xlib en X11, GDI en Windows, así como PostScript y OpenGL.
- ✓ **Scribe:** Es el nuevo motor de representación de texto que soporta Unicode, texto enriquecido y la posibilidad de texto rodeando y formas irregulares.
- ✓ **Mainwindow:** Es una nueva arquitectura basada en acciones de menús, ventanas, barras de herramientas y widgets anclados en cualquier parte de la ventana (arriba, debajo, izquierda y derecha).

Qt Creator.

Qt Creator es un IDE (Integrated Development Environment) liberado por Nokia para el desarrollo de aplicaciones basadas en el conjunto de librerías Qt. La cual es multiplataforma, actualmente existen versiones para Linux, Microsoft Windows XP, Microsoft Windows Vista y Mac OS.

Qt Creator trae herramientas de navegación que permiten administrar el código de origen. Posee revisor de código estático y suministra sugerencias de estilo a medida que se tipea, code folding y edición avanzada.

Su objetivo no es remplazar Visual Studio o Eclipse, sino ser un IDE rápido, simple y usable para el desarrollo de aplicaciones para Linux, Windows y Mac OS así como embebidas en dispositivos portátiles como teléfonos y PDAs.

Proporciona un conjunto de características para aumentar la productividad de los desarrolladores experimentados de Qt, y para ayudar a los nuevos usuarios a poner en marcha sus proyectos más rápidamente. [17]

Principales Funcionalidades:

- ✓ Presenta un asistente para la generación de proyectos.
- ✓ Tiene una pantalla de bienvenida en la que se muestran los últimos proyectos o sesiones usados.
- ✓ Está integrado con el sistema de ayuda de Qt.
- ✓ Incluye módulos de desarrollo como Phonon, XML, OpenGL, entre otros.

- ✓ Presenta un avanzado editor de código C++.
- ✓ El completamiento de código es muy preciso.
- ✓ Tiene una potente herramienta para navegar entre los archivos y clases.
- ✓ Incluye la herramienta Qt Designer para la creación de interfaces gráficas.
- ✓ Presenta una interfaz gráfica para GDB (Depurador Simbólico de GNU).
- ✓ Tiene dentro la estructura de clases de Qt para acelerar el proceso de depuración.
- ✓ Construye los proyectos de Qt integrados a qmake. Herramienta de construcción multiplataforma.
- ✓ Permite la gestión de proyectos.

Qt Designer.

Es una herramienta muy potente que permite diseñar de forma muy sencilla y rápida ventanas de diálogo con las librerías Qt. Esta es una aplicación mediante la cual se puede realizar el diseño de aplicaciones gráficas de forma intuitiva y fácil. [18]

Principales Características:

- ✓ Dispone de una paleta de widgets muy completa, que incluyen los widgets más comunes de las librerías QT. Si además se han instalado las librerías para el desarrollo de aplicaciones KDE, se tendrán widgets adicionales.
- ✓ El funcionamiento es de estilo Selecciona y Dibuja, es decir, basta con seleccionar un tipo de widget en la paleta y luego se pone sobre el formulario, dibujándose con la geometría que se desee.
- ✓ Las propiedades de un widget cualquiera se pueden cambiar fácilmente en tiempo de diseño con el panel de propiedades.

Signal y Slot.

Los slots y signals (señales) son un mecanismo de comunicación entre objetos, esta es la principal característica de Qt y es el rasgo que hace distintas las librerías Qt del resto de las herramientas para la elaboración de GUI (Graphic User Interface). El mismo consiste en un mecanismo de comunicación seguro, flexible y totalmente orientado a objetos y por supuesto implementado en C++.

En la programación con GUI se busca que los cambios producidos en un objeto sean comunicados a otros objetos, por ejemplo cuando se hace clic en un botón para que se cierre una ventana, lo que se hace es posibilitar la comunicación entre los dos objetos.

Otras herramientas de diseño de GUI llevan a cabo la comunicación entre objetos usando los llamados callbacks. Un callback es un puntero a una función, con este mecanismo si se quiere procesar una determinada función cada vez que ocurre un evento en un objeto, lo que se hace es pasar un puntero a otra función (el callback) a la función deseada y será esta la que se encargue de llamar al callback en el momento apropiado.

Este tipo de comunicación tiene el inconveniente de no ser totalmente seguro puesto que no se sabe si se llamará al callback con los argumentos apropiados y además la función que llama al callback debe saber exactamente a que callback llamar, además es un sistema inflexible y no está orientado a objetos. [19]

1.3.2. Mksquashfs.

Como parte de las aplicaciones que se usarán por la herramienta que se desarrollará se encuentra mksquashfs, la cual es una aplicación de línea de comando usada para crear sistemas de ficheros comprimidos con squashfs, un poderoso sistema de ficheros de solo lectura muy útil para almacenar grandes cantidades de datos. Es un sistema de ficheros estable que ha sido probado en varias arquitecturas como PowerPC, i586, Sparc y ARM.

Esta aplicación también puede ser usada para modificar ficheros comprimidos con squashfs, de la misma forma la descompresión es posible, pero con otra herramienta llamada unsquashfs.

Mksquashfs tiene un gran número de opciones para modificar su sistema de ficheros comprimidos de forma rápida, pero la mayoría son para cambiar opciones avanzadas que raramente hace falta cambiar. Con solo un comando en la consola se obtiene el sistema comprimido. [20]

1.3.3. Aufs.

Aufs (Another UnionFS) es una versión alternativa de unionfs, un servicio para sistemas de archivos Linux que permite montar un sistema de ficheros formado por la unión de otros sistemas de archivos. Esta siendo desarrollado por Junjiro Okajima desde principios del 2006.

Esta tecnología es una re-implementación completa de unionfs, con el ánimo de mejorar la estabilidad, la confiabilidad y el rendimiento. Este también incluye algunos nuevos conceptos como el balance entre ramas escribibles entre otras muchas mejoras. Algunas de estas nuevas ideas han sido implementadas por unionfs en su versión 2.

Se utiliza principalmente en LiveCD/DVDs o sistemas sin disco, donde el directorio raíz de sólo lectura se combina con un sistema de archivos tmpfs (que reside en memoria y es modificable). De este modo, todos los archivos de la raíz de sólo lectura pueden ser modificados, y la modificación se mantiene en memoria.

Aufs ha sido escogido como reemplazo para unionfs en la popular distribución autónoma Knoppix desde finales del 2006. Así como en la distribución Slax desde su versión 6. [21]

1.3.4. Squashfs.

Es una tecnología para el correcto funcionamiento de los LiveCD/DVDs generados, el cual es un sistema de archivos comprimido de sólo lectura para Linux. Squashfs comprime archivos, inodos y directorios, y soporta tamaños de bloque de hasta 1024 KB para mayor compresión. Squashfs es software libre (licenciado como GPL).

Está pensado para ser usado como sistema de archivos genérico de sólo lectura en dispositivos de bloques/sistemas de memoria limitados (por ejemplo, sistemas integrados), donde se requiere poca sobrecarga. La versión estándar de Squashfs utiliza compresión mediante gzip, aunque existe también otro proyecto que dota de compresión LZMA a Squashfs.

Este sistema de ficheros se utiliza en las versiones en LiveCD/DVD de Debian, Finnix, Gentoo Linux, Ubuntu y Mandriva, y en sistemas integrados como firmwares de dispositivos. A menudo se combina con un sistema de archivos de unión como UnionFS o aufs, para proveer un entorno de lectura-escritura para LiveCD/DVD. De este modo se combinan las ventajas de la alta velocidad de compresión de SquashFS con la posibilidad de alterar la distribución mientras se ejecuta esta desde un LiveCD/DVD. Distribuciones como Slax, Debian Live, Mandriva One y Nova usan esta combinación. [22]

1.3.5. Rsync.

Rsync es una aplicación para sistemas Unix que ofrece transmisión eficiente de datos incrementales comprimidos y cifrados. Mediante una técnica de delta encoding, permite sincronizar archivos y directorios entre dos máquinas de una red o entre dos ubicaciones en una misma máquina, minimizando el volumen de datos transferidos.

La aplicación rsync provee otras funciones que asisten en la transferencia. Estas incluyen compresión y descompresión de los datos bloque por bloque, utilizando zlib, y soporte para protocolos de cifrado, tal como SSH. Adicionalmente puede utilizarse una aplicación de tunneling para asegurar los datos.

Los principales usos de rsync incluyen espejado (mirroring) o Respaldo de múltiples clientes Unix dentro de un servidor Unix central. Habitualmente se lo ejecuta mediante herramientas de scheduling como cron, para automatizar procesos de sincronización.

Además de archivos, el algoritmo permite copiar directorios, aún recursivamente, así como vínculos, dispositivos, grupos y permisos. No requiere por defecto privilegios de root para su uso. [23]

1.3.6. Lenguaje para el modelado (UML).

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

Este lenguaje de modelado capta la información sobre la estructura estática y el comportamiento dinámico de un sistema, el cual se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

Este lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. Es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes.

Existían diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques, lo que generó la creación del UML como estándar para el modelado de sistemas de software principalmente, pero con posibilidades de ser aplicado a todo tipo de proyectos. [24]

1.3.7. Intérprete de Comandos (Bash).

Es un intérprete de comandos para Unix, escrito para el proyecto GNU. Su nombre es el acrónimo de bourne-again Shell (otro Shell Bourne) — haciendo un juego de palabras (born-again significa renacimiento en Ingles) sobre el Bourne Shell (sh), que fue uno de los primeros intérpretes importantes de Unix.

Hacia 1978 el intérprete Bourne era el intérprete distribuido con el Unix Versión 7. Stephen Bourne, por entonces investigador de los Laboratorios Bell, escribió el intérprete Bourne original. Brian Fox escribió el intérprete Bash en 1987. En 1990, Chet Ramey se convirtió en su principal desarrollador. Bash es el intérprete predeterminado en la mayoría de sistemas GNU/Linux, además de Mac OS X Tiger, y puede ejecutarse en la mayoría de los sistemas operativos tipo Unix. También se ha llevado a Microsoft Windows por el proyecto Cygwin.

La sintaxis de órdenes de Bash es un súper conjunto de la sintaxis del intérprete Bourne. La especificación definitiva de la sintaxis de órdenes de Bash, puede encontrarse en el Bash Reference Manual distribuido por el proyecto GNU.

La mayoría de los Shell scripts (guiones de órdenes) Bourne pueden ejecutarse por Bash sin ningún cambio, con la excepción de aquellos scripts de Shell Bourne que hacen referencia a variables especiales de Bourne o que utilizan una orden interna de Bourne. [25]

1.3.8. Lenguaje de Programación (C/ C++).

Hoy en día los ordenadores son capaces de realizar muchas tareas diferentes, desde simples operaciones matemáticas, hasta sofisticadas simulaciones animadas. Pero el equipo no crea estas tareas por sí mismo, se realizan estos a raíz de una serie de instrucciones predefinidas que conforman lo que llamamos un programa.

En 1970, Ken Thompson, inmerso en el desarrollo de UNIX en los laboratorios Bell, creó el lenguaje B. Se trataba de una versión de BCPL (Basic Combined Programming Language), para una máquina y sistema específico (DEC PDP-7 y UNIX), el cual se adaptó a su funcionamiento particular y necesidades.

En 1971, Dennis Ritchie, con su equipo de desarrollo de los laboratorios Bell, inició el desarrollo de un compilador llamado B que, entre otras cosas, fuera capaz de generar código ejecutable directamente. El "Nuevo B" luego llamado C introducía además algunos conceptos nuevos al lenguaje como los tipos de datos.

C es un lenguaje muy eficiente puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas. A pesar de su bajo nivel es el lenguaje más portado en existencia, habiendo compiladores para casi todos los sistemas conocidos. Proporciona facilidades para realizar programas modulares así como utilizar código o bibliotecas existentes. [26]

Entre las características de C podemos mencionar:

- ✓ Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de ficheros, proporcionadas por bibliotecas.
- ✓ Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado no llevado al extremo (permitiendo ciertas licencias rupturistas).
- ✓ Un sistema de tipos que impide operaciones sin sentido.
- ✓ Usa un lenguaje de preprocesado, el preprocesador de C, para tareas como definir macros e incluir múltiples ficheros de código fuente.
- ✓ Acceso a memoria de bajo nivel mediante el uso de punteros.
- ✓ Un conjunto reducido de palabras clave.
- ✓ Los parámetros se pasan por valor. El paso por referencia se puede simular pasando explícitamente el valor de los punteros.
- ✓ Punteros a funciones y variables estática, que permiten una forma rudimentaria de encapsulado y polimorfismo.

- ✓ Tipos de datos agregados (struct) que permiten que datos relacionados se combinen y se manipulen como un todo.

C++ es un lenguaje imperativo orientado a objetos derivado de C. En realidad un conjunto de C, que nació para añadirle cualidades y características de las que carecía. El resultado es que como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero se le han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción.

C++ tiene ciertas características respecto a otros lenguajes de programación. Las más notables son:

Programación Orientada a Objetos: La posibilidad de la programación orientada a objetos le permite al programador diseñar las aplicaciones desde un punto de vista comunicativo entre objetos más que una secuencia estática de código. Además de posibilitar la reusabilidad de código de un modo más lógico y productivo.

Portabilidad: Es posible compilar el mismo código C++ en cualquier tipo de computadora o sistema operativo, sin que ocurra algún cambio. C++ es el lenguaje de programación más usado y portado de todo el mundo.

Menos Código: El código escrito en C++ es mucho más corto que el mismo escrito en otro lenguaje, pues el uso de caracteres especiales es mucho más usado que las palabras claves. Haciendo que el programador ahorre espacio y tiempo.

Programación Modular: El cuerpo de una aplicación en C++ puede estar separado en múltiples ficheros de código, que son compilados de forma separada y luego enlazarlos. Esto ahorra tiempo pues no es necesario compilar todos los ficheros cuando se hace cambios solo en un archivo. Además esta característica permite que se pueda enlazar código C++ con otros como por ejemplo código ensamblador o C.

Compatibilidad con C: C++ es compatible con el lenguaje C, cualquier código escrito en C puede ser fácilmente incluido en un programa C++ sin hacer ningún cambio.

Velocidad: El resultado del proceso de compilación es muy eficiente, debido a su dualidad para el trabajo a bajo y alto nivel, y a la reducción del tamaño del código en sí.

1.3.9. Metodología de desarrollo (SXP).

Es una metodología compuesta por la metodologías XP y SCRUM que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo y ayuda al líder del proyecto a tener un mejor control del mismo.

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que se sepa el trabajo que se ha realizado.

XP más bien es una metodología encaminada para el desarrollo; 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.

SXP consta de 4 fases principales:

- ✓ **Planificación-Definición:** Se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto;
- ✓ **Desarrollo:** Se realiza la implementación del sistema hasta que este listo para ser entregado;
- ✓ **Entrega y Puesta en marcha:** Se entrega el software y se comienza a usar.
- ✓ **Mantenimiento:** Se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añada una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo. [27]

A continuación se muestran los artefactos que se generan con esta metodología de software.

Artefactos Generados en SXP:

- Concepción del Sistema.
- Estado del Arte.
- Modelo de Historias de Usuarios del Negocio.
- LRP (Lista de Reservas de Producto).
- Historia de Usuario.
- Lista de Riesgo.
- Modelo de Diseño.
- Tarea de Ingeniería.
- Plan de Releases.
- Manual de Usuario.
- Caso de Prueba de Aceptación.
- Estándar de Programación.
- Manual de Identidad.
- Manual de Desarrollo.
- Gestión de Cambios.

En este capítulo se realizó un análisis de las principales aplicaciones para la creación de LiveCD/DVDs, así como las disímiles herramientas y sistemas que se emplearán en el desarrollo de la misma. El diseño de la interfaz gráfica de la aplicación se realizará con la herramienta Qt Designer, la cual permite diseñar de una forma rápida y sencilla los componentes visuales de las librerías Qt. El sistema será comprimido con la herramienta mksquashfs, la cual se usa para crear sistemas de ficheros comprimidos con squashfs el cual permite grandes ratios de compresión.

Se utilizará como metodología de desarrollo de software SXP, que es una metodología compuesta por la metodología XP y SCRUM que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles.

Capítulo 2: Análisis y Diseño del Software NovaFAR Live.

En este capítulo se realizará una breve comparación entre las herramientas que en la actualidad se emplean en la UCID, sobre el sistema operativo NovaFAR para la generación de LiveCD/DVDs en el desarrollo ágil de la aplicación Nova LiveCD Kit, utilizando la metodología ágil SXP para el desarrollo de software que se realizó en el año 2008 por la Ingeniera Gladys Marsi Peñalver Romero (GLADYS M., P. R. 2008), además se explicará también toda la dinámica del proyecto en forma de historias de usuarios, prototipos de interfaz de usuario y algunos modelos auxiliares.

2.1. Roles del proyecto.

Rol	Nombre
Gerente	Carlos A. García Rodríguez. Elisa Gamez López.
Cliente	NovaFAR.
Programador	Carlos A. García Rodríguez. Elisa Gámez López.
Analista	Elisa Gámez López. Carlos A. García Rodríguez.
Diseñador	Elisa Gámez López.
Probador	Elisa Gámez López.
Arquitecto	Carlos A. García Rodríguez.

Tabla 1. Roles del proyecto.

2.2. Modelo de dominio.

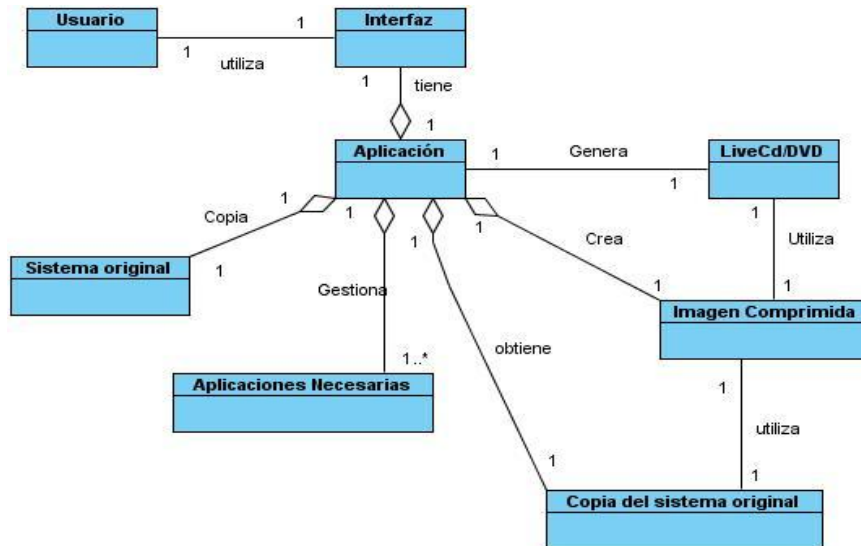


Figura 1. Modelo de Dominio.

En el modelo de dominio de la aplicación a desarrollar que se ha expuesto anteriormente, se explica el proceso general que involucra el funcionamiento de la misma. En el primer paso el usuario utiliza la interfaz gráfica que provee la herramienta y esta a su vez se encarga de gestionar todas las aplicaciones necesarias para que la misma funcione correctamente. De la misma forma la aplicación copia el sistema original para evitar que se dañe durante el proceso de creación del LiveCD. Usando esta copia una vez que se ha modificado y optimizado, se procede a comprimir el mismo. Finalmente la aplicación usando esta imagen comprimida genera el LiveCD.

2.3. Historias de usuario y prototipo de interfaz.

A continuación se dan a conocer las distintas historias de usuarios que están presentes en el sistema; así como su relación con los prototipos de interfaz usuario y las tareas asociadas a cada historia de usuario; estas se desarrollan por la prioridad que tienen y por los usuarios encargados de las mismas. Ésta es solo una planificación inicial, el proceso es cambiante para ir adecuándolo a las necesidades y nuevas propuestas. Todas las decisiones se toman de conjunto con el cliente ya que es parte del equipo de desarrollo.

Historia de Usuario.	
Número: HU-NFL-1	Nombre: Programar Scripts Auxiliares.
Modificación de Historia de Usuario: Ninguna	
Usuario: Carlos A. García Rguez. Elisa Gamez López.	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Se programan los scripts para optimizar las configuraciones y los permisos de la aplicación. Estos scripts deben permitir montar las particiones /dev y /proc, ejecutar tareas de mantenimiento como la creación y eliminación de ficheros entre otras.	
Observaciones: Los scripts se programarán usando Bash. Los mismos permitirán optimizar el sistema del LiveCD.	

Tabla 2. HU Programar Scripts Auxiliares.

Tarea de Ingeniería.	
Número Tarea: 1	Número Historia de Usuario: HU-NFL-1
Nombre Tarea: Crear script para optimizar el sistema.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 3/3/2010	Fecha Fin: 14/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	

Descripción: Este script permite modificar el sistema para que sea más óptimo. Entre las tareas que debe realizar se encuentra montar las particiones necesarias y ejecutar tareas de mantenimiento como la creación y eliminación de ficheros entre otras.

Tabla 3. Tarea de Ingeniería #1 de la HU Programar Scripts Auxiliares.

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: HU-NFL-1
Nombre Tarea: Crear scripts que copia los datos personales del usuarios seleccionado.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 3/3/2010	Fecha Fin: 14/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Este script permite copiar los datos personales del usuario seleccionado para el LiveCD/DVD. Los datos se copian a /etc/skel.	

Tabla 4. Tarea de Ingeniería #2 de la HU Programar Scripts Auxiliares.

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: HU-NFL-1
Nombre Tarea: Crear script para desmontar las particiones usadas por el script chroot.sh.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1

Fecha Inicio: 3/3/2010	Fecha Fin: 14/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Este script permite desmontar las particiones usadas por el script chroot.sh.	

Tabla 5. Tarea de Ingeniería #3 de la HU Programar Scripts Auxiliares.

Historia de Usuario	
Número: HU-NFL-2	Nombre: Copia del sistema origen.
Modificación de Historia de Usuario: Ninguna.	
Usuario: Carlos A. García Rodríguez. Elisa Gamez López.	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta/ Media/ Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Alta (Alta/ Media/ Baja)	Puntos Reales: 1
Descripción: Se hace una copia del sistema del cual se creará el LiveCD/DVD. Usando rsync, donde se excluirán los ficheros y directorios previamente seleccionados.	
Prototipo de interface:	

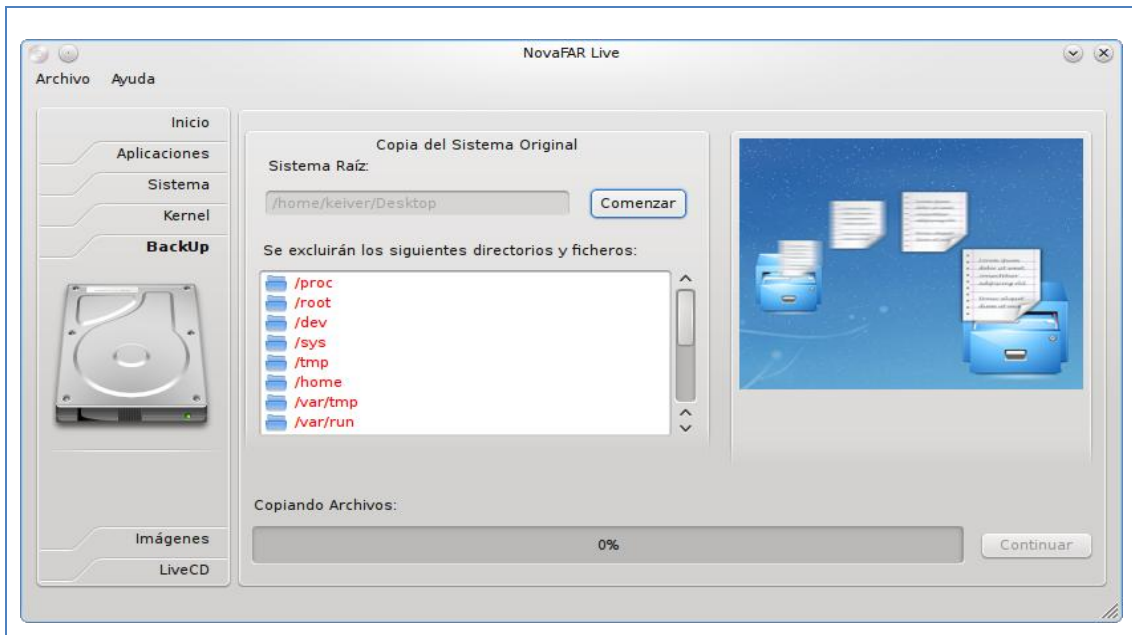


Tabla 6. HU Copia del sistema origen.

Tarea de Ingeniería.	
Número Tarea: 1	Número Historia de Usuario: HU-NFL-2
Nombre Tarea: Definir expresiones regulares para obtener el por ciento de copia de un fichero.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18/3/2010	Fecha Fin: 22/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Se determina la expresión regular para el por ciento de copia del sistema. Necesario para mostrarlo en la interfaz gráfica. El nombre de la expresión es <code>int_percent_exp</code> y la podemos encontrar en la clase <code>MainWindow.cpp</code> , en el método <code>Rsync</code> .	

Tabla 7. Tarea de Ingeniería #1 de la HU Copia del sistema origen.

Tarea de Ingeniería.	
Número Tarea: 2	Número Historia de Usuario: HU-NFL-2
Nombre Tarea: Definir expresiones regulares para obtener el nombre del fichero que se está copiando.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 18/3/2010	Fecha Fin: 22/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Se determina la expresión regular para obtener el nombre del fichero que se está copiando. Necesario para mostrarlo en la interfaz gráfica. El nombre de la expresión es <i>name_expy</i> la podemos encontrar en la clase MainWindow.cpp, en el método Rsync.	

Tabla 8. Tarea de Ingeniería #2 de la HU Copia del sistema origen.

Tarea de Ingeniería.	
Número Tarea: 3	Número Historia de Usuario: HU-NFL-2
Nombre Tarea: Realizar la copia del sistema original.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 22/3/2010	Fecha Fin: 22/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Se copia el sistema original previamente seleccionado usando la	

herramienta rsync.

Tabla 9. Tarea de Ingeniería #3 de la HU Copia del sistema origen.

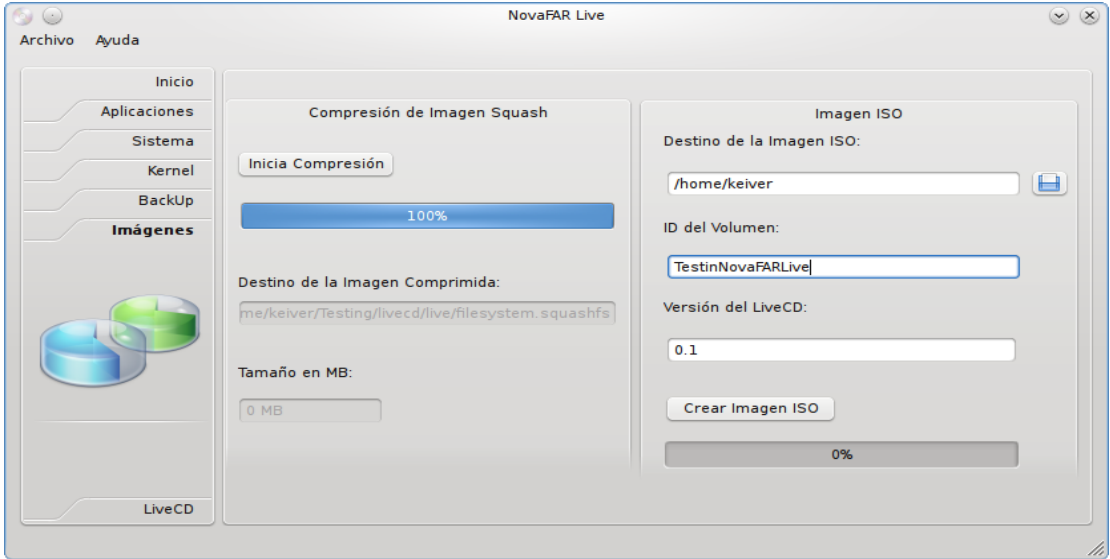
Historia de Usuario.	
Número: HU-NFL-3	Nombre: Compactar Imagen.
Modificación de Historia de Usuario: Ninguna.	
Usuario: Carlos A. García Rodríguez. Elisa Gamez López.	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Comprime el sistema creando una imagen comprimida con squashfs. Usando la aplicación mksquashfs.	
Prototipo de interface:	
	

Tabla 10. HU Compactar Imagen.

Tarea de Ingeniería.	
Número Tarea: 1	Número Historia de Usuario: HU-NFL-3
Nombre Tarea: Crear expresión regular para obtener el por ciento de creada de la imagen.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 20/3/2010	Fecha Fin: 23/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Se determina la expresión regular para obtener el por ciento de la creación de la imagen. Necesaria para mostrarlo en la interfaz gráfica. El nombre de la expresión es <i>int_percent_exp</i> .	

Tabla 11. Tarea de Ingeniería #1 de la HU Compactar Imagen.

Tarea de Ingeniería.	
Número Tarea: 2	Número Historia de Usuario: HU-NFL-3
Nombre Tarea: Calcula el tamaño de la imagen comprimida.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 20/3/2010	Fecha Fin: 23/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Se calcula el tamaño de la imagen comprimida. Necesaria para mostrarlo en la interfaz gráfica.	

Tabla 12. Tarea de Ingeniería #2 de la HU Compactar Imagen.

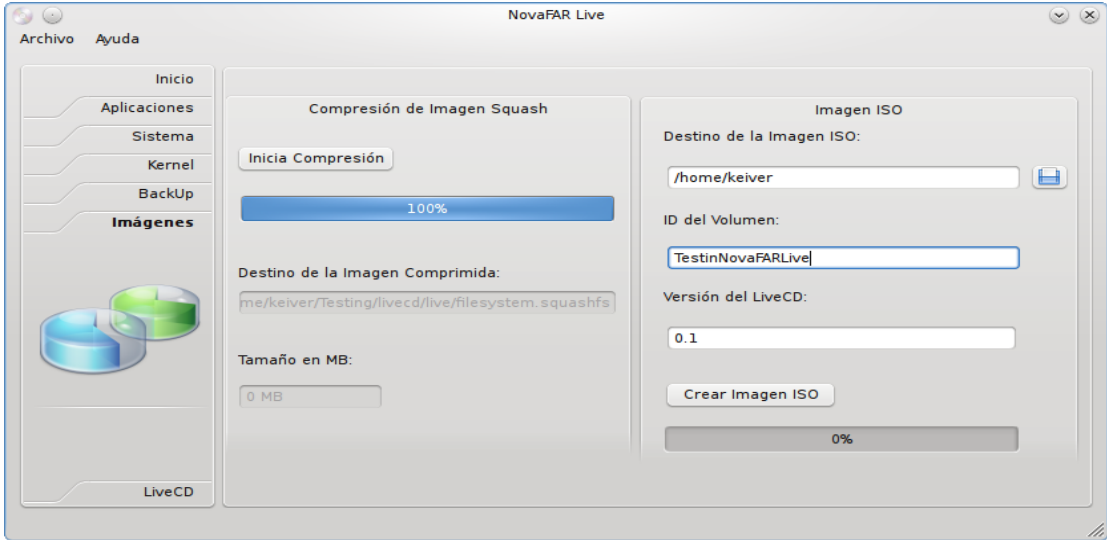
Historia de Usuario.	
Número: HU-NFL-4	Nombre: Crear imagen ISO.
Modificación de Historia de Usuario: Ninguna	
Usuario: Carlos A. García Rodríguez. Elisa Gamez López.	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
<p>Descripción: Crea una imagen ISO lista para usarse a partir del sistema comprimido. Se calcula su tamaño y se aconseja al usuario si debe usar un CD o un DVD.</p>	
<p>Prototipo de interface:</p> 	

Tabla 13. HU Crear imagen ISO.

Tarea de Ingeniería.	
Número Tarea: 1	Número Historia de Usuario: HU-NFL-4
Nombre Tarea: Obtener valores de configuración de la imagen ISO.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 20/3/2010	Fecha Fin: 23/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Obtiene los valores para la imagen ISO. Entre ellos, la versión del LiveCD/DVD, el ID del volumen y la dirección donde se guardará la imagen ISO.	

Tabla 14. Tarea de Ingeniería #1 de la HU Crear imagen ISO.

Tarea de Ingeniería.	
Número Tarea: 2	Número Historia de Usuario: HU-NFL-4
Nombre Tarea: Crear expresión regular para obtener el por ciento de creada de la imagen ISO.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 20/3/2010	Fecha Fin: 23/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Obtiene el por ciento de creada de la imagen ISO. Necesaria para mostrarlo en la interfaz gráfica. El nombre de la expresión es <i>int_percent_exp</i> .	

Tabla 15. Tarea de Ingeniería #2 de la HU Crear imagen ISO.

Historia de Usuario.	
Número: HU-NFL-5	Nombre: Adicionar Perfil.
Modificación de Historia de Usuario: Ninguna.	
Usuario: Carlos A. García Rodríguez. Elisa Gamez López.	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Se crea el perfil con el nombre y en la ruta seleccionada por el usuario.	
Prototipo de interface:	
	

Tabla 16. HU Adicionar Perfil.

Tarea de Ingeniería.	
Número Tarea: 1	Número Historia de Usuario: HU-NFL-5
Nombre Tarea: Nombrar el perfil.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 3/3/2010	Fecha Fin: 10/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Se crea el perfil con el nombre elegido por el usuario.	

Tabla 16. Tarea de Ingeniería #1 de la HU Adicionar Perfil.

Tarea de Ingeniería.	
Número Tarea: 2	Número Historia de Usuario: HU-NFL-5
Nombre Tarea: Ruta del área de trabajo.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 3/3/2010	Fecha Fin: 10/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: El perfil se guardara en la ruta seleccionada por usuario.	

Tabla 17. Tarea de Ingeniería #2 de la HU Adicionar Perfil.

Historia de Usuario.	
Número: HU-NFL-6	Nombre: Configuración.
Modificación de Historia de Usuario: Ninguna.	
Usuario: Carlos A. García Rodríguez. Elisa Gamez López.	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1
Descripción: Se gestionan los parámetros para la configuración a utilizar en la creación del proyecto. Entre ellos la verificación del usuario con que se ejecuta la aplicación, la gestión del kernel y del script management.sh.	

Tabla 18. HU Configuración.

Tarea de Ingeniería.	
Número Tarea: 1	Número Historia de Usuario: HU-NFL-6
Nombre Tarea: Copia del script management.sh necesario para chroot.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha Inicio: 23/3/2010	Fecha Fin: 30/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Copia el script necesario para el chroot, este script es el que hace las tareas de mantenimiento dentro del sistema recién copiado.	

Tabla 19. Tarea de Ingeniería #1 de la HU Configuración.

Tarea de Ingeniería.	
Número Tarea: 2	Número Historia de Usuario: HU-NFL-6
Nombre Tarea: Gestionar la selección del Kernel del sistema.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23/3/2010	Fecha Fin: 30/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Gestiona todo lo relacionado con el Kernel a utilizar por el LiveCD/DVD. Se muestra una lista de los Kernels disponibles en el sistema, y es posible seleccionar uno personalizado.	

Tabla 20. Tarea de Ingeniería #2 de la HU Configuración.

Tarea de Ingeniería.	
Número Tarea: 3	Número Historia de Usuario: HU-NFL-6
Nombre Tarea: Gestionar los permisos de la aplicación.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23/3/2010	Fecha Fin: 30/3/2010
Programador Responsable: Carlos A. García Rodríguez. Elisa Gamez López.	
Descripción: Gestiona los permisos para ejecutar la aplicación como root. Al inicio de la aplicación se chequea el usuario que ejecuta la misma, si es root, se ejecuta normalmente, de lo contrario se muestra un mensaje de información y se cierra la aplicación.	

Tabla 21. Tarea de Ingeniería #3 de la HU Configuración.

2.4. Diagrama de clases.

La siguiente figura representa el diagrama de clases propuesto para la aplicación a desarrollar, donde se ven representadas las principales clases que tendrá. En una breve explicación de las clases representadas en la aplicación, podemos decir que la clase LiveCD se recogen las principales características del mismo. Por su parte MainWindow es la clase que se encarga de realizar las principales operaciones relacionadas con el proceso de creación del LiveCD, en ella se controlan los diferentes procesos que se ejecutan con el fin de logran un correcto funcionamiento de la aplicación. También se encuentra la clase Mensaje la cual nos permite mostrar cualquier mensaje y por último la clase help que es la encargada de guiar al usuario durante el empleo de la aplicación.

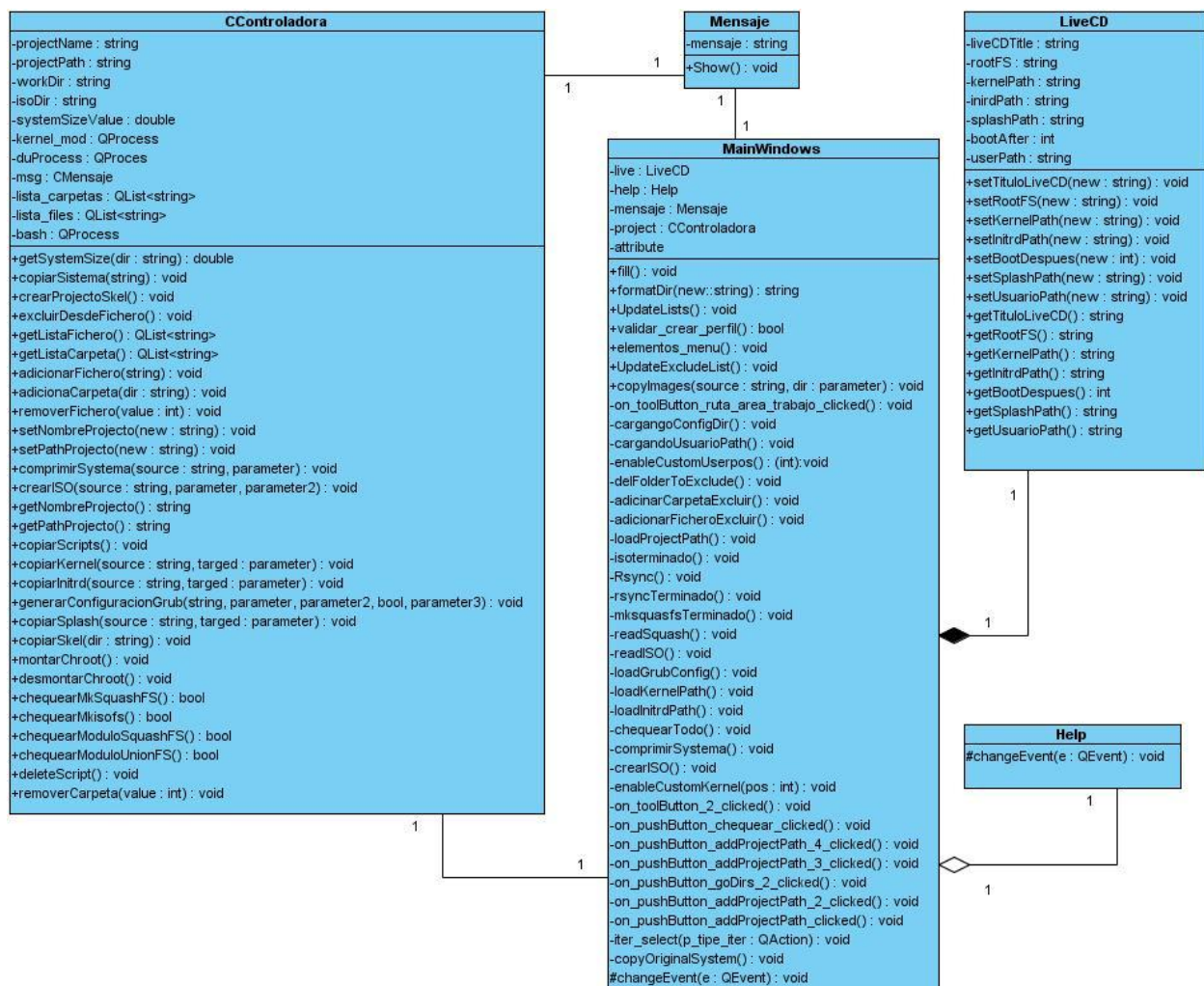


Figura 2: Diagrama de Clases.

2.5. Diagrama de componentes.

El diagrama de componente describe los elementos físicos del sistema y sus relaciones. Muestra las opciones de realización incluyendo códigos fuente, binarios y ejecutables. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros. A continuación se presente el diagrama de componentes para el sistema propuesto.

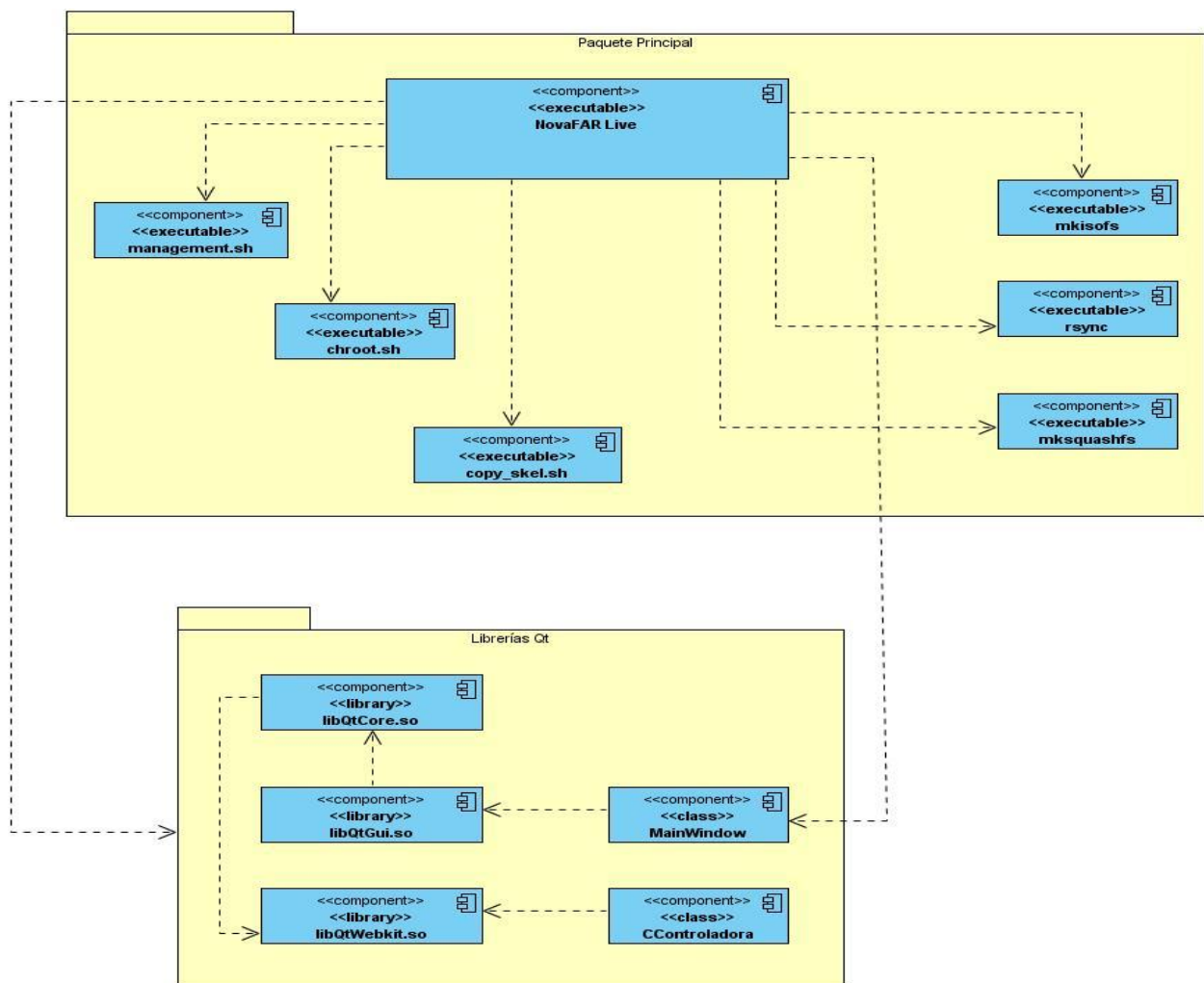


Figura 3: Diagrama de Componentes.

2.6. Plan de releases.

En este paso se define el plan de releases e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una

historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha historia. Como resultado de la priorización de historias se llegó a la siguiente planificación:

Release	Historias de Usuarios	Tiempo Estimado(Semanas)
1	1,5,6	1
2	2	1
3	4,3	1

Tabla 22. Plan de releases.

En el desarrollo de este capítulo se planificó el proyecto en forma de historias de usuario, tareas de ingeniería y diagramas auxiliares que nos hacen más fácil la comprensión del sistema. Se analizaron algunos diagramas auxiliares como el de componentes, el diagrama de clases y de dominio, así como la lista de reservas del producto. De la misma forma se hizo la planificación de los releases teniendo en cuenta las principales funcionalidades que la aplicación debe tener.

Capítulo 3: Validación de la propuesta del Software NovaFAR Live.

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

3.1. Casos de Pruebas.

La metodología SXP define entre iteración e iteración un conjunto de casos de pruebas de aceptación para poder avanzar a una iteración superior. Durante el desarrollo del presente trabajo, no solo se diseñó un conjunto de pruebas a la que fue sometida la herramienta para comprobar el funcionamiento de acuerdo a las Historias de Usuarios, si no que se confeccionó un Plan de Prueba para organizar el desarrollo de las mismas; para más información ver Anexo # 2.

3.2. Propuesta del sistema a implementar.

Antes de pasar a definir los casos de pruebas de aceptación para cada historia de usuario, es necesario ver la propuesta del sistema a implementar. Después de un estudio realizado a nivel mundial de todas las herramientas usadas para generar LiveCD/DVDs, y echo una comparación de las dos herramientas usadas en el centro, se concluye que ninguna es una solución factible para generar un LiveCD/DVD usando una interfaz gráfica de forma fácil y configurable. Por lo que se decide comenzar la implementación de una interfaz visual para la herramienta NovaFAR Live la cual hereda gran parte de las características y funcionalidades de estos.

En el proceso de implementación de la aplicación se debe utilizar una capa de presentación y una capa lógica de negocio. En la primera estará la interfaz que el usuario observará y que se implementará en Qt, mientras que la capa lógica de negocio contendrá todo el núcleo de la aplicación, principales clases y funcionalidades. La interfaz gráfica de la aplicación estará compuesta por una barra de menú que dará acceso a funciones básicas de aplicación como la opción de salir y el acceso a la ayuda. El funcionamiento general de la misma está basado en las distintas opciones de configuración que el usuario deberá ir completando para ir realizando los pasos necesarios para

generar el LiveCD/DVD. Primeramente la aplicación solicitará el nombre del proyecto el cual será el nombre del LiveCD/DVD que se quiera generar, así como la ruta donde se guardarán los ficheros necesarios para que la aplicación genere el mismo de forma correcta. Seguidamente se pasará a la sección donde se configurarán los directorios principales del LiveCD/DVD a generar. Se seleccionará el directorio raíz que no es más que la dirección donde está el sistema a usar en el LiveCD/DVD. Es aquí también donde se deben seleccionar los directorios y ficheros a excluir en el proceso, así como la dirección de la carpeta personal con la configuración del usuario que se usará en el LiveCD/DVD.

Una vez configurados los directorios se procederá a la selección del kernel y la imagen initrd a usar en el LiveCD/DVD, se listarán los disponibles en el sistema y además se dará la posibilidad de seleccionar uno personalizado. Finalmente se copiará el sistema previamente seleccionado y se comprimirá usando un sistema squashfs y se generará la imagen ISO lista para grabar en un medio extraíble. El sistema debe permitir chequear las aplicaciones necesarias para el correcto funcionamiento del LiveCD/DVD, informándole al usuarios si están o no instaladas.

3.3. Arquitectura de la aplicación.

La aplicación propuesta estará desarrollada en Qt, que es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario. Dentro de las principales clases usadas en el desarrollo de la aplicación, se encuentran QProcess y QObject. QProcess es una clase usada para gestionar la ejecución de procesos dentro de las aplicaciones desarrolladas con Qt. Entre otras funcionalidades esta clase permite, ejecutar un programa y a la vez adicionar parámetros al mismo. Una vez ejecutado un proceso con dicha clase, es posible leer desde la salida estándar o la salida de error; este parámetro es configurable una vez inicializado el objeto de tipo QProcess. Esta clase permite además poner variables de entorno independientemente de la plataforma en que se esté ejecutando la aplicación. Otra de las funcionalidades fundamentales de QProcess, es que puede ejecutar proceso como hijos de otros procesos. QObject es la clase base de todos los objetos de Qt y es el centro del modelo de objetos de Qt. La característica fundamental de este poderoso mecanismo es el sistema de comunicación entre objetos llamado signals y slots, los cuales no son más que las señales que emiten los mismos y las funciones que se ejecutan al ser llamadas estos. Como parte de la arquitectura básica de la aplicación, se implementan las clases LiveCD, CControladora y MainWindow, encargadas de gestionar los objetos principales de la misma. LiveCD es la clase que contiene todos los atributos del LiveCD. CControladora es la clase que contiene las funcionalidades que debe tener el sistema para su buen funcionamiento. En ella se realiza la copia del sistema origen, la compresión de dicha copia y la

creación de la imagen ISO, entre otras importantes funcionalidades. La clase MainWindow es la responsable de gestionar los elementos gráficos.

3.4. Listado de Historias de Usuarios a Probar.

Luego de realizar un estudio de todo el sistema a implementar y de la arquitectura de la aplicación, se decide realizar pruebas a las historias de usuarios.

3.4.1. Caso de Prueba Historias de Usuarios: HU-NFL-1

Este caso de prueba pertenece a la Historia de Usuario: Programar Scripts Auxiliares, se valida que los script auxiliares funcionen correctamente. Se analiza el script management.sh que es el encargado de realizar tareas de optimización en el sistema. También se comprobará el correcto funcionamiento del script chroot.sh, el cual está encargado de montar las particiones para un correcto funcionamiento del comando chroot en el sistema del cual se creara el LiveCD. Finalmente se comprueba que las particiones estén desmontadas usando el script umount.sh.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 1	Nombre Historia de Usuario: HU-NFL-1
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Después de terminar la copia del sistema original, se montan las particiones en los directorios definidos, posteriormente se ejecuta el script management.sh el cual optimiza el sistema copiado.	
Condiciones de Ejecución: Que la aplicación se esté ejecutando como root y que se copie un sistema de archivos válidos.	
Entrada / Pasos de ejecución: El script chroot monta las particiones y ejecuta al script management.sh en el nuevo sistema.	
Resultado Esperado: Se debe ejecutar el script management.sh de forma correcta y este debe hacer todas las tareas definidas.	

Evaluación de la Prueba: Prueba Satisfactoria.

Tabla 23: Caso de Prueba de Aceptación #1 de la HU: NFL-1.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 2	Nombre Historia de Usuario: HU-NFL-1
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se copia la información personal del usuario la cual aparecerá con el objetivo de que las configuraciones estén en el LiveCD/DVD. Se da la posibilidad de seleccionar los datos.	
Condiciones de Ejecución: Se debe seleccionar previamente un usuario.	
Entrada/Pasos de ejecución: Se copia el nombre del usuario o la dirección de su carpeta personal y se copian los datos en la carpeta /etc/skel del sistema a usar en el LiveCD/DVD.	
Resultado Esperado: Que se copien los datos del usuario satisfactoriamente.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Tabla 24: Caso de Prueba de Aceptación #2 de la HU: NFL-1.

3.4.2. Caso de Prueba Historias de Usuarios: HU-NFL-2.

Este caso de prueba pertenece a la Historia de Usuario: Copia del sistema origen, aquí primeramente debe de estar instalada la herramienta rsync. Se comprobará que se haya realizado una copia del sistema original correctamente. Se probarán las expresiones regulares para obtener los valores de por ciento y el nombre del fichero que se está copiando respectivamente.

Caso de Prueba de Aceptación.

Código Caso de Prueba: 1	Nombre Historia de Usuario: HU-NFL-2.
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se realiza la copia del sistema original con las carpetas de directorios a incluir, para eso se usa la herramienta rsync.	
Condiciones de Ejecución: Debe especificarse el sistema original que se va a copiar.	
Entrada/Pasos de ejecución: Se le pasa a la aplicación rsync los parámetros como ficheros y directorios que no son necesarios para el LiveCD/DVD y las direcciones del sistema a copiar.	
Resultado Esperado: Que se realice la copia del sistema raíz correctamente.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Tabla 25: Caso de Prueba de Aceptación #1 de la HU: NFL-2.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 2	Nombre Historia de Usuario: HU-NFL-2
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: La copia del sistema produce una salida que se empleara para saber el por ciento de ejecución de este proceso chequeando la expresión regular que permite obtener dicho valor.	
Condiciones de Ejecución: Que se esté ejecutando la copia del sistema.	
Entrada/Pasos de ejecución: Se usa la salida del rsync para saber el porcentaje de ejecución de la copia y se le pasa a la barra de progreso de la copia el patrón	

encontrado por la expresión regular.
Resultado Esperado: Que la expresión regular para mostrar el por ciento de la copia funcione correctamente.
Evaluación de la Prueba: Prueba Satisfactoria.

Tabla 26: Caso de Prueba de Aceptación #2 de la HU: NFL-2.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 3	Nombre Historia de Usuario: HU-NFL-2
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: La copia del sistema produce una salida que se empleara para saber el por ciento de ejecución de este proceso chequeando la expresión regular que permite obtener el nombre del fichero que se está copiando.	
Condiciones de Ejecución: Que se esté ejecutando la copia del sistema.	
Entrada/Pasos de ejecución: Se usa la salida del rsync para saber el por ciento de ejecución de la copia. Se lee la salida estándar del proceso y luego la expresión regular obtiene el nombre del fichero que se está copiando.	
Resultado Esperado: Que la expresión regular usada para obtener el nombre del fichero, funcione correctamente.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Tabla 27: Caso de Prueba de Aceptación #3 de la HU: NFL-2.

3.4.3. Caso de Prueba Historias de Usuarios: HU-NFL-3.

Capítulo 3: Validación de la propuesta del Software NovaFAR Live.

Este caso de prueba pertenece a la Historia de Usuario: Compactar imagen, primeramente debe de estar instalada la herramienta mksquashfs. Se verifica que la compresión se haya realizado correctamente. También se comprobará la expresión regular usada para obtener el por ciento de progreso de compresión de la imagen. Finalmente se calcula el tamaño de la imagen comprimida para darle una noción al usuario de cual dispositivo de almacenamiento usar, ya sea CD o DVD.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 1	Nombre Historia de Usuario: HU-NFL-3
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se realiza la compresión del sistema original copiado.	
Condiciones de Ejecución: Se tiene que haber realizado la copia del sistema raíz.	
Entrada/Pasos de ejecución: Con la herramienta mksquashfs se realiza la compresión del sistema, esta herramienta recibe la dirección del directorio a comprimir y la dirección donde se guardara la imagen comprimida.	
Resultado Esperado: Se debe realizar la compresión del sistema de forma correcta.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Tabla 28: Caso de Prueba de Aceptación #1 de la HU: NFL-3.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 2	Nombre Historia de Usuario: HU-NFL-3
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: La compresión del sistema produce una salida que se	

usara para saber el por ciento de ejecución de este proceso, se chequeará la expresión regular para obtener dicho por ciento.
Condiciones de Ejecución: Que se esté ejecutando la compresión del sistema.
Entrada/Pasos de ejecución: La salida de la compresión se empleara para saber el por ciento de ejecución de este proceso. Se obtiene el por ciento de compresión que se le pasara a la barra de progreso gracias a la expresión regular usada para encontrar patrones dentro de las cadenas leídas.
Resultado Esperado: La expresión regular para obtener el por ciento de compresión debe funcionar correctamente.
Evaluación de la Prueba: Prueba Satisfactoria.

Tabla 29: Caso de Prueba de Aceptación #2 de la HU: NFL-3.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 3	Código Caso de Prueba: 3
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se calcula el tamaño de la imagen.	
Condiciones de Ejecución: Que la compresión de la imagen se haya realizado.	
Entrada/Pasos de ejecución: Una vez realizada la compresión se calcula el tamaño de la imagen comprimida para mostrarlo en la interfaz.	
Resultado Esperado: Se debe mostrar el tamaño de una forma correcta.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Tabla 30: Caso de Prueba de Aceptación #3 de la HU: NFL-3.

3.4.4. Caso de Prueba Historias de Usuarios: HU-NFL-4.

Este caso de prueba pertenece a la Historia de Usuario: Crear imagen ISO. Primeramente debe de estar instalada la herramienta mkisofs, se verificará la creación de la imagen ISO. También se probará la expresión regular usada para obtener el por ciento de progreso de creación de la imagen. Por último se probará que la entrada de datos para la creación de la imagen ISO.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 1	Nombre Historia de Usuario: HU-NFL-4
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se creará la imagen ISO del sistema comprimido.	
Condiciones de Ejecución: La compresión del sistema original debe haberse realizado.	
Entrada/Pasos de ejecución: Se emplea la herramienta mkisofs para crear la imagen, dicha herramienta recibe la dirección del directorio donde está preparada la estructura del LiveCD/DVD y la dirección donde se guardará el ISO a crear.	
Resultado Esperado: Que el archivo .iso se genere de forma correcta.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Tabla 31: Caso de Prueba de Aceptación #1 de la HU: NFL-4.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 2	Nombre Historia de Usuario: HU-NFL-4
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	

<p>Descripción de la Prueba: La generación de la imagen ISO produce una salida la cual se emplea para saber el porcentaje de creado el .iso y se chequea la expresión regular encargada de dicho proceso.</p>
<p>Condiciones de Ejecución: Que la creación de la imagen ISO se este ejecutando.</p>
<p>Entrada/Pasos de ejecución: La salida obtenida de la generación de la imagen se usara para saber el por ciento de creación de este proceso, dicho por ciento obtenido a través de una expresión regular usada para encontrar patrones dentro de la cadena leída, se le pasa a la barra de progreso.</p>
<p>Resultado Esperado: La expresión regular usada para obtener el por ciento de creada la imagen ISO debe funcionar correctamente.</p>
<p>Evaluación de la Prueba: Prueba Satisfactoria.</p>

Tabla 32: Caso de Prueba de Aceptación #2 de la HU: NFL-4.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 3	Nombre Historia de Usuario: HU-NFL-4
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se probará la entrada de los datos para la creación del LiveCD/DVD.	
Condiciones de Ejecución: Que se haya realizado la compresión del sistema.	
Entrada/Pasos de ejecución: Se chequea que se entre el ID de Volumen y la versión del LiveCD/DVD. Si el usuario no entra algún dato necesario, se muestra un mensaje de error.	
Resultado Esperado: Se deben entrar todos los datos de una forma correcta.	

Evaluación de la Prueba: Prueba Satisfactoria.

Tabla 33: Caso de Prueba de Aceptación #3 de la HU: NFL-4.

3.4.5. Caso de Prueba Historias de Usuarios: HU-NFL-5.

Este caso de prueba pertenece a la Historia de Usuario: Adicionar Perfil. Aquí el usuario debe de introducir el nombre del proyecto y la ruta de trabajo, de no ser así, se le mostrará un mensaje pidiéndole que llene las casillas en blanco de lo contrario no pasará al siguiente paso.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 1	Nombre Historia de Usuario: HU-NFL-5
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se debe comprobar que el usuario haya ingresado el nombre y la ruta.	
Condiciones de Ejecución: Que el perfil no esté creado.	
Entrada/Pasos de ejecución: Se chequea que el usuario haya entrado el nombre del proyecto y la ruta de trabajo, de no ser así se muestra un mensaje de alerta.	
Resultado Esperado: Que el usuario no deje casilla en blanco.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Tabla 34: Caso de Prueba de Aceptación #1 de la HU: NFL-5.

3.4.6. Caso de Prueba Historias de Usuarios: HU-NFL-6.

Este caso de prueba pertenece a la Historia de Usuario: Configuración. Aquí se verifica que la gestión de la configuración funcione correctamente. Se chequeará la copia del script management.sh, encargado de ejecutar la optimización del sistema original después de copiado. Se comprobará los

Capítulo 3: Validación de la propuesta del Software NovaFAR Live.

permisos necesarios para la ejecución de la aplicación y se verá también la gestión de la configuración del Kernel.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 1	Nombre Historia de Usuario: HU-NFL-6
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se copia el script management.sh además de cambiársele los permisos para que pueda ser ejecutado en el nuevo sistema usando chroot.	
Condiciones de Ejecución: Que el sistema original ya este copiado.	
Entrada/Pasos de ejecución: Se copia el script management.sh y se le cambian los permisos para ser ejecutado en el nuevo sistema usando chroot. Entre las tareas que debe hacer este script se encuentran la de borrar archivos y usuarios innecesarios así como crear otros ficheros importantes.	
Resultado Esperado: Que se copie correctamente el archivo management.sh al sistema que recién se copio.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Tabla 35: Caso de Prueba de Aceptación #1 de la HU: NFL-6.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 2	Nombre Historia de Usuario: HU-NFL-6
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Esta prueba es necesaria para arrancar el	

LiveCD/DVD.
Condiciones de Ejecución: Se debe haber seleccionado el kernel y el initrd.
Entrada/Pasos de ejecución: Se copian los ficheros del Kernel y el initrd previamente seleccionados con los nombres vmlinuz e initrd.img respectivamente a la carpeta del LiveCD/DVD para ser usados en el arranque del mismo.
Resultado Esperado: Que se copien correctamente los ficheros vmlinuz y initrd.img.
Evaluación de la Prueba: Prueba Satisfactoria.

Tabla 36: Caso de Prueba de Aceptación #2 de la HU: NFL-6.

Caso de Prueba de Aceptación.	
Código Caso de Prueba: 3	Nombre Historia de Usuario: HU-NFL-6
Nombre de la persona que realiza la prueba: Elisa Gamez López. Carlos A. García.	
Descripción de la Prueba: Se probará que la aplicación solo sea accesible por el usuario root, pues se necesitan los permisos para hacer chroot al sistema copiado a usar en el LiveCD/DVD.	
Condiciones de Ejecución: Que el usuario que inicie la aplicación sea root.	
Entrada/Pasos de ejecución: Al iniciar la aplicación si no es root el usuario que la inicia, se muestra un mensaje alertando al usuario que la aplicación debe tener permisos de administración y la aplicación termina. De lo contrario la aplicación se ejecuta de forma normal.	
Resultado Esperado: Que si no es root el usuario que ejecuta la aplicación, que se muestre un mensaje de error y termine. De lo contrario que se ejecute normalmente.	

Evaluación de la Prueba: Prueba Satisfactoria.

Tabla 37: Caso de Prueba de Aceptación #3 de la HU: NFL-6.

3.5. Resultados Obtenidos.

En este epígrafe se dan a conocer los resultados obtenidos hasta el momento por el equipo de desarrollo de NovaFAR Live. Resaltar que como resultado de este trabajo, NovaFAR Live está disponible en su versión 0.1.2, lo cual significa que se pueden esperar resultados en versiones posteriores.

3.5.1. Acerca del tiempo de desarrollo.

Desde el curso pasado se viene analizando que la herramienta existente en el centro para la generación de LiveCD/DVDs no cumple con las necesidades de los desarrolladores del sistema operativo ya que esta era a través de consola, es por esto que se decide llevar a cabo un estudio sobre el tema, principalmente en aplicaciones que hacían este trabajo en varias distribuciones de Linux.

Aproximadamente nos tomó un tiempo aprender a usarlas y probarlas. Luego se comenzó a pensar ya en realizarle una interfaz visual a una de las herramientas existentes en el centro, hasta que por fin se decide llevar a cabo el desarrollo de la misma. Luego se comenzó el análisis y el diseño de la aplicación propuesta la cual incluía las principales funcionalidades y características de los sistemas analizados.

De todas las tareas de implementación que se llevaron a cabo la que más tiempo consumió fue la creación de los scripts escritos en Bash, usados para optimizar el sistema a usar en el LiveCD/DVD, aproximadamente tomó un mes para que estuviesen listos. Luego se comenzó a desarrollar la interfaz gráfica para la aplicación la cual fue desarrollada en un tiempo muy corto pues las bondades de las librerías Qt4 son muchas y fáciles de usar.

3.5.2. Acerca de las Funcionalidades Obtenidas.

Entre las principales funcionales que posee NovaFAR Live hasta su última versión, se pueden mencionar:

Capítulo 3: Validación de la propuesta del Software NovaFAR Live.

1. Adición o eliminación de ficheros y directorios a excluir del LiveCD/DVD.
2. Selección del sistema raíz personalizado, no tiene porque ser el del sistema que se está usando.
3. Posibilita seleccionar el directorio personal de un usuario para usar esa configuración en el LiveCD/DVD.
4. Configuración del Boot Splash y el título del LiveCD/DVD, así como del tiempo en que demorará en arrancar el mismo.
5. Es posible seleccionar el Kernel y el Initrd a usar en el LiveCD/DVD de manera fácil seleccionando desde una lista o se puede especificar manualmente otro cualquiera.
6. La aplicación es capaz de chequear si está instalada la herramienta rsync. Necesaria para hacer una copia del sistema original. .
7. La aplicación es capaz de chequear si están instalados los módulos del Kernel necesarios para el correcto funcionamiento de la misma.
8. Es capaz de hacer chroot al sistema a usar en el LiveCD/DVD y hacer modificaciones para optimizar el sistema. Esta funcionalidad le da robustez a la aplicación pues sin tener que recompilar, solo modificando los scripts necesarios, se pueden hacer nuevas tareas de mantenimiento dentro del sistema a usar en el LiveCD/DVD.
9. Comprime el sistema usando squashfs.
10. Es capaz de mostrar el tamaño de la imagen comprimida, dándole al usuario una idea del tamaño que será el LiveCD/DVD final.
11. Genera un LiveCD/DVD auto-arrancable usando Grub como gestor de arranque.

En este capítulo se presentó una propuesta de la aplicación que se desarrollaría en la que se explicó con detalles cuales serían las principales funcionalidades y características de la misma que serían implementadas. Complementando esta sección, se presentó una propuesta de la arquitectura de la misma.

Se presentaron algunos casos de pruebas que guiaron la calidad del sistema, y determinaron en cada momento si se estaba o no en condiciones de continuar avanzando. Dichas pruebas ayudaron a que la aplicación en su fase de entrega tuviese muy pocos errores en cuanto a funcionamiento en general. Todas ellas fueron. En el análisis de los resultados obtenidos se muestran las funcionalidades alcanzadas por el sistema en el período que se ha estado trabajando en su desarrollo.

Conclusiones.

EL Proyecto Personalización para Sistemas Operativos en la UCID, necesita mejorar la creación de LiveCD/DVDs para facilitar el proceso de instalación y despliegue. Por lo que, con la confección de este Trabajo de Diploma se obtuvo una herramienta con una interfaz visual agradable al usuario y mucho más cómoda; donde se logró a través de un estudio sobre las principales aplicaciones existentes a nivel mundial para la generación de LiveCD/DVDs en diferentes plataformas, de las cuales la mayor parte de ellas son a nivel de consola lo que hace que el proceso de creación de este sea una tarea complicada para el usuario. Además se analizaron las principales características y funcionalidades de varias aplicaciones llegando a un consenso de cuales debería tener la aplicación propuesta. Donde se puede observar como resultado de la creación de esta herramienta:

- ✓ La manera más adecuada de gestionar los directorios y ficheros que se incluirán en el LiveCD/DVD a generar.
- ✓ La realización de configuraciones personales del usuario.
- ✓ El chequeo de las herramientas necesarias para que el LiveCD/DVD generado funcione de forma correcta. Con el objetivo de incluir gran cantidad de software dentro del medio auto-arrancable donde se comprime el sistema usando un sistema de ficheros que permite amplios radios de compresión.

De esta manera se implementó y probó la aplicación NovaFAR Live que permite generar un LiveCD/DVD de forma fácil y rápida a partir de un sistema Nova GNU/Linux instalado en el ordenador.

Recomendaciones.

Como producto final del presente trabajo, la aplicación NovaFAR Live permite la generación de un LiveCD/DVD de forma rápida y fácil para usuarios finales. Pero con el objetivo de lograr una mayor robustez, se recomienda:

1. Implementar las funcionalidades seleccionar perfil y quemar CD/DVD.
2. Incorporar la opción de “Instalar”, una vez obtenido el ISO facilitando al usuario iniciar la instalación.
3. Probar el producto a gran escala a fin de validar su funcionamiento.

Referencias Bibliográficas.

[1] Free Software Foundation. Free Software. [Fecha de consulta: 19 de noviembre del 2009]. Disponible en: <<http://www.gnu.org/philosophy/free-sw.html>>

[2] [Castro Fidel, 2002] Discurso de Fidel Castro, 2002.

[3] Ing. Goñi Oramas, Ángel. XVI Fórum de Ciencia y Técnica. Nova, Distribución Cubana de GNU/Linux. UCI 2009.

[4] Hdez. Fdez., Keiver. *Aplicación para Generar LiveCD/DVDs de Sistemas Debian GNU/Linux*. UCI, 2009. Disponible en: <<http://gforge.f10.uci.cu/projects/inouelive/>>

[5] Debian GNU/Linux Packages. Dfsbuild. [Fecha de consulta: 21 de noviembre del 2009]. Disponible en: <<http://packages.debian.org/es/etch/dfsbuild>>

[6] Debian Administration Team. Bootcd. [Fecha de consulta: 22 de noviembre del 2009]. Disponible en: <<http://www.debian-administration.org/articles/148>>

[7] Fedora Project Team. LiveCD-Tools. [Fecha de consulta: 22 de noviembre del 2009]. Disponible en: <<http://fedoraproject.org/wiki/FedoraLiveCD>>

[8] Debian GNU/Linux Packages. Live-Magic. [Fecha de consulta: 23 de noviembre del 2009]. Disponible en: <<http://packages.debian.org/lenny/live-magic>>

[9] EsDebian.org Team. Live-Helper. [Fecha de consulta: 23 de noviembre del 2009]. Disponible en: <<http://www.esdebian.org/wiki/live-helper>>

[10] GuiaUbuntu.org Team. Remastersys. [Fecha de consulta: 23 de noviembre del 2009]. Disponible en: <<http://www.guia-ubuntu.org/index.php?title=Remastersys>>

[11] Linux Live Team. Linux Live Scripts. [Fecha de consulta: 24 de noviembre del 2009]. Disponible en: <<http://www.linux-live.org/>>

[12] UCK Team. Ubuntu Customization Kit. [Fecha de consulta: 24 de noviembre del 2009]. Disponible en: <<http://uck.sourceforge.net/>>

[13] Reconstructor Team. Reconstructor. [Fecha de consulta: 24 de noviembre del 2009]. Disponible en: <<http://reconstructor.aperantis.com/>>

[14] Nova Team. Nova LiveCD-Kit. [Fecha de consulta: 24 de noviembre del 2009]. Disponible en: <<http://gforge.f10.uci.cu/projects/nova>>

- [15] Calculate Linux [Fecha de consulta: 4 enero 2010]. Disponible en: <http://old.calculate-linux.org/Calculate_Linux>
- [16] Qt Software. Librerías Qt. [Fecha de consulta: 25 de noviembre del 2009]. Disponible en: <<http://doc.trolltech.com/>>
- [17] Qt Creator Team. QtCreator. [Fecha de consulta: 26 de noviembre del 2009]. Disponible en: <<http://www.qtsoftware.com/developer/qt-creator>>
- [18] Qt Designer Manual [Fecha de consulta: 2 de febrero del 2010]. Disponible en: <<http://doc.trolltech.com/4.3/designer-manual.html>>
- [19] Signals and Slots [fecha de consulta: 2 de febrero del 2010]. Disponible en: <<http://doc.trolltech.com/4.6/signalsandslots.html>>
- [20] Tomas M. Squash LZMA. [Fecha de consulta: 28 de noviembre del 2009]. Disponible en: <<http://www.squashfs-lzma.org>>
- [21] AufS Team. Another UnionFS. [Fecha de consulta: 29 de noviembre del 2009]. Disponible en: <<http://aufs.sourceforge.net/>>
- [22] SquashFS Team. SquashFS. [Fecha de consulta: 6 diciembre del 2009]. Disponible en: <<http://squashfs.sourceforge.net/>>
- [23] Wayne Davison. Rsync. [Fecha de consulta: 29 de noviembre del 2009]. Disponible en: <<http://samba.anu.edu.au/rsync>>
- [24] OMG Team. UML. [Fecha de consulta: 6 diciembre del 2009]. Disponible en: <<http://www.omg.org/technology/documents/formal/uml.htm>>
- [25] GNU.org Team. Bash Reference Manual. [Fecha de consulta: 6 diciembre del 2009]. Disponible en: <<http://www.gnu.org/software/bash/manual/bashref.html>>
- [26] Juan Soulie. Programming with C/C++. [Fecha de consulta: 7 diciembre del 2009]. Disponible en: <<http://www.cplusplus.com/info/description.html>>
- [27] GLADYS M P. R., 2008, MA-GMPR-UR2, Metodología ágil para proyectos de software libre. Disponible en: <<http://gforge.f10.uci.cu/projects/magmpr/>>

Bibliografía.

Aufs Team. Another UnionFS. Disponible en: <<http://aufs.sourceforge.net/>>

Capink Ubuntu Forums User. LiveCD/DVD HowTo. Disponible en: <<http://www.remastersys.klikit-linux.com/capink.html>>

CDRKit Team. Portable Command-Line CD/DVD Recorder Software. Disponible en: <http://www.cdrkit.org/>>

Fidel C. R., 2004, Discurso en la clausura del VIII Congreso de la UJC. Disponible en <http://www.cuba.cu/gobierno/discursos/2004/esp/f051204e.html>>

Free Software Foundation. Free Software. Disponible en: *Inoue Live Creator 92*
<<http://www.gnu.org/philosophy/free-sw.html>>

GLADYS M P. R., 2008, MA-GMPR-UR2, Metodología ágil para proyectos de software libre. Disponible en:
<<http://gforge.f10.uci.cu/projects/magmpr/>>

GNU.org Team. Bash Reference Manual. Disponible en:
<http://www.gnu.org/software/bash/manual/bashref.html>

GNU.org Team. Bash. Disponible en: <<http://www.gnu.org/software/bash/>>

Juan Soulie. C/C++. Disponible en: <<http://www.cplusplus.com/info/description.html>>

Linux Live Team. Linux Live Scripts. Disponible en: < <http://www.linux-live.org/>>

LiveCD-Tools. Disponible en: < <http://fedoraproject.org/wiki/FedoraLiveCD>>

Nova Team. LiveCD-Kit. Disponible en: < <http://gforge.f10.uci.cu/projects/nova> >

OMG Team. UML. Disponible en: <<http://www.omg.org/technology/documents/formal/uml.htm>>

Qt Creator Team. QtCreator. Disponible en: < <http://www.qtsoftware.com/developer/qt-creator>>

Qt Designer Team. Qt Designer. Disponible en: <<http://doc.trolltech.com/3.3/designer-manual.html>>

Reconstructor Team. Reconstructor. Disponible en: <<http://reconstructor.aperantis.com/>>

SquashFS Team. SquashFS. Disponible en: <<http://squashfs.sourceforge.net/>>

Tomas M. Squash LZMA. Disponible en: < <http://www.squashfs-lzma.org/>>

Glosario.

A continuación en orden alfabético, se muestra el significado de algunos términos usados en el documento cuyo uso no es común y puede dificultar la comprensión del mismo:

B.

- ✓ **Backup:** Copia de seguridad realizada cada periodo corto de tiempo por si se pierden los datos originales de un sistema.

F.

- ✓ **Framework:** es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.
- ✓ **FSF:** La Fundación para el Software Libre (Free Software Foundation) es una organización creada en Octubre de 1985 por Richard Matthew Stallman y otros entusiastas del Software Libre con el propósito de difundir este movimiento. La Fundación para el Software Libre (FSF) está dedicada a eliminar las restricciones sobre la copia, redistribución, entendimiento, y modificación de programas de computadoras. Con este objeto, promociona el desarrollo y uso del software libre en todas las áreas de la computación, pero muy particularmente, ayudando a desarrollar el sistema operativo GNU.

G.

- ✓ **GPL:** La GNU General Public License (inglés: Licencia Pública General) es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es Software Libre.
- ✓ **Grub Splash:** Un Grub Splash es una imagen de 14 colores que se pone detrás del menú de selección del gestor de arranque Grub.
- ✓ **Grub:** En computación, el Gestor de Arranque Unificado (GRand Unified Bootloader), es un gestor de arranque múltiple que se usa comúnmente para iniciar dos o más sistemas operativos instalados en un mismo ordenador.

I.

- ✓ **IDE:** Entorno de desarrollo integrado o en inglés Integrated Development Environment (IDE). Es un programa compuesto por un conjunto de herramientas para un programador desde el que se pueden editar programas, compilarlos y depurarlos.
- ✓ **Init:** El propósito principal de init en el initramfs es preparar el montaje y el acceso al sistema de archivos raíz real. Entre otras tareas es responsable de cargar los módulos del núcleo y gestionar la configuración de la red.
- ✓ **Initramfs:** Es un pequeño sistema de archivos que el núcleo puede cargar en un disco RAM. Proporciona un entorno Linux mínimo que habilita la ejecución de programas antes de que se monte el sistema de archivos raíz. El entorno Linux mínimo se carga en la memoria mediante las rutinas de la BIOS y no necesita requisitos específicos de hardware, únicamente una memoria suficiente. Initramfs siempre debe proporcionar un ejecutable denominado init que ejecuta el programa init actual en el sistema de archivos raíz para que se lleve a cabo el proceso de arranque.
- ✓ **Instalar:** Incorporar a la computadora un programa o dispositivo para ser utilizado.
- ✓ **Interfaz:** Se refiere a la interfaz gráfica de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar.
- ✓ **ISO 9660:** El estándar ISO 9660 es una norma publicada inicialmente en 1986 por la ISO, que especifica el formato para el almacenaje de archivos en los soportes de tipo disco compacto. El estándar ISO 9660 define un sistema de archivos para CD-ROM. Su propósito es que tales medios sean legibles por diferentes sistemas operativos, de diferentes proveedores y en diferentes plataformas, por ejemplo, MS-DOS, Microsoft Windows, Mac OS y UNIX.
- ✓ **ISO:** Un ISO es un archivo donde se almacena una copia o imagen exacta de un sistema de ficheros, normalmente un disco compacto (como un CD o un DVD). Se rige por el estándar ISO 9660 de la Organización Internacional para la Normalización.

K.

- ✓ **Kernel:** Núcleo. Parte esencial de un sistema operativo que provee los servicios más básicos del sistema. Se encarga de gestionar los recursos como el acceso seguro al hardware de la

computadora. Se encarga también del multiplexado, determinando qué programa accederá a un determinado hardware si dos o más quieren usarlo al mismo tiempo.

L.

- ✓ **Libre:** O Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.
- ✓ **LiveCD/DVD:** Un Live CD o Live DVD, más genéricamente Live Distro, (traducido en ocasiones como CD vivo o CD autónomo), es un sistema operativo (normalmente acompañado de un conjunto de aplicaciones) almacenado en un medio extraíble, tradicionalmente un CD o un DVD (de ahí sus nombres), que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de ficheros.
- ✓ **LiveUSB:** Un Live USB es una Memoria USB que contiene un completo sistema operativo, el cual permite arrancar una espacio que ocupan en el disco y memoria. Haciendo transparente su descompresión. Ésta sólo se hace cuando se va a utilizar algo, con lo que no hay que descomprimir todo el sistema a RAM, sólo lo que se esté usando.

S.

- ✓ **SCRUM:** SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos.
- ✓ **Swap:** La swap o espacio de intercambio es una zona del disco (un fichero o partición) que se usa para guardar información sobre los procesos que no han de mantenerse en memoria física. Permitiendo liberar la memoria principal para cargar otros procesos.
- ✓ **SXP:** SXP es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

U.

- ✓ **UnionFS:** Es un servicio para sistemas de archivos de Linux que permite montar un sistema de archivos formado por la unión de otros sistemas de archivos de Linux. Permite que archivos y directorios de sistemas de archivos distintos, conocidos como ramas, se superpongan de forma transparente, formando un único sistema de archivos. Los contenidos de directorios que tienen la

misma ruta en las ramas que se combinan aparecerán juntos en un único directorio en el nuevo sistema de archivos virtual.

- ✓ **UNIX:** Unix (registrado oficialmente como UNIX®) es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

X.

- ✓ **XP:** Más bien es una metodología encaminada para el desarrollo; 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.

Anexos.

Anexo # 1: Lista de Reserva del producto.

Prioridad	Descripción	ítem	Estimación	Estimado por
Requisitos Funcionales				
Muy Alta	Programar script auxiliares.	1	1	Programador
	Crear script para la copia de la configuración personal.	2	1	Programador
	Obtener lista de directorios y ficheros a excluir. Obtener valores de configuración del gestor de arranque.	3	0.5	Analista
	Chequear aplicaciones necesarias en el sistema.	4	0.5	Analista
	Chequear los módulos necesarios en el Kernel.	5	0.5	Analista
	Copiar el sistema origen.	6	1	Programador
	Crear imagen comprimida con squashfs.	7	1	Programador
	Generar imagen ISO.	8	1	Programador
Alta	Obtener nombre y ruta del proyecto.	1	0.5	Analista
	Obtener parámetros de configuración de directorios.	2	0.5	Analista
	Obtener la ruta del sistema origen.	3	1	Analista
	Obtener valores de configuración del gestor de arranque.	4	0.5	Analista

Alta	Obtener ruta del Kernel y la imagen initrd a usar.	5	0.5	Analista
	Definir expresiones regulares para obtener el por ciento de copia de un fichero.	6	1	Analista
	Definir expresiones regulares para obtener el nombre del fichero que se está copiando.	7	1	Analista
	Crear expresión regular para obtener el por ciento de creación de la imagen.	8	1	Analista
	Obtener valores de configuración de la imagen ISO.	9	0.5	Analista
	Crear expresión regular para obtener el por ciento de creación de la imagen ISO.	10	1	Analista
	Mostrar cuales aplicaciones están instaladas o no.	11	0.5	Analista
	Mostrar la lista de los Kernels e Initrd disponibles en el sistema.	12	0.5	Analista
	Mostrar lista de carpetas a excluir.	13	0.5	Analista
	Mostrar lista de ficheros a excluir.	14	0.5	Analista
	Mostrar progreso de copia del sistema original.	15	0.5	Analista
	Mostrar progreso de creación de la imagen comprimida.	16	0.5	Analista

	Mostrar progreso de creación de la imagen ISO.	17	0.5	Analista
	Mostrar datos sobre la imagen ISO generada.	18	0.5	Analista
Baja	Chequear la herramienta rsync.	1	0.5	Analista
	Permitir seleccionar una imagen Splash personalizada por el usuario.	2	1	Programador
	Permitir seleccionar un Kernel personalizado por el usuario.	3	1	Analista
	Permitir seleccionar un Initrd personalizado por el usuario.	4	1	Analista
	Chequear la herramienta mkisofs.	5	0.5	Analista
	Chequear herramienta Squash.	6	0.5	Analista
	Calcula espacio de la imagen comprimida.	7	1	programador
Requisitos No Funcionales				
	NovaFAR-Live debe ser instalado en sistemas pertenecientes a NovaFAR/GNU LINUX.	1	0.5	Analista
	NovaFAR-Live debe ser una interfaz gráfica amigable, y agradable para el uso de los usuarios que interactúen con ella, la cual estará compuesta por un Front-end.	2	0.5	Analista

	NovaFAR-Live se caracterizará como un Software seguro, manteniendo siempre la integridad del sistema. Siempre estará disponible al acceso de los usuarios administradores en el sistema operativo NovaFAR para su uso.	3	0.5	Analista
	NovaFAR-Live se caracterizará por ser un Software de ambiente agradable, conteniendo una ayuda formalizada de la aplicación que crea expectativas de uso más flexibles.	4	0.5	Analista
	El sistema debe ser bien documentado de forma tal que en caso de mantenimiento el tiempo que se requiera sea el mínimo.	5	0.5	Analista
	El sistema permitirá su extensibilidad, permitiendo agregar nuevas funcionalidades en un futuro.	6	0.5	Analista

Anexo #2: Plantilla de Plan de Prueba para NovaFAR Live

I. HU a probar:

- ✓ Programar Scripts Auxiliares.
- ✓ Copia del sistema origen.
- ✓ Compactar Imagen.
- ✓ Crear imagen ISO.
- ✓ Adicionar Perfil

✓ Configuración.

II. Cronograma:

A continuación se muestra el cronograma de prueba para la herramienta NovaFAR Live:

No.	Tarea	Fecha	Responsable	Participantes	Observaciones
1	Programar script auxiliares.	3/3/2010	Carlos A. García.	Elisa Gamez, Keiver Hdez.	Prueba Satisfactoria
2	Copia del Sistema Origen.	18/3/2010	Carlos A. García.	Elisa Gamez, Keiver Hdez.	Prueba Satisfactoria
3	Compactar Imagen.	20/3/2010	Elisa Gamez.	Carlos A. García, Keiver Hdez.	Prueba Satisfactoria
4	Crear imagen ISO.	20/3/2010	Elisa Gamez.	Carlos A. García, Keiver Hdez.	Prueba Satisfactoria
5	Adicionar Perfil.	3/3/2010	Elisa Gamez.	Carlos A. García, Keiver Hdez.	Prueba Satisfactoria
6	Configuración	23/3/2010	Carlos A. García.	Elisa Gamez, Keiver Hdez.	Prueba Satisfactoria

III. Evaluación de las pruebas:

El proceso de prueba es clave a la hora de detectar errores. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Cuando se realiza una prueba se dan ciertos criterios de evaluación como son: Criterio de No Conformidad, Pedidos de Cambios y Lista de Chequeo.

A continuación se describen estos criterios:

No Conformidad: La no conformidad vendrá dada si se viola algún requisito funcional definido por el cliente, es decir, que lo que se le brinde no sea lo que el cliente solicitó.

Pedidos de Cambios: El pedido de cambio vendrá dado por cambios en requerimientos, funcionalidad, diseño o por errores y/o defectos.

Listas de Chequeos: Proporciona un apoyo mayor mediante preguntas que los probadores deben de responder mientras leen el artefacto. Esta técnica proporciona listas que ayudan al probador a saber qué tipo de faltas buscar.