

Universidad de las Ciencias Informáticas

Facultad 1

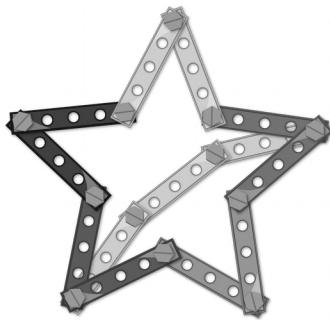


## TÍTULO

---

# PERSONALIZACIÓN DE GNU/LINUX NOVA PARA EL SISTEMA DE EDUCACIÓN PRIMARIA EN CUBA

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



### **Autor**

Manuel Alejandro Sánchez del Campo

### **Tutores**

MSc. Allan Pierra Fuentes

Ing. Dairelys García Rivas

**La Habana**

**2015**

**“Año 57 de la Revolución”**



*“La libertad no es poder elegir entre unas pocas opciones impuestas, sino tener el control de tu propia vida... Enseñar a los niños el uso de software libre en las escuelas, formará individuos con sentido de libertad.”*

*Richard Matthew Stallman*

## **Declaración de Autoría**

---

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a que haga el uso que estime pertinente con el mismo.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ de \_\_\_\_\_.

---

**Autor**

Manuel Alejandro Sánchez del Campo

---

**Tutor**

MsC. Allan Pierra Fuentes

---

**Tutor**

Ing. Dairelys García Rivas

## **Dedicatoria**

---

A mi madre por sus sacrificios, por ser la luz de mis ojos y la voz de mi conciencia.  
Por siempre estar ahí sin pedir nada a cambio.

A mi padre por alentarme a continuar creciendo en la vida y aconsejarme a cada momento.

A mi hermano por haber venido al mundo a hacerme compañía, brindarme su apoyo y hacer mas  
completa la vida.

## Agradecimientos

---

A mis padres y mi hermano, por el sacrificio de todos estos años, por ser siempre mi bastón de apoyo, por estar siempre ahí.

A la comunidad de software libre “humanOS” por formarme y ayudarme a crecer tanto en lo profesional como en lo espiritual.

A Jorge amigo y compañero de trastadas, el mago del código y alquimista de los Elefantes Azules.

A Kceres, Jako, Osiel, Alexis y Yaisel por su amistad y sus consejos.

A Pepe por hacer todos los diseños que le pedí, aún cuando se lo pedía de carrera.

A mis tutores por ayudar en todo el proceso y tener paciencia.

A los amigos que son muchos y que siempre estuvieron para darme una mano cuando lo necesité.

A los elefantes azules por meterme y sacarme en más de una ocasión de los más complicados embrollos,  
*¡Ellos siempre serán azules!*

A todos los que de una forma u otra han formado parte de mi vida y me han ayudado a ser quien soy.

Cuba se encuentra inmersa en la migración a software libres. Sin embargo los contenidos definidos en la dosificación informática no están en consonancia con este proceso. Como consecuencia el estudiante, una vez egresado no cuenta con los conocimientos necesarios sobre el uso de herramientas de software y tecnologías libres. Aun cuando se cuenta con una Distribución Cubana de GNU/Linux adaptada a la realidad nacional, su adopción dentro del sistema educativo no constituye una opción viable dado que esta fue concebida para su uso en entornos profesionales y carece de los software necesarios para apoyar el proceso de enseñanza aprendizaje. Partiendo de la situación problemática planteada con anterioridad, el presente trabajo tiene como objetivo desarrollar una personalización de la distribución de GNU/Linux Nova que facilite su incorporación en la dosificación informática para la enseñanza primaria en Cuba. En el documento, se recogen los resultados del estudio del estado del arte, las herramientas y técnicas empleadas, así como los diferentes elementos de la metodología de desarrollo Nova OpenUp. El resultado esperado constituye una personalización de la distribución de GNU/Linux Nova que facilite la integración de esta distribución en los centros de educación primaria del país.

**Palabras Claves:** migración, personalización, dosificación informática, educación primaria, GNU/Linux Nova.

# Índice de contenido

Introducción.....	1
Capítulo1. Fundamentación Teórica.....	7
1.1 Conceptos fundamentales.....	7
1.2 El software libre en la educación.....	11
1.2.1 Las TIC en la educación.....	12
1.2.2 El software libre en las escuelas.....	12
1.2.3 El software libre en la enseñanza de la informática.....	14
1.3 GNU/Linux en la educación.....	15
1.3.1 Distribuciones de GNU/Linux educativas.....	15
1.3.2 ¿Por qué Nova?.....	17
1.4 Tecnologías a emplear en el proceso de construcción de la personalización.....	19
1.4.1 Herramienta para la creación del Sistema Operativo Base.....	19
1.4.2 Herramienta para la gestión de proyecto.....	20
1.4.3 Herramienta de modelado.....	21
1.4.4 Herramientas para la creación del ISO.....	21
1.4.5 Herramienta para la personalización del Repositorio.....	22
1.5 Metodología de desarrollo de software.....	23
1.5.1 Metodología de desarrollo ágil.....	23
1.5.2 Metodología de desarrollo Nova OpenUp.....	24
1.6 Conclusiones del capítulo.....	25
Capítulo2. Planificación y Diseño de la Solución Propuesta.....	26
2.1 Estructura de componentes propuesta para el sistema.....	26
2.2 Gestión del proceso de desarrollo de la personalización de GNU/Linux Nova.....	27
2.3 Modelo del proceso de construcción de Nova Educativo.....	32
2.4 Requisitos de software.....	33
2.4.1 Requisitos funcionales.....	33
2.4.2 Descripción de los requisitos ágiles.....	33
2.4.3 Requisitos no funcionales.....	42

2.5 Conclusiones del capítulo.....	43
Capítulo3. Implementación y Pruebas.....	44
3.1 Proceso de construcción.....	44
3.1.1 Elementos necesarios para la construcción.....	44
3.1.2 Preparativos previos a la construcción del Sistema Operativo Base.....	45
3.1.3 Construcción del SOB.....	46
3.1.4 Configuraciones iniciales.....	46
3.1.5 Instalación de paquetes.....	47
3.1.6 Configuraciones al sistema.....	52
3.1.7 Restauración de las configuraciones del sistema.....	54
3.1.8 Empaquetado del sistema.....	55
3.1.9 Creación del repositorio personalizado.....	58
3.1.10 Creación del ISO.....	58
3.2 Pruebas al sistema.....	58
3.2.1 Nivel de prueba.....	59
3.2.2 Métodos de prueba.....	59
3.2.3 Diseño de casos de prueba.....	60
3.2.4 Resultados obtenidos.....	60
3.3 Conclusiones del capítulo.....	61
Conclusiones Generales.....	62
Recomendaciones.....	63
Referencias Bibliográficas.....	64
Bibliografía Consultada.....	67
Anexo 1: Dosificación de informática en la educación primaria y especial del municipio La Lisa.....	69
Anexo 2: Casos de Pruebas.....	77
Anexo 3: Guía de la entrevista realizada a profesores de la escuela primaria "Celia Sánchez".....	90



## Índice de tablas

Tabla 1: Comparativa entre distribuciones de GNU/Linux educativas.....	17
Tabla 2: Comparación entre una metodología ágil y una metodologías tradicionales.....	24
Tabla 3: Tareas para el desarrollo de la personalización de GNU/Linux Nova.....	28
Tabla 4: Tareas para el proceso de personalización de GNU/Linux Nova para el sistema de educación primaria en función de riesgos.....	31
Tabla 5: Descripción del RF1 Proveer software educativo que se ajuste a las necesidades del proceso docente educativo de la enseñanza primaria.....	34
Tabla 6: Descripción del RF2: Crear nuevos documentos con un procesador de textos, darle formato a los mismos y permitir guardar en diferentes formatos.....	35
Tabla 7: Descripción del RF3: Permitir al usuario la realización de acciones en hojas de cálculos en diferentes formatos.....	36
Tabla 8: Descripción del RF4: Crear y editar presentaciones en diferentes formatos.....	37
Tabla 9: Descripción del RF5: Permitir al usuario la facilidad de visualizar contenido web.....	38
Tabla 10: Descripción del RF6: Permitir la visualización de imágenes con opciones de tamaño y rotación. ....	39
Tabla 11: Descripción del RF7: Permitir la reproducción de música y vídeos en diferentes formatos.....	40
Tabla 12: Descripción del RF8: Permitir al usuario escoger si desea salir de la sesión: suspender, apagar o reiniciar el equipo.....	41
Tabla 13: Descripción del RF9: Permitir la instalación del sistema desde un CD, DVD o dispositivo USB..	42
Tabla 14: Paquetes básicos del sistema.....	48
Tabla 15: Paquetes educativos.....	51
Tabla 16: Otros paquetes complementarios.....	52
Tabla 17: CPR1: Proveer software educativo.....	77
Tabla 18: CPR2: Abrir y editar documentos de texto.....	79
Tabla 19: CPR3: Abrir y editar Hoja de cálculo.....	81
Tabla 20: CPR4: Abrir y editar presentaciones.....	83
Tabla 21: CPR5: Abrir ficheros en los formatos (htm,html) desde URL u otras direcciones físicas.....	84
Tabla 22: CPR6: Visualización, manipulación de imágenes soportadas.....	85

Tabla 23: CPR7: Reproducción vídeos en los formatos soportados.....	86
Tabla 24: CPR8: Reproducción audio en los formatos soportados.....	86
Tabla 25: CPR9: Salir de la sesión.....	87
Tabla 26: CPR10: Instalar el sistema.....	89

## Índice de ilustraciones

Figura 1: Diagrama de relación entre componentes del sistema.....	27
Figura 2: Diagrama del proceso de construcción de la personalización de GNU/Linux Nova para la educación primaria.....	32
Figura 3: Menú de aplicaciones educativas.....	34
Figura 4: LibreOffice Writer.....	35
Figura 5: LibreOffice Calc.....	36
Figura 6: LibreOffice Impress.....	37
Figura 7: Navegador web Mozilla Firefox.....	38
Figura 8: Visor de Imágenes.....	39
Figura 9: Reproductor de video VLC y reproductor de audio Audacious.....	40
Figura 10: Menú con opciones para abandonar la sección actual.....	41
Figura 11: Porcentaje de aceptación y errores en pruebas al sistema.....	61

## Introducción

El desarrollo alcanzado por las nuevas tecnologías de la informática y las comunicaciones ha permitido la evolución de otras esferas en el ámbito político, cultural, económico y social. Cuba en particular ha venido avanzando en este sentido a partir de la aprobación en 1997 de la estrategia de informatización de la sociedad cubana<sup>1</sup> (Vidal 2008) y la posterior inclusión de este aspecto como parte de los Lineamientos de la Política Económica y Social del Partido y la Revolución<sup>2</sup> aprobados en el sexto Congreso del Partido Comunista de Cuba (PCC) el 18 de abril de 2011.

Sin embargo, la difícil situación internacional que rodea a Cuba tanto en el ámbito político como económico, asediada por un férreo bloqueo económico y comercial impuesto por el gobierno de los Estados Unidos (EUA) y el dominio del mercado por grandes transnacionales del software<sup>3</sup> como Microsoft o Google, hacen imposible para la nación caribeña mantener el vertiginoso ritmo del modelo de consumo impuesto por el mercado o hacer uso de forma legal de las licencias requeridas por los productos desarrollados por compañías de EUA. A esto se suman los riesgos que representa para la seguridad y soberanía nacional el hecho de depender de software privativo desarrollado por empresas extranjeras cuyo código fuente es inaccesible y por tanto su funcionamiento real desconocido; de ahí que resulte indispensable lograr la soberanía tecnológica a través del uso en las instituciones y empresas del estado cubano de tecnología de software libre y código abierto.

En el año 2004 es presentado por el Consejo de Ministros el Acuerdo 084/2004, donde se indicaba al entonces Ministerio de la Informática y las Comunicaciones (MIC) ordenar el proceso paulatino de migración a software libre y de código abierto<sup>4</sup> a los Organismos de la Administración Central del Estado

---

<sup>1</sup> Proceso de utilización ordenada y masiva de las tecnologías de la información y las comunicaciones para satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad.

<sup>2</sup> Constituyen el documento rector que define que el sistema económico que prevalecerá continuará basándose en la propiedad socialista de todo el pueblo sobre los medios fundamentales de producción, donde deberá regir el principio de distribución socialista "de cada cual según su capacidad, a cada cual según su trabajo".

<sup>3</sup> Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. («Real Academia Española. Diccionario Usual.»)

<sup>4</sup> Un proceso ordenado en el cual se sustituye, parcial o totalmente, el software existente en la organización por alternativas liberadas bajo

(OACE). Desde entonces, y con el fin de dar cumplimiento a este objetivo, fue creado el Grupo Ejecutivo Nacional y cuatro subgrupos asociados, entre los que se encuentra el Grupo Técnico Nacional (GTN), encargado de desarrollar un sistema operativo cubano y definir los lineamientos y estrategias a seguir. Estas estrategias fueron presentadas en el año 2009 como parte de la Guía Cubana de Migración a Software Libre<sup>5</sup>, documento basado en las experiencias internacionales de empresas, entidades públicas y gubernamentales.

Sin embargo, aún con la aplicación de la Guía Cubana de Migración a Software Libre y la existencia de una distribución GNU/Linux cubana diseñada para adaptarse a las particularidades tecnológicas con que cuenta el país, los esfuerzos por conseguir un proceso de migración satisfactorio en las instituciones del estado cubano se han visto hasta cierto punto obstaculizados, encontrando en la mayoría de los casos un fuerte rechazo y resistencia al cambio por parte del personal de dichos centros.

Esta cuestión se asocia generalmente a que las personas encargadas de llevar a cabo la migración se centran fundamentalmente en aspectos tecnológicos y obvian el componente humano asociado a este (Sánchez Ortiz 2014) y por tanto, la necesidad de llevar a cabo la migración social en la que los seres humanos son el centro de la migración y por tanto requieren de un proceso de cambio en su forma de pensar, a fin de romper la cultura de fidelización a los programas privativos provocada por la costumbre y adaptándose a la filosofía asociada al movimiento de software libre y código abierto (Pérez Villazón 2009).

Pero incluso en las empresas y organismos donde se ha brindado un proceso de capacitación previo a la migración de cualquier sistema informático, mantener los resultados alcanzados a lo largo del tiempo resulta complicado, puesto que con la llegada de cada nuevo trabajador se debe comenzar una vez más el ciclo de entrenamiento y cambio de mentalidad.

Esto ocurre debido a que no se ha conseguido una buena sinergia entre el proceso de migración llevado a cabo en las instituciones y empresas del estado y los contenidos definidos en la dosificación informática<sup>6</sup> según la cual se rigen los centros educacionales para impartir dicha asignatura, donde se continúa

---

*licencias libres o de código abierto. Debe incluir la adopción de estándares abiertos para la documentación.*

*5 Define las etapas, procesos y actividades que se deben seguir para la migración de los Organismos de la Administración Central del Estado hacia Código Abierto.*

*6 Establece la dosificación de los contenidos de la asignatura de informática que se imparten en cada nivel de enseñanza.*

haciendo uso exclusivo de software y sistemas operativos privativos como Microsoft Windows y la suite ofimática Microsoft Office (ver Anexo 1: Dosificación de informática en la educación primaria y especial del municipio La Lisa.), resultando complejo mantener la estabilidad del proceso de adopción tecnológica dado que el estudiante, una vez egresado de su último año académico no cuenta con los conocimientos necesarios sobre el uso de herramientas de software libre y código abierto a las cuales deberá hacer frente por primera vez al ingresar en el centro donde desarrollará sus actividades laborales.

Aún cuando se cuenta con una Distribución Cubana de GNU/Linux adaptada a la realidad nacional, esta no está optimizada para su uso en entornos educativos, ya que carece por defecto de los software educativos necesarios para apoyar el proceso de enseñanza aprendizaje. De igual forma el diseño de la interfaz, según opina el cliente (ver Anexo 3: Guía de la entrevista realizada a profesores de la escuela primaria "Celia Sánchez".) no resulta del agrado de los estudiantes.

La inserción de las tecnologías de la informática y las comunicaciones en la enseñanza cubana, tiene como objetivo general elevar la calidad de la educación y garantizar la necesaria formación en el uso de las mismas de las nuevas generaciones, tal y como se plantea en el Programa de Informática Educativa del MINED.

De igual forma, no se debe olvidar la importante misión social que tienen las escuelas de inculcar los valores necesarios para crear ciudadanos de bien que puedan formar parte de una sociedad fuerte, capaz, solidaria y próspera. Pero cuando se hace uso de software privativo para impartir las asignaturas, el profesor aún sin saberlo se coloca en una encrucijada moral, ya que obliga al estudiante a copiar o hacer uso del software de manera ilegal, mostrando al estudiante que es posible violar la ley siempre que esta no le guste o vaya contra sus propios intereses. No se trata solo de sustituir un sistema operativo por otro porque sea más barato, seguro y fiable. Enseñar con GNU/Linux, no es solo usarlo sino transmitir el espíritu de colaboración y cooperativismo que implica cualquier empresa del conocimiento. El software libre es en sí mismo educativo por los valores que le acompañan.

De manera que no se trata de llevar a cabo una migración forzada en el sistema educativo cubano, sino de sentar las bases necesarias a partir de la creación de una herramienta que permita comenzar este

proceso desde las edades más tempranas de los estudiantes cubanos, evitando así el trauma futuro de un proceso de migración redundante en las instituciones y empresas del estado.

De los elementos anteriormente expuestos se origina el siguiente **problema de investigación** ¿Cómo facilitar el empleo de la Distribución Cubana de GNU/Linux Nova en la dosificación informática para la enseñanza primaria en Cuba?

Siendo el **objeto de estudio** el proceso de personalización de distribuciones de GNU/Linux para entornos educativos, teniendo como **campo de acción** la personalización de la Distribución Cubana de GNU/Linux Nova para la educación primaria en Cuba.

Trazando como **objetivo general** desarrollar una personalización de la Distribución Cubana de GNU/Linux Nova que facilite su incorporación en la dosificación informática para la enseñanza primaria en Cuba.

Para alcanzar este objetivo general se plantean los siguientes **objetivos específicos**:

- Elaborar el marco teórico conceptual relacionado con los aspectos teóricos que sustentan la investigación.
- Desarrollar una personalización de la Distribución Cubana de GNU/Linux Nova que permita dar solución al problema planteado.
- Validar la aplicación a partir de los diferentes casos de estudio.

Se propone como **idea a defender** que el desarrollo de una personalización de la Distribución Cubana de GNU/Linux Nova orientada al sistema de educación primaria en Cuba, permitirá facilitar el proceso de migración a software libre en la sociedad cubana.

Para dar cumplimiento al objetivo se definieron las siguientes **tareas de investigación**:

- Revisión de bibliografía asociada al proceso de construcción y personalización de distribuciones de GNU/Linux.

- Definición de las herramientas a utilizar para el desarrollo de la distribución.
- Selección de los software a utilizar en el repositorio de la distribución educativa de GNU/Linux.
- Diseño de los casos de prueba para la evaluación de las herramientas desarrolladas.
- Realización de la validación propuesta.
- Corrección de los posibles errores encontrados en la validación.

Para dar cumplimiento a las tareas de investigación fueron empleados los **métodos**:

Métodos teóricos:

**Histórico-Lógico**, para estudiar la evolución y desarrollo de diferentes soluciones GNU/Linux orientadas al ámbito educativo, analizando en cada caso las ventajas y errores presentados.

**Analítico-Sintético** para la descomposición del objeto de estudio en cada una de las tareas, para luego determinar el nexo existente entre ellas a fin de alcanzar un resultado.

Métodos Empíricos:

**Entrevista:** con el objetivo de recaudar información sobre el funcionamiento de las aplicaciones utilizadas hoy día en el proceso docente-educativo en el sistema de educación primaria en Cuba.

**Experimental:** a fin de estudiar el desempeño de la personalización bajo condiciones que permitan confirmar la idea a defender.

Quedando dividida la investigación en tres capítulos.

**Capítulo 1 - Fundamentación Teórica:** Se realizará un estudio del estado del arte de las soluciones ya existentes, analizando las características que puedan ser adaptadas a la propuesta de solución. Además se definirán una serie de conceptos necesarios para la comprensión del objeto de estudio.

**Capítulo 2 - Planificación y Diseño de la Solución Propuesta:** En este capítulo se hace alusión a la



solución propuesta, se definen las funcionalidades a implementar y se plantea cómo se va a desarrollar la misma a partir de la metodología de desarrollo de software seleccionada.

**Capítulo 3 - Implementación y Pruebas:** Se verá la implementación y pruebas de la personalización de la Distribución Cubana de GNU/Linux Nova. Se explican sus características principales, el diseño de casos de prueba y el análisis de los errores encontrados.

## **Capítulo 1. Fundamentación Teórica**

En aras de construir una personalización de la Distribución Cubana de GNU/Linux Nova que cumpla con las características necesarias para su funcionamiento óptimo en el sistema de educación primaria en Cuba, se hace necesario llevar a cabo una investigación sobre los principales conceptos y características presentes en otras distribuciones construidas con similar propósito, comparando estos aspectos en cada caso con las particularidades presentes en nuestro país. De igual forma en el capítulo se presenta la metodología a utilizar en el proceso de construcción de la personalización.

### **1.1 Conceptos fundamentales**

**Software libre:** La filosofía del Software Libre fue creada por Richard M. Stallman<sup>7</sup> en 1996 y en ella se estipulan los criterios que se tienen que cumplir para que un programa sea considerado libre.

Para entender el significado de esta filosofía es necesario primeramente pensar en la acepción de libre como en “libertad de expresión” y no como en “barra libre de cerveza” (Stallman 2004). En inglés en ocasiones se expresa “libre software”, en lugar de “free software”, para mostrar que no se trata de un producto gratuito.

El software libre es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

Un programa es software libre si los usuarios cuentan con las cuatro libertades esenciales (Free Software Foundation 2015):

- **Libertad 0:** la libertad para ejecutar el programa sea cual sea nuestro propósito.
- **Libertad 1:** la libertad para estudiar el funcionamiento del programa y adaptarlo a tus

---

<sup>7</sup> Programador estadounidense, fundador y figura principal del movimiento por el software libre en el mundo.

necesidades (el acceso al código fuente es condición indispensable para esto).

- **Libertad 2:** la libertad para redistribuir copias y ayudar así a tu vecino.
- **Libertad 3:** la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad (el acceso al código fuente es condición indispensable para esto).

**Software privativo:** El software privativo posee licencias que están limitadas por usuarios y son en su mayoría de pago. Estas restringen las libertades de los usuarios a usar, modificar, copiar y distribuir el software (Stallman 2004).

El desarrollo, programación y actualización de este software solo lo hace la empresa que tiene los derechos; ejemplo de esto son los productos de Microsoft como Windows y Office. Lo anterior puede ocasionar retrasos tecnológicos y poca creatividad en los productos (Stallman 2004).

En el software comercial se suelen esconder los avances y descubrimientos tecnológicos entre las empresas que lo desarrollan y el futuro del software que el usuario adquirió solo va a depender de la empresa que lo comercia (Stallman 2004).

En ocasiones suele suceder que con la utilización de estrategias comerciales los usuarios actualicen su software, sin que exista una verdadera necesidad de ello; para de esta forma hacer que el usuario invierta en nuevas licencias, la mayoría de las veces de forma innecesaria (Stallman 2004).

**Software educativo:** La expresión software educativo identifica genéricamente los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje. (Marqués 2012)

**Sistema operativo:** Según la Real Academia de la Lengua Española, un sistema operativo se define como un “programa o conjunto de programas que efectúan la gestión de los procesos básicos de un sistema informático, y permite la normal ejecución del resto de las operaciones.”

Los sistemas operativos tienen como funciones principales (Prieto 2002):

- Evitar que el usuario/aplicación tenga que conocer demasiados detalles de hardware<sup>8</sup> para su interacción.
- Ofrecer una interfaz de usuario o servicios que permitan ejecutar aplicaciones u operaciones.

**Sistema operativo base:** Un sistema operativo base (SOB), es la combinación mínima necesaria de aplicaciones que permite la instalación de paquetes en un sistema GNU/Linux, que pueden o no soportar el arranque de la computadora. (Beekmans 2011).

**Distribución de GNU/Linux:** Una distribución de GNU/Linux puede ser descrita como una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios. Las mismas pueden contener o no aplicaciones o controladores privativos (Pierra 2011).

En general, las distribuciones Linux pueden ser:

- Comerciales o no comerciales.
- Completamente libres o incluir software privativo.
- Diseñadas para uso en el hogar o en las empresas.
- Diseñadas para servidores, escritorios o dispositivos empujados.
- Orientadas a usuarios regulares o usuarios avanzados.
- De uso general o para dispositivos altamente especializados, como un cortafuegos, un enrutador o un cluster computacional.
- Diseñadas e incluso certificadas para un hardware o arquitectura específicos.
- Orientadas hacia grupos en específico, por ejemplo a través de la internacionalización y localización del lenguaje, o por la inclusión de varios paquetes para la producción musical o para

---

<sup>8</sup> Conjunto de los componentes que integran la parte material de una computadora. («Real Academia Española. Diccionario Usual.»)

computación científica.

- Configuradas especialmente para ser más seguras, completas, portables o fáciles de usar.
- Soportadas bajo distintos tipos de hardware.

**Personalización de sistema operativo:** A partir de la definición brindada por la Real Academia de la Lengua Española a la palabra personalizar como: “Dar carácter personal a algo”, es posible establecer como personalización de un sistema operativo, el conjunto de acciones realizadas sobre el conjunto de aplicaciones que forman al mismo, con el objetivo de agregar características al sistema y adaptarlo a las necesidades de sus usuarios finales (Miranda y Bayón 2014).

**Paquete de software:** Un paquete de software es una serie de objetos ejecutables y archivos de configuración que se distribuyen conjuntamente. Estos paquetes pueden ser binarios o de código fuente (Alegsa 2012).

**Paquetes binarios:** Los paquetes binarios contienen código máquina entendible por un sistema operativo y compilado en dependencia de la arquitectura del microprocesador donde vayan a ser utilizados. Algunos ejemplos de formatos de estos paquetes son (Miranda y Bayón 2014):

- **dpkg:** Usado originalmente por Debian y derivados. Fue el primero en poseer una herramienta de resolución de dependencias ampliamente conocida, APT.
- **RPM:** Creado por Red Hat y usado por un gran número de distribuciones de GNU/Linux, es el formato de paquetes del Linux Standard Base. Para trabajar con este sistema de paquetes existen diversas herramientas como apt4rpm, up2date (de Red Hat), urpmi (de Mandriva), YaST (de SuSE) y YUM (usado por Fedora y Yellow Dog Linux).
- **Pacman:** para Arch Linux usa binarios precompilados distribuidos en un fichero *pkg.tar.xz*.

**Paquetes de código fuente:** Contienen el código fuente del programa, vienen con los archivos necesarios para compilar e instalar el programa manualmente. Suelen presentarse en formato .tar.gz o tar.bz2 o sea, compactado con tar y comprimido con gzip o bzip (Miranda y Bayón 2014).

## 1.2 El software libre en la educación

Las razones vertidas por los partidarios que defienden las ventajas y superioridad del software libre y de código abierto frente al software privativo pueden ser divididas en dos grupos correspondientes a las dos principales corrientes de pensamiento existentes sobre el tema. Por un lado se encuentra la opinión brindada por la *Open Source Initiative* (OSI), representada por personalidades como Eric Raymond<sup>9</sup>, Tim O'Reilly<sup>10</sup>, Bruce Perens<sup>11</sup> o Linus Torvalds<sup>12</sup> en la que se defiende el código abierto por razones pragmáticas, como su mayor calidad, su menor coste, mayor seguridad, estabilidad, eficiencia e integración. Mientras que por otro lado la *Free Software Foundation* (FSF) encabezada por Richard M. Stallman defiende su superioridad ética, social y política. De cierta forma ambos conjuntos de argumentos son reales y complementarios.

Algunos autores, conforme se han ido extendiendo las ideas sobre el software libre, han apuntado también argumentos que tienen en cuenta el contexto escolar, es decir, por qué es más adecuado o ventajoso usar software libre en educación, además de las razones genéricas que se aplican a cualquier usuario informático. Entre ellos merecen mención aparte los relacionados con la enseñanza de la informática que tiene en el software libre un fenomenal recurso didáctico (Adell y Bernabé 2007).

### 1.2.1 Las TIC en la educación

La presencia de las nuevas tecnologías de la Información y la comunicación (TIC) en la educación ya no es una novedad casual, sino una realidad cotidiana. Los contextos de enseñanza-aprendizaje han cambiado con su sola aparición en las escuelas. Tanto es así que la Organización de las Naciones Unidas (ONU) recoge entre los *Objetivos de Desarrollo del Milenio* la necesidad del acceso universal a los beneficios de las nuevas tecnologías y de igual forma la UNESCO en su iniciativa *Educación para Todos* da especial importancia a la integración de las tecnologías de la informática en el proceso docente

---

9 *Escritor, historiador de la cultura hacker y una de las figuras líder en el Movimiento del Open Source y el Código abierto*

10 *Fundador y presidente de O'Reilly Media, uno de los autores del concepto Web 2.0 y participante en el desarrollo del lenguaje Perl. Fuerte impulsor de los movimientos de software libre y código abierto.*

11 *Autor de la definición del concepto de "código abierto", fundador y primer líder del proyecto Linux Standard Base, cofundador de la OSI Open Source Initiative.*

12 *Una de las figuras más importantes del movimiento Open Source y el Código abierto. Creador del núcleo Linux y actualmente coordinación del proyecto.*

(Stienen 2007).

### **1.2.2 El software libre en las escuelas**

En el artículo “Por qué las escuelas deben usar exclusivamente software libre”, (Stallman 2003) remarca el hecho de que los usuarios que hacen uso de tecnologías libres son capaces de controlar lo que hacen sus ordenadores y desarrollan la capacidad de cooperar entre ellos. Ambas razones son también válidas para la educación, pero hay razones netamente educativas (Adell y Bernabé 2007).

1. La primera es que el software libre se puede copiar y redistribuir a precio de coste. La administración educativa puede dotar de software a todos sus centros docentes a muy bajo precio y dedicar los recursos ahorrados a otros temas necesarios para la educación: más ordenadores, formación del profesorado y desarrollo de software libre educativo. En los países menos desarrollados, el software libre puede ayudar a dotar de infraestructura tecnológica a sus escuelas y a paliar la brecha digital con el mundo desarrollado. Los vendedores de software privativo, que saben de la importancia de la educación para sus futuras ventas, pueden ofrecer software a muy bajo coste o gratuito a las escuelas. Pero se trata en realidad de una estrategia comercial para captar futuros clientes y para formarlo en sus productos a costa del erario público.
2. La escuela ha de enseñar a los estudiantes valores y estilos de vida que beneficien a toda la sociedad. La escuela ha de promover el uso de software libre por la misma razón que promueve el reciclaje: porque nos beneficia a todos. Si los estudiantes usan el software libre y aprenden que es mejor que el privativo, cuando sean adultos seguirán usando el software libre. Eso permitirá a la sociedad liberarse de los abusos y del control de las multinacionales que controlan el software privativo.
3. El software libre favorece que los estudiantes aprendan cómo funcionan los ordenadores y el propio software. Los futuros programadores se inician en la programación durante la adolescencia. Es una etapa clave en la que necesitan buenos modelos y ejemplos para modificar, copiar y experimentar con ellos. Necesitan desafíos. El software libre, al permitir el acceso al código fuente del programa, les facilita el aprendizaje. El software privativo es como una caja negra que no aporta nada para satisfacer su curiosidad y sus ansias de saber. El mensaje que les envía el

software privativo es “el conocimiento es una mercancía, lo que quieres saber es un secreto comercial, aprender está prohibido por la ley”. El software privativo mantiene a la gente alejada del conocimiento, sacraliza la tecnología y contribuye interesadamente a la ignorancia tecnológica que tan buenos resultados económicos les proporciona a las empresas que lo comercializan.

4. Pero, aunque muchos adolescentes no sientan curiosidad por cómo están hechos los programas de ordenador, hay valores generales que persigue la educación que están en claro conflicto con el mensaje que transmite el software privativo. Las escuelas deben enseñar hechos, conceptos, principios y procedimientos, pero también valores. La misión de la escuela es enseñar a las personas a ser buenos ciudadanos, a cooperar con los demás, a ser solidarios. Ésta es la base de la sociedad. En informática, cooperar significa, entre otras cosas, compartir software, poder hacer copias a todos los compañeros de clase, llevarse a casa el software que se usa en la escuela. Y todo eso, con el software privativo es un delito.
5. Finalmente, enseñar a los estudiantes a usar software libre y a participar en la comunidad de usuarios y desarrolladores de software libre es una lección cívica llevada a la práctica. También enseña a los estudiantes que el ideal es el modelo de servicio público y la solidaridad, no el modelo del beneficio a cualquier precio de los magnates. Todos los niveles pueden y deben usar software libre (Stallman 2003).

(Amatrian 2006) resume perfectamente la coincidencia en valores del software libre y la educación: “los valores que una institución educativa tendría que promover están muy relacionados con aquellos que promueve el software libre: libertad de pensamiento y expresión, igualdad de oportunidades, esfuerzo y beneficio colectivo en lugar del beneficio individual. De hecho, la libertad puede que sea el valor más importante relacionado con la educación: la educación sin libertad se convierte en adoctrinamiento”.

### **1.2.3 El software libre en la enseñanza de la informática**

En su infancia los niños poseen un alto nivel de absorción del conocimiento, permitiéndoles aprender y comprender cosas en muy poco tiempo, que años más tarde durante su vida adulta tal vez les costaría



meses o quizás años de práctica. Es por ello que resulta de vital importancia comenzar a temprana edad a inculcarles las habilidades necesarias para el manejo de las nuevas tecnologías, asegurando de esta forma la construcción de una sociedad con personas plenamente capacitadas para innovar y sacar provecho del uso de las nuevas tecnologías (Pfaffenberger 2000).

El uso de software libre en este contexto resulta una estrategia primordial, ya que permitirá al estudiante conocer como funciona el sistema, liberándolo de la abstracción que impone el software privativo, donde el usuario desconoce por completo el funcionamiento del mismo, lo cual hace más engorroso y lento el proceso de enseñanza aprendizaje en el educando (Pfaffenberger 2000).

La alfabetización informática suele estar centrada en productos comerciales y adopta el enfoque “Qué tecla hay que apretar”. Es decir, está basada en procedimientos y destrezas concretos y de corto alcance. La justificación de este enfoque no es pedagógica, sino pragmática y errónea. Se afirma que el entorno de sistemas y aplicaciones privativas dominantes será el que encontrarán los estudiantes una vez concluido sus estudios. Es lo que los empresarios demandan y las grandes compañías demandan, por tanto, es lo que hay que enseñarles. Sin embargo, el enfoque “Qué tecla hay que apretar” olvida el rápido desarrollo de las tecnologías de la información: posiblemente la versión que dominan a la perfección los estudiantes ya no exista cuando busquen empleo. Lo que los empresarios necesitan no es alguien que domine versiones viejas del software, sino alguien capaz de aprender de manera rápida cualquier aplicación informática, alguien que posea también conocimientos y competencias generales (Pfaffenberger 2000).

El software libre, según (Pfaffenberger 2000), por su apertura y flexibilidad, facilitaría la formación basada en competencias genéricas, transferibles a otras situaciones y entornos, y el desarrollo de la capacidad de seguir aprendiendo por su cuenta a lo largo de toda la vida de los estudiantes.

### **1.3 GNU/Linux en la educación**

A lo largo de los años, varios países y proyectos han intentado ir cavando una brecha que permita la inserción del software libre en las instituciones educativas. Con este fin se han creado algunas distribuciones de GNU/Linux educativas, concepto que se pudiera definir como distribución de software

basada en el núcleo Linux y construida a partir de software orientados a cubrir las necesidades de estudiantes y docentes, sean estos privativos o no.

### 1.3.1 Distribuciones de GNU/Linux educativas

**Skolelinux/Debian Edu:** Es un proyecto creado en Noruega por un conjunto de docentes, activistas y entusiastas del Software Libre con el objetivo de proveer una solución completa de software para la educación basado en Software Libre.

Skolelinux es una combinación de la palabra Noruega “*skole*” que significa escuela y Linux, el kernel del sistema operativo libre GNU/Linux. Esta distribución de GNU/Linux viene adaptada a las necesidades de escenarios educativos y preconfigurada para su fácil instalación (Mora 2011).

**Fedora Education Spin:** Es la edición de Fedora para la educación tiene como objetivo principal facilitar a docentes y estudiantes el acceso a una gran variedad de herramientas para la educación tales como el galardonado ambiente de escritorio *Sugar*<sup>13</sup> y las populares aplicaciones educativas de KDE<sup>14</sup>.

Este sistema operativo también viene equipado con el entorno de desarrollo integrado Eclipse y otras herramientas que brindan la oportunidad a sus usuarios tanto de explorar las aplicaciones como de contribuir con la producción de código para las mismas.

La última versión liberada de esta distribución estuvo basada en *Fedora 12 Constantine* actualmente sin soporte (Mora 2011).

**Qimo for kids:** Es una distribución de GNU/Linux basada en Xubuntu con un escritorio diseñado exclusivamente para los más chicos. Viene equipado con juegos educativos para niños mayores de tres años. Su interfaz gráfica es sencilla e intuitiva y está diseñado con iconos grandes y llamativos.

La última actualización del sistema ocurrió en el año 2012, estaba basada en la versión 10.10 de

---

13 *Sugar es un ambiente de escritorio diseñado originalmente para ser utilizado en las computadoras del proyecto “One Laptop per Children” (OLPC). Su original diseño orientado a niños y su facilidad de uso lo han hecho muy popular y posee una comunidad de voluntarios creciente, que incluye: educadores, desarrolladores, estudiante y otros.*

14 *KDE es un entorno de escritorio desarrollado en Qt muy utilizado en distribuciones de GNU/Linux.*

*Xubuntu*, distribución sin soporte en la actualidad (Mora 2011).

**Edubuntu:** Está orientado hacia la educación y contiene una gran cantidad de programas en distintas categorías para niños y jóvenes. Está basado en la popular distribución de GNU/Linux Ubuntu, cuyo eslogan es “*Linux for Human Beings*” (Linux para seres humanos) y su nombre proviene del concepto Ubuntu que quiere decir «humanidad hacia otros».

Edubuntu es también un movimiento liderado por estudiantes, docentes, padres de familia y hackers que creen que el conocimiento debe estar disponible libremente para todos y buscan promover la adopción de Ubuntu en las escuelas, hogares y comunidades, además de brindar apoyo a los usuarios para su instalación y mantenimiento (Mora 2011).

A continuación se muestra una tabla comparativa referente a varios aspectos importantes de las distribuciones mencionadas con anterioridad (Tabla 1: Comparativa entre distribuciones de GNU/Linux educativas.):

Nombre	Entorno de escritorio	Nivel educacional al que está dirigida	Última versión estable
<b>Skolelinux</b>	KDE, GNOME, LXDE, XFCE	Primaria, Secundaria, Superior	2013 - Debian Wheezy
<b>Fedora Education Spin</b>	GNOME	Primaria, Secundaria, Superior	2009 - Fedora 12 “Constantine”
<b>Qimo for kids</b>	XFCE	Primaria, Secundaria	2012 - Versión 10.10 de Xubuntu
<b>Edubuntu</b>	GNOME	Primaria, Secundaria, Superior	2014 - Ubuntu 14.10

Tabla 1: Comparativa entre distribuciones de GNU/Linux educativas.

### 1.3.2 ¿Por qué Nova?

Nova es una distribución de GNU/Linux desarrollada y mantenida en la Universidad de las Ciencias

informáticas (UCI) por un grupo de programadores, ingenieros, másteres y miembros de la comunidad de software libre de Cuba. Esta distribución es construida en base a los principios fundamentales para el desarrollo de aplicaciones informáticas del país reflejados en las *4S de Pierra* que se exponen a continuación (Pierra 2011):

**Seguridad:** El modelo de desarrollo colaborativo que nos propone el movimiento de software libre, el acceso al código fuente y el exhaustivo proceso de revisión y auditoría de código garantizará un sistema seguro de ataques y sin puertas traseras.

**Soberanía Tecnológica:** Es la capacidad de nuestro país para desarrollarse en dicho campo en forma autónoma. No supone autarquía (independencia absoluta) sino capacidad decisional sobre su uso y desarrollo.

**Socio-adaptabilidad:** Las bases tecnológicas para la informatización de Cuba, deben ser hechas por cubanos y para los cubanos, logrando inigualable adaptabilidad a las condiciones de nuestro país.

**Sostenibilidad:** La constante asimilación e investigación de las nuevas tecnologías, la planificación, los modelos novedosos de comercialización y el uso racional de los recursos humanos, materiales y naturales, garantizarán a nuestras soluciones vigencia y sostenibilidad a largo plazo.

Estos principios no contradicen la filosofía de código abierto, ni sustituyen las cuatro libertades del movimiento de software libre, sino que, sumados a estas pretenden materializar las razones que llevan a nuestro país a ejecutar el necesario proceso de migración tecnológica.

El hecho de que el desarrollo de la distribución sea llevado a cabo en nuestro país, permite brindar soporte de primera mano y de calidad, con un tiempo de respuesta corto a los centros educativos donde sea desplegada la personalización.

Así mismo, GNU/Linux Nova cuenta con una versión ligera, cuyo entorno de escritorio Guano ha sido optimizado para su uso en ordenadores de bajas prestaciones, como los existentes en una gran parte de las escuelas primarias cubanas y su interfaz mejorada, siguiendo estilos de diseños de otros sistemas operativos con los que los estudiantes se encuentran más familiarizados.

Si bien es cierto que las distribuciones anteriormente mencionadas cuentan con características que las hacen aptas para ser usadas en entornos educativos, presentan algunos inconvenientes que no las hacen óptimas como propuesta de solución. En el caso de dos de ellas, *Fedora Education Spin* y *Qimo for kids*, se encuentran descontinuadas y sin recibir soporte desde hace más de tres años. *Edubuntu* cuenta con un entorno de escritorio que exige de un rendimiento mayor al existente en varios de los centros educativos del país. Y en general estas distribuciones fueron construidas para ser usadas en varios niveles de enseñanza, por lo que cuentan en su catálogo de software con un número importante de aplicaciones que no son aptas para su uso en el sistema de educación primaria.

## **1.4 Tecnologías a emplear en el proceso de construcción de la personalización.**

A continuación se describen las herramientas a emplear en el proceso de construcción de la personalización de la Distribución Cubana de GNU/Linux Nova para su uso en el sistema de educación primaria en Cuba.

### **1.4.1 Herramienta para la creación del Sistema Operativo Base.**

La construcción de un Sistema Operativo Base (SOB) resulta el primer paso antes de llevar a cabo el desarrollo de la personalización de la Distribución Cubana de GNU/Linux Nova. Ya que será sobre este SOB que se realicen las configuraciones necesarias. Para su creación se decidió hacer uso de la herramienta Debootstrap dado que esta permite instalar un sistema basado en Debian dentro de un subdirectorio de otro sistema ya instalado sin necesidad de un CD de instalación, solamente el acceso a un repositorio.

Para hacer uso de la herramienta resulta necesario abrir un terminal de líneas de comando y escribir la sintaxis siguiente:

```
debootstrap [OPCIONES...] $DISTRO $MONTAJE $MIRROR
```

donde:

- **\$DISTRO** - Identifica la versión de la distribución de GNU/Linux (Debian, Ubuntu, Nova, etc) que se

desea instalar.

- **\$MONTAJE** - Hace referencia al punto de montaje o subdirectorio donde se quiere instalar la distribución.
- **\$MIRROR** - Se refiere al repositorio o espejo (mirror en inglés), de donde se van a descargar los paquetes para la instalación. (Esta herramienta solo permite usar un repositorio para sus paquetes. Si se necesita obtener paquetes desde diferentes repositorios, es recomendable hacer uso de Multistrap).

Otras opciones de debootstrap:

- **--arch=ARCH** - Selecciona la arquitectura que poseerá el nuevo sistema, se sustituye normalmente ARCH por i386 o amd64.
- **--include=alpha,beta** - Lista de paquetes separados por coma que se van a agregar a la descarga.
- **--exclude=alpha,beta** - Lista de paquetes separados por coma que serán retirados de la descarga. Hay que tener mucho cuidado con esta opción, ya que se podrían excluir paquetes esenciales.
- **--components=alpha,beta** - Utilizar paquetes de los componentes listados.
- **--variant=minbase | buildd | fakechroot | scratchbox** - Nombre de la variante a utilizar.

Actualmente, las variantes admitidas son:

- **minbase** - solo incluye paquetes esenciales y apt;
- **buildd** - Instala el build-essential<sup>15</sup> en \$MONTAJE.
- **fakechroot** - Instala los paquetes sin privilegios de root.
- **scratchbox**<sup>16</sup> - Es utilizado en la creación de destinos para usar *scratchbox*. De forma predeterminada, sin un argumento --variante = X, se crea una instalación de Debian base en

---

<sup>15</sup> Es un paquete que contiene herramientas necesarias para la creación, compilación e instalación de programas.

<sup>16</sup> Es un conjunto de herramientas de compilación cruzada diseñado para hacer que el desarrollo de aplicaciones Linux embebido sea más fácil. También proporciona un conjunto completo de herramientas para integrar y realizar una compilación cruzada de una distribución Linux completa.

## \$MONTAJE.

- **--verbose** - Brinda más información acerca de la descarga.

### 1.4.2 Herramienta para la gestión de proyecto

Para la gestión de proyecto se hará uso de la herramienta ProjectLibre, software similar a Microsoft Project, con una interfaz de usuario y una metodología similar para la construcción de un plan de proyecto: se crea una lista de tareas indentadas o una estructura de descomposición del trabajo (WBS), se establecen duraciones, se crean enlaces y se asignan recursos. Las columnas (campos) son las mismas en ambos productos.

La versión actual incluye:

- Interfaz de usuario basado en cintas (*Ribbon UI*).
- Gestión del Valor Ganado.
- Diagrama de Gantt.
- Gráficos PERT.
- Diagrama de estructura analítica de recursos (RBS).
- Diagrama de Estructura de descomposición del trabajo (WBS).
- Informes de uso de tareas

### 1.4.3 Herramienta de modelado

Para el modelado fue escogido Visual Paradigm, una herramienta de Ingeniería de Software Asistida por Computadora (CASE) definida por la universidad para el desarrollo de aplicaciones utilizando el lenguaje de modelado UML<sup>17</sup>. Fue diseñado para una amplia gama de usuarios interesados en la construcción de

---

<sup>17</sup> *Lenguaje de Modelado Unificado.*

sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. Es multiplataforma y cuenta con una versión llamada *Community Edition* que se caracteriza por ser de uso no comercial (Sierra 2007).

#### **1.4.4 Herramientas para la creación del ISO**

Una vez concluido el proceso de personalización de la distribución de GNU/Linux, será el momento de pasar a la creación del ISO que la contendrá. Para esto se decidió emplear la herramienta Genisoimage, la cual es un *fork* de mkisofs, una aplicación utilizada para crear imágenes de sistemas de archivos ISO-9660 para CD-ROM, que pueden grabarse en CD o DVD usando un programa específico para quemado o en dispositivos USB por medio de aplicaciones que permitan grabar la imagen *bootable* (Debian Admin 2014).

#### **1.4.5 Herramienta para la personalización del Repositorio**

Para la construcción del repositorio personalizado que sería incorporado dentro del ISO, se decidió hacer uso de la aplicación comunitaria Tinp, dado que otras herramientas como Repoman se encuentran actualmente sin soporte. Esta es una herramienta de línea de comandos para crear repositorios personalizados de Debian y similares (Ubuntu, Mint, Nova), creando dicho repositorio con los paquetes seleccionados y todo su árbol de dependencias.

Entre las principales características de la herramienta están:

- Es portable.
- Multiplataforma.
- Permite crear los repositorios personalizados a partir de repositorios en la web o guardados localmente.
- Hecha en python.



- Compatible con python 2.7 y 3.x.

El funcionamiento básico de la herramienta sería:

```
python tinp-cli.py -a "scite apache2" /home/miusuario/mirepo
```

Esto crea un repositorio con los paquetes scite y apache2 junto a sus árboles de dependencias en la ruta `/home/miusuario/mirepo` a partir de los repositorios configurados en el sistema. Otras opciones de interés serían:

- `-s (--sources)` para señalar un fichero de fuentes (`sources.list`) predefinido (por defecto lo busca en `/etc/apt/sources.list`).
- `-c (--arch)` para definir la arquitectura (por defecto `"binary-i386"`).

Para una lista completa de las opciones se puede correr con `-h` o `--help`.

## 1.5 Metodología de desarrollo de software

Una metodología es el conjunto de pasos o procedimientos a seguir para llevar a cabo un proyecto de cualquier índole o disciplina, tiene determinado objetivo, espera cierto resultado y brinda solución a un problema (Monte 2013).

### 1.5.1 Metodología de desarrollo ágil

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software, se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil por la habilidad de responder de forma versátil al cambio para maximizar los beneficios, de forma que una metodología ágil es la que tiene como principios (Monte 2013):

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- El software que funciona es más importante que la documentación exhaustiva.

- La colaboración con el cliente en lugar de la negociación de contratos.
- La respuesta delante del cambio en lugar de seguir un plan cerrado.

En la tabla siguiente se puede observar una comparación entre una metodología tradicional y una ágil. (Tabla 2: Comparación entre una metodología ágil y una metodologías tradicionales.)

Criterio	Metodologías Ágiles	Metodologías Tradicionales
Roles	Poseen pocos roles.	Poseen gran cantidad de roles.
Artefactos	Poseen pocos artefactos.	Poseen gran cantidad de artefactos.
Arquitectura	Se define y mejora a lo largo del proyecto.	Se define al principio del proyecto.
Aplicación	Deben aplicarse a proyectos pequeños de corta duración, con no más de 10 integrantes y que residan en el mismo sitio.	Puede aplicarse a proyectos de cualquier tamaño, pero son más eficientes en proyectos grandes y equipos dispersos.
Contrato	No tiene que existir.	Sí existe un contrato prefijado.
Interacción con el Cliente	Constante, es parte del equipo de desarrolladores.	solo en reuniones oficiales.

Tabla 2: Comparación entre una metodología ágil y una metodologías tradicionales.

### 1.5.2 Metodología de desarrollo Nova OpenUp

Nova OpenUp es una metodología de desarrollo de software ágil, creada con el propósito de mejorar el proceso de construcción de la Distribución Cubana de GNU/Linux Nova y por tanto aplicable al resto de distribuciones de GNU/Linux. La metodología abarca todo el ciclo de vida de proyectos que hacen productos de este tipo mediante la ejecución de un conjunto de disciplinas compuestas por actividades que se relacionan entre sí, roles y artefactos (Monte 2013).

Esta metodología se desarrolla en 4 fases:

- **Inicio:** En esta fase se debe entender el propósito y objetivos a cumplir, conocer los requisitos que debe cumplir el producto, y realizar un análisis para determinar costos y riesgos asociados al proyecto.
- **Elaboración:** Los objetivos fundamentales de esta fase son obtener los requisitos de manera más detallada, diseñar, validar e implementar la arquitectura de las distribuciones GNU/Linux así como mitigar los riesgos necesarios.
- **Construcción:** Esta fase logra dar al producto capacidad operacional cumpliendo con los objetivos, su finalidad es completar el desarrollo del sistema, para ello se debe crear una versión del producto que cumpla con los requisitos especificados y listo para usarse por los usuarios, así como optimizar los recursos necesarios y reutilizar componentes para minimizar los costos de desarrollo.
- **Transición:** En esta fase es donde se obtiene una liberación candidata del sistema, la cual se debe validar para saber si cumple con las expectativas y desplegar en los entornos de los usuarios finales.

## 1.6 Conclusiones del capítulo

Al realizar el análisis de los conceptos relacionados con la investigación, se lograron comprender los elementos teóricos que respaldan el presente trabajo. Como parte del estudio realizado se obtuvo como resultado que las distribuciones educativas actualmente disponibles no ofrecen una solución que se adapte al sistema educativo cubano, por lo que se decidió desarrollar la personalización a partir de la versión base de la Distribución Cubana de GNU/Linux Nova, usando como herramienta Debootstrap. Para la construcción del ISO final se definió el uso de la aplicación Genisoimage. Como metodología de trabajo a utilizar para la personalización quedó establecida Nova OpenUp.

## **Capítulo2. Planificación y Diseño de la Solución Propuesta**

En el presente capítulo se procede a realizar la descripción de la propuesta de solución a fin de facilitar su comprensión y obtener un producto con la calidad estipulada para su uso en el proceso de enseñanza y aprendizaje del sistema de educación primaria en Cuba.

### **2.1 Estructura de componentes propuesta para el sistema**

Con la finalidad de dar solución al problema planteado y con ello brindar un impulso al proceso de migración que se viene llevando a cabo en el país, se decide construir una personalización de GNU/Linux para el sistema de educación primaria en Cuba, tomando como base la Distribución Cubana de GNU/Linux Nova, dado que esta se adapta a la realidad y particularidades que enfrenta el proceso de migración de la sociedad cubana. De igual forma, Nova cuenta entre sus repositorios con una gran variedad de paquetes de aplicaciones educativas que facilitarán y enriquecerán el proceso de enseñanza y aprendizaje de los estudiantes.

Para una mayor comprensión de la propuesta de solución por parte del cliente se hace necesario la creación de un diagrama donde se muestra las partes que componen el sistema y la relación existente entre las mismas (Figura 1).

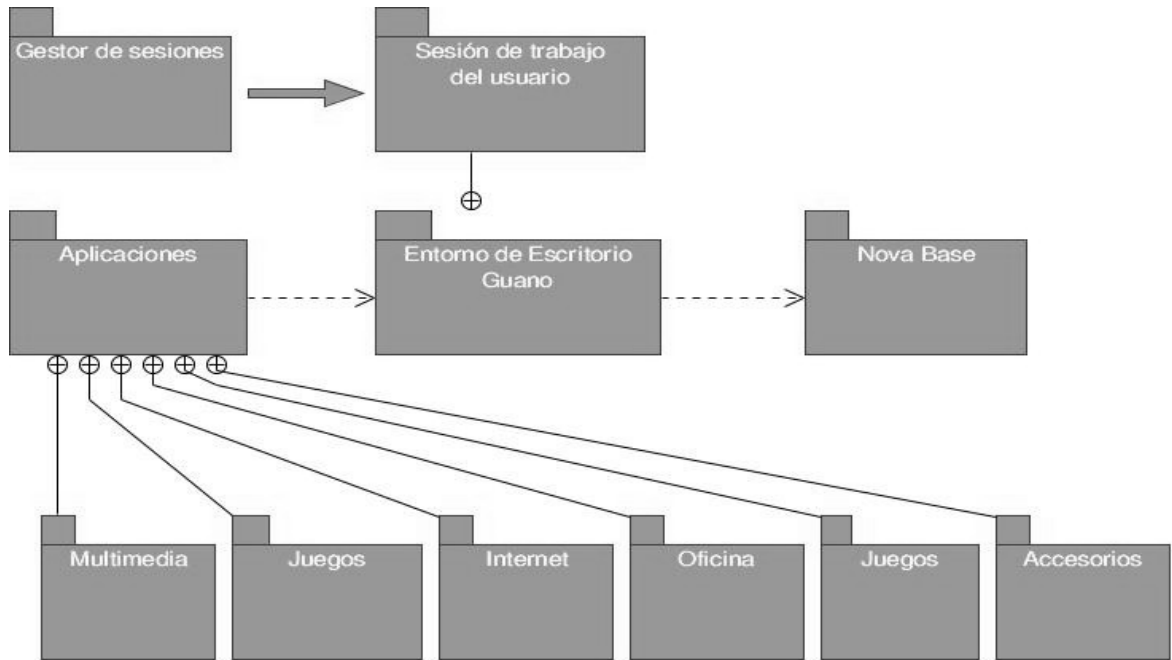


Figura 1: Diagrama de relación entre componentes del sistema.

## 2.2 Gestión del proceso de desarrollo de la personalización de GNU/Linux Nova

El proceso de construcción de la personalización de GNU/Linux Nova para el sistema de educación primaria en Cuba, se estructura para abarcar un período de duración de 150 días, estableciendo un total de 17 tareas que fueron distribuidas en la totalidad del ciclo de desarrollo, quedando conformado como se observa en la Tabla 3: Tareas para el desarrollo de la personalización de GNU/Linux Nova.

Nombre	Duracion	Inicio	Terminado
Proceso de desarrollo de Nova Educativo	150 days?	3/11/14 8:00	29/05/15 17:00
Recopilación de información	130 days?	3/11/14 8:00	1/05/15 17:00
Identificación de requisitos	10 days?	3/11/14 8:00	14/11/14 17:00
Especificación y verificación de requisitos	10 days?	10/11/14 8:00	21/11/14 17:00
Descripción de requisitos ágiles	10 days?	17/11/14 8:00	28/11/14 17:00
Definición de vista de sistema	10 days?	28/11/14 8:00	11/12/14 17:00
Definición de vista de infraestructura	10 days?	5/12/14 8:00	18/12/14 17:00
Definición de vista de seguridad	10 days?	12/12/14 8:00	25/12/14 17:00
Definición de vista de despliegue	10 days?	26/12/14 8:00	8/01/15 17:00
Generación de repositorios binarios	5 days?	5/01/15 8:00	9/01/15 17:00
Creación del sistema base	15 days?	12/01/15 8:00	30/01/15 17:00
Modificación de paquetes	20 days?	2/02/15 8:00	27/02/15 17:00
Instalación de paquetes	10 days?	2/03/15 8:00	13/03/15 17:00
Creación de sistema instalable	10 days?	16/03/15 8:00	27/03/15 17:00
Creación Manual de usuario	15 days?	25/03/15 8:00	14/04/15 17:00
Realización de pruebas de aceptación	15 days?	15/04/15 8:00	5/05/15 17:00
Plantilla de no conformidades	10 days?	6/05/15 7:00	19/05/15 17:00
Incorporación de mejoras	9 days?	19/05/15 7:00	29/05/15 17:00

*Tabla 3: Tareas para el desarrollo de la personalización de GNU/Linux Nova.*

En la Tabla 4: Tareas para el proceso de personalización de GNU/Linux Nova para el sistema de educación primaria en función de riesgos se muestran las tareas establecidas para el proceso de construcción de la personalización de Nova en función de los riesgos a que se encuentran expuestas cada una de ellas y las acciones tomadas para mitigar estos.

Asignada a: Manuel Alejandro Sánchez del Campo

Nombre	Descripción	Fecha - Inicio	Fecha - Fin	Riesgo	Mitigación	Estado
Recopilación de Información.	<p>Estudio del estado del arte.</p> <p>Estudio de herramientas para la personalización y construcción de distribuciones GNU/Linux.</p> <p>Estudio de herramientas para la construcción de repositorios personalizados.</p> <p>Estudio sobre software educativos disponibles para GNU/Linux.</p>	03/11/14	01/05/15	<ul style="list-style-type: none"> <li>- Contar con poca información para consulta.</li> <li>- Información de calidad dudosa.</li> </ul>	<ul style="list-style-type: none"> <li>- Establecer métodos de búsqueda de información</li> <li>- Correcta gestión de la información</li> </ul>	Hecho
<b>Levantamiento de Requisitos</b>						
Identificación de Requisitos	- Se realizará una entrevista con el cliente, recopilando información sobre las necesidades y particularidades que debe satisfacer el sistema	03/11/14	14/11/14	<ul style="list-style-type: none"> <li>- Falta de requisitos necesarios para la realización de la personalización.</li> <li>- Dificultad para reunirse con el cliente.</li> </ul>	<ul style="list-style-type: none"> <li>- Hacer preguntas claras al cliente.</li> <li>- Elaboración de un cuestionario</li> <li>- Contactar con anticipación con el cliente para coordinar los encuentros.</li> </ul>	Hecho
Especificación y verificación de requisitos del sistema.	Como artefacto de expediente de proyecto, según la metodología Nova OpenUp, se redactará el documento oficial que contiene los requisitos del sistema(010113_Especificacion_de_requisitos_de_software).	10/11/14	21/11/14	<ul style="list-style-type: none"> <li>- Poca Comprensión por parte del cliente.</li> <li>- Desacuerdo por parte del cliente con alguno(s) de los requisitos.</li> </ul>	<ul style="list-style-type: none"> <li>- Detallar claramente las definiciones y acciones propuestas.</li> <li>- Establecer un efectivo control de las tareas.</li> </ul>	Hecho
Descripción de Requisitos Ágiles.	- Se detallarán los requisitos en función del resultado a esperar, teniendo en cuenta su procesamiento Interno (010114c_Descripcion_de_requisito_agil).	17/11/14	28/11/14	- Demora en la entrega.	- Establecer un control efectivo de las tareas.	Hecho

Definición de Arquitectura						
Definición de vista del sistema.	- Es una descripción del sistema en cuanto a los componentes del mismo. Se establecerá la relación de prioridad entre los requisitos, se muestran los paquetes y componentes y los escenarios presentes en el sistema. (010216_0_Arquitectura_de_software).	28/11/14	11/12/14	- Demora en la entrega.	- Establecer un control efectivo de las tareas.	Hecho
Definición de vista de infraestructura.	Definir las herramientas a utilizar en el desarrollo del sistema y como deben ser configuradas. (010216_8_Arquitectura_vista_de_infraestructura).	05/12/14	18/12/14	- Demora en la entrega.	- Establecer un control efectivo de las tareas.	Hecho
Definición de vista de seguridad.	- Permitirá presentar los diferentes métodos para garantizar la seguridad de la distribución GNU/Linux a partir de los requisitos a cumplir y teniendo en cuenta los diferentes niveles del producto (010216_7_Arquitectura_vista_de_seguridad).	12/12/14	25/12/15	- Demora en la entrega.	- Establecer un control efectivo de las tareas.	Hecho
Definición de vista de despliegue	Describirá las propiedades que deben poseer los Ordenadores para que la distribución GNU/Linux sea capaz de funcionar sobre ellos.	26/12/15	08/01/15	- Demora en la entrega.	- Establecer un control efectivo de las tareas.	Hecho
Desarrollo del Sistema						
Generación de repositorios binarios	Se descarga el repositorio el repositorio de fuentes y binario que constituirán la base del repositorio a crear	05/01/15	09/01/15	- Exceder el tamaño establecido para repositorio.  - No incluir alguno(s) de los software incluidos en el catálogo.	- Establecer un control efectivo sobre el catálogo de software seleccionado.	Hecho
Creación del Sistema Operativo Base.	- Se conformará el sistema operativo base para para su posterior configuración	12/01/15	30/02/2015	- Demora en la entrega.	- Establecer un control efectivo de las tareas.	Hecho



Instalación de Paquetes.	- Se instalarán los paquetes necesarios para el funcionamiento del sistema operativo final.	02/03/15	13/03/15	- Error de dependencias.	- Buscar e instalar dependencias necesarias.	Hecho
Creación de sistema instalable	Se empaquetará la personalización en un archivo ISO que podrá ser guardado en un CD, DVD o dispositivo USB y posteriormente	16/03/15	27/03/15	- No presenta.		Hecho
Creación del manual de usuario	Se readactará una guía que ayude a los usuarios a comprender la estructura y uso del sistema	25/03/15	14/04/15	- No presenta.		Hecho
<b>Mejoras al sistema</b>						
Plantilla de no conformidades	Los problemas reportados por el cliente serán recogidos en un documento para su posterior solución.	06/05/15	19/05/15	- No presenta.		Hecho
Incorporación de mejoras	Se da solución a los problemas recogidos en la plantilla de no conformidades.	19/05/15	29/05/15	- No presenta.		Hecho

*Tabla 4: Tareas para el proceso de personalización de GNU/Linux Nova para el sistema de educación primaria en función de riesgos*

## 2.3 Modelo del proceso de construcción de Nova Educativo

A continuación se muestran las actividades llevadas a cabo durante la construcción de la personalización de Nova para el sistema de educación primaria en Cuba, teniendo en cuenta las decisiones a tomar y los artefactos generados en cada actividad, como se muestra en la Figura 2: Diagrama del proceso de construcción de la personalización de GNU/Linux Nova para la educación primaria.

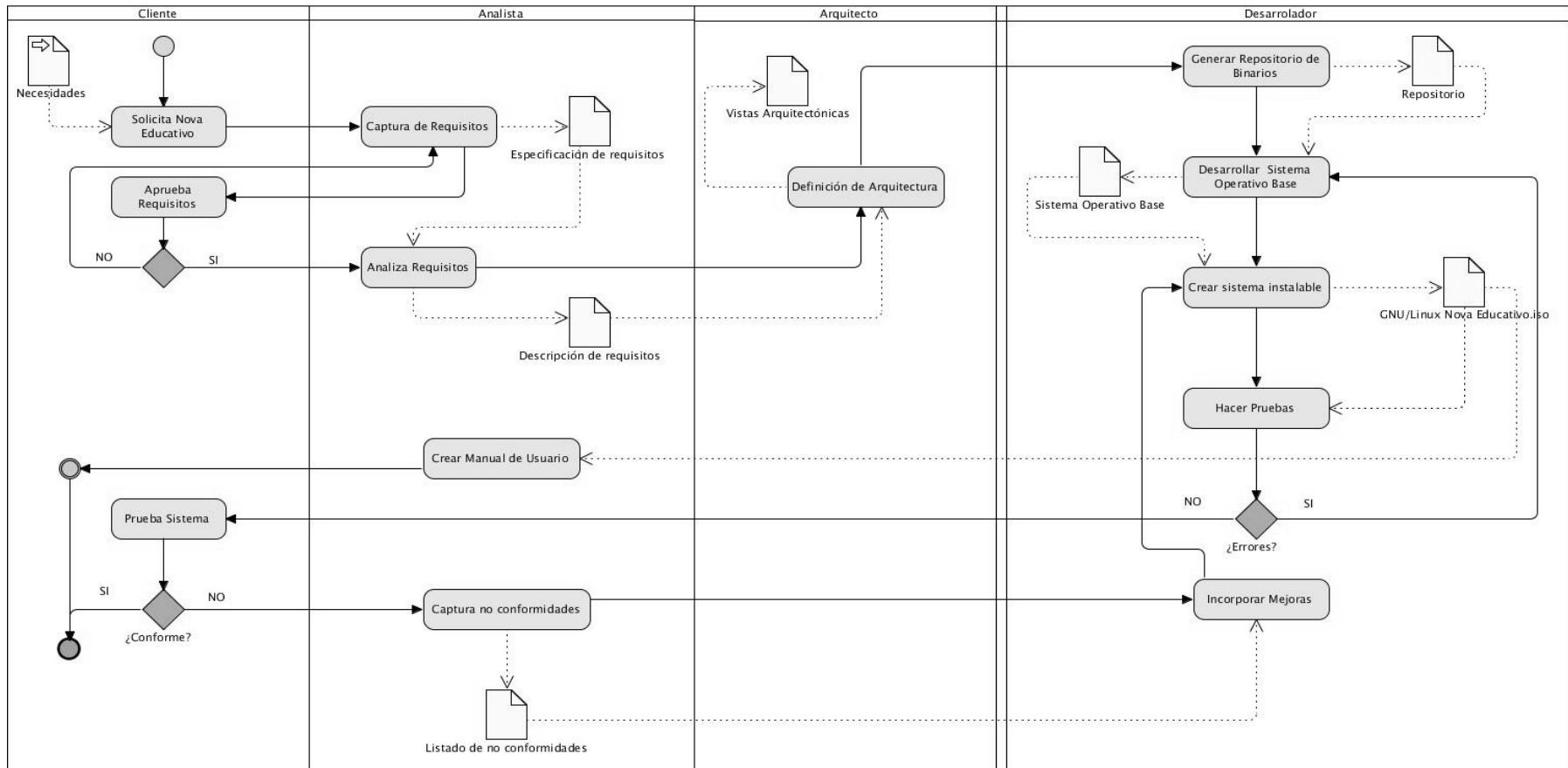


Figura 2: Diagrama del proceso de construcción de la personalización de GNU/Linux Nova para la educación primaria.

## **2.4 Requisitos de software**

Los requerimientos de sistemas deben mostrar todo lo que el sistema debe hacer más todas las restricciones sobre la funcionalidad. Deben ser suficientemente entendibles como para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre qué debe y qué no debe hacer el sistema (Rota 2000).

### **2.4.1 Requisitos funcionales**

**RF1:** Proveer software educativo que se ajuste a las necesidades del proceso docente educativo de la enseñanza primaria.

**RF2:** Crear nuevos documentos con un procesador de textos, darle formato a los mismos y guardarlos en diferentes formatos.

**RF3:** Permitir al usuario la realización de acciones en hojas de cálculos en diferentes formatos.

**RF4:** Crear y editar presentaciones en diferentes formatos.

**RF5:** Permitir al usuario visualizar contenido web.

**RF6:** Permitir la visualización de imágenes con opciones de tamaño y rotación.

**RF7:** Permitir la reproducción de música y vídeos en diferentes formatos.

**RF8:** Permitir al usuario escoger si desea salir de la sesión: suspender, apagar o reiniciar el equipo.

**RF9:** Permitir la instalación del sistema desde un CD, DVD o dispositivo USB.

### **2.4.2 Descripción de los requisitos ágiles**

Una correcta descripción de los requisitos ágiles según establece la metodología Nova OpenUp, permitirá contar con un punto de vista más detallado en el momento de definir la arquitectura de la distribución, estableciendo un punto de partida para las pruebas de funcionamiento del sistema y la

posterior incorporación de mejoras al mismo.


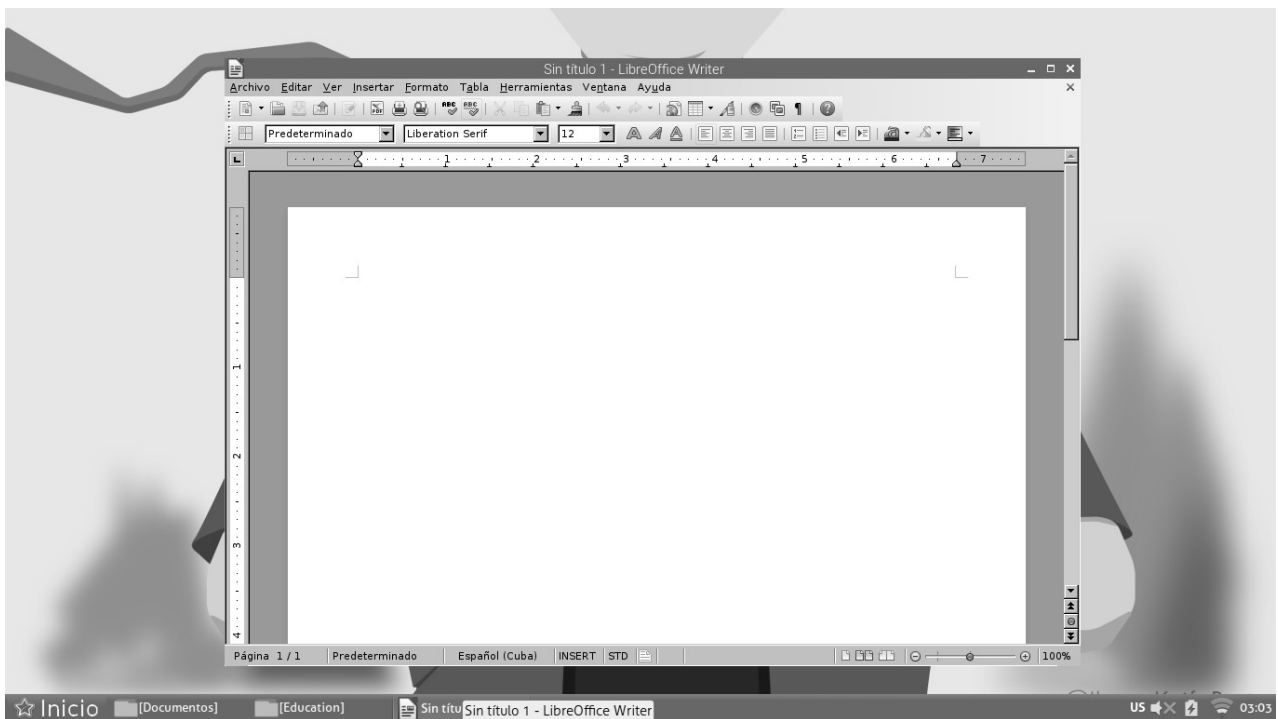
<b>Número:</b> RF1	<b>Nombre del requisito:</b> Proveer software educativo que se ajuste a las necesidades del proceso docente educativo de la enseñanza primaria.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 24 días	
<b>Riesgo en Desarrollo:</b> Mala selección del software educativo	<b>Tiempo Real:</b> 4 semanas	
<b>Descripción:</b> El sistema debe proveer al usuario de aplicaciones educativas que le provean de las herramientas necesarias para nutrir su proceso de enseñanza-aprendizaje. (Figura 3)		
<b>Prototipo de Interfaz</b>		
		
<i>Figura 3: Menú de aplicaciones educativas</i>		

Tabla 5: Descripción del RF1 Proveer software educativo que se ajuste a las necesidades del proceso docente educativo de la enseñanza primaria.

<b>Número:</b> RF2	<b>Nombre del requisito:</b> Crear nuevos documentos con un procesador de textos, darle formato a los mismos y guardarlos en diferentes formatos.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 7 días	
<b>Riesgo en Desarrollo:</b> No posee riesgo.	<b>Tiempo Real:</b> 1 semana	
<b>Descripción:</b> El sistema debe contar con un procesador de textos con una calidad igual o superior a Microsoft Office Word. (Figura 4)		

**Prototipo de Interfaz**

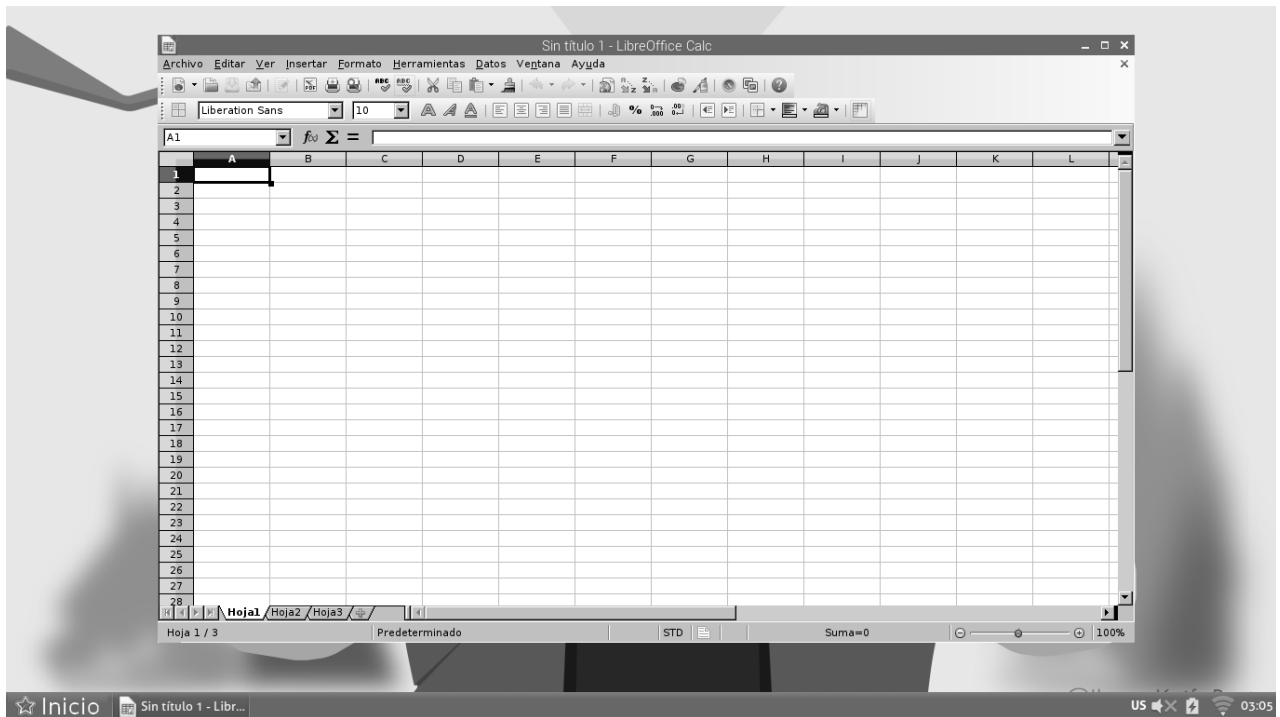


*Figura 4: LibreOffice Writer*

*Tabla 6: Descripción del RF2: Crear nuevos documentos con un procesador de textos, darle formato a los mismos y permitir guardar en diferentes formatos.*

<b>Número:</b> RF3	<b>Nombre del requisito:</b> Permitir al usuario la realización de acciones en hojas de cálculos en diferentes formatos.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 7 días	
<b>Riesgo en Desarrollo:</b> No posee riesgo.	<b>Tiempo Real:</b> 1 semana	
<b>Descripción:</b> El sistema debe contar con una aplicación para la elaboración y uso de hojas de calculo con una calidad igual o superior a Microsoft Office Excel. (Figura 5)		

**Prototipo de Interfaz**



*Figura 5: LibreOffice Calc*

*Tabla 7: Descripción del RF3: Permitir al usuario la realización de acciones en hojas de cálculos en diferentes formatos.*

<b>Número:</b> RF4	<b>Nombre del requisito:</b> Crear y editar presentaciones en diferentes formatos.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 7 días	
<b>Riesgo en Desarrollo:</b> No posee riesgo.	<b>Tiempo Real:</b> 1 semana	

**Descripción:** El sistema debe contar con una potente aplicación para la elaboración de presentaciones con una calidad igual o superior a Microsoft Office Power Point. (Figura 6)

**Prototipo de Interfaz**

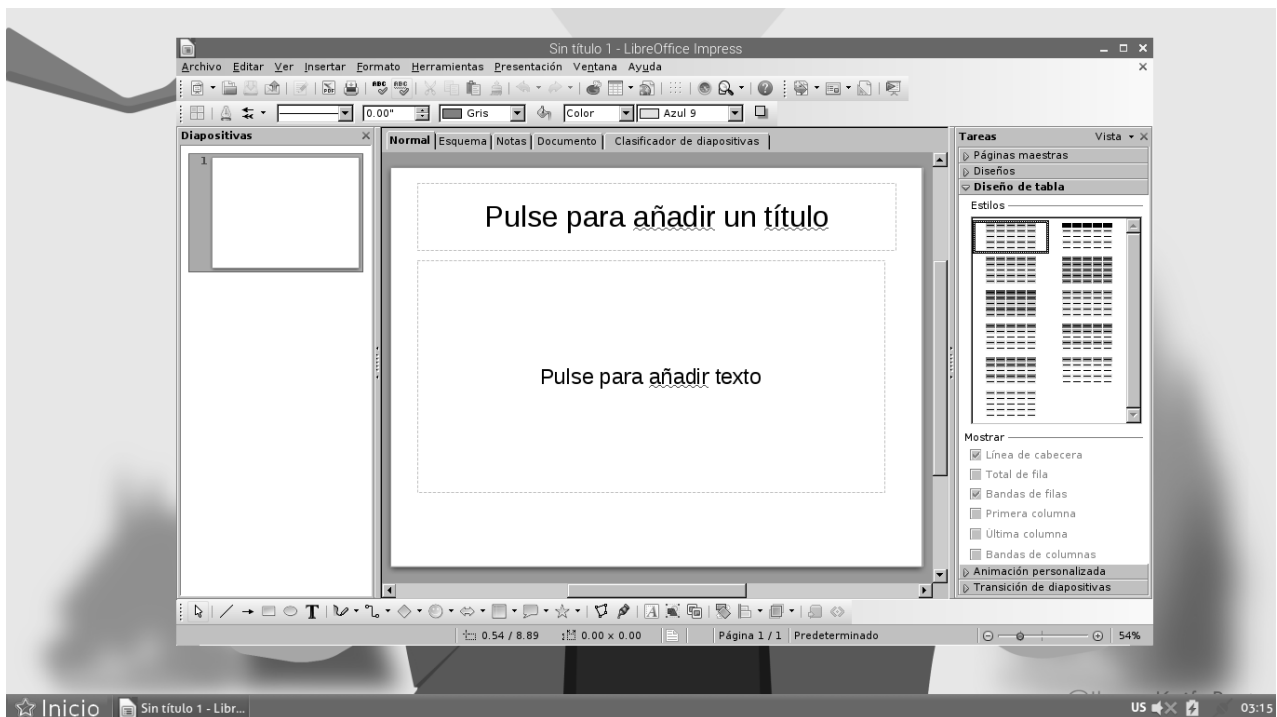


Figura 6: LibreOffice Impress

Tabla 8: Descripción del RF4: Crear y editar presentaciones en diferentes formatos.

<b>Número:</b> RF5	<b>Nombre del requisito:</b> Permitir al usuario visualizar contenido web.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 7 días	
<b>Riesgo en Desarrollo:</b> No posee riesgo.	<b>Tiempo Real:</b> 1 semana	
<b>Descripción:</b> El sistema debe contar con un navegador web, que permita visualizar contenidos desde la web o ficheros locales con formatos html, mht, htm, etc. (Figura 7)		

**Prototipo de Interfaz**



*Figura 7: Navegador web Mozilla Firefox*

*Tabla 9: Descripción del RF5: Permitir al usuario la facilidad de visualizar contenido web.*



<b>Número:</b> RF6	<b>Nombre del requisito:</b> Permitir la visualización de imágenes con opciones de tamaño y rotación.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 7 días	
<b>Riesgo en Desarrollo:</b> No posee riesgo.	<b>Tiempo Real:</b> 1 semana	
<b>Descripción:</b> La distribución ha de contar con un visor de imágenes. (Figura 8)		

**Prototipo de Interfaz**

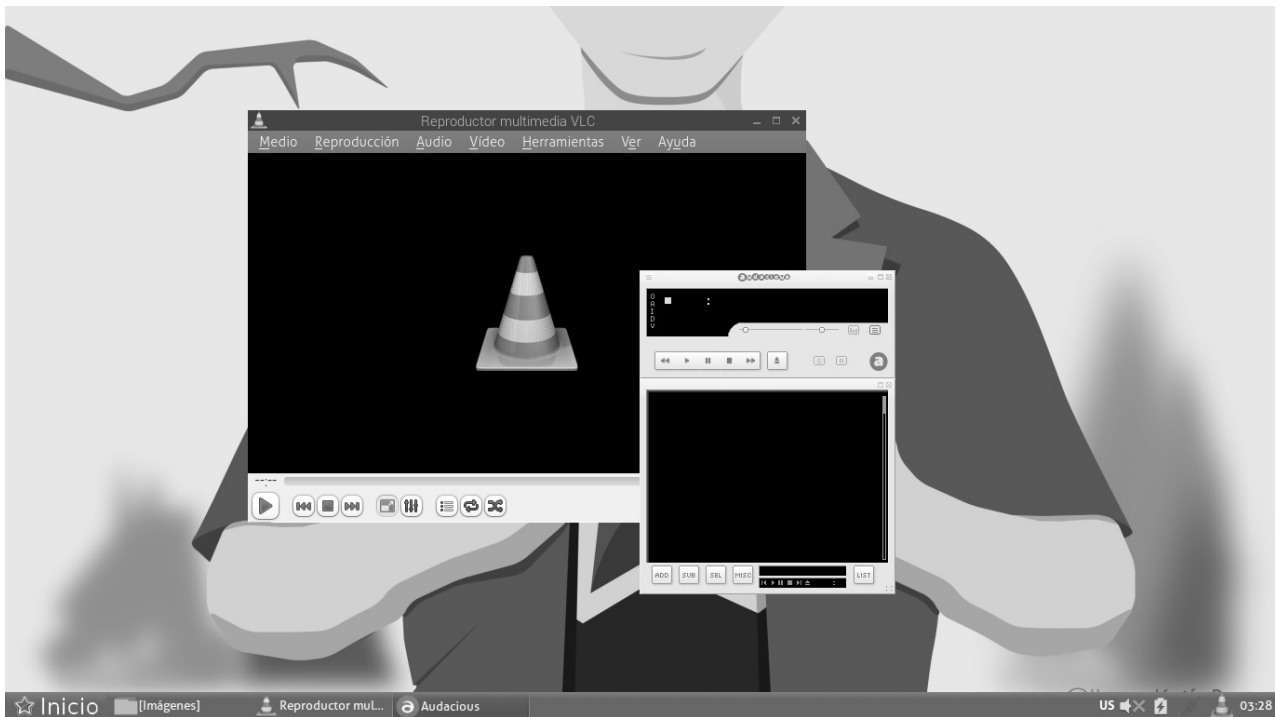


*Figura 8: Visor de Imágenes*

*Tabla 10: Descripción del RF6: Permitir la visualización de imágenes con opciones de tamaño y rotación.*

<b>Número:</b> RF7	<b>Nombre del requisito:</b> Permitir la reproducción de música y vídeos en diferentes formatos.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 7 días	
<b>Riesgo en Desarrollo:</b> No posee riesgo.	<b>Tiempo Real:</b> 1 semana	
<b>Descripción:</b> El sistema debe proporcionar al usuario al menos un reproductor de video y un reproductor de audio. (Figura 9)		

**Prototipo de Interfaz**

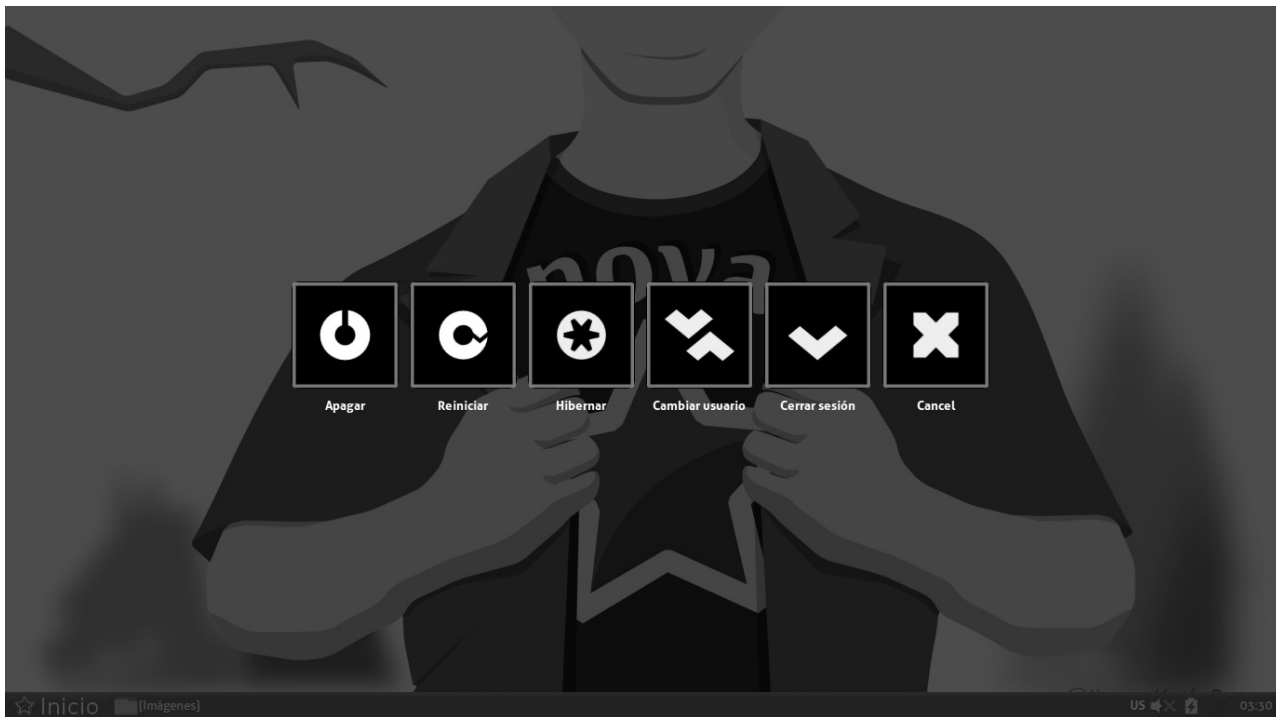


*Figura 9: Reproductor de video VLC y reproductor de audio Audacious*

*Tabla 11: Descripción del RF7: Permitir la reproducción de música y vídeos en diferentes formatos.*

<b>Número:</b> RF8	<b>Nombre del requisito:</b> Permitir al usuario escoger si desea salir de la sesión: suspender, apagar o reiniciar el equipo.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> No posee riesgo.	<b>Tiempo Real:</b> 1 semana	
<b>Descripción:</b> El sistema debe brindar las opciones para que el usuario suspenda, apague o reinicie el ordenador. (Figura 10)		

**Prototipo de Interfaz**



*Figura 10: Menú con opciones para abandonar la sección actual*

*Tabla 12: Descripción del RF8: Permitir al usuario escoger si desea salir de la sesión: suspender, apagar o reiniciar el equipo.*

<b>Número:</b> RF9	<b>Nombre del requisito:</b> Permitir la instalación del sistema desde un CD, DVD o dispositivo USB.	
<b>Programador:</b> Manuel Alejandro Sánchez del Campo	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 días	
<b>Riesgo en Desarrollo:</b> No posee riesgo.	<b>Tiempo Real:</b> 1 semana	
<b>Descripción:</b> El sistema podrá ser instalado desde un CD, DVD o USB.		

Tabla 13: Descripción del RF9: Permitir la instalación del sistema desde un CD, DVD o dispositivo USB.

### 2.4.3 Requisitos no funcionales

#### Requerimientos de usabilidad:

**RnF 1:** El sistema debe estar apto para su uso por usuarios a partir de los 6 años de edad.

#### Requerimientos de hardware

**RnF 2:** El sistema debe correr en ordenadores de pocas prestaciones, con al menos 256 MB de memoria RAM.

**RnF3:** Se debe disponer de al menos 5GB disponibles para la instalación del sistema.

#### Requerimientos de interfaz:

**RnF 4:** El sistema utilizará como entorno de escritorio el *shell* de Nova Ligero Guano, con un tema personalizado.

#### Requerimientos de mantenibilidad:

**RnF 5:** Las no conformidades reportadas deberán ser corregidas en un período no mayor a 20 días.

#### Requerimientos de licencia:

**RnF 6:** La personalización de Nova se liberará bajo la licencia GPL v3.

**Requerimientos legales, de derecho de autor y otros:**

**RnF 7:** Registrar el producto cuando culmine su desarrollo.

## **2.5 Conclusiones del capítulo**

En el capítulo se definió el proceso de construcción de la personalización de la Distribución Cubana de GNU/Linux Nova para el sistema de educación primaria en Cuba, brindando una panorámica de las funcionalidades presentes en el sistema, lo que permitió establecer las bases para la implementación y comprensión del mismo.

De igual forma se generaron los artefactos correspondientes a la metodología Nova OpenUp y se definieron los requisitos funcionales y no funcionales recopilados durante las entrevistas con el cliente.

## **Capítulo3. Implementación y Pruebas**

En el presente capítulo se explicará detalladamente el proceso de construcción de la personalización de la Distribución Cubana de GNU/Linux Nova para el sistema de enseñanza primaria en Cuba. Y al término de este, las pruebas realizadas al sistema final.

### **3.1 Proceso de construcción**

La construcción de una distribución de GNU/Linux se realiza dependiendo de lo que se desea obtener, sea un sistema completo, un sistema mínimo, un sistema particular con aplicaciones específicas, un sistema con kernel determinado, etc. Todos los pasos requieren de un conjunto de aplicaciones y configuraciones necesarias para la construcción del sistema funcional y completo.

#### **3.1.1 Elementos necesarios para la construcción**

Para la construcción de una distribución de GNU/Linux es necesario contar con algunos componentes específicos y habilidades en el manejo y manipulación de sistemas GNU/Linux.

Entre los componentes requeridos se incluyen:

- Kernel con soporte *aufs* y *squashfs*.
- Herramientas que acompañan a *aufs* y *squashfs*.
- Una versión de Grub.
- Conjunto de paquetes mínimos del live.
- Debootstrap.
- Herramientas para entornos *JAIL*.

Algunos conocimientos y habilidades necesarias serán:

- Manejo del `initrd`, creación, modificación.
- Manejo de montaje & desmontaje de unidades virtuales.
- Manejo de administración de procesos virtuales.
- Manejo de servicios virtuales.
- Manejo en manipulación de paquetes: Creación, modificación.
- Manejo de herramientas como: `dpkg`, `apt`, `aptitude`, `synaptic`, etc.
- Manejo de archivos temporales.
- Manejo del administrador de servicios de arranque.
- Manipulación de archivos `rc.S*` y archivos contenidos en el `init.d`.

### 3.1.2 Preparativos previos a la construcción del Sistema Operativo Base

Como quedó definido en el primer capítulo del presente trabajo de diploma, la herramienta a utilizar en el proceso de construcción del sistema operativo base será **debootstrap**, la cual será instalada desde los repositorios de GNU/Linux Nova.

```
$ sudo apt-get install debootstrap
```

Nota: Dado que todo el proceso a partir de este paso se realizará con permisos de *root*, lo más conveniente será registrarse como tal en el sistema, evitando tener que escribir la orden **sudo** al inicio de cada instrucción. Para ello se hace uso del comando:

```
$ sudo su
```

El próximo paso antes de la creación del SOB será crear un directorio donde alojar el mismo, el cual se llamará `nova-edu`.

```
# mkdir /mnt/nova-edu
```

### 3.1.3 Construcción del SOB

Una vez concluidos los preparativos, se procede a la creación del SOB con la herramienta `debootstrap`.

```
# debootstrap --arch=i386 --components=principal,extendido 2013  
/mnt/nova-edu/ http://nova.f10.uci.cu/nova
```

Este proceso tardará de acuerdo a la velocidad de conexión con el repositorio, si se dispone de un repositorio local solo serán unos pocos minutos. Las opciones establecidas en el proceso fueron **--arch** para definir la arquitectura y **--components** para especificar que los paquetes a utilizar serán de las ramas principal y extendido del repositorio, por otra parte **2013** es el nombre de la versión de Nova a emplear (en el caso de otro sistema operativo como Ubuntu este valor podría ser Trusty, Saucy o Utopic, por solo mencionar algunos ejemplos). Los próximos parámetros definidos en el comando constituyen la ruta donde se creará el SOB y la dirección del repositorio desde donde se obtendrán los paquetes.

### 3.1.4 Configuraciones iniciales

Una vez concluido el proceso de construcción del SOB, este quedará almacenado en el directorio `/mnt/nova-edu` y el próximo paso lógico será darle vida, para lo cual se requiere del sistema anfitrión; éste proveerá de configuración en tiempo de ejecución, dispositivos y carpetas esenciales para la manipulación del *chroot*.

Para ello se montarán los directorios:

- **/proc:** Contiene un sistema de archivos imaginario o virtual. Este no existe físicamente en disco, sino que el núcleo lo crea en memoria. Se utiliza para ofrecer información relacionada con el sistema (originalmente acerca de procesos, de aquí su nombre). (Weeks y Stafford 2003)
- **/dev:** Contiene los archivos de dispositivos especiales para todos los dispositivos de hardware. (Weeks y Stafford 2003)
- **/sys:** Contiene información similar a la que se encuentra en `/proc/`, pero muestra una vista jerárquica de la información de dispositivos específica con relación a los dispositivos de conexión en caliente. (Red Hat 2005)



- **/dev/pts**: Este sistema de ficheros vive exclusivamente en la memoria, las entradas en */dev/pts* son pseudo-terminales (pty). (Weeks y Stafford 2003)

El montaje se realizará haciendo uso de los siguientes comandos en la terminal:

```
# mount --bind /dev/ /mnt/nova-edu/dev/  
# mount none -t proc /mnt/nova-edu/proc/  
# mount none -t sysfs /mnt/nova-edu/sys/  
# mount none -t devpts /mnt/nova-edu/dev/pts/
```

Para disponer de acceso a la red y poder hacer uso de los repositorios, se deben copiar los ficheros **resolv.conf** y **hosts** del equipo anfitrión al SOB creado.

```
# cp /etc/resolv.conf /mnt/nova-edu/etc/resolv.conf  
# cp /etc/hosts /mnt/nova-edu/etc/hosts
```

Posteriormente se monta el entorno *chroot*, permitiendo interactuar con el sistema que se está creando como si se tratara de un sistema en marcha.

```
# chroot /mnt/nova-edu
```

Una vez dentro, el próximo paso será crear una salva del archivo **initctl**, para luego crear un enlace simbólico del mismo a **/bin/true**.

```
# mv /sbin/initctl /sbin/initctl.nim_blocked  
# ln -s /bin/true /sbin/initctl
```

Luego se exportan algunas variables del sistema y se genera la variable local.

```
# export PS1="(chroot) $PS1" export LC_ALL=C  
# locale-gen es_ES.utf8
```

Una vez concluidos los pasos anteriores, se procede a sincronizar el índice de paquetes desde el repositorio fuente para posteriormente instalar y correr el *script* de post-instalación de **dbus**.

```
# apt-get update  
# apt-get install --yes dbus  
# dbus-uuidgen > /var/lib/dbus/machine-id
```

### 3.1.5 Instalación de paquetes

Tras construir el SOB y establecer las configuraciones iniciales, se procede a la instalación de los

paquetes y software que integrarán el sistema funcional y completo, así como el proceso de personalización de algunos de sus componentes.

Para realizar la instalación de los paquetes se hará uso del comando:

```
# apt-get install paquete1 paquete2
```

El listado de paquetes a instalar puede ser consultado en las tablas siguientes: (Tabla 14: Paquetes básicos del sistema, Tabla 15: Paquetes educativos, Tabla 16: Otros paquetes complementarios.).

Paquete	Descripción
nova-light	Metapaquete en el cual se incluyen todas las dependencias necesarias para la conformación de la distribución “Nova Ligero”.
casper	Permite cargar un sistema en modo <i>live</i> desde un dispositivo de sola lectura.
serere3-gtk3	Instalador de la Distribución Cubana de GNU/Linux Nova 4.0.
linux-image-generic	Imagen genérica del kernel del sistema operativo Linux.
squashfs-tools	Herramienta para crear y añadir sistemas de archivos <i>squashfs</i> .
ubuntu-restricted-extras	Paquetes necesarios para la lectura de varios tipos de archivos multimedia.
lightdm-webkit-greeter	Gestor de sesiones.

Tabla 14: Paquetes básicos del sistema

Paquete	Descripción
Jclic	Paquete que contiene una serie de aplicaciones que pueden ser utilizadas para llevar a cabo diferentes actividades educativas: rompecabezas, asociaciones, ejercicios de texto, crucigramas, entre otros (Mora 2011).
Gcompris	Colección de juegos educativos para niños de dos a diez años. Contiene más de cien programas que ofrecen una amplia gama de actividades para el entretenimiento y aprendizaje (Mora 2011).
Ktuberling	Juego de construcción. A través de esta aplicación se pueden arrastrar y colocar ojos, nariz, orejas, ropa y accesorios a una papa con forma humana. De igual manera el programa posee otros ambientes denominados <i>Zona de Juego</i> , en los cuales se pueden insertar objetos para crear ambientes personalizados, tales como: la luna, navidad, pizzería, antiguo Egipto, entre otras (Mora 2011).
Pysycache	Programa diseñado principalmente para niños y niñas de 4 a 7 años. Enseña a manipular e identificar las partes del ratón de la computadora a través de juegos variados con imágenes. También puede ser utilizado por personas adultas que se encuentran por primera vez con una computadora. Sirve para enseñar el movimiento del ratón, hacer clic, arrastrar y soltar (Mora 2011).
Blinken	Versión para GNU/Linux del popular juego ¡Simón dice!, en donde se debe tratar de memorizar un patrón de sonidos y luces presentado para luego pulsar el botón de inicio de juego y tratar de repetir la secuencia en el mismo orden. Para ganar la secuencia debe tener el orden correcto y completo (Mora 2011).
Childsplay	Programa de juegos para la memoria, aprendizaje de sonidos, imágenes, letras y números, actividades que entrenan a niños en el uso del ratón y el

Paquete	Descripción
	teclado. Además tiene actividades tales como rompecabezas y juego de billar (Mora 2011).
Klettres	Programa de apoyo para leer y pronunciar el alfabeto, asociando letras con el sonido al pronunciarlas. Cuenta con dos modos de uso: adulto e infantil (Mora 2011).
Kbruch	Ayuda a aprender a calcular problemas con fracciones (Mora 2011).
Tuxmath	Juego arcade educativo que protagoniza la mascota de Linux, el pingüino Tux. Está basado en el juego de <i>Missile Comand</i> y el objetivo principal es que Tux defienda sus ciudades, en este caso resolviendo operaciones matemáticas (Mora 2011).
Celestia	Simulador del espacio que no se limita a la superficie de la Tierra. Con Celestia es posible viajar por todo el sistema solar, por cualquiera de sus más de cien mil estrellas, o incluso más allá de la galaxia. Viene con un gran catálogo de estrellas, planetas, lunas, asteroides, cometas y naves espaciales (Mora 2011).
Tuxpaint	Es un programa de dibujo simple para jóvenes o niños. No pretende ser una herramienta de dibujo de uso general, solo ser divertido y fácil de usar. Un personaje animado con efectos de sonido ayuda a que el usuario sepa lo que está pasando, y lo mantiene entretenido. Tux Paint es extensible. Por ejemplo, un profesor puede poner una colección de figuras de animales y pedir a sus estudiantes que dibujen ecosistemas. Cada figura tiene un sonido que se reproduce y texto que se muestran cuando el usuario selecciona la figura (Mora 2011).
Ktouch	Es un ayudante para aprender a escribir con rapidez y precisión en el

Paquete	Descripción
	computador. Proporciona un texto de ejemplo e indica que dedos se debe utilizar para cada tecla. Incluye una colección de lecciones en una amplia gama de diferentes idiomas y diseños de teclado. Posee distintos niveles de dificultad y las estadísticas se utilizan para ajustar dinámicamente los mismos (Mora 2011).
Tuxtype	Es un juego educativo protagonizado por Tux, el pingüino de Linux. El jugador guía a Tux para que se coma el pescado que cae desde la parte superior de la pantalla. Cada pez tiene una letra escrita en él, cuando el jugador pulsa la tecla correspondiente, Tux se posicionará y se come a los peces. El juego está dirigido a los niños que empiezan a escribir en computadora, aunque tiene mayores niveles de dificultad que, incluso mecanógrafos con experiencia pueden encontrar difícil (Mora 2011).
Dr.geo	Programa que permite construir diferentes tipos de figuras geométricas a partir de puntos creados por el usuario. La aplicación permite construir desde segmentos de rectas hasta polígonos y ofrece herramientas basadas en propiedades y transformaciones geométricas (Mora 2011).
Scratch	Entorno educativo de programación que puede ser utilizado tanto por niños como por jóvenes para facilitar el aprendizaje de la programación en computadoras. Esto lo hace ideal en la enseñanza de los fundamentos de programación, matemáticas y geometría (Mora 2011).

Tabla 15: Paquetes educativos

Paquete	Descripción
Audacious	Sencillo reproductor de audio.
Vlc	reproductor multimedia que soporta formatos tales como MPEG, MPEG2, MPEG4, DivX, MOV, WMV, QuickTime, mp3, Ogg, DVD, discos compactos y canales multimedia de una gran variedad de fuentes. (Mora 2011)
Gnomine	Popular juego del buscaminas.
Aisleriot	Popular juego de cartas.

Tabla 16: Otros paquetes complementarios.

Una vez finalizada la instalación de todos los paquetes que conformarán el sistema, se procede a eliminar todos los paquetes **.deb** almacenados en cache.

```
# aptitude clean
```

### 3.1.6 Configuraciones al sistema

Para la obtención del sistema final resulta necesario cambiar algunas configuraciones, de manera que la personalización se adapte a las especificaciones recogidas en las entrevistas con el cliente.

#### Gestor de Sesiones

El primer paso consistirá en el cambio del gestor de sesiones presente por defecto en *Nova Ligero* por otro desarrollado propiamente para la personalización. Para ello es necesario copiar desde fuera del *chroot* la carpeta con el tema del nuevo gestor de sesiones.

```
$ sudo cp -rf /home/user/antergos mnt/nova-edu/usr/share/lightdm-  
webkit/themes
```

Luego se edita desde dentro del *chroot* el archivo de configuración de Lightdm.

```
# vim /etc/lightdm/lightdm.conf
```

Y se sustituye el gestor de sesiones *lightdm-gtk-greeter* por *lightdm-webkit-greeter* cambiando el valor del parámetro **greeter-session**.

```
greeter-session=lightdm-webkit-greeter
```

Por último se establece el tema de *Lightdm* anteriormente copiado.

```
# vim /etc/lightdm/lightdm-webkit-greeter.conf
webkit-theme=antergos
```

### Creación y configuración del usuario

Resulta necesario crear una cuenta de usuario **Estudiante** que inicie por defecto en el sistema, pero que no cuente con permisos de administración, evitando así que dicho usuario pueda comprometer la estabilidad del sistema.

```
# adduser estudiante
```

Para establecer la cuenta creada como la entrada automática del sistema, es necesario modificar el fichero de configuración **lightdm.conf**, incluyendo dentro el parámetro **autologin-user**.

```
# vim /etc/lightdm/lightdm.conf
autologin-user=estudiante
```

Luego se procede a quitar la contraseña del usuario creado, modificando la entrada correspondiente a este en el fichero **passwd** localizado en **/etc/**.

```
# vim /etc/passwd
```

Modificando la línea:

```
estudiante:x:1000:::1000:Estudiante,,,:/home/estudiante:/bin/bash
```

De forma que quede de la siguiente manera:

```
estudiante::1000:1000:Estudiante,,,:/home/estudiante:/bin/bash
```

Para garantizar que se ejecute el instalador se deberá otorgar los permisos en **/etc/sudoers**.

```
estudiante ALL=NOPASSWD:/usr/bin/serere3-gtk3
```

### Configuración del tema visual

Al estar orientada para su uso por niños, resulta necesario que los diferentes componentes de la distribución cuenten con un aspecto visual colorido y llamativo. Por lo tanto, fue necesaria la creación de un nuevo diseño estético para el sistema.

Para copiar el tema debe hacerse desde fuera de la consola donde estamos trabajando con el *chroot*.

```
$ sudo cp -r /home/user/NovaTheme /mnt/nova-edu/usr/share/themes/
```

Y posteriormente se incluyen igualmente, el nuevo tema para *lxpanel*, el fondo de escritorio y el tema de iconos a utilizar.

```
$ sudo cp -r /home/user/lxpanel /mnt/nova-edu/usr/share/  
$ sudo cp /home/user/novaEducativo.png /mnt/nova-  
edu/usr/share/backgrounds/  
$ sudo cp -r /home/user/Numix /mnt/nova-edu/usr/share/icons/
```

Con todos los componentes visuales incluidos dentro del sistema, el próximo paso será realizar las configuraciones necesarias para que sean usados por defecto.

Para definir el tema, la iconografía y la tipografía a utilizar, resulta necesario modificar el archivo **desktop.conf**.

```
# vim /etc/xdg/guano-session/Guano/desktop.conf
```

Quedando de la siguiente forma:

```
[GTK]  
sNet/ThemeName=Nova-Educativo  
sNet/IconThemeName=Numix  
sGtk/FontName=Novason Light 11  
...
```

El fondo de escritorio que utilizará el sistema debe ser establecido en el fichero de configuración **pcmanfm.conf**.

```
# vim /etc/xdg/pcmanfm/guano/pcmanfm.conf
```

Una vez dentro del archivo, se establecen los parámetros como se muestra a continuación.

```
[desktop]  
wallpaper_mode=1  
wallpaper=/usr/share/backgrounds/nova-2013-ligero_1024x768.jpg  
...
```

### 3.1.7 Restauración de las configuraciones del sistema

Llegado a este punto y sabiendo que no se realizarán más cambios dentro del sistema, se procede a la restauración de varias de las configuraciones para garantizar que una vez construido el Sistema Operativo



Final (SOF), éste se ejecute por sí mismo.

Limpiar el entorno del *chroot*.

```
# rm /var/lib/dbus/machine-id
```

Salir del *chroot*.

```
exit
```

Eliminar el archivo *inictl* creado.

```
$ sudo rm /mnt/nova-edu/sbin/initctl
```

Restaurar el archivo *inictl* por el original movido al principio hacia otra dirección.

```
$ sudo mv /mnt/nova-edu/sbin/initctl.nim_blocked /mnt/nova-edu/sbin/initctl
```

Igualmente se limpiarán los archivos temporales en busca de disminuir el tamaño final del ISO.

```
$ sudo rm -rf /mnt/nova-edu/tmp/*  
$ sudo rm /mnt/nova-edu/etc/resolv.conf
```

Luego se desmontan las unidades montadas inicialmente.

```
$ sudo umount -lf /mnt/nova-edu/proc/  
$ sudo umount -lf /mnt/nova-edu/sys/  
$ sudo umount -lf /mnt/nova-edu/dev/pts/  
$ sudo umount -lf /mnt/nova-edu/dev/
```

### 3.1.8 Empaquetado del sistema

Se crea en */mnt* la carpeta *image* la cual contendrá las subcarpetas *casper*, *isolinux*, e *install*.

```
$ sudo mkdir -p /mnt/image/{casper,isolinux,install}
```

Una vez creados los directorios, se procede a incluir el *kernel* y el archivo *initrd* generado en el *chroot* dentro de la carpeta *casper*.

```
$ sudo mv /mnt/nova-edu/boot/vmlinuz-3.8.0-35-generic  
/mnt/image/casper/vmlinuz  
$ sudo mv /mnt/nova-edu/boot/initrd.img-3.8.0-35-generic  
/mnt/image/casper/initrd.lz
```

Se copia el contenido de la carpeta **isolinux**, presente en cualquier ISO de Nova, que ya esté previamente configurada con un *splash* y un menú en concordancia con el patrón de diseño de Nova, para la carpeta creada con el mismo nombre.

Con el objetivo de modificar las opciones de entrada del *LiveCD* de manera que se adapten a la nueva personalización, se modifica dentro de la carpeta **isolinux** el archivo **txt.cfg**.

```
$ vim /mnt/image/isolinux/txt.cfg
```

Quedando como se muestra a continuación.

```
default live
label live
  menu label ^Probar Nova Educativo sin instalar
...
```

Posteriormente se instala la herramienta **squashfs-tools** y se genera el listado de los paquetes instalados en el sistema dentro de la carpeta **casper** en un fichero nombrado **filesystem.manifest**.

```
$ sudo apt-get install squashfs-tools
chroot /mnt/nova-edu/ dpkg-query -W --showformat='${Package} $
{Version}\n' | tee /mnt/image/casper/filesystem.manifest
```

De los paquetes instalados para ser usados en el *LiveCD* no todos son necesarios una vez instalado el sistema en un ordenador, por lo que se excluyen estos paquetes.

Los paquetes a excluir son:

- serere3
- serere3-gtk3
- casper
- user-setup
- live-initramfs

```
cp -v /mnt/image/casper/filesystem.manifest
/mnt/image/casper/filesystem.manifest-desktop
REMOVE='serere3 serere3-gtk3 casper user-setup live-initramfs'
for i in $REMOVE
```

```
do
  sudo sed -i "${i}/d" /mnt/image/casper/filesystem.manifest-desktop
done
```

Luego se crea el archivo **filesystem.squashfs**, donde estará comprimido el sistema en formato **squashfs**.

```
$ sudo mksquashfs /mnt/nova-edu/ /mnt/image/casper/filesystem.squashfs
```

Luego se escribe el fichero **filesystem.size**, requerido por el instalador.

```
$ sudo printf $(du -sx --block-size=1 /mnt/nova-edu | cut -f1) >
/mnt/image/casper/filesystem.size
```

Se crea el archivo **README.diskdefines**.

```
$ sudo vim /mnt/image/README.diskdefines
```

Dentro del fichero se especificará la arquitectura de la computadora utilizada, la numeración del disco y una descripción del sistema, quedando de la forma siguiente.

```
#define DISKNAME Nova Educativo
#define TYPE binary
#define TYPEbinary 1
#define ARCH i386
#define ARCHi386 1
#define DISKNUM 1
#define DISKNUM1 1
#define TOTALNUM 0
#define TOTALNUM0 1
```

Una vez terminado, se guardan los cambios y se cierra el fichero.

La próxima acción a tomar será crear una carpeta oculta con nombre **.disk** y dentro se generan otros archivos con información pertinente.

```
mkdir /mnt/image/.disk
cd /mnt/image/.disk/
touch base_installable
echo "Nova Educativo" > info
echo "http://www.nova.cu" > release_notes_url
```

Para calcular la suma de verificación **MD5**, se regresa a la carpeta **image** dentro de **/mnt**.

```
cd /mnt/image/
```

Y por último se crea el archivo con la suma de verificación MD5.

```
find . -type f -print0 | xargs -0 md5sum | grep -v "\./md5sum.txt" >  
md5sum.txt
```

### 3.1.9 Creación del repositorio personalizado

En el caso de los centros de educación primaria que no cuenten con la infraestructura necesaria para hacer uso del repositorio completo de la distribución de GNU/Linux Nova, se proveerá dentro del ISO un repositorio personalizado con algunas de las aplicaciones de uso más frecuente.

Para la construcción del mismo, se hará uso de la herramienta **Tinp**, con lo cual se escribirá en la consola la línea siguiente, sustituyendo los valores *paquete#* por los paquetes a incluir dentro del repositorio:

```
python tinp-cli.py -s nova.f10.uci.cu/nova -a "paquete1 paquete2  
...paquete100" /mnt/image/minirepo
```

### 3.1.10 Creación del ISO

La fase final del proceso la constituye la creación de la imagen de instalación ISO-9660 que contendrá al SOF.

```
mkisofs -r -V "Nova Educativo" -cache-inodes -J -l -b  
isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size  
4 -boot-info-table -o /mnt/nova-educativo.iso .
```

## 3.2 Pruebas al sistema

En todo proceso de desarrollo de software es común que aparezca la falibilidad humana, debido a la imposibilidad de este ser de trabajar y comunicarse de forma perfecta. Por lo tanto, resulta indispensable la presencia de un proceso de pruebas de software que garantice el buen funcionamiento y la calidad del

producto final.

“Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente”. (EVA 2011)

### **3.2.1 Nivel de prueba**

Las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. En el caso del sistema que da solución al problema planteado, se distingue el siguiente nivel de prueba empleado para la identificación de errores.

**Pruebas de aceptación:** Son básicamente pruebas funcionales sobre el sistema completo, y buscan comprobar que se satisfacen los requisitos establecidos. Su ejecución es facultativa del cliente, y en el caso de que no se realicen explícitamente, se dan por incluidas dentro de las pruebas de sistema. Es decir, las pruebas de aceptación son, a menudo, responsabilidad del usuario o del cliente, aunque cualquier persona involucrada en el negocio puede realizarlas. La ejecución de las pruebas de aceptación requiere un entorno de pruebas que represente el entorno de producción. (Valdez Huaraca 2013)

### **3.2.2 Métodos de prueba**

Las estrategias de pruebas del software se integran a las técnicas del diseño de casos de prueba, en una serie de pasos planificados que dan como resultado una correcta construcción del software, con tal grado de confianza que se detectarán la mayor parte de los errores existentes en él, dado que nunca se debe considerar que el producto está libre de errores.

**Pruebas de caja negra:** También conocidas como Pruebas Funcionales, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una Caja Negra en las que las pruebas se derivan a partir de la especificación

del sistema. Y el comportamiento del sistema solo debe ser determinado estudiando sus entradas y salidas relacionadas. Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en el sistema, y como consecuencia se producen una serie de salidas que revelan la presencia de defectos. (Valdez Huaraca 2013)

### **3.2.3 Diseño de casos de prueba**

El objetivo de los casos de prueba es validar que el sistema cumple con el funcionamiento esperado, de manera que cuando se dan las entradas apropiadas se producen los resultados esperados. Los datos de prueba se escogerán atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, al fin de verificar que el programa se ejecute forma correcta. Los casos de pruebas diseñados se muestran en el Anexo 2: Casos de Pruebas.

### **3.2.4 Resultados obtenidos**

Se realizaron un total de tres iteraciones de pruebas, llevadas a cabo en la escuela primaria Celia Sánchez del municipio La Lisa con estudiantes y profesores de esta institución. Estas pruebas consistieron en la confección de una clase a partir de la cual el profesor guiaba a los estudiantes teniendo en cuenta los casos de pruebas definidos. Como resultado fue posible obtener un producto con el cual los usuarios mostraron un mayor nivel de aceptación, solucionando las no conformidades detectadas antes de la terminación del sistema.

El resultado de cada una de las iteraciones se muestra en la Figura 11: Porcentaje de aceptación y errores en pruebas al sistema:

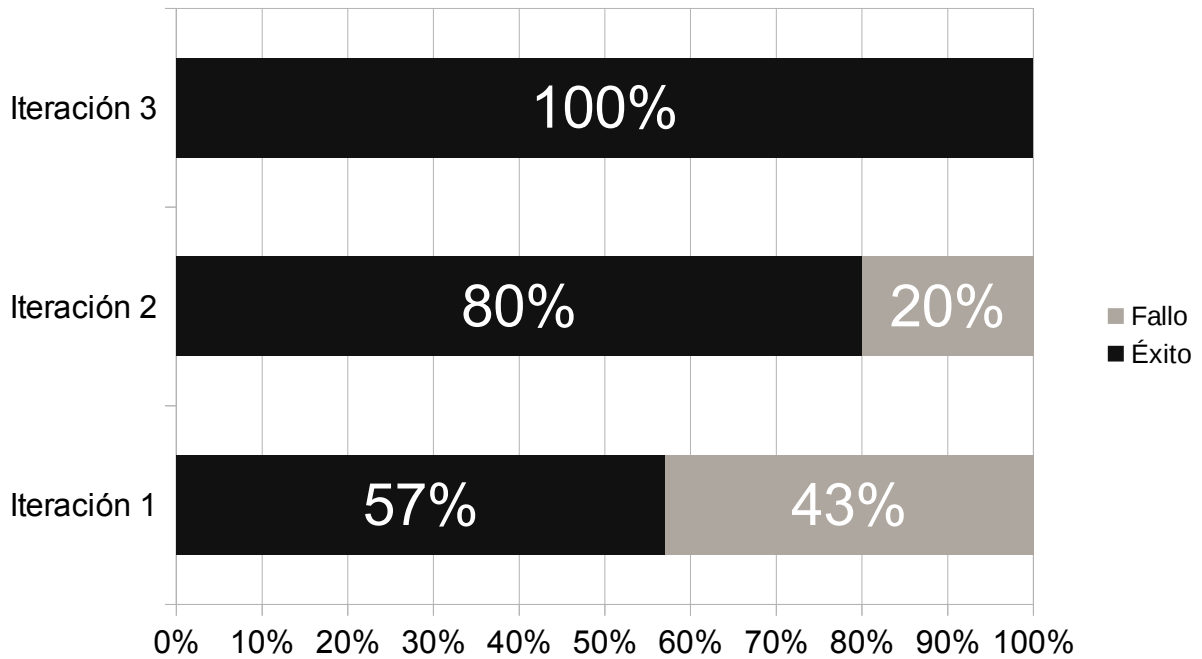


Figura 11: Porcentaje de aceptación y errores en pruebas al sistema

### 3.3 Conclusiones del capítulo

En este capítulo se describió con detalle el proceso de personalización de la distribución de GNU/Linux Nova para su uso en los centros de educación primaria en Cuba, dando como resultado una imagen lista para su instalación.

El proceso llevado a cabo y las pruebas realizadas al sistema a partir de los casos de prueba definidos teniendo en cuenta los requisitos funcionales, permitieron validar luego de tres iteraciones una personalización que cumple con las necesidades planteadas por el cliente.

## **Conclusiones Generales**

Entre los principales resultados obtenidos durante el desarrollo de la investigación y obtención del producto final se pueden destacar que:

- El estudio realizado sobre las distribuciones de GNU/Linux educativas y las herramientas para la construcción de personalizaciones, permitieron elaborar el marco teórico conceptual relacionado con los aspectos teóricos que sustentan la investigación.
- Con el desarrollo de la solución presentada en este trabajo de diploma, se crea un sistema que facilitará el empleo de la distribución de GNU/Linux Nova en la dosificación informática para la enseñanza primaria en Cuba.
- Las pruebas realizadas durante el proceso de desarrollo y por parte del cliente a partir de los casos de prueba, validan el nivel de aceptación del sistema final.



## **Recomendaciones**

Se recomienda por parte del autor de este trabajo:

- Que el documento generado en esta investigación sea estudiado e incluido dentro de la “Guía cubana de migración a software libre”.
- Divulgar y distribuir esta personalización de Nova a todos los centros de educación primaria del país.
- Desarrollar personalizaciones basadas en esta solución, dirigidas al resto de los niveles educativos del país.
- Incluir los software educativos actualmente desarrollados por el centro FORTES de la Universidad de las Ciencias Informáticas dentro de posteriores actualizaciones de la distribución.

## Referencias Bibliográficas

1. ADELL, J. y BERNABÉ, I., 2007. *Software libre en educación*. 2007. S.l.: s.n.
2. ALEGSA 2012. Paquete de software. [en línea]. [Consulta: 11 diciembre 2014]. Disponible en: <http://www.alegsa.com.ar/Dic/paquete%20de%20software.php>.
3. AMATRIAN, X. 2006. *Free software in education: a guide for its justification and implementation* [en línea]. S.l.: s.n. Disponible en: <http://www.create.ucsb.edu/~xavier/FreeSoftware/ProgramariLliureEducacio.pdf>.
4. BEEKMANS, G. 2011. Linux From Scratch. [en línea]. [Consulta: 11 diciembre 2014]. Disponible en: <http://www.linuxfromscratch.org>.
5. EVA 2011. Ingeniería de Software II. *Entorno Virtual de Aprendizaje* [en línea]. Disponible en: [http://eva.uci.cu/file.php/259/Curso\\_2010-2011/Semana\\_9/Conferencia\\_7/Materiales\\_Basicos/Sobre\\_la\\_disciplina\\_de\\_Prueba.pdf](http://eva.uci.cu/file.php/259/Curso_2010-2011/Semana_9/Conferencia_7/Materiales_Basicos/Sobre_la_disciplina_de_Prueba.pdf).
6. FREE SOFTWARE FOUNDATION 2015. ¿Qué es el software libre? [en línea]. [Consulta: 23 febrero 2015]. Disponible en: <http://www.gnu.org/philosophy/free-sw.es.html>.
7. MIRANDA, L.E. y BAYÓN, L., 2014. *Personalización de la distribución GNU/Linux Nova para el control de acceso a los comedores*. 2014. S.l.: s.n.
8. PÉREZ VILLAZÓN, Y., 2009. *Guía Cubana de Migración a plataformas de Código Abierto*. 2009. S.l.: s.n.
9. PFAFFENBERGER, B., 2000. *Linux in Higher Education: Open Source, Open Minds, Social* [en línea]. 2000. S.l.: s.n. Disponible en: <http://www2.linuxjournal.com/article/5071>.

10. PIERRA, A., 2011. *NOVA, DISTRIBUCIÓN CUBANA DE GNU/LINUX. REESTRUCTURACIÓN ESTATÉGICA DE SU PROCESO*. 2011. S.l.: s.n.
11. PRIETO, A. 2002. *Introducción a la Informática*. S.l.: Mc-Hill/Interamericana de España.
12. Real Academia Española. Diccionario Usual. [en línea] [sin fecha]. [Consulta: 20 mayo 2015 a]. Disponible en: <http://buscon.rae.es/drae/srv/search?val=software>.
13. Real Academia Española. Diccionario Usual. [en línea] [sin fecha]. [Consulta: 20 mayo 2015 b]. Disponible en: <http://buscon.rae.es/drae/srv/search?val=hardware>.
14. RED HAT 2005. Red Hat Enterprise Linux 4: Manual de referencia. *Capítulo 3. Estructura del sistema de archivos* [en línea]. [Consulta: 25 diciembre 2014]. Disponible en: <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/s1-file-system-fhs.html>.
15. SÁNCHEZ ORTIZ, S., 2014. *Estrategia de soporte técnico para el proceso de migración a software libre en los Organismos de la Administración Central del Estado*. 2014. S.l.: s.n.
16. SIERRA, D., 2007. *Visual Paradigm para UML* [en línea]. 2007. S.l.: s.n. Disponible en: <http://es.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
17. STALLMAN, R., 2003. *Por qué las escuelas deben usar exclusivamente software libre* [en línea]. 2003. S.l.: s.n. Disponible en: <http://www.gnu.org/philosophy/schools.es.html>.
18. STALLMAN, R.M. 2004. *Software libre para una sociedad libre*. S.l.: Traficantes de Sueños. ISBN 84-933555-1-8.
19. STIENEN, J., 2007. *Las TIC para el sector educativo* [en línea]. 2007. S.l.: s.n. Disponible en: <http://www.iicd.org/files/Education-impactstudy-Spanish.pdf>.

20. VALDEZ HUARACA, A.G., 2013. *PRUEBAS DE SISTEMAS Y PRUEBAS DE ACEPTACION*. 2013. S.l.: s.n.
21. VIDAL, M. 2008. Alfabetización digital e información de la sociedad. Un reto para el presente. *Revista Cubana de Informática Médica*.
22. WEEKS, A. y STAFFORD, S. 2003. Guía Para Administradores de Sistemas GNU/Linux. *Guía Para Administradores de Sistemas GNU/Linux* [en línea]. Disponible en: <http://www.tldp.org/pub/Linux/docs/ldp-archived/system-admin-guide/translations/es/html/index.html>.

## Bibliografía Consultada

- Acobson, I.; Booch, G. y Rumbaugh, J.; “El Proceso Unificado de Desarrollo de software”. 2000. Addison-Wesley.
- Booch, G.; Rumbaugh, J. y Jacobson, I.; “El Lenguaje Unificado de Modelado”. 2000. Addison-Wesley. Capítulo 11
- Pressman, Roger; Ingeniería de software. Un enfoque práctico. 2002. McGraw.Hill/Interamericana de España.
- GNU BASH. Gnu Operating Sistem. GNU Bash [online]. 2 February 2014. [Accessed 15 February 2014]. Available from: <http://www.gnu.org/software/bash/bash.html>
- [MANUAL] Openbox, instalación y configuración paso a paso, [Septiembre, 2012]. Disponible en: <http://foro.noticias3d.com/vbulletin/showthread.php?t=397929&s=bb52df3d8a1f25189028a08d41cb32a2>
- LiveCD Customization From Scratch, [Diciembre, 2014]. Disponible en: <https://help.ubuntu.com/community/LiveCDCustomizationFromScratch?action=fullsearch&value=linkto%3A%22LiveCDCustomizationFromScratch%22&context=180>
- SOFTWARE.COM.AR. Visual Paradigm para UML [online]. 2013. [Accessed 25 December 2014]. Available from: <http://www.software.com.ar/visual-paradigm-para-uml.html>.
- SOMMERVILLE. Ingeniería de Software [online]. Available from: <http://eva.uci.cu/mod/resource/view.php?id=9352>
- ECURED. metodología ágil. ECURED [online]. [no date]. [Accessed 4 April 2014]. Available from: [http://www.ecured.cu/index.php/Metodolog%C3%ADa\\_C3%A1gil#Fuente](http://www.ecured.cu/index.php/Metodolog%C3%ADa_C3%A1gil#Fuente).

- YUSLEYDI FERNÁNDEZ DEL MONTE. Metodología Nova OpenUp. June 2013. 4.0 Metodología para desarrollar la distribución cubana de GNU/Linux Nova
- DANIEL HERNANDEZ BAHR, Yunier Soler Franco and ALLAN PIERRA FUENTES, Belkis Soler Blanco. PROCESO DE CREACIÓN DEL SISTEMA OPERATIVO NOVA.
- JOANNA OJA. Guía Para Administradores de Sistemas GNU/Linux [online]. 2003. [Accessed 15 March 2014]. Available from: <http://www.tldp.org/pub/Linux/docs/ldp-archived/system-admin-guide/translations/es/html/>
- Construyento Lihuen 3 Live CD, [Noviembre, 2013]. Disponible en: [https://lihuen.linti.unlp.edu.ar/index.php?title=Construyento\\_Lihuen\\_3\\_Live\\_CD](https://lihuen.linti.unlp.edu.ar/index.php?title=Construyento_Lihuen_3_Live_CD)
- Curso de Introducción a GNU/Linux: Historia, Filosofía, Instalación y Conceptos Básicos. El sistema X- Window, entornos de escritorio gráficos. Entornos KDE y Gnome, [Febrero, 2009]. Disponible en: [http://www.ant.org.ar/cursos/curso\\_intro/x3359.html](http://www.ant.org.ar/cursos/curso_intro/x3359.html)
- Guia Ubuntu. Entorno de escritorio Xfce (2007), [Febrero, 2009]. Disponible en: <http://www.guia-ubuntu.org/index.php?title=XFce>
- [MasterMagazine, 2004] Definición y significado de Escritorio. MasterMagazine archivo 2004, [Febrero, 2009]. Disponible en: <http://www.mastermagazine.info/termino/4920.php>
- Metacity. Capítulo 3. Tablero del Escritorio de la Base, tabla consultada, [Marzo, 2009]. Disponible en: <http://www.gnome.org/~bmsmith/gconf-docs/es/metacity.html>
- Red Hat Enterprise Linux 4: Manual de referencia Capítulo 7. El Sistema X Window. Archivos de configuración del servidor X, [Marzo, 2009]. Disponible en: <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/s1-x-server-configuration.html>