

Universidad de las Ciencias Informáticas
Facultad 1



*Título: Soporte para la Arquitectura de
Computadora ARM en Nova para Cliente
Ligero.*

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

Autor: Ernesto Puente Fuentes.

Tutores: Ing. Anielkis Herrera González.

Ing. Mijail Hurtado Fedorovích.

Consultante: Ing. Jorge Luis Machín Castillo.

Mayo, 2011

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 25 días del mes de mayo del año 2011.

Ernesto Puente Fuentes

Firma del Autor

Anielkis Herrera González

Firma del Tutor

Mijail Hurtado Fedorovich

Firma del Tutor

AGRADECIMIENTOS

El primer y más importante de los agradecimientos es para toda mi familia, en especial para mi papá, mi mamá, mi hermanita que se encuentra a millas de distancia físicamente pero está muy presente en mi corazón en todo momento, y a mi tía Silvia, que más que una tía es una madre. A todos gracias por el apoyo que me han brindado durante el transcurso de mi vida y en especial de mi Universidad.

Agradecer además a mi querida novia Maidolys, que me ha sabido comprender y me ha ayudado en los momentos de tensión que he tenido para el desarrollo de esta Tesis.

A todos mis amigos, los de la UCI, los de Guantánamo y en especial a José Ernesto, mi hermano menor por horas, sin él la Universidad hubiese sido totalmente diferente.

Agradecer a todos los profesores que he tenido durante el transcurso de los 5 cursos de la Universidad, los cuales me han aportado una gran enseñanza en mi formación como un profesional.

Agradecer de manera especial a mis amigos de Nova, por todo el apoyo, la confianza y el conocimiento que supieron transmitir durante el tiempo que he estado junto a ellos y principalmente durante el desarrollo de este trabajo. En especial a Mijail y Abel por convencerme que este trabajo sería el cierre de oro para mi carrera.

De manera general, agradecer a todos los que de una manera u otra han aportado su grano de arena para que este sueño se hiciera realidad.

DEDICATORIA

A las primeras personas que va dedicado este trabajo son a mi papá, mi mamá, mi hermanita y mi tía Silvia, por confiar siempre en mi.

Está dedicado además:

A mi novia Maidolys.

A José Ernesto, por ser mi hermanito de la UCI.

A Pascual y Serguey, por ser mis hermanos de toda la vida.

A mi primo Valodia.

A mi entrenador Leonardo Cardona.

A todos los que confiaron en mí para el desarrollo de esta tarea.

RESUMEN

Mediante la integración del Instituto Central de Investigación Digital (ICID), Gente De Mérito (GEDEME) y la Universidad de las Ciencias Informáticas (UCI), surge la idea de crear el Cliente Ligerero Cubano (CLC). Para esto se utilizará la Computadora en una Tarjeta basada en la Arquitectura de Computadora Avanced RISC Machine (ARM), diseñada por el ICID. Por tal motivo es necesario un sistema operativo (SO) que posibilite la interacción entre el usuario final y el CLC. Al no existir ningún SO que cumpla con esos requisitos, es objetivo de esta investigación describir el proceso de construcción de una distribución GNU/Linux basada en Nova, que sus aplicaciones estén compiladas para la Arquitectura de Computadora ARM, capaz de funcionar sobre la Computadora en una Tarjeta diseñada por el ICID y brinde la funcionalidad de Cliente Ligerero. Durante el desarrollo de este trabajo se realiza un estudio del arte de los SO para cliente ligero existentes. Se utiliza el Proceso Unificado de Desarrollo de Software (RUP) como metodología de desarrollo, Lenguaje de Modelado Unificado (UML) y Notación de Modelado de Procesos de Negocio (BPMN) para el modelado de procesos, y los convenios definidos por el libro Linux From Scracht (LFS) para describir el proceso de construcción de SO basados en Linux.

PALABRAS CLAVE

Sistema Operativo, Arquitectura de Computadora, Cliente Ligerero, Computadora en una Tarjeta.

TABLA DE CONTENIDOS

<u>INTRODUCCIÓN.....</u>	<u>1</u>
<u>CAPITULO I: HISTORIA DEL ARTE DE LAS DISTRIBUCIONES GNU/LINUX PARA CLIENTE LIGERO.....</u>	<u>7</u>
<u>1.1 Arquitectura de Computadora.....</u>	<u>7</u>
<u>1.2 La MBTI2440: Computadora en una Tarjeta.....</u>	<u>9</u>
<u>1.3 Cliente Ligero.....</u>	<u>10</u>
<u>1.4 Método para la construcción del Sistema.....</u>	<u>13</u>
<u>1.5 Metodología de desarrollo.....</u>	<u>14</u>
<u>1.6 Modelado de Proceso.....</u>	<u>16</u>
<u>1.7 Métodos de descripción del proceso de construcción.....</u>	<u>17</u>
<u>1.8 Conclusiones Parciales.....</u>	<u>18</u>
<u>CAPÍTULO 2: DISEÑO DEL SISTEMA.....</u>	<u>19</u>
<u>2.1 Requisitos del Sistema.....</u>	<u>19</u>
<u>2.2 Diagrama de Proceso del Funcionamiento del Sistema.....</u>	<u>20</u>
<u>2.3 Arquitectura del Sistema.....</u>	<u>24</u>
<u>2.4 Núcleo.....</u>	<u>25</u>
<u>2.5 Sistema Base.....</u>	<u>25</u>
<u>2.6 Cliente Ligero.....</u>	<u>26</u>
<u>2.7 Conclusiones Parciales.....</u>	<u>27</u>
<u>CAPÍTULO 3: CONSTRUCCIÓN DEL SISTEMA.....</u>	<u>29</u>

3.1 Definir y configurar Sistema Anfitrión.....	29
3.2 Núcleo.....	31
3.3 Sistema Base.....	37
3.4 Cliente Ligero.....	42
3.5 Grabar el sistema para la computadora real.....	44
3.6 Conclusiones Parciales.....	45
CAPÍTULO 4: PRUEBAS AL SISTEMA.....	46
4.1 Pruebas a Funcionalidades.....	46
4.2 Pruebas de Integración.....	49
4.3 Pruebas al Sistema.....	52
4.4 Conclusiones Parciales.....	53
CONCLUSIONES.....	55
RECOMENDACIONES.....	56
REFERENCIAS BIBLIOGRÁFICAS.....	57
BIBLIOGRAFÍA	59
GLOSARIO.....	71
ANEXO I.....	75
ANEXO II.....	79
ANEXO III.....	89

ANEXO IV.....	100
ANEXO V.....	105
ANEXO VI.....	108
ANEXO VII.....	111
ANEXO VIII.....	118

INTRODUCCIÓN

“La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos.”(MIC 2010)

Cuba se ha encaminado a la modernización informática mediante un programa integral que involucra a varias organizaciones económicas, políticas y sociales, para lograr más y mejores productos y servicios. Potenciando en este orden el desarrollo de la industria nacional como llave primaria en la sustitución de importaciones y búsqueda de nuevas líneas de soporte económico y aporte financiero al país.

“El modelo de informatización de la sociedad cubana, basado en máquinas convencionales, mostró sus debilidades: es poco eficaz, limitado y sobre todo muy costoso. Convencidos de estas limitaciones se comenzaron a buscar alternativas que han dado en llamar Proyecto de Desarrollo de Clientes Ligeros aplicados a la Informatización de la Sociedad.”(Hurtado et al. 2010)

En este punto, el Grupo de Desarrollo de las Tecnologías de la Información (GDTI) perteneciente a la empresa GEDEME, pretenden desarrollar productos competitivos en el mercado nacional e internacional, por concepto de costo de producción,

consumo de energía y versatilidad, además con gran valor agregado por sus desarrolladores. El GDTI proyecta desarrollar un *Cliente Ligero Cubano (CLC)*.

El CLC apoyará el proceso de informatización de la sociedad cubana. Además permitirá la sustitución de importaciones y se convertirá en un nuevo producto para futuras exportaciones a mercados internacionales. Para el desarrollo de este Cliente Ligero, GEDEME utilizará la Placa Madre diseñada por personal del Instituto Central de Investigación Digital (ICID).

La Placa Madre del ICID nombrada MBTI2440 está basada en el procesador S3C2440A de SAMSUNG desarrollado para la arquitectura de computadora ARM. Diseñada con el objetivo inicial de ser utilizada en equipos médicos de mediana complejidad, se caracteriza por poseer un pequeño tamaño, bajo consumo de potencia, con gran variedad de dispositivos de interfaz y elevada conectividad. Las características anteriores, sumado a un procesamiento adecuado, le completa las prestaciones necesarias para poder ser utilizada como base para el CLC.

El CLC para su completo funcionamiento necesita un Sistema Operativo que sea capaz de funcionar sobre la MBTI2440. Para obtener esta parte vital para el desarrollo del Cliente Ligero, GEDEME opta por la integración con la Universidad de las Ciencias Informáticas (UCI) y en especial con el proyecto Nova perteneciente a la Facultad 1.

Nova como parte del grupo de proyectos de la Facultad 1 de la UCI, es la distribución cubana de GNU/Linux surgida para apoyar la migración a tecnologías de Software Libre y Código Abierto que experimenta Cuba como parte del proceso de informatización de la sociedad. Nova posee un grupo de trabajo experimentado en el proceso de construcción y personalización de distribuciones GNU/Linux, por esta razón será Nova el proyecto encargado de llevar a cabo la construcción de una distribución GNU/Linux para Cliente Ligero sobre la arquitectura ARM, que permitirá el desarrollo y funcionamiento del **Cliente Ligero Cubano**.

Por lo antes expresado definimos como **Problema Científico**:

¿Cómo construir una distribución GNU/Linux para Cliente Ligero sobre la arquitectura ARM?

Planteando como **Objeto de Estudio**, el proceso de construcción de distribuciones GNU/Linux, ubicando como **Campo de Acción** las distribuciones GNU/Linux para Cliente Ligero sobre la arquitectura ARM.

El **Objetivo General** de este trabajo es: construir una distribución GNU/Linux basada en Nova, que permita realizar las operaciones requeridas para los Clientes Ligeros sobre la arquitectura ARM.

Como **Objetivos Específicos** se plantearon los siguientes:

1. Sistematizar el proceso de construcción de distribuciones GNU/Linux.

2. Construir la distribución GNU/Linux basada en Nova para la arquitectura ARM.

Para el cumplimiento de los Objetivos Específicos definimos las siguientes **Tareas de Investigación**:

1. Sistematización de distribuciones GNU/Linux utilizadas en Cliente Ligero para definir el proceso de construcción y aplicaciones a instalar.
2. Definición de las herramientas a utilizar en el proceso de compilación cruzada para construir e instalar las aplicaciones necesarias en el Sistema.
3. Estudio de las herramientas que permitan *emular* otras arquitecturas de computadora, para emular el Sistema construido.
4. Modelación del proceso de construcción de la distribución para organizar y priorizar las actividades a realizar durante este proceso.
5. Construcción del Sistema.
6. Confección de las aplicaciones especializadas para lograr los objetivos deseados con los Clientes Ligeros.

Estas tareas pueden sufrir variaciones en la medida que avance el desarrollo de la investigación, debido al surgimiento de nuevas líneas de investigación para lograr el *Objetivo General*.

Para dar solución a las Tareas antes mencionadas se utilizaron los siguientes métodos:

Método Empírico

- **Observación:** Permitted hacer un análisis del comportamiento de algunas herramientas que apoyan el proceso de construcción de distribuciones GNU/Linux y el proceso de compilación de aplicaciones.

Método Teórico

- **Analítico-Sintético:** Se realizó el análisis de documentos, libros, artículos, Trabajos de Diplomas. También, se dividió el problema en subproblemas y se realizó un mejor análisis, para de esta manera interpretar mejor los resultados obtenidos en cada tarea realizada.
- **Histórico Lógico:** Se analizó la evolución de diferentes teorías y herramientas que apoyan la construcción de estos sistemas, mediante la búsqueda en diferentes bibliografías.

El presente trabajo está estructurado en 4 capítulos organizados de la siguiente forma:

Capítulo 1. Historia del Arte de las distribuciones GNU/Linux para Cliente Ligerito.

Se realiza un estudio de diferentes distribuciones GNU/Linux para Cliente Ligerito y el proceso de construcción de las mismas. Además, se definen las herramientas para modelar el proceso, compilar aplicaciones y emular el sistema. Se establecen los estándares que se utilizará para describir el proceso.

Capítulo 2. Diseño del Sistema.

Se realiza la captura de requisitos del sistema y basado en estos, se crea un flujo de actividades para describir el proceso de funcionamiento del sistema. Además, se define la Arquitectura del sistema y se mencionan los principales componentes de esta.

Capítulo 3. Construcción del Sistema.

Se describe el proceso de construcción del sistema en conjunto con las técnicas empleadas.

Capítulo 4. Pruebas al Sistema.

Se describen las pruebas realizadas al sistema, para asegurar el correcto funcionamiento de sus funcionalidades.

CAPITULO I: HISTORIA DEL ARTE DE LAS DISTRIBUCIONES GNU/LINUX PARA CLIENTE LIGERO.

Este capítulo contendrá los principales conceptos necesarios para el entendimiento de las etapas por donde transitará el proceso de construcción. Se definirá el término de Arquitectura de Computadora y Cliente Ligero, y a su vez, algunas características de interés de estudio. Además, se abordará sobre la metodología de desarrollo a utilizar y los métodos de descripción de procesos utilizados.

1.1 Arquitectura de Computadora.

“Arquitectura de Computadora es la ciencia y el arte de seleccionar e interconectar componentes de hardware para crear computadoras que cumplen con el rendimiento funcional y los objetivos de coste.”(Hower et al. 2010) Describe y diseña varias partes de la computadora, principalmente la *Unidad Central de Procesamiento* (CPU, por sus siglas en Inglés).

La Arquitectura de una computadora explica la situación de sus componentes. Permite determinar las posibilidades de que un sistema informático, con una determinada configuración, pueda realizar las operaciones para las que se va a utilizar.

Entre los fabricantes de computadoras más notables se encuentran Intel y ARM. Los cuales, aunque con puntos de vistas diferentes han logrado dominar el mercado mundial en la producción de procesadores.

Intel fue la primera compañía en fabricar un procesador alrededor de los años 70. Convirtiéndose en los pioneros en el campo de producción de estos productos, contando actualmente con el liderazgo mundial del mercado.

El desarrollo de los procesadores de Intel ha transitado por un proceso de actualización y mejora constante. Desde su surgimiento han buscado obtener el procesador más potente en cuanto a velocidad de procesamiento de datos. Por esta razón son los más usados en la producción de computadoras.

Por su parte ARM, “fue creado alrededor de los años 80 por el *Acorn Computer Group*, como el primer procesador *RISC* de gran impacto en el mundo. Partiendo de la filosofía de que la eficiencia está en hacer las cosas menos complicadas. Gracias a su diseño sencillo, el ARM tiene relativamente pocos componentes en el chip, por lo que no alcanza altas temperaturas y tiene bajos requerimientos de energía. Lo anterior lo ha hecho candidato perfecto para el mercado de sistemas embebidos que van desde un celular hasta una lavadora.” (Parra 2001)

Desde su creación ARM ha producido un gran número de procesadores. Alrededor del año 1998 se extiende la familia de los procesadores ARM9. Punto de partida para la

producción del procesador S3C2440 de Samsung basado en el ARM920, “este procesador fue diseñado para proporcionar dispositivos portátiles y aplicaciones generales con bajo consumo de energía y costo efectivo.” (Samsung Electronics co 2004) Basado en este procesador de Samsung surge la MBTI2440 como la computadora en una tarjeta diseñada por el ICID.

1.2 La MBTI2440: Computadora en una Tarjeta.

“MBTI2440 fue diseñada con el objetivo de obtener una tarjeta de procesamiento de bajo costo y pequeño tamaño, bajo consumo de potencia, con gran variedad de dispositivos de interfaz y elevada conectividad y una capacidad de procesamiento adecuada a los requerimientos de los procesadores empleados en equipos médicos de complejidad media desarrollados por el ICID, como son monitores de parámetros vitales, desfibriladores y electrocardiogramas entre otros.”(ICID)

La MBTI2440 no posee conectores estándar de interconexión con los periféricos y dispositivos de interfaz. Por tal motivo, los usuarios para poder utilizarla debe hacerlo a través de una tarjeta base, en la cual se colocarán los conectores requeridos.

Características principales de la MBTI2440:

1. Velocidad del Procesador: hasta 400 MHz.
2. Memoria
 - SDRAM: 128MB
 - NOR FLASH: 1MB

- NAND FLASH: 64MB
- 3. Interfaz Ethernet 10 Mbit/s.
- 4. Entrada y salida de audio estéreo.
- 5. Interfaz para LCD (STN o TFT) resolución hasta 1024 x 768 con 24 bpp.

Estas características convierten a la MBTI2440 en la candidata para la confección del *Cliente Ligero Cubano*. Proyecto en el cual se integran las instituciones GEDEME, ICID y UCI.

1.3 Cliente Ligero.

El término Cliente Ligero surge a finales de los años 90, definiéndolos como máquinas poco potentes, sin medios de almacenamiento, que sustituirían a los ordenadores de sobremesa normales. Estos son computadoras clientes que dependen de un servidor central para las tareas de procesamiento, todos los programas que en un momento determinado tiene un Cliente Ligero en ejecución residen en la memoria del servidor central, en este punto, la memoria y CPU del servidor central son compartidas entre todos los Clientes Ligeros conectados. Estas computadoras se enfocan principalmente en transportar la entrada y salida entre el usuario y el servidor.

Ventajas de los Clientes Ligeros.

- Menores costos de administración.
- Información centralizada.
- Más fácil de asegurar.

- Más bajos costos de hardware.
- Menos costos por mantenimiento.
- Menos consumo de energía eléctrica.
- Menor ruido.
- Más fácil gerencia de fallas.
- Uso más eficiente de los recursos computacionales.

Sistema Operativo para Cliente Ligerero.

El Sistema Operativo (SO) instalado en los Clientes Ligeros realiza la función de interfaz de usuario, posee solo algunas aplicaciones frecuentemente usadas y un sistema enfocado en la red. Entre los principales programas instalados en este Sistema están los protocolos de comunicación cliente - servidor, aplicaciones necesarias para configurar la red, sonido y vídeo.

El SO instalado en los Clientes Ligeros, puede llegar a ser muy específico, dependiendo de las necesidades, recursos y uso real donde se utilice esta tecnología. Actualmente el exponente principal de SO para Cliente Ligerero lo posee Linux Terminal Server Project (LTSP).

Linux Terminal Server Project (LTSP).

LTSP es un software que brinda soporte para Cliente Ligerero en Servidores Linux. Posee dos líneas principales de desarrollo: Servidor de Cliente Ligerero y Terminal de Cliente Ligerero. LTSP posibilita la creación automática de un Sistema para ser instalado

en las Terminales de Cliente Ligero, este sistema es el encargado de realizar la conexión entre la Terminal de Cliente Ligero y el Servidor de Cliente Ligero.

LTSP como característica necesaria para construir el Sistema a instalar en los Clientes Ligeros, necesita que la arquitectura de computadora del Cliente sea soportada por el ordenador donde se va a construir el Sistema. En el caso particular de esta investigación se pretende construir el Sistema para instalar en los Clientes Ligeros, con la inconveniente que la arquitectura de los Clientes Ligeros y el ordenador donde se van a construir el Sistema son diferentes e incompatibles, por lo que la solución brindada por LTSP no es factible.

Es objetivo de este trabajo llegar a construir un Sistema para la arquitectura de computadora ARM, tomando como base el Sistema que construye LTSP. Las etapas principales por las que transita LTSP para construir el sistema para cliente ligero son:

1. Construye un Sistema Base utilizando la herramienta *debootstrap*.
2. Instala y configura las herramientas utilizadas para establecer la conexión del Cliente Ligero al Servidor de Cliente Ligero.

En el caso de estudio, de manera general, el proceso de construcción del sistema, transitará por las mismas etapas de LTSP. La diferencia estará en la forma en que se construye el sistema base y se instalan las herramientas para la conexión.

1.4 Método para la construcción del Sistema.

Para la construcción del sistema se realizarán 5 actividades principales:

1. Definir las aplicaciones a instalar.
2. Descargar las aplicaciones.
3. Desempaquetar las aplicaciones.
4. Configurar el sistema base.
5. Instalar todas las aplicaciones.

El proceso de instalar las aplicaciones del sistema base y las herramientas para la conexión entre el cliente y el servidor, se puede realizar de dos métodos distintos:

1. A partir del *paquete de código binario* de la aplicación a instalar.
2. A partir del *paquete de código fuente* de la aplicación a instalar.

Para instalar utilizando el primer método, mediante la herramienta *dpkg*, las aplicaciones son instaladas y configuradas en el sistema. Al utilizar el segundo método, se compila - mediante la técnica de compilación cruzada - el código fuente y luego se copian los ficheros binarios generados para el sistema en construcción.

Compilación

Compilación se denomina al proceso de transformar el código de una aplicación escrito en algún lenguaje de programación a código binario que pueda ser entendido por la computadora.

Compilación cruzada

Se define como compilación cruzada como el proceso de compilar una aplicación sobre una Arquitectura de Computadora generando código binario para otra Arquitectura de Computadora.

1.5 Metodología de desarrollo.

Las metodologías de desarrollo se pueden definir como guías para la construcción de un producto. Tienen como objetivo aumentar la calidad del producto y definen un conjunto de etapas por las cuales debe transitar el proceso de desarrollo. Estas definen el qué, cómo y cuándo deben desarrollarse cada una de las actividades. A su vez, no existe metodología universal, estas deben ser adaptas a las peculiaridades de cada proyecto donde se aplique.

Entre las metodologías de desarrollo existentes, se encuentran de interés de estudio:

- Proceso Unificado de Desarrollo de Software (RUP, por sus siglas en Inglés).
- Desarrollo Basado en Funcionalidades (FDD, por sus siglas en Inglés).
- Programación Extrema (XP, por sus siglas en Inglés).

Rational Unified Process (RUP)

Proceso Unificado de Rational (RUP, por sus siglas en ingles), es una de las metodologías más amplia de las existentes actualmente. Es el resultado de varios años de desarrollo, en la cual se han unificado varias técnicas y experiencias de otras metodologías. No se limita solo a los proyectos de software informático, de manera

que esta diseñada para adaptarse a la mayoría de los proyectos de desarrollo. Se define como una metodología robusta y principalmente usada en grandes proyectos.

RUP se caracteriza por ser:

- Dirigido por casos de uso

“Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.”(Kruchten 2000)

- Centrado en la arquitectura.

“La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.”(Kruchten 2000)

- Iterativo e incremental.

“El trabajo se divide en partes pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y Arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos flujos de trabajos fundamentales) del cual se obtiene un incremento

que produce crecimiento en el producto.”(Jacoboson, I., Booch, G., y Rumbaugh, J. 2000)

Esta metodología divide el proceso de desarrollo en 4 fases:

- Inicio
- Elaboración
- Construcción
- Transición

En base a las particularidades del producto informático a desarrollar, el proceso de desarrollo transitará por las 3 tres primeras fases en que está dividido RUP. Describiendo cada una de ellas como sigue:

- Inicio: Se define el alcance y los requerimientos del sistema.
- Elaboración: Se define la arquitectura del sistema.
- Construcción: Se construye y prueba el sistema.

1.6 Modelado de Proceso.

Para el modelado de procesos se utilizará Lenguaje de Modelado Unificado (UML, por sus siglas en Inglés) y la Notación de Modelado de Procesos de Negocio (BPMN, por sus siglas en Inglés).

“UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema software. Utilizar herramientas de modelado visual facilita la gestión de dichos modelos, permitiendo ocultar o exponer detalles cuando sea

necesario. El modelado visual también ayuda a mantener la consistencia entre los artefactos del sistema: requisitos, diseños e implementaciones. En resumen, el modelado visual ayuda a mejorar la capacidad del equipo para gestionar la complejidad del software. "(Universidad Politécnica de Valencia.)

"La Notación BPMN define un diagrama de procesos de negocio que se utiliza para modelar procesos de negocio, sub-procesos y tareas que lo componen. Intenta rescatar las mejores ideas de las notaciones anteriormente existentes y consolidarlas en una sola. Además es un estándar internacional de modelado de procesos aceptado por la comunidad, rico gráficamente y fácil de comprender."(Machín y Corso 2010)

1.7 Métodos de descripción del proceso de construcción.

"Para la descripción del proceso se utilizarán las convenciones establecidas en el libro Linux From Scratch (LFS)"(Machín y Corso 2010) y otras adicionales.

1. Los textos que aparecen en un rectángulo gris y la letra negrita se pueden teclear exactamente como aparecen, a menos que se indique lo contrario en el texto subyacente. También se utiliza en las secciones explicativas para identificar el comando al que se hace referencia.
2. Los textos escritos en letras cursivas se utilizan para señalar aspectos importantes.
3. Los hipervínculos son direcciones de descarga.
4. Los textos que se encuentran entre <> se utilizan para encapsular texto que no debe ser escrito tal y como aparece.

5. Las notas aclaratorias se presentan con un rectángulo en amarillo.
6. Las líneas de comando a ejecutar que comiencen “#” significa que hay que ejecutarlas como superusuario.
7. Las líneas de comando a ejecutar que comiencen “\$” significa que hay que ejecutarlas como usuario normal del sistema.

1.8 Conclusiones Parciales

Con el objetivo de construir una distribución de GNU/Linux basada en Nova para Cliente Ligeró, y del buen entendimiento de los lectores de esta investigación, se analizaron en este capítulo algunos conceptos de gran importancia para el desarrollo del producto. De esta forma se describió el estado actual de las distribuciones GNU/Linux para Cliente Ligeró y algunos conceptos relacionados con la Arquitectura de Computadora. Se definió como metodología de desarrollo RUP, así como UML y BPMN para el modelado de procesos. Se acordaron además, los métodos para la descripción de procesos a utilizar, los cuales serán regidos en gran medida por los explicados en el libro *Linux From Scratch*.

CAPÍTULO 2: DISEÑO DEL SISTEMA.

En el presente capítulo se definirá la Arquitectura del sistema, enfocada en dar cumplimiento a los requisitos del mismo. De igual forma, se describirá de forma general el proceso del funcionamiento del sistema, especificando aspectos importantes pertenecientes a los subprocesos definidos.

2.1 Requisitos del Sistema.

Se construirá una distribución GNU/Linux basada en Nova, para ser instalada en la computadora en un tarjeta diseñada por el ICID. Esta distribución deberá cumplir con los siguientes requisitos:

Requisitos funcionales

1. Reconocer los dispositivos.
2. Iniciar y configurar la red.
3. Permitir la conexión a un servidor de cliente ligero mediante el protocolo *RDP*.
4. Permitir la conexión a un servidor de cliente ligero mediante el protocolo *VNC*.
5. Permitir la conexión a un servidor de cliente ligero mediante el protocolo *XDMCP*.
6. Importar audio desde el servidor de cliente ligero a través de la red.

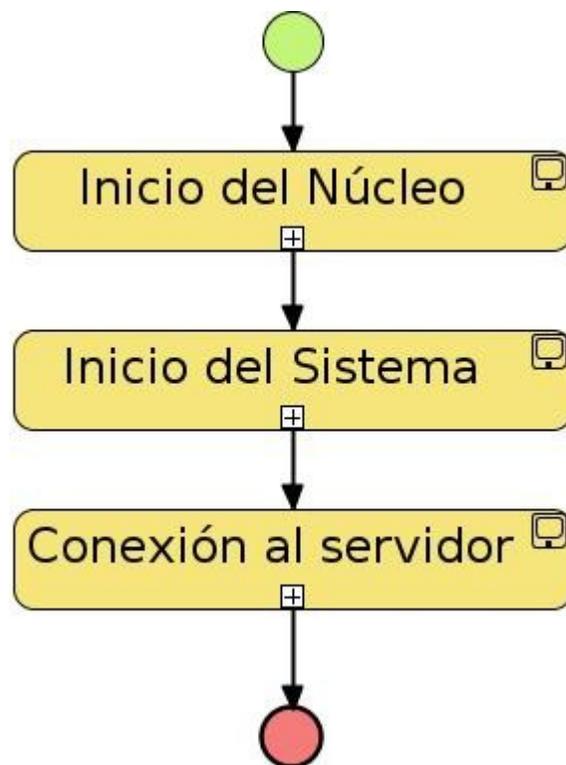
Requisitos no funcionales

- El sistema tiene que estar compilado para la arquitectura ARM versión 4.

- El sistema debe funcionar sobre la computadora en una tarjeta, diseñada por el ICID.
- Debe permitir a usuarios finales la interacción con las estaciones de trabajo de manera sencilla y transparente.

2.2 Diagrama de Proceso del Funcionamiento del Sistema.

El proceso completo del funcionamiento del sistema, desde su arranque hasta lograr los objetivos finales, transita por tres subprocessos: Inicio del Núcleo, Inicio del Sistema y Conexión al Servidor, como se muestra en la siguiente figura.



Cada uno de estos subprocessos realizan una serie de actividades, de las cuales las más importante se mencionan a continuación.

Inicio del Núcleo

El núcleo del sistema, es el primer elemento que se inicia al encender al computadora. Su funcionamiento no puede verse como un proceso aislado que tiene un inicio y un fin, sino como un elemento que se está ejecutando durante todo el tiempo que el sistema está en funcionamiento. Entre sus principales tareas están la administración de memoria y tiempo del procesador a los programas y procesos que se estén ejecutando en el sistema, reconocer los dispositivos físicos e iniciar sus controladores, lo cual permite la comunicación entre el software y los periféricos de manera transparente, brindando una serie de herramientas para esto.

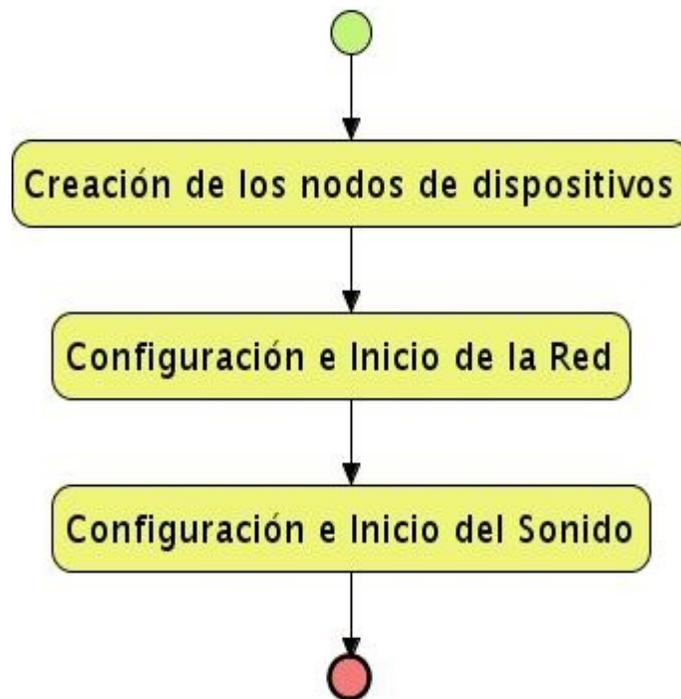
Diagrama de Funcionamiento:



Inicio del Sistema

Una vez terminado el inicio del núcleo, este le cede el control del sistema al primer proceso, el proceso INIT. Este proceso es el encargado de iniciar el resto de los procesos del sistema, y se conoce como el padre de todos los procesos. Entre sus principales funciones están: la creación de los nodos de dispositivos e iniciar los controladores de estos, mediante los cuales los programas pueden interactuar con los dispositivos físicos de la computadora; chequea y monta los sistemas de ficheros definidos en el sistema; configura e inicia los dispositivos de red y audio.

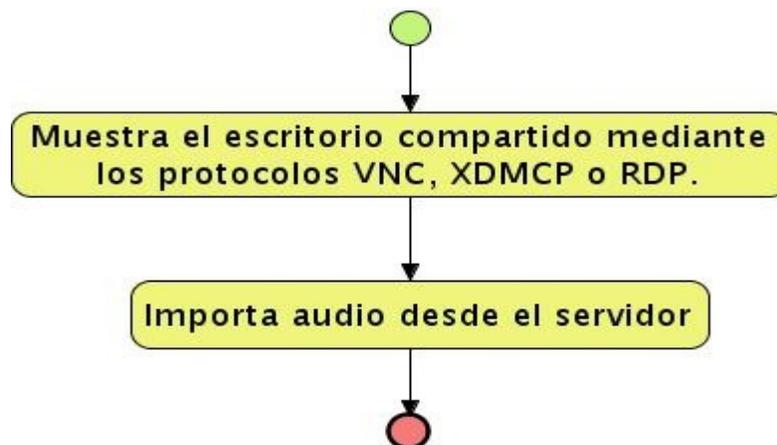
Diagrama de Funcionamiento:



Conexión al Servidor

Al terminar de iniciar las funcionalidades básicas del sistema, se configuran e inician un conjunto de aplicaciones que darán soporte a la funcionalidad de cliente ligero al sistema. Según el protocolo de comunicación configurado por el administrador del sistema, se creará una conexión entre el servidor de cliente ligero y la terminal cliente. Mediante esta conexión y según las particularidades de cada protocolo, se procede al transporte de datos entre ambos lados. En cuanto a la interfaz gráfica, desde el servidor se envían los datos para que se muestre en la terminal cliente el escritorio compartido en el servidor. A su vez, son enviados los paquetes de sonido desde el servidor a la terminal cliente, mediante la utilización de una herramienta que funcionará como servidor de sonido, y creará un puente de comunicación entre las dos computadoras. La terminal cliente por su parte, enviará los eventos realizados por los periféricos (mouse y teclado) al servidor.

Diagrama de funcionamiento:

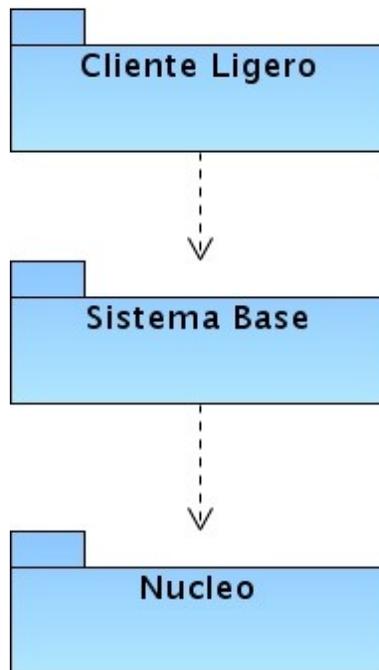


2.3 Arquitectura del Sistema

El sistema estará definido por una arquitectura en capas, la cual no se rige por ningún patrón arquitectónico, donde las capas se pueden definir como niveles, los cuales se listan a continuación:

1. Núcleo
2. Sistema base
3. Cliente Ligero

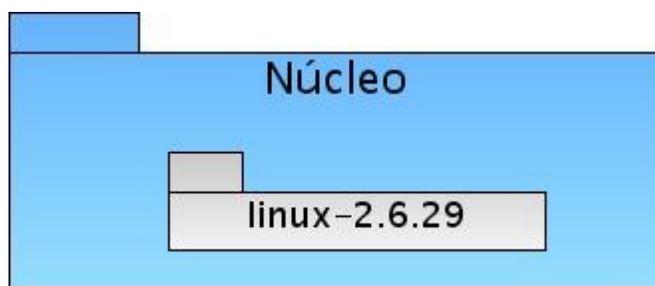
Diagrama de la Arquitectura del Sistema



2.4 Núcleo

En el *núcleo* del sistema se encuentran los controladores de los dispositivos y permite la comunicación entre el sistema y los elementos físicos de la computadora. Esta capa está compuesta por el núcleo de Linux versión 2.6.29.

Diagrama del Núcleo



2.5 Sistema Base

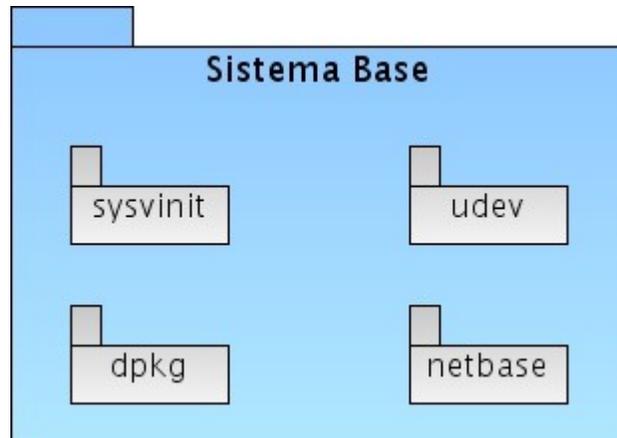
El sistema base de la distribución contendrá los paquetes esenciales del sistema. Estos paquetes comprenden los necesario para el arranque del sistema, detección de dispositivos, configuración de la red, trabajo básico con ficheros, entre otros.

El listado total de los elementos que componen esta capa se listan en el **ANEXO I**. A continuación se describen los elementos principales:

- Sysvinit: Se encarga del inicio del sistema.
- Udev: Se encarga de crear los nodos de los dispositivos físicos del sistema.
- Dpkg: Aplicación utilizada para instalar las aplicaciones del sistema.

- Netbase: Se encarga del arranque y configuración de la red.

Diagrama del Sistema Base



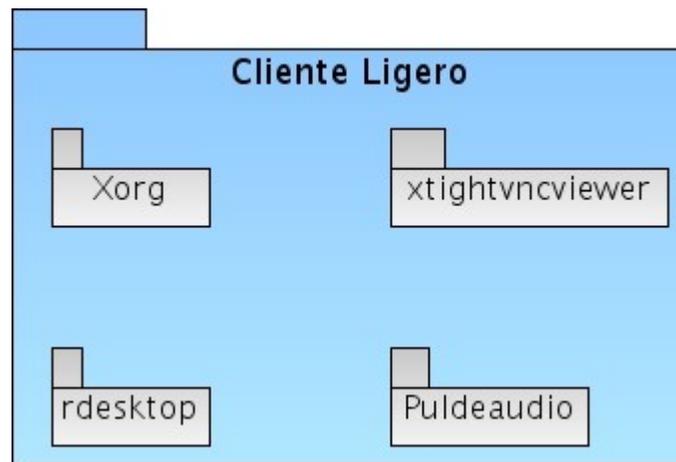
2.6 Cliente Ligero

En la capa de cliente ligero se encuentran las herramientas necesarias para crear la conexión entre el cliente ligero y el servidor de cliente ligero. Estas herramientas están compuestas principalmente por el Servidor Gráfico y los protocolos de comunicación cliente-servidor.

El listado total de los elementos que componen esta capa se listan en el **ANEXO II**. Los elementos principales se describen a continuación:

- Xorg: Servidor gráfico que permite visualizar en el cliente ligero las aplicaciones ejecutadas en el servidor. Permite además realizar la conexión mediante el protocolo XDMCP.
- Rdesktop: Permite realizar la conexión a un servidor de cliente ligero mediante el protocolo RDP.
- Xtightvncviewer: Permite realizar la conexión a un servidor de cliente ligero mediante el protocolo VNC.
- Pulseaudio: Permite el transporte de audio desde el servidor para el cliente ligero.

Diagrama del Cliente Ligero



2.7 Conclusiones Parciales

Fue de interés de estudio en este capítulo, la captura y análisis de los requisitos del sistema, tomando como críticos los relacionados con la conexión al servidor de cliente

ligero. Además, se describió el proceso de funcionamiento del sistema en general, especificando las actividades que se realizan en cada uno de los subprocesos. Se definió una arquitectura en capas para el sistema, así como los principales elementos que se encontrarán en cada una de esas capas.

CAPÍTULO 3: CONSTRUCCIÓN DEL SISTEMA.

En el presente capítulo se definirá y configurará el sistema anfitrión. Además, se describirá el proceso de construcción de cada una de las capas definidas en la arquitectura.

3.1 Definir y configurar Sistema Anfitrión

Para llevar a cabo el proceso de construcción del sistema se utilizará como sistema anfitrión la distribución cubana de GNU/Linux Nova versión 3.0. Con motivo de tener organizado el proceso de construcción del sistema se crean un conjunto de directorios.

```
# mkdir /trabajo
# chmod 777 /trabajo
$ cd /trabajo
$ mkdir herramientas kernel paquetes sistema
$ cd paquetes
$ mkdir cliente_ligero sistema_base
```

trabajo: Directorio dentro del cual se guardarán y realizarán todos los cambios durante el proceso de construcción del sistema.

herramientas: Directorio en el cual se guardarán e instalarán algunas herramientas necesarias.

kernel: Directorio que guardará el paquete de código fuente de los núcleos que se utilizarán.

paquetes: Se guardarán todos los paquetes que se instalarán en el sistema.

cliente_ligero: Para mejor organización de los paquetes, se dividirán en dos grandes grupos. En este directorio se guardarán los paquetes necesarios para las funcionalidades de cliente ligero.

sistema_base: En este directorio se guardarán los paquetes necesarios para construir el sistema base.

sistema: En este directorio se creará y configurará el sistema a construir.

Herramientas

Durante el proceso de construcción y configuración del sistema se necesitarán algunas herramientas. Estas permitirán emular la arquitectura ARM en el sistema anfitrión, realizar la compilación cruzada para la arquitectura ARM, copiar los resultados hacia la computadora real, entre otras funciones.

Las herramientas utilizadas se listan a continuación. El proceso de instalación y configuración de cada una se encuentra detallado en el **ANEXO III**.

- Qemu
- NFS Server
- Red virtual
- Minicom
- CodeSourcery

3.2 Núcleo

Durante el proceso de construcción se utilizarán dos versiones del núcleo de linux: 2.6.33 y 2.6.29. La primera versión se configurará y compilará para ser utilizada para emular el sistema. La versión 2.6.29, contiene los controladores configurados de la computadora real, se configurará para añadirle nuevos soportes y se compilará, este núcleo es el que se grabará en la computadora real, la versión utilizada no es la oficial, sino una adaptación hecha por el ICID para que funcionaran los controladores en la tarjeta construida.

Descargar ambos códigos fuentes en el directorio **kernel**.

Configurando y compilando 2.6.33:

```
$ cd /trabajo/kernel/  
$ tar -xjf linux-2.6.33.tar.bz2  
$ cd linux-2.6.33
```

Editar el fichero Makefile, y aplicar los siguientes cambios en las líneas correspondientes:

```
ARCH = arm  
CROSS_COMPILE = arm-none-linux-gnueabi-  
CC = $(CROSS_COMPILE)gcc -march=armv4t
```

De esta forma se indica la arquitectura para la cual se compilará el núcleo y el nombre de la herramienta de compilación cruzada a usar. La opción **-march=armv4t**, se especifica para generar código binario para la versión 4 de la arquitectura de computadora ARM.

```
$ make mrproper
$ make versalite_defconfig
$ make menuconfig
```

Se activan un conjunto de *módulos del núcleo*, para el correcto funcionamiento de la red y soporte para código binario compilado para la versión 4 de ARM.

File systems -> Pseudo filesystems

[*] Virtual memory file system support (former shm fs)

Device Drivers -> Generic Driver Options

[*] Create a kernel maintained /dev tmpfs (EXPERIMENTAL)

[*] Automount devtmpfs at /dev

Kernel Features

[*] Use the ARM EABI to compile the kernel

[*] Allow old ABI binaries to run with this kernel (EXPERIMENTAL)

Networking support -> Networking options

[*] IP: DHCP support

File systems

<*> Kernel automounter support

Compilar y copiar la imagen del núcleo para **trabajo**:

```
$ make all  
$ cp arch/arm/boot/zImage /trabajo/zImageV
```

Se crea el comprimido con los módulos y la imagen del núcleo.

```
$ make targz-pkg
```

Configurando y compilando 2.6.29:

```
$ cd /trabajo/kernel/  
$ tar -xjf linux-2.29-last.tar.bz2  
$ cd linux-2.6.29/  
$ make menuconfig
```

Se activan un conjunto de módulos del núcleo, para el correcto funcionamiento de la red, audio, soporte para código binario compilado para la versión 4 de ARM, soporte para sistema de ficheros *ext3*, *ext4*, *nfs*, y la tarjeta SD de la computadora real.

General setup

[] Configure standard kernel features (for small systems)

Kernel Features

[*] Use the ARM EABI to compile the kernel

[*] Allow old ABI binaries to run with this kernel (EXPERIMENTAL)

Networking support -> Networking options

[*] IP: kernel level autoconfiguration

[*] IP: DHCP support

[*] IP: BOOTP support

Device Drivers -> Network device support -> Ethernet (10 or 100Mbit)

<*> Generic Media Independent Interface device support

<*> CS8900 support

Device Drivers

<*> Sound card support

Device Drivers -> Sound card support

<*> Advanced Linux Sound Architecture

Device Drivers -> Sound card support -> Advanced Linux Sound Architecture

<*> ALSA for SoC audio support

Device Drivers -> Sound card support -> Advanced Linux Sound Architecture -> ALSA for SoC audio support

<*> SoC Audio for the Samsung S3CXXXX chips

<*> SoC I2S Audio support UDA134X wired to a S3C24XX

<*> Build all ASoC CODEC drivers

File systems

<*> Ext3 journalling file system support

[*] Ext3 extended attributes (NEW)

<M> The Extended 4 (ext4) filesystem

[*] Ext4 extended attributes

<*> Kernel automounter support

[*] Network File Systems

File systems -> Network File Systems

<*> NFS client support

[*] NFS client support for NFS version 3

[*] Root file system on NFS

Boot options

root=/dev/mmcblk0p1 rw console=ttySAC0,115200 rootwait init=/sbin/init

Salir y guardar los cambios.

Editar Makefile y realizar los siguientes cambios en las líneas correspondientes:

```
ARCH           = arm  
CROSS_COMPILE = arm-none-linux-gnueabi-  
CC             = $(CROSS_COMPILE)gcc -march=armv4t
```

Para poder ejecutar la interfaz gráfica se debe modificar el código el siguiente fichero en el núcleo:

[linux-2.6.29/drivers/video/fbmem.c](#)

Buscar donde aparezca el siguiente código:

```
case FBIOPUT_VSCREENINFO:
```

Añadir justo debajo:

```
return 0;
```

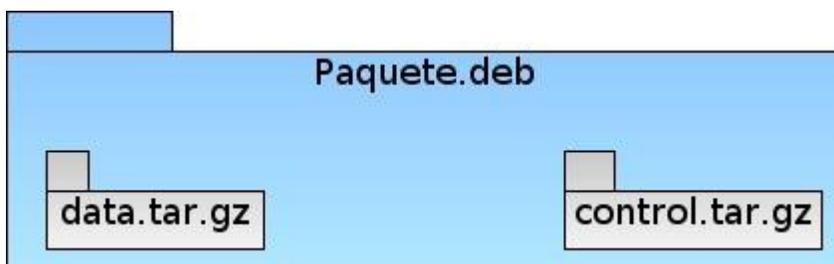
Compilar:

```
$ make all  
$ make targz-pkg  
$ cp arch/arm/boot/zImage /trabajo/zImageR
```

3.3 Sistema Base

Se procederá a construir la base del sistema. Se descargan todos los paquetes listados en el **ANEXO I**.

Para el entendimiento de los pasos siguientes, es importante aclarar que un paquete **.deb** está estructurado de la siguiente forma:



En el comprimido *data.tar.gz* contiene los directorios y ficheros a copiar en el sistema. Por su parte, en *control.tar.gz* se encuentran los ficheros de configuración del paquete, que se ejecutan durante el proceso de pre-instalación y post-instalación del mismo.

Se creará una estructura de directorios, copiando el contenido que cada paquete tiene en *data.tar.gz* hacia el directorio **sistema**. Se realizará esto utilizando los paquetes que se encuentran en el directorio **sistema_base**.

Dentro del directorio **/trabajo/paquetes/sistema_base/** se ejecuta el siguiente comando para todos los paquetes listados en el **ANEXO I**.

```
$ ar p <nombre_paquete>.deb data.tar.gz | tar zx -C /trabajo/sistema/
```

Se sustituye *<nombre_paquete>* por el nombre de cada paquete a procesar, uno a la vez. El proceso completo de desempaquetado de todos los paquetes se encuentra en el **ANEXO IV**.

Una vez realizadas las operaciones anteriores, el directorio **sistema** posee todos los directorios y ficheros del sistema base, pero no están configurados y algunas aplicaciones no funcionarán correctamente. Para suprimir el problema anterior, se emulará el sistema usando la herramienta **qemu-system-arm**, para esto se realizan las siguientes operaciones:

```
$ cd /trabajo/sistema/  
$ cp -r /trabajo/paquetes/sistema_base/
```

Se copian los paquetes de la base del sistema dentro del directorio **sistema**, de esta forma podrán ser utilizados cuando se emule el sistema. Estos paquetes se procederán a instalar dentro del emulador usando la herramienta *dpkg*, así quedarán bien configurados, propiciando estabilidad al sistema.

Se crean un conjunto de ficheros de configuración, para el correcto funcionamiento de la herramienta *dpkg*. Estos ficheros controlan la configuración de la red, el nombre de la computadora, entre otras funciones. Estos ficheros se listan a continuación, el proceso de creación y edición de cada uno de ellos se encuentra en el **ANEXO V**.

Ficheros creados:

- sistema/var/lib/dpkg/status
- sistema/var/lib/dpkg/available
- sistema/var/lib/dpkg/diversions
- sistema/etc/hostname
- sistema/etc/fstab
- sistema/etc/resolv.conf
- sistema/etc/network/interfaces

Se copian los módulos del núcleo para el sistema de ficheros creado:

```
$ tar xzf /trabajo/kernel/linux-2.6.33/linux-2.6.33.tar.gz -C /trabajo/sistema/
```

Se procede a emular el sistema:

```
$ cd /trabajo/  
$ sudo ./red  
$ sudo qemu-system-arm -M versatilepb -m 256M -kernel zImageV -append  
"root=/dev/nfs nfsroot=192.168.1.1:/trabajo/sistema rw ip=192.168.1.101  
init=/bin/bash" -net nic -net tap,ifname=tap1,script=no
```

A partir de este momento los comandos a ejecutar serán dentro del emulador, hasta que se indique lo contrario. Se procederá a instalar todos los paquetes del sistema base.

```
# dpkg -i <nombre_paquete>.deb
```

Se sustituye *<nombre_paquete>* por el nombre de cada uno de los paquetes listados en el **ANEXO I**, uno a la vez y necesariamente en el orden en que aparecen. Al instalarlos en el orden estricto en que aparecen se asegura que ningún paquete se instale antes de estar instaladas todas sus *dependencias*. El proceso completo de instalación se encuentra en el **ANEXO VI**. Una vez culminado el proceso de instalación de cada paquete se cierra la ventana del emulador.

En el directorio **sistema**, se encuentran instalados y configurados los paquetes necesarios del sistema base. Se volverá iniciar el emulador pero ejecutando como primer proceso **/sbin/init**, para verificar que todo está funcionando correctamente. A partir de este punto, cada vez que se refiera a emular el sistema se utilizan los comandos siguientes:

```
$ sudo ./red
```

```
$ sudo qemu-system-arm -M versatilepb -m 256M -kernel zImageV -append  
"root=/dev/nfs nfsroot=192.168.1.1:/trabajo/sistema rw ip=192.168.1.101  
init=/sbin/init" -net nic -net tap,ifname=tap1,script=no
```

Para evitar tener que cambiar la contraseña del usuario **root** cada vez que se emule el sistema, se aplican los siguientes cambios dentro del emulador. Al finalizar se cierra la ventana del emulador.

Ejecutar dentro del emulador:

```
# date --set="2011-02-7 16:30"  
# passwd
```

El primer comando modifica la fecha del sistema. Con el segundo comando se le asigna una nueva contraseña al usuario **root**.

En este momento ya está listo el sistema base. Se eliminan los directorios **lib/modules** y **boot/** del sistema construido, para copiar los generados con el núcleo 2.6.29, ya que es con el que se trabajará en el hardware real. Se eliminan además, los paquetes de la base del sistema que ya fueron instalados.

```
$ rm -r /trabajo/sistema/lib/modules/  
$ rm -r /trabajo/sistema/boot/  
$ tar xzf /trabajo/kernel/linux-2.6.29/linux-2.6.29-MBTI2440.tar.gz -C  
/trabajo/sistema/  
$ rm -r /trabajo/sistema/sistema_base/
```

Se crea un comprimido con el sistema base, como salva del sistema.

```
$ cd /trabajo/sistema/  
$ sudo tar czf ../sistemabase.tar.gz .
```

3.4 Cliente Ligero

A partir de este momento se empezará la construcción del sistema con las funcionalidades de cliente ligero. Se descargan todos los paquetes listados en el **ANEXO II** en el directorio **cliente_ligero**, correspondiente al listado de los paquetes para brindar las funcionalidades de cliente ligero. Se copiarán los paquetes del directorio **cliente_ligero** para el sistema, para poder instalarlos dentro del emulador.

```
$ cd /trabajo/  
$ cp -r paquetes/cliente_ligero/ sistema/
```

Se reemplazarán los directorios **lib/modules** y **boot/** del sistema, con los del núcleo 2.6.33 que está compilado para utilizarlo en el emulador.

```
$ rm -r sistema/lib/modules/  
$ rm -r sistema/boot/  
$ tar xzf kernel/linux-2.6.33/linux-2.6.33.tar.gz -C sistema/
```

Se emula el sistema para instalar los paquetes pertenecientes al grupo de cliente ligero. Se instalarán cada uno de estos paquetes utilizando la herramienta *dpkg*, de la

misma forma que se instalaron los paquetes del sistema base, ejecutando dentro del emulador el siguiente comando.

```
# dpkg -i <nombre_paquete>.deb
```

Se sustituye *<nombre_paquete>* por el nombre de cada paquete listado en el **ANEXO II**, este procedimiento se realizará a cada paquete y estrictamente en el orden en que son listados, para evitar problemas de dependencias. El proceso de instalación completo esta descrito en el **ANEXO VII**.

Finalizada la instalación de los paquetes se procede a realizar algunas configuraciones:

Para indicar el *controlador* gráfico que debe usar el servidor gráfico se modifica el fichero */etc/X11/xorg.conf*, en la sección "Device" añadir la siguiente línea:

```
Driver "fbdev"
```

En este punto ya está el sistema completo. Se eliminan los paquetes, y se reemplaza los directorios **lib/modules** y **boot/** por los del núcleo 2.6.29 y se comprime como salva.

Ejecutar en el sistema anfitrión:

```
$ rm -r sistema/lib/modules/  
$ rm -r sistema/boot/  
$ tar xzf kernel/linux-2.6.29/linux-2.6.29-MBTI2440.tar.gz -C sistema/  
$ rm -r sistema/cliente_ligero/  
$ cd sistema/  
$ sudo tar czf ../sistemacompleto.tar.gz .
```

3.5 Grabar el sistema para la computadora real

Al tener todo listo se procederá a copiar el sistema en la tarjeta SD de la computadora real. Usando un lector de tarjetas o un adaptador, se conecta la tarjeta SD a la computadora anfitrión. En el caso de estudio el dispositivo donde se instaló fue en **/dev/sdc1**, se monta el dispositivo, se elimina todo el contenido que tenga, se copia todo el contenido del sistema y se desmonta el dispositivo.

```
$ cd /trabajo/  
$ mkdir tem  
$ sudo mount /dev/sdc1 tem/  
$ cd tem  
$ sudo rm -r *  
$ sudo tar xzf /trabajo/sistemacompleto.tar.gz .  
$ sudo umount tem/
```

Ya está grabado el contenido del sistema en la tarjeta SD. Para copiar el núcleo para la memoria NAND de la computadora real, se conecta la computadora real con la

computadora anfitrión mediante un cable serial, y utilizando la herramienta **minicom**, se procede a realizar esta operación. Este proceso se describe en el **ANEXO VIII**.

3.6 Conclusiones Parciales

Este capítulo puede considerarse como el de mayor importancia, ya que es en el mismo donde se describe el proceso de construcción del sistema. Para esto se utilizaron los métodos definidos en el capítulo 1. El proceso de construcción transitó de forma general por cinco etapas. La primera fue la selección y configuración del sistema anfitrión, seguida de tres etapas en las cuales se describe el proceso de construcción de cada una de las capas definidas en la arquitectura del sistema, y por último, se explica el proceso de instalar el sistema construido en la computadora real.

CAPÍTULO 4: PRUEBAS AL SISTEMA.

Una vez concluida la construcción de la distribución para cliente ligero, con soporte para la arquitectura de computadora ARM, se le aplicarán un conjunto de pruebas para asegurar el correcto funcionamiento de la solución. Las pruebas de software permiten encontrar errores o comportamientos inadecuados en los sistemas, de esta forma se prueba la integridad y el correcto funcionamiento del sistema.

4.1 Pruebas a Funcionalidades

Inicio del sistema

Escenario	Descripción	Respuesta del sistema	Flujo central
Iniciar la computadora	Iniciar el sistema operativo mediante el botón de inicio de la computadora.	El sistema operativo iniciado completamente, mostrando la opción de autenticarse en el sistema.	Iniciar la computadora

Autenticación

Condiciones de ejecución:

El sistema debe estar iniciado correctamente.

Escenario	Descripción	Respuesta del sistema	Flujo central
Autenticarse en el sistema.	Autenticarse en el sistema utilizando un usuario y	Si el usuario y la contraseña son válidos, debe mostrar una consola ejecutando el intérprete de comandos Bash. En caso de ser incorrecto el usuario y/o	Insertar usuario Insertar contraseña.

	contraseña.	la contraseña, impedir la entrada al sistema.	
--	-------------	---	--

Conexión al servidor de cliente ligero mediante el protocolo RDP.

Escenario	Descripción	Respuesta del Sistema	Flujo Central
	Se realizará la conexión al escritorio compartido del servidor de cliente ligero mediante el protocolo RDP.	Se muestra el escritorio compartido en el servidor, mostrando la pantalla de bienvenida con la opción de autenticar usuarios.	Ejecutando: # rdesktop <ip_servidor> Sustituyendo <ip_servidor> por la dirección IP del servidor de cliente ligero.

Conexión al servidor de cliente ligero mediante el protocolo VNC.

Escenario	Descripción	Respuesta del Sistema	Flujo Central
	Se realizará la conexión al escritorio compartido del servidor de cliente ligero mediante el protocolo VNC.	Se muestra el escritorio compartido en el servidor, mostrando la pantalla de bienvenida con la opción de autenticar usuarios.	Ejecutando: # xtightvncviewer <ip_servidor>:1 Sustituyendo <ip_servidor> por la dirección IP del servidor de cliente ligero.

Conexión al servidor de cliente ligero mediante el protocolo XDMCP.

Escenario	Descripción	Respuesta del Sistema	Flujo Central
	Se realizará la conexión al escritorio compartido del servidor de cliente ligero mediante el protocolo XDMCP.	Se muestra el escritorio compartido en el servidor, mostrando la pantalla de bienvenida con la opción de autenticar usuarios.	Ejecutando: # X -query <ip_servidor> :1 Sustituyendo <ip_servidor> por la dirección IP del servidor de cliente ligero.

Configuración del Sonido Local

Escenario	Descripción	Respuesta del Sistema	Flujo Central
	Se configurará el sonido para reproducir algún archivo de sonido y probar si se escucha en el dispositivo local de la computadora real. Para esto se instalará el	El sistema debe reproducir el archivo seleccionado mediante algún reproductor instalado y permitir escuchar mediante el dispositivo de salida de audio de la computadora.	Ejecutar: # mpg321 Archivo.mp3

	reproductor de audio mpg321		
--	------------------------------------	--	--

Configuración del Sonido por la red.

Escenario	Descripción	Respuesta del Sistema	Flujo Central
	Se configurará el sonido para permitir el transporte de audio desde el servidor para el cliente. Para esto se utilizará la herramienta <i>pulseaudio</i> .	Iniciar correctamente la herramienta <i>pulseaudio</i> y detección de los dispositivos de sonido de la tarjeta.	Ejecutar: # pulseaudio --system

4.2 Pruebas de Integración.

Las pruebas de integración permiten verificar el correcto funcionamiento del sistema a medida que se le integren nuevos componentes.

Xorg

Se realiza la prueba al sistema a partir de instalar el Servidor Gráfico Xorg y configurarlo para que inicie forma automática.

Escenario	Descripción	Respuesta del Sistema	Flujo Central
	Una vez instalado el Servidor Gráfico Xorg, y configurado adecuadamente para su inicio automático en el fichero <i>/etc/inittab</i> , se inicia la Computadora.	Iniciar correctamente el sistema, mostrando un recuadro gris como muestra que se instaló correctamente el Servidor Gráfico.	Iniciar la Computadora.

Rdesktop

Se instala la aplicación *rdesktop* que brinda el soporte para el protocolo de comunicación RDP, luego se inicia el sistema para verificar su funcionamiento adecuado con esta nueva herramienta.

Escenario	Descripción	Respuesta del Sistema	Flujo Central
	Se configura de forma adecuada el fichero <i>/etc/inittab</i> , para que inicie el Servidor Gráfico y se conecte de forma	Iniciar correctamente el sistema, mostrando el escritorio compartido en el servidor al cual se conecto de forma	Iniciar la Computadora.

	automática a un servidor mediante el protocolo RDP utilizando la aplicación <i>rdesktop</i> .	automática el sistema.	
--	---	------------------------	--

Xtightvncviewer

Se instala la aplicación *xtightvncviewer* que brinda el soporte para el protocolo de comunicación VNC, luego se inicia el sistema para verificar su funcionamiento adecuado con esta nueva herramienta.

Escenario	Descripción	Respuesta del Sistema	Flujo Central
	Se configura de forma adecuada el fichero <i>/etc/inittab</i> , para que inicie el Servidor Gráfico y se conecte de forma automática a un servidor mediante el protocolo VNC utilizando la aplicación <i>xtightvncviewer</i> .	Iniciar correctamente el sistema, mostrando el escritorio compartido en el servidor al cual se conecto de forma automática el sistema.	Iniciar la Computadora.

4.3 Pruebas al Sistema.

Las pruebas al Sistema, permitirá comprobar el correcto funcionamiento del sistema completo con todos sus componentes integrados.

Conectar al servidor de cliente ligero de forma automática.

Condiciones de Ejecución:

De acuerdo al protocolo de comunicación a utilizar, se debe configurar el inicio del sistema para que se conecte al servidor mediante el protocolo elegido.

Escenario	Descripción	Respuesta del Sistema	Flujo Central
Utilizando el protocolo XDMCP	Se realizará la conexión al escritorio compartido del servidor de cliente ligero mediante el protocolo XDMCP.	Se muestra el escritorio compartido en el servidor, mostrando la pantalla de bienvenida con la opción de autenticar usuarios.	Se inicia el cliente ligero.
Utilizando el protocolo VNC	Se realizará la conexión al escritorio	Se muestra el escritorio compartido en el servidor,	Se inicia el cliente ligero.

	compartido del servidor de cliente ligero mediante el protocolo VNC.	mostrando la pantalla de bienvenida con la opción de autenticar usuarios.	
Utilizando el protocolo RDP	Se realizará la conexión al escritorio compartido del servidor de cliente ligero mediante el protocolo RDP.	Se muestra el escritorio compartido en el servidor, mostrando la pantalla de bienvenida con la opción de autenticar usuarios.	Se inicia el cliente ligero.

4.4 Conclusiones Parciales

Durante el desarrollo del presente capítulo y teniendo en cuenta el resultado de las pruebas realizadas al sistema, se pudo comprobar el correcto funcionamiento de la mayoría de las funcionalidades mismo. La conexión a los servidores mediante los protocolos VNC, RDP y XDMCP se comprobó de manera satisfactoria.

Al probar la funcionalidad del sonido se detectaron algunos problemas. Se comprobó el correcto funcionamiento de audio local, mediante la reproducción de un archivo de sonido, el cual no arrojó errores y pudo ser escuchado mediante el dispositivo de salida de audio de la computadora.

Pro su parte al tratar de transportar audio por la red, mediante la utilización de la herramienta *pulseaudio* no se lograron los objetivos deseados. Entre las posibles causas se encuentra, que hasta la versión del núcleo de linux utilizada (2.6.29) este no le permite el acceso a los dispositivos a los procesos que no entren dentro del grupo de los privilegiados, grupo dentro del cual no se encuentra *pulseaudio*. Con una simple actualización del núcleo de linux se podría intentar resolver el problema, pero es para la versión utilizada que están configurados los controladores de audio, red y tarjeta SD, y al utilizar otra versión del núcleo no estarían disponibles estos controladores indispensables para el funcionamiento de la computadora en una tarjeta del ICID.

La solución al problema anterior estaría en migrar los controladores de red, audio y la tarjeta SD a una versión actual del núcleo de linux, pero esto conllevaría a emplear gran cantidad de tiempo en estudio del funcionamiento del núcleo de linux. Por tal razón, se opta por dejar la funcionalidad de transportar sonido por la red para futuras versiones del sistema.

Como todo sistema operativo las pruebas realizadas y las prácticas utilizadas no validan al 100% la solución ofrecida. Se mantiene además, un sistema de gestión de fallas, mediante las cuales se podrán reportar nuevos errores encontrados por los usuarios que utilicen el sistema.

CONCLUSIONES

Una vez concluida la construcción de la distribución GNU/Linux basada en Nova, para la arquitectura ARM, y con soporte para cliente ligero se puede concluir lo siguiente:

1. Se construyó el primer sistema operativo para usuarios finales, para la Computadora en una Tarjeta diseñada por el ICID.
2. Para la construcción del sistema se utilizaron las técnicas de compilación cruzada de aplicaciones a partir del paquete de código fuente y la instalación directa de paquetes binarios compilados para la arquitectura de computadora ARM de la distribución de GNU/Linux Debian.
3. El sistema brinda las funcionalidades de conexión remota a servidores de cliente ligero en diferentes plataformas y por varios protocolos de comunicación.
4. Se logró un correcto funcionamiento del audio local en la Computadora en una Tarjeta, no siendo así para el transporte de audio entre el servidor y el cliente ligero.

RECOMENDACIONES

El autor de esta investigación recomienda:

1. Continuar la investigación del proceso de construcción de sistemas operativos en dispositivos incrustados.
2. Utilizar este trabajo como guía para la construcción de sistemas operativos para cliente ligero.
3. Realizar un estudio en profundidad de los controladores modificados en el ICID para poder actualizar la versión del núcleo de linux a utilizar en futuras versiones del producto para añadir nuevos soportes y funcionalidades.

REFERENCIAS BIBLIOGRÁFICAS

Debian. Debian -- Acerca de Debian. *Debian.org* 2011. [cited 8 Abril 2011]. Available from world wide web: <<http://www.debian.org/intro/about.es.html>>.

Hower, Derek et al. WWW Computer Architecture Page. *WWW Computer Architecture Page* 2010. [cited 1 Marzo 2011]. Available from world wide web: <<http://arch-www.cs.wisc.edu/home>>.

Hurtado, Mijail et al. Nova al servicio de los Clientes Ligeros y máquinas sin disco. 2010. [cited 2 Diciembre 2010]. Available from world wide web: .

ICID. Manual de Usuario MBTI2440. [cited 10 Diciembre 2010]. Available from world wide web: .

Jacobson, I., Booch, G., y Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*. Addison Wesley, 2000 [cited 30 Marzo 2011]. Available from world wide web: .

Kruchten, Philippe. *The Rational Unified Process: An Introduction*. 3 Addison Wesley, 2000 [cited 30 Marzo 2011]. Available from world wide web: .

Machín, Jorge Luis, y Anay Corso. Descripción del proceso de construcción del sistema operativo base de la distribución cubana de GNU/Linux Nova. 2010. [cited 25 Noviembre 2010]. Available from world wide web: .

MIC. ¿Qué es INFOSOC? *Ministerio de Informatica y Comunicaciones* 2010. [cited 30 Marzo 2011]. Available from world wide web: <<http://www.mic.gov.cu/>>.

Parra, Mauro. Un vistazo a la arquitectura ARM. Junio 2001. [cited 30 Marzo 2011]. Available from world wide web: .

Samsung Electronics co. Instalación Manual para S3C2440. 2004. [cited 30 Marzo 2011]. Available from world wide web: .

Scott, Gary, y Tracy LaQuey. RFC 1392. *Internet Users Glossary* 1993. [cited 8 Abril 2011]. Available from world wide web: <<http://tools.ietf.org/html/rfc1392>>.

Universidad Politécnica de Valencia. Rational Unified Process (RUP). [cited 30 Marzo 2011]. Available from world wide web: .

BIBLIOGRAFÍA

- Arquitectura de computadoras - Wikipedia, la enciclopedia libre. *Wikipedia* 2010a. [cited 1 Marzo 2011]. Available from world wide web: <http://es.wikipedia.org/wiki/Arquitectura_de_computadoras>.
- Cliente liviano - Wikipedia, la enciclopedia libre. *Wikipedia* 2010b. [cited 1 Marzo 2011]. Available from world wide web: <http://es.wikipedia.org/wiki/Cliente_liviano>.
- Como desempaquetar un DEB y RPM [Extraer Contenido - "Descomprimir"]. *Skamasle* 2010c. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.skamasle.com/como-desempaquetar-un-deb-y-rpm-extraer-contenido-descomprimir/>>.
- Conceptos básicos de make y los Makefile. *Ejemplos java y C/linux* 2007. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.chuidiang.com/clinix/herramientas/makefile.php>>.
- Escritorio Remoto con sonido. *Ubuntu-es* 2009. [cited 24 Marzo 2011]. Available from world wide web: <<http://www.ubuntu-es.org/index.php?q=node/93024>>.
- HOWTO: Set up VNC server with resumable sessions - Ubuntu Forums. [cited 22 Marzo 2011]. Available from world wide web: <<http://ubuntuforums.org/showthread.php?t=122402>>.
- Installing Debian (ARM) on QEMU. *Your Electronics Open Source* 2009. [cited 1 Marzo 2011]. Available from world wide web: <<http://dev.emcelettronica.com/installing-debian-arm-qemu>>.

- PulseAudio. *Gentoo Linux Wiki* 2011. [cited 22 Marzo 2011]. Available from world wide web: <<http://en.gentoo-wiki.com/wiki/PulseAudio>>.
- Remote Desktop Protocol - Wikipedia, la enciclopedia libre. *Wikipedia* 2010. [cited 1 Marzo 2011]. Available from world wide web: <http://es.wikipedia.org/wiki/Remote_Desktop_Protocol>.
- RFB protocol - Wikipedia, the free encyclopedia. *Wikipedia* 2011. [cited 1 Marzo 2011]. Available from world wide web: <http://en.wikipedia.org/wiki/RFB_protocol>.
- Tutorial para configurar XDMCP. *Configuración de XDMCP*. [cited 14 Marzo 2011]. Available from world wide web: <<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Interfaces/XWindows/tutoXDMCP.html>>.
- Ubuntu Forums - HOWTO: Setting up a VNC to GDM connection. [cited 22 Marzo 2011]. Available from world wide web: <<http://ubuntuforums.org/printthread.php?t=42941>>.
- VNC - Virtual Network Computing from AT&T Laboratories Cambridge. *VNC, Virtual Network Computing* 2001. [cited 1 Marzo 2011]. Available from world wide web: <http://www.hep.phy.cam.ac.uk/vnc_docs/index.html>.
- VNC - Wikipedia, la enciclopedia libre. *Wikipedia* 2011. [cited 1 Marzo 2011]. Available from world wide web: <<http://es.wikipedia.org/wiki/VNC>>.
- Aoki, Osamu. Debian Reference (version 1). *Debian* 2008. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.debian.org/doc/manuals/debian-reference/#contents>>.

- Balau. Booting Linux with U-Boot on QEMU ARM « Balau. *Balau Freedom Embedded* 2010a. [cited 1 Marzo 2011]. Available from world wide web: <<http://balau82.wordpress.com/2010/04/12/booting-linux-with-u-boot-on-qemu-arm/>>.
- Balau. Busybox for ARM on QEMU « Balau. *Balau Freedom Embedded* 2010b. [cited 1 Marzo 2011]. Available from world wide web: <<http://balau82.wordpress.com/2010/03/27/busybox-for-arm-on-qemu/>>.
- Balau. Compiling Linux kernel for QEMU ARM emulator « Balau. *Balau Freedom Embedded* 2010c. [cited 1 Marzo 2011]. Available from world wide web: <<http://balau82.wordpress.com/2010/03/22/compiling-linux-kernel-for-qemu-arm-emulator/>>.
- Balau. Linux NFS Root under QEMU ARM emulator « Balau. *Balau Freedom Embedded* 2010d. [cited 1 Marzo 2011]. Available from world wide web: <<http://balau82.wordpress.com/2010/04/27/linux-nfs-root-under-qemu-arm-emulator/>>.
- Barrios, Joel, y Sandino Flores. Cómo configurar el sistema para sesiones gráficas remotas. *Linux Para Todos*. 2007. [cited 14 Marzo 2011]. Available from world wide web: <<http://www.linuxparatodos.net/portal/staticpages/index.php?page=11-como-x-remoto>>.
- Basso, Minerva Hernández. La nueva GEDEME. *Opciones.cu* Octubre 2005. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.opciones.cu/leer.asp?idnuevo=1592>>.

Beekmans, Gerard. *Linux From Scratch*. 2007 [cited 30 Marzo 2011]. Available from world wide web: .

Beekmans, Gerard. Welcome to Linux From Scratch! *Linux from scratch*. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.linuxfromscratch.org/>>.

Berry, Andrew. Start a VNC Server on Ubuntu on Boot. *Andrew Berry Development* 2008. [cited 22 Marzo 2011]. Available from world wide web: <<http://www.abdevelopment.ca/blog/start-vnc-server-ubuntu-boot>>.

Brody, Tim. Ubuntu 10.04 vnc-based login server. *An Open Access Peon* 2010. [cited 22 Marzo 2011]. Available from world wide web: <<http://oapeon.blogspot.com/2010/05/ubuntu-1004-vnc-based-login-server.html>>.

Carter, Jonathan et al. Theory of operation. *EDUBUNTU*. [cited 4 Marzo 2011]. Available from world wide web: <<http://doc.ubuntu.com/edubuntu/edubuntu/handbook/C/ltsp-theory.html>>.

Ciccione, Joe, Jim Gifford, Maarten Lankhorst, y Ryan Oliver. Cross-Compiled Linux From Scratch - Embedded. *Cross Linux From Scratch* 2010. [cited 1 Marzo 2011]. Available from world wide web: <<http://cross-lfs.org/view/clfs-embedded/arm/index.html>>.

CodeSourcery. CodeSourcery: Company: About. *CodeSourcery* 2010a. [cited 15 Marzo 2011]. Available from world wide web: <<http://www.codesourcery.com/company>>.

CodeSourcery. Sourcery G++ Lite. 2010b. [cited 5 Dicembre 2010]. Available from world wide web: .

Cooper, Mendel. Building and Installing Software Packages for Linux. *The Linux Documentation Project* 1999. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.tldp.org/HOWTO/Software-Building-HOWTO.html>>.

Debian. LTSP/Howto - Debian Wiki. *LTSP How To* 2011. [cited 4 Marzo 2011]. Available from world wide web: <<http://wiki.debian.org/LTSP/Howto>>.

Developers, TightVNC. TightVNC: VNC-Compatible Free Remote Control / Remote Desktop Software. *tightVNC* 2010. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.tightvnc.com/>>.

Dooks, Ben. Linux/Documentation/arm/Samsung-S3C24XX/Overview.txt - Linux Cross Reference - Free Electrons. *Linux Cross Reference* 2006. [cited 1 Marzo 2011]. Available from world wide web: <<http://lxr.free-electrons.com/diff/Documentation/arm/Samsung-S3C24XX/Overview.txt?v=2.6.25;diffval=sh;diffvar=a>>.

Dubois, Paul. imake Frequently Asked Questions. *imake Frequently Asked Questions* 1997. [cited 1 Marzo 2011]. Available from world wide web: <http://www.snake.net/software/imake-stuff/imake-faq.html#TOC_1>.

Free Electrons. Tiny qemu arm system with a DirectFB interface - Free Electrons. *Free Electrons* 2010. [cited 1 Marzo 2011]. Available from world wide web: <<http://free-electrons.com/community/demos/qemu-arm-directfb/>>.

- Hower, Derek et al. WWW Computer Architecture Page. *WWW Computer Architecture Page* 2010. [cited 1 Marzo 2011]. Available from world wide web: <<http://arch-www.cs.wisc.edu/home>>.
- Hurtado, Mijail et al. Nova al servicio de los Clientes Ligeros y máquinas sin disco. 2010. [cited 2 Diciembre 2010]. Available from world wide web: .
- ICID. Manual de Usuario MBTI2440. [cited 10 Diciembre 2010]. Available from world wide web: .
- Jacobson, I., Booch, G., y Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*. Addison Wesley, 2000 [cited 30 Marzo 2011]. Available from world wide web: .
- Jarno, Aurélien. Debian on an emulated ARM machine. *Debian on an emulated ARM machine* 2008. [cited 1 Marzo 2011]. Available from world wide web: <http://www.aurel32.net/info/debian_arm_qemu.php.en>.
- King, Russell. Kernel Compilation. *The ARM Linux Project* 2004. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.arm.linux.org.uk/docs/kerncomp.php>>.
- Kruchten, Philippe. *The Rational Unified Process: An Introduction*. 3 Addison Wesley, 2000 [cited 30 Marzo 2011]. Available from world wide web: .
- LTSP. LTSP. *Linux Terminal Server Project*. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.ltsp.org/>>.
- Machín, Jorge Luis, y Anay Corso. Descripción del proceso de construcción del sistema operativo base de la distribución cubana de GNU/Linux Nova. 2010. [cited 25 Noviembre 2010]. Available from world wide web: .

Mankinen, Veli, y Valtteri Rahkonen. Cross-Compiling tutorial with Scratchbox. *Scratchbox* 2005. [cited 1 Marzo 2011]. Available from world wide web: <<http://scratchbox.org/documentation/user/scratchbox-1.0/html/tutorial.html>>.

Martínez, Emilio. GEDEME Artículo Principal. *Grupo de la Electrónica*. [cited 1 Marzo 2011]. Available from world wide web: <http://www.gec.cu/index.php?option=com_content&view=article&id=64&Itemid=62.%20%20GEDEME>.

Meza, Jorge. Acceder al escritorio remoto de GNU/Linux Ubuntu 10.04 utilizando Xephyr. *Jorge Iván Meza Martínez* 2010. [cited 24 Marzo 2011]. Available from world wide web: <<http://blog.jorgeivanmeza.com/2010/09/acceder-al-escritorio-remoto-de-gnulinix-ubuntu-10-04-utilizando-xephyr/>>.

MIC. ¿Qué es INFOSOC? *Ministerio de Informatica y Comunicaciones* 2010. [cited 30 Marzo 2011]. Available from world wide web: <<http://www.mic.gov.cu/>>.

Microsoft. Remote Desktop Protocol (Windows). *MSDN* 2010. [cited 1 Marzo 2011]. Available from world wide web: <[http://msdn.microsoft.com/en-us/library/aa383015\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa383015(VS.85).aspx)>.

Parra, Mauro. Un vistazo a la arquitectura ARM. Junio 2001. [cited 30 Marzo 2011]. Available from world wide web: .

Pérez, Eloy. Configuración xdmcp en Ubuntu 9.10 Karmic Koala. *Eloy Mier Pérez Site* 2010. [cited 22 Marzo 2011]. Available from world wide web: <<http://eloymp.com/wordpress262/?p=1719>>.

Proyecto Nova. Que es Nova ? *Nova, Distribución Cubana de GNU/Linux* 2010. [cited 1 Marzo 2011]. Available from world wide web: <http://www.nova.cu/?q=es_nova>.

RealVNC. RealVNC - VNC® remote control software. *Real VNC*. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.realvnc.com/>>.

Reimer, Klaus. ARM cross-compiling howto - K's cluttered loft. *K's cluttered loft* 2010. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.ailis.de/~k/archives/19-ARM-cross-compiling-howto.html>>.

S-4. Cliente Ligero. S-4. [cited 1 Marzo 2011]. Available from world wide web: <http://www.s-4.es/mt_ClienteLigero.aspx>.

Samsung Electronics co. Instalación Manual para S3C2440. 2004. [cited 30 Marzo 2011]. Available from world wide web: .

systemadmin.es. Conectar desde Linux a un equipo Windows por RDP | systemadmin.es. *systemadmin.es* 2010. [cited 14 Marzo 2011]. Available from world wide web: <<http://systemadmin.es/2010/10/conectar-desde-linux-a-un-equipo-windows-por-rdp>>.

Ulloa, Sergio. Referencias Bibliográficas / Norma ISO 690. *Norma ISO 690* 2009. [cited 1 Marzo 2011]. Available from world wide web: <<http://tutorialsibusach.pbworks.com/w/page/22541893/Norma-ISO-690>>.

Universidad Politécnica de Valencia. Rational Unified Process (RUP). [cited 30 Marzo 2011]. Available from world wide web: .

UPC. Departament d'Arquitectura de Computadors. *UPC* 2009. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.ac.upc.edu/>>.

Valroff, Julien. Clear dpkg status file « Julien Valroff. *Julien Valroff* 2011. [cited 1 Marzo 2011]. Available from world wide web: <<http://www.kirya.net/tips/clear-dpkg-status-file/>>.

Verens, Kae. using vnc to create a thin client | klog. *KLOG* 2004. [cited 1 Marzo 2011]. Available from world wide web: <<http://verens.com/2004/09/25/using-vnc-to-create-a-thin-client/>>.

WordReference.com. compilar - Definición. *WordReference.com* 2005. [cited 30 Marzo 2011]. Available from world wide web: <<http://www.wordreference.com/definicion/compilar>>.

Yero, Arlenys, y Darianna Gonzalez. Intranet de la Empresa Albet S.A. 2010. [cited 29 Marzo 2011]. Available from world wide web: .

GLOSARIO

Computadora en una Tarjeta: Es una computadora diseñada por el ICID, basada en la Arquitectura de Computadora ARM, con el objetivo inicial de utilizarla en equipos médicos de mediana complejidad.

Sistema Operativo: Conjunto de programas y utilidades básicas que hacen funcionar la computadora.

Componentes de hardware: Dispositivos físicos que componen una computadora.

Dispositivos: Cualquier elemento que se pueda unir al sistema mediante algún medio de conexión.

Unidad Central de Procesamiento: Central Processing Unit (CPU) o **procesador**, es el cerebro de la computadora, es donde se produce la mayoría de los cálculos.

Acorn Computer Group: Compañía informática británica, fundada en el año 1978. Actualmente se encuentra dividida en varias unidades, pero es reconocida por la construcción del primer procesador RISC.

RISC: Reduced Instruction Set Computer, es una filosofía para la construcción de procesadores, se define además como una Arquitectura de Computadora.

Sistema embebido: Un sistema embebido o incrustado es un sistema de computación diseñado para realizar funciones dedicadas, frecuentemente en un sistema de computación en tiempo real.

Debootstrap: Herramienta que permite crear sistemas bases, a partir de un listado de aplicaciones esenciales definido por la propia herramienta.

Paquete de código fuente: Es el software escrito en algún lenguaje de programación.

Paquete de código binario: Conjunto de ficheros creados como el resultado de la compilación de un paquete de código fuente.

Dpkg: Ver Gestor de Paquetes.

Gestor de Paquetes: Es una colección de herramientas que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software.

Dependencias: Conjunto de paquetes de los cuales depende otro paquete ya sea para compilarse, instalarse o ejecutarse.

RDP: Remote Desktop Protocol, es un protocolo de comunicación propietario desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre una computadora cliente y una computadora servidor.

VNC: Virtual Network Computing, es un protocolo de comunicación que permite controlar de forma remota un escritorio compartido en otra computadora.

XDMCP: X Display Manager Control Protocol, es un protocolo utilizado en redes para comunicar un ordenador servidor que ejecuta un sistema operativo con un gestor de ventanas basado en X-Window, con el resto de clientes que se conectarán a éste con propósitos interactivos.

Servidor Gráfico: Es el encargado de dotar de una Interfaz Gráfica a los sistemas Unix.

Pulseaudio: Es un servidor de sonido multi-plataforma, capaz de funcionar por red. Funciona bajo sistemas compatibles con POSIX como GNU/Linux y otros sistemas operativos como Microsoft Windows.

Núcleo: Es la parte fundamental de un sistema operativo. Es el software responsable de facilitar a los distintos programas acceso seguro a los dispositivos físicos de la computadora. Es el encargado de gestionar recursos, a través de servicios de llamadas al sistema.

Módulos del núcleo: Son unidades de programa discretas e identificables con respecto a la compilación, la combinación con otras unidades y la carga.

Sistema anfitrión: Se define como el sistema operativo sobre el cual se realizará el proceso de construcción del nuevo sistema.

Computadora anfitrión: Es la computadora donde está instalado el sistema anfitrión.

Computadora real: Es la computadora en un Tarjeta diseñada por el ICID.

Emular: Simular el hardware o software de otra computadora.

Sistema de ficheros: Conjunto algoritmos y estructuras auxiliares que nos van a permitir de manera sencilla y transparente acceder nuestros datos en dispositivos de almacenamiento.

Tarjeta SD: La memoria Secure Digital (también conocida como *SD* o *Tarjeta SD*) es un tipo de tarjeta de memoria creada por Matsushita Electronic, SanDisk y Toshiba en enero de 2000.

Controlador: Un controlador es software que permite que el equipo se comuniquen con hardware o dispositivos.

Memoria NAND: Dispositivo de almacenamiento fijo, incrustado en tarjeta madre de la computadora, que puede eléctricamente escribir y borrar datos.

Cable serial: Es un cable que se usa para transportar datos entre dos dispositivos mediante una comunicación serial.

Comunicación Serial: Es el proceso de enviar datos bit a bit, sobre un canal de comunicación.