

Universidad de las Ciencias Informáticas

Facultad 1



Virtualización de Android para la CID 300/9

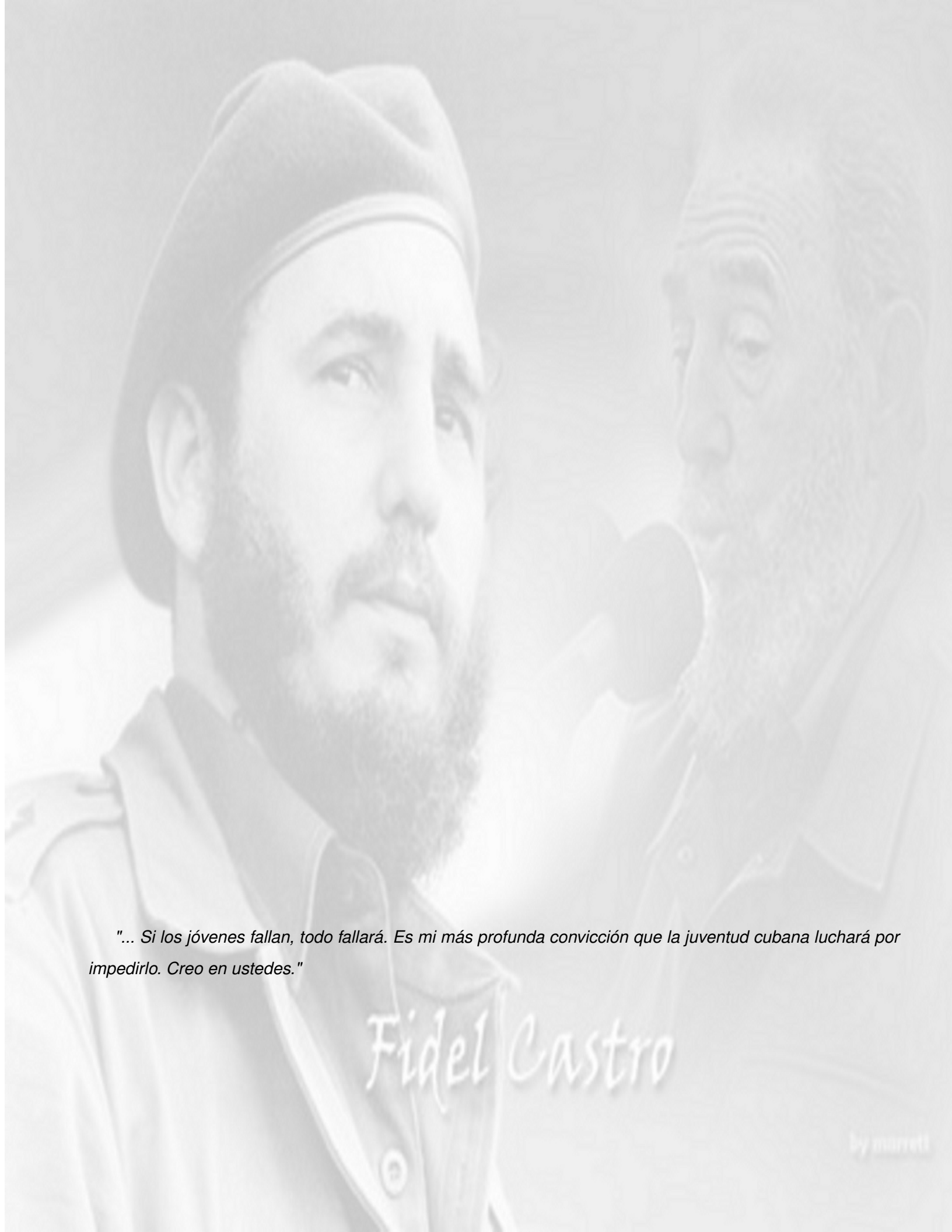
Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor: Danier Parra Barbán

Tutor: MSc. Dariem Pérez Herrera

Co-Tutor: Ing. Ernesto Puente Fuentes

La Habana, 10 de junio de 2013



"... Si los jóvenes fallan, todo fallará. Es mi más profunda convicción que la juventud cubana luchará por impedirlo. Creo en ustedes."

Fidel Castro

by marrett

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la UCI los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Danier Parra Barbán

Firma del Autor

Dariem Pérez Herrera

Firma del Tutor

Ernesto Puente Fuentes

Firma del Tutor

Agradecimientos

A mi mami linda, por existir siempre, por darme fuerza, por confiar en mi, por demostrarme que de cada acontecimiento, de cada suceso, de cada persona se pueden extraer cosas positivas.

A mi papá, por confiar en mi y darme fuerzas para seguir adelante, por hacer de mi la persona que no se cansa de buscar siempre la perfección en los detalles y con la necesidad de estar aprendiendo en todo momento.

A mi hermana por ser esa personita que hace que me esfuerce más cada día por ser su ejemplo, por alegrarme cada momento que pasamos juntos, por hacer inigualable su presencia.

A mis abuelos queridos, por ser más que abuelos padres para mí, por complacerme en cada uno de mis gustos, por educarme e inculcarme a lo largo de mi vida.

A mi tía Damicela, que aunque no se encuentre entre nosotros, fue mi segunda madre, mi faro, mi luz, mi sostén, mi guía.

A mi familia por estar siempre ahí para mí, por acojermme con un cariño sincero e inigualable, por confiar en mí en cada momento, aún cuando las circunstancias no eran las mejores.

A mi novia por alegrarme cada día y estar conmigo en los momentos buenos y fusionarse conmigo en los malos, por hacer que cada día sea un reto por volverla a conquistar, por regalarme momentos inigualables e inolvidables.



A todos los profesores que me impartieron clases a lo largo de la carrera por transmitirme sus conocimientos.

A mi compañeros, por ser estar conmigo en los momentos difíciles, por hacer suyos mis problemas, por compartir mis alegrías.

A todas las amistades que he cultivado a lo largo de estos cinco años, sobre todo a aquellos que hemos estado juntos desde primer año: Samuel, Roberto, Daniel, El flaco, Dennis, Nuricel, Iliana, Elizabeth.

A mis compañeros del proyecto por ayudarme en cada momento y hacer que cada momento que pasé junto a ellos fuese especial, por ser especiales.

A Alfonso y Yannier por ser mis hermanos en esta etapa de mi vida, por ser incondicionales, leales y sobre todo amigos de verdad.

A mi tutor Puente, que por ser más que mi tutor fue mi amigo, por sacar siempre lo mejor de mi, por hacer que cada día creciera como profesional, por confiar en mi y sobre todo por las experiencias que me transmitió.

A Haniel y Edito, por ser excelentes, tanto en lo personal como en lo profesional, por ayudarme en cada momento, por siempre estar dispuestos a ayudarme, por ser ejemplos.

Al equipo de trabajo de Nova, por ser como una familia grande, por compartir los conocimientos, por forjarme como un mejor profesional y como una mejor persona, en especial a a Abel, El Migue, Marlies, Luis Daniel, Daniel, a todos.

A todos aquellos que de una forma u otra han tenido que ver con mi formación profesional, así como a aquellos que han formado parte en el desarrollo de este trabajo.



Dedicatoria

Danier Parra Barbán

El presente trabajo de diploma está dedicado especialmente a mis padres; mi mami que me apoya siempre y es mi faro y guía en este mundo, mi inspiración y mis logros son de ella por ser la gran mujer que es. A mi padre por enseñarme a ser un hombre y por forjar en mí los valores que poseo, por estar a mi lado en los momentos duros y difíciles que me han tocado vivir; GRACIAS A AMBOS. A ustedes le dedico este trabajo por ser los mejores padres del mundo. Gracias por existir...

A mi mamá y a mi papá, dos personas especiales y únicas, quisiera tenerlos siempre.

A mi hermana, mi hermanita del alma.

A mi familia toda.



Resumen

Para cumplir con los objetivos trazados por el proyecto Nova, perteneciente a la Facultad 1 de la Universidad de Ciencias Informáticas (UCI) se realizó una solución de virtualización, la cual permitió el empleo del sistema operativo Android sobre la CID 300/9. El desarrollo de esta solución fomenta en Cuba el dinamismo en la explotación de nuevas tecnologías y su integración con la Computadora en una Tarjeta creó un producto con gran valor agregado por sus desarrolladores. Para el desarrollo de esta solución se realizó un estudio de las tecnologías de virtualización y consumo para sistemas operativos, protocolos de comunicación remota y software de virtualización, y se definió como núcleo el kernel Linux en su versión 2.6.29, para la construcción del sistema de ficheros se empleó la herramienta Buildroot. Se definió como software de virtualización a VirtualBox, como herramienta de compilación cruzada fue usada CodeSourcery, para la comunicación con la computadora real a través de la puerta serial se empleó Minicom y como protocolo de comunicación se usó RDP.

Palabras claves: Computadora en una Tarjeta, Puerta Serial , Sistema de Ficheros, Sistema Operativo, Virtualización.

Índice General

Índice de figuras	x
Índice de tablas	xii
Introducción	1
1. Capítulo 1 Fundamentación teórica	6
1.1. La MBTI2440: Computadora en una Tarjeta.	6
1.2. Android	8
1.2.1. Características	8
1.2.2. Arquitectura	10
1.2.3. Diseño y desarrollo	11
1.2.4. Aplicaciones	11
1.3. Virtualización	12
1.3.1. Virtualización de Android	13
1.4. Software de Virtualización	16
1.4.1. VirtualBox	17
1.4.2. VMware WorkStation	18
1.4.3. Qemu	19
1.5. Protocolos de comunicación remota	20
1.5.1. Protocolo VNC	21
1.5.2. Protocolo RDP	22
1.6. Protocolo NFS	23

1.7. Modelado de proceso	24
1.8. Conclusiones parciales	24
2. Capítulo 2 Diseño del Sistema	25
2.1. Funcionamiento del Sistema.	25
2.1.1. Funcionamiento del Cliente	26
2.2. Requerimientos del Sistema	29
2.2.1. Requerimientos del Cliente	29
2.2.2. Requerimientos del Servidor	30
2.3. Arquitectura del Sistema	30
2.3.1. Núcleo	31
2.3.2. Sistema Base	31
2.3.3. Aplicaciones de conexión remota	32
2.4. Arquitectura del Servidor	33
2.5. Conclusiones parciales	33
3. Capítulo 3 Construcción del Sistema	34
3.1. Núcleo	35
3.2. Sistema de Ficheros	37
3.3. Administración del servidor de virtualización	42
3.4. Asignación de imágenes virtuales	44
3.5. Emulación del sistema	44
3.6. Grabar Sistema de Ficheros creado a la computadora real	45
3.7. Conclusiones parciales	45
4. Capítulo 4 Pruebas al Sistema.	46
4.1. Pruebas a funcionalidades	46
4.2. Pruebas de integración	48
4.3. Pruebas al sistema	49
4.4. Conclusiones parciales	49

Conclusiones	51
Recomendaciones	52
Glosario de Términos	53
Referencias bibliográficas	55
Bibliografía	61
Anexo I	67
Anexo II	78
Anexo III	79

Índice de figuras

1.	Cuota de mercado de Android entre los años 2009-2012	2
1.1.	Vista general de la solución “Virtual SmartPhone over IP”	15
1.2.	Vista general de la virtualización de Android sobre la arquitectura x86	16
2.1.	Vista general del funcionamiento del sistema	26
2.2.	Vista general del funcionamiento del cliente	26
2.3.	Vista general del Inicio del Núcleo	27
2.4.	Vista general del Inicio del Sistema	28
2.5.	Vista general de la conexión al Servidor	28
2.6.	Vista general de la arquitectura del Sistema	31
2.7.	Vista general del núcleo del Sistema	31
2.8.	Vista general de los componentes del Sistema	32
2.9.	Vista general de las aplicaciones de conexión remota	32
2.10.	Vista general de arquitectura del Servidor de Virtualización	33
4.1.	Vista general de la configuración de Minicom	69
4.2.	Vista general de la configuración de la puerta serial	69
4.3.	Vista general de la configuración de la puerta serial	70
4.4.	Vista general de la configuración de la puerta serial	70
4.5.	Vista general de la configuración de la puerta serial	71
4.6.	Vista general de la configuración de dash	72
4.7.	Vista general de la configuración de CodeSourcery	72

4.8. Vista general de la configuración de CodeSourcery	73
4.9. Vista general de la configuración de CodeSourcery	73
4.10. Vista general de la configuración de CodeSourcery	74
4.11. Vista general de la configuración de CodeSourcery	74
4.12. Vista general de la configuración de CodeSourcery	75
4.13. Vista general de la configuración de CodeSourcery	75
4.14. Vista general de la configuración de CodeSourcery	76
4.15. Vista general de la configuración de CodeSourcery	76
4.16. Vista general de la configuración de CodeSourcery	77
4.17. Vista general de la configuración de CodeSourcery	77
4.18. Vista general del proceso de cargar un nuevo kernel.	79
4.19. Vista general del proceso de cargar un nuevo kernel.	80
4.20. Vista general del proceso de cargar un nuevo kernel.	80
4.21. Vista general del proceso de cargar un nuevo kernel.	81

Índice de tablas

4.1. <i>Inicio del sistema</i>	46
4.2. <i>Inicio del servidor gráfico</i>	47
4.3. <i>Autenticación</i>	47
4.4. <i>Conexión al servidor de virtualización mediante el protocolo RDP.</i>	48
4.5. <i>Aplicación rdesktop.</i>	49
4.6. <i>Pruebas al sistema.</i>	49

Introducción

Android es un sistema operativo de código abierto basado en el núcleo Linux, diseñado originalmente para telefonía móvil. Con el transcurso de los años, logra expandir su desarrollo para otros dispositivos como tabletas, reproductores de audio y video, computadoras con funcionalidades limitadas (*netbook*) [1], hasta llegar a microondas y lavadoras. Inicialmente, Android fue desarrollada por Google Inc. aunque poco después se unió Open Handset Alliance, un consorcio de 48 compañías de hardware, software y telecomunicaciones, las cuales llegaron a un acuerdo para promocionar los estándares de códigos abiertos para dispositivos móviles [2]. En la actualidad posee alrededor de 700 000 aplicaciones disponibles y tiene como promedio la liberación de 25 000 aplicaciones mensuales [3]. Android es un sistema operativo reconocido a escala mundial por sus características, algunas de las más distintivas son [4]:

- **Framework de aplicaciones:** permite reutilización y reemplazo de componentes.
- **Navegador integrado:** basado en el motor WebKit de código abierto.
- **Gráficos optimizados** por una colección de gráficos personalizados en 2D y gráficos 3D basados en OpenGL ES 1.0.
- **SQLite:** para almacenamiento de datos estructurados.
- **Cámara, GPS, brújula, y acelerómetro** (dependiente del hardware)

Android, al estar basado en el núcleo Linux es del agrado de los desarrolladores de GNU/Linux, lo que ha traído como consecuencia que sea un mejor sistema operativo y posea mayor demanda en el mercado mundial. Las características más distintivas, que hacen que Android sea del agrado de los desarrolladores de GNU/Linux se mencionan a continuación:

- Eliminación de capas de abstracción.

- Mejora de la comunicación entre procesos.
- Estandarización de tecnologías.
- Estabilidad.
- El código sigue siendo abierto.

En los últimos años Android ha ido escalando peldaños en la venta de unidades de Teléfonos Inteligentes (Smartphone de ahora en lo adelante) en planos mundiales, lo que evidencia gran aceptación por parte de los usuarios. El mismo ha logrado aumentar considerablemente sus ventas anuales y superar a sistemas operativos reconocidos a nivel mundial como Symbian e iOS [5].

Para el año 2011 se evidenciaba claramente la tendencia por parte de los usuarios al uso de Android como sistema operativo para Smartphone, al mismo tiempo que Symbian e iOS veían reducidas sus aspiraciones a crecer. En ese año, la cuota de mercado de Android para telefonía móvil ocupaba más del 40 %, mientras que sus principales rivales; Symbian e iOS se conformaban con el 20 % y 18 % respectivamente [6]. En el año 2012 aumenta la rivalidad entre Google y Apple, donde el primero de estos amplió considerablemente su ventaja. Android aumentó significativamente al 53 % de dominio en el mercado de Smartphone, mientras que el sistema operativo de Apple, iOS, se quedó cómodamente en segundo lugar con 20 % de los envíos de Smartphone, según nuevas estadísticas de la empresa de investigación IDC (Corporación Internacional de Datos). Juntos, los dos sistemas operativos sumaron el 73 % de todos los Smartphone nuevos en 2012, lo que dejó a los competidores BlackBerry y Windows muy atrás [7].

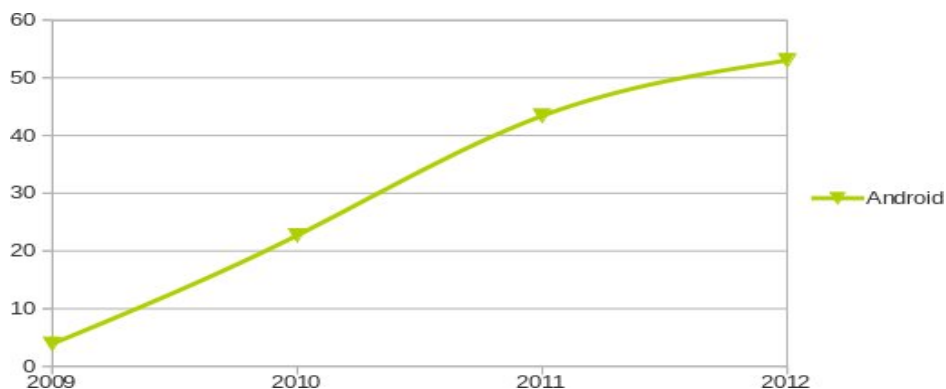


Figura 1: Cuota de mercado de Android entre los años 2009-2012

El gráfico anterior muestra la tendencia del aumento de la cuota de mercado a nivel mundial de Android entre los años 2009 y 2012.

Paralelo al desarrollo de Android, en Cuba se desarrollaría la CID 300/9, computadora cubana diseñada por personal del Instituto Central de Investigación Digital (ICID) basada en la Placa Madre MBTI2440. Esta placa permite eludir las trabas del bloqueo impuesto por los Estados Unidos a Cuba, además de proveer una vía accesible para fomentar el proceso de informatización de la sociedad cubana.

Con el desarrollo de la CID 300/9 se le brinda a Cuba la capacidad de desarrollarse en este campo de forma autónoma. Al ser el hardware de producción nacional, los cubanos pueden adaptarlo a sus necesidades, cumpliendo siempre con los principios de Seguridad, Soberanía Tecnológica, Socio-adaptabilidad y Sostenibilidad [8].

Como parte de los objetivos trazados en el proyecto Nova, perteneciente a la Facultad 1 de la Universidad de Ciencias Informáticas (UCI) está la utilización de Android sobre la CID 300/9. El uso de este sistema operativo fomenta el dinamismo en la explotación de nuevas tecnologías en Cuba, y su integración con hardware cubano crea un producto con gran valor agregado por sus desarrolladores. Para cumplir con estos objetivos se decide el empleo de una solución de virtualización que permita la utilización de Android en la CID 300/9.

Partiendo de la presente problemática se plantea como **pregunta científica** de la presente investigación: ¿Cómo implementar una solución de virtualización de Android para la CID 300/9?

Definiéndose como **objeto de estudio** de la investigación: Las tecnologías de virtualización y consumo para Sistemas Operativos, y se enmarca como **campo de acción**: Las tecnologías de virtualización y consumo para Android.

Se plantea como **idea a defender**: La creación de una solución de virtualización de Android para su consumo en la CID 300/9, para permitir la utilización de nuevas tecnologías en el campo de los sistemas operativos sobre hardware diseñado en Cuba.

Tomando como punto de partida esta premisa, la presente investigación define como **objetivo general** Implementar una solución que permita la ejecución de Android en la CID 300/9, utilizando el conocimiento acumulado por los desarrolladores del proyecto Nova para la integración de hardware cubano con nuevas tecnologías.

Para darle solución al mismo se definen los siguientes **objetivos específicos**:

1. Sistematizar las soluciones de virtualización, para aumentar el dominio sobre el campo de acción de la investigación
2. Sistematizar el proceso de construcción de sistemas operativos para la CID 300/9, para la futura obtención de un producto estable
3. Definir la nueva solución para darle cumpliendo a objetivo general
4. Implementar una prueba de concepto para validar la nueva solución

Para darle cumplimiento a los objetivos específicos se plantean las siguientes **tareas de la investigación**:

1. Análisis de las soluciones de virtualización para comprender su funcionamiento interno
2. Estudio de las herramientas asociadas a estas soluciones
3. Concepción de la nueva solución para su adaptación a entornos cubanos
4. Implementación del componente en el servidor
5. Implementación del componente en la CID 300/9
6. Realización de pruebas para validar la nueva solución

Para solucionar las tareas antes mencionadas se utilizaron los **métodos científicos**:

- Del nivel teórico:
 - Método histórico-lógico: Se analiza la evolución de diferentes teorías y herramientas que apoyan la construcción de estos sistemas, mediante la búsqueda en diferentes bibliografías.
- Del nivel empírico:
 - Método de la observación: Permite 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 experimental: El experimento permite estudiar el objeto, crear las condiciones para verificar la idea a defender.

El presente trabajo de diploma está compuesto por un resumen, introducción, cuatro capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y un glosario de términos. A continuación se describen los principales aspectos abordados en cada uno de los capítulos:

Capítulo 1 Fundamentación teórica: Se realiza un estudio del estado del arte de las tecnologías de virtualización y consumo de Android. Además de los software de virtualización y protocolos de comunicación remota.

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 sus principales componentes.

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.

Capítulo 1

Capítulo 1 Fundamentación teórica

En el presente capítulo se realiza un estudio de: las diferentes tecnologías de virtualización y consumo para sistemas operativos; las herramientas para la construcción y emulación de sistema de ficheros y los diferentes protocolos para la conexión remota. Además, se definen las herramientas para modelar los procesos y compilar aplicaciones.

1.1. La MBTI2440: Computadora en una Tarjeta.

ARM (*Advanced RISC Machine* o máquina avanzada RISC) es una arquitectura RISC (*Reduced Instruction Set Computer* u Ordenador con Conjunto de Instrucciones Reducidas) de 32 bits, desarrollada por ARM Holdings [9]. Concebida originalmente por Acorn Computers para su uso en ordenadores personales, los primeros productos basados en ARM eran los Acorn Archimedes, lanzados en 1987. 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.” Basado en este procesador de Samsung surge la MBTI2440 [10].

“La MBTI2440 fue diseñada con el objetivo de obtener una tarjeta de procesamiento de bajo costo, pequeño tamaño, bajo consumo de potencia, gran variedad de dispositivos de interfaz, 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 [11].”

La MBTI2440 no posee conectores estándar de interconexión con los periféricos y dispositivos de interfaz. Por tal motivo, los usuarios para poder utilizarla deben hacerlo a través de una tarjeta base, en la cual se colocan los conectores requeridos [11].

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.

Inicialmente, la CID 300/9 funcionaba con un sistema operativo Linux, el cual no mostró ser estable, motivo por el cual se realizaron posteriores trabajos para solucionar dicho problema. Como parte de la búsqueda de una solución que permitiese el uso de esta placa, surge la idea de construir un sistema operativo GNU/Linux basado en la distribución cubana de GNU/Linux Nova, que permitiera el empleo de la CID 300/9 como un cliente ligero [11]. Esta solución no mostró ser efectiva, ya que para la construcción del sistema era necesario la compilación e instalación manual de cada una de las aplicaciones. Además, no contaba con las restricciones de tamaño del sistema operativo para en un futuro almacenarlo de forma interna en la CID 300/9.

Como parte de la búsqueda realizada para aumentar el dinamismo en la explotación de nuevas tecnologías en la CID 300/9 se encuentra Android, un sistema operativo que posee una marcada tendencia de uso a nivel mundial. La plataforma de hardware principal de Android es la arquitectura ARM, que es la que típicamente utilizan teléfonos inteligentes y tabletas [12], motivo por el cual se apuesta por la integración de Android con la computadora en una tarjeta. Aunque Android no se considere un sistema operativo embebido, si puede ser empotrado en todo tipo de sistemas que puedan ser considerados embebidos [12].

1.2. Android

Android es un sistema operativo basado en Linux, diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas, inicialmente desarrollados por Android Inc., que Google respaldó económicamente y más tarde compró en 2005 [13]. En diciembre de 2007 se estrenó Android como una plataforma de soporte para equipos móviles, utilizando la versión 2.6 del núcleo Linux [14]. Ese mismo año se le unió a Google la Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones entre las que están Texas Instruments, Broadcom Corporation, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, Intel, LG, Marvell Technology Group, Motorola, y T-Mobile aquí, para avanzar en los estándares abiertos de los dispositivos móviles [2]. El 9 de diciembre de 2008, se anunció que 14 nuevos miembros se unirían al proyecto Android, incluyendo PacketVideo, ARM Holdings, Atheros Communications, Asustek, Garmin, Softbank, Sony Ericsson, Toshiba, Vodafone y ZTE [15].

Android ha visto numerosas actualizaciones desde su liberación inicial. Estas actualizaciones al sistema operativo base, típicamente arreglan errores y agregan nuevas funciones. Generalmente cada actualización del sistema operativo Android es desarrollada bajo un nombre en código de un elemento relacionado con postres [16].

1.2.1. Características

- **Diseño de dispositivos:** La plataforma es adaptable a pantallas de mayor resolución, ya que utiliza Adaptador de Tarjeta Gráfica (VGA por sus siglas en inglés), biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
- **Almacenamiento:** SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
- **Conectividad:** Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+ y WiMAX.
- **Mensajería:** SMS y MMS son formas de mensajería, incluyendo mensajería de texto.

- **Navegador web:** El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador por defecto de Ice Cream Sandwich obtiene una puntuación de 100/100 en el test Acid3.
- **Soporte de Java:** Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El código independiente del hardware (*bytecode*) [17] de Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.
- **Soporte multimedia:** Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
- **Soporte para hardware adicionales:** Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, termómetro, aceleración por GPU 2D y 3D.
- **Entorno de desarrollo:** Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4, 3.5 o 3.6) usando programas para añadirle funcionalidades (*plugin*) [18] a las Herramientas de Desarrollo de Android.
- **Multi-táctil:** Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel, posiblemente para evitar infringir patentes de otras compañías. Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa.
- **Multitarea:** Las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj, a

diferencia de otros sistemas de la competencia en la que la multitarea es congelada, como por ejemplo: iOS, en el que la multitarea se limita a servicios internos del sistema y no a aplicaciones externas.

- **Características basadas en voz:** La búsqueda en Google a través de voz está disponible como “Entrada de Búsqueda” desde la versión inicial del sistema [19].

1.2.2. Arquitectura

A continuación se describe la arquitectura de los principales componentes del sistema operativo Android.

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** los desarrolladores tienen acceso completo a los mismos APIs del marco de trabajo (framework de ahora en lo adelante) usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Ejecutable (.dex), el cual está optimizado para memoria mínima. La

Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato .dex por la herramienta incluida "dx".

- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, gestión de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del software.

1.2.3. Diseño y desarrollo

Android, al contrario de otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar nuevos problemas.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del programa informático que se encuentra registrado en una memoria ROM, flash o similar (firmware) [20] para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria RAM y velocidad de procesador. En sus comienzos, Android era eminentemente un sistema operativo pensado para usar con teclado y gracias a un cursor poder navegar entre las aplicaciones. Desde su comienzo, Android ha sido altamente personalizable. Poco después, antes del lanzamiento del primer teléfono Android, esta filosofía cambió para convertirse en eminentemente táctil.

1.2.4. Aplicaciones

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, donde también es posible usar las bibliotecas Qt gracias al proyecto *Necesitas SDK*.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación.

Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente.

Todas las aplicaciones están comprimidas en formato APK (formato similar al jar de Java), que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

En una primera fase para ejecutar Android sobre la CID 300/9 se define el empleo de una solución de virtualización, la cual permita darle cumplimiento a los objetivos trazados en el proyecto Nova.

1.3. Virtualización

La virtualización consiste en la creación de un entorno virtual, sobre una máquina real, en el que se pueden ejecutar programas sobre otro sistema operativo, de manera independiente. Es decir, la virtualización es un programa que proporciona soporte para instalar otros sistemas operativos sobre la misma máquina, y que funcionarán de manera encapsulada dentro de ese entorno virtual. Habrá por tanto, un sistema físico que será la máquina real y un sistema virtual o lógico que se ejecutará virtualizado sobre el sistema físico. Y si se dispone de suficientes recursos se podrían tener varios sistemas virtuales ejecutándose sobre un único sistema físico. Y en sentido contrario también, es decir que varios recursos físicos (servidores o dispositivos de almacenamiento) aparecen como un único recurso lógico [21].

Esta tecnología permite la separación del hardware y el software, lo cual posibilita a su vez que múltiples sistemas operativos, aplicaciones o plataformas se ejecuten simultáneamente en un solo servidor o equipo, según sea el tipo de aplicación. El software que implementa la virtualización permite que el hardware ejecute múltiples instancias de diferentes sistemas operativos de forma concurrente sin que interfieran entre sí, ni con las aplicaciones. Por último, definiendo el concepto de máquina virtual como software que crea un entorno virtual entre la plataforma utilizada en la máquina real y el usuario final [21].

Ventajas:

- Simultáneamente los usuarios pueden trabajar con varios entornos diferentes e independientes.
- Si la estrategia de la organización es cambiar a menudo de aplicaciones, el uso de la virtualización permite conservar los puntos de trabajo idénticos y todos los cambios hacerlos sobre el entorno virtualizado y desde un servidor a través de la red.

- Facilita la realización de copias de seguridad y su recuperación.
- Optimiza la utilización de los recursos en aquellos servidores dedicados que proporcionan servicios sin grandes requerimientos.
- Otras ventajas serían la menor dependencia del hardware y la facilidad de cambio del puesto de trabajo.

En consecuencia se puede decir que la virtualización proporciona alta disponibilidad, ahorra costos, optimiza el uso y control sobre los recursos y mejora la seguridad de los sistemas virtualizados.

Desventajas:

- Problemas de licencias. La utilización de la virtualización supone un cambio en cuanto a las políticas de licencias por usuario.
- Utilización de recursos. El software de virtualización tiene unos requerimientos de hardware muy exigentes, sobre todo en cuanto a capacidad de proceso y de memoria RAM. Esto supone una pérdida de rendimiento.
- Existe una dependencia del sistema operativo anfitrión y del sistema de virtualización elegido. Es decir, el anfitrión limita y es el punto débil del sistema ya que está compartido por todos los sistemas virtualizados.

1.3.1. Virtualización de Android

Como parte del estudio realizado a soluciones de virtualización existentes se encuentran las soluciones “Virtual Smartphone over IP” mostrada en el 2010 y la “Virtualización de Android sobre la arquitectura x86” desarrollada en el proyecto Nova. Estas soluciones sirven como punto de partida para definir la nueva solución de virtualización.

- **“Virtual Smartphone over IP”**

Esta solución está concebida para la ejecución de imágenes virtuales desde la “nube” o servidor de virtualización en el smartphone físico. La utilización de los smartphone de forma virtual sobre el

protocolo de internet se realiza a través del consumo de imágenes virtuales almacenadas en la nube, permitiendo a los usuarios la ejecución remota de estas aplicaciones de forma local [22].

El sistema Virtual Smartphone over IP está compuesto por un número de clientes de teléfonos inteligentes externos, un servidor de interfaz (front-end de ahora en lo adelante), una granja de teléfonos inteligentes virtuales, un servidor de gestión y una red del sistema de archivos (NFS), donde la granja de teléfonos inteligentes virtuales es el componente más importante del sistema, esta alberga la colección de imágenes virtuales. El servidor front-end admite solicitudes por parte de los usuarios de teléfonos inteligentes a través de Internet y establece sesiones remotas en el teléfono inteligente virtual correspondiente. El servidor front-end también permite a los usuarios crear, configurar y destruir estas imágenes virtuales. Una vez que la sesión remota es establecida los usuarios pueden ejecutar e instalar aplicaciones en una de estas imágenes en lugar del teléfono físico.

El servidor de gestión se utiliza para administrar la granja de teléfonos inteligentes virtuales, el mismo permite las operaciones típicas de un servidor de administración, incluyendo la creación de imágenes virtuales y solución de problemas. Los usuarios controlan sus imágenes virtuales a través de una aplicación específica instalada en el teléfono físico, esta aplicación cliente recibe la salida de la pantalla de una imagen virtual y la presenta en la pantalla de forma local y a su vez, envía los eventos de hardware del teléfono físico a la imagen virtual para su procesamiento, funcionando como la tecnología de cliente ligeroada al reconectarse la imagen debe continuar en el mismo estado.

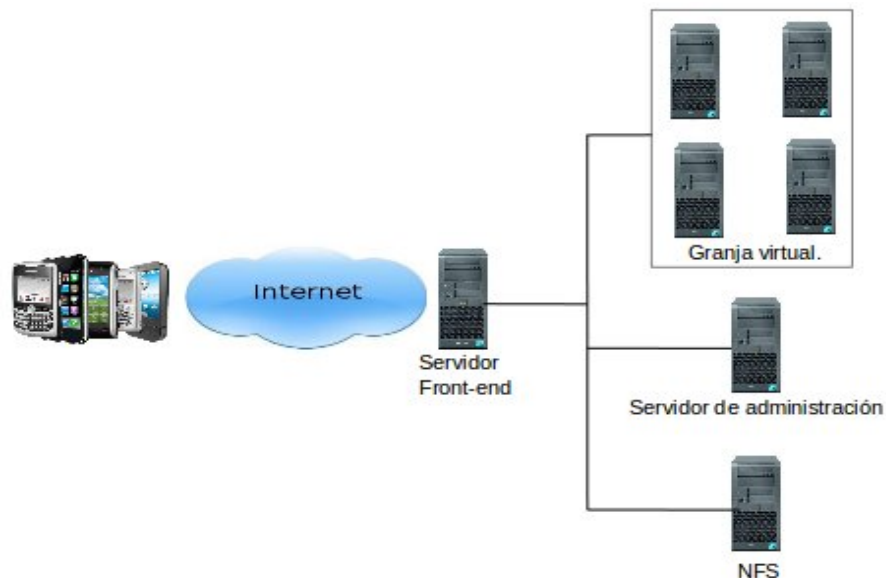


Figura 1.1: Vista general de la solución “Virtual SmartPhone over IP”

Para la implementación de esta solución se utiliza para establecer la conexión entre el smartphone físico y el servidor de virtualización el protocolo VNC, al cual se le agregan una serie de funcionalidades, entre las que se encuentran: importe de audio desde el servidor, acelerómetro, orientación, campo magnético y temperatura, entre otras.

Como software de virtualización se utiliza VM Ware para la ejecución de la imagen virtual en el servidor de virtualización y posterior consumo desde el smartphone local. Para el almacenamiento de los datos se emplea en protocolo NFS, que por ser fácilmente escalable brinda un almacenamiento prácticamente ilimitado. Este sistema de almacenamiento funciona como la tarjeta SD de un smartphone físico. Esta solución ofrece una serie de ventajas, las cuales se listan a continuación [22]:

- Almacenamiento
- Personalización
- Seguridad
- Control de la información
- Poder de cómputo

- Drenaje de la batería
 - Tráfico de red
- **“Virtualización de Android sobre la arquitectura x86”**

Esta solución ofrece una vía de ejecutar de forma remota Android sobre la arquitectura de computadoras x86. El prototipo posee el funcionamiento cliente-servidor, permitiendo el consumo de las imágenes del sistema operativo Android que se estén ejecutando en el servidor de virtualización.



Figura 1.2: Vista general de la virtualización de Android sobre la arquitectura x86

Para ejecutar las imágenes del sistema operativo Android almacenadas en el servidor se utiliza VirtualBox; un software de virtualización escogido por la potencia que posee, además de permitir la configuración de la resolución de pantalla en la imagen virtual ejecutada.

A la hora de establecer la comunicación entre el cliente y el servidor de virtualización se emplean los protocolos VNC y RDP, para el envío de los eventos de teclado y mouse desde el cliente al servidor y el envío de actualizaciones gráficas desde el servidor de virtualización hasta el cliente. En esta solución el protocolo RDP mostró mejores resultados ya que con el protocolo VNC no se logró importar el audio.

1.4. Software de Virtualización

En la actualidad existe una alta y variada gama de softwares especializados en la virtualización de arquitecturas de computadoras, estos permiten la creación de un entorno virtual sobre una máquina física, lo que es útil a la hora de implementar una solución cubana por sus características y particularidades. A continuación se hará mención de algunos de ellos con sus características particulares.

1.4.1. VirtualBox

VirtualBox es un software de virtualización para arquitecturas x86 que fue desarrollado originalmente por la empresa alemana Innotek GmbH, pero que pasó a ser propiedad de la empresa Sun Microsystems en febrero de 2008 cuando la compró a Innotek. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como “sistemas invitados”, dentro de otro sistema operativo “anfitrión”, cada uno con su propio ambiente virtual. Por ejemplo, se pueden instalar diferentes distribuciones de GNU/Linux en VirtualBox instalado en Windows XP o viceversa.

Entre los sistemas operativos soportados (en modo anfitrión) se encuentran GNU/Linux, Mac OS X, OS/2 Warp, Windows, y Solaris/OpenSolaris, y dentro de éstos es posible virtualizar los sistemas operativos FreeBSD, GNU/Linux, OpenBSD, OS/2 Warp, Windows, Solaris, MS-DOS y muchos otros.

Virtualbox provee funcionalidades como la ejecución de máquinas virtuales de forma remota, por medio del Remote Desktop Protocol (RDP), soporte iSCSI [23].

Ventajas de VirtualBox

Además de las ventajas descritas anteriormente, las que sin duda hacen por si solas a VirtualBox un software potente, se pueden mencionar otras ventajas que son de suma importancia para los usuarios finales, algunas de estas son [24]:

- Posee licencia GPL (*General Public Licence*).
- Actualización constante y cada vez más compatible.
- Virtualiza diferentes Sistemas Operativos en distintas versiones o distribuciones (dependiendo del S.O.).
- Tiene documentación completa.
- Configuración de red con distintas opciones.
- Resolución de pantalla configurable.
- Compatible para distintas arquitecturas.

- Muy estable al momento de virtualizar el Sistema Operativo.

Desventajas de VirtualBox

Al usar VirtualBox se debe de tener presente que se está utilizando un sistema operativo habitual o físico y sobre él se ejecuta otra súper aplicación, como es otro sistema operativo utilizando la herramienta VirtualBox. Esta acción puede afectar de cierta forma el rendimiento del ordenador, esto marca una desventaja en la utilización de VirtualBox. A continuación se mencionan algunas de estas desventajas [25]:

- No reconoce algunas distribuciones de Linux que son poco conocidas.
- El rendimiento teórico de una máquina virtual no va a ser igual que el de una instalación en una partición.

1.4.2. VMware Workstation

VMware Inc., (VM de Virtual Machine) filial de EMC Corporation que proporciona la mayor parte del software de virtualización disponible para ordenadores compatibles X86. Entre este software se incluyen VMware Workstation, y los gratuitos VMware Server y VMware Player. El software de VMware puede funcionar en Windows, Linux, y en la plataforma Mac OS X que corre en procesadores INTEL, bajo el nombre de VMware Fusion. El nombre corporativo de la compañía es un juego de palabras usando la interpretación tradicional de las siglas «VM» en los ambientes de computación, como máquinas virtuales (Virtual Machines) [26].

VMware Workstation es una de las herramientas de virtualización más potentes en la actualidad, imprescindible para correr varias versiones de Windows y Linux en un ordenador sin reparticionar el disco ni reiniciar el ordenador.

Con la utilización de VMware Workstation se pueden recrear las características técnicas de un ordenador e instalar un sistema operativo. Cualquier problema que ocurra durante este proceso o posterior al mismo no llegará a afectar al Sistema instalado en el ordenador físico.

Ventajas de VMware Workstation

- Modo en que integra los dos sistemas
- Comparte máquinas virtuales en red
- Capturas de imágenes y vídeos
- Asistente de instalación de máquinas

Desventajas de VMware Workstation

- Requiere un equipo potente
- No es compatible con todos los sistemas operativos

Vistas ya las principales características, ventajas y desventajas de esta potente herramienta se concluye que no sería óptimo utilizarla puesto al requerimiento de hardware que trae asociada, causaría incompatibilidad a la hora de utilizar hardware de bajas prestaciones.

1.4.3. Qemu

Qemu es un programa que ejecuta máquinas virtuales dentro de un sistema operativo, ya sea Linux, Windows u otro sistema operativo. Esta máquina virtual puede ejecutarse en cualquier tipo de Microprocesador o arquitectura (x86, x86-64, PowerPC, MIPS, SPARC, ARM, entre otras). Está licenciado en parte con la LGPL y la GPL de GNU [27].

Qemu emula el funcionamiento de varios procesadores y utiliza la traducción dinámica para ganar en rapidez. Puede utilizarse para lanzar distintos sistemas operativos sin reiniciar el PC o para depurar el código. En el otro modo, llamado usuario, Qemu puede lanzar procesos compilados de una CPU a otra. ¿Su utilidad? Por ejemplo, lanzar el emulador Wine Windows API o facilitar la compilación y depuración cruzadas [28].

El programa no dispone de interfaz gráfica, pero existe otro programa llamado QEmu Manager que hace la función de interfaz gráfica si se utiliza Qemu desde Windows. También existe una versión para Linux llamado QEmu-Launcher, en Mac OS X puede utilizarse el programa Q que dispone de una interfaz gráfica para crear y administrar las máquinas virtuales.

Ventajas de Qemu

- Soporte para una gran variedad de arquitecturas, tanto en el host, como en la máquina emulada.
- Implementa el modo Copy-On-Write de formato de imagen de disco. Se puede declarar una imagen de disco de varios gigas y realmente ocupara en disco el tamaño que se esté utilizando
- El sistema host no ha de ser parcheado.
- Lleva integrado un servidor VNC para controlar remotamente las máquinas virtuales.

Desventajas de Qemu

- Usado sobre arquitectura x86 es menos eficiente que otras soluciones como VMWare y VirtualBox.
- Más difícil de instalar y ejecutar que otras soluciones.

1.5. Protocolos de comunicación remota

Los protocolos son reglas de comunicación que permiten el flujo de información entre computadoras distintas que manejan lenguajes distintos, por ejemplo, dos computadoras conectadas en la misma red pero con protocolos diferentes no podrían comunicarse, para ello, es necesario que ambas “hablen” el mismo idioma, es decir, utilicen el mismo protocolo de comunicación [29]. Los protocolos de comunicación remota permiten la comunicación con otras computadoras de la misma red o conectadas a Internet. Estos protocolos permiten que por un mismo canal de información sea enviada simultáneamente cualquier información [30].

1.5.1. Protocolo VNC

El protocolo de Computación de Redes Virtuales o *Virtual Network Computing* (VNC de ahora en lo adelante) se creó en Cambridge, UK Olivetti & Oracle Research Lab (ORL), que fue poseído entonces por Olivetti y Oracle Corporation. En 1999 AT&T adquirió el laboratorio, y en 2002 cerró los esfuerzos de investigación del laboratorio. Después del cierre de ORL en 2002, varios miembros del equipo de desarrollo formaron RealVNC a fin de seguir trabajando en código abierto y software VNC, comercial bajo ese nombre.

Las actualizaciones de pantalla se hacen enviando las partes de la pantalla que hayan cambiado, enviando solo pequeños rectángulos con las variaciones. Esto funciona muy bien cuando los cambios en pantalla son pocos, pero cuando hay grandes variaciones, por ejemplo, cuando se arrastra una ventana por la pantalla. Para evitar tener que enviar tantos datos por la red y acelerar la tasa de refresco, el protocolo VNC puede utilizar diferentes tipos de compresión en las actualizaciones que se envían.

Por lo que respecta a la seguridad del protocolo, la contraseña no se envía en texto plano, como en el caso de Telnet, sino que se utiliza un sistema llamado desafío-respuesta, en el que el servidor envía un número aleatorio al cliente, el cual lo cifra con la contraseña y se lo devuelve, con lo que el servidor puede comprobar si esta contraseña es correcta. El resto del tráfico que se transmite si se hace en texto plano, por lo que un atacante que escuche el tráfico puede obtener los datos que se envían [31].

Un sistema VNC consiste en un cliente, un servidor y un protocolo de comunicación:

- El servidor VNC es el programa en la máquina que comparte su pantalla. El servidor pasivamente permite que el cliente tome el control de ello.
- El cliente VNC (o espectador) es el programa que mira, controla y se relaciona con el servidor.
- El protocolo VNC es muy simple, basado en una obra primitivista gráfica del servidor al cliente ("Puesto un rectángulo de datos del píxel especificado en la posición X, Y ") y

mensajes del acontecimiento del cliente al servidor.

En condiciones ideales, VNC puede ser utilizado para que muchos ordenadores de baja potencia puedan acceder a un servidor de aplicaciones con altas prestaciones. De esta manera en un sistema multiusuario, como Linux, cada persona puede visualizar simultáneamente un perfil único de trabajo (Pantalla, programas, espacio en disco, seguridad, así como otras funcionalidades) [31].

1.5.2. Protocolo RDP

Protocolo de Escritorio Remoto (RDP de ahora en lo adelante) es un protocolo propietario desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre un terminal (muestra la información procesada que recibe del servidor) y un servidor (recibe la información dada por el usuario mediante el ratón o el teclado).

El modo de funcionamiento del protocolo es sencillo. La información gráfica que genera el servidor es convertida a un formato propio RDP y enviada a través de la red al terminal, que interpretará la información contenida en el paquete del protocolo para reconstruir la imagen a mostrar en la pantalla del terminal. En cuanto a la introducción de órdenes en el terminal por parte del usuario, las teclas que pulse el usuario en el teclado del terminal así como los movimientos y pulsaciones de ratón son redirigidos al servidor, permitiendo el protocolo un cifrado de los mismos por motivos de seguridad. El protocolo también permite que toda la información que intercambien cliente y servidor sea comprimida para un mejor rendimiento en las redes menos veloces [32].

Este servicio utiliza por defecto el puerto TCP 3389 en el servidor para recibir las peticiones. Una vez iniciada la sesión desde un punto remoto el ordenador servidor mostrará la pantalla de bienvenida, no se verá lo que el usuario está realizando de forma remota.

Características:

- Permite el uso de colores de 8, 16, 24 y 32 bits.
- Cifrado de 128 bits utilizando el algoritmo criptográfico RC4. Los clientes más antiguos pueden utilizar cifrados más débiles.

- El redireccionamiento del audio permite al usuario ejecutar un programa de audio en una ventana remota y escuchar el sonido en el ordenador local.
- El redireccionamiento de puertos permite utilizar los puertos serie y paralelo directamente.
- Ancho de banda ajustado para clientes RDP.

1.6. Protocolo NFS

El protocolo de Sistema de Ficheros en Red (NFS de ahora en lo adelante) fue desarrollado por SUN Microsystems en 1984. Su función en una red es permitir que un equipo GNU/Linux pueda montar y trabajar con un sistema de archivos de otro equipo de la red como si fuera local [33].

NFS permite compartir directorios y ficheros a través de la red. Los usuarios del sistema NFS pueden acceder a ficheros que se encuentran físicamente en máquinas remotas de una forma transparente, como si se tratara de ficheros locales [34].

NFS asume un sistema de ficheros jerárquico (directorios). Los ficheros son secuencias de bytes sin estructura y carentes de significado inherente: es decir, todos los ficheros se ven como una secuencia contigua de bytes, sin ninguna estructura de registros.

Con NFS, todas las operaciones sobre ficheros son síncronas. Esto significa que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esa operación. En caso de una solicitud de escritura, el servidor escribirá físicamente los datos en el disco, y si es necesario, actualizará a estructura de directorios, antes de devolver una respuesta al cliente. Esto garantiza la integridad de los ficheros.

El protocolo NFS tiene múltiples usos prácticos. Los más típicos se enumeran a continuación:

- Compartición de la unidad de CDROM entre varias máquinas.
- En grandes redes puede ser más adecuado configurar un servidor central de NFS en el cual se almacenen todos los “homes” de los distintos usuarios [35].

1.7. 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 uno de los lenguajes de modelado de procesos más conocidos y utilizados en la actualidad. Está respaldado por el OMG (Object Management Group o Grupo de Gestión de Objetos). UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Además, ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

El principal objetivo de BPMN es proveer una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio (stakeholders). Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorizan y gestionan los procesos). En síntesis BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación [36].

1.8. Conclusiones parciales

Con este capítulo fueron sentadas las bases teóricas de la aplicación. Una vez analizados los diferentes software de virtualización y los protocolos de comunicación, definiéndose como software de virtualización a VirtualBox y a RDP como protocolo de comunicación. Para el modelado de procesos se define a UML y BPMN.

Capítulo 2

Capítulo 2 Diseño del Sistema

En el presente capítulo se define la arquitectura del sistema, enfocada a dar cumplimiento a los requisitos del mismo. Se abordan aspectos generales del funcionamiento del sistema, también se tratan aspectos relacionados con los subprocesos definidos.

2.1. Funcionamiento del Sistema.

El proceso de virtualización de Android sobre la arquitectura ARM se realiza a través de la conexión del cliente con el servidor de virtualización, que será donde se esté ejecutando una imagen virtual del sistema operativo Android. Desde el cliente, con la utilización de aplicaciones de conexión remota se consumen los servicios desde el servidor, permitiendo obtener en la pantalla del cliente las actualizaciones gráficas enviadas desde el servidor de virtualización. Este funcionamiento consiste en la utilización de un protocolo de conexión remota entre el cliente y el servidor, el cual tiene la funcionalidad de enviar los eventos de hardware o pulsaciones de teclado y mouse desde el cliente al servidor y el envío de actualizaciones gráficas desde el servidor hacia el cliente, este proceso se muestra en la siguiente figura.

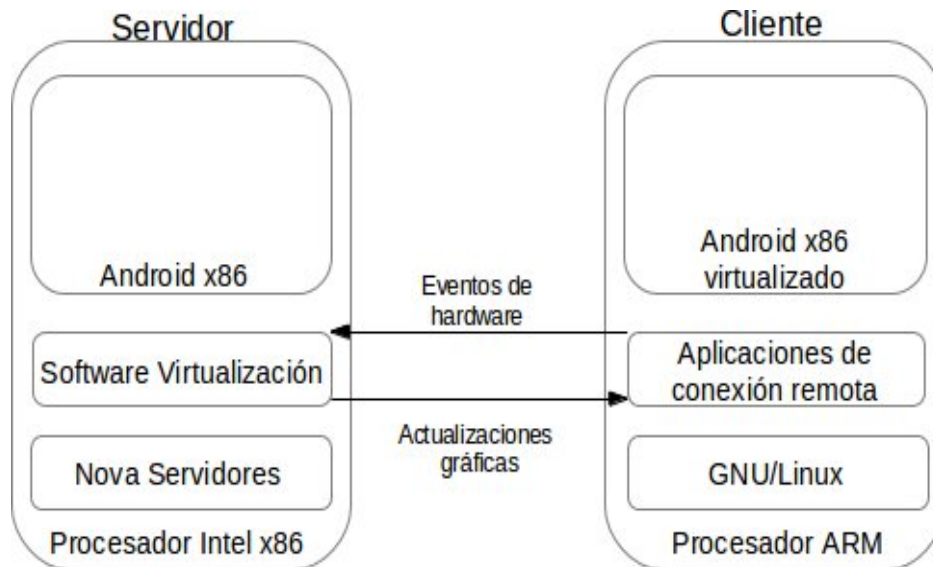


Figura 2.1: Vista general del funcionamiento del sistema

2.1.1. Funcionamiento del Cliente

Para el correcto funcionamiento del cliente es necesario tener en cuenta una serie de aspectos que son relevantes, estos son los subprocesos que se realizan en el cliente para obtener un funcionamiento fctén. Los subprocesos relacionados con el funcionamiento del cliente son: inicio del núcleo, inicio del sistema y conexión al servidor.

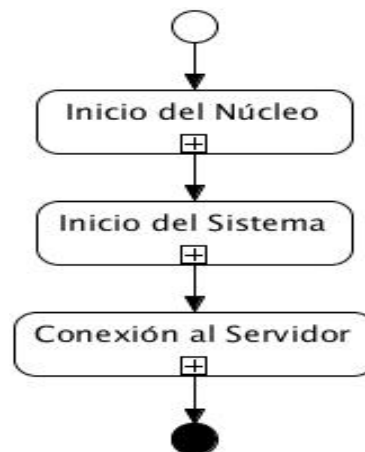


Figura 2.2: Vista general del funcionamiento del cliente

Cada uno de estos subprocesos realiza una función importante y específica, a continuación se mencionan las más relevantes.

Inicio del Núcleo

El núcleo del sistema, es el primer elemento que se inicia al encender la 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.

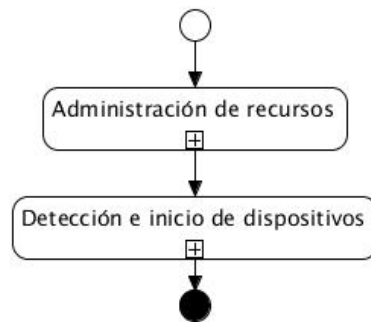


Figura 2.3: Vista general del Inicio del Núcleo

Inicio del Sistema

Una vez terminado el inicio del núcleo, este le delega 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.

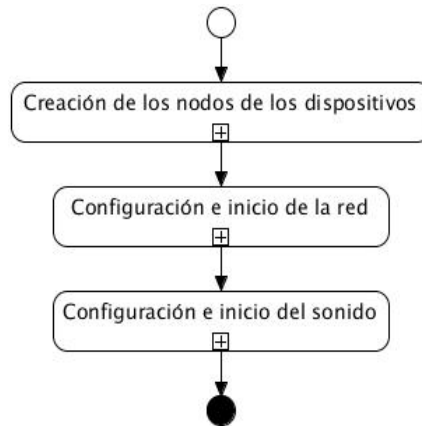


Figura 2.4: Vista general del Inicio del Sistema

Conexión al Servidor

Al terminar de iniciar las funcionalidades básicas del sistema, se configuran e inician un conjunto de aplicaciones que dan soporte a la funcionalidad de cliente ligero al sistema. Según el protocolo de comunicación configurado por el administrador del sistema, se crea una conexión entre el servidor de virtualización y el 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 el terminal cliente el escritorio compartido en el servidor. A su vez, son enviados los paquetes de sonido desde el servidor al terminal cliente, mediante la utilización de una herramienta que funciona como servidor de sonido, y crea un puente de comunicación entre las dos computadoras. La terminal cliente por su parte, envía los eventos realizados por los periféricos (mouse y teclado) al servidor.



Figura 2.5: Vista general de la conexión al Servidor

2.2. Requerimientos del Sistema

Un requerimiento es una condición o capacidad que debe exhibir o poseer un sistema para satisfacer un contrato, estándar, especificación u otra documentación formalmente impuesta. Los requerimientos del sistema son descripciones de servicios y restricciones. Los requerimientos deben de cumplir con varias características, entre las que se encuentran: necesario, no ambiguo, conciso, consistente, completo, alcanzable y verificable [37].

2.2.1. Requerimientos del Cliente

Se construirá una distribución de GNU/Linux basada en Nova, para ser instalada en la CID 300/9, esta debe cumplir con los siguientes requisitos:

- **Requerimientos Funcionales:**

1. Permitir la conexión remota al servidor de virtualización
2. Enviar los eventos del teclado al servidor
3. Enviar los eventos del mouse
4. Recibir las actualizaciones gráficas del servidor
5. Recibir sonido del servidor
6. Permitir el trabajo con USB conectadas en el cliente

- **Requerimientos no Funcionales:**

- Restricción de diseño e implementación

- Implementar la solución utilizando la CID 300/9

- Personalizar el sistema bajo los estándares de Nova

- Personalizar el núcleo para que utilice como imagen de arranque el logo de Nova

- Requerimiento de software

- El tamaño del sistema no debe sobrepasar los 60 MB

2.2.2. Requerimientos del Servidor

- **Requerimientos Funcionales:**

1. Virtualizar diferentes imágenes de Android
2. Asignar la imagen correspondiente a cada cliente
3. Enviar actualizaciones gráficas al cliente
4. Transportar sonido al cliente
5. Permitir el trabajo con USB conectadas en el cliente

- **Requerimientos no Funcionales:**

- Restricción de diseño e implementación

- Implementar la solución utilizando Nova para Servidores

2.3. Arquitectura del Sistema

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

1. Núcleo
2. Sistema Base
3. Aplicaciones de conexión remota

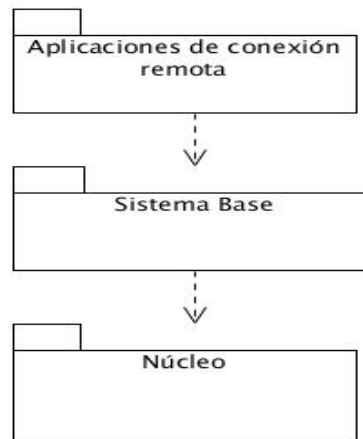


Figura 2.6: Vista general de la arquitectura del Sistema

2.3.1. 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.

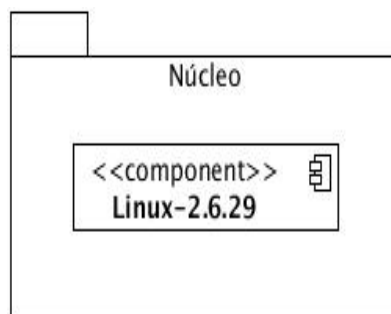


Figura 2.7: Vista general del núcleo del Sistema

2.3.2. Sistema Base

El sistema base de la distribución contiene los paquetes esenciales del sistema. Estos paquetes comprenden lo necesario para el arranque del sistema, detección de dispositivos, configuración de la red, trabajo básico con ficheros, entre otros. A continuación se describen los elementos

primordiales:

- Udev: Se encarga de crear los nodos de los dispositivos físicos del sistema.
- BusyBox: Permitirá la ejecución de archivos binarios.

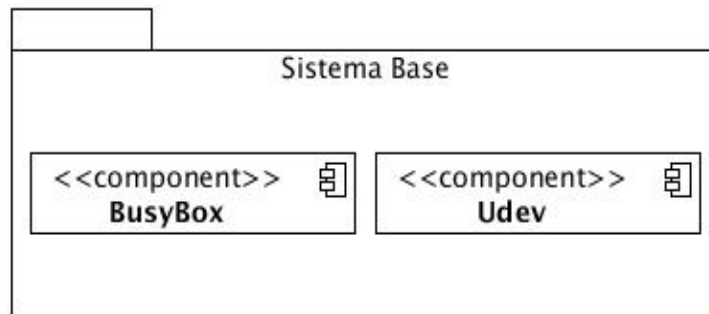


Figura 2.8: Vista general de los componentes del Sistema

2.3.3. Aplicaciones de conexión remota

En esta capa se encuentran las herramientas necesarias para establecer la conexión entre el cliente y el servidor de virtualización.

Los elementos primordiales se listan a continuación:

- Xorg: Servidor gráfico que permite visualizar en el cliente ligero las aplicaciones ejecutadas en el servidor.
- Rdesktop: Permite realizar la conexión a un servidor de cliente ligero mediante el protocolo RDP.

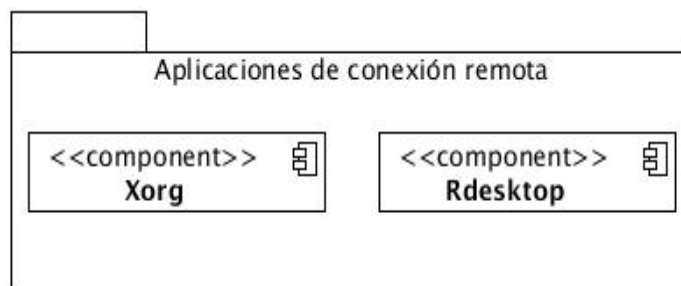


Figura 2.9: Vista general de las aplicaciones de conexión remota

2.4. Arquitectura del Servidor

La arquitectura del Servidor está compuesta por un Software de Virtualización, en este caso VirtualBox, donde este está relacionado con un grupo de imágenes virtuales del Sistema Operativo Android. A continuación se representa la arquitectura del servidor.

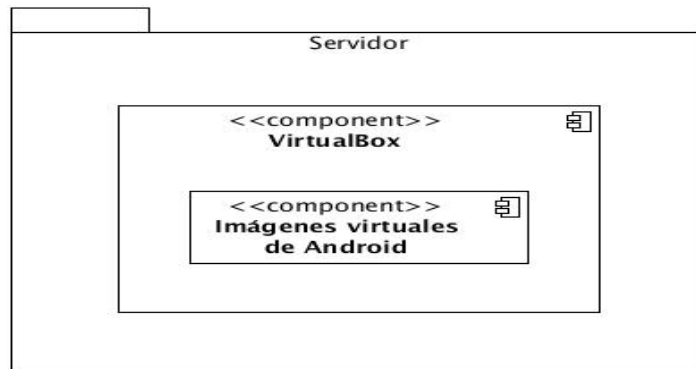


Figura 2.10: Vista general de arquitectura del Servidor de Virtualización

2.5. Conclusiones parciales

En este capítulo fue diseñado el sistema para su exitosa implementación. Fue de interés de estudio, la captura y análisis de los requerimientos del sistema. Además se describe el proceso de funcionamiento del sistema, así como cada uno de sus componentes. Se define una arquitectura en capas para el sistema, así como los elementos notables que se encuentran en cada uno de ellos.

Capítulo 3

Capítulo 3 Construcción del Sistema

En el presente capítulo se describe el proceso de construcción de cada una de las capas definidas en la arquitectura.

Para llevar a cabo el proceso de construcción del sistema se utilizará la distribución cubana de GNU/Linux Nova, en su versión 4.0. Para contar con una organización en el proceso de construcción del sistema se crean una serie de directorios, los cuales contendrán las herramientas y aplicaciones necesarias.

```
$ mkdir /media/Datos/Trabajo
$ chmod 777 /media/Datos/Trabajo
$ cd /media/Datos/Trabajo
$ mkdir Herramientas Kernel Sistema
```

Trabajo: En este directorio se guardan todos los cambios ocurridos durante el proceso de construcción del sistema.

Herramientas: Contiene todas las herramientas utilizadas en el proceso de construcción del sistema.

Kernel: Aquí se encuentra el código fuente de los núcleos utilizados.

Sistema: Se almacena el sistema construido.

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 I**.

- Minicom
- CodeSourcery
- VirtualBox
- Buildroot
- Qemu
- NFS Server

3.1. Núcleo

El kernel a utilizar será el kernel Linux en su versión 2.6.29, esta no es la versión oficial del kernel, sino una modificada por el personal del ICID para adaptarlo a la computadora en una tarjeta.

Descargar el código fuente en la carpeta Kernel.

Configurando y compilando el kernel 2.6.29

```
$ cd /media/Datos/Trabajo/Kernel
$ tar -xjf linux-2.6.29.tar.bz2
$ cd linux-2.6.29
```

Aplicando la configuración por defecto.

```
$ make ARCH = arm versatile_defconfig
$ make menuconfig
```

Seleccionando las opciones de configuración

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: 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

<*> 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

[*] 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/mmcbk0p1 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**

Compilar:

```
$ make all
$ make targz-pkg
```

Una vez concluida la compilación se ha generado una imagen (zImage) del kernel, la cual será utilizada para crear el sistema de ficheros y grabarla en la memoria interna de la CID 300/9. Esta imagen se encuentra en el directorio *boot*.

```
$ cd arch/arm/boot
```

Con el objetivo de conservar la configuración utilizada para la compilación de este kernel, y utilizarla para la creación del sistema de ficheros es guardada en otro fichero.

```
$ cd /media/Datos/Trabajo/Kernel/linux-2.6.29
$ cp .config config_kernel
```

3.2. Sistema de Ficheros

Para la creación del sistema de fichero se utiliza la herramienta Buildroot en su versión 2011.02. Esta herramienta permite la creación de un sistema de ficheros a partir de paquetes definidos.

Descargar el código fuente dentro de la carpeta *Herramientas*.

Configurando y compilando el sistema de ficheros.

```
$ cd /media/Datos/Trabajo/Herramientas/buildroot-2011.02
```

Limpiando compilaciones anteriores.

```
$ make distclean
```

Crear directorio dl, esta carpeta contendrá los paquetes necesarios para la construcción del sistema.

```
$ mkdir dl
```

Copiar paquetes a la carpeta dl. La carpeta **repository** contiene una salva de los paquetes necesarios para la construcción del sistema.

```
$ cp ../repository/* dl
```

Antes de comenzar el proceso de selección de cada una de las aplicaciones que serán compiladas es necesario copiar la configuración generada por el proceso de configuración del kernel para la carpeta **buildroot-2011.02**.

```
$ cp /media/Datos/Trabajo/Kernel/linux-2.6.29/config_kernel config_kernel
```

Ejecutar interfaz de configuración.

```
$ make menuconfig
```

Seleccionar las siguientes opciones:

Como arquitectura se selecciona **arm** que es la arquitectura de computadora utilizada por la CID 300/9. Como variante de la arquitectura arm se selecciona **arm920t** ya que la Computadora en una Tarjeta está basada en esta versión de arquitectura de computadora.

```
Target Architecture - - -> arm
Target Architecture Variant - - -> arm920t
```

Para lograr compatibilidad con el sistema operativo se utiliza el estándar EABI.

Target ABI - - ->

EABI

Ahora es necesario la utilización de una herramienta de compilación cruzada (toolchain), en el caso de estudio se utiliza *arm-2010.09-50-arm-none-linux-gnueabi*, para el correcto funcionamiento se especifica la dirección donde se encuentra, así como el prefijo que utiliza.

Toolchain - - - - ->Toolchain type - - ->

External Toolchain

Toolchain - - - - ->Toolchain - - ->

Custom toolchain

Toolchain path - - ->

/home/danier/CodeSourcery/Sourcery_G++_Lite

Toolchain - - - - ->Toolchain prefix - - ->

\$(ARCH)-none-linux-gnueabi

Toolchain - - - - ->External toolchain C library - - ->

glibc

Toolchain - - ->

Enable MMU support

Para establecer la comunicación a través de la puerta serial es necesario especificar el puerto serial, en este caso sería **SAC0**.

System configuration - - ->

Generic serial port config

System configuration - - ->Generic serial port config - - ->Serial port to run a getty on - - ->

ttySAC0

System configuration - - ->Generic serial port config - - ->Baudrate to use - - ->

115200

En esta sección se especifica que el sistema utiliza a **Xorg** como servidor gráfico, a **MatchBox** como manejador de ventanas, a **rdesktop** como herramienta para la comunicación con el servidor de virtualización, además se selecciona la fuente a utilizar.

Package Selection for the target- - ->

BusyBox

Package Selection for the target - - ->Graphic libraries and applications - - ->

X.org X Window System, X11R7

Liberation (Free fonts)

MatchBox Window Manager)

rdesktop)

xterm)

Package Selection for the target - - ->Graphic libraries and applications - - ->X.org

X Window System, X11R7 - - ->X Window System server type- - ->

Modular X.org

En esta sección se seleccionan las aplicaciones para el servidor gráfico, estas opciones especifican el inicio del servidor, reloj y soporte para dispositivos de entrada/salida.

Package Selection for the target - - ->Graphic libraries and applications - - ->X.org

X Window System X11R7 - - ->X11R7 Applications - - ->

setxkbmap

twm

xclock

xinit

xinput

Una vez seleccionadas las aplicaciones para el servidor gráfico es necesario la incorporación al sistema de los controladores necesarios para darle soporte a estas aplicaciones.

Package Selection for the target - - ->Graphic libraries and applications - - ->X.org

X Window System X11R7 - - ->X11R7 Drivers - - ->

- [*] xf86-input-evdev
- [*] xf86-input-keyboard
- [*] xf86-input-mouse
- [*] xf86-video-cirrus
- [*] xf86-video-fbdev
- [*] xf86-video-vesa

Es necesario la inclusión de **Udev** para el manejo y control de los dispositivos incluidos en el kernel.

Package Selection for the target - - ->Hardware handling - - ->

- [*] udev

Como opciones para crear el sistema de ficheros se seleccionan las siguientes.

Target filesystem options - - ->

- [*] tar the root filesystem
- [*] cpio the root filesystem
- [*] initramfs for initial ramdisk of linux kernel

Para configurar el sistema de ficheros para el kernel que se utiliza en el caso de estudio es necesario modificar una serie de opciones. Aquí es necesario especificar la versión del kernel, además de la configuración que se utilizará para el kernel especificado. En este caso se utilizará la configuración generada en el proceso de compilación del núcleo (*config_kernel*).

```
Kernel - - ->
[*] Linux Kernel
Kernel - - ->Kernel version - - ->
(x) Custom version
Kernel - - ->Kernel configuration - - ->
(x) Using a custom config file - - -> config_kernel
Kernel - - ->Kernel binary format - - ->
(x) zImage
```

Compilar.

```
$ make
```

3.3. Administración del servidor de virtualización

En el servidor de virtualización será donde se esté ejecutando la imagen virtual del sistema operativo Android. Para esto se utiliza VirtualBox como software de virtualización, a continuación se explican los pasos para la instalación, configuración y ejecución de las estas imágenes:

Comprobar versión del kernel utilizada en el sistema anfitrión.

```
$ uname -a
```

Instalar módulos del kernel.

```
$ sudo apt-get install linux-headers-3.2.0-38-escritorio
```

Instalar VirtualBox.

```
$ sudo apt-get install virtualbox
```

Crear y registrar la nueva máquina virtual. En este caso se nombrará Android.

```
$ VBoxManage createvm --name "Android" --register
```

Especificar tamaño de memoria, modo de boteo y adaptador de red.

```
$ VBoxManage modifyvm "Android" --memory 512 --acpi on --boot1 dvd --nic1 bridged --bridgeadapter1 eth0
```

Crear el disco duro y asignarle el tamaño.

```
$ VBoxManage createhd --filename Android.vdi --size 10000
```

Especificarle el id del controlador.

```
$ VBoxManage storagectl "Android" --name 'IDE Controller' --add ide
```

Especificar puerto y tipo de dispositivo.

```
$ VBoxManage storageattach "Android" --storagectl "IDE Controller" --port 0 --device 0 --type hdd --medium Android.vdi
```

Especificar imagen ISO a utilizar.

```
$ VBoxManage storageattach "android" --storagectl "IDE Controller" --port 1 --device 0 --type dvddrive --medium /media/Datos/Trabajo/Isos/droid-20121212.iso
```

Una vez realizadas estas operaciones la máquina virtual está creada, ahora para ponerla en marcha se ejecuta lo siguiente:

```
vboxheadless -startvm <nombre_máquina_virtual>
```

Se sustituye <nombre_máquina_virtual> por el nombre con que se creó la máquina. En este caso se nombró **Android**, por lo que quedaría así:

```
$ vboxheadless -startvm Android
```


3.4. Asignación de imágenes virtuales

Para asignarle una imagen del sistema operativo Android a un determinado cliente se le cambia el puerto de conexión de esa imagen.

Se modifica el parámetro *vrdeport*, este por defecto utiliza el puerto 3389.

```
$ VBoxManage modifyvm "<nombre_máquina_virtual>" -vrdeport <puerto>
```

Se sustituye *<nombre_máquina_virtual>* por el nombre de la máquina virtual que se quiera modificar y *<puerto>* por el puerto por el cual se quiere que se comunique.

```
$ VBoxManage modifyvm "Android" -vrdeport 5000
```

Una vez realizado esto en el cliente de virtualización se especificará el puerto de conexión.

```
$ rdesktop <ip_servidor>:<puerto>
```

```
$ rdesktop 10.53.7.235:5000
```

3.5. Emulación del sistema

Una vez construido el sistema se emulará para comprobar su correcto funcionamiento. Para esto serán necesarias las herramientas Qemu y NFS Server, la primera será utilizada para emular el sistema construido y la segunda para compartir el sistema construido.

Para el correcto funcionamiento del sistema a la hora de emular se configurará la red, este proceso será explicado en el **ANEXO II**. Una vez configurada la red hará falta exportar el sistema creado y crear la red virtual. Para poder emular el sistema es necesario tener el sistema de ficheros construido en el directorio */trabajo/sistema* y la imagen del kernel en */trabajo*.

```
$sudo cp <directorio_del_buildroot>/output/target/* /trabajo/sistema
```

```
$ sudo cp <directorio_del_buildroot>/images/zImage /trabajo
```

```
$ cd /trabajo
```

```
# ./red
```

```
$ sudo qemu-system-arm -M versatilepb -m 256M -kernel zImage -append  
"root=/dev/nfs nfsroot=10.53.7.251:/trabajo/sistema rw ip=10.53.7.25 -initrd init"  
-net nic -net tap,ifname=tap1,script=no
```

3.6. Grabar Sistema de Ficheros creado a 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. Se monta el dispositivo, se elimina todo el contenido que tenga, se copia todo el contenido del sistema y se desmonta el dispositivo.

```
$ sudo rm -rf /media/CID/*
```

```
$ cd /media/Datos/Trabajo/Herramientas/buildroot-2011.02/output/target
```

```
$ sudo rsync -av . /media/cid
```

```
$ umount /media/cid
```

Concluido el proceso de grabación del sistema de ficheros a la tarjeta SD es necesario cargar la imagen del kernel Linux compilado, este proceso es explicado en el ANEXO III.

3.7. Conclusiones parciales

En el presente capítulo fue implementado un sistema que permite la virtualización de Android sobre la Computadora en una Tarjeta. El proceso de construcción de la solución de virtualización transitó de forma general por 6 etapas. En la primera se configuró y compiló el núcleo a utilizar, en la segunda se construyó el sistema de ficheros, en las dos siguientes se implementó el servidor de virtualización, posteriormente se emuló el sistema para comprobar su correcto funcionamiento y por último se grabó en la computadora real.

Capítulo 4

Capítulo 4 Pruebas al Sistema.

Una vez concluida la construcción de la distribución para el sistema, 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

Las actividades de esta etapa se refieren a hacer chequeos completos respecto a las funcionalidades y aplicaciones que posee el sistema.

Tabla 4.1: *Inicio del sistema*

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

Tabla 4.2: *Inicio del servidor gráfico*

Escenario	Descripción	Respuesta del sistema	Flujo central
Iniciar Xorg	Iniciar el servidor gráfico.	El sistema operativo levanta el servidor gráfico.	Ejecutando startx

Condiciones de ejecución: El sistema debe estar iniciado correctamente

Tabla 4.3: *Autenticación*

Escenario	Descripción	Respuesta del sistema	Flujo central
Autenticarse en el sistema	Autenticarse en el sistema utilizando usuario y contraseña.	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 la contraseña, impedir la entrada al sistema.	Insertar usuario y contraseña

Tabla 4.4: *Conexión al servidor de virtualización mediante el protocolo RDP.*

Escenario	Descripción	Respuesta del sistema	Flujo central
	Se realiza la conexión al escritorio compartido del servidor de virtualización mediante el protocolo RDP.	Se muestra el escritorio compartido en el servidor de virtualización.	Ejecutando: rdesktop <ip>:<puerto> Sustituyendo <ip> por la dirección IP del servidor y <puerto> por el puerto utilizado por la máquina virtual.

4.2. Pruebas de integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras al mismo tiempo, se lleva a cabo pruebas para detectar errores asociados con la interacción. La integración puede ser descendente si se integran los módulos desde el control o programa principal, o bien, ascendente, si la verificación del diseño empieza desde los módulos más bajos y de allí al principal. En el caso de estudio las pruebas realizadas fueron del tipo descendente.

Rdesktop

La aplicación fue compilada con el sistema de ficheros. Para comprobar su correcto funcionamiento se inicia el sistema y se verifica.

Tabla 4.5: *Aplicación rdesktop.*

Escenario	Descripción	Respuesta del sistema	Flujo central
	Se inicia la aplicación rdesktop, la cual permitirá conectarse a un servidor de virtualización utilizando el protocolo RDP.	iniciar correctamente el sistema, mostrando el escritorio compartido del servidor de virtualización al cual se conectó	Iniciar rdesktop.

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.

Tabla 4.6: *Pruebas al sistema.*

Escenario	Descripción	Respuesta del sistema	Flujo central
Utilizando el protocolo RDP	Se realiza la conexión al escritorio compartido del servidor de virtualización mediante el protocolo RDP.	Se muestra el escritorio compartido en el servidor, mostrando la pantalla de bienvenida.	Se inicia la aplicación rdesktop.

4.4. Conclusiones parciales

Durante el desarrollo del presente capítulo se realizaron pruebas de funcionalidades, integración y al sistema, las cuales permitieron comprobar el correcto funcionamiento del sistema. Estas pruebas fueron realizadas emulando el sistema en Qemu, debido a que la computadora real

presentaba problemas técnicos. El desarrollo de estas pruebas permitieron comprobar el cumplimiento de los requisitos definidos en el Capítulo 2, exceptuando los requisitos de transporte del audio desde el servidor de virtualización y trabajo con dispositivos USB conectados en el cliente, las cuales no fueron posibles por los problemas técnicos antes mencionados.

Conclusiones

Una vez concluido el proceso de Virtualización de Android para la CID 300/9 se puede concluir lo siguiente:

1. Se creó una solución de virtualización de Android, la cual fue probada utilizando el software de virtualización Qemu para la conexión remota, configurado para una plataforma similar a la de la CID 300/9.
2. El uso de la herramienta Buildroot para la construcción de sistema propició un fácil método de construcción, obtención de un sistema integrado, estable y que cumple con las restricciones de tamaño y funcionalidades deseadas para un sistema operativo embebido.
3. El sistema brinda la opción de conexión remota con el servidor de virtualización a través del protocolo RDP.
4. La solución de virtualización creada puede ser utilizada además en sistemas basados en la arquitectura x86.

Recomendaciones

El autor del presente trabajo recomienda:

1. Lograr la ejecución directa de Android sobre la Computadora en una Sola Tarjeta diseñada por el ICID.
2. Utilizar este trabajo como guía para la virtualización de Android.
3. Agregarle soporte para otros protocolos de comunicación.

Glosario de Términos

API (Application Programming Interface): Término inglés que significa Interfaz de Programación de Aplicaciones.

Aplicación: Programa informático que permite a un usuario utilizar una computadora con un fin específico.

Arquitectura de computadora: Diseño conceptual y la estructura operacional fundamental de un sistema de computadora.

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.

Controlador: Programa informático que posibilita la interacción entre el sistema operativo de una computadora y un periférico o hardware.

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

Encapsulamiento: Mecanismo que consiste en organizar datos y métodos de una estructura, conciliando el modo en que el objeto se implementa, es decir, evitando el acceso a datos por cualquier otro medio distinto a los especificados.

Framework: Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

Hardware: Dispositivos físicos que componen una computadora.

Imagen ISO: Archivo donde se almacena una copia o imagen exacta de un sistema de ficheros.

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.

Plataforma: Sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible.

Recurso: Medios utilizados por los dispositivos para ejecutar sus funciones, provistos por los elementos del ordenador.

Rendimiento: Medida o cuantificación de la velocidad/resultado con que se realiza una tarea o proceso.

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

Software: Programa informático que trata de resolver necesidades concretas del usuario.

Referencias bibliográficas

- [1] **Netbook**, “*Definición de netbook - Qué es, Significado y Concepto.*”
[En línea] Disponible en la dirección web: <http://definicion.de/netbook/>, 2012.
[Citado: Mayo 2013].
- [2] **Open Handset Alliance**, “*Industry Leaders Announce Open Platform for Mobile Devices Open Handset Alliance.*” [En línea] Disponible en la dirección web:
http://www.openhandsetalliance.com/press_110507.html, 2007. [Citado: Marzo 2013].
- [3] **Domenech, Jaime**, “*Android iguala a iOS en número de aplicaciones disponibles.*”
[En línea] Disponible en la dirección web: <http://www.theinquirer.es/2012/11/01/android-igual-a-ios-en-numero-de-aplicaciones-disponibles.html>, 2012.
[Citado: Abril 2013].
- [4] **Móviles-Android**, “*Estadísticas Globales, Android, líder en ventas globales.*”
[En línea] Disponible en la dirección web: <http://alexistechblog.com/2011/11/15/estadisticas-globales-android-lider-en-ventas>, 2013. [Citado: Abril 2013].
- [5] **Marco, Antonio**, “*Estadísticas de smartphones 2010: Symbian sigue superando a Android.*”
[En línea] Disponible en la dirección web <http://www.celularis.com/smartphones/estadisticas-de-smartphones-2010-symbian-sigue-superando-a-android/>, 2012. [Citado: Abril 2013].
- [6] **Corredera, Mark**, “*Estadísticas Globales! Android, líder en ventas globales.*”
[En línea] Disponible en la dirección web: <http://alexistechblog.com/2011/11/15/>

- [estadisticas-globales-android-lider-en-ventas-globales](#), 2011. [Citado: Abril 2013].
- [7] **Android**, “*Estadísticas Mundiales de Smartphones y Celulares 2012.*” [En línea] Disponible en la dirección web: <http://www.poderpda.com/investigacion-y-desarrollo/estadisticas-mundiales-de-uso>, 2012. [Citado: Abril 2013].
- [8] **Pierra Fuentes, Allan**, “*Nova, distribución cubana de GNU/Linux: reestructuración estratégica de su proceso de desarrollo.*” [En línea] Disponible en la dirección web: <http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=11161>, 2011. [Citado: Abril 2013].
- [9] **Espeso, Pablo**, “*Xataka - ARM, la 'navaja suiza' de los procesadores.*” [En línea] Disponible en la dirección web: <http://m.xataka.com/componentes-de-pc/arm-la-navaja-suiza-de-los-procesadores-1>, 2012. [Citado: Mayo 2013].
- [10] **Hatchman, Mark**, “*ARM Cores Climb Into 3G Territory ExtremeTech.*” [En línea] Disponible en la dirección web: <http://www.extremetech.com/extreme/52180-arm-cores-climb-into-3g-territory>, 2002. [Citado: Abril 2013].
- [11] **Puente Fuentes, Ernesto**, “*Soporte para la Arquitectura de Computadora ARM en Nova para Cliente Ligero.*” [En línea] Disponible en la dirección web: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_05034_11/1/TD_05034_11.pdf, 2011. [Citado: Marzo 2013].
- [12] **Perso**, “*Programación de Aplicaciones Android.*” [En línea] Disponible en la dirección web: <http://perso.wanadoo.es/emiliotoboso/android/introduccion.htm>, 2013. [Citado: Mayo 2013].
- [13] **Elgin, Ben**, “*WebCite query result.*” [En línea] Disponible en la dirección web: <http://www.webcitation.org/5wk7sIvVb>, 2008. [Citado: Marzo 2013].

- [14] **Android**, “*Todo sobre sistema operativo Android.*” [En línea] Disponible en la dirección web: <http://www.consejosgratis.es/todo-sobre-sistema-operativo-android/>, 2011. [Citado: Abril 2013].
- [15] **Martínez, Jennifer**, “*More mobile phone makers back Google’s Android.*” [En línea] Disponible en la dirección web: <http://www.reuters.com/article/2008/12/10/openhandset>, 2008. [Citado: Marzo 2013].
- [16] **Marbmora**, “*Android - Ensayos.*” [En línea] Disponible en la dirección web: <http://www.buenastareas.com/ensayos/Android/7559392.html>, 2013. [Citado: Mayo 2013].
- [17] **Diccionario Babylon**, “*Definición de Bytecode por el diccionario gratuito de Babylon.*” [En línea] Disponible en la dirección web: <http://diccionario.babylon.com/bytecode/>, 2013. [Citado: Mayo 2013].
- [18] **Torres, Jorge Luis**, “*Plugins. Qué son y cómo utilizarlos inteligentemente.*” [En línea] Disponible en la dirección web: <http://artesonoro.com.mx/1847/plugins-que-son-y-como-utilizarlos-inteligentemente/>, 2011. [Citado: Mayo 2013].
- [19] **Developer**, “*Android, the world’s most popular mobile platform.*” [En línea] Disponible en la dirección web: <http://developer.android.com/about/index.html>, 2012. [Citado: Marzo 2013].
- [20] **Firmware**, “*Definición de firmware. Qué es, Significado y Concepto.*” [En línea] Disponible en la dirección web: <http://definicion.de/firmware/>, 2008. [Citado: Abril 2013].
- [21] **Mifsud, Elvira**, “*Introducción a la virtualización con XEN.*” [En línea] Disponible en la dirección web: <http://recursostic.educacion.es/observatorio/web/es/>

[software/servidores/1080-introduccion-a-la-virtualizacion-con-xen](#), 2012. [Citado: Febrero 2013].

- [22] **Itoh Mitsutaka and Eric Y. Chen**, “*Wowmom.*” [En línea] Disponible en la dirección web: <http://android-x86.googlecode.com/files/Wowmom-CR2.pdf>, 2012. [Citado: Febrero 2013].
- [23] **Virtualbox**, “*Virtual Box, un software de virtualización para arquitecturas x86.*” [En línea] Disponible en la dirección web: <http://virtualbox.es/>, 2013. [Citado: Febrero 2013].
- [24] **Meyner**, “*Software y web. Genbeta.*” [En línea] Disponible en la dirección web: <http://www.genbeta.com/>, 2009. [Citado: Febrero 2013].
- [25] **BlogTecnos**, “*Utilizar VirtualBox para trabajar y probar otros Sistemas Operativos en tu ordenador.*” [En línea] Disponible en la dirección web: <http://blogtecnos.blogspot.com/2011/09/utilizar-virtualbox-para-trabajar-y.html>, 2011. [Citado: Febrero 2013].
- [26] **Bermudez, Daniel**, “*Aprender VMware. ¿Que es Vmware?.*” [En línea] Disponible en la dirección web: <http://aprendervmware.blogspot.com/2010/08/que-es-vmware.html>, 2010. [Citado: Febrero 2013].
- [27] **Qemu**, “*Qemu.*” [En línea] Disponible en la dirección web: <http://www.arcos.inf.uc3m.es/folcina/pfc-html/node18.html>, 2008. [Citado: Febrero 2013].
- [28] **Navarro, David**, “*Qemu - Descargar.*” [En línea] Disponible en la dirección web: <http://qemu.softonic.com/>, 2012. [Citado: Febrero 2013].

- [29] **Protocolos**, "*Protocolos de Comunicación.*" [En línea] Disponible en la dirección web: <http://www.forest.ula.ve/mana/cursos/redes/protocolos.html>, 2012. [Citado: Mayo 2013].
- [30] **Miles, Arlene**, "*Remote Communication Protocol.*" [En línea] Disponible en la dirección web: http://www.ehow.com/facts_7910475_remote-communication-protocol.html, 2013. [Citado: Mayo 2013].
- [31] **Fuentes, Sacha**, "*VNC: especial software control remoto.*" [En línea] Disponible en la dirección web: <http://www.genbeta.com/comparativa/vnc-especial-software-control-remoto>, 2006. [Citado : Febrero 2013].
- [32] **RDP**, "*Remote Desktop Protocol.*" [En línea] Disponible en la dirección web: http://es.wikipedia.org/w/index.php?title=Remote_Desktop_Protocol&oldid=64622122, 2013. [Citado: Marzo 2013].
- [33] **Ministerio de educación, España**, "*NFS: Sistema de archivos de red.*" [En línea] Disponible en la dirección web: <http://recursostic.educacion.es/observatorio/web/es/software/software-general/733-nfs-sistema-de-archivos-de-red>, 2009. [Citado: Febrero 2013].
- [34] **Swingle, Bill**, "*NFS.*" [En línea] Disponible en la dirección web: <http://www.freebsd.org/doc/es/books/handbook/network-nfs.html>, 2010. [Citado: Febrero 2013].
- [35] **Brook**, "*Protocolo Nfs.*" [En línea] Disponible en la dirección web: <http://www.buenastareas.com/ensayos/Protocolo-Nfs/1807582.html>, 2011. [Citado: Febrero 2013].

- [36] **Coni**, "*MODELADO DE PROCESOS*." [En línea] Disponible en la dirección web:
<http://otroblogmas.fullblog.com.ar/modelado-de-procesos.html>, 2010.
[Citado: Febrero 2013].
- [37] **Labeybaanirak**, "*Definicion De Requerimientos*." [En línea] Disponible en la dirección web:
<http://www.buenastareas.com/ensayos/Definicion-De-Requerimientos/1695580.html>, 2011. [Citado: Mayo 2013].

Bibliografía

1. “Análisis de Symantec de las plataformas de Apple y Android de Google revela mayor seguridad en comparación con las PCs, pero aún existen algunas brechas”, [En línea], 2013, [Consultado: Enero 2013]. Disponible en la dirección web:, http://www.symantec.com/es/mx/about/news/release/article.jsp?prid=20110823_01
2. “Android, the world’s most popular mobile platform”, [En línea], 2012, [Consultado: Marzo 2013]. Disponible en la dirección web:, <http://developer.android.com/about/index.html>
3. “Android - Ensayos”, [En línea], 2013, [Consultado: Mayo 2013]. Disponible en la dirección web:, <http://www.buenastareas.com/ensayos/Android/7559392.html>
4. “Android dominó el mercado de los smartphones en 2012”, [En línea], 2013, [Consultado: Abril 2013]. Disponible en la dirección web:, <http://mexico.cnn.com/tecnologia/2013/02/15/android-domino-el-mercado-de-los-smartphones-en-2012>
5. “Android iguala a iOS en número de aplicaciones disponibles”, [En línea], 2012, [Consultado: Abril 2013]. Disponible en la dirección web:, <http://www.theinquirer.es/2012/11/01/android-igual-a-ios-en-numero-de-aplicaciones-disponibles.html>
6. “Aprender VMware. ¿Que es Vmware?”, [En línea], 2010, [Consultado: Febrero 2013]. Disponible en la dirección web: <http://aprendervmware.blogspot.com/2010/08/que-es-vmware.html>
7. “ARM Cores Climb Into 3G Territory”, [En línea], 2012, [Consultado: Mayo 2013]. Disponible en la dirección web:, <http://qemu.softonic.com/>

8. "ARM Cores Climb Into 3G Territory ExtremeTech", [En línea], 2002, [Consultado: Abril 2013]. Disponible en la dirección web:, <http://www.extremetech.com/extreme/52180-arm-cores/climb-into-3g-territory>
9. "Definición de Bytecode por el diccionario gratuito de Babylon" , [En línea], 2013, [Consultado: Febrero 2013]. Disponible en la dirección web: <http://diccionario.babylon.com/bytecode/>
10. "Definición de firmware. Qué es, Significado y Concepto", [En línea], 2008, [Consultado: Mayo 2013]. Disponible en la dirección web:, <http://definicion.de/firmware/>
11. "Definición de netbook - Qué es, Significado y Concepto" , [En línea], 2013, [Consultado: Mayo 2013]. Disponible en la dirección web: <http://definicion.de/netbook/>
12. "Definicion De Requerimientos", [En línea], 2011, [Consultado: Mayo 2013]. Disponible en la dirección web:, <http://www.buenastareas.com/ensayos/Definicion-De-Requerimientos/1695580.html>
13. "Delitos Informaticos. Análisis comparativo de Firma Electrónica" ,[En línea], 2010, [Consultado: Febrero 2013]. Disponible en la dirección web: <http://www.delitosinformaticos.com/firmaelectronica/analisis2.shtml>
14. 'Estadísticas Globales, Android, líder en ventas globales"', [En línea], 2013, [Consultado: Abril 2013]. Disponible en la dirección web:, <http://alexistechblog.com/2011/11/15/estadisticas-globales-android/lider-en-ventas-globales-en-q3-2011-525-de-marketshare/>
15. "Estadísticas de smartphones 2010: Symbian sigue superando a Android", [En línea], 2011, [Consultado: Abril 2013]. Disponible en la dirección web:, <http://www.celularis.com/smartphones/estadisticas-de-smartphones-2010-symbian-sigue-superando-a-android>

16. “Estadísticas Mundiales de Smartphones y Celulares 2012”, [En línea], 2012, [Consultado: Abril 2013]. Disponible en la dirección web:, <http://www.poderpda.com/investigacion-y-desarrollo/estadisticas-mundiales>
17. “Introducción a la virtualización con XEN” , [En línea], 2012, [Consultado: Mayo 2013]. Disponible en la dirección web:, <http://recursostic.educacion.es/observatorio/web/es/software/servidores/1080-introduccion-a-la-virtualizacion-con-xen>
18. “Informática de la red virtual”, [En línea], 2011, [Consultado: Febrero 2013]. Disponible en la dirección web:, <http://www14.knowledgres.com/00227863/InformaticaDeLaRedVirtual>
19. “Industry Leaders Announce Open Platform for Mobile Devices Open Handset Alliance”, [En línea], 2007, [Consultado: Marzo 2013]. Disponible en la dirección web:, http://www.openhandsetalliance.com/press_110507.html
20. “Microsoft Virtual PC - Descargar”, [En línea], 2007, [Consultado: Febrero 2013]. Disponible en la dirección web: <http://microsoft-virtual-pc.softonic.com/>
21. “Microsoft Word - CW_C.doc - CW_C.pdf ”, [En línea], 2012, [Consultado: Mayo 2013]. Disponible en la dirección web: , http://www.dsp.efn.unc.edu.ar/documentos/CW_C.pdf
22. “MODELADO DE PROCESOS”, [En línea], 2010, [Consultado: Marzo 2013]. Disponible en la dirección web:, <http://otroblogmas.fullblog.com.ar/modelado-de-procesos.html>
23. “More mobile phone makers back Google’s Android”, [En línea], 2008, [Consultado: Marzo 2013]. Disponible en la dirección web:, <http://www.reuters.com/article/2008/12/10/openhandset/idUSN0928595620081210>

24. "Network File System. NFS", [En línea], 2010, [Consultado: Febrero 2013]. Disponible en la dirección web:
<http://www.freebsd.org/doc/es/books/handbook/network-nfs.html>
25. "NFS: Sistema de archivos de red", [En línea], 2009, [Consultado: Febrero 2013]. Disponible en la dirección web: <http://recursostic.educacion.es/observatorio/web/es/software/software-general/733-nfs-sistema-de-archivos-de-red>
26. "Nova, distribución cubana de GNU/Linux: reestructuración estratégica de su proceso de desarrollo" [En línea], 2011, [Consultado: Abril 2013]. Disponible en la dirección web:
<http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=11161>
27. "Protocolos de Comunicación", [En línea], 2012, [Consultado: Mayo 2013]. Disponible en la dirección web:
<http://www.forest.ula.ve/~mana/cursos/redes/protocolos.html>
28. "Protocolo NFS", [En línea], 2011, [Consultado: Febrero 2013]. Disponible en la dirección web: <http://www.buenastareas.com/ensayos/Protocolo-Nfs/1807582.html>
29. "Programación de Aplicaciones Android", [En línea], 2013, [Consultado: Mayo 2013]. Disponible en la dirección web:
<http://perso.wanadoo.es/emiliotoboso/android/introduccion.htm>
30. "PT3:GPSL:UF6 - Formacion", [En línea], 2010, [Consultado: Marzo 2013]. Disponible en la dirección web: http://formacion.morfeo-project.org/wiki/index.php/PT3:GPSL:UF6#Principios_de_OpenUp
31. "Plugins. Qué son y cómo utilizarlos inteligentemente" ,[En línea], 2011, [Consultado: Marzo 2013]. Disponible en la dirección web: <http://artesonoro.com.mx/1847/plugins-que-son-y-como-utilizarlos-inteligentemente/>

32. “Qemu - Descargar”, [En línea], 2012, [Consultado: Febrero 2013]. Disponible en la dirección web:, <http://qemu.softonic.com/>
33. “Qemu”, [En línea], 2008, [Consultado: Febrero 2013]. Disponible en la dirección web:, <http://www.arcos.inf.uc3m.es/folcina/pfc-html/node18.html>
34. “Remote Desktop Protocol”, [En línea], 2013, [Consultado: Marzo 2013]. Disponible en la dirección web:, http://es.wikipedia.org/w/index.php?title=Remote_Desktop_Protocol&oldid=64622122
35. “Remote Communication Protocol”, [En línea], 2013, [Consultado: Mayo 2013]. Disponible en la dirección web:, http://www.ehow.com/facts_7910475_remote-communication-protocol.html
36. “Software y web. Genbeta”, [En línea], 2009, [Consultado: Febrero 2013]. Disponible en la dirección web:, <http://www.genbeta.com/>
37. “Soporte para la Arquitectura de Computadora ARM en Nova para Cliente Liger”, [En línea], 2011, [Consultado: Enero 2013]. Disponible en la dirección web:, http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_05034_11/1/TD_05034_11.pdf
38. “Todo sobre sistema operativo Android”, [En línea], 2011, [Consultado: Abril 2013]. Disponible en la dirección web:, <http://www.consejosgratis.es/todo-sobre-sistema-operativo-android/>
39. “Tutorial y descripción técnica de TCP/IP”, [En línea], 2010, [Consultado: Febrero 2013]. Disponible en la dirección web:, http://www.cicei.com/ocon/gsi/tut_tcpip/3376c412.html
40. “Virtual Box, un software de virtualización para arquitecturas x86”, [En línea], 2013, [Consultado: Febrero 2013]. Disponible en la dirección web: <http://virtualbox.es/>

41. "VMware Workstation", [En línea], 2012, [Consultado: Febrero 2013]. Disponible en la dirección web: <http://vmware-workstation.softonic.com/>
42. "VNC: especial software control remoto", [En línea], 2006, [Consultado: Febrero 2013]. Disponible en la dirección web:, <http://www.genbeta.com/comparativa/vnc-especial-software-control-remoto>
43. "Virtual Smartphone over IP", [En línea], 2012, [Consultado: Febrero 2013]. Disponible en la dirección web:, <http://android-x86.googlecode.com/files/Wowmom-CR2.pdf>
44. "Virtualbox en linea de comandos", [En línea], 2011, [Consultado: Mayo 2013]. Disponible en la dirección web: , <http://www.ubuntu-es.org/node/176583>
45. "VBoxHeadless - Running Virtual Machines With VirtualBox 2.0 On A Headless Ubuntu 8.04 Server" [En línea], 2012, [Consultado: Mayo 2013]. Disponible en la dirección web: , <http://www.howtoforge.com/vboxheadless-running-virtual-machines-with-virtualbox-2.0-on-a-headless-ubuntu-8.04-server>
46. "WebCite query result", [En línea], 2008, [Consultado: Marzo 2013]. Disponible en la dirección web:, <http://www.webcitation.org/5wk7sIvVb>
47. "Xataka - ARM, la 'navaja suiza' de los procesadores", [En línea], 2012, [Consultado: Mayo 2013]. Disponible en la dirección web:, <http://m.xataka.com/componentes-de-pc/arm-la-navaja-suiza-de-los-procesadores-1>

Anexo I

Proceso de instalación y configuración de cada una de las herramientas a utilizar:

Qemu

Qemu permitirá emular el sistema a construir. Permitiendo dar una abstracción del hardware ARM al sistema en construcción.

Instalación:

```
$ sudo apt-get install qemu-kvm-extras
```

NFS Server

Al emular el sistema se necesitará que los cambios realizados se mantengan en el sistema de ficheros que se construirá. Para lograr esto se exportará el sistema de fichero desde el sistema anfitrión mediante NFS. Creando una red virtual para conectar Qemu con el sistema anfitrión se podrá lograr el objetivo anterior.

Instalación:

```
$ sudo apt-get install nfs-kernel-server
```

Exportar el sistema en el servidor NFS:

```
# cat >/etc/exports << "EOF"
```

```
/trabajo/sistema/
```

```
*(rw,sync,no_subtree_check,all_squash,insecure,anonuid=0,anongid=0)
```

```
EOF
```

```
# exportfs -av
```

Red Virtual

```
# apt-get install uml-utilities
```


Para la configuración automática de la red virtual se creará un fichero ejecutable, el cual contendrá los comandos necesarios para iniciarla, así una vez que se inicie la red virtual solo habrá que ejecutar el fichero creado.

```
$ cd /trabajo/
```

```
$ cat >red << "EOF"
```

```
sudo tuncctl -u $(whoami) -t tap1
```

```
sudo ifconfig tap1 10.53.7.243
```

```
sudo route add -net 10.53.7.0 netmask 255.255.255.0 dev tap1
```

```
sudo sh -c "echo 1 >/proc/sys/net/ipv4/ip_forward"
```

```
EOF
```

```
$ chmod +x red
```

Minicom

Minicom es un programa de comunicaciones, permitirá el intercambio de información desde la computadora anfitrión hacia la computadora real. Para el intercambio de información se utilizará el puerto serial de ambas computadoras.

Instalación:

```
$ apt-get install minicom
```

Configuración:

```
$ sudo minicom -s
```

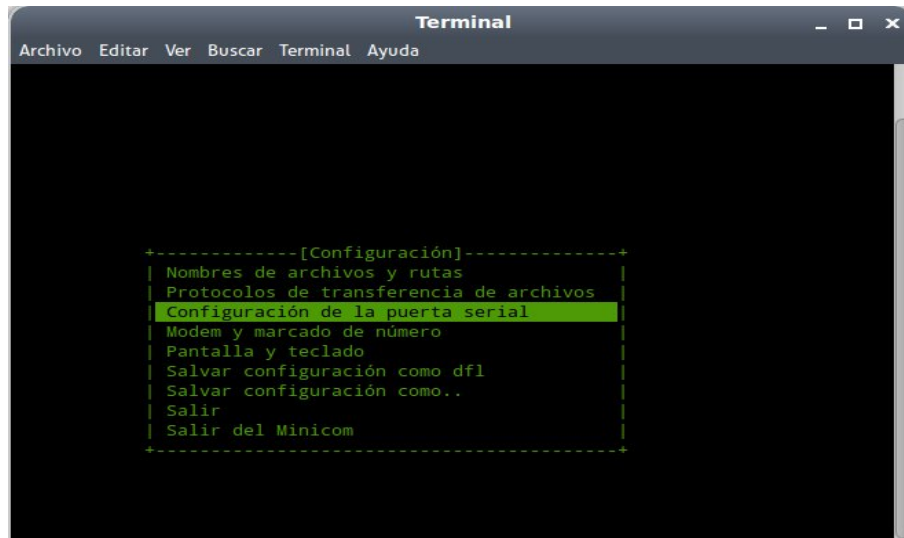


Figura 4.1: Vista general de la configuración de Minicom

Seleccionar la opción ***Configuración de la puerta serial***

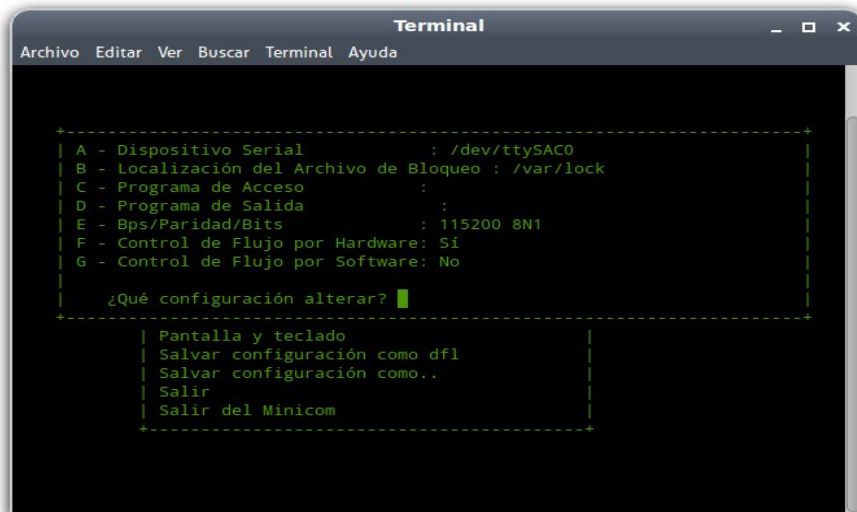


Figura 4.2: Vista general de la configuración de la puerta serial

Presionar la tecla “**A**” y luego teclear “**/dev/ttyS0**”, y si usa otro puerto serial que no sea el cero se cambiará “**tyS0**” por “**tyS#**”, donde “**#**” sería el número del puerto que se utilice, al finalizar presionar la tecla “**Enter**”. Luego presionar la tecla “**F**”. Al finalizar debe estar configurado como se muestra en la figura anterior. Para volver al menú anterior se presiona “**Enter**”.

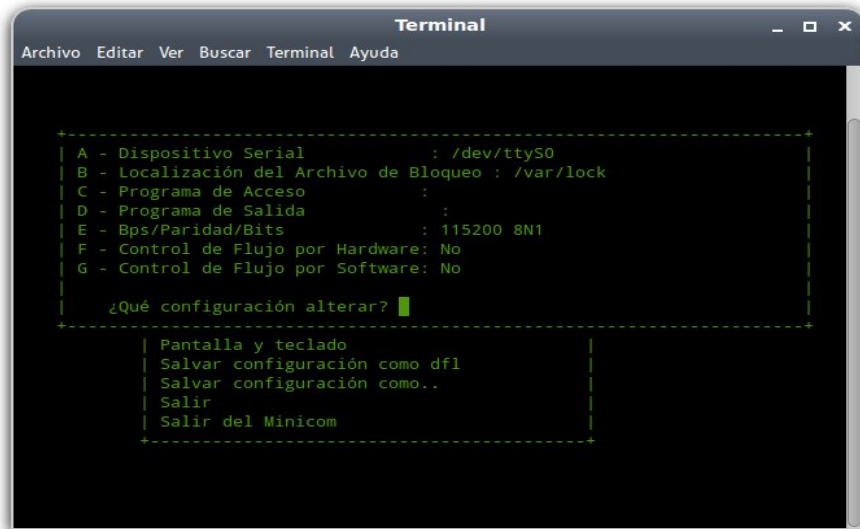


Figura 4.3: Vista general de la configuración de la puerta serial

Una vez configurado se guardará la configuración anterior, para esto se selecciona la opción **“Salvar configuración como dfl”**.

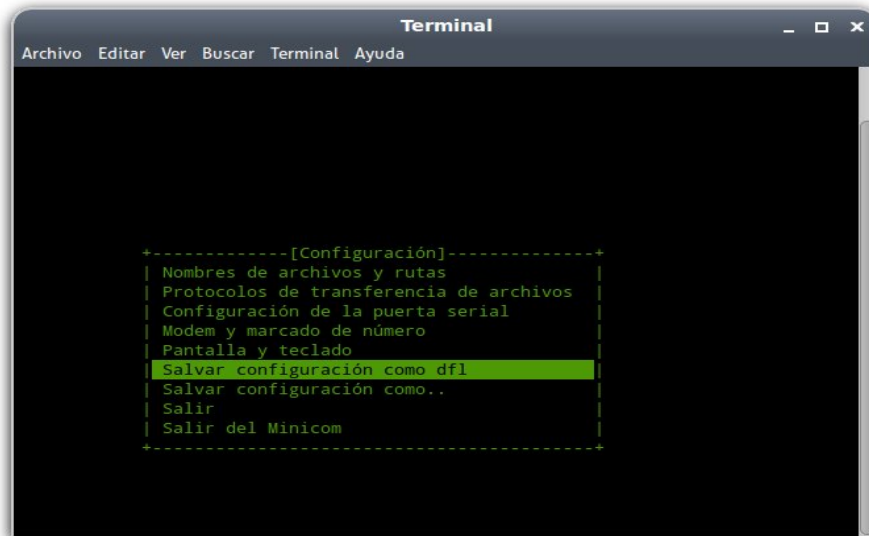


Figura 4.4: Vista general de la configuración de la puerta serial

Para salir se selecciona la opción **“Salir del Minicom”**.

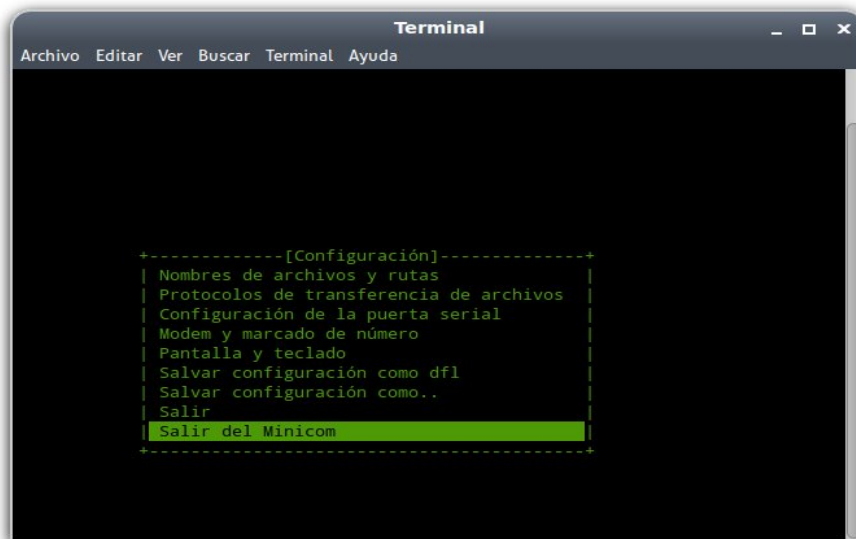


Figura 4.5: Vista general de la configuración de la puerta serial

CodeSourcery

Se necesitará las herramientas de compilación cruzada para la arquitectura ARM, en el caso de estudio se utilizó las que proporciona CodeSourcery.

Para su instalación se descarga en el directorio **Herramientas** y se procede:

- ◇ ***\$ cd /media/Datos/Trabajo/Herramientas/CodeSourcery/***
- ◇ ***\$ chmod +x arm-2010.09-50-arm-none-linux-gnueabi.bin***
- ◇ ***\$ sudo dpkg-reconfigure -plow dash***

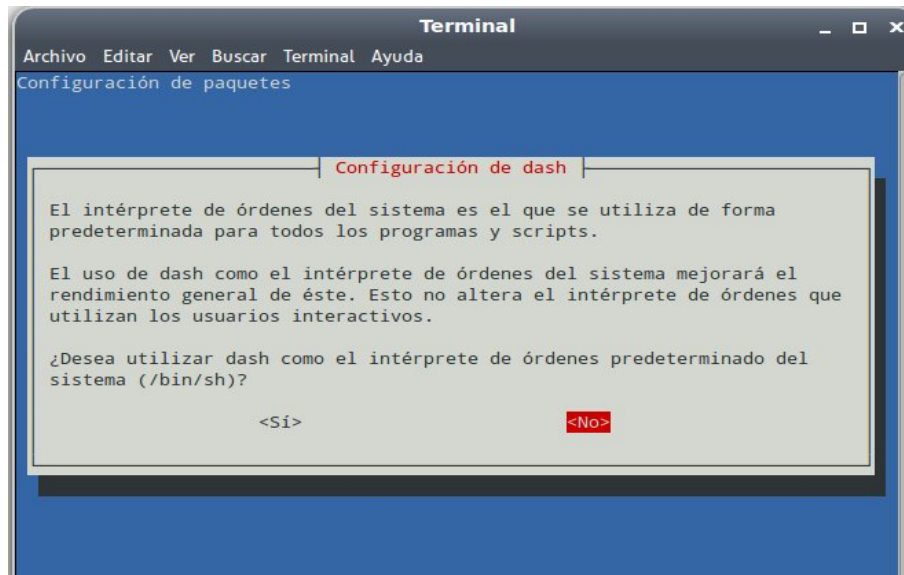


Figura 4.6: Vista general de la configuración de dash

◇ **`$. /arm-2010.09-50-arm-none-linux-gnueabi.bin`**

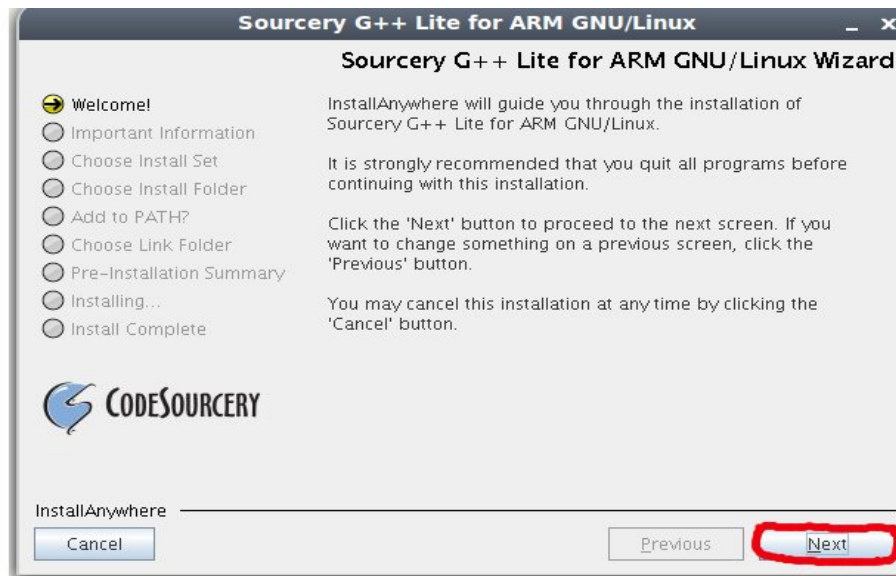


Figura 4.7: Vista general de la configuración de CodeSourcery

◇ Seleccionar la opción '**Accept the Term of the License Agreement**' y presionar **Next**.

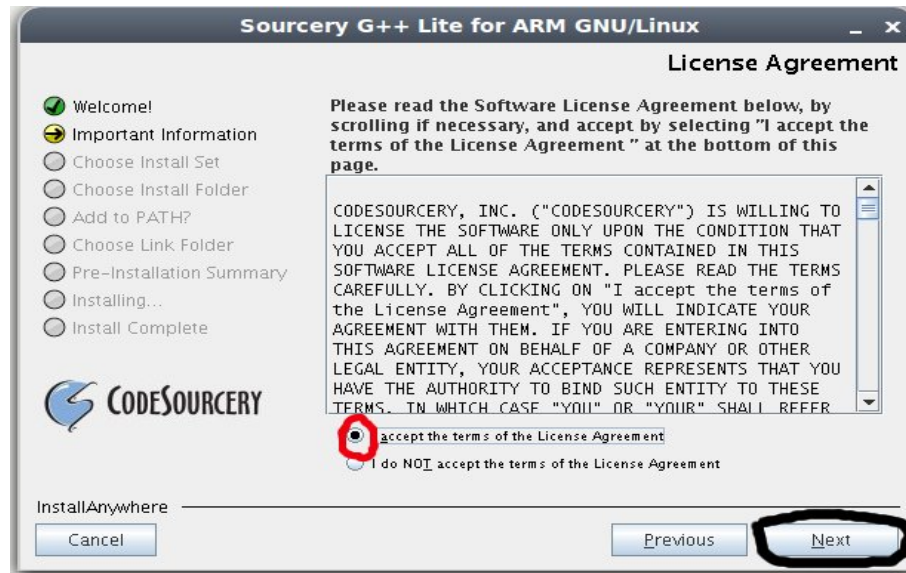


Figura 4.8: Vista general de la configuración de CodeSourcery

◇ Presionar **Next**.

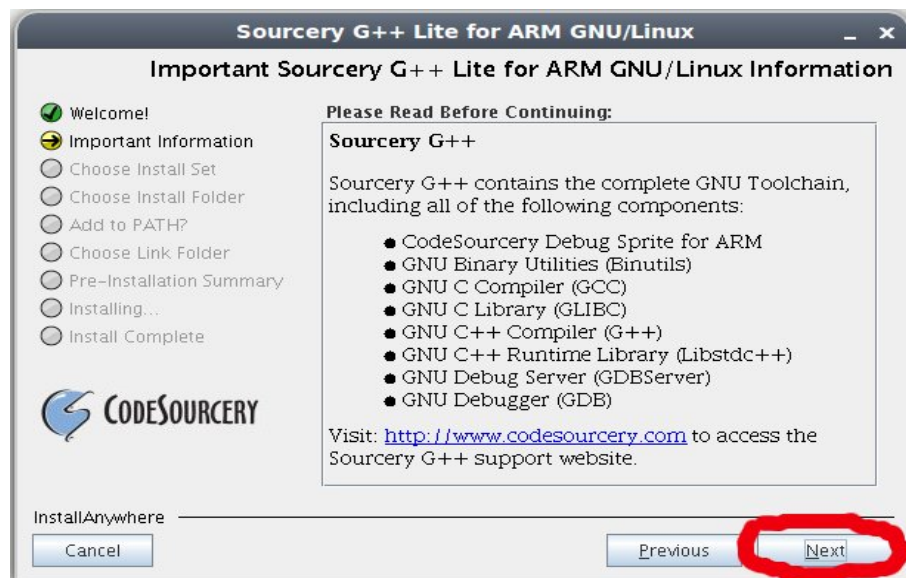


Figura 4.9: Vista general de la configuración de CodeSourcery

◇ Seleccionar la opción '**Custom**' y precionar **Next**.

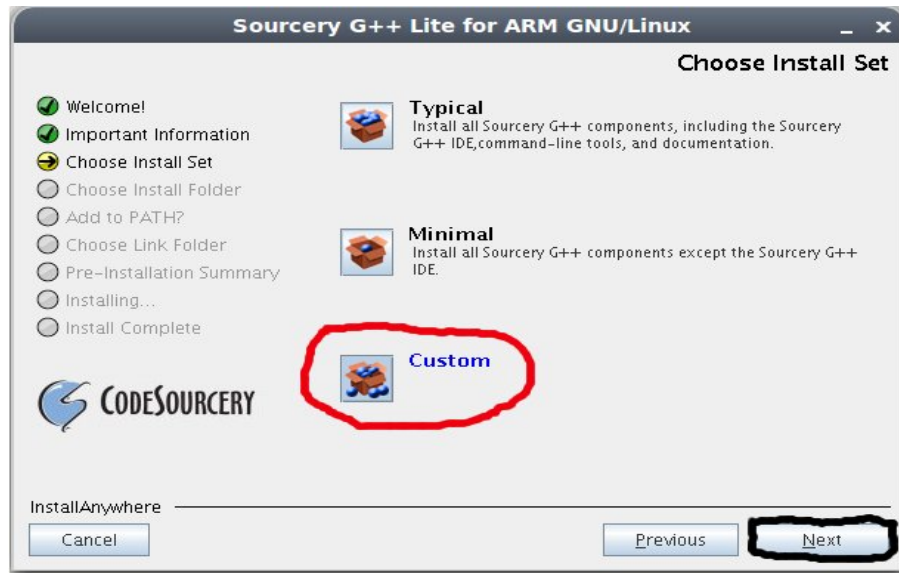


Figura 4.10: Vista general de la configuración de CodeSourcery

◇ Presionar **Next**.

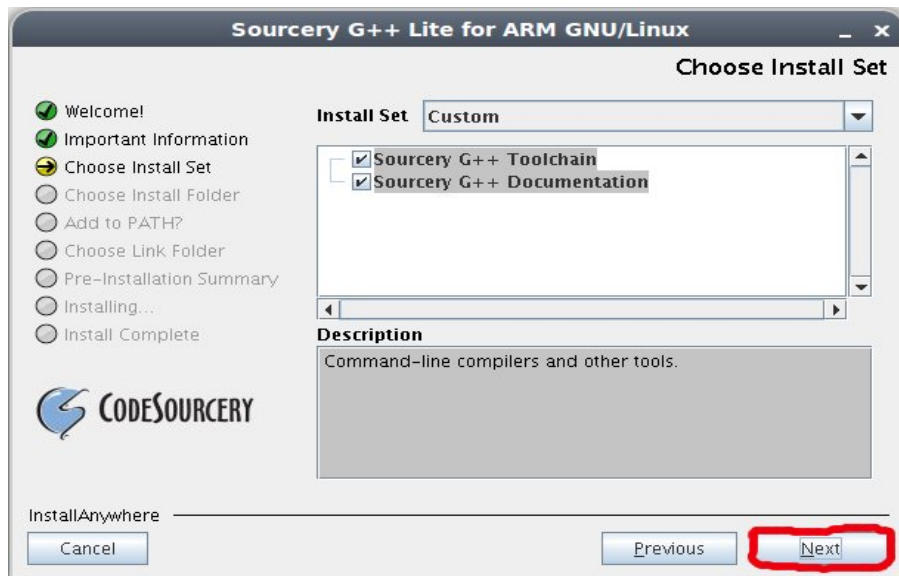


Figura 4.11: Vista general de la configuración de CodeSourcery

◇ Seleccionar directorio donde se desea instalar y precionar **Next**.

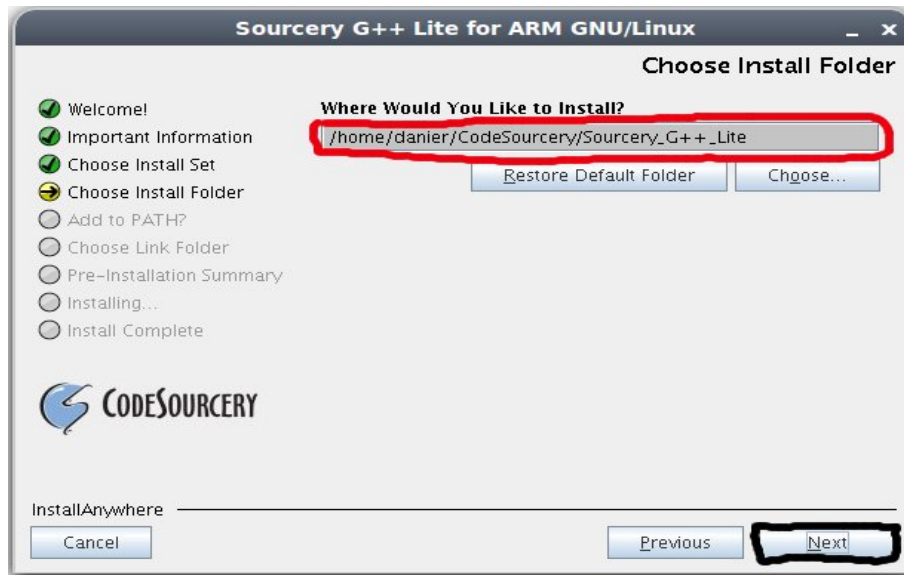


Figura 4.12: Vista general de la configuración de CodeSourcery

- ◇ Seleccionar la opción **'Modify PATH for current user'** y precionar **Next**.

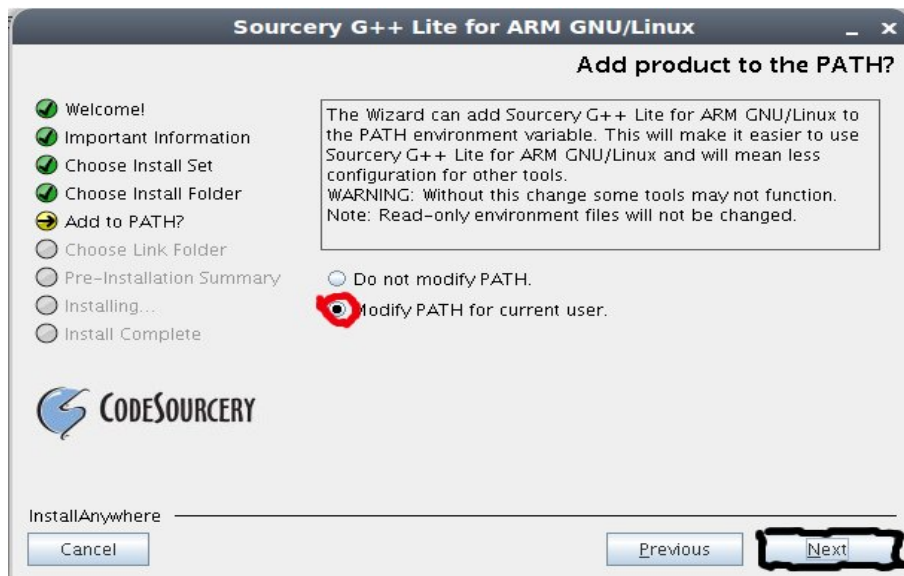


Figura 4.13: Vista general de la configuración de CodeSourcery

- ◇ Seleccionar la opción **'Don't create links'** y precionar **Next**.

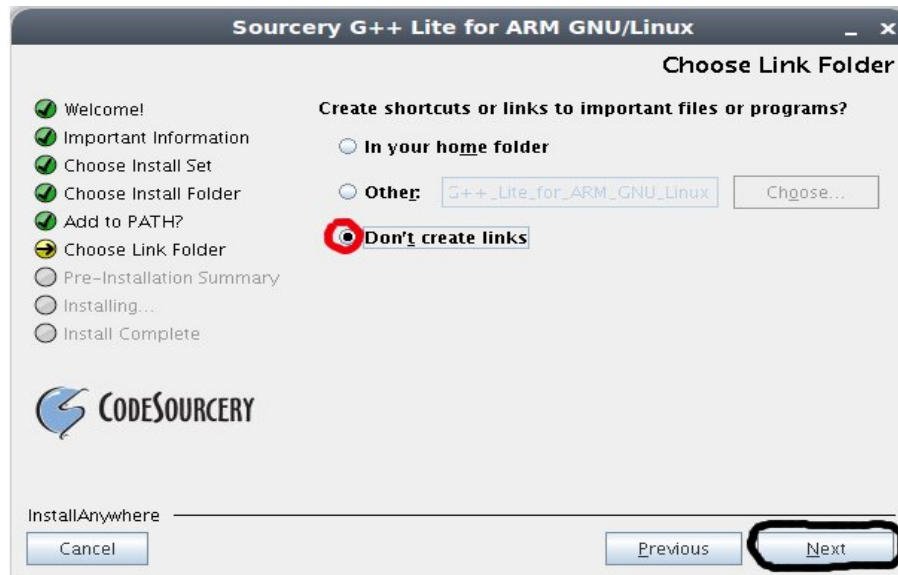


Figura 4.14: Vista general de la configuración de CodeSourcery

- ◇ Presionar la opción **Install**.

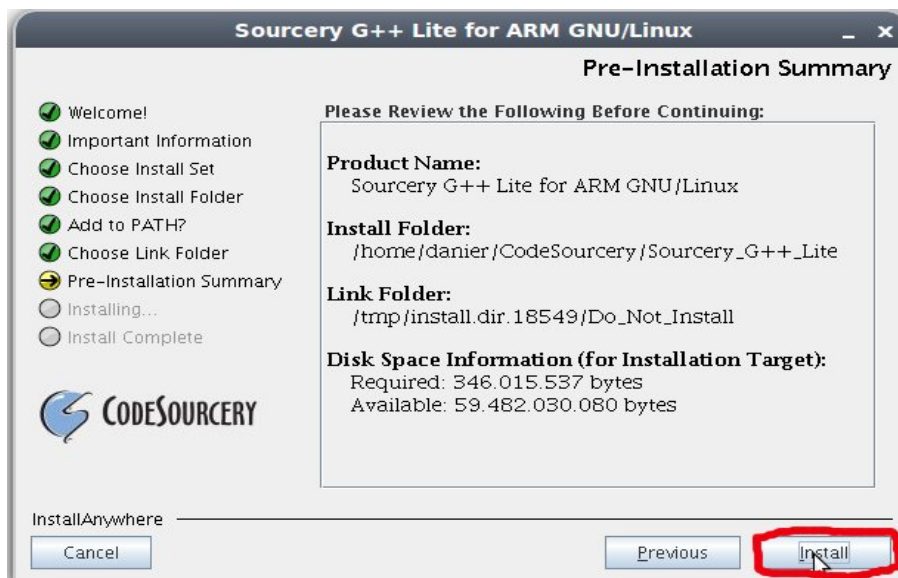


Figura 4.15: Vista general de la configuración de CodeSourcery

- ◇ Desmarcar la opción '**View Getting Started' guide ?** y precionar **Next**.

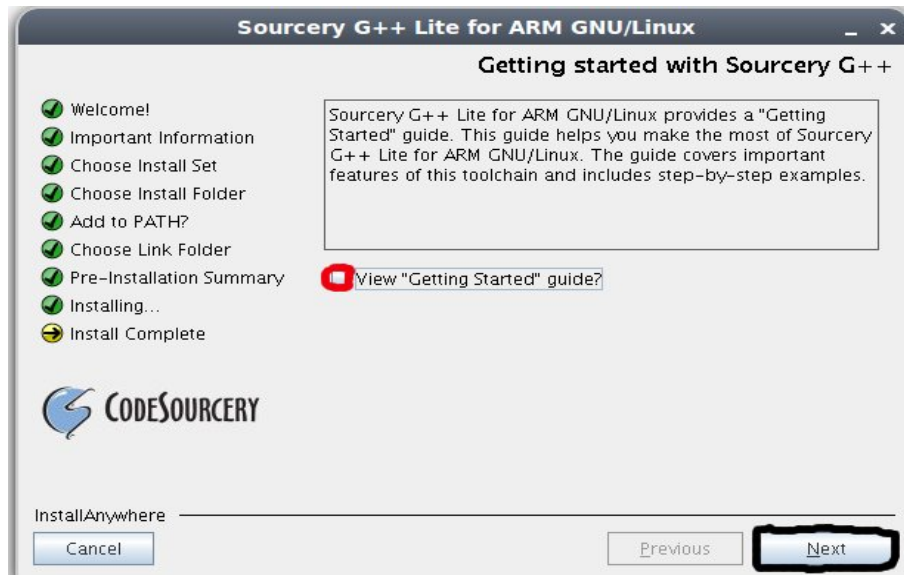


Figura 4.16: Vista general de la configuración de CodeSourcery

◇ Presionar **Done** para salir del instalador.

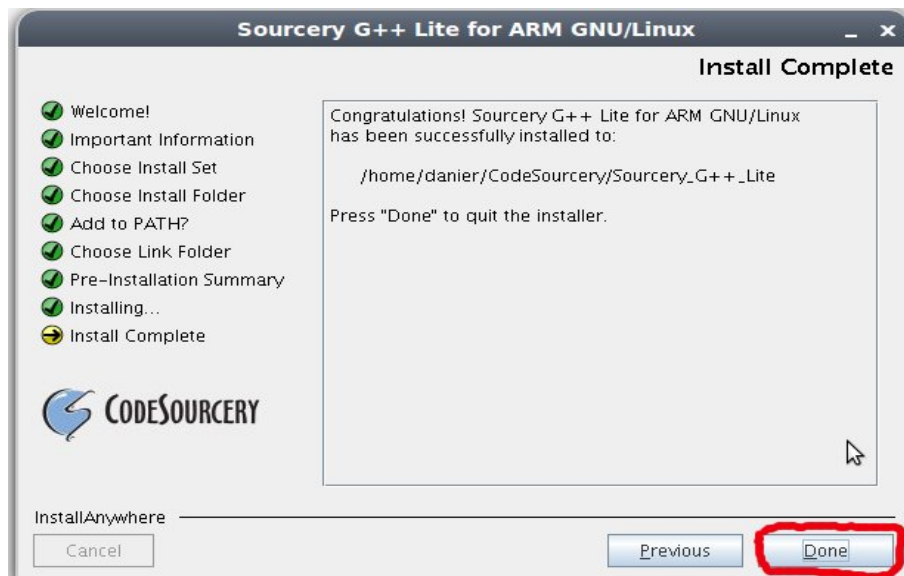


Figura 4.17: Vista general de la configuración de CodeSourcery

Se reinicia el sistema anfitrión para que tengan efecto todas las configuraciones realizadas durante le proceso de instalación de CodeSourcery.

Anexo II

Configuración de la red

Para el correcto funcionamiento del sistema creado se tendrán que crear y configurar una serie de ficheros, este proceso se explicará a continuación:

Moverse a la dirección donde se encuentre el sistema.

◇ **\$ cd /media/Datos/Trabajo/Sistema**

◇ **\$ cd etc/network**

\$ cat >interfaces < < “EOF”

auto lo

iface lo inet loopback

auto eth1

iface eth1 inet static

address 10.53.7.25

netmask 255.255.255.0

gateway 10.53.7.254

#iface eth0 inet dhcp

EOF

\$ cd ..

\$ cat >resolv.conf < < “EOF”

nameserver 10.0.0.3

nameserver 10.0.0.4

EOF

Anexo III

Proceso de copiar la imagen del núcleo compilada para la computadora real, mediante el puerto serie, utilizando la herramienta **minicom**.

Ejecutar en el sistema anfitrión:

minicom

Mantener presionada una tecla cualquiera y reiniciar la computadora real. Se debe cargar la interfaz de cargador de arranque instalado en la computadora real, en el caso de estudio “**vivi**”.

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
VIVI version 0.1.4 (puente@Nova) (gcc version 2.95.3 20010315 (release)) #0.1.42
MMU table base address = 0x37DFC000
Succeed memory mapping.
NAND device: Manufacture ID: 0xec, Chip ID: 0x76 (Samsung K9D1208V0M)
Found saved vivi parameters.
Press Return to start the LINUX now, any other key for vivi
Copy linux kernel from 0x00030000 to 0x30008000, size = 0x00280000 ... done
Warning: this binary is not compressed linux kernel image
zImage magic = 0xffffffff
Setup linux parameters at 0x30000100
linux command line is: "root=/dev/mmcblk0p1 rw rootwait init=/sbin/init console"
MACH_TYPE = 362
NOW, Booting Linux.....

VIVI version 0.1.4 (puente@Nova) (gcc version 2.95.3 20010315 (release)) #0.1.42
MMU table base address = 0x37DFC000
Succeed memory mapping.
NAND device: Manufacture ID: 0xec, Chip ID: 0x76 (Samsung K9D1208V0M)
Found saved vivi parameters.
Press Return to start the LINUX now, any other key for vivi
type "help" for help.
puente>
```

Figura 4.18: Vista general del proceso de cargar un nuevo kernel.

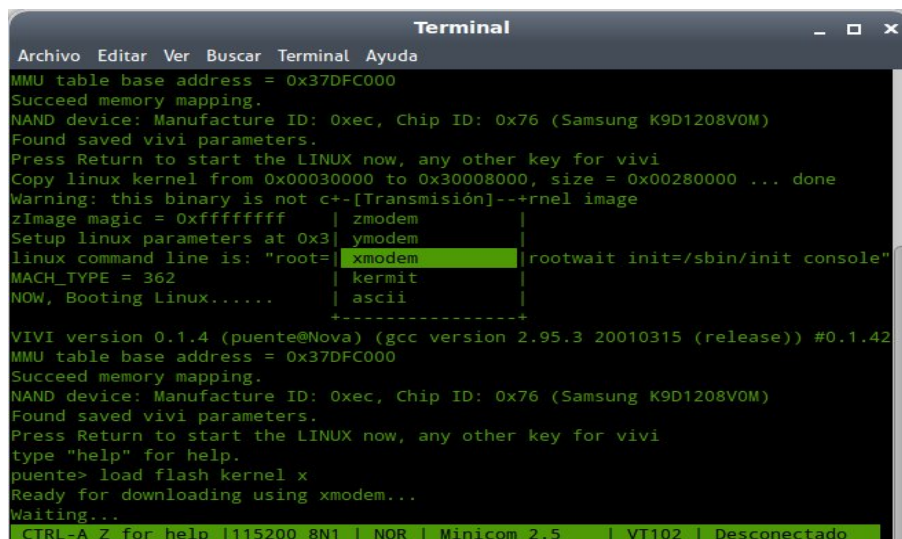
Ejecutar en **vivi**:

>**flash erase kernel**

>**load flash kernel x**

Presionar **Ctrl + A**, luego presionar **S**. IMPORTANTE!! Al presionar **Ctrl + A**, aparentemente no

ocurre nada, se debe presionar de forma inmediata la letra **S**. Se mostrará lo que aparece a continuación:



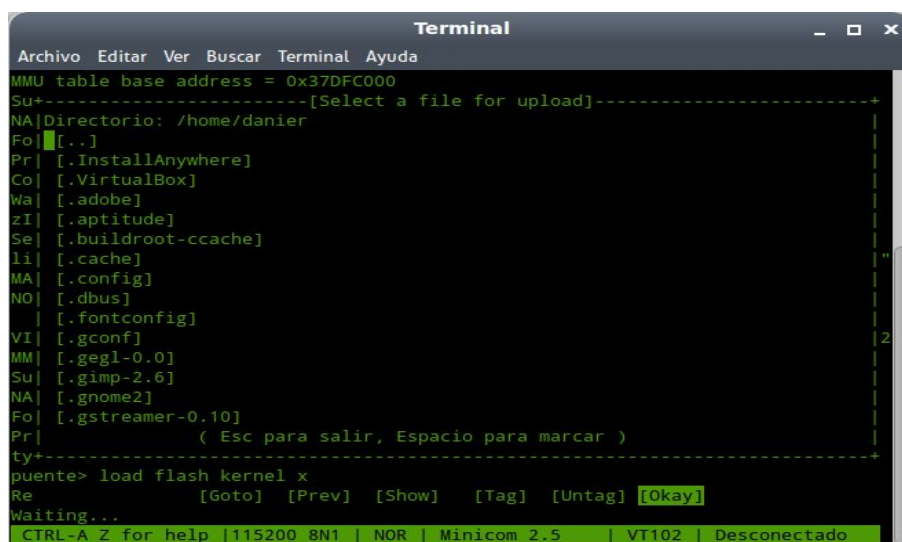
```

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
MMU table base address = 0x37DFC000
Succeed memory mapping.
NAND device: Manufacture ID: 0xec, Chip ID: 0x76 (Samsung K9D1208V0M)
Found saved vivi parameters.
Press Return to start the LINUX now, any other key for vivi
Copy linux kernel from 0x00030000 to 0x30008000, size = 0x00280000 ... done
Warning: this binary is not c+-[Transmisión]--+rnel image
zImage magic = 0xffffffff | zmodem |
Setup linux parameters at 0x3 | ymodem |
linux command line is: "root= | xmodem | rootwait init=/sbin/init console"
MACH_TYPE = 362 | kermit |
NOW, Booting Linux..... | ascii |
+-----+
VIVI version 0.1.4 (puente@Nova) (gcc version 2.95.3 20010315 (release)) #0.1.42
MMU table base address = 0x37DFC000
Succeed memory mapping.
NAND device: Manufacture ID: 0xec, Chip ID: 0x76 (Samsung K9D1208V0M)
Found saved vivi parameters.
Press Return to start the LINUX now, any other key for vivi
type "help" for help.
puente> load flash kernel x
Ready for downloading using xmodem...
Waiting...
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.5 | VT102 | Desconectado

```

Figura 4.19: Vista general del proceso de cargar un nuevo kernel.

Elegir “**xmodem**”



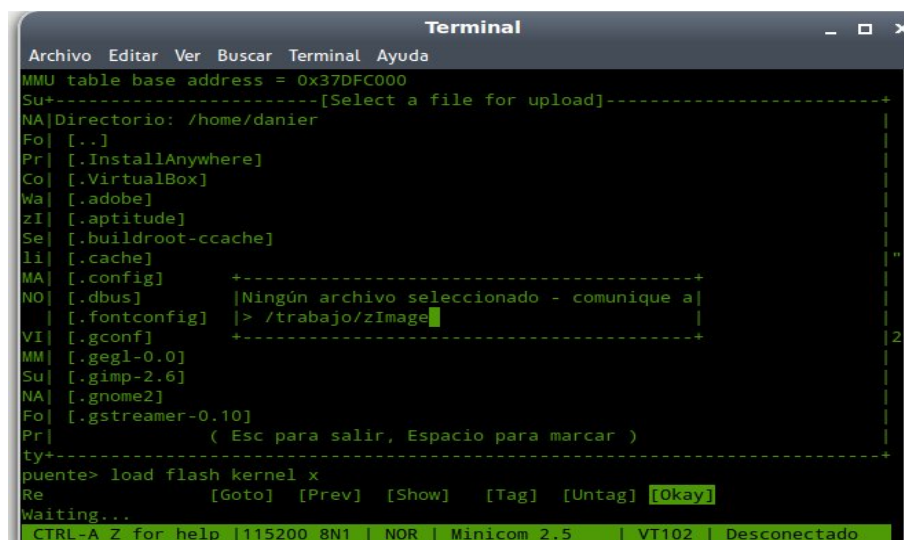
```

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
MMU table base address = 0x37DFC000
Su+-----[Select a file for upload]-----+
NA|Directorio: /home/danier
Fo| [..]
Pr| [.InstallAnywhere]
Co| [.VirtualBox]
Wa| [.adobe]
zI| [.aptitude]
Se| [.buildroot-ccache]
li| [.cache]
MA| [.config]
NO| [.dbus]
 | [.fontconfig]
VI| [.gconf]
MM| [.gegl-0.0]
Su| [.gimp-2.6]
NA| [.gnome2]
Fo| [.gstreamer-0.10]
Pr|
 | ( Esc para salir, Espacio para marcar )
ty+-----+
puente> load flash kernel x
Re | [Goto] [Prev] [Show] [Tag] [Untag] [Okay]
Waiting...
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.5 | VT102 | Desconectado

```

Figura 4.20: Vista general del proceso de cargar un nuevo kernel.

Presionar **Enter**



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
MMU table base address = 0x37DFC000
Su+-----[Select a file for upload]-----+
NA|Directorio: /home/danier
Fo| [..]
Pr| [.InstallAnywhere]
Co| [.VirtualBox]
Wa| [.adobe]
zI| [.aptitude]
Se| [.buildroot-ccache]
li| [.cache]
MA| [.config] +-----+
NO| [.dbus] |Ningún archivo seleccionado - comuniqua|
| [.fontconfig] |> /trabajo/zImage|
VI| [.gconf] +-----+
MM| [.gegl-0.0]
Su| [.gimp-2.6]
NA| [.gnome2]
Fo| [.gstreamer-0.10]
Pr| ( Esc para salir, Espacio para marcar )
ty+-----+
puente> load flash kernel x
Re [Goto] [Prev] [Show] [Tag] [Untag] [Okay]
Waiting...
CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.5 | VT102 | Desconectado
```

Figura 4.21: Vista general del proceso de cargar un nuevo kernel.

Teclear la dirección donde se encuentra la imagen del núcleo: **/trabajo/zImageR**. Luego presionar **Enter**. Esperar a que termine. Introducir la tarjeta SD en la computadora real y reiniciar la computadora real. Debe iniciar el sistema construido correctamente.