



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA, EN
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**TEMA: EVALUACIÓN DEL RENDIMIENTO Y PRESTACIONES
DE UN DECODIFICADOR DE CONTENIDOS INTERACTIVOS
BASADOS EN GINGA-NCL SOBRE UNA RASPBERRY PI**

AUTOR: ESPINEL RIVERA, KARLA VERÓNICA

DIRECTOR: ING. ALULEMA FLORES, DARWIN

SANGOLQUÍ

2016



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “***EVALUACIÓN DEL RENDIMIENTO Y PRESTACIONES DE UN DECODIFICADOR DE CONTENIDOS INTERACTIVOS BASADOS EN GINGA-NCL SOBRE UNA RASPBERRY PI***” realizado por la señora ***KARLA VERÓNICA ESPINEL RIVERA***, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a la señora ***KARLA VERÓNICA ESPINEL RIVERA*** para que lo sustente públicamente.

Sangolquí, 26 de Enero del 2016

Ing. Darwin Alulema.
DIRECTOR



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**

AUTORÍA DE RESPONSABILIDAD

Yo, **KARLA VERÓNICA ESPINEL RIVERA**, con cédula de identidad N°1724325160, declaro que este trabajo de titulación “**EVALUACIÓN DEL RENDIMIENTO Y PRESTACIONES DE UN DECODIFICADOR DE CONTENIDOS INTERACTIVOS BASADOS EN GINGA-NCL SOBRE UNA RASPBERRY PI**” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 26 de Enero del 2016

KARLA VERÓNICA ESPINEL RIVERA

C.C. 1724325160



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, ***KARLA VERÓNICA ESPINEL RIVERA***, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación ***“EVALUACIÓN DEL RENDIMIENTO Y PRESTACIONES DE UN DECODIFICADOR DE CONTENIDOS INTERACTIVOS BASADOS EN GINGA-NCL SOBRE UNA RASPBERRY PI”*** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 26 de Enero del 2016

KARLA VERÓNICA ESPINEL RIVERA

C.C. 1724325160

DEDICATORIAS

A Dios por darme la vida, a mis padres por su esfuerzo y apoyo, a mis hermanas por ser los pilares de mi vida, a mi hijo Dario Sebastian por ser mi motivación de superación y a mi esposo Dario porque siempre ha estado conmigo.

Espinel Rivera Karla Verónica

AGRADECIMIENTOS

A Dios y a la Virgen por protegerme día a día, a mis padres por darme la oportunidad de superarme siempre brindándome su apoyo incondicional y su amor, a mis hermanas quienes son los pilares de mi vida, mis amigas y confidentes, a mi hijo Dario Sebastian que con sus sonrisas es la motivación para superarme cada día más, a mi esposo Dario quien ha estado a mi lado apoyándome y animándome siempre, a mi abuelita y mi tía que con sus consejos nunca me han dejado sola, a mis sobrinos por ser mis compañeros de locuras y como mis hermanos.

A mis amigos, por compartir esta experiencia de aprendizaje, de momentos difíciles que superamos juntos. Finalmente a cada uno de los ingenieros quienes han compartido sus conocimientos y vivencias. Gracias a cada uno de ustedes por ser parte de este logro.

Espinel Rivera Karla Verónica

INDICE DE CONTENIDO

CERTIFICACIÓN	II
AUTORÍA DE RESPONSABILIDAD	III
AUTORIZACIÓN	IV
DEDICATORIAS	V
AGRADECIMIENTOS	VI
RESUMEN	XI
ABSTRACT	XII
PRESENTACIÓN	XIII
CAPÍTULO 1 : INTRODUCCIÓN	1
1.1. Antecedentes.....	1
1.2. Objetivos.....	2
1.2.1. General.....	2
1.2.2. Específicos.....	2
1.3. Justificación e importancia del proyecto.....	3
1.4. Alcance del proyecto.....	4
1.5. Estudio del Estado del Arte.....	5
CAPÍTULO 2 : MARCO TEÓRICO	8
2.1. Hardware Libre.....	8
2.1.1. Definición.....	8
2.1.2. Clasificación de Hardware Libre.....	9
2.2. Raspberry PI.....	10
2.2.1. Características físicas básicas de la Raspberry PI.....	13
2.2.2. Descripción de pines GPIO.....	14
2.2.3. Descripción de Sistemas Operativos.....	16
2.3. ISDB-Tb.....	17
2.3.1. Características (Piscotta, 2010).....	18
2.3.2. Modulación OFDM.....	19
2.4. Televisión Digital Interactiva.....	21
2.4.1. Ventajas de la TVD.....	22

2.4.2. Elementos tecnológicos	22
2.4.3. Beneficios de la Interactividad	23
2.5. Middleware GINGA.....	24
2.5.1. Arquitectura del Middleware Ginga	25
2.6. Ginga CC.....	26
2.7. Ginga NCL	27
2.7.1. Arquitectura de Ginga NCL	27
2.7.2. NCL- lua	29
2.8. Ginga J	30
2.8.1. Arquitectura de Ginga J	30
2.9. Ginga.ar	31
2.10. Ginga Brasil	31
CAPÍTULO 3 : DISEÑO E IMPLEMENTACIÓN.....	33
3.1. Determinación de requerimientos	33
3.2. LIFIA (Argentina).....	33
3.3. TELEMÍDIA (Brasil)	34
3.4. Descripción de requerimiento de Ginga Telemídia	35
3.4.1. Instalación de XERCES-C.....	35
3.4.1.1. Funciones.....	36
3.4.2. Instalación de CURL.....	36
3.4.2.1. Funciones.....	37
3.4.3. Instalación de SDL2	37
3.4.4. Instalación de SDL2_image	38
3.4.5. Instalación de SDL2_ttf	38
3.4.6. Instalación de FFMPEG	38
3.4.6.1. Función	39
3.4.7. Instalación de NCLUA.....	40
3.4.8. Instalación de Berkelium.....	41
3.5. Análisis de la tarjeta Raspberry Pi	41
3.5.1. Raspberry Pi Modelo B	42
3.5.2. Raspberry Pi 2.....	42

3.6. Integración del sistema operativo en la Raspberry Pi.....	43
3.6.1. Instalación de la imagen en la tarjeta SD	44
3.6.2. Instalación de GINGA.....	45
3.7. Resumen	46
CAPÍTULO 4 : PRUEBAS Y RESULTADOS	47
4.1. Ejecución de los programas.....	47
4.1.1. Programa 1	48
4.1.2. Programa 2	48
4.1.3. Programa 3	48
4.2. Pruebas con la Raspberry Pi Modelo B	49
4.2.1. Raspbian Jessie	49
4.3. Pruebas con la Raspberry Pi 2	49
4.3.1. Ubuntu Mate	50
4.3.2. Raspbian Jessie	51
4.4. Medición de rendimiento.....	52
4.5. Medición de calentamiento.....	55
CAPÍTULO 5 : CONCLUSIONES Y RECOMENDACIONES	58
5.1. Conclusiones	58
5.2. Recomendaciones	60
REFERENCIAS	62

ÍNDICE TABLAS

Tabla 1. Sistemas Operativos que soporta la Raspberry PI	10
Tabla 2. Características de la Raspberry PI modelo antiguo.....	11
Tabla 3. Características de la Raspberry PI modelo nuevo	11
Tabla 4. Características principales de la Raspberry Pi Modelo B	42
Tabla 5. Características principales y beneficios de la Raspberry Pi 2.....	43
Tabla 6. Medición de Rendimiento con Escritorio Remoto.....	52
Tabla 7. Medición de Rendimiento con Hardware Externo	53
Tabla 8. Medición de Calentamiento.....	55

ÍNDICE DE FIGURAS

Figura 1. Partes de la Raspberry PI Modelo B	12
Figura 2. Partes de la Raspberry PI 2.....	12
Figura 3. Entradas exteriores de la Raspberry PI	13
Figura 4. Pines GPIO de Raspberry Modelo B.....	15
Figura 5. Pines de GPIO de la Raspberry Pi 2.....	16
Figura 6. Distribución de portadoras	20
Figura 7. Transmisión Analógica	21
Figura 8. Set top box.....	21
Figura 9. Arquitectura de la Televisión Digital	25
Figura 10. Arquitectura Middleware Ginga.....	26
Figura 11. Arquitectura de Ginga CC.....	27
Figura 12. Arquitectura de Ginga NCL.....	28
Figura 13. Arquitectura de Ginga J.....	30
Figura 14. Logotipos de Ginga Argentina.....	31
Figura 15. Logotipos de Ginga Brasil.....	32
Figura 16. Funciones de Xerces-C.....	36
Figura 17. Funciones de Curl	37
Figura 18. Funciones de FFMPEG.....	40
Figura 19. Advertencia de Berkelium	41
Figura 20. Colocación del micro SD en la ranura	45
Figura 21. Prueba 1 de aplicación en Raspberry Modelo B.....	49
Figura 22. Aplicación corriendo sobre Ubuntu	50
Figura 23. Interactividad Programa 1	50
Figura 24. Aplicación corriendo sobre Raspbian Jessie	51
Figura 25. Estadísticas de medición de rendimiento con escritorio remoto ..	53
Figura 26. Medición de rendimiento con hardware externo	54
Figura 27. Rendimiento Escritorio Remoto vs. Hardware Externo.....	54
Figura 28. Nivel de procesamiento.....	56

RESUMEN

El presente proyecto evalúa el rendimiento y las prestaciones de montar un decodificador Ginga en una Raspberry Pi. Se cuenta con un estudio del estado del arte relacionado con la televisión interactiva, ISDB-Tb, hardware libre, Raspberry Pi, ventajas, beneficios, especificaciones, Ginga, lenguaje NCL de programación y aplicaciones realizadas con Ginga NCL. Se realizó un análisis del funcionamiento y manejo de la minicomputadora Raspberry Pi, a fin de conocer los sistemas operativos que posee para su ejecución siendo seleccionados para el este proyecto Raspbian Jessie y Ubuntu Mate (14.05). Para embeber Ginga en la Raspberry Pi, se analizó el mejor escenario realizando pruebas en dos modelos de la tarjeta, así como también evaluando los requisitos para la instalación del middleware, siendo escogido Ginga que provee TELEMIDIA (Brasil), adjuntando todos los pasos a seguir conjuntamente con los comandos adecuados para la instalación y funcionamiento del middleware. Se diseñó tres aplicaciones para realizar las pruebas de funcionamiento con actividades interactivas, las cuales tuvieron mejor reproducción en la Raspberry Pi 2 por su procesamiento y características.

Palabras clave:

- **GINGA**
- **RASPBERRY PI**
- **NCL**
- **ISDBT-B**
- **LUA**
- **FFMPEG**

ABSTRACT

This project tries to evaluate the performance and benefits of making a Ginga decoder on a Raspberry Pi. After an analysis on themes related to interactive television, ISDB-Tb, free hardware, Raspberry Pi, assets, benefits, specifications, Ginga, NCL language in programming and applications done with Ginga-NCL. It was performed an analysis on the performance and management of the minicomputer Raspberry, so that we could learn about the running system that it possess, for its later execution being Raspbian Jessie and Ubuntu mate (14.05) chosen for this project. In order to introduce Ginga in Raspberry Pi and to find the best scenario, there were made test with two different cards, at the same time there were evaluated those requirements needed for the installation of the middleware, being Ginga chosen, which it provided by TELEMIDIA (Brazil), attaching all steps to follow with the appropriate commands for its appropriate operation of the middleware. Three interactive applications were made to test the performance, as a result we concluded that Raspberry pi 2 has a better reproduction, thanks to its processing and characteristics.

Key words:

- **GINGA**
- **RASPBERRY PI**
- **NCL**
- **ISDBT-B**
- **LUA**
- **FFMPEG**

PRESENTACIÓN

El presente proyecto abarca el proceso de evaluación del rendimiento y las prestaciones disponibles de un decodificador de contenidos interactivos basados en GINGA-NCL sobre una Raspberry Pi.

Inicialmente se desarrolla los contenidos del proyecto como son: la presentación, resumen, objetivos, estado del arte, justificación e importancia y alcance.

En el segundo capítulo se describe los fundamentos teóricos que se requieren para poder conocer y comprender el funcionamiento de la Raspberry Pi, para ello se detalla sobre hardware libre, características, beneficios y sistemas operativos. Con respecto a la televisión interactiva se necesita tener en claro los conceptos de ISDB-Tb, OFDM y el middleware Ginga, detallando la arquitectura y los componentes como son: Ginga CC, Ginga NCL y Ginga J. Por último se detalla Ginga.ar y Ginga Telemidia, que son empresas que proveen el paquete completo del middleware.

El tercer capítulo está dedicado al estudio de los requerimientos para ser instalado el Ginga en la minicomputadora Raspberry Pi, así como la determinación de los sistemas operativos a ser ocupados y el método de instalación más adecuado del middleware.

En el cuarto capítulo se presenta la evaluación del funcionamiento de las aplicaciones interactivas sobre el middleware Ginga previamente embebido en la tarjeta Raspberry Pi modelo B y modelo 2.

El quinto capítulo se presenta las conclusiones y recomendaciones del proyecto, además se incluye la información, respaldada en formato digital, para la instalación de middleware Ginga en la Raspberry Pi, así como también las aplicaciones interactivas.

CAPÍTULO 1 : INTRODUCCIÓN

1.1. Antecedentes

En el Ecuador, el Ministerio de Telecomunicaciones y de la Sociedad de la Información y entidades gubernamentales del sector de las telecomunicaciones han trabajado arduamente para que se llegue a adoptar oficialmente el estándar japonés-brasileño (ISDB-Tb) para la Televisión Digital Terrestre. Este reto se consiguió el 26 de marzo del 2010 abriendo las puertas a la era digital, donde se ha logrado mejorar la calidad del servicio de televisión.

La implementación de la Televisión Digital Terrestre (TDT) genera una clara oportunidad de desarrollar la industria televisiva y aumenta el grado de pluralismo; democratizando el uso del espectro, así como el acceso a los contenidos garantizando la expansión comercial, esto ayuda en gran medida al país ya que se puede fundamentar en la implementación de software propio, siendo esta una oportunidad de negocio para conseguir el desarrollo tecnológico.

La digitalización de la señal de televisión y el mejor aprovechamiento del ancho de banda, permiten que las televisoras ofrezcan servicios adicionales al contenido audiovisual emitido, como: la recepción de la señal de televisión en los teléfonos móviles, información de la programación televisiva, acceso a Internet, interactividad, portabilidad y multiservicios.

La interactividad es el cambio más importante con relación al sistema analógico, dando la visión que los computadores unidos con las emisiones televisivas digitales, permitan el desarrollo de los sistemas interactivos, generando la relación entre la persona y el entorno digital definido por el hardware que los conecta. En este ámbito, el estándar ISDB-Tb trabaja mediante aplicaciones interactivas, las cuales son programadas por

informáticos y se ejecutan en el receptor de TDT; éstas pueden ser transmitidas por el aire al igual que los programas televisivos y se ejecutan al mismo tiempo que se presenta el programa de televisión. (Villanueva & Velásquez, 2010)

En el país existen grupos dedicados a la investigación de la Televisión Interactiva, uno de ellos se encuentra en La Universidad de las Fuerzas Armadas con el nombre de ESPETV que forma parte de la “Red temática en aplicaciones y usabilidad de la televisión digital interactiva” y de la “Red Iberoamericana para el desarrollo de la televisión digital terrestre y aplicaciones interactivas”, aportando con el Estado en la investigación a fin de conseguir nuevos retos y descubrimientos para plasmar estas ideas, y mejorar el ámbito tecnológico en el Ecuador.

1.2. Objetivos

1.2.1. General

Evaluar el rendimiento y las prestaciones disponibles de un decodificador de contenidos interactivos basados en GINGA-NCL sobre una Raspberry PI.

1.2.2. Específicos

- Estudiar el funcionamiento de la minicomputadora Raspberry PI.
- Determinar los parámetros para embeber GINGA dentro de la minicomputadora Raspberry PI.
- Diseñar una interfaz programable en NCL.
- Establecer el procedimiento para embeber aplicaciones interactivas.

- Comprobar el rendimiento y las prestaciones de la Raspberry PI al integrar aplicaciones interactivas.
- Implementar la reproducción, administración y actualización de contenidos interactivos.

1.3. Justificación e importancia del proyecto

El Ecuador está en un proceso de transición de analógico a digital, lo cual ha hecho que los canales actuales de televisión den paso a canales digitales que ya se están transmitiendo a la población, pero solo en ciertos equipos los televidentes puede observar el cambio, debido a que los televisores deben contar con características especiales; lo que se espera que en un futuro esto sea algo normal y cotidiano para todas las personas.

Además en el país existen ciertos proyectos de lanzamiento de la televisión interactiva, para que sea el televidente participe de los programas de su mayor interés, pero aún no se sabe a ciencia cierta si se lanzará al mercado la televisión interactiva por varias razones políticas, ya que se debe aceptar universalmente el lenguaje middleware superando el conflicto de uso del software libre y la utilización correcta del espectro.

La televisión interactiva hace uso de un middleware llamado GINGA mediante el cual se puede programar en JAVA o NCL, consiguiendo aplicaciones que funcionan por medio de la utilización de un control remoto. En el mercado tecnológico se cuenta con hardware libre, el cual permite realizar la integración del middleware GINGA obteniendo así varias funcionalidades, y sobre todo sería de un costo accesible a la población.

Se soluciona un problema de decodificación a bajo costo, aprovechando la tecnología mediante recursos multimedia como videos e imágenes; a pesar

de que no se llegue a concretar la TDT y la IPTV, se puede tener un equipo prototipo del diseño de un set top box para llegar a familiarizar a la población con esta tecnología, lo cual aporta al ámbito educativo, gubernamental, cultural, entre otros.

Además puede ayudar en gran medida a la población de menores ingresos y baja escolaridad, proveyendo de este dispositivo a escuelas del milenio, como también de personas especiales, facilitándoles el aprendizaje por su simplicidad en el funcionamiento por medio del control y la variedad de aplicaciones que pueden ser cargadas en el dispositivo para dar diferente uso según la necesidad.

1.4. Alcance del proyecto

El presente trabajo pretende analizar las características de cada uno de los accesorios para conformar un decodificador del middleware GINGA, como son: una minicomputadora Raspberry, el sistema operativo Ubuntu y GINGA NCL.

Se procede a la instalación del sistema operativo Linux (Ubuntu) en la minicomputadora Raspberry para embeber el middleware de tal manera que se pueda implementar una interfaz programable que permita ejecutar varias aplicaciones interactivas, que sean manejadas desde el teclado que se integrará al hardware.

Las pruebas se realizan en un televisor por medio de un cable HDMI comprobando el rendimiento y las prestaciones que se obtenga de la Raspberry PI, dichos programas serán manipulados por niños y personas con capacidades especiales, ya que el teclado se puede adaptar a las necesidades que presenta cada persona y así lograr un aprovechamiento de la tecnología para fines de aprendizaje.

Se logra integrar un decodificador independiente de la TDT, de memoria expandible mediante SD cards o flash memories que se pueden conectar en la Raspberry, que se acopla a las necesidades de las personas que hagan uso del contenido interactivo, además será un sistema de fácil administración y actualización.

1.5. Estudio del Estado del Arte

El estudio del estado del arte se ha realizado a trabajos de investigación y artículos que tienen relación con la temática de la televisión digital, en este artículo que tiene por nombre *“La Televisión Digital Terrestre y su incidencia en las estaciones televisivas ecuatorianas”* se realiza un análisis de la Televisión Digital Terrestre y su oportunidad de realizar actualizaciones tecnológicas dentro de la industria, ya que las preferencias de los usuarios están orientadas a recibir información y contenidos con multiservicios, multiplataformas y en multidispositivos. Planteando oportunidades de negocios, en la transición de Analógico a Digital, así como estimaciones de las inversiones que deben hacer los Broadcasters para digitalizarse en los sistemas de transmisión, y de esta forma evolucionar para convertirse en la puerta de acceso universal a la Sociedad de la Información. Además estudia las políticas adoptadas en el Ecuador para la transición nacional de TV Análoga a TV Digital, las frecuencias a utilizar, las normas técnicas, regulaciones, etc.

El proyecto de tesis titulado *“Implementación de un laboratorio de televisión digital que cumpla el ESTÁNDAR ISDB-TB”*, permite desarrollar aplicaciones interactivas que funciona sobre la plataforma Ginga-NCL/LUA, como también un servidor de aplicaciones interactivas y un servidor de base de datos para el almacenamiento de información generada por los televidentes. Como resultado de este proyecto, la Universidad de Cuenca cuenta con un laboratorio de televisión digital y una aplicación interactiva que permite capturar la actividad del usuario y almacenarla en una base de datos.

El proyecto titulado *“Estado del Arte de Receptores Set-Top-Box – Aplicaciones”* basa su estudio al sistema de televisión digital y su paso de la era analógica, además del análisis de la decodificación por el propio televisor o a través de un receptor llamado set-top-box, con la finalidad de que el telespectador participe indirectamente de la programación. Con la televisión digital interactiva se puede concretizar un dialogo del espectador con el programa a través del canal de retorno o del acceso de datos recibido y almacenados en el set-top-box. Por ultimo estudia las aplicaciones de la Televisión Digital, tales como: Guía de TV personalizada, Monitoreando y Control: Aplicaciones en Domótica y el voto electrónico T-Voto.

En el artículo *“Hardware libre: clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux”* hace referencia a los dos tipos de hardware, el estático, que tiene existencia física, y el reconfigurable, descrito mediante lenguajes HDL, centrándose en los criterios para considerarlos libres. Aborda también el hardware reconfigurable, centrándose en las herramientas de desarrollo para GNU/Linux. Además contiene ejemplos de sistemas diseñados enteramente bajo GNU/Linux, tanto de hardware estático como reconfigurable.

El proyecto de tesis titulado *“Manejo de software Ginga para el desarrollo de aplicaciones interactivas para televisión digital, basado en el estándar Brasileño ISDB-T”* estudia el manejo del middleware Ginga mediante el lenguaje de programación NCL. Empezando con una introducción a la TV Digital interactiva, los estándares de middlewares desarrollados para que los países en función de sus necesidades sociales y geográficas puedan adoptar por uno u otro. Se realiza un análisis de Ginga-J y Ginga NCL, para la construcción de aplicaciones, generando un manual de programación para investigaciones futuras.

Este proyecto *“Diseño e implementación de un prototipo para un sistema de generación de aplicaciones interactivas con Ginga-Ncl para la evaluación de Servicios Masivos”* presenta el desarrollo y funcionamiento del Sistema de Generación de Aplicaciones Interactivas (SGAi), el cual posee dos elementos

fundamentales: el programa “*Survey Composer*”, que permite crear y personalizar aplicaciones interactivas, y una Aplicación Web Administrable, encargada de proporcionar el sistema de gestión de encuestas, de recepción de los votos y visualización de resultados. La recepción de televisión digital y la ejecución de aplicaciones interactivas se ve realizada gracias al uso de un Set-Top-Box conectado al televisor, de esta manera los televidentes pueden interactuar con el televisor haciendo uso del control remoto, utilizando un canal de retorno, el cual permite que la interacción dúplex y así por medio de una aplicación acceder al sistema para poder utilizar y archivar los resultados.

CAPÍTULO 2 : MARCO TEÓRICO

2.1. Hardware Libre

En los años setenta las futuras grandes empresas del hardware y software de la actualidad se encontraban abiertos a compartir todos sus diseños electrónicos de aparatos informáticos, es decir las personas que asistían podían acceder a los planos, esquemas, PCBs de construcción de los dispositivos. (Espinola, 2014)

En los años noventa surge en el sitio Web de *Reinoud Lamberts*, *Open Design Circuits*, que fue el primero en proponer la creación de una comunidad del diseño del hardware con la visión de crear software libre. Las primeras tarjetas que permitieron esta práctica fueron las *FPGA*, proporcionando la práctica del intercambio de diseños libres electrónicamente, tal como como los programas pueden ser intercambiados.

Pero la comunidad que creció alrededor del sitio nunca agregó diseños libres por la carencia de software libre, por este motivo se debatió sobre el uso del software libre o el *freeware* comercial, donde no se llegó a concretar con ningún diseño real para el sitio.

Esto llevó a varias discusiones del tema dejando como base la creación de este sitio web, *Open Design Circuits* para una comunidad entera.

2.1.1. Definición

Hardware libre se denomina al soporte físico o al conjunto de elementos materiales que componen un computador. Hardware también son los componentes físicos de una computadora tales como el disco duro, dispositivo de CD-ROM, disquete, etc. El hardware se refiere a lo que es tangible, es decir todos los componentes físicos que se pueden tocar de la computadora.

El software libre y código abierto ofrece al usuario libertades. Existen licencias que las garantizan y que dan una cobertura legal, como por ejemplo la GNU GPL. El hardware abierto o libre toma estas mismas ideas del software libre y código abierto para aplicarlas en su campo. (Espinola, 2014)

Existen lenguajes de descripción de hardware que han impulsado a crear diseños abiertos como son *VHDL (Very high speed integrated circuit hardware description language)* o *Verilog HDL* de la misma manera que el movimiento *Open Source*.

2.1.2. Clasificación de Hardware Libre

Al hardware libre se lo ha clasificado por su naturaleza en:

Hardware estático: Conjunto de elementos materiales de los sistemas electrónicos tangibles.

Existen varios problemas por la confusión con la definición de software libre, debido que no se pueden aplicar las cuatro libertades del software al hardware, dando como resultado los siguientes puntos:

1. Un diseño físico es único
2. La compartición tiene asociado un coste
3. Disponibilidad de los componentes

Al desarrollar un hardware libre se tiene varias dificultades como las planteadas, que pueden ser superadas si se tiene para realizar una inversión de un coste y se difunde los planos para su construcción.

Hardware reconfigurable. Es el que se describe mediante un lenguaje HDL, (*Hardware Description Language*) y que permite especificar con todo detalle su estructura y funcionalidad. A partir de este código se generan unos









ficheros de configuración (*bitstreams*) para que los dispositivos del tipo FPGA se reconfiguren. (Rayo., Chimbo, & Valdivieso, 2013)

2.2. Raspberry PI

La Raspberry PI es una minicomputadora, de tamaño pequeño y de bajo costo. Es basada en ARM. Se le utiliza para estudiantes y aficionados.

Para iniciar la tarjeta, se debe instalar un sistema operativo y Linux es el sistema operativo (S.O.) base de ininidad de equipos a través de una SD card, la tarjeta soporta versiones de Linux como:

Tabla 1. Sistemas Operativos que soporta la Raspberry PI

<u>SISTEMA OPERATIVO</u>	<u>IMAGEN</u>
RASPBIAN	
UBUNTU MATE	
SNAPPY UBUNTU CORE	
OSMC	
OPENELEC	
PINET	
RISC OS	
WINDOWS 10	

El hardware de Raspberry PI es muy variado, ya que la pasar de los años se han desarrollado varios modelos como son:

- Modelos antiguos: A, A+, B, B+ e industrial

Tabla 2. Características de la Raspberry PI modelo antiguo

Característica	Especificación
CPU	700MHZ ARM1176-JZFS
GPU	Broadcom VideoCore 4
Memoria	256MB LPDDR2-800
Video	HDMI, compuesto
Audio	HDMI, stereo analog
USB	2 x USB2.0 (modelo B)
Almacenamiento	SD card
Red	10/100 Ethernet
Alimentación	5V micro USB

- Modelo nuevo: En enero 2015 apareció la Raspeberry Pi 2 modelo B.

Tabla 3. Características de la Raspberry PI modelo nuevo

Característica	Especificación
CPU	900MHZ quad-core ARM Cortex-A7
GPU	Broadcom VideoCore 4
Memoria	1GB de RAM
Video	HDMI, RCA
Audio	HDMI, estéreo analógico
USB	4 x USB2.0
Almacenamiento	SD card
Red	10/100 Ethernet
Alimentación	5V micro USB

Las partes de la Raspberry PI se especifican a continuación:

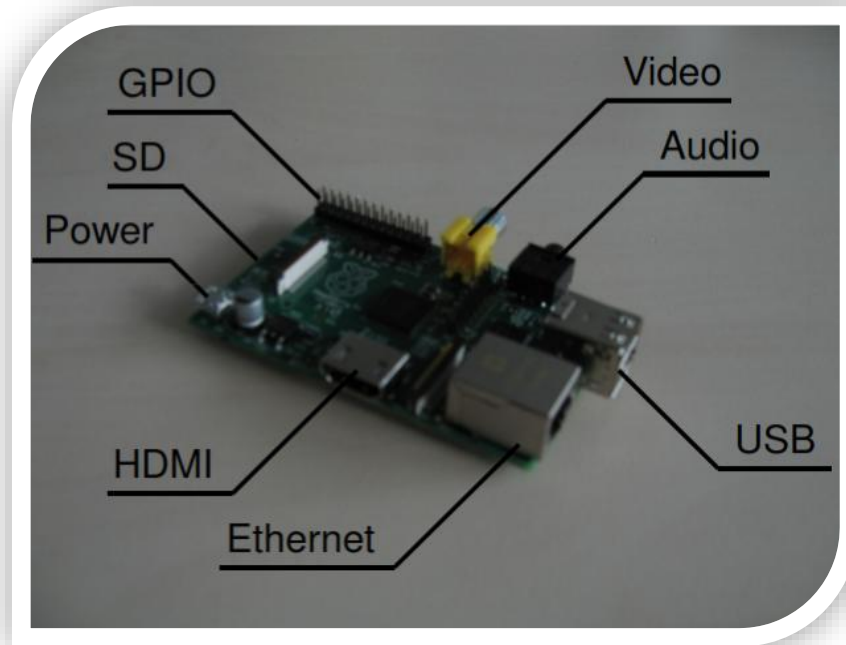


Figura 1. Partes de la Raspberry PI Modelo B

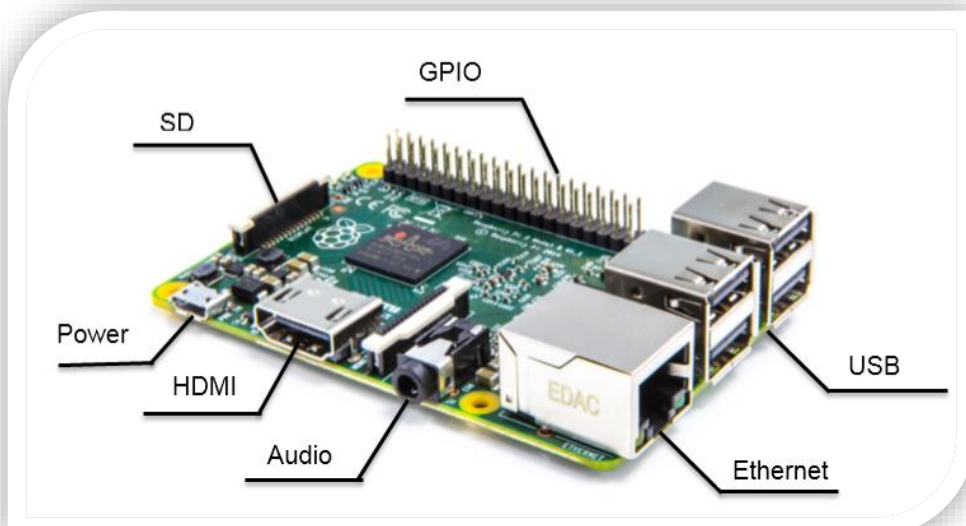


Figura 2. Partes de la Raspberry PI 2

Fuente: (Fundación Raspberry, 2015)

Para la utilización de esta tarjeta se necesita de entradas exteriores como son:

- Tarjeta SD de 8 GB.
- Alimentador con conector micro-USB 5V, 1.2 A.
- Monitor o TV con conexión HDMI o DVI.
- Cable HDMI 1.3a o HDMI-DVI.
- Teclado y mouse con cable USB 2.0
- Cable red ethernet RJ-45 para acceso a Internet.



Figura 3. Entradas exteriores de la Raspberry PI

2.2.1. Características físicas básicas de la Raspberry PI

1. Procesador BMC2835

El procesador es aquel que se encarga de los procesos del audio, video, procesamiento gráfico y comunicación de las interfaces del hardware. Estos procesos se realizan con la memoria RAM y con el coprocesador GPU.

La memoria RAM fue diseñada para el procesamiento de los datos. El coprocesador GPU cumple con las funciones de un acelerador de video.

Un total de 256 Mb (Modelo B) está repartido entre la memoria y el coprocesador. (Rayo., Chimbo, & Valdivieso, 2013)

2. Salida de Video

La Raspberry PI soporta tres estándares como son: RCA de televisores analógicos, HDMI de televisores digitales y DSI que se necesita de hardware adicional.

3. Conexión de Mouse, Teclado e Internet

La Raspberry PI posee 2 puertos USB (Modelo B) o 4 puertos USB (Raspberry PI 2), que se utilizan para la conexión de teclado y mouse.

Además posee un puerto Ethernet para la salida a Internet, ya que la tarjeta no posee WIFI.

4. Fuente de Alimentación

Posee un consumo de energía de 2,5 W, se necesita de un cable con salida a USB y micro USB. Teniendo cuidado con los parámetros de 5V y 1A.

2.2.2. Descripción de pines GPIO

GPIO (General Purpose Input/Output) es un sistema de E/S (Entrada/Salida) de propósito general, se puede realizar una serie de conexiones para usar como entradas o salidas. Estos pines están incluidos en todos los modelos de Raspberry Pi.

Todos los pines son de tipo unbuffered, que tiene como significado que no disponen de buffers de protección, por lo cual se debe tener cuidado con las magnitudes de voltajes e intensidad.

Las funciones de los pines GPIO son:

- **Pines de alimentación:** Existen pines de 5v, 3v3 (limitados a 50mA) y tierra (GND o Ground), que aportan alimentación a estos voltajes para los circuitos.
- **DNC (Do Not Connect):** Son pines que no tienen función, éstos se encuentran en las primeras versiones de la Raspberry Pi. En las actuales placas han sido marcados como GND.
- **GPIO normales:** Son conexiones configurables.
- **GPIO especiales:** Son pines destinados a una interfaz UART, con conexiones TXD y RXD que sirven para comunicaciones en serie.



Figura 4. Pines GPIO de Raspberry Modelo B

Fuente: (PE, 2015)



Figura 5. Pines de GPIO de la Raspberry Pi 2

Fuente: (PE, 2015)

2.2.3. Descripción de Sistemas Operativos

- **Ubuntu Mate**

Ubuntu MATE es un sistema operativo estable, fácil de usar, con un entorno de escritorio configurable. Es ideal para aquellos que quieren el máximo rendimiento de sus ordenadores y prefieren una metáfora de escritorio tradicional.

Martin Wimpress y Rohith Madhavan han realizado una imagen de Ubuntu Mate para Raspberry Pi 2.

La imagen es funcional y basada en el regular Ubuntu, lo que significa que el procedimiento de instalación de aplicaciones es el mismo que el de la versión de escritorio, es decir, utilizando apt-get.

Se puede utilizar cómodamente aplicaciones como LibreOffice, pero el rendimiento microSDHC es un problema que se soluciona utilizando una clase 6 o 10 de tarjeta microSDHC.

La primera vez que arranque la imagen MATE Ubuntu se ejecutará a través de un asistente de configuración donde se puede crear su propia cuenta de usuario y colocar la configuración regional. El primer arranque es bastante lento, pero las posteriores son mucho más rápidos. (Ubuntu, 2016)

- **Raspbian**

El sistema operativo Raspbian está basada en Debian Linux. Las versiones recientes de Raspbian se han basado en Debian Wheezy, pero Raspbian ahora se ha actualizado a la nueva versión estable de Debian, que se llama Jessie.

Jessie viene con un nuevo sistema de arranque predeterminado, *systemd*. El conjunto de software *systemd* proporciona muchas características interesantes como tiempos de arranque más rápidos, *cgroups* para servicios, la posibilidad de aislar parte de los servicios. El sistema de arranque *sysvinit* sigue disponible en Jessie. (Debian, 2016)

2.3. ISDB-Tb

ISDB-T (“Integrated Service Digital Broadcasting – Terrestrial” – Transmisión Digital de Servicios Integrados – Terrestre) ha sido desarrollado y está operando en Japón, y ha sido adoptado por Brasil.

ISDB-T ha probado el haber dado el mejor desempeño de DTTB (Transmisión Terrestre de TV Digital) en Japón durante más de 3 años. El servicio DTTB comenzó en Japón desde Diciembre del 2003, se ha podido migrar rápidamente al servicio DTTB debido a sus ventajas y nuevo servicio, llamado “One-Seg”, que es el servicio de recepción portátil en el mismo canal de transmisión, el cual comenzó en Abril del 2006. (Piscotta, 2010)

2.3.1. Características (Piscotta, 2010)

1. Alta calidad

En el ISDB-T, se han adoptado las siguientes tecnologías:

- Tecnología multiplex flexible (sistema MPEG-2)
- Sistema de codificación de video/audio flexible y de alta eficiencia (MPEG AAC)

Con esto se ha logrado varias clases de transmisión como son: HDTV, HDTV+SDTV y multicanales SDTV.

2. Robustez

Para dar robustez contra tales factores de degradación, ISDB-T adoptó el sistema de transmisión OFDM con la tecnología de "Time Interleave", lo cual proporciona menor potencia de transmisión, posibilidad de usar antenas de recepción internas y servicios de recepción móvil / portátil.

3. Flexibilidad

- **Del servicio:** Se utiliza MPEG-2 y MPEG AAC
- **De recepción:** OFDM

4. Utilización efectiva del recurso de las frecuencias

Adoptando el sistema de transmisión OFDM, es posible la construcción de una red de Isofrecuencia (SFN), dando como resultado la reducción de frecuencias para transmisores de relevo (repetidores), no es necesario cambiar el canal de recepción.

5. Interactividad

ISDBT-B adopto el Sistema de Transmisión Segmentada OFDM, que permite servicios de recepción fija, móvil, portátil en un mismo canal.

Además ISDBT-B adopto el servicio único “One-seg”, que usa 1 segmento de los 6MHz. Se instala fácilmente en los teléfonos celulares, PDA portátiles, sintonizadores USB, etc., permitiendo la transmisión en cualquier lugar.

6. Transferencia de datos

El servicio One-seg, usa un segmento del ancho de banda de 6MHz, no necesita otro canal, por lo que no necesita otro transmisor, permite ahorrar frecuencias y costos de infraestructura a la compañía transmisora. También opera con una recepción de banda muy estrecho, esta operación ahorra consumo de energía. (Piscotta, 2010)

7. Compatibilidad

Para reducir el costo de los receptores, se usa una tecnología común para los receptores digitales para satélite, terrestres y cable. Como resultado el chip de recepción que usan estos receptores digitales es común.

2.3.2. Modulación OFDM

Las características del canal radioeléctrico no se mantienen constantes en el tiempo, en cambio las características de propagación del canal radioeléctrico tienden a mantenerse más o menos estables dentro de un cierto período de tiempo, donde se dice que las condiciones son estables.

Determinando que:

- El dominio de la frecuencia se divide en una cierta cantidad de “sub-bandas” de reducida anchura de banda comparada con el total disponible en el canal.
- El dominio del tiempo se divide en pequeños “intervalos de tiempo”.

Donde cada sector de la partición frecuencia/tiempo se utiliza, para colocar una portadora específica, además:

- Un determinado número de portadoras transmitidas en un intervalo de tiempo se denomina “Símbolo OFDM”.
- Una sucesión de n Símbolos OFDM se denomina “Cuadro OFDM”. Más adelante se verá que el sistema ISDB-T emplea cuadros compuestos por 204 símbolos y se justificará su valor.

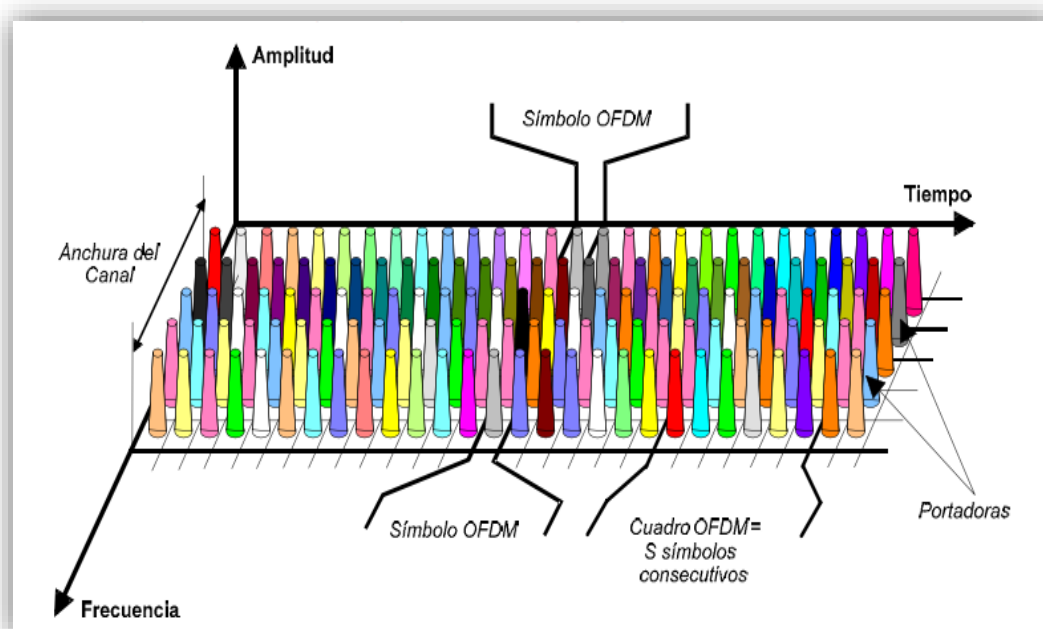


Figura 6. Distribución de portadoras

Fuente: (Piscotta, 2010)

Durante cada intervalo de tiempo, las portadoras son moduladas con unos pocos bits (entre 2 y 6) de datos codificados. El número de bits transmitidos por cada portadora dependerá del tipo de modulación empleado.

Concretamente, para ISDB-T:

- 2 bits para QPSK y DQPSK
- 4 bits para 16-QAM
- 6 bits para 64-QAM. (Galabay & Vivar, 2012)

2.4. Televisión Digital Interactiva

La televisión digital convierte las señales analógicas a un formato binario, que es emitido por un satélite, terrestre o por cable, para luego pasar por un proceso de decodificación por el televisor a través de un set top box.

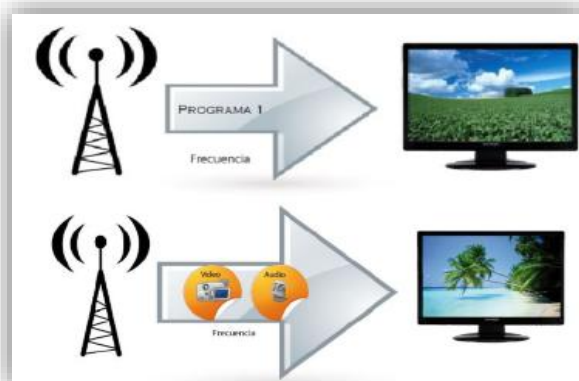


Figura 7. Transmisión Analógica

Fuente: (Galabay & Vivar, 2012)

El **set top box** es un hardware que permite que un televisor convencional pueda reproducir señales televisivas producidas con tecnología digital, a los cuales se les instala un middleware, para soportar características de interactividad con los usuarios y además que brinden funcionalidades.



Figura 8. Set top box

Fuente: (Villanueva & Velásquez, 2010)

La televisión interactiva funciona a través de datos y servicios interactivos, para mejorar la señal HDTV en la recepción y visualización de señales en varios dispositivos como fijos y móviles. Además se logra mayor eficiencia en el aprovechamiento del espectro radioeléctrico, ofreciendo más canales al usuario.

La Superintendencia de Telecomunicaciones, adoptó oficialmente el estándar Brasileño ISDB-Tb, que es una modificación del estándar Japonés de TV Digital ISDB-T. Entre las modificaciones están el uso de MPEG-4 en lugar de MPEG-2 para la codificación de video y la aparición del middleware Ginga como plataforma de desarrollo y presentación de los contenidos interactivos. (González, González, & Gómez, 2003)

2.4.1. Ventajas de la TVD

- Mejora de la calidad de imagen y sonido.
- Aumento del número de canales de televisión ofertados por banda UHF.
- Mayor flexibilidad de las emisoras y servicios adicionales.
- Se puede transmitir audio, video, imágenes y texto.

2.4.2. Elementos tecnológicos

Para lograr la interactividad con el receptor de TDT del estándar ISDB-Tb, se requieren una serie de elementos que se detallan a continuación:

- **Aplicación Interactiva:** permite la interacción entre televidente y receptor.
- **Servidor de Aplicaciones:** es el encargado de almacenar y preparar las aplicaciones interactivas antes de su emisión; puede o no estar conectado al canal de retorno.

- **Middleware:** software que se encuentra instalado en el Set Top Box (STB) que tiene como objetivo ejecutar aplicaciones interactivas.
- **Canal de retorno o interactividad:** permite al televidente recibir y enviar datos hacia un servidor.
- **Set Top Box (STB):** es un dispositivo conectado al televisor que decodifica y demultiplexa la señal (audio, video y datos) para convertirla en señal analógica (para el caso de video y audio) que se envía al televisor, pudiendo ser esta de alta definición. Contiene el *middleware* y permite el uso del canal de retorno para las aplicaciones interactivas. (Alvarez, Balanguer, Costa, & Woites)

2.4.3. Beneficios de la Interactividad

Los beneficios de la interactividad en ISDB-Tb son los siguientes:

- **Complementa los programas televisivos:** Las aplicaciones pueden ir ampliando la información del programa de televisión que se observa.
- **Aplicaciones interactivas amplias y variadas:** Abordan temas de entretenimiento, educación información, etc.
- **Sistemas de prevención y de alerta temprana:** Las aplicaciones interactivas pueden contener campañas de información sobre la prevención de enfermedades y alerta ante desastres.
- **Fácil acceso:** Se necesita que le televidente cuente con un receptor de TDT instalado *middleware* para poder acceder a una programación interactiva.
- **Mayor conocimiento de los clientes:** Ayuda en casos de estudios a la población por medio de encuestas de diferentes temas según la necesidad.

- **Personalización de las aplicaciones interactivas:** Las aplicaciones son realizadas de manera creativa por los programadores.
- **Interacción externa:** Mediante el canal de retorno el televidente puede acceder a sitios de Internet.

2.5. Middleware GINGA

El middleware es un componente fundamental para la reproducción de aplicaciones interactivas en el lado del receptor, con la ayuda de una serie de herramientas de software permite el desarrollo de aplicaciones interactivas que se ejecutarán sobre el middleware.

El middleware que se utiliza es Ginga, el cual permite la ejecución de aplicaciones interactivas en un receptor de TV Digital terrestre. La especificación de Ginga se encuentra descrita en la norma ABNT NBR 15606.

Ginga es una especificación abierta de fácil aprendizaje, permitiendo la producción de contenido interactivo, lo que dará un nuevo impulso a los canales de televisión comunitarios y a la producción de contenido por las grandes emisoras. Adoptando la licencia GPLv2.

La arquitectura de la televisión digital, es modificada agregándole una capa del middleware Ginga, para que sé de paso a la televisión interactiva, y funcionen sobre esta capa las aplicaciones para que estén accesibles al usuario. En la figura 9, se puede observar la arquitectura base y la adición del middleware.

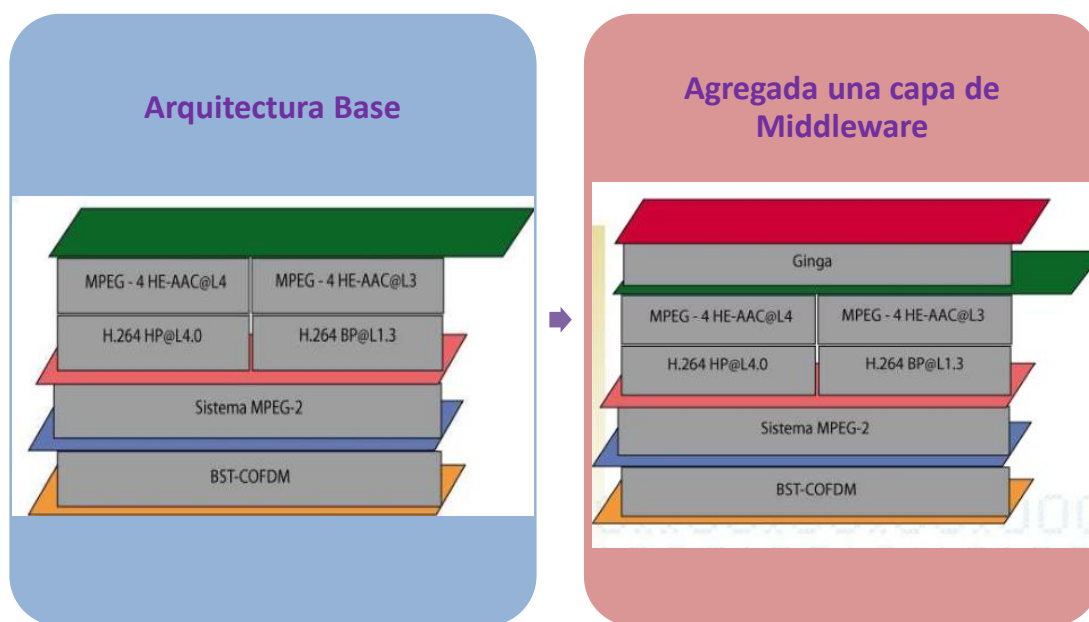


Figura 9. Arquitectura de la Televisión Digital

Fuente:(Zambrano., 2015)

Ginga es la capa de software intermedio, que permite el desarrollo de aplicaciones interactivas para TV Digital, independientemente de la plataforma de hardware de fabricantes de terminales de acceso (STB).

El middleware Ginga se divide en dos subsistemas interconectados:

- **Ginga J:** Se utiliza para aplicaciones de procedimiento Java.
- **Ginga NCL:** Se utiliza para la presentación de aplicaciones declarativas escritos en lenguaje NCL.

En Ginga una aplicación declarativa puede hacer referencia a una aplicación procedural, de la misma manera, una aplicación procedural puede hacer referencia a una aplicación declarativa.

2.5.1. Arquitectura del Middleware Ginga

La arquitectura del middleware Ginga, puede ser dividida en tres módulos:

1. Ginga CC (Common Core)
2. Entorno de presentación Ginga NCL (declarativo)
3. Entorno de ejecución Ginga J (procedural)



Figura 10. Arquitectura Middleware Ginga

Fuente: (Cabezas. & Quezada, 2012)

2.6. Ginga CC

GINGA-CC es el subsistema lógico que provee toda la funcionalidad común al soporte de los ambientes: declarativo (GINGA-NCL) y procedural (GINGA-J).

La arquitectura del sistema permite que únicamente el módulo Ginga-CC deba ser adaptado a la plataforma en la cual será implementado, de esta manera el Ginga-CC provee un nivel de abstracción de la plataforma de hardware y del sistema operativo, accesible a través de las APIs. Esto le permite interactuar con el acceso al sintonizador de canal, con el sistema de archivos, el terminal gráfico entre otros. (Cabezas. & Quezada, 2012)

Posee un conjunto de exhibidores monomedia comunes, los cuales presentan las medias de audio, video, texto e imagen. El acceso a tales exhibidores es a través de adaptadores, que se encargan de notificar eventos de presentación y selección del usuario. Entre los exhibidores también se encuentra el exhibidor HTML ofrecido por el *middleware* Ginga.

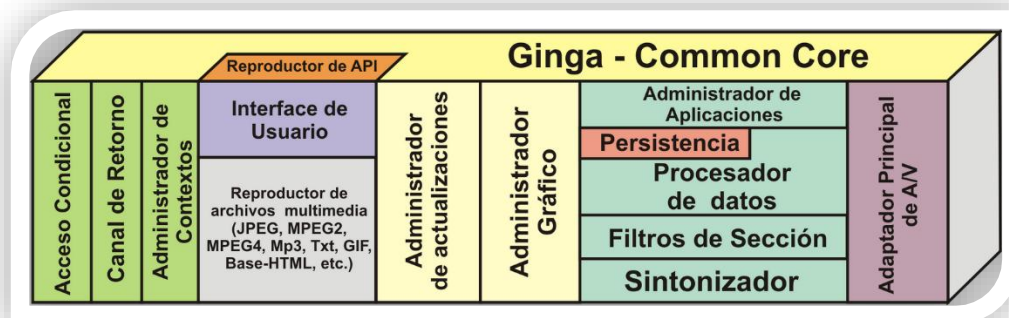


Figura 11. Arquitectura de Ginga CC

Fuente: (Galabay & Vivar, 2012)

2.7. Ginga NCL

Ginga NCL fue desarrollado por la Pontificia Universidad Católica de Río de Janeiro, la cual tiene una presentación multimedia para aplicaciones declarativas escritas en el lenguaje NCL.

NCL es un lenguaje de aplicación XML con elementos diseñados para la especificación de los aspectos de la interactividad, la sincronización espacio-temporal de los objetos multimedia, capacidad de adaptación y soporte para múltiples dispositivos.

Ginga NCL permite el soporte a dos lenguajes procedurales LUA y JAVA.

2.7.1. Arquitectura de Ginga NCL

La arquitectura de Ginga NCL se la puede observar en la figura 12.



Figura 12. Arquitectura de Ginga NCL

Fuente:(Torres, 2011)

- **Formateador**

Es el núcleo del entorno de presentación Ginga-NCL, que tiene como función recibir y controlar la ejecución de una aplicación NCL, además asegura las relaciones espacio-temporales de la aplicación. Como también se encarga de recibir las aplicaciones entregadas por el Ginga-CC y las deposita en una estructura de datos llamada “base privada”.

- **Analizador de XML y Conversor**

Cuando se ejecuta una aplicación el formateador solicita al Analizador XML y al Conversor realizar la traducción de la aplicación NCL a la estructura interna de datos GingaNCL, para verificar los datos de la aplicación.

- **Programador y Administrador de Reproducción**

El programador es iniciado para organizar el orden de la presentación del documento NCL, capturando los objetos media, programando sus enlaces y guiando el flujo de la presentación.

El administrador de reproducción que se encarga de crear una instancia de reproducción y ejecutar el objeto media cuando sea requerido.

- **Administrador de Bases Privadas**

El administrador de la base privada tiene la función de recibir comandos para activar y manipular las aplicaciones.

- **Administrador del Diseño**

Ginga-NCL soporta múltiples dispositivos de presentación al mismo tiempo, debido a que asigna todas las regiones definidas en una aplicación.

- **Administrador de Contexto NCL**

Soporta el contenido por la información procedente del Gestor de Contexto del Ginga-CC.

- **Administrador de Recuperación**

El Administrador de Recuperación detecta, controla e implementa los procedimientos de recuperación de fallas.

2.7.2. NCL- lua

Lua es un lenguaje script de NCL diseñado para una programación procedimental con utilidades para la descripción de datos, además da soporte para la programación orientada a objetos, programación funcional y programación orientada a datos.

Una aplicación NCL-Lua puede correr en pantalla completa o junto con el video principal, mientras recibe eventos del Flujo de Transporte o el control remoto. La ejecución de aplicaciones NCL-Lua se basa en dos elementos fundamentales.

1. La existencia de un carrusel de objetos que contiene las aplicaciones a correr y los archivos de datos.
2. La señalización por medio de eventos que llegan por el Flujo de Transporte (BTS). (Alvarez, Balanguer, Costa, & Woites)

2.8. Ginga J

Ginga J fue desarrollado por la Universidad Federal de Paraiba, para proveer una infraestructura de ejecución de aplicaciones basadas en lenguaje Java, con facilidades para el ambiente de interactividad en TV Digital.

El Ginga-J es el subsistema lógico del *middleware* Ginga responsable por el procesamiento de aplicaciones imperativas escritas utilizando el lenguaje Java.

Una aplicación Java puede actuar como entidad hija de una aplicación declarativa, o como una entidad inicial controlando el ciclo de vida de una entidad hija declarativa.

2.8.1. Arquitectura de Ginga J

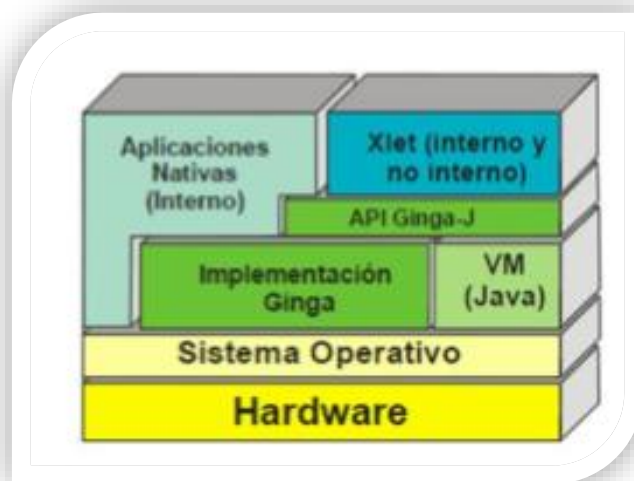


Figura 13. Arquitectura de Ginga J

Fuente: (Galabay & Vivar, 2012)

La arquitectura de Ginga J conforman las entidades de hardware o de recursos, software del sistema y las aplicaciones.

Las aplicaciones nativas están conformadas de funciones no estandarizadas, provistas por el sistema operativo del dispositivo Ginga, también pueden estar provistas por las APIs estandarizadas.

2.9. Ginga.ar

Ginga.ar es una implementación desarrollada por el laboratorio LIFIA de la Universidad Nacional de La Plata, a partir de la implementación de referencia de Ginga-NCL creada por la PUC de Rio de Janeiro. Ginga.ar está desarrollado en C++, es Software Libre y las licencias utilizadas son GPLv2 y LGPLv2. Ginga.ar permite ejecutar aplicaciones interactivas escritas en NCL (Nested Context Language). Ginga.ar además se ejecuta en distribuciones Linux desktop de 32 y 64 bits, además de Windows. (Lifia, 2013)



Figura 14. Logotipos de Ginga Argentina

Fuente:(Lifia, 2013)

2.10. Ginga Brasil

Ginga Brasil tiene como premisa fundamental que la verdadera inclusión no ocurra solamente mediante el acceso a la información, sino también el derecho de saber acerca de la generación de información y el derecho a emitir.

Ginga NCL es un entorno de presentación multimedia para aplicaciones declarativas escritas en NCL y su lenguaje de scripting Lua.

El ambiente de presentación Ginga-NCL es el subsistema lógico necesario de Ginga, responsable de la ejecución de aplicaciones NCL. Las aplicaciones se entregan en el Ginga-NCL por el subsistema “Núcleo Común Ginga” (Ginga-CC).

Ginga es el resultado del desarrollo de proyectos de investigación coordinados por los laboratorios Telemídia en la PUC-Rio y LAViD en la UFPB. (Gomes, 2016)



Figura 15. Logotipos de Ginga Brasil

Fuente: (Gomes, 2016)

CAPÍTULO 3 : DISEÑO E IMPLEMENTACIÓN

3.1. Determinación de requerimientos

En el presente proyecto se presenta una topología de una Raspberry Pi embebido GINGA NCL, con programas de interacción con el usuario.

El middleware GINGA es formado por un conjunto de software ubicado entre el código de las aplicaciones y la infraestructura de ejecución, está formado por tecnologías estandarizadas e innovaciones que lo convierten en la especificación de middleware más avanzada.

Ginga es proveído por la LIFIA (Argentina) y por TELEMÍDIA (Brasil). Existen versiones tanto para Windows como para Linux, cada una de las versiones cuenta con requerimientos, que deben ser instalados para obtener un funcionamiento satisfactorio del middleware.

3.2. LIFIA (Argentina)

Ginga.ar es un software Libre, las licencias utilizadas son GLPV2 y LGPLv2, e incluye librerías con otras licencias libres.

La versión 2.2 de Ginga.ar forma parte de Kuntur. **Kuntur** contiene software de recepción (zapper) y el middleware para soportar la interactividad (**Ginga.ar**) ya integrados. A partir de este software se puede construir firmware de recepción que soporta la descarga y ejecución de aplicaciones interactivas.

Versiones previas a la versión 2.2 pueden descargarse e instalarse, el requerimiento para este proyecto solo es Ginga.ar, por ese motivo se analiza la versión 2.1., la cual requiere los siguientes componentes:

- Ubuntu 12.04
- Usuario: ginga, clave: ginga

Además se debe instalar las siguientes dependencias, en Ubuntu:

- Boost
- XERCEC
- EV
- DB
- GTK2
- GDK-PIXBUF
- GLIB PANGO
- CAIRO
- LIBSOUP2
- WEBKIT
- LIBVLC
- LUA
- CURL

La versión del sistema operativo Ubuntu 12.04, no se puede obtener a través de las imágenes disponibles para Raspberry Pi, por lo cual requerirá que se cree la imagen.

3.3. TELEMÍDIA (Brasil)

La versión actual de Ginga Telemídia es 0.13.6, que tiene varios requisitos para su instalación los cuales son:

- Sistema Operativo: Debian

Las dependencias que se debe instalar en Debian:

- XERCEC
- CURL
- SDL2
- SDL2_image

- SDL2_ttf
- FFMPEG
- NCLUA
- Berkelium

El sistema operativo Debian es equivalente a Raspbian, la versión actual que soporta la Raspberry Pi es Jessie, por lo cual la instalación puede ser directamente de la imagen que provee la página oficial de la tarjeta. Además las pruebas se realizarán en Ubuntu para comprobar la portabilidad del middleware.

Por este motivo, el middleware escogido para el presente proyecto es Ginga de Telemidia, para lo cual se debe instalar todos los paquetes antes mencionados, los que se encuentran detallados en el Anexo 4.

3.4. Descripción de requerimiento de Ginga Telemidia

3.4.1. Instalación de XERCES-C

Xerces-C ++ es un analizador XML, es validado y escrito en un subconjunto portable de C ++. Xerces-C ++ hace que sea fácil de darle a su solicitud, la capacidad de leer y escribir XML de datos. Se proporciona una biblioteca compartida para analizar, generar, manipular y validar documentos XML utilizando el DOM, SAX, y APIs SAX2. (The Apache XML Project, 2015)

3.4.1.1. Funciones

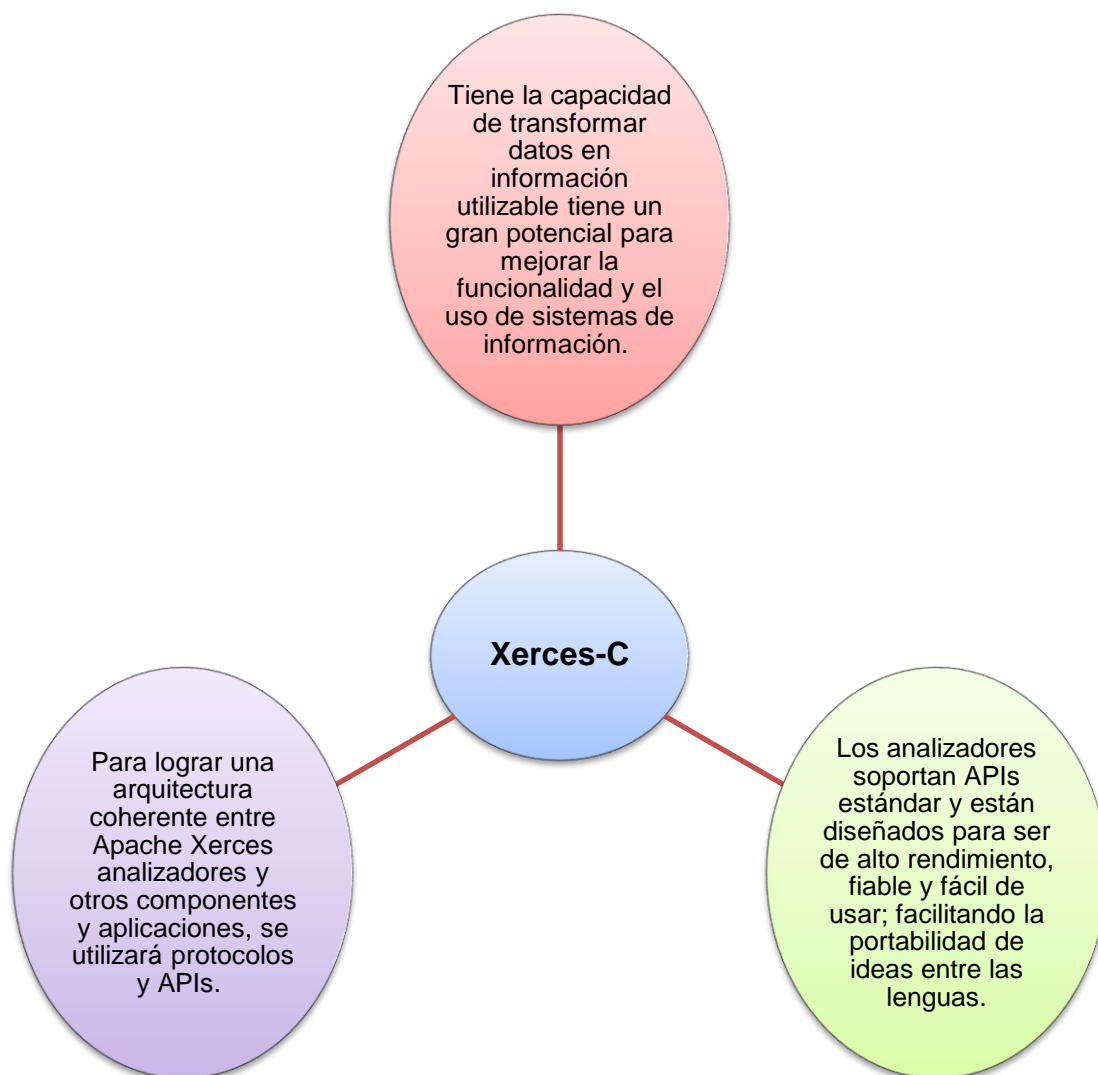


Figura 16. Funciones de Xerces-C

3.4.2. Instalación de CURL

CURL es una herramienta software para transferencia de archivos con sintaxis URL mediante intérpretes de comandos que soporta FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, FILE y LDAP. (Wikipedia, 2015)

3.4.2.1. Funciones

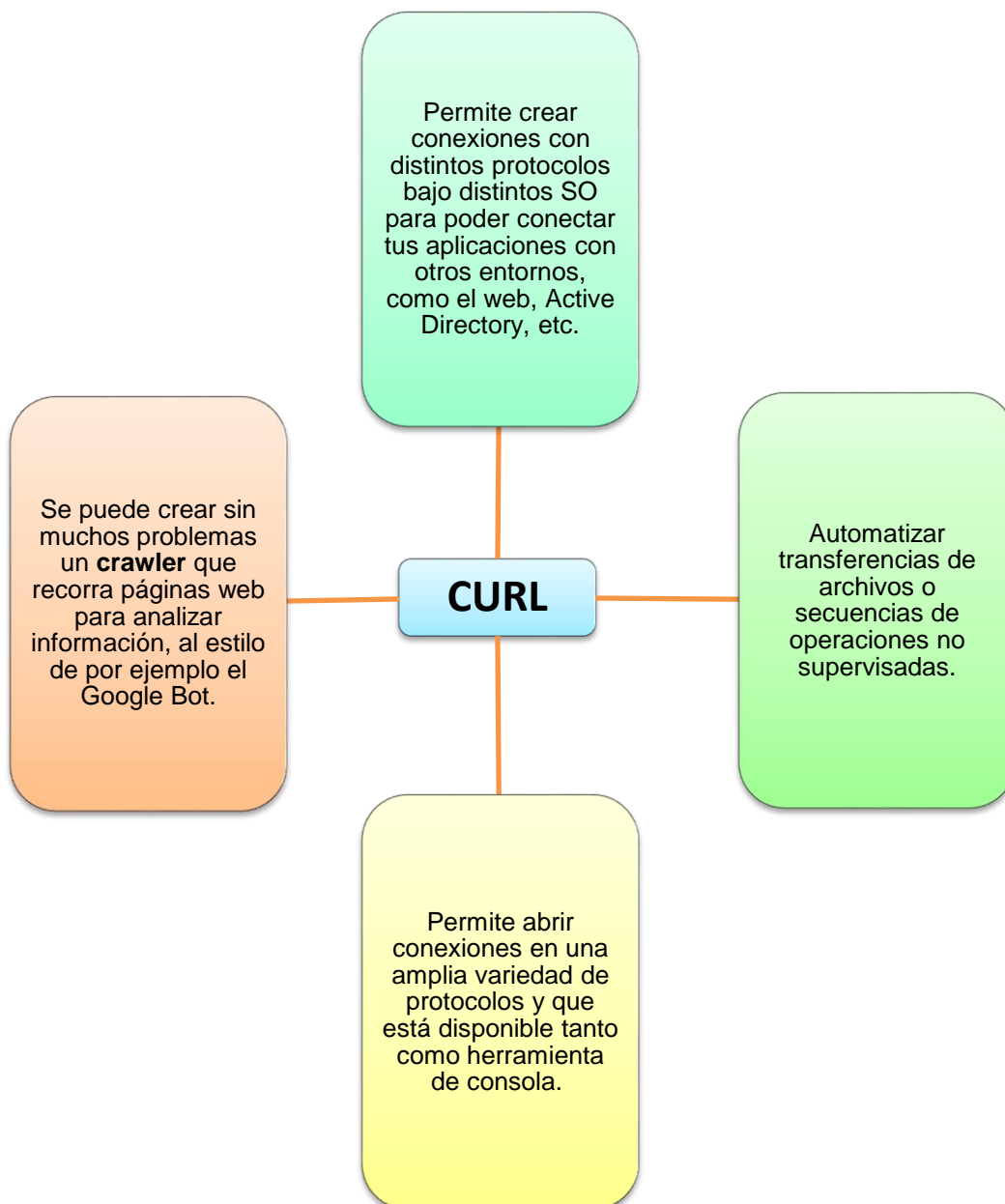


Figura 17. Funciones de Curl

3.4.3. Instalación de SDL2

Simple DirectMedia Layer es una librería multiplataforma de desarrollo diseñado para proporcionar acceso a bajo nivel para audio, teclado, ratón,

joystick, y los gráficos de hardware mediante OpenGL y Direct3D. Es utilizado por el software de reproducción de vídeo, emuladores y juegos populares.

SDL apoya oficialmente Windows, Mac OS X, Linux, iOS y Android. Soporte para otras plataformas se puede encontrar en el código fuente.

SDL está escrito en C, funciona de forma nativa con C ++, y hay enlaces disponibles para varios otros idiomas, incluyendo C # y Python.

SDL 2.0 se distribuye bajo la licencia zlib. Esta licencia le permite utilizar SDL libremente en cualquier software. (Simple Direct Layer, 2015)

3.4.4. Instalación de SDL2_image

SDL_image es una biblioteca de carga de archivos de imagen.

Carga imágenes como superficies SDL y texturas , y es compatible con los siguientes formatos : BMP , GIF , JPEG , LBM , PCX , PNG , PNM , TGA, TIFF, WebP, XCF , XPM , XV. (Lantinga Sam, 2015)

3.4.5. Instalación de SDL2_ttf

SDL2_ttf se trata de una biblioteca de muestras que le permite utilizar las fuentes TrueType en sus aplicaciones SDL. Viene con un "ShowFont" programa de ejemplo que muestra una cadena de ejemplo para un archivo de fuentes TrueType dado. (Sam., 2015)

3.4.6. Instalación de FFMPEG

FFmpeg es el marco multimedia líder, capaz de decodificar, codificar, transcodificación, mux, demux, **filtrar** y **reproducir** casi cualquier cosa que los humanos y las máquinas han creado. Es

compatible con los formatos antiguos más oscuros hasta la vanguardia. No importa si fueron diseñados por algunos estándares, la comunidad o una sociedad anónima. También es altamente portátil: FFmpeg compila, corre y pasa nuestras pruebas infraestructura FATE través de Linux, Mac OS X, Microsoft Windows, los BSD, Solaris, etc. en una amplia variedad de entornos de compilación, arquitecturas y configuraciones. (FFMPEG, 2015)

FFmpeg contiene:

- libavcodec
- libavutil
- libavformat
- libavfilter
- libavdevice
- libswscale
- libswresample

Cada una de estas librerías son indispensables para la reproducción de GINGA, pero para la compilación de GINGA se necesita añadir el sufijo ffmpeg, por ejemplo: libavcodecffmpeg.

Para ello se debe compilar con los comandos expuestos en el Anexo 4, teniendo en cuenta que lo que añade el sufijo es:

`build-suffix=ffmpeg`

3.4.6.1. Función

Las librerías que complementan el paquete FFmpeg permiten:

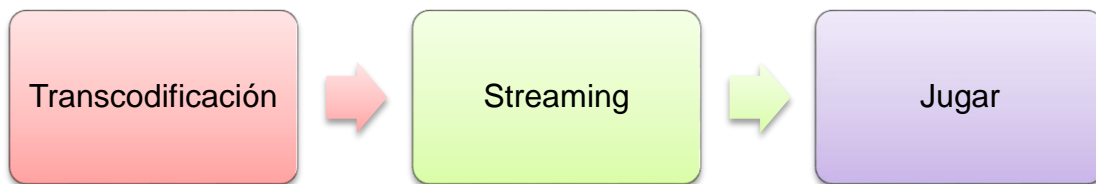


Figura 18. Funciones de FFmpeg

3.4.7. Instalación de NCLUA

Lua es un lenguaje de programación extensible diseñado para una programación procedimental general con utilidades para la descripción de datos.

NCL está estructurado de una manera que separa el contenido y la estructura de la aplicación permitiendo definir objetos media y relacionarlos con el tiempo y el espacio.

NCLUA es lenguaje por el cual se puede programar las actividades interactivas, y por el cual pueden ejecutarse las aplicaciones sobre GINGA.

Esta librería provee TELEMEDIA, es decir GINGA Brasil, por ello solo se descarga y se ejecuta para cumplir con los requisitos de GINGA.

Además depende de otras librerías que deben ser instalados para no tener inconvenientes en la instalación como son:

- Lua
- Pango
- Cairo

Los comandos que fueron ejecutados para dicha instalación se puede ver en el Anexo 4.

3.4.8. Instalación de Berkelium

Berkelium es una biblioteca con licencia BSD que proporciona una representación del navegador fuera de la pantalla, a través de código abierto del navegador web de Google Chromium.

En el tutorial que provee TELEMEDIA indica que se necesita la librería BERKELIUM, la cual se encuentra obsoleta y descontinuada, por ese motivo no se debe realizar la instalación de la misma, al momento de compilar el GINGA dará un cuadro de advertencia como se muestra en la figura 19 indicando que es necesaria la librería, pero no afecta a la funcionalidad de las aplicaciones.

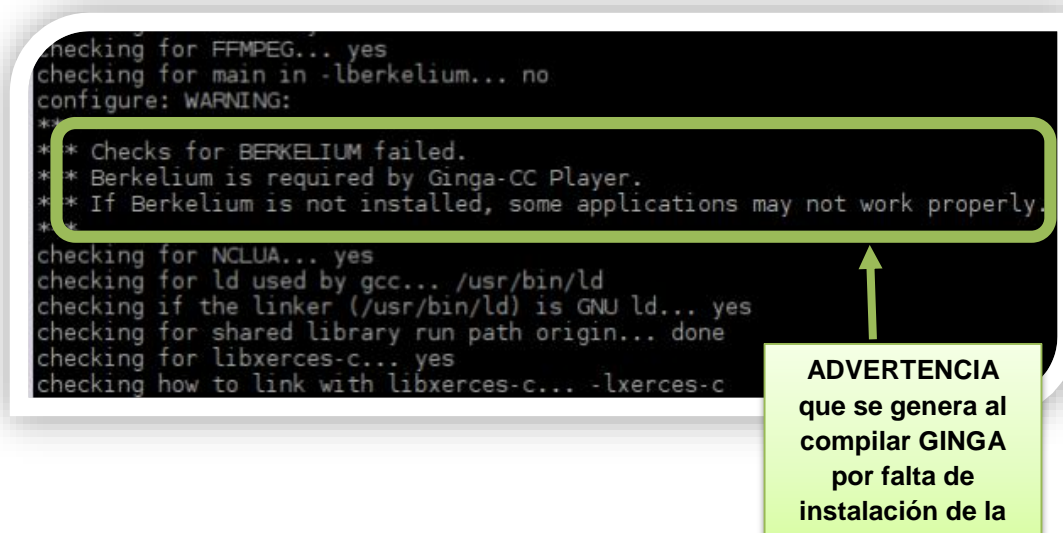


Figura 19. Advertencia de Berkelium

3.5. Análisis de la tarjeta Raspberry Pi

Raspberry Pi es un ordenador de placa reducida que cuenta con varios modelos, para el presente proyecto se utilizará el modelo B (antiguo) y el modelo 2 (actual).

3.5.1. Raspberry Pi Modelo B

La Raspberry Pi modelo B, presenta las siguientes características:

Tabla 4. Características principales de la Raspberry Pi Modelo B

Fuente: (Fundación Raspberry, 2015)

Características principales	
-	QUAD Core Broadcom BCM2835 CPU con 512MiB RAM.
-	40 pines de extensión GPIO.
-	Slot de SD, ranura para SDIO.
-	Múltiples puertos: 2 puertos USB, tamaño completo HDMI, conector de 3.5mm, CSI puerto de la cámara y DSI puerto de pantalla.
-	Fuente de alimentación micro USB.
-	Consumo energético de 700mA (3.5W)

Esta Raspberry no permite la instalación del sistema operativo Ubuntu, por ello se ha procedido hacer las pruebas solo en Raspbian Jessie, para establecer una comparación con la versión más actual de la minicomputadora.

3.5.2. Raspberry Pi 2

La Raspberry Pi versión 2 es la más actual en el mercado y cuenta con características muy interesantes como:

Tabla 5. Características principales y beneficios de la Raspberry Pi 2**Fuente: (Fundación Raspberry, 2015)**

Características principales y beneficios	
-	QUAD Core Broadcom BCM2836 CPU con 1GB RAM.
-	40 pines de extensión GPIO.
-	Slot de Micro SD.
-	Múltiples puertos: 4 puertos USB, tamaño completo HDMI, salida estéreo de 4 polos y el puerto de vídeo compuesto, CSI puerto de la cámara y DSI puerto de pantalla.
-	Fuente de alimentación micro USB.

La Raspberry Pi 2 necesita de una tarjeta SD para su funcionamiento, en la cual debe ser instalado el sistema operativo que se ha elegido, en este caso es Raspbian Jessie o Ubuntu que son versiones Linux, por lo cual su instalación se debe realizar por medio de comandos.

3.6. Integración del sistema operativo en la Raspberry Pi

La Raspberry Pi debería funcionar con cualquier tarjeta SD, pero es importante tener en cuenta el tamaño mínimo recomendado que es de 4GB, sin embargo para no tener inconvenientes con la falta de espacio dependiendo la utilización que se le vaya a dar, recomiendo una tarjeta de 8GB.

Además la tarjeta SD tiene como característica la clase, la cual determina la velocidad de escritura sostenida, sin embargo no determina el uso general que se dé a la misma, porque a menudo esta velocidad de escritura se logra de la velocidad de lectura y el aumento de los tiempos de búsqueda.

La Raspberry Pi 2 Modelo B requiere tarjetas micro-SD, ya que cuenta con el slot para la misma sin requerir adaptador de SD de tamaño completo. En

caso de utilizar una Raspberry Pi de versiones anteriores se requiere un lector de tarjetas para que pueda ser corrido el sistema operativo escogido.

La fuente de alimentación de la Raspberry debe ser de buena calidad junto con el cable para evitar daños y bajas de voltaje para el equipo.

Para acceder al sistema operativo se lo hace desde la página oficial de la Fundación Raspberry Pi. (Ver Anexo 1)

3.6.1. Instalación de la imagen en la tarjeta SD

La instalación de la imagen en la tarjeta SD se puede realizar desde tres sistemas operativos como son: (Ver Anexo 2)

- Linux
- Windows
- Mac OS

Al instalar la imagen en la tarjeta SD, la memoria se particiona, por lo cual se debe extender la utilización de toda la tarjeta para poder tener espacio para la instalación de los programas requeridos.

Después se de ingresar la SD a la ranura como se puede observar en la imagen 20, y dar arranque a la Raspberry. Además se debe realizar la instalación del escritorio remoto, a fin de no ocupar periféricos y solo ocupar una laptop. (Ver Anexo 3)



Figura 20. Colocación del micro SD en la ranura

3.6.2. Instalación de GINGA

Ginga es el programa que da vida a la televisión digital convencional, ya que por medio de la interactividad permite la interacción con el usuario, el aprendizaje y la diversión.

Ginga es parte de TELEMEDIA, este programa está disponible para realizar la descarga y empezar a compilar una vez instalado todas las librerías y paquetes dependientes, para poder empezar a realizar las pruebas correspondientes a la interactividad.

Este programa está basado en scripts que solo se deben ejecutar, ya que están previamente probados y listos para ser ocupados por aplicaciones que pueden ser programadas o creadas por cualquier usuario.

Los comandos utilizados para la instalación se pueden ver en el Anexo 4.

3.7. Resumen

La instalación del middleware Ginga que se encuentra detallada en el Anexo 4, es un proceso que se puede seguir para los sistemas operativos Raspbian Jessie, así como también Ubuntu, la diferencia básicamente es en la descarga de la imagen, ya que la instalación de la imagen en la tarjeta SD card se puede escoger dependiendo del sistema operativo de la computadora que se esté utilizando como se detalla en el Anexo 2.

CAPÍTULO 4 : PRUEBAS Y RESULTADOS

Después de haber pasado por una selección de hardware y software para determinar las características apropiadas para el funcionamiento de Ginga, en este capítulo se realizó un análisis de los posibles problemas que se pueden presentar en cuanto a la ejecución de aplicaciones.

Para realizar las pruebas se utilizaron tres programas, de los cuales dos fueron realizados por la autora y uno fue facilitado por un colaborador, además se encuentran en el Anexo 6.

4.1. Ejecución de los programas

Para ejecutar los programas se utiliza la siguiente sentencia:

```
ginga --ncl main.ncl --vmode 800x600
```

Aparece una ventana donde se reproduce la aplicación, se debe activar la ventana del terminal donde se está ejecutando los comandos, seguido de la activación de la ventana de Ginga, finalizando por la revisión del volumen bajando a cero el volumen y después subiendo al nivel máximo del audio de la Raspberry, a fin de que se reconozca las teclas que se pulsa para observar la interactividad.

El audio de las aplicaciones que se ejecutan en Ubuntu Mate se escucha de manera óptima, en cambio las aplicaciones que se ejecutan en Raspbian Jessie se escuchan con un nivel alto de ruido, además de generar un retardo en la reproducción del sonido.

4.1.1. Programa 1

Este programa tiene como temática “Las Islas Galápagos”, al empezar a ejecutar el programa da inicio un video con una imagen, después de 15 segundos aparecen unos cuadros de colores: amarillo, azul, verde y rojo, que son botones de interactividad que se activa por medio del teclado, simulando los botones del control del televisor y por medio de los cuales podemos tener más información acerca del mismo tema.

4.1.2. Programa 2

Este programa tiene como temática “Ecuador”, al empezar a ejecutar el programa da inicio un video con un mensaje de presionar ok para empezar la interactividad, el ok hace referencia al enter del teclado, al presionar esta tecla se redimensiona la pantalla del video y aparecen los nombres de cuatro ciudades del país, cada una de ella con un color: rojo para Quito, verde para Guayaquil, amarillo para Cuenca y azul para Manta. Los colores hacen referencia a los botones del control remoto del televisor, activándose para este programa con el teclado, y dando a conocer una reseña de cada una de las ciudades.

4.1.3. Programa 3

Este programa tiene como temática “Leyendas Quiteñas”, al empezar a ejecutar el programa da inicio un video de Quito, después de 15 segundos dará inicio los botones de la interactividad amarillo, azul, rojo y verde, Los colores hacen referencia a los botones del control, activándose para este programa con el teclado, y dando a conocer por cada botón una leyenda de la ciudad de Quito.

4.2. Pruebas con la Raspberry Pi Modelo B

4.2.1. Raspbian Jessie

Al ejecutar el programa 1, se observó que el video se pixelaba, simulaba como una secuencia de imágenes. Además existió un retraso en salir los botones completos correspondientes a las actividades interactivas, a pesar que se encontraban programados para salir juntos, se puede ver que en la imagen 1 se tiene dos botones, completándose los botones en la imagen 2 luego de unos segundos. Finalmente el video colapsa en ciertas ocasiones quedando totalmente en negro la pantalla, no siempre sucede en la misma parte del video.

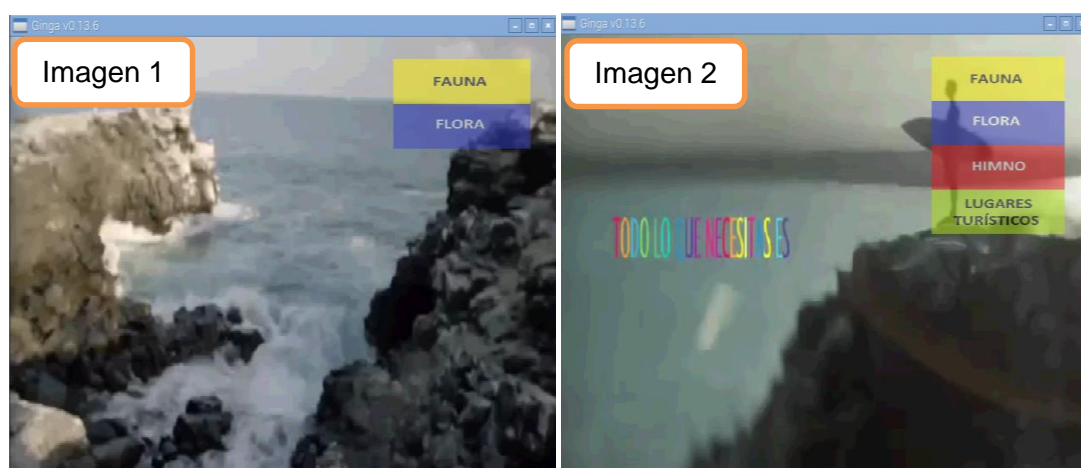


Figura 21. Prueba 1 de aplicación en Raspberry Modelo B

4.3. Pruebas con la Raspberry Pi 2

Las pruebas que se realizaron en la Raspberry Pi 2 fueron en base a dos sistemas operativos como fueron:

- Ubuntu
- Raspbian jessie

4.3.1. Ubuntu Mate

Se realizó pruebas con el Programa 1, donde la aplicación se ejecutó sin ningún problema, el video tuvo una resolución buena, sin pixelar la imagen, sin embargo en ocasiones aparecieron unas líneas, debido a la resolución de la pantalla. Los botones de la interactividad aparecieron juntos dispuesto a los programado.

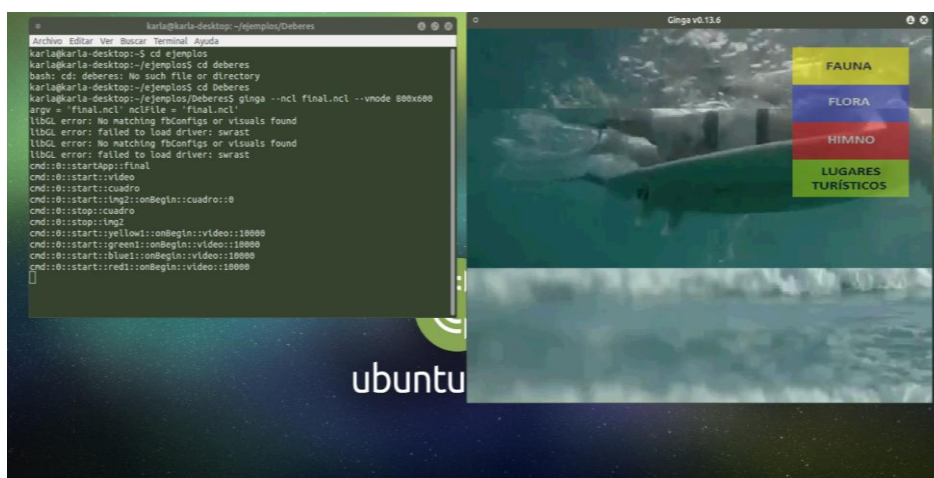


Figura 22. Aplicación corriendo sobre Ubuntu

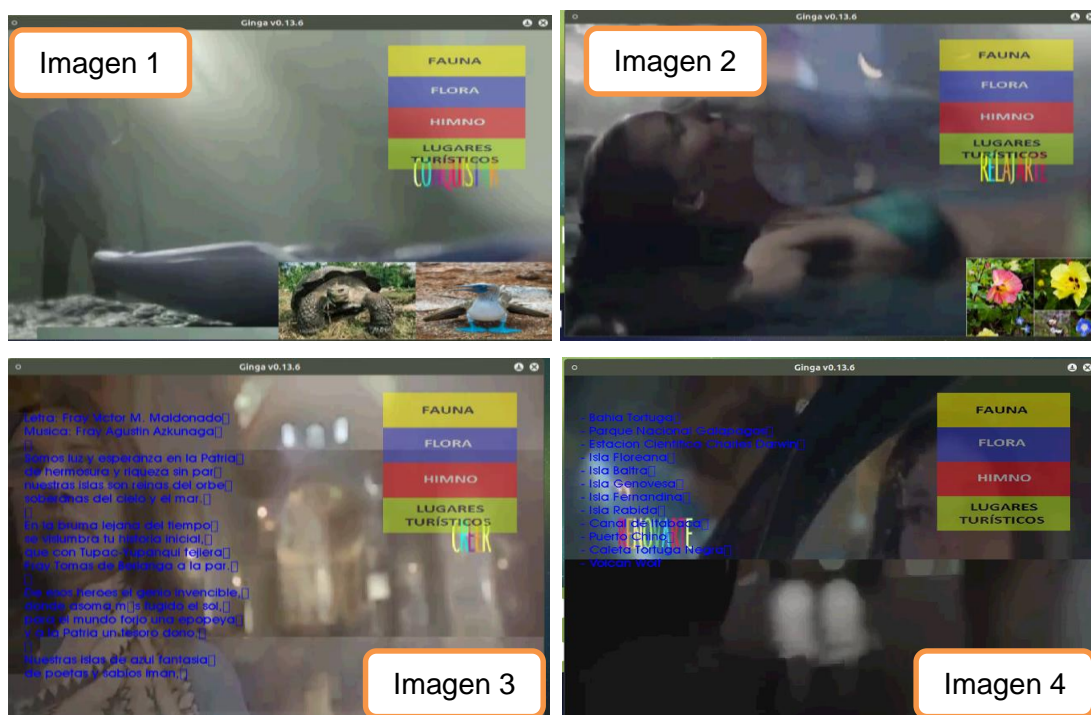


Figura 23. Interactividad Programa 1

La interactividad de la aplicación está plasmada en la Figura 23, se puede observar que en la imagen 1 al presionar el botón amarillo, se da a conocer fotos de los animales típicos de las Islas. En la imagen 2 al presionar el botón azul, se puede conocer la flora. En la imagen 3 al presionar el botón rojo, se puede visualizar el Himno de la región y en la imagen 4 al presionar el botón verde, se visualiza los lugares turísticos. La interactividad no muestra inconvenientes al momento de ser utilizada, rápidamente cambia según sea la necesidad de escoger el ícono deseado.

4.3.2. Raspbian Jessie

La prueba se realizó con el programa 1, al ejecutar el programa funcionó de manera correcta, el video avanzó con imágenes poco pixeladas, apareciendo los botones de la interactividad según lo programado, al igual que al presionar las teclas referentes a la interactividad mostró las imágenes y los textos correspondientes.

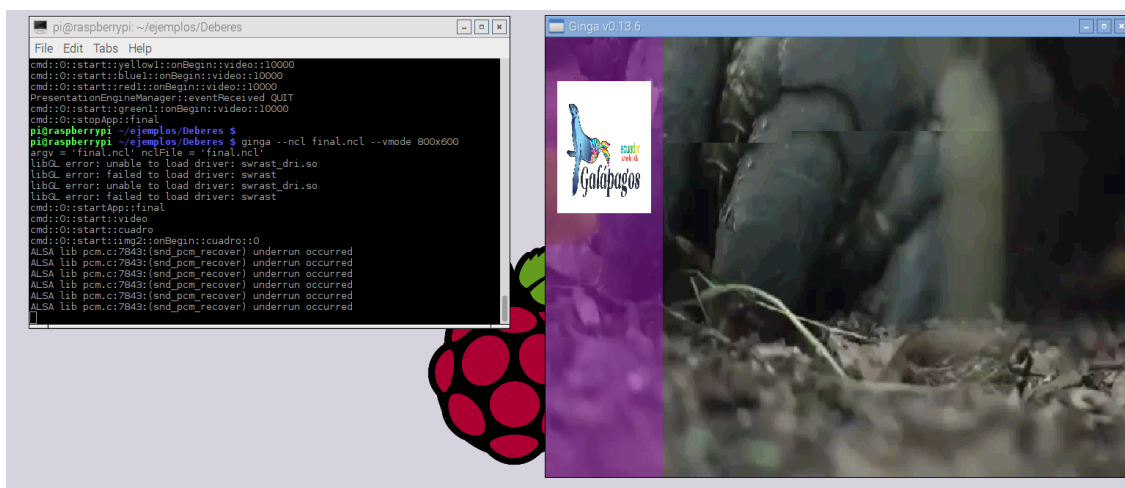


Figura 24. Aplicación corriendo sobre Raspbian Jessie

4.4. Medición de rendimiento

Para evaluar el rendimiento que posee cada una de las tarjetas junto a los sistemas operativos, se procedió a realizar la toma del tiempo en segundos (s) de lo que se demora el Programa 1 en arrancar la aplicación hasta que aparezcan los botones referentes a la interactividad por completo.

Las pruebas se realizaron utilizando escritorio remoto, así como también con hardware externo por medio de un cable HDMI para utilizar la pantalla, además de teclado y mouse.

Los datos obtenidos de la prueba que se realizó utilizando escritorio remoto se muestran en la Tabla 6.

Tabla 6. Medición de Rendimiento con Escritorio Remoto

Modelo de Raspberry	Raspberry Pi Modelo B	Raspberry Pi 2	Raspberry Pi 2
Sistema Operativo	Raspbian Jessie	Ubuntu Mate	Raspbian Jessie
Medida 1 (s)	23,70	13,76	15,55
Medida 2 (s)	28,90	13,90	14,36
Medida 3 (s)	26,11	13,58	15,55
Medida 4 (s)	24,56	13,25	14,25
Medida 5 (s)	22,71	13,86	15,03
Medida 6 (s)	26,45	13,71	15,23
PROMEDIO (s)	25,41	13,68	15,00

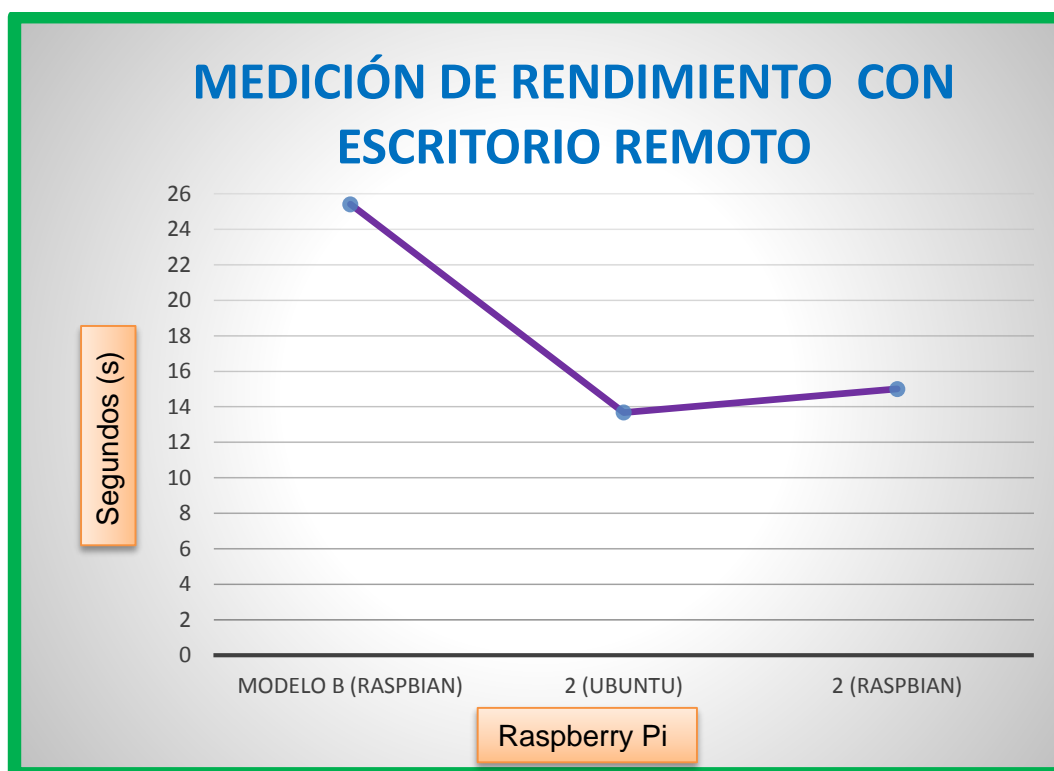


Figura 25. Estadísticas de medición de rendimiento con escritorio remoto

Los datos obtenidos de la prueba que se realizó utilizando hardware externo se muestran en la Tabla 7.

Tabla 7. Medición de Rendimiento con Hardware Externo

Modelo de Raspberry	Raspberry Pi Modelo B	Raspberry Pi 2	Raspberry Pi 2
Sistema Operativo	Raspbian Jessie	Ubuntu Mate	Raspbian Jessie
Medida 1 (s)	19,68	12,20	14,70
Medida 2 (s)	22,20	13,55	14,65
Medida 3 (s)	20,80	13,70	14,80
Medida 4 (s)	24,63	13,71	14,21
Medida 5 (s)	23,28	13,30	14,53
Medida 6 (s)	22,13	13,45	14,75
PROMEDIO (s)	22,12	13,32	14,61

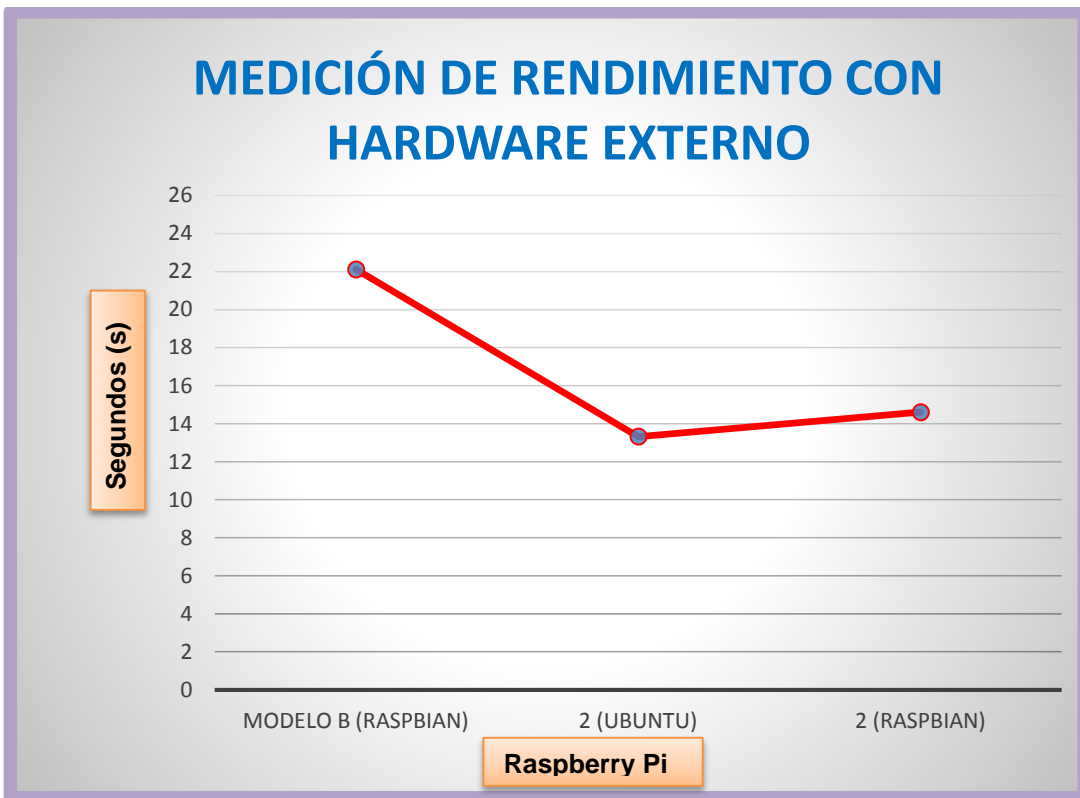


Figura 26. Medición de rendimiento con hardware externo

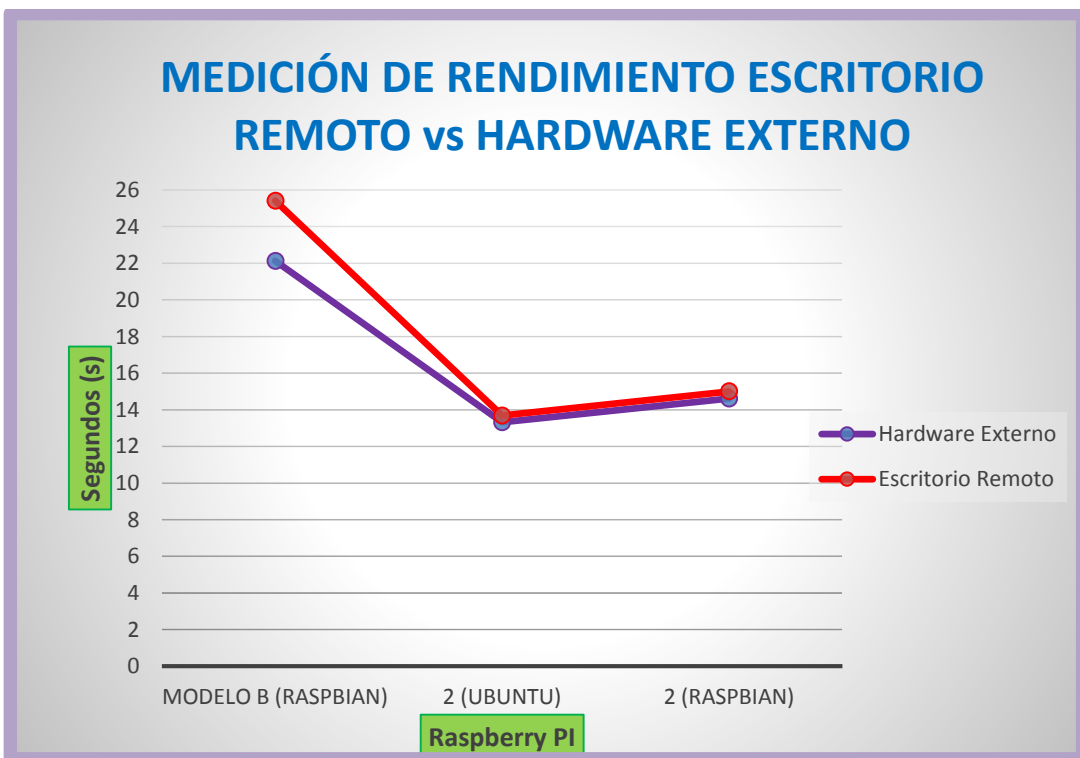


Figura 27. Rendimiento Escritorio Remoto vs. Hardware Externo

Al realizar la toma de tiempos se pudo comprobar que al ejecutar las aplicaciones utilizando escritorio remoto, tienen una demora aproximadamente de 0,5 segundos en la Raspberry Pi 2 y de aproximadamente 3 segundos en la Raspberry Pi Modelo B, más que al ejecutar las aplicaciones utilizando hardware externo. Además se observa que el menor tiempo en ejecutarse una aplicación se da en el sistema operativo Ubuntu montado en la Raspberry Pi 2.

El comportamiento de la Raspberry Pi 2 fue más rápido en cuanto a la ejecución de los programas, debido que a que obtuvo menores tiempos, en cambio la Raspberry Pi Modelo B obtuvo mayores tiempos en la ejecución de los programas debido al procesador que contiene la tarjeta.

4.5. Medición de calentamiento

0	Nada	Temperatura Ambiente + 10°C aprox.
1	Poco	Temperatura Ambiente + 15°C aprox.
2	Mucho	Temperatura Ambiente + 20°C aprox.

Tabla 8. Medición de Calentamiento

Programas	Raspberry Modelo B	Raspberry Pi 2	Raspberry Pi 2
Sistema Operativo	Raspbian Jessie	Ubuntu Mate	Raspbian Jessie
Programa 1	1	0	0
Programa 2	1	0	1
Programa 3	2	1	1

El calentamiento producido en la tarjeta se da por el procesamiento al ejecutar los programas, según la tabla 8 se puede observar que el programa 1 es el que genera menos calor en la Raspberry, en cambio el programa 3 genera más calor por poseer una mayor cantidad de videos. Con respecto a la Raspberry que produce más calor es el modelo B, por sus características y el procesador que trabaja con una memoria RAM de 512 Mb a diferencia de la Raspberry Pi 2 que es la más actual en el mercado y posee una memoria RAM de 1 Gb.

El consumo medio de la placa Raspberry Pi es de 3,5 W, es una cantidad mínima en comparación con un computador de escritorio o un portátil, dando como una ventaja en servidores de datos, web, domótica, central telefónica o servidor de impresión; sin embargo para trabajar como un decodificador Ginga, el consumo de energía será mucho mayor, ya que el calentamiento subirá a parámetros altos, por ello no se puede tener conectado varias horas a fin de no dañar los dispositivos o la placa en su totalidad.

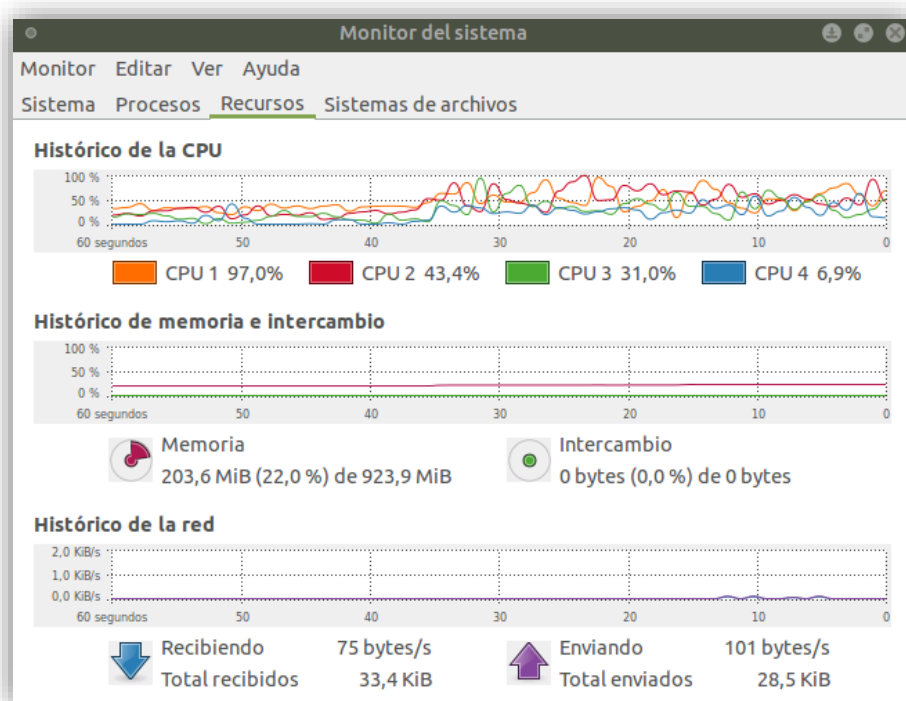


Figura 28. Nivel de procesamiento

En la figura 28 se puede observar el nivel de procesamiento de los CPU's, al ejecutar una aplicación se comprueba un incremento de manera acelerada, en varias ocasiones los CPU's se colocan al 100%, ya que la reproducción de imágenes y videos le colocan a la tarjeta en los niveles más altos de procesamiento, esta es una las causas por las cuales la Raspberry pasa por etapas de calentamiento, ya que conmutan las señales produciendo un consumo de potencia mayor del nivel promedio.

CAPÍTULO 5 : CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Se evaluó las prestaciones y rendimiento de un decodificador de contenidos interactivos basados en GINGA-NCL sobre una Raspberry Pi, efectuando las pruebas en dos modelos de la Raspberry Pi para establecer una comparación en cuanto hardware, y determinar la usabilidad de la misma.
- Se realizó un análisis del comportamiento de la minicomputadora Raspberry Pi, en cuanto a reproducción de videos, imágenes y audios, dando un resultado aceptable en la tarjeta modelo 2, a diferencia de la tarjeta modelo B que tiene un comportamiento pausado en la reproducción del video, debido a que posee un procesador con ARM6, a diferencia del modelo actual posee un procesador con ARM7.
- Las pruebas en las tarjetas se han realizado con tres aplicaciones programables de Ginga NCL, que presentan interfaces con interactividad, a la cual se accede por medio del teclado, simulando el control remoto del televisor.
- Se describió el procedimiento para embeber Ginga en la Raspberry Pi, detallado los comandos y los pasos que se debe seguir en dicha instalación.
- Los comandos encontrados para la instalación son propios del middleware, ya que si se realiza la instalación de las librerías de la manera convencional no tendrá el correcto funcionamiento de Ginga y dará como resultado una serie de errores.
- El rendimiento del funcionamiento de Ginga en la tarjeta, se comprobó mediante las mediciones de tiempo que se realizaron en el programa 1

hasta un punto de la aplicación, además se realizó en dos escenarios: el primero fue mediante escritorio remoto y el otro con hardware externo, obteniendo en las dos pruebas que la tarjeta que ejecuta en menor tiempo las aplicaciones es la Raspberry Pi 2 montada el sistema Operativo Ubuntu.

- Se comprobó que al ejecutar las aplicaciones en la Raspberry, daba un calentamiento debido al procesamiento y al contenido de los programas, por ello se daba un incremento acelerado en los CPU's que conforman el procesador de la tarjeta, determinando que la tarjeta que presento más calentamiento fue la Raspberry Pi Modelo B.
- El middleware Ginga, ha sido desarrollado por Argentina y Brasil, donde cada uno de ellos presenta una versión del programa con diferentes requisitos. Por ello se buscó el mejor acoplamiento con la tarjeta y se determinó que Ginga Brasil requería el sistema operativo Debian el cual equivalía a Raspbian y se podía obtener la imagen del página oficial; en cambio Ginga Argentina requería el sistema operativo Ubuntu pero en versión 12.04, la cual no se encuentra disponible para Raspberry y se debería crear la imagen.
- Se determinó que la Raspberry Pi Modelo B no tiene un buen funcionamiento embebido Ginga, ya que la reproducción de las aplicaciones se da de manera lenta y no permite que se ocupe alguna función de sistema operativo adicional, esto se debe a que posee un procesador con una memoria RAM de 512Mb a diferencia de la tarjeta actual posee un procesador con memoria RAM de 1Gb.
- En la Raspberry Pi 2 se pudo montar dos sistemas operativos que fueron Raspbian Jessie y Ubuntu Mate, donde se comprobó que el desempeño en cuanto a tiempo de instalación y ejecución de las aplicaciones dio mejor resultado en Ubuntu Mate, con una diferencia no muy extensa de Raspbian Jessie.

5.2. Recomendaciones

- Se recomienda utilizar una tarjeta SD de 8 Gb como mínimo y de clase 6 como la más baja, para la instalación del sistema operativo de la Raspberry Pi obteniendo un correcto funcionamiento, sin tener problemas en cuanto a la usabilidad del sistema operativo.
- Se recomienda extender la tarjeta SD en su totalidad, debido que después de escribir la imagen particiona la tarjeta, dejando inutilizable gran parte de la misma, y al realizar la instalación del middleware no se tendrá espacio para los paquetes requeridos.
- Para la selección de Ginga se recomienda analizar los requisitos que posee cada uno de los paquetes, y la afinidad que posean con la Raspberry, ya que no todas las librerías son compatibles con Ginga, debido al código fuente que compone el middleware.
- Si se requiere realizar la instalación de Ginga en la Raspberry se recomienda seguir los pasos escritos en el Anexo 3, ya que posee comandos que se utilizan solo para la compilación de Ginga, caso contrario no funcionará de manera correcta el middleware.
- Es necesario que se alimente con un cable que de 5V y 2,5mA a la Raspberry Pi ya que si no recibe la corriente necesaria puede presentar problemas como: la tarjeta se pueda bloquear, no permita el arranque del sistema operativo, etc.
- Se recomienda analizar el funcionamiento del sistema operativo con relación a los planos en que se ejecutan las aplicaciones, a fin de no tener que activar y desactivar las ventanas en el momento de la ejecución de la aplicación y la configuración del audio.

- Como trabajo futuro se podría realizar el estudio de la implementación de un circuito de recepción que se adhiriera a la Raspberry Pi, por medio de módulos de recepción bajo el estándar ISDB-Tb, para enviar un Ts y de esta manera comprobar la parte de recepción, acoplándole al decodificador de contenidos interactivos basados en Ginga NCL, y comprobando la llegada del Ts y la reproducción del mismo, obteniendo así un prototipo de set top box completo a bajo costo.

REFERENCIAS

- Alvarez, A., Balanguer, F., Costa, F., & Woites, L. (s.f.). *Aplicaciones Casuales de Televisión Digital con Ginga NCL/Lua*. Universidad Nacional de La Plata.
- Cabezas., G., & Quezada, M. (2012). Diseño e implementación de un prototipo para un sistema de generación de aplicaciones interactivas con GINGA-NCL para la evaluación de servicios masivos.
- Debian. (2016). *Debian Jessie*. Obtenido de <https://www.debian.org/News/2015/20150426>
- Espinola, R. (2014). *Hardware Libre, Recopilación de Información*.
- FFMPEG. (2015). *Acerca de FFMPEG*. Obtenido de <https://www.ffmpeg.org/about.html>
- Fundación Raspberry. (2015). *Raspberry Pi*. Obtenido de <https://www.raspberrypi.org/documentation/installation/>
- Galabay, P., & Vivar, F. (2012). *Manejo de software Ginga para el desarrollo de aplicaciones interactivas para televisión digital, basado en el estándar Brasileño ISDB-Tb*. Cuenca: Universidad Politécnica Salesiana .
- Gomes, L. (2016). *Ginga NCL*. Obtenido de <http://www.ginga.org.br/es/inicio>
- González, I., González, J., & Gómez, F. (2003). *Hardware libre: clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux*.
- Lantinga Sam, E. M. (2015). *SDL_image2.0*. Obtenido de https://www.libsdl.org/projects/SDL_image/
- Lifia. (2013). *Documentación Ginga.ar*. Obtenido de <http://www.lifia.com>
- Medina, J., & Villa, C. (2014). *Implementación de un laboratorio de televisión digital que cumpla el estándar ISDB-TB*.
- PE, I. (26 de 05 de 2015). *comohacer.eu*. Obtenido de <http://comohacer.eu/gpio-raspberry-pi/>
- Piscotta, N. (2010). *Sistema ISDBT-b (Primera parte)*.
- Rayo, & Chimbo. (2013). Utilización de la Minicomputadora Raspberry Pi para la adquisición y evaluación de datos de consumo de energía eléctrica de equipos a 220 Voltios.

- Rayo., J., Chimbo, J., & Valdivieso, C. (2013). Utilización de la minicomputadora Raspberry PI para la adquisición y evaluación de datos de consumo de energía eléctrica de Equipos a 220 voltios. *Revista Tecnológica ESPOL*.
- Reporte Técnico de ARIB*. (20 de 09 de 2015). (Reporte Técnico) Recuperado el 10 de Julio de 2015, de http://www.dibeg.org/techp/feature/isdb-t_Spanish.pdf
- Rojas, P. (2012). La Televisión Digital Terrestre y su incidencia en las estaciones televisivas ecuatorianas. *Advicom Cia. Ltda.*
- Sam., L. (2015). *SDL_ttf2.0*. Obtenido de https://www.libsdl.org/projects/SDL_ttf/
- Simple Direct Layer. (2015). *SDL*. Obtenido de <https://www.libsdl.org/index.php>
- The Apache XML Project. (2015). *Xerces ++ XML Parser*. Obtenido de <https://xerces.apache.org/xerces-c/index.html>
- Torres, J. (28 de 06 de 2011). *Comunidad Ginga Ecuador*. Obtenido de <http://comunidadgingaec.blogspot.com/2011/06/middleware-ginga.html>
- Ubuntu. (2016). *Ubuntu Mate*. Obtenido de <https://ubuntu-mate.org/>
- Villanueva, J., & Velásquez, G. (2010). *Informe preliminar: Estado del Arte de Receptores Set-Top-Box-Aplicaciones*.
- Wikipedia. (2015). *Curl*. Obtenido de <https://es.wikipedia.org/wiki/CURL>
- Zambrano., A. (2015). *Introducción a la TV Digital Interactiva y Ginga.ar*. Lifia.