

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**DISEÑO DE UN SISTEMA INTERACTIVO PARA LA
TELEVISIÓN DIGITAL TERRESTRE USANDO
MIDDLEWARE GINGA**

**INFORME DE SUFICIENCIA
PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO ELECTRÓNICO**

**PRESENTADO POR:
RONALD RICARDO MARTINEZ CHUNGA**

**PROMOCIÓN
2007-II**

**LIMA-PERÚ
2013**

**DISEÑO DE UN SISTEMA INTERACTIVO PARA LA TELEVISIÓN
DIGITAL TERRESTRE USANDO MIDDLEWARE GINGA**

Dedico este trabajo a Dios, a mi Alma Mater, la Universidad Nacional de Ingeniería, a mi preciada Facultad de Ingeniería Eléctrica y Electrónica, a mi madre Ana María Chunga que ha sido promotora incansable de este trabajo, y a mi bella Maribel Ruiz, inspiración de un logro más en mi trayectoria de vida, muchas gracias.

SUMARIO

Estamos inmersos en un proceso de cambio de la televisión analógica a la digital, que será definitivo con el inminente apagón analógico. Son muchas las ventajas que aporta la Televisión Digital, como mayor calidad de imagen, mayor número de canales, y entre ellas, destaca la interactividad, una opción aún desconocida por la mayoría de la sociedad.

La interactividad, es un mundo en desarrollo y sus posibilidades son ilimitadas, además, que supone un cambio en el papel del telespectador frente a la televisión, pasando a desempeñar el usuario un rol mucho más activo.

Este trabajo pretende dar a conocer cómo funciona el entramado de la Televisión Digital Interactiva, revisando las capas básicas hasta las aplicaciones del usuario final. El objetivo es aportar información sobre esta tecnología bastante desconocida, pero cercana a todos y con un futuro prometedor, y así proporcionar una visión más global de la televisión del futuro, la televisión digital interactiva.

Para lograr este objetivo se propone un diseño de una aplicación interactiva que será implementada íntegramente con Software Libre, el desarrollo empieza usando el Sistema operativo Ubuntu y las librerías del lenguaje NCL para implementarlo, la primera prueba de su funcionamiento se realizará con emuladores también basados en sistemas Linux, y finalmente, colocaremos este aplicativo en el sistema embebido del set-top box EITV DeveloperBox, una caja de desarrollo de producción brasileña, que nos permitirá comprobar la interactividad de la aplicación propuesta sobre la TDT en el Perú.

El aplicativo interactivo final desplegado en este trabajo, responde a un diseño básico para demostrar las funcionalidades de interactividad de las que se pueden disponer en una extensa gama de aplicaciones interactivas, como el hacer uso de las capacidades de redimensionamiento de la pantalla, la exhibición y manipulación de textos, la manipulación del formato de una caja contextual a través de la selección de botones vía el control remoto en pantalla, también se observará una imagen recibida vía el canal de retorno. Todo esto resume una gran cantidad de aplicativos interactivos que se quieren ejemplificar en el alcance de este trabajo.

ÍNDICE

INTRODUCCIÓN	1
Capítulo I	
PLANTEAMIENTO DEL PROBLEMA	2
1.1 Descripción del problema	2
1.2 Objetivos generales	4
1.3 Objetivos específicos	4
1.4 Evaluación del problema.....	5
1.5 Alcance del trabajo.....	5
1.6 Síntesis de trabajo	6
Capítulo II	
MARCO TEÓRICO	7
2.1 Definición.....	7
2.2 Plataformas de envío	7
2.3 El espectro radioeléctrico	8
2.4 Ventajas.....	9
2.4.1 Mejor transmisión	9
2.4.2 Más canales de TV	9
2.4.3 Más calidad de video.....	10
2.4.4 Más calidad de audio.....	12
2.4.5 Más contenidos e interactividad.....	13
2.4.6 Movilidad y portabilidad.....	13
2.5 Estándar de Televisión Digital Terrestre.....	14
2.6 Que se necesita para acceder a la TV Digital.....	15
Capítulo III	
PLATAFORMA DE DESARROLLO	16
3.1 El Middleware Ginga	16
3.2 Ambiente declarativo y ambiente imperativo	16
3.3 NCL como lenguaje declarativo de Ginga.....	18

3.4	NCM el modelo conceptual de datos del lenguaje NCL.....	19
3.5	Edición Declarativa con NCL	23
3.5.1	El que tocar	23
3.5.2	Donde tocar	23
3.5.3	Cómo tocar.....	24
3.5.4	Cuándo tocar	24
3.6	Estructura de un programa en NCL	24
3.7	Herramientas de Desarrollo para el NCL.....	25
3.7.1	NCL Composer	26
3.7.2	La máquina virtual Ginga-NCL set-top box	32
3.7.3	VMware player.....	32
3.8	Upload de un aplicativo NCL a la máquina virtual Ginga-NCL	37
3.9	Ejecución del aplicativo NCL en la máquina virtual Ginga-NCL	39
Capítulo IV		
DESPLIEGUE DE LA APLICACIÓN DE TDT INTERACTIVA.....		41
4.1	Descripción general del Equipo EITV DeveloperBox.....	41
4.2	Opción de Interactividad Ginga	44
4.3	Opción de Red.....	45
4.4	Servidor Web del EITV DeveloperBox	46
4.5	Aplicaciones de Red.....	47
4.6	Aplicaciones embebidas.....	47
4.7	Configuración de registro.....	49
4.8	Empaquetamiento del aplicativo para ser registrado en el set-top box.....	50
4.9	Implementación y ejecución del aplicativo interactivo.....	50
4.9.1	Descripción del aplicativo	50
4.9.2	Implementación del código fuente	52
4.10	Simulación en la máquina virtual.....	55
4.11	Estimación de costos para la implementación de un Laboratorio de TV Digital.....	57
CONCLUSIONES Y RECOMENDACIONES.....		58
Anexo A		
EITV DeveloperBox		60
Anexo B		
Glosario.....		63

Anexo C

Código fuente del aplicativo demoFinal.ncl.....	65
BIBLIOGRAFÍA.....	69

INTRODUCCIÓN

En este trabajo se presenta inicialmente una revisión sobre el panorama que actualmente tiene la Televisión Digital Terrestre, como un conjunto de tecnologías de transmisión, de recepción de imagen, sonido y datos, a través de señales digitales, además de la clasificación de los servicios digitales que se despliegan por este medio.

La Televisión Digital Interactiva, coloca al televidente como un usuario activo con la posibilidad de personalizar los servicios ofrecidos a través de este medio. La Televisión Digital interactiva posibilita nuevas oportunidades para el Estado y todas aquellas Instituciones que tengan participación en la Televisión, permitiendo desplegar todo tipo de servicios y aplicaciones en campos tan diversos como: la educación, la salud, la información, el comercio, la banca, entre otros.

Este trabajo describe un modelo para desarrollar una aplicación interactiva en T-Marketing, publicidad que es provista a través de un canal de Televisión Digital Terrestre, accesible mediante un set-top box que tiene la capacidad de acceder a aplicaciones interactivas, como se verá en su momento, a través del middleware Ginga.

Dentro de los principales objetivos de este trabajo está la descripción de los elementos participantes mínimos y necesarios para el desarrollo de aplicaciones interactivas usando software libre, en diferentes tipos de escenarios de recepción; en este caso, la Televisión Digital Terrestre (TDT) la cual permite movilidad y portabilidad; asimismo por ser una tecnología de difusión abierta, fomentar la inclusión social con acceso a la información. Aprovechando además, la tendencia decreciente en costo que implica la adquisición de los receptores digitales, en particular de los set-top box.

El aplicativo final responde a un diseño básico que tiene como propósito, demostrar las principales funcionalidades de interacción con el usuario, el cual podemos describir como un simple manipulador de formato de una caja de texto, que es aplicado con la selección de los botones disponibles en pantalla vía el control remoto. También se podrá observar una imagen proveniente del canal de retorno, lo que nos demuestra las bondades de comunicación con el exterior.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

Este capítulo tiene como finalidad analizar y desglosar el escenario bajo el cual este trabajo es motivado, revelando así su propósito. Se define el marco de trabajo, los alcances del mismo y una síntesis es detallada al término.

1.1 Descripción del problema

Una de las mayores y más importantes innovaciones que nos propone la TV Digital es la interactividad, es decir los servicios interactivos que ofrece, haciendo así al usuario un elemento activo que participa y no un simple espectador. Al ser la televisión digital interactiva una tecnología aun en desarrollo, algunos de estos servicios son aún ideas, aunque, a continuación vamos a exponer algunas de las posibilidades que surgen. Podemos clasificar los servicios interactivos según:

Servicios de Información: Son los servicios que proporcionan al usuario información de cualquier tipo (sin estar ésta relacionada con la programación que se emite en ese momento). Algunos ejemplos de estos servicios pueden ser información meteorológica, económica o bursátil, de tráfico, entre otros.

Servicios de Entretenimiento y Comerciales: Esta categoría engloba, por un lado, servicios orientados al ocio del usuario y, por otro, servicios para que las empresas capten la atención de éste. Podemos observar elementos que permitan la fidelización del usuario a determinados contenidos, como permitirle participar en concursos, votar en programas, consultar información sobre actores o concursantes de la programación que está viendo, entre otros. Aquí también entrarían la compra de eventos deportivos o películas en pay-per-view. En el segundo caso tenemos como ejemplo más claro la publicidad interactiva orientada al usuario según sus intereses. [1]

Servicios ligados a la programación: Estos proporcionan información complementaria sobre los contenidos audiovisuales que se están emitiendo en el momento actual. Entre estos servicios destacan la Guía Electrónica de Programación (EPG) avanzada, teletextos mejorados.

Servicios de T-Comercio: Son aquellos que permiten al espectador adquirir productos a través de su televisor (por ejemplo, si en un programa se hace una crítica de un libro que el espectador pueda comprarlo a través de Amazon).

Servicios de T-Administración: Permiten al usuario realizar trámites a través de la TV, como solicitar cita en servicios sanitarios, gestionar impuestos, solicitar impresos a administraciones públicas.

Servicios de T-Asistencia: Son servicios de asistencia a personas dependientes a través de sus receptores de TV (por ejemplo recordarles tomas de medicamentos o campañas médicas en determinados Centros de Salud).

En el mercado peruano la televisión digital está en una fase muy temprana de su desarrollo por lo que los servicios ofrecidos son muy pocos y bastante básicos, encuadrándose inicialmente en el apartado de servicios Informativos y de Marketing.

A fin a de disponer de estos contenidos interactivos proveídos por la TV Digital, es inherente el cambio de la tecnología tras ella, que permite alcanzar así la interactividad. Los elementos transmisores y receptores se reformulan dando paso al contenido digital, la Internet toma participación de esta revolución sirviendo como un canal de realimentación con el mundo exterior, donde ya no solo las televisoras y los espectadores participan, sino toda aquella entidad que encaja en los diferentes servicios de Interactividad que ya se han mencionado. La figura 1.1 muestra un bosquejo del escenario de TV Digital Terrestre planteado desde el punto de vista de los proveedores de soluciones interactivas, incorporando las tecnologías que se usan en la solución.

Desde el punto de vista de la recepción, o sea por el lado del usuario, quien posee el rol de receptor de TV Digital, observa la llegada de un nuevo elemento que inicialmente lo acompaña en el establecimiento de esta nueva era en la televisión, aquel que hace capaz de que nuestros aún televisores con receptores analógicos, puedan ser usados para la recepción de TV Digital, nos referimos por su puesto a los decodificadores y en particular a aquellos que permiten la interactividad, los set-top box.

Este novedoso dispositivo motiva la comprensión de su desempeño, tanto en componentes de hardware como de software, por un lado es un decodificador digital que es ya un componente conocido, dado que distribuidores de servicios de TV Digital los usan, por ejemplo por mencionar uno de ellos, la empresa Movistar TV; y por otro lado, tenemos al componente de Software denominado Middleware que permite presentar el contenido in-

teractivo en la pantalla. Esta conjunción de componentes tiene una denominación extendida la cual es set-top box.

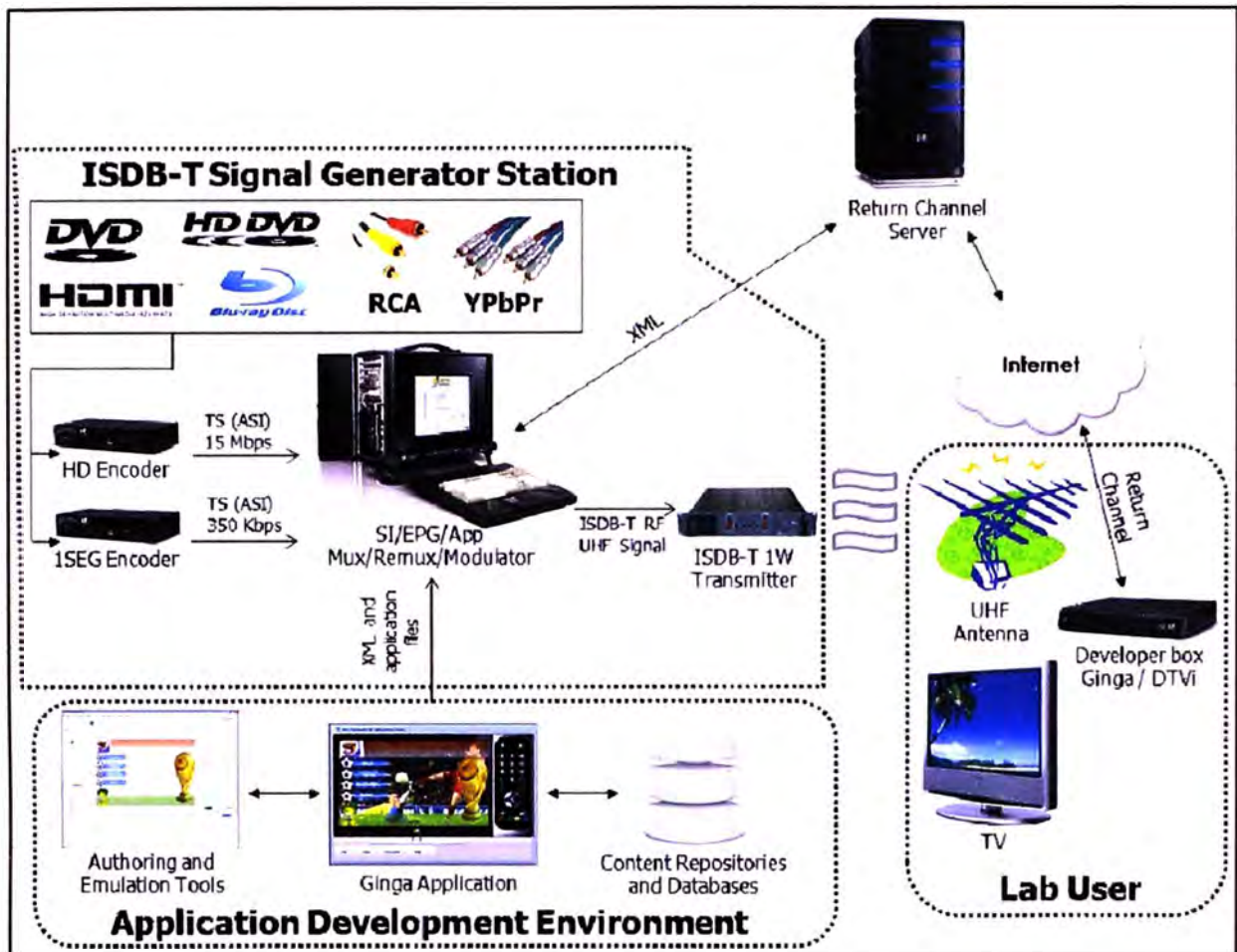


Figura 1.1 Sistema actual de Televisión Digital

Fuente: Entretenimiento e Interactividad para la TV Digital

Posteriormente, analizaremos el término Middleware, y más aún, revisaremos su funcionamiento básico, y las herramientas que facilita para la implementación de una aplicación interactiva colocada en un set-top box real.

1.2 Objetivos generales

Conocer y analizar desde el punto de vista del usuario – en la recepción – la infraestructura y la normatividad que la norma ISDB-Tb permite, a fin de crear aplicaciones interactivas en un tipo particular de Decodificador de TV Digital, comúnmente denominado set-top box.

1.3 Objetivos específicos

Revisar la arquitectura básica del sistema operativo embebido del set-top box, tal y como es en el equipo usado en este trabajo, con la participación del Middleware Ginga.

Sugerir una plataforma de desarrollo conveniente, suministrada por el middleware Ginga para el desarrollo de aplicaciones interactivas.

Diseñar una aplicación interactiva para TV Digital, usando herramientas de desarrollo actuales, utilizando además software libre, lo cual reduce el costo operativo en la implementación, permitiendo así, la integración de conocimientos en esta todavía reciente área de interés.

1.4 Evaluación del problema

La interactividad en la Televisión Digital Terrestre no solo puede ser observada como una novedad desde el punto de vista de la tecnología. también, coloca al usuario en una plataforma, donde su rol, juega un papel determinante en cómo se usa esta nueva forma de experimentar la televisión, facilitando así, la inclusión social a toda escala en nuestro País.

La interactividad además, interviene de modo transversal en diferentes áreas de la sociedad, desde la perspectiva del Estado. Los denominados aplicativos interactivos, revelan su participación en la sociedad desde su tipología, ya mencionadas según su campo de acción en la sociedad, vemos entonces como el T-Learning, el T-Voting, el T-Health, entre otros, llegan para quedarse en un mundo donde la Internet ya los ha presentado a la sociedad.

1.5 Alcance del trabajo

Este trabajo está enmarcado en la línea de desarrollo e investigación para la TDT sobre la base del estándar ISDB-Tb, adoptada por el Estado peruano. El trabajo pretende cubrir todos los aspectos útiles que deben ser considerados para un desarrollo óptimo y eficiente con herramientas disponibles para comenzar a implementar aplicativos para la TV Digital Terrestre en el Perú

La utilización de herramientas de Software libre en su totalidad para este trabajo, reduce la brecha económica que distancia a los nuevos desarrolladores de esta novedosa área de las Tecnologías de Información. Este trabajo pretende informar al desarrollador que su uso es sencillo y accesible, motivando así, la concepción de soluciones reflejadas en el buen diseño de estos aplicativos.

El trabajo no busca decir que esta es la única forma o la mejor forma de empezar a crear interactividad para la TV Digital, busca ser una guía introductoria a esta novedosa área de conocimientos que de paso a otros investigadores en el país para extender y mejorar lo que ya este trabajo pretende ilustrar.

1.6 Síntesis de trabajo

Se exponen aspectos teóricos que permiten la práctica recomendada para el mejor diseño de aplicaciones interactivas para la TV Digital.

Se revisa la estructura del middleware, a fin de poder discernir, cual es lo soporte que Ginga ofrece, y cómo podemos usarlo para comenzar a crear interactividad en la TV Digital Terrestre en el Perú.

Finalmente se expone la implementación de una aplicación interactiva demostrativa que logra ilustrar las bondades del set-top box adquirido para la sustentación de este trabajo, reflejando así, la veracidad de esta propuesta.

Este aplicativo ha sido diseñado para mostrar las bondades interactivas que en general los aplicativos de interactividad poseen, se trata de una caja contextual que es manipulada predeterminadamente a voluntad por el espectador. Esto lo realizará vía la selección de los botones disponibles en pantalla accesibles vía el control remoto, reflejando así la interactividad del proceso. También otras características como el redimensionamiento de pantalla, o la obtención de una imagen vía el canal de retorno, serán exhibidas como muestra de todo lo que una aplicación interactiva permite al momento de ser ejecutado.

CAPÍTULO II MARCO TEÓRICO

Este capítulo revisa el marco teórico en cuanto a la televisión digital se refiere, y en particular, a la situación actual de la televisión digital terrestre. Estos ítems son desenvueltos en forma básica sin llegar al detalle de los mismos. Se persigue conocer toda la parafernalia que envuelve a la TDT en el Perú, sus escenarios actuales, y cuál es el plan que el Estado viene implementando.

2.1 Definición

La televisión digital es un avance tecnológico que transmite los contenidos audiovisuales a través de una codificación digital. A diferencia de la señal analógica, la señal digital envía los sonidos e imágenes sin distorsión o ruidos, con mejor detalle y resolución, además de permitir que el televidente interactúe con el televisor (que puede estar en nuestra sala, en un bus en movimiento, o integrado en un teléfono celular).

El motivo de las ventajas de la tecnología digital es que los datos que se transmiten hacia nuestros televisores son recibidas a través de números discretos; es decir, una transmisión que no permite que el TV interprete de manera distinta la información que recibe, por lo que la información es recibida exactamente como fue emitida. La televisión analógica, en cambio, recibe la imagen y el sonido interpretando la señal según las variaciones en la amplitud y frecuencia de la onda, lo cual hace que la transmisión pueda ser inexacta. Así, mientras lo digital es una interpretación exacta, lo analógico es una interpretación aproximada sensible al ruido y distorsiones en su transmisión.

2.2 Plataformas de envío

La televisión digital puede ser transmitida bajo distintas plataformas, como a través de un satélite, de un cable, de la línea telefónica, de la señal abierta (espectro radioeléctrico), entre otros. Cuando la televisión digital se transmite a través de la señal abierta se le denomina Televisión Digital Terrestre.

Televisión Digital por Cable. La señal de televisión llega al usuario a través de un cable, ya sea el coaxial convencional, el cable telefónico (también conocido como IPTV) o

incluso por los cables de electricidad. Para ello, las empresas de TV instalan una caja decodificadora que recibe señales de cualquier vía (aérea o terrestre) y las distribuyen a los usuarios por medio de un cable.

Televisión Digital por Satélite. La señal llega al usuario desde un satélite de distribución directa. Las empresas más conocidas que prestan este servicio en el Perú son: DirecTV, Cable Mágico Satelital y Claro TV Sat. Estas instalan al usuario una antena parabólica y un receptor satelital conectado directamente a un televisor, que recibe el paquete de programación ofrecido. La señal por satélite no admite vías de retorno (que el televisor envíe información al canal de TV).

Televisión Digital Terrestre (TDT). El canal de televisión utiliza antenas ubicadas en torres, y utilizando el espectro electromagnético (el cual se utiliza actualmente en la televisión convencional o analógica), difunde la señal en un área determinada. La señal es recibida en cada hogar por medio de las antenas que están comúnmente integradas al televisor o que se encuentran en los techos de las casas. La TDT, por utilizar un canal de comunicación de propiedad pública, de uso masivo y en la actualidad de acceso gratuito, se le considera el medio más importante para la expansión de la televisión digital.

2.3 El espectro radioeléctrico

El espectro radioeléctrico (o electromagnético) es la distribución de todas las ondas electromagnéticas que se encuentran segmentadas según su frecuencia. Así, existen desde las frecuencias más bajas (muy baja frecuencia - radio) hasta las más altas (rayos gamma), pasando por las ondas microondas, infrarrojo, ultravioleta y por la frecuencia que corresponde a las ondas de radio y televisión.

Las ondas de televisión se encuentran divididas en las ondas VHF (Very High Frequency) y UHF (Ultra High Frequency). Según las disposiciones del Ministerio de Transportes y Comunicaciones del Perú (MTC) y las recomendaciones del Comité Consultivo Internacional de Radiocomunicaciones (CCIR), el rango de frecuencias VHF que se utilizan para los servicios de televisión son: 54 - 72 MHz (canales 2 - 4), 76 - 88 MHz (canales 5 y 6) y 174 - 216 MHz (canales 7 - 13), mientras que en el caso de la banda UHF, la reciente implementación de la Televisión Digital Terrestre ha establecido el rango de frecuencias entre los 470 y 698 MHz (canales 14 - 51). El ancho de banda que utiliza un canal de TV analógica en el Perú es de 6 MHz.

El espectro radioeléctrico es un recurso natural limitado que forma parte del patrimonio de la Nación y por tanto es de propiedad de todos los peruanos. Es administrado y re-

gulado, según la Ley 28278 - Ley de Radio y Televisión, por el Ministerio de Transportes y Comunicaciones, quien entrega los espacios o bandas por concesión a los canales de televisión y emisoras de radio.

2.4 Ventajas

2.4.1 Mejor transmisión

Al codificarse digitalmente, la televisión digital terrestre transmite los contenidos casi sin ninguna interferencia o distorsión de las imágenes y el sonido. La televisión analógica, por diversos factores, puede generar una distorsión en las imágenes que se traducen en el efecto lluvia, la duplicidad de imagen (fantasma) o cambios en la nitidez, contraste y color de las imágenes, tal y como lo muestra la figura 2.1.

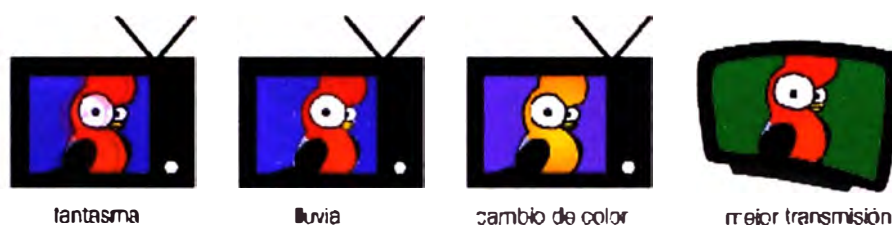


Figura 2.1 Mejor transmisión

Fuente: Consejo Consultivo de Radio y Televisión

La televisión digital mejora la capacidad de transmisión de la señal. Sin embargo, si la señal digital llega a nuestro televisor de manera parcial, no se verá ninguna imagen, a diferencia de la analógica que recibe la imagen con calidad media. Esto se debe a que la señal digital necesita de la recepción total de toda la información, en caso contrario, se verá una pantalla negra.

2.4.2 Más canales de TV

El espectro radioeléctrico utilizado para prestar servicios de televisión cuenta con un total de 300 MHz (50 canales) entre las bandas VHF y UHF. En el Perú, cada canal de televisión analógico utiliza 6 MHz del espectro radioeléctrico para poder transmitir sus contenidos. Con la digitalización de la televisión (tomando en consideración los factores técnicos del estándar ISDB-Tb elegido por el Perú), este intervalo de frecuencia puede albergar un canal en Full HDTV (1920 x 1080 pixeles), un canal HDTV (1280 x 720 pixeles) más un canal SDTV (720 x 480 pixeles), o hasta 4 canales en SDTV. Los tres tipos de calidad de televisión digital son superiores a la calidad de la televisión analógica. El estándar de compresión que se utiliza es MPEG-4 AVC (H.264) y compresión de audio HE-AAC, esto está descrito en la figura 2.2.

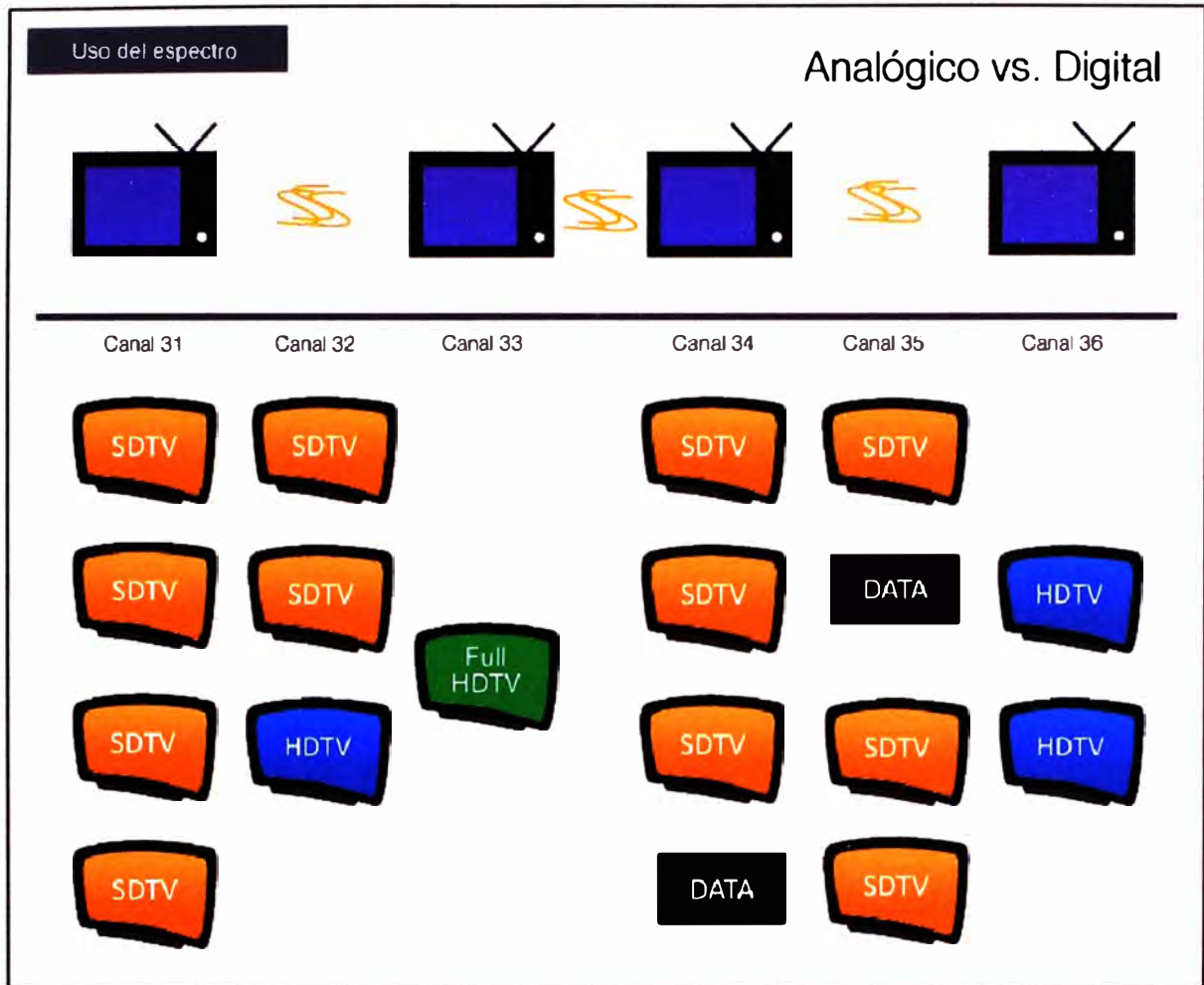


Figura 2.2 Más canales de TV

Fuente: Consejo Consultivo de Radio y Televisión

Es importante comentar que la televisión digital genera menor ruido y su transmisión es más efectiva, por lo que los canales de televisión que actualmente sirven de espacio para evitar interferencias entre los canales vecinos (por ejemplo, los canales 24, 26, 28 entre otros, según la canalización de Lima), podrán ser utilizados, lo que aumenta aun más la cantidad de canales de televisión digital disponibles.

Bajo este precepto, en un escenario teórico podría haber, en la banda UHF, hasta 38 canales de televisión digital en Full HD, o hasta 152 canales en SDTV, tomando en consideración que se darán por concesión 6MHz para cada radiodifusor como se viene dando con la televisión analógica en el Perú.

2.4.3 Más calidad de video

La resolución de la televisión analógica en el Perú está determinada por el sistema norteamericano NTSC que fue adoptado por el Perú en los años 60. Este sistema utiliza 480 líneas de barrido (resolución equivalente en digital a 640 x 480 pixeles). En la televisión digital, por otro lado, la resolución puede ser personalizada, siendo las usadas:

- **Standard Definition Television (SDTV):** cuenta con una resolución de 720 x 480 píxeles.
- **High Definition Television (HDTV) o HDTV 720p:** que cuenta con una resolución de 1280 x 720 píxeles.
- **Full HDTV o HDTV 1080p** que cuenta con 1920 x 1080 píxeles.

La posibilidad de disfrutar contenidos de alta definición depende del tamaño y resolución de los televisores digitales (LCD, plasma o LED). Para televisores entre 17 y 22 pulgadas, la diferencia de ver contenidos HDTV 720p y HDTV 1080p no es muy notoria. Sin embargo, para televisores de 32 pulgadas o más, la diferencias se hace más evidente. En el caso de televisores análogos, bastaría con la resolución estándar SDTV.

Es importante considerar que existen televisores que pese a contar con más de 32 pulgadas, no cuentan con la resolución necesaria para disfrutar de la calidad Full HDTV y por tal motivo el precio del aparato es menor.

Bajo esta premisa, si contamos con un televisor analógico, se obtendrán imágenes de calidad similar tanto con señal analógica, SDTV, HDTV 720p o HDTV 1080p, ya que el límite tecnológico se encuentra impuesto por el televisor.

En caso que tengamos un televisor con resolución HDTV 720p, podríamos diferenciar la calidad de señal analógica con SDTV y HDTV 720p, mas la señal 1080p se vería igual que la de 720p.

En el caso de un televisor con resolución HDTV 1080p, todas las calidades de señal se verán diferenciadamente de acuerdo a su calidad.

Otro aspecto a tomar en cuenta es que la televisión digital ha estandarizado el ratio de aspecto 16:9 (panorámico) sobre el 4:3 (convencional), lo que promueve imágenes más panorámicas y paisajísticas, además de que se estandariza con las producciones cinematográficas. Este cambio en el ratio de aspecto genera dos escenarios distintos:

- Si tenemos un televisor analógico (4:3), veremos los contenidos digitales (16:9) más pequeños o recortados.
- Si tenemos un televisor digital (16:9), veremos los contenidos analógicos o digitales (con ratio de aspecto 4:3) ensanchados o centrados.

Cuando el ratio de aspecto del televisor y del contenido coincide, la imagen se ve en su totalidad y sin distorsiones.

Algunas estaciones de transmisión realizan adecuaciones personalizadas. La figura 2.3 describe lo antes mencionado.

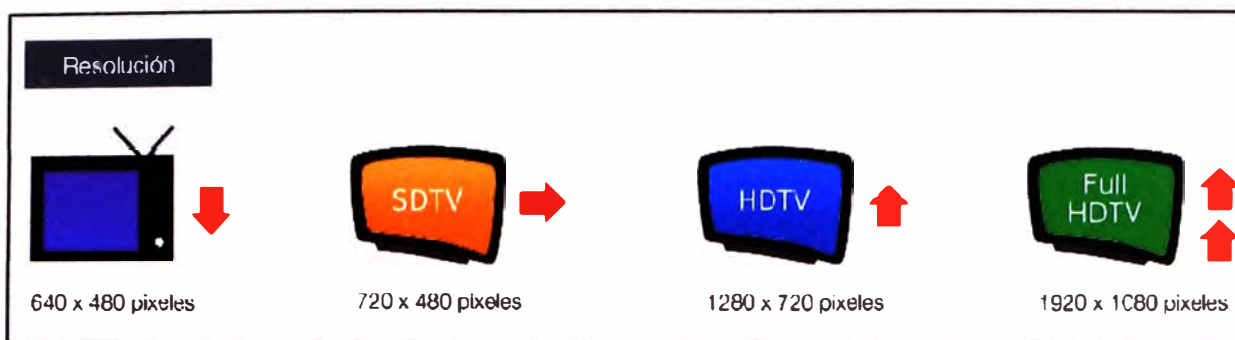


Figura 2.3 Más calidad de video

Fuente: Consejo Consultivo de Radio y Televisión

2.4.4 Más calidad de audio

La calidad del sonido en la televisión analógica es similar a la proporcionada por una cinta magnética o casete, pudiendo ser transmitida a través de un canal indiferenciado (mono) o a través de dos canales (estéreo).

La figura 2.4 describe esta ventaja.

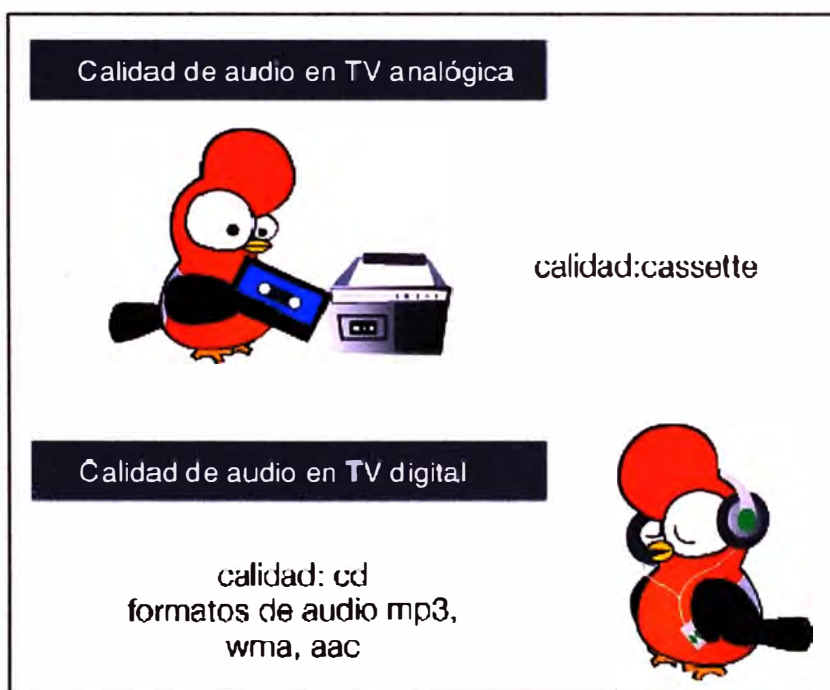


Figura 2.4 Más calidad de audio

Fuente: Consejo Consultivo de Radio y Televisión

Con la TDT, la calidad de audio puede compararse a la de un disco compacto (CD) o a la de los formatos de audio populares – códec – como mp3, wma, aac, con ratios de bits por segundo mayores a 128 Kb/s. Además de esta ventaja, la TDT permite enviar hasta seis canales de audio diferenciados acorde con los sistemas de sonido, minicomponentes y home theater's. Para contar con este beneficio, el contenido de TV debe haber sido grabado con los seis canales diferenciados (como ya se viene realizando con algunas películas).

Si se cuenta con un televisor analógico con un decodificador instalado para captar la señal digital, se podrá disfrutar de la mejora de la calidad del audio sin inconvenientes. No obstante, para gozar de los seis canales de audio diferenciados, esta opción debe estar disponible tanto en el contenido digital como en el decodificador, que debe contar con conectores de audio independientes para cada parlante.

2.4.5 Más contenidos e interactividad

Además de contenidos de audio y video, la TDT permitirá el envío de cualquier tipo de datos, lo que se amplía la posibilidad de ofrecer productos y servicios que anteriormente no eran posibles. Así, los canales de TV podrán enviar una guía de programación interactiva, juegos, opción de incluir subtítulos o canales de audio en otros idiomas, entre otros.

La interactividad de la TDT se potencia cuando se activa un canal de retorno entre los televisores y la estación de TV, ya sea a través del mismo espectro radioeléctrico (en frecuencias más altas, como las utilizadas por los servicios de telefonía celular a través de tecnologías 3.5 G o WiMax) o a través de conexión por cable, ADSL, satélite, entre otros. La existencia de un canal de retorno permite servicios interactivos totales como el acceso a Internet, juegos en red, e-shopping, encuestas, telefonía, banca por TV, entre otros, la figura 2.5 describe lo antes dicho.

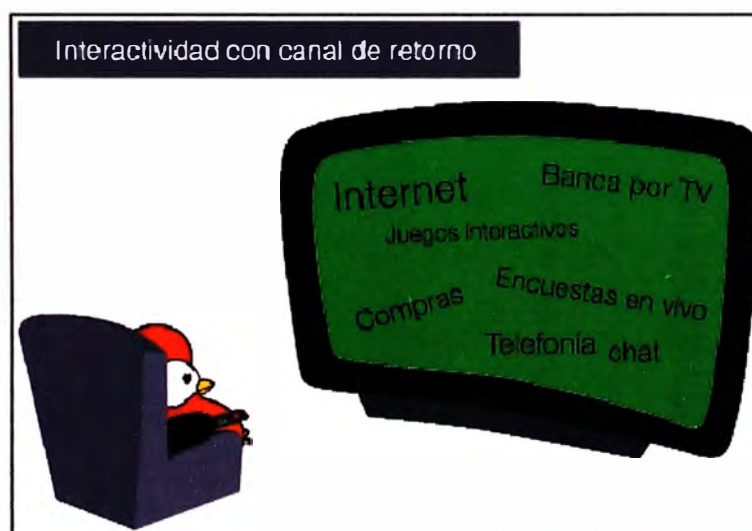


Figura 2.5 Interactividad

Fuente: Consejo Consultivo de Radio y Televisión

2.4.6 Movilidad y portabilidad

La movilidad es la capacidad de recepción de la televisión a través de dispositivos móviles como teléfonos celulares, mientras que la portabilidad es la posibilidad de utilizar los televisores en situaciones de movimiento, como equipos incorporados en buses, autos y

trenes. Con la TDT, se podrán recibir las señales de TV en ambas situaciones sin pérdida de calidad y de manera gratuita, como lo describe la figura 2.6.

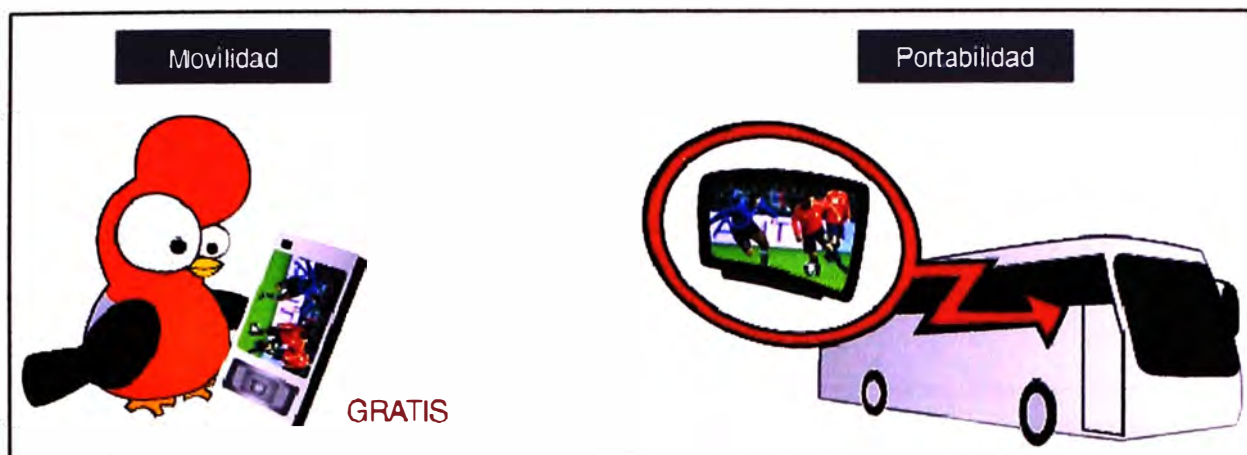


Figura 2.6 Movilidad y portabilidad

Fuente: Consejo Consultivo de Radio y Televisión

Es importante destacar que, a diferencia de los estándares ATSC (norteamericano), DVB-T (europeo) Y DTMB (chino), el estándar japonés y brasilero ISDB-T la transmisión de televisión para dispositivos móviles no requiere de un espacio adicional en el espectro radioeléctrico gracias al servicio de transmisión one-seg.

2.5 Estándar de Televisión Digital Terrestre

El estándar de Televisión Digital Terrestre es el conjunto de principios técnicos acordados para el uso digital del espectro radioeléctrico. Es el requisito básico para iniciar la transmisión de TDT en un área geográfica, ya que todo el ciclo de transmisión y recepción debe estar acorde al estándar definido. Por tanto, los televisores, decodificadores, transmisores, consolas y demás implementos de transmisión, además del software, deben 'compartir' el mismo lenguaje, estándar o característica tecnológica que permita una intercomunicación exitosa.

El escenario mundial ha definido actualmente cuatro estándares de TDT:

- **ATSC Forum (Advanced Televisión Systems Committee)** de los Estados Unidos de América. Sistema adoptado (hasta abril de 2010) por más de 10 países: Estados Unidos, Canadá, México, Corea del Sur, Bahamas, Bermudas, El Salvador, Honduras, Samoa Americana, Guam e Islas Marianas del Norte.
- **DVB-T (Digital Video Broadcasting - Terrestrial)** de la Unión Europea. Sistema adoptado (hasta abril de 2010) por aproximadamente 70 países, entre ellos los países integrantes de la Unión Europea, la India, Uruguay, Colombia, Panamá, Sudáfrica, Arabia Saudita, Hong Kong, Singapur, Taiwán, Vietnam y Australia.

- **ISDB-T International (Terrestrial Integrated Services Digital Broadcasting International)** de Japón y Brasil. Sistema adoptado (hasta abril de 2010) por 7 países: Japón, Brasil, Perú, Chile, Argentina, Venezuela y Ecuador.
- **DTMB o DMB-T (Digital Terrestrial Multimedia Broadcast)** de la República Popular China. Sistema adoptado (hasta abril de 2010) por China (incluyendo Hong Kong).

2.6 Que se necesita para acceder a la TV Digital

Estar dentro de la zona con cobertura de TDT y para recibir señal digital necesita una de las configuraciones descritas en la figura 2.7.

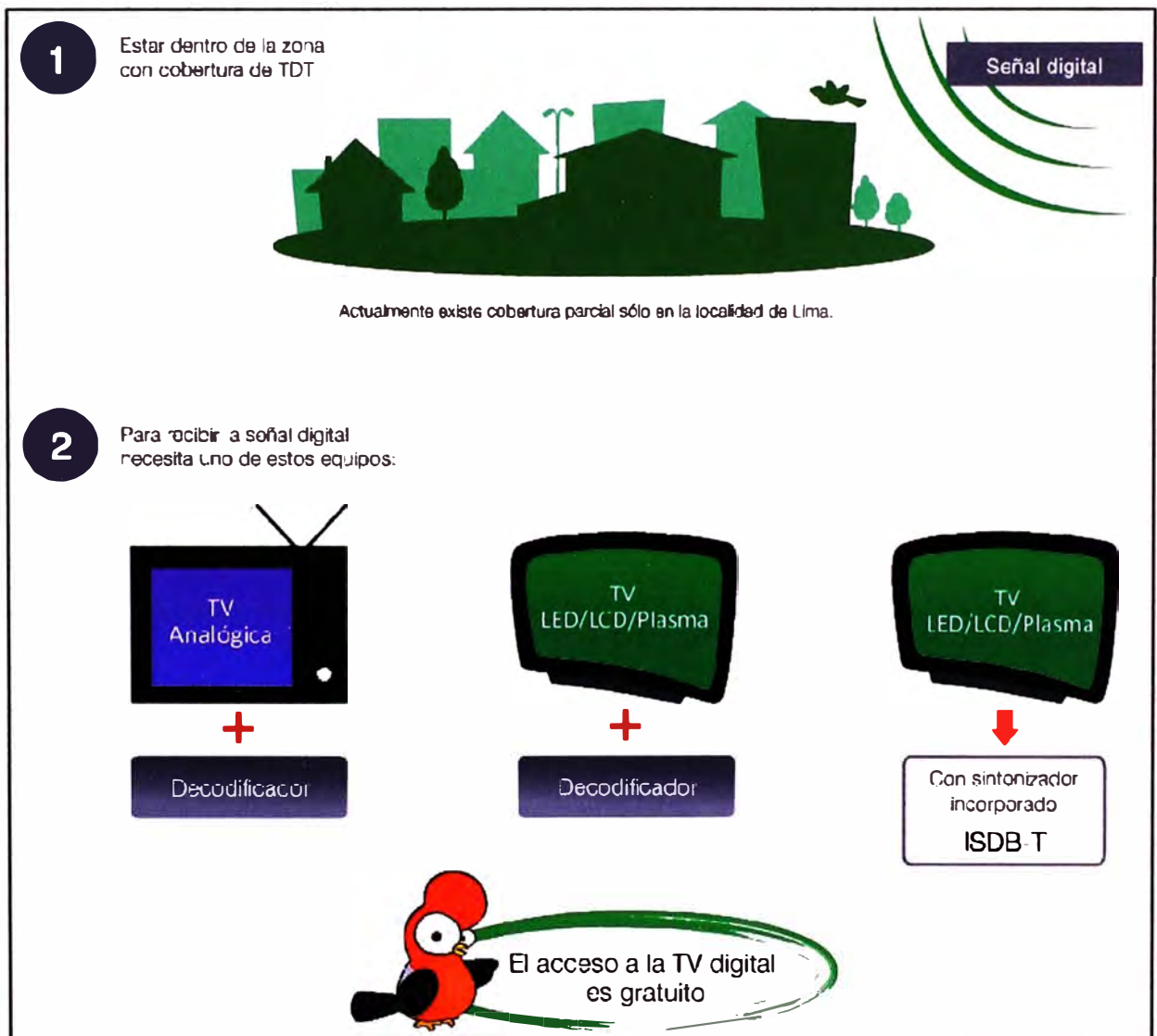


Figura 2.7 Acceso a la TV Digital

Fuente: Consejo Consultivo de Radio y Televisión

El Informe anual de evaluación del proceso de implementación de la Televisión Digital Terrestre (TDT) en el Perú, muestra un panorama al detalle actual del Estado [2].

CAPÍTULO III PLATAFORMA DE DESARROLLO

En este capítulo se señala la plataforma de desarrollo usada en este trabajo, veremos la implicancia de la inclusión de Ginga, en la norma ISDB-Tb, repasaremos la arquitectura de este llamado middleware y las interfaces de desarrollo que hacen posible la implementación de aplicaciones interactivas; así como también revisaremos los diseños propuestos para la concepción de estos aplicativos. Las diferentes herramientas para el desarrollo que existen en la actualidad, así como una primera prueba del aplicativo interactivo propuesto sobre la máquina virtual Ginga, emulador distribuido por el sitio oficial del middleware Ginga.

3.1 El Middleware Ginga

Middleware es una capa de software que yace entre el código de las aplicaciones y la infraestructura de ejecución (plataforma de hardware y los sistemas operacionales embebidos). Un middleware para aplicaciones de tv digital generalmente consiste de motores de lenguajes y librerías de funciones que habilitan un fácil y rápido desarrollo de aplicaciones de TV.

Ginga es el nombre del Sistema de TV Digital terrestre Japonés-Brasileño (ISDB-Tb) middleware y que sigue las recomendaciones "estándares" de la Unión Internacional de Telecomunicaciones ITU para servicios de televisión IP.

3.2 Ambiente declarativo y ambiente imperativo

El universo de las aplicaciones de TVD (TV Digital) puede ser distribuido en un conjunto de aplicaciones declarativas y un conjunto de aplicaciones imperativas. La entidad inicial de una aplicación, es decir, aquella que dispara la aplicación, es la que define a qué conjunto la aplicación pertenece, dependiendo si esta entidad es codificada de acuerdo a un lenguaje declarativo o imperativo.

Note que aplicaciones declarativas pueden contener entidades imperativas y viceversa, lo que las caracteriza es solamente la entidad inicial. Los lenguajes declarativos enfatizan la descripción declarativa de una tarea, en vez de su descomposición paso a paso, en una

definición algorítmica del flujo de ejecución de una máquina, como lo hacen las descripciones imperativas.

Debido a su alto nivel de abstracción, tareas descritas de forma declarativa son más fáciles de ser concebidas y entendidas, sin exigir un programador especialista, como normalmente es necesario en las tareas descritas de forma imperativa. Sin embargo, un lenguaje declarativo normalmente propone un determinado dominio de aplicaciones y define un modelo específico para este dominio. Cuando una tarea enlaza con el modelo de lenguaje declarativo, el paradigma declarativo es, en general, la mejor opción.

Los lenguajes imperativos son bien expresivos y de propósito general, no obstante, a un costo elevado. Ellos normalmente exigen un programador especialista, que generalmente colocan en riesgo la portabilidad de una aplicación, y el control de la aplicación está expuesto a errores cometidos por el programador. Sin embargo, en los casos donde el foco de realización de una tarea no enlaza con el foco del lenguaje declarativo, el paradigma imperativo es, en general, la mejor opción.

En el caso del middleware del estándar brasileño, los dos ambientes son exigidos en los receptores fijos y móviles, entretanto, solamente el ambiente declarativo es exigido en los receptores portátiles.

En su ambiente declarativo el Ginga da soporte para el desarrollo de aplicaciones declarativas desarrolladas en el lenguaje NCL (Nested Context Language), que pueden contener entidades imperativas especificadas en el lenguaje Lua. Principalmente por su gran eficiencia y facilidad de uso, Lua es el lenguaje de script del NCL.

En su ambiente imperativo, el Ginga da soporte a las aplicaciones desarrolladas en Java. Un puente formado entre los dos ambientes da soporte a las aplicaciones híbridas con entidades especificadas en NCL, Lua y Java.

La arquitectura de referencia del middleware Ginga puede ser dividida en tres módulos principales, Ginga-CC, Ginga-NCL, y Ginga-J, los dos últimos módulos componen los servicios ofrecidos por Ginga, tal y como lo muestra la figura 3.1.

El Ginga-J es el subsistema lógico del middleware Ginga responsable por el procesamiento de aplicaciones imperativas escritas en lenguaje Java. Este subsistema sigue la norma ABNT NBR 15606-4.

Ginga-NCL es el subsistema lógico del middleware Ginga responsable por el procesamiento de aplicaciones declarativas NCL. NCL (Nested Context Language) y su lenguaje

script Lua componen la base para el desarrollo de aplicaciones declarativas. La especificación de este subsistema tiene como base las Normas NBR 15606-2 y ABNT NBR 15606-1.

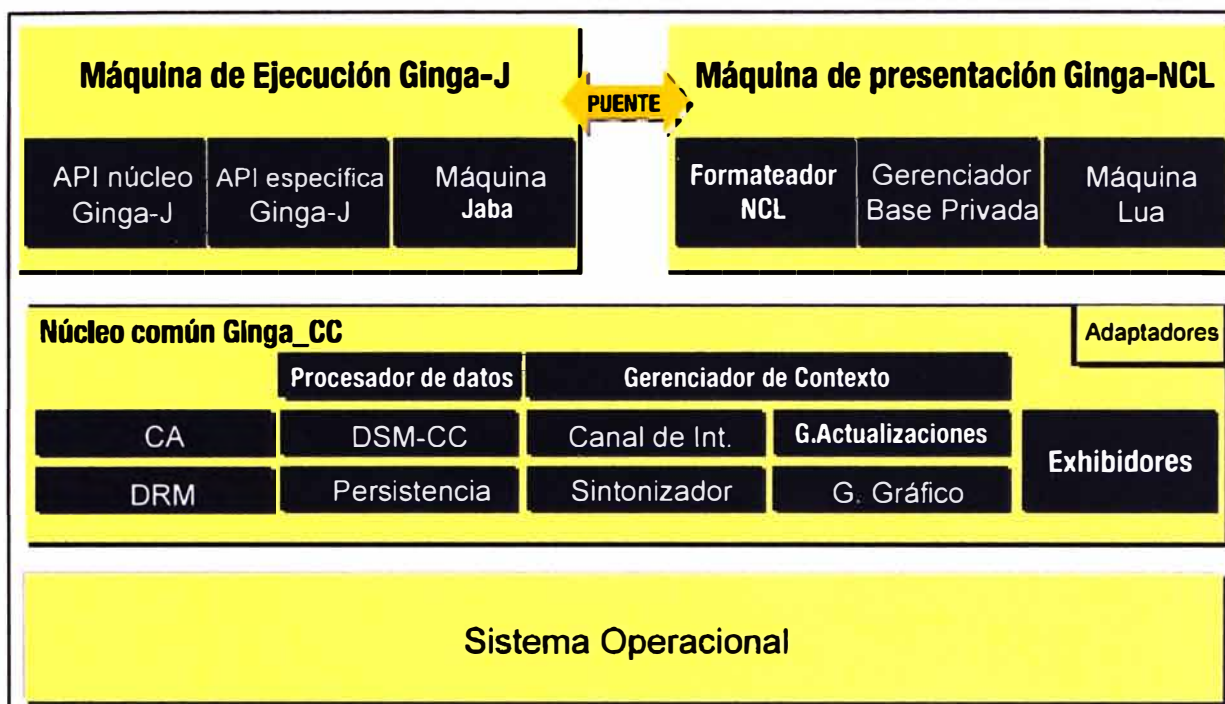


Figura 3.1 Middleware Ginga

Fuente: Foro SBTVD

El Ginga-CC (Ginga Common Core) es el subsistema lógico que provee toda la funcionalidad común al soporte de los ambientes tanto imperativo como declarativo. La arquitectura del Sistema permite que únicamente el módulo Ginga-CC deba ser adaptado donde la plataforma será embebida. De esta manera el Ginga-CC provee un nivel de abstracción de la plataforma de hardware y del sistema operativo [3].

A diferencia de otros sistemas por ejemplo el sistema europeo, no existe cualquier relacionamiento maestro-esclavo entre los ambientes declarativo e imperativo del middleware Ginga. Al contrario, ellos son ambientes pares con procesos de comunicación muy bien definidos especificado por las APIs simbolizadas en la figura 3.1 por el Puente.

Como veremos a continuación, debido a que la problemática del diseño de aplicaciones interactivas de TV Digital encaja con el ámbito declarativo. El alcance de este trabajo se circunscribe al desarrollo de aplicaciones interactivas usando el Lenguaje NCL. No es parte de este trabajo el desarrollo de aplicaciones interactivas usando el modelo imperativo, que usa el lenguaje Java a través de Ginga-J.

3.3 NCL como lenguaje declarativo de Ginga

NCL (Nested Context Language) es un lenguaje de aplicación XML que permite a los autores editar presentaciones multimedia interactivas. NCL es parte de las especificaciones

de codificación de datos del Sistema Brasileño de Televisión Digital (SBTVD). Es la especificación del lenguaje utilizado por el motor de presentación del middleware Ginga.

Con NCL, un autor puede describir el comportamiento espacio-temporal de una presentación multimedia, asociar hiperenlaces (interacción con el usuario) entre los objetos de media, definir comportamientos alternativos para una presentación (adaptación) y describir el layout de presentación entre varios dispositivos.

3.4 NCM el modelo conceptual de datos del lenguaje NCL

Toda lengua declarativa tiene como base un modelo conceptual de datos. Un modelo conceptual debe presentar los conceptos estructurales de los datos, así como los eventos relacionados entre los mismos. El modelo debe también definir las reglas estructurales y las operaciones sobre los datos para la manipulación y actualización de las estructuras.

El lenguaje declarativo NCL tiene por base el modelo NCM (NESTED CONTEXT MODEL) [4]. En este trabajo haremos apenas una introducción de los conceptos del NCM, de un modelo bastante informal. La figura 3.2 presenta las entidades básicas del modelo. En la figura, los términos han sido colados deliberadamente en inglés para una mejor armonía con los elementos del lenguaje NCL.

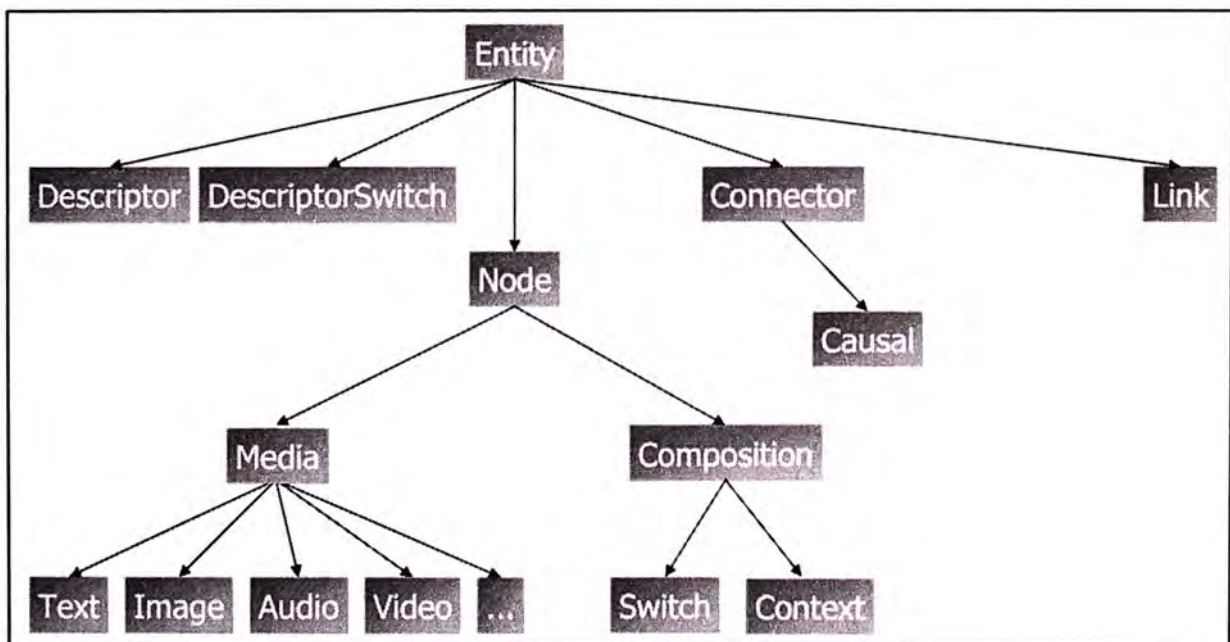


Figura 3.2 Modelo conceptual NCM básico

Fuente: Programando en NCL 3.0

Un nodo (Node en la figura 3.2) NCM tiene un identificador, un contenido y un conjunto de anclajes. El contenido de un nodo es formado por un conjunto de unidades de información. La noción exacta del que constituye una unidad de información es parte de la definición del nodo y depende de su especialización, conforme será ejemplificado más

adelante. Un anclaje (Anchor en la figura 3.3) es un subconjunto de las unidades de información de un nodo. Un nodo de media, o un nodo de contenido (media en la figura 3.2) representan lo que llamamos un objeto de media. Los nodos de contenido deben ser especializados en subclases para definir mejor la interpretación del contenido (texto, imagen, audio, video, objetos con código imperativo Lua, objetos con código imperativo Java, entre otros).

Por ejemplo, en un nodo de video, el conjunto de información puede ser formado por los cuadros que, en secuencia, componen el contenido. Un anclaje, en ese caso, puede definir un trecho del video. Un anclaje especial es aquel que delimita todo el contenido. Anclajes en NCM son definidas por separado del contenido de un nodo. Como ya dijimos, NCM es un modelo basado en estructura (structure-based), y no en el contenido de la media (media-based) como XHTML.

Los nodos NCM poseen propiedades (property en la figura 3.3), como, por ejemplo para nodos de media, su color de fondo, su posición en la pantalla, entre otros, Los objetos de media se relacionan a través de propiedades y anclajes que constituyen sus interfaces.

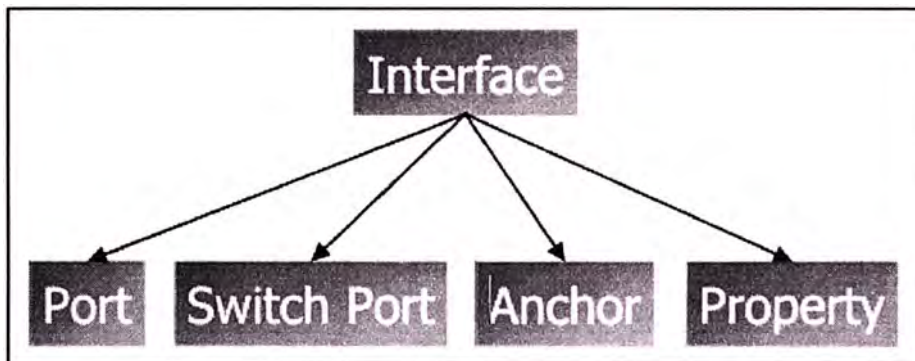


Figura 3.3 Interfaces de un nodo NCM

Fuente: Programando en NCL 3.0

Objetos, (nodos) de media pueden ser exhibidos de formas diferentes, conforme la situación. La forma como un objeto de media debe ser exhibida, donde y por quien es especificada en la entidad descriptor (Descriptor en la figura 3.2). Es en esa entidad que dadas las características iniciales en el inicio de la presentación, pueden cambiar con el tiempo. Debemos notar que el descriptor define donde el objeto debe ser presentado, incluyendo al dispositivo de exhibición. Así, esa entidad es la base para el soporte de los múltiples dispositivos de exhibición. Podemos asociar varios descriptores a un nodo de media, que serán escogidos de acuerdo con la situación y de acuerdo con una regla (rule).

El conjunto de descriptores alternativos debe ser definido dentro de la entidad switch de los descriptores (descriptorSwitch en la figura 3.2). Esa entidad es la que va a permitir a

la NCL la definición declarativa de la forma como un contenido debe ser presentado “adaptada” a cierto contexto de exhibición.

Los objetos de media se pueden relacionar. Por ejemplo, los relacionamientos temporales y espaciales entre objetos de media. Un relacionamiento es definido por una relación y por los participantes de la relación. En NCM, una relación es definida por la entidad conector (connector en la figura 3.2). Un conector define una relación a través de sus papeles (roles) y de la “cola (glue)” entre los papeles. Por ejemplo: un trecho de nodo de media, al ser iniciado, debe iniciar la presentación de un trecho del nodo de media. Los papeles de “un trecho de nodo de media” pueden ser ejercidos por cualquier anclaje, por ejemplo: en el momento exacto de una novela (trecho de un nodo de vídeo) en que una actriz da una mordida en un pedazo de pizza, debe ser iniciada la propaganda de la pizzería. (Un nodo de imagen).

En NCM, los conectores pueden ser casuales y de restricción, pero en el caso de la NCL 3.0, en su perfil para la TV digital, los conectores son siempre casuales (la “cola” es siempre de causalidad), o sea, cuando las condiciones en un conjunto de papeles fueran satisfechas, un conjunto de acciones deben ser aplicadas en un conjunto de papeles. Conectores definen relaciones n:m entre interfaces de nodos. En NCM, el enlace entre el papel de un conector y una interfaz de un nodo se da a través de entidad enlace (link en la figura 3.2). Así, un enlace es compuesto por un conector y por lazos (binds) entre los papeles de los conectores e interfaces de los nodos que ejercen el papel. A través de enlaces, relacionamientos de sincronismo temporal y espacial pueden ser definidos, en particular relacionamientos con interacción del usuario.

Como se dijo, un descriptor o un switch de descriptores, puede ser asociado a un nodo, definiendo como, donde y por qué será exhibido. Esas entidades pueden ser asociadas al nodo a través de un atributo del nodo, a través de un lazo de un enlace. En el caso de doble asociación, las definiciones contenidas en las asociaciones realizadas por ellos se superponen aquellas realizadas por atributos de los nodos.

Tenemos ahora un ejemplo típico de una aplicación de TV. Una novela es compuesta de capítulos que a su vez, son compuestas de escenas, que son compuestas por tomas, compuestas de trechos de video, trechos de audio, textos de leyenda, informaciones adicionales sobre un autor, propagandas insertadas (que por sí solos pueden contener otros objetos de media) y otros objetos de media. Organizar los objetos independientes de forma temporal y espacial como serán presentados es muy importante. Las entidades NCM que

permiten tal organización son llamadas composición (composition en la figura 3.2) y forman la base de la organización lógica de un documento NCM. Note que los objetos de media se pueden relacionar en cada uno de esos niveles organizacionales, o sea, en cada una de esas composiciones.

Un nodo de contexto (context en la figura 3.2) es un tipo de nodo de composición cuyo contenido incluye un conjunto de nodos de contenido y un conjunto de nodos de composición, recursivamente. De ahí el nombre de contextos anidados, Los nodos de contexto poseen como atributo adicional un conjunto de enlaces relacionando sus objetos (nodos). Un nodo de contexto posee en sus interfaces, tanto anclajes como propiedades, puertas. Las puertas son mapeadas hacia otras interfaces interiores en la composición, y son ellas quienes permiten la exposición de composiciones, controlando, las interfaces de sus nodos internos. Así, cada enlace contenido en el conjunto de enlaces de un nodo de contexto C define un relacionamiento entre interfaces de nodos recursivamente contenidos en C.

Con el objetivo de permitir la especificación de alternativas de nodos que serán exhibidos, el NCM define una entidad llamada nodo switch. El nodo switch (switch en la figura 3.2) es otra especialización de los nodos de composición. El contenido de un nodo switch es un conjunto que puede incluir nodos de contexto y de contenido. El nodo switch tiene un atributo adicional que define, para cada nodo contenido en su conjunto de nodos, una regla asociada. Las reglas son definidas en una lista ordenada. En un instante dado de exhibición, el primero nodo que tenga su regla habilitada como verdadera debe ser elegido como la alternativa seleccionada. Así como un switch de descriptores es el soporte necesario para la definición de alternativas de la forma de exhibición de un mismo contenido de media, un nodo switch es el soporte necesario para la definición de alternativas de contenidos.

Además de la ya mencionada lista ordenada de puertos, común a todo nodo de composición, los switch poseen una lista ordenada de puertos-switch (switch port en la figura 3.3). Puertos-switch constituyen a lo más un punto de interfaz. La definición de puertos-switch permite que los enlaces sean definidos anclando en nodos switch, independiente del nodo hijo que será seleccionado. Los puertos tradicionales del switch permiten que enlaces sean asociados a alternativas específicas.

Si esa alternativa no fuera seleccionada, el enlace será simplemente ignorado durante la presentación del documento.

3.5 Edición Declarativa con NCL

Para construir un documento hipermedia, es necesario definir el que se quiere tocar, donde, cómo y cuándo.

3.5.1 El que tocar

En general la primera cosa que consideramos cuando comenzamos a concebir un programa audiovisual interactivo es su contenido: que videos, imágenes, textos y otras medias o programas serán presentados por el programa. Ese contenido es representado a través de nodos de media. Todo nodo de media es definido dentro de un contexto, En la NCL, el elemento `body` es el contexto que contienen todos los nodos del documento, sean nodos de media o contextos. En la figura 3.4 se ilustra un documento con 5 nodos de media, 4 de los cuales están dentro de un contexto `ctxMenu` anidado al `body`.

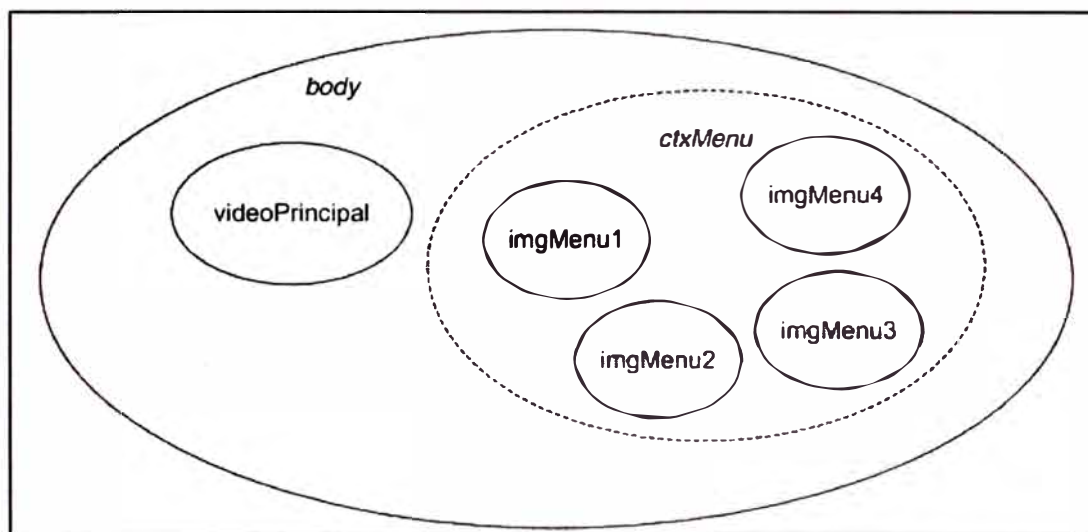


Figura 3.4 Representación de nodos de media y composición

Fuente: TV digital interactiva en el Brasil se hace con Ginga: Fundamentos, Patrones, Edición Declarativa y Usabilidad.

3.5.2 Donde tocar

A medida que definimos el contenido de nuestro programa, también comenzamos a definir las áreas donde cada media será presentada en la pantalla, a través de elementos denominados regiones. Una región (`region`) indica una posición y las dimensiones del área donde una media será presentada. En otras palabras, una región sirve para inicializar una posición de los nodos de media en una locación específica. Es importante observar que una región define donde las medias podrán ser presentadas, mas no una asociación con cada media particular. Esa asociación es hecha a través de un descriptor, como veremos a continuación.

3.5.3 Cómo tocar

La asociación de una media a una región es definida a través de un descriptor (descriptor). Descriptores son utilizados también para definir la forma como la media deberá ser presentada. Por ejemplo, un descriptor de una media de audio puede definir su volumen; de una media de imagen, algún grado de transparencia; o de una media de texto, que será presentada visualmente o leída por un sintetizador de voz, entre otros. Al ser definido un descriptor, se define una región a la cual él estará asociado. Toda media que utiliza aquel descriptor estará asociada a la región correspondiente. Un descriptor no posee una representación gráfica definida. Es importante observar que, así no se quiera alterar la forma como una media será presentada, es necesario un descriptor para asociar la media a la región donde deberá ser presentada.

3.5.4 Cuándo tocar

Para definir cuál es el primer nodo del documento a ser presentado, se debe crear un puerto (port) en el contexto body para ese nodo. En caso haya más de un puerto en el contexto body, los nodos mapeados por todos los puertos son iniciados en paralelo. Identificar por donde el documento puede comenzar a ser presentado es apenas una característica específica de un puerto. De hecho, los puertos son necesarios para dar acceso a los nodos (de media o contextos) internos a un contexto cualquiera, y no solo a body.

Para definir cuando un nodo de media será presentado en relación a otros, son creados enlaces, que son utilizados para establecer el sincronismo entre los nodos y para definir la interactividad del programa. El comportamiento de cada enlace es dado por los conectores.

3.6 Estructura de un programa en NCL

Un programa en NCL es un archivo escrito en XML. Todo documento NCL posee la siguiente estructura:

- Una cabecera del archivo NCL (líneas 1 y 2)
- Una sección de cabecera del programa (sección head, líneas 3 a la 13), donde se definen las regiones, los descriptores, los conectores y las reglas utilizadas por el programa
- El cuerpo del programa (sección body, líneas 14 a la 17), donde se definen los contextos, nodos de media, enlaces y otros elementos que definen el contenido y la estructura del programa;
- Por lo menos un puerto que indica donde el programa comienza a ser exhibido (port pInicio, línea 15); y

- Un cierre del documento (línea 18).

La figura 3.5 muestra el documento antes descrito [5].

```

1: <?xml version="1.0" encoding="ISO-8859-1"?>
2: <ncl id="exemplo01"
   xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile" >
3: <head>
4:   <regionBase>
5:     <!-- regiões da tela onde as mídias são apresentadas -->
6:   </regionBase>
7:   <descriptorBase>
8:     <!-- descritores que definem como as mídias são
   apresentadas -->
9:   </descriptorBase>
10:  <connectorBase>
11:    <!-- conectores que definem como os elos são ativados e
   que eles disparam -->
12:  </connectorBase>
13: </head>
14: <body>
15:   <!-- porta(s) -->
16:   <!--
   contextos,
   nós de mídia e suas âncoras,
   elos e
   outros elementos -->
17: </body>
18: </ncl>

```

Figura 3.5 Estructura básica de un documento NCL

Fuente: TV digital interactiva en el Brasil se hace con Ginga: Fundamentos, Patrones, Edición Declarativa y Usabilidad.

3.7 Herramientas de Desarrollo para el NCL

Una vez revisado prudentemente el aspecto teórico para la concepción de una aplicación interactiva para TV Digital, se menciona las herramientas utilizadas para el desarrollo de este trabajo. Para cubrir todo el aspecto práctico requerido para la concepción de una aplicación interactiva, se necesita de herramientas para el desarrollo de documentos “programas” hipermedia. El diseño de estas aplicaciones contempla la visualización de diagramas estructurales así como de diagramas de layout que definen los prototipos del aplicativo, y por supuesto, de editores del lenguaje XML en particular como sabemos el lenguaje NCL.

También será necesario un sistema que haga las veces del sistema embebido donde se prueben los programas diseñados, esto corresponde a una máquina virtual que haga las

veces de un set-top box, en el que podamos probar lo que deseamos que nuestras aplicaciones interactivas realicen. Finalmente, cabe resaltar que estas herramientas son del tipo Software libre, cuidando con ello, el evitar costos de copyright.

3.7.1 NCL Composer

Es una herramienta de edición multimedia flexible y multiplataforma para crear aplicaciones para la TV Digital con NCL, también creada por el Laboratorio Telemidia de la PUC de Rio de Janeiro. [6]

A la fecha de realización de este trabajo, la dirección web: <http://composer.telemidia.puc-rio.br/en/downloadmuestra> las diferentes plataformas en las que podemos usar el NCL Composer, para este caso, el Sistema Linux, la distribución Ubuntu 12.04. En enlace directo de descarga es: http://composer.telemidia.puc-rio.br/downloads/latest_unstable.php?system=deb. El archivo descargado es el siguiente: composer_0.1.4_i386.deb

En un terminal de Linux, instalamos con:

```
$ sudo dpkg -i composer_0.1.4_i386.deb
```

Después de la instalación conforme, en un terminal de Linux, ejecutamos:

```
$ composer
```

Las figuras 3.6 y 3.7 muestran las pantallas de arranque e inicio del Composer.



Figura 3.6 NCL Composer

Fuente: Software NCL Composer



Figura 3.7 Pantalla inicial del NCL Composer

Fuente: Software NCL Composer

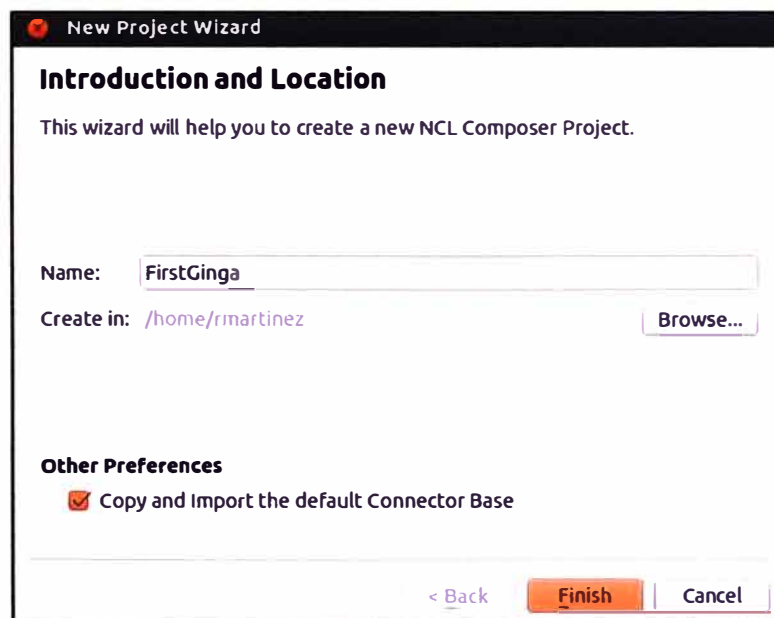


Figura 3.8 Creación del nuevo proyecto con NCL Composer

Fuente: Software NCL Composer

Podemos crear un nuevo proyecto en tres pasos, según la figura 3.7, damos clic sobre New Project, y en la ventana según la figura 3.8, en el campo [Name], colocamos el nombre, y en el campo [Create in], colocamos la ruta donde será creado el Proyecto.

Enseguida, una nueva perspectiva del proyecto creado es mostrada, evidentemente, sin contenido aún, como se ve en la figura 3.9.

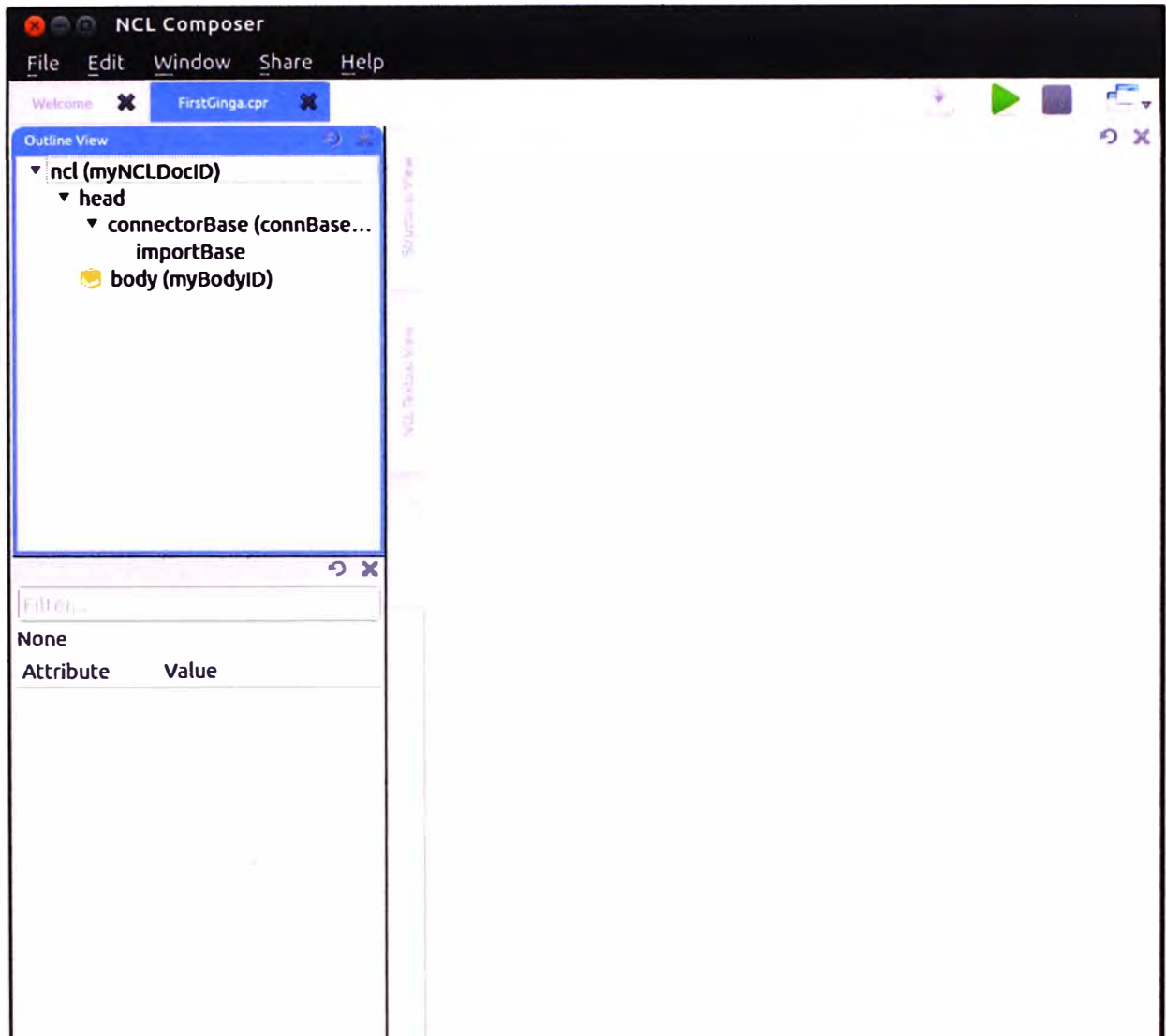


Figura 3.9 Nuevo proyecto creado en el NCL Composer

Fuente: Software NCL Composer

Todo proyecto generado con esta herramienta contempla la visualización de vistas que en la etapa de diseño y concepción de la aplicación interactiva para TV Digital son usadas comúnmente; tenemos primero, la vista de layout, donde diseñamos la interfaz del aplicativo, o sea, el prototipo; la figura 3.10 muestra un ejemplo de esta vista.

El Composer por cada proyecto creado, conserva dos archivos que están sincronizados internamente, un primer archivo binario de extensión .cpr el cual guarda información gráfica del proyecto, esta se refleja en las vistas de layout y estructural, mientras que el otro archivo de extensión .ncl contiene el código ejecutable del proyecto. Este último archivo es el que finalmente se trasladará con los objetos de media necesarios conjuntamente empaquetados al set-top box para su posterior ejecución.

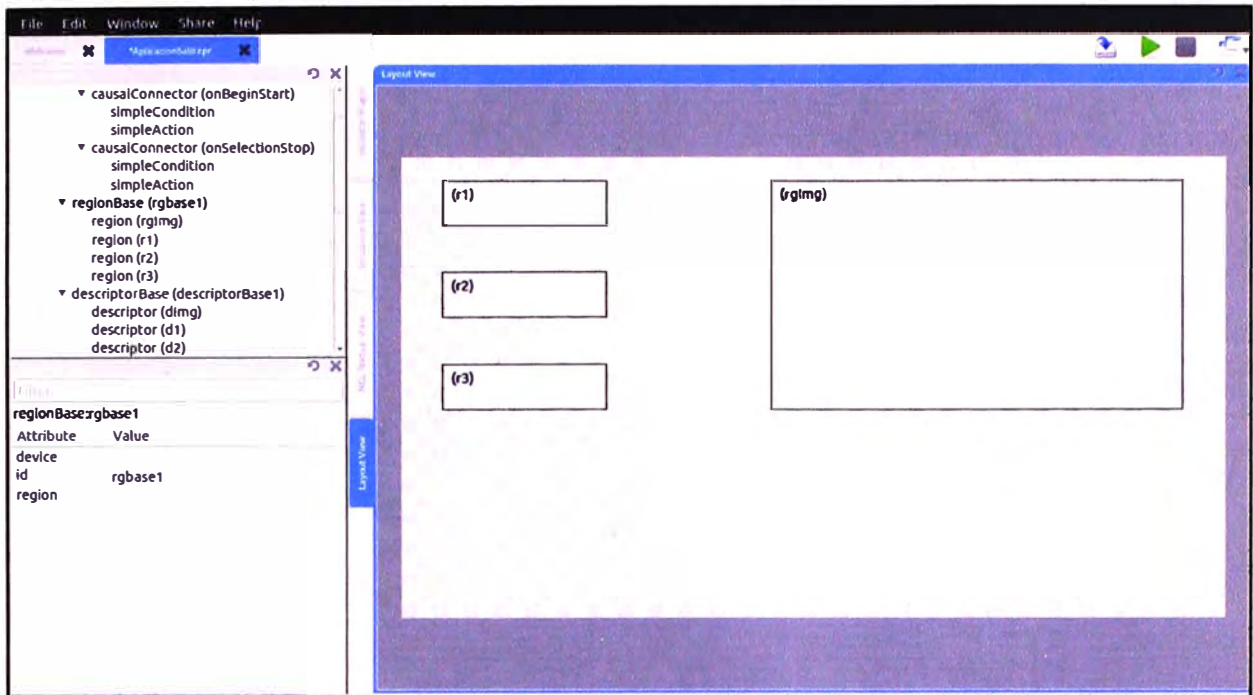


Figura 3.10 Vista de Layout de un proyecto con NCL Composer

Fuente: Software NCL Composer

Otra vista importante es la vista estructural, la cual nos permite visualizar los nodos de media y sus relacionamientos entre ellos, la figura 3.11 describe esta vista.

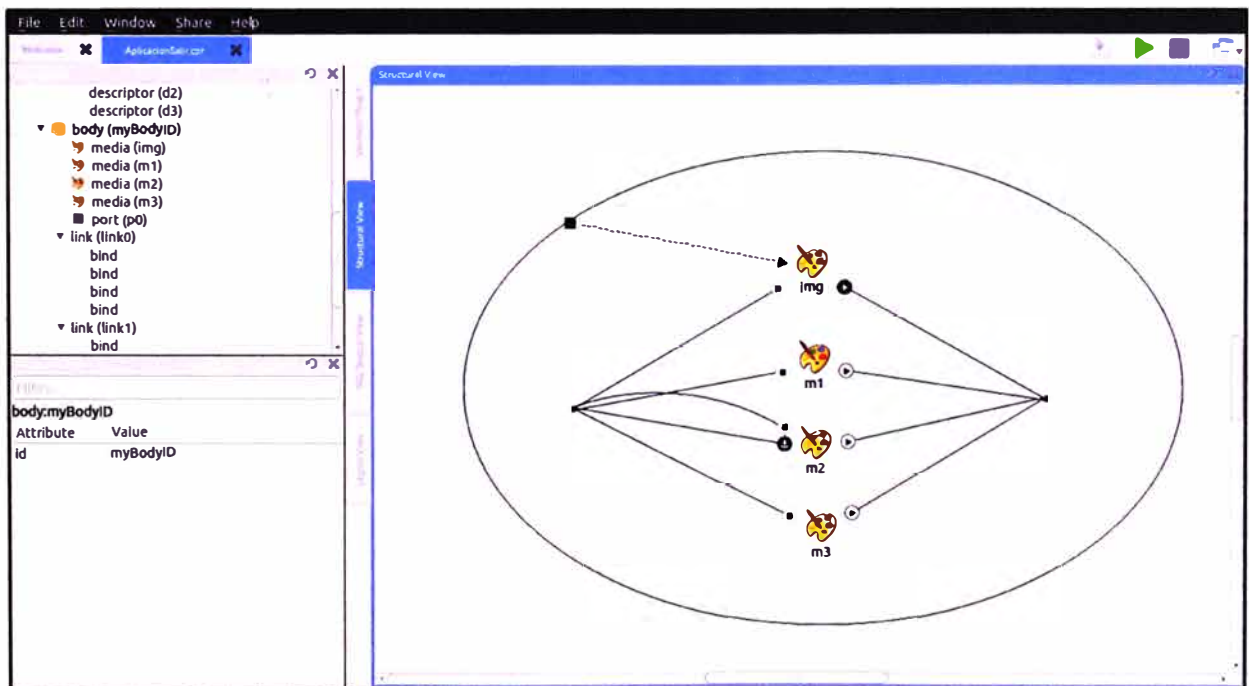


Figura 3.11 Vista estructural de un proyecto con NCL Composer

Fuente: Software NCL Composer

Finalmente, no menos importante, podemos observar el código NCL generado a partir del diseño del prototipo en la vista de layout, tanto como las inclusiones de nodos de media y sus relacionamientos en la vista estructural. Como se ha indicado anteriormente, algunos

de los elementos usados para la creación de este código, por ejemplo, los descriptores, no tienen representación gráfica, por tanto NCL Composer, brinda la posibilidad de ingresar estos elementos usando el clic derecho del mouse sobre los elementos de la vista de outline, lo que nos permite insertar este tipo de elementos en proyecto, donde finalmente sí pueden ser visualizados en la representación de árbol de esta vista y como texto, un ejemplo es mostrado en la figura 3.12.

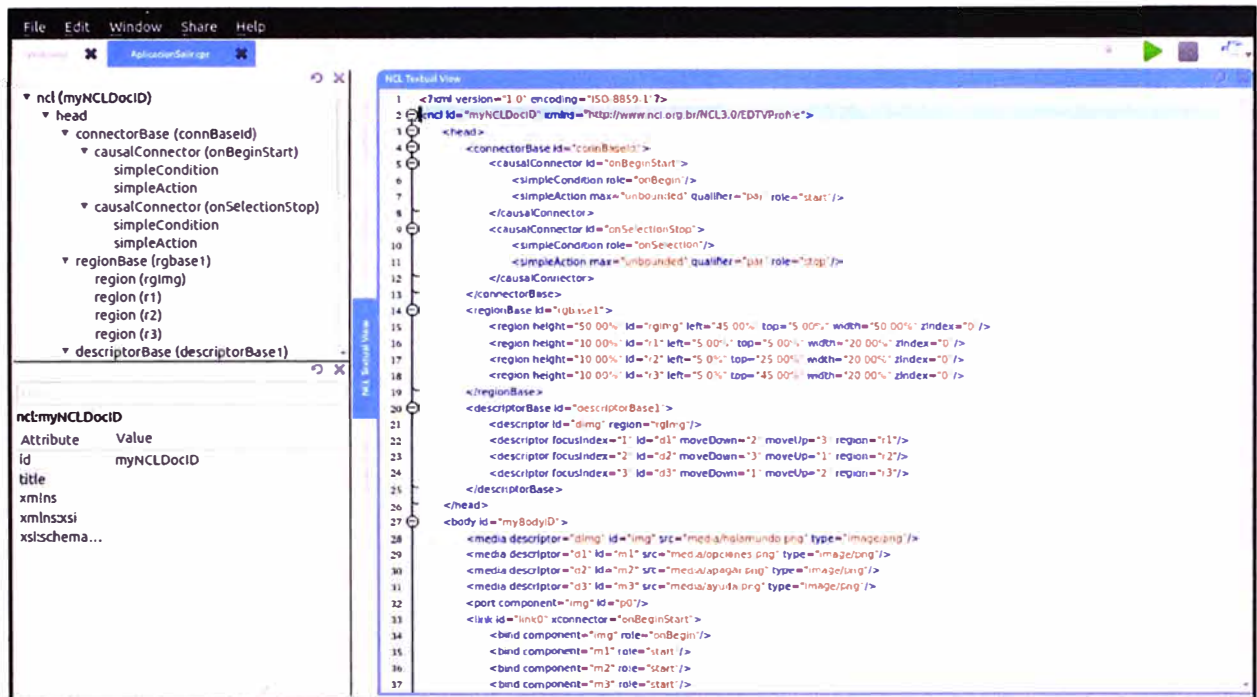


Figura 3.12 Vista textual de un proyecto con NCL Composer

Fuente: Software NCL Composer

Es importante mencionar que en esta vista, la del documento NCL resultante debido a la elaboración del proyecto con la asistencia de las otras vistas, no es editable. Estas primeras liberaciones de la herramienta NCL Composer, no contemplan tal característica; por ello, es importante, siempre revisar esta vista, puesto que muestra la conformidad de la sintaxis del documento final. La vista de propiedades, permite como manifiesta su nombre poder visualizar, editar y guardar las propiedades de todos los nodos del proyecto, siendo ésta la única que permite ingreso textual de datos según sea su origen, tanto de los visuales, como los que no lo son.

Además de disponer de las vistas mencionadas para el desarrollo de los aplicativos interactivos, NCL Composer también provee una interfaz para el upload de aplicativos al escenario de prueba, dígame, máquina virtual donde colocaremos nuestros documentos NCL y elementos de media que son usados por estos documentos. La interfaz hacia esta máquina virtual es un cliente ssh que necesita elementos mínimos para establecer la comu-

nicación con ella, información tal y como la dirección IP asignada a la máquina virtual, y el registro de usuario (login) de la misma. Los campos a llenar pueden ser visualizados en la figura 3.13.

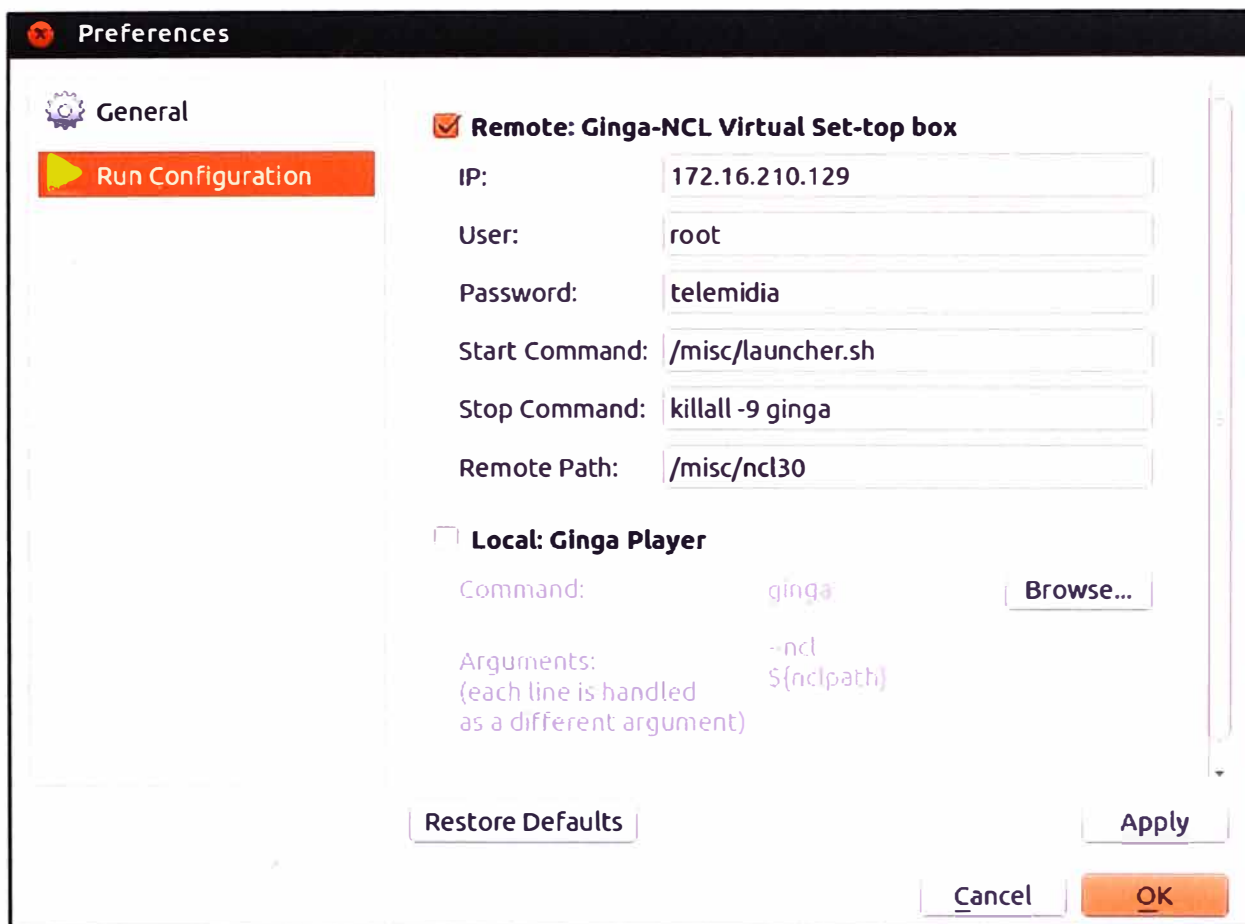


Figura 3.13 Preferencias del NCL Composer

Fuente: Software NCL Composer

A la fecha, la versión instalada del NCL Composer es la mostrada en la figura 3.14.



Figura 3.14 Versión del NCL Composer

Fuente: Software NCL Composer

3.7.2 La máquina virtual Ginga-NCL set-top box

A la fecha de realización de este trabajo, la dirección web: <http://www.gingancl.org.br/en/ferramentas> provee la máquina virtual comprimida en formato zip, para que pueda ser descargada.

El acceso directo a la descarga de la herramienta es el siguiente: <http://www.gingancl.org.br/sites/gingancl.org.br/files/ferramentas/ubuntu-server10.10-ginga-v.0.12.4-i386.zip> como se observa el archivo descargado tiene como nombre: ubuntu-server10.10-ginga-v.0.12.4-i386.zip

En un terminal de Linux:

```
$ unzip ubuntu-server10.10-ginga-v.0.12.4-i386.zip -d ./vm
```

Y el directorio ./vm será creado donde se encontrará la estructura de directorios de la máquina virtual, el archivo que representa al proyecto “máquina virtual” es el ubuntu-server10.10-ginga-i386.vmx.

Cabe mencionar que esta máquina virtual no es la única disponible y libre para realizar nuestras pruebas, otros países de la región tal y como el Perú lo ha hecho, tienen su propia versión de máquina virtual, e incluso, existen un Live CD distribuida por la misma web, que permite cómodamente transportar en un USB la máquina virtual de pruebas.

3.7.3 VMware player

A la fecha de realización de este trabajo, la dirección web: https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/5_0 suministra la descarga del reproductor de la máquina virtual en múltiples plataformas, que, anteriormente hemos descargado, la opción elegida para sistemas Linux, es: VMware-Player-5.0.1-894247.i386.bundle

La instalación de este reproductor, es simple, en un terminal, digitamos así:

```
$ sudo sh ./VMware-Player-5.0.1-894247.i386.bundle
```

Donde seguiremos los pasos para la instalación del mismo; finalmente, podemos comprobar la instalación, digitando en un terminal:

```
$ vmplayer
```

Lo que nos muestra la pantalla inicial del reproductor VMware, según la figura 3.15, en la siguiente página.

En los sistemas libres tal y como las distribuciones de Linux, podemos encontrar también otros reproductores para máquinas virtuales, una opción interesante para revisar es el reproductor de Oracle VM VirtualBox, no cubierto en este trabajo.

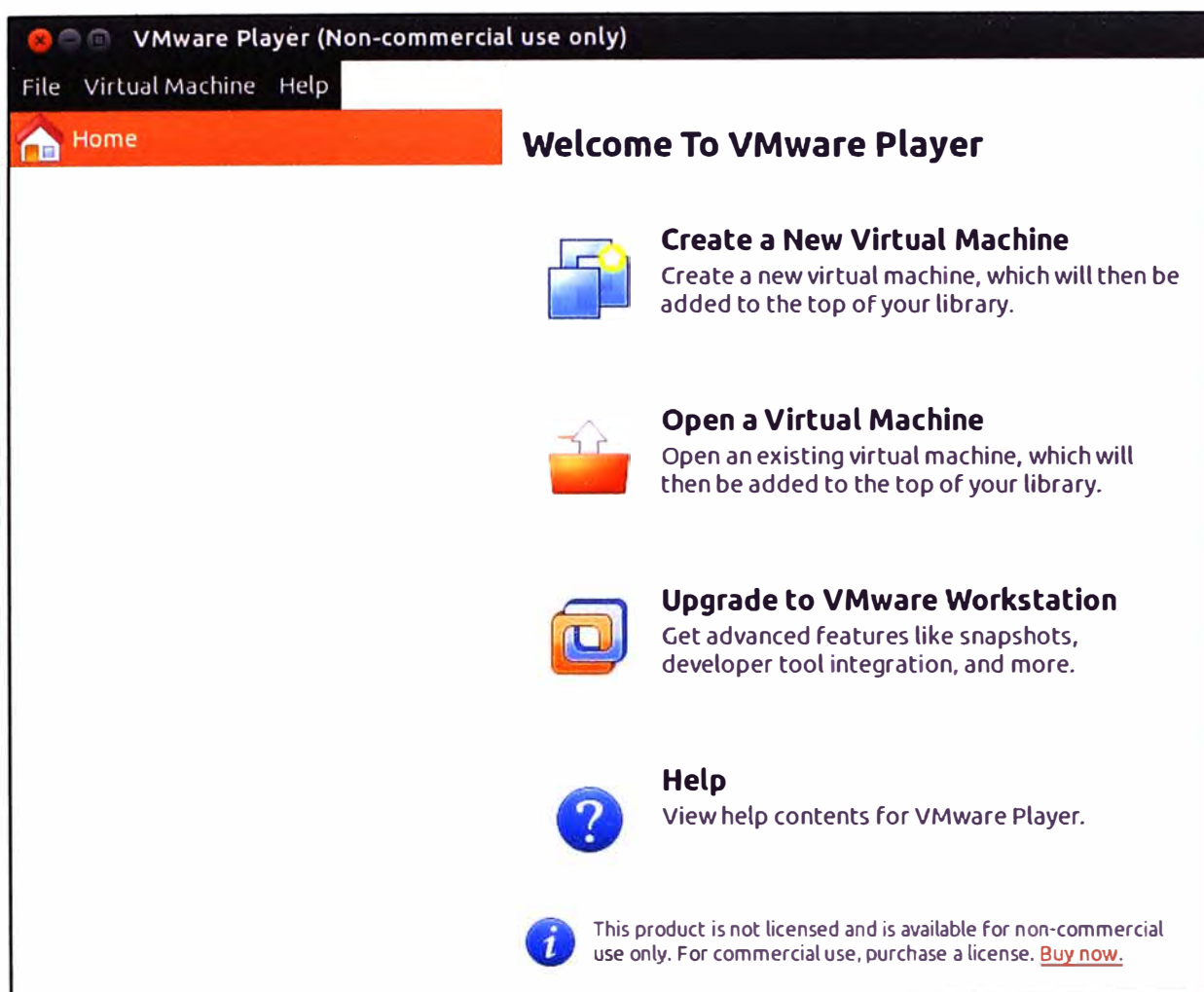


Figura 3.15 Pantalla inicial del reproductor de máquinas virtuales VMware

Fuente: Software VMware

En seguida, visualizamos, las opciones [Create a New Virtual Machine], [Open a Virtual Machine], entre otras. Como se ha mencionado antes, hemos descargado nuestra máquina virtual, elegimos la segunda opción y nos dirigimos al directorio Base, donde se encuentra la carpeta que contiene la máquina virtual, un ejemplo de lo antes mencionado, se describe en la figura 3.16.



Figura 3.16 Abriendo la máquina virtual Ginga-NCL

Fuente: Software VMware

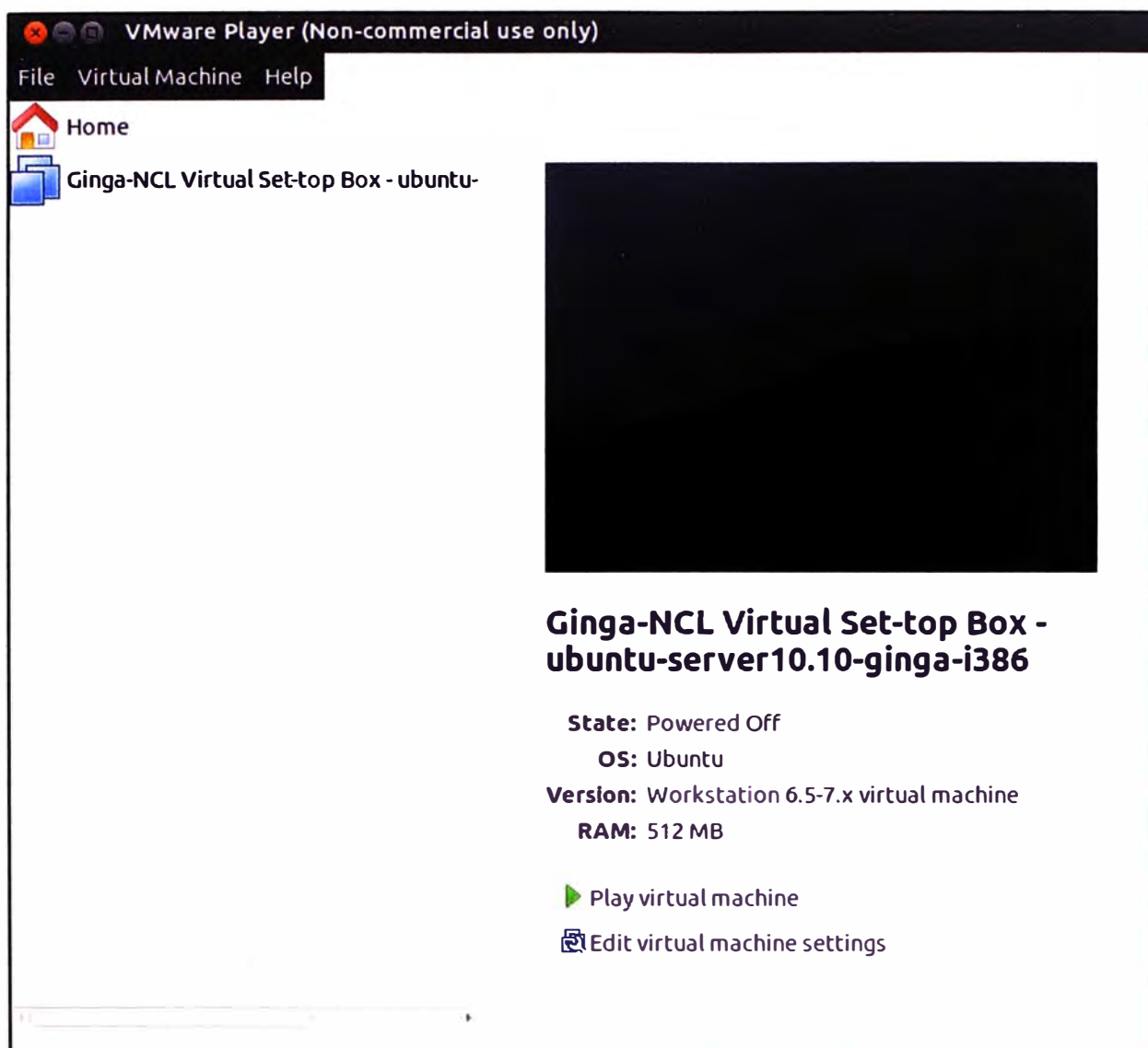


Figura 3.17 Máquina virtual Ginga-NCL registrada en el reproductor de VMware

Fuente: Software VMware

En la figura 3.17 se observa el registro de la máquina virtual en el reproductor, también se observa en la parte inferior, la opción [Edit virtual machine settings], se personaliza la configuración, dando clic sobre esta opción y veremos lo que en la figura 3.18 se muestra.

Se observan las pestañas [Hardware] y [Options], elegimos la primera.

En la pestaña [Hardware], seleccionamos la opción [Network Adapter], seguidamente seleccionamos la opción [NAT], para garantizar la asignación correcta de la dirección IP, en el caso de que nuestra PC pertenezca a diferentes redes, y siempre pueda existir comunicación con la máquina virtual. Esta asignación es dinámica, o sea, no tenemos que colocarla; para conocer cuál es, damos clic en [Save] y a continuación clic en [Play virtual machine], observaremos el mensaje mostrado en la figura 3.19. Y Como lo dice el mensaje,

damos clic en [I copied it], entonces, veremos el grub y la lista de máquinas virtuales disponibles, según lo muestra la figura 3.20.

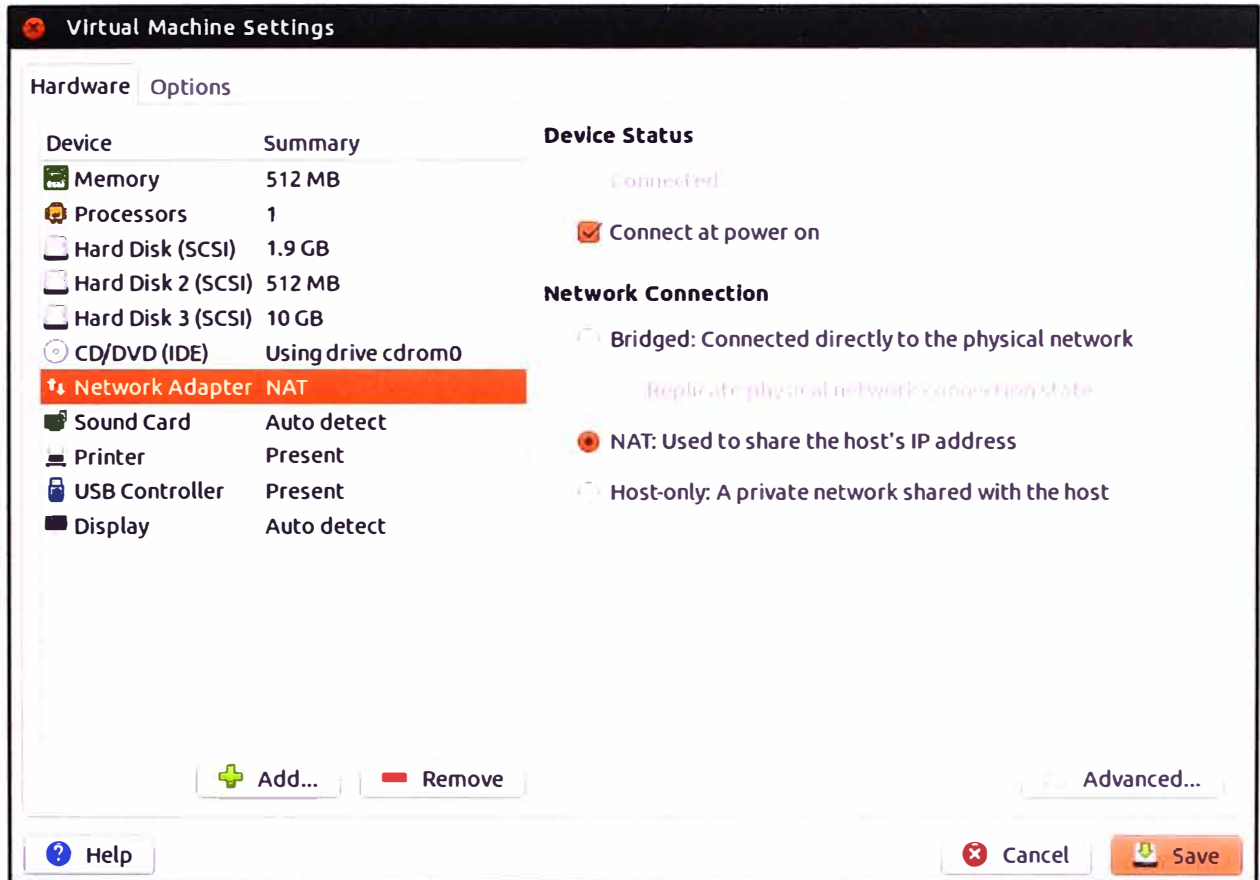


Figura 3.18 Configuración de la máquina virtual Ginga-NCL

Fuente: Software VMware

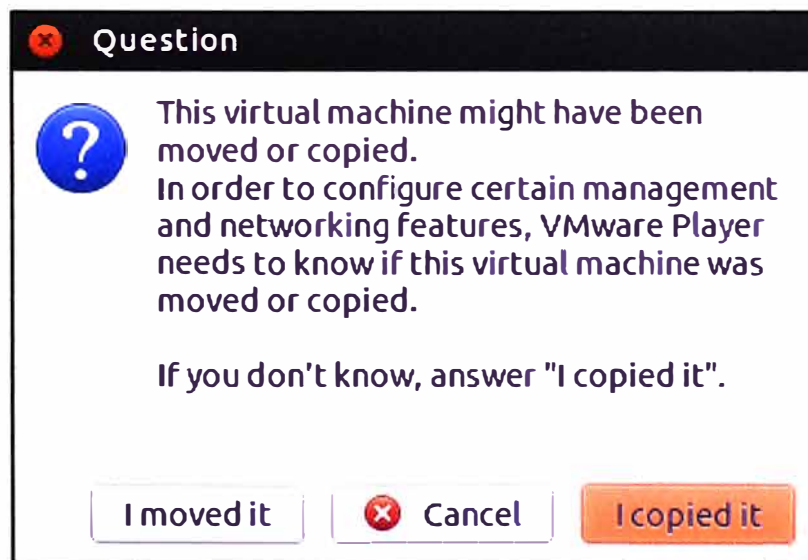


Figura 3.19 Confirmación de copia de la máquina virtual

Fuente: Software VMware

Como se puede observar en el menú del grub, existen diferentes resoluciones en las que se puede iniciar la máquina virtual, todas ellas están disponibles.

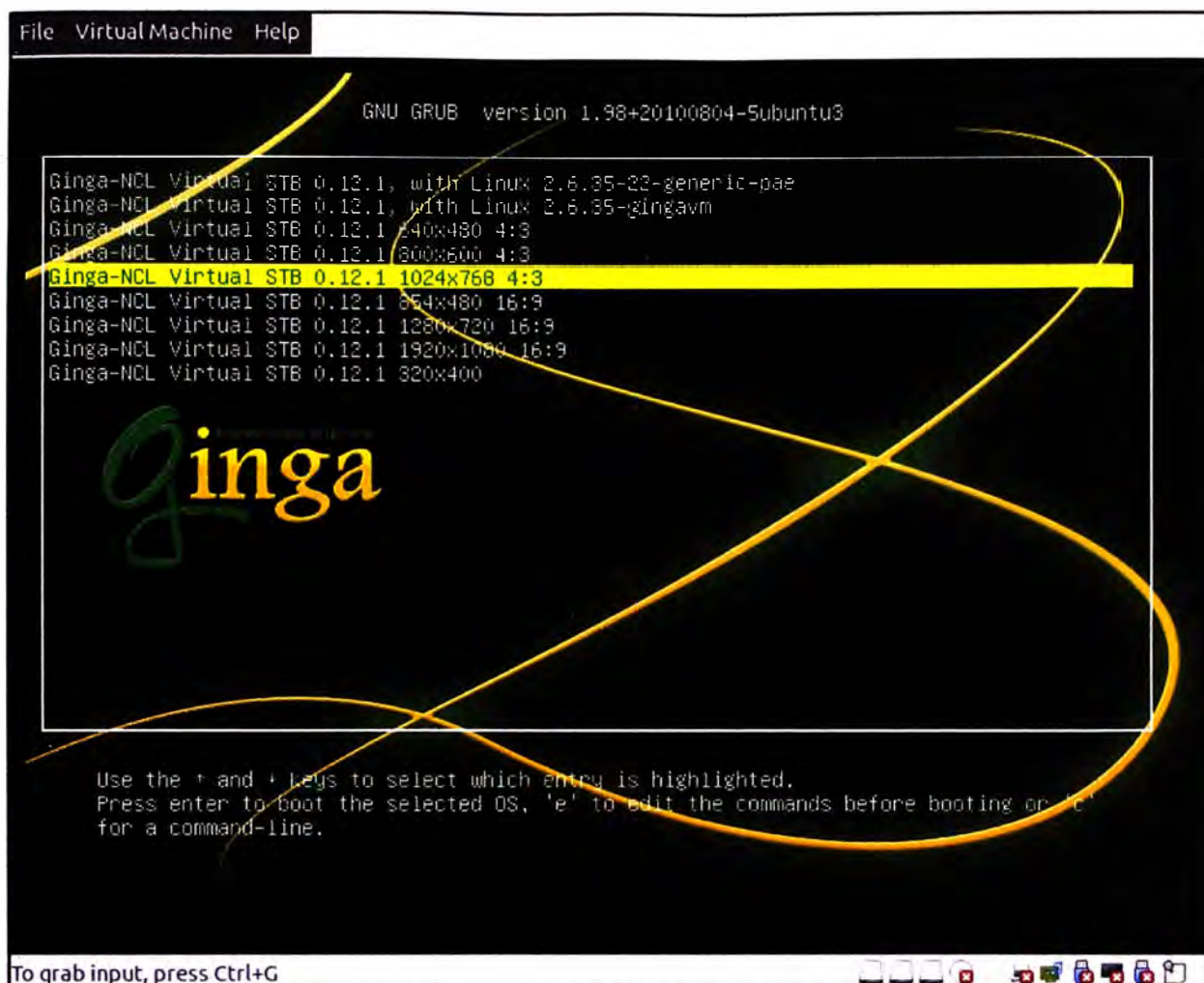


Figura 3.20 Iniciando la máquina virtual Ginga-NCL

Fuente: Software VMware

Las diferentes máquinas virtuales que aparecen en esta lista, en realidad apuntan a la misma máquina virtual, la diferencia entre ellas, consiste en que cada una de ellas, posee un formato de resolución de pantalla diferente, indica el tamaño que tomará cuando la máquina virtual esté iniciada, elegimos una de ellas, y al término del arranque, la figura a continuación, nos muestra la pantalla final.

La selección de la resolución a elegir dependerá del desarrollador y del tipo de aplicativo al que se esté orientando el diseño por implementar; como sabemos es diferente, crear un diseño para una pantalla de un celular, que crear un aplicativo para una pantalla HD o más aún, una pantalla Full HD.

Como se muestra en la figura 3.21, en la parte inferior izquierda, podemos reconocer la dirección IP asignada a la máquina virtual; la que será usada para comunicarnos, ya sea vía el NCL Composer, según la figura 3.13, o directamente, desde una línea de comandos en un terminal de Linux, usando el cliente ssh.



Figura 3.21 Máquina virtual Ginga-NCL iniciada

Fuente: Software VMware

3.8 Upload de un aplicativo NCL a la máquina virtual Ginga-NCL

El comando mostrado en la figura 3.22, manifiesta la intención de conectarse a la máquina virtual cuya dirección es 172.16.210.129 y cuyo usuario es el root, la máquina virtual, solicita el password correspondiente al usuario. El password del root como ya hemos mostrado

```
rmartinez@VPCSB15GL:~$ ssh root@172.16.210.129
Ginga-NCL Virtual STB 0.12.4
Ubuntu 10.10 Server
root@172.16.210.129's password:
Welcome to Ginga-NCL Virtual STB!
New release 'natty' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Sun Dec 30 03:33:22 2012 from 172.16.210.1
root@gingavm:~#
```

Figura 3.22 Login cliente SSH

Fuente: Terminal Sistema Ubuntu

anteriormente es “telemidia”, al ingresarla, la máquina virtual nos da la bienvenida, con el mensaje mostrado y el prompt del sistema de la máquina virtual. Desde este terminal podemos ejecutar directamente la aplicaciones NCL que queremos probar, pero para ello será necesario primero subir estas aplicaciones desde nuestra máquina de desarrollo a la máquina virtual.

Tenemos como ejemplo una aplicación NCL, que está contenido en un directorio llamado [Ginga], que contiene otro directorio llamado [media], donde se colocan los elementos de media típicos en una aplicación interactiva de TV Digital.

En el Directorio raíz, se encuentran el o los documentos NCL que contienen el código explícito, aquel que se desea ejecutar o que contiene el inicio del aplicativo. En la figura 3.23 tenemos un ejemplo, la máquina de desarrollo es “VPCSB15GL”.

```
rmartinez@VPCSB15GL:~$ cd Ginga; pwd; tree
/home/rmartinez/Ginga
├── AplicacionSalir.ncl
├── media
│   ├──
│   ├──
│   └── banner.png
└──
1 directory, 5 files
rmartinez@VPCSB15GL:~/Ginga$
```

Figura 3.23 Aplicación NCL contenida en el directorio Ginga

Fuente: Terminal Sistema Ubuntu

Y a continuación realizaremos la subida del directorio [Ginga] a la máquina virtual cuyo nombre es “gingavm”, cuya dirección IP es la 172.16.210.129 y cuyo contenido a subir en la máquina virtual será colocado en la carpeta raíz [/misc], en la figura 3.24 observamos en un terminal la línea de comando usada.

```
rmartinez@VPCSB15GL:~$ cd; pwd; scp -r Ginga root@172.16.210.129:"/misc/."
/home/rmartinez

Ginga-NCL Virtual STB 0.12.4
Ubuntu 10.10 Server

root@172.16.210.129's password:
opciones.png          100%  58KB  58.2KB/s  00:00
apagar.png            100%  58KB  57.5KB/s  00:00
banner.png            100%  84KB  83.8KB/s  00:00
ayuda.png             100%  58KB  58.0KB/s  00:00
AplicacionSalir.ncl  100% 2343   2.3KB/s  00:00
rmartinez@VPCSB15GL:~$
```

Figura 3.24 Upload del proyecto NCL a la máquina virtual Ginga-NCL

Fuente: Terminal Sistema Ubuntu

```

root@gingavm:~# cd /misc; pwd; tree
/misc
├── carousel
│   └── modules
├── Ginga
│   ├── AplicacionSalir.ncl
│   ├── defaultConnBase.ncl
│   └── media
│       ├── apadon.png
│       ├── apadon.jpg
│       └── banner.png
├── launcher
├── ncl30
│   ├── lost+found
│   └── sample03
│       ├── img
│       │   ├── banner.png
│       │   ├── gray.gif
│       │   ├── green.gif
│       │   ├── start.gif
│       │   └── yellow.gif
│       ├── sample03.ncl
│       ├── txt
│       │   ├── text1.txt
│       │   └── text2.txt
│       └── video
└──
10 directories, 17 files
root@gingavm:/misc#

```

Figura 3.25 Comprobación del proyecto subido a la máquina virtual Ginga-NCL

Fuente: Terminal Sistema Ubuntu

Los porcentajes mostrados en la lista de la figura 3.24 confirman que el upload de cada uno de los archivos se realizó con éxito, a continuación, como ya se ha visto en la figura 3.22, accedemos a la máquina virtual. Y confirmamos que en el directorio [misc] de la máquina virtual “gingavm” ha sido colocado el directorio [Ginga], esto, se observa en la figura 3.25. En este punto estamos listos para ejecutar nuestro aplicativo interactivo, con la línea de comando mostrada en la siguiente sección.

3.9 Ejecución del aplicativo NCL en la máquina virtual Ginga-NCL

Una simple línea de comando ejecuta el aplicativo interactivo demostrativo que se ha implementado en la máquina de desarrollo y se ha subido a la máquina virtual, a efectos de comprobar lo hasta aquí descrito, aquí aquella línea de comando en un terminal.

```
root@gingavm:/misc# ginga --ncl ./Ginga/AplicaciónSalir.ncl
```

La figura 3.26 muestra lo que sucede en la máquina virtual, como se observa; es una aplicación estática. Más adelante, veremos algo funcional en el demostrativo final.



Figura 3.26 Prueba de ejecución del aplicativo NCL en la máquina virtual gingavm

Fuente: Software VMware

Después de realizar nuestras pruebas unitarias, y confirmar la lógica del aplicativo implementado, en la máquina virtual; podemos pasar a la ejecución del aplicativo demostrativo final en el set-top box que en este trabajo estamos considerado.

CAPÍTULO IV DESPLIEGUE DE LA APLICACIÓN DE TDT INTERACTIVA

Este capítulo tiene como protagonista al Set-top Box “EITV DeveloperBox” decodificador de señal digital que permite la ejecución de aplicativos para la interactividad con el usuario. Este set-top box desarrollado en Brasil en el 2010 y puesto a disposición de los desarrolladores de soluciones de TV Interactiva desde el 2011, incorpora el Middleware Ginga ya cubierto, en el capítulo anterior. Además, es parte de una solución conjunta que integra tanto desarrollo de Software como implementación de Hardware.

Este set-top box permite que el desarrollador en el lado receptor, pueda hacer pruebas de software utilizando tanto señal real de video, como señal local emulada a partir de un simple archivo de video. También, dispone de una interfaz Ethernet y un servicio web que permite mediante protocolo HTTP subir los aplicativos desarrollados para ser probados desde el sistema operativo embebido.

4.1 Descripción general del Equipo EITV DeveloperBox

El panel frontal del equipo se muestra en la figura 4.1, y la tabla 4.1 la describe. [7]



Figura 4.1 Set-Top Box EITV DeveloperBox (Panel Frontal)

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Tabla 4.1 Descripción del panel frontal del Set-Top Box

Fuente: EITV Entretenimiento e Interactividad para TV Digital

[Botón]	Descripción
1	Botón de encendido / apagado
2	Selector de canales sintonizados

Las interfaces disponibles en el equipo se pueden observar en la figura 4.2.

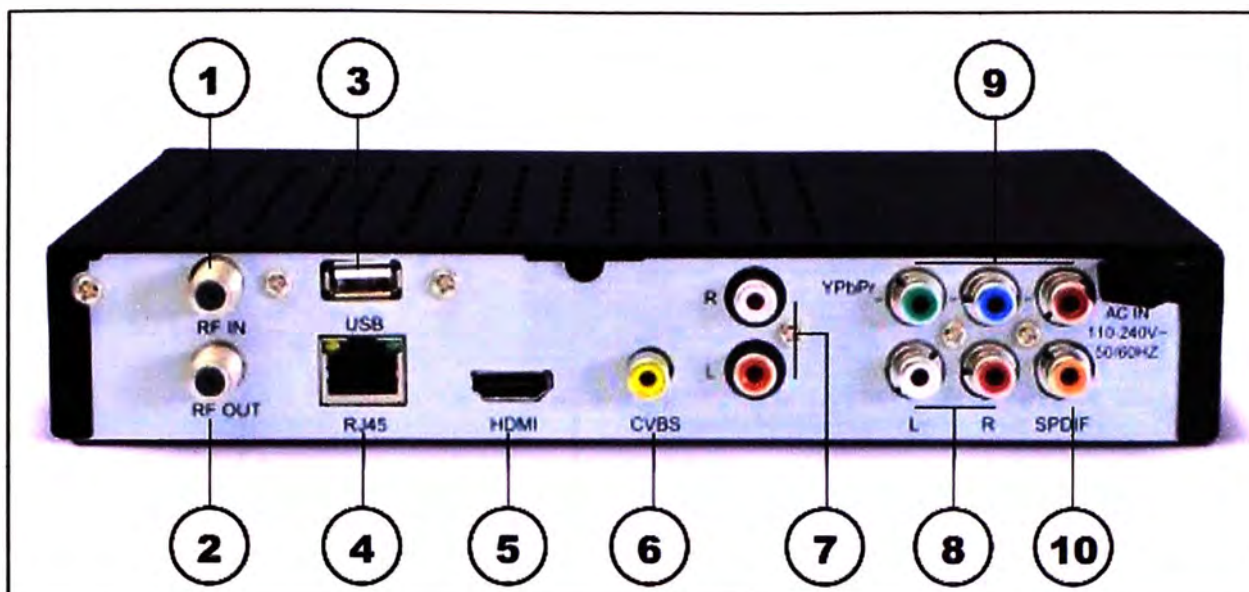


Figura 4.2 Interfaces del set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Las descripciones de la figura 4.2, pueden observarse en la tabla 4.2.

Tabla 4.2 Descripción de las interfaces del Set-top Box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Ítem	Conector	Descripción
1	RF In	Entrada de RF (Antena)
2	RF Out	Salida de RF (Antena)
3	USB	Conector USB
4	RJ-45	Conector de red Ethernet
5	HDMI	Audio y Video Digital
6	CVBS	Audio y video compuesto (CVBS-AV)
7	Audio 1 (R + L)	Audio estéreo 1 (Derecho + Izquierdo)
8	Audio 2 (R + L)	Audio estéreo 2 (Derecho + Izquierdo)
9	RGB	Video componente (YPbPr)
10	SPDIF	Audio Digital (SPDIF coaxial)
11	AC IN	Entrada de alimentación 110/220 VAC – 50/60 Hz.

En la imagen en la figura 4.3 se describe el escenario en el que participa este set-top box. Una conexión de antena UHF es necesaria para la sintonía de señales de TV digital abierta. Una conexión Ethernet está disponible para permitir la sintonía de canales IP y para el uso de aplicaciones que requieren el canal de retorno (web). Al realizar la primera conexión del set-top box según lo descrito anteriormente, se visualiza la inicialización (boot) del set-top box EITV DeveloperBox, mostrado en la pantalla según la figura 4.4.

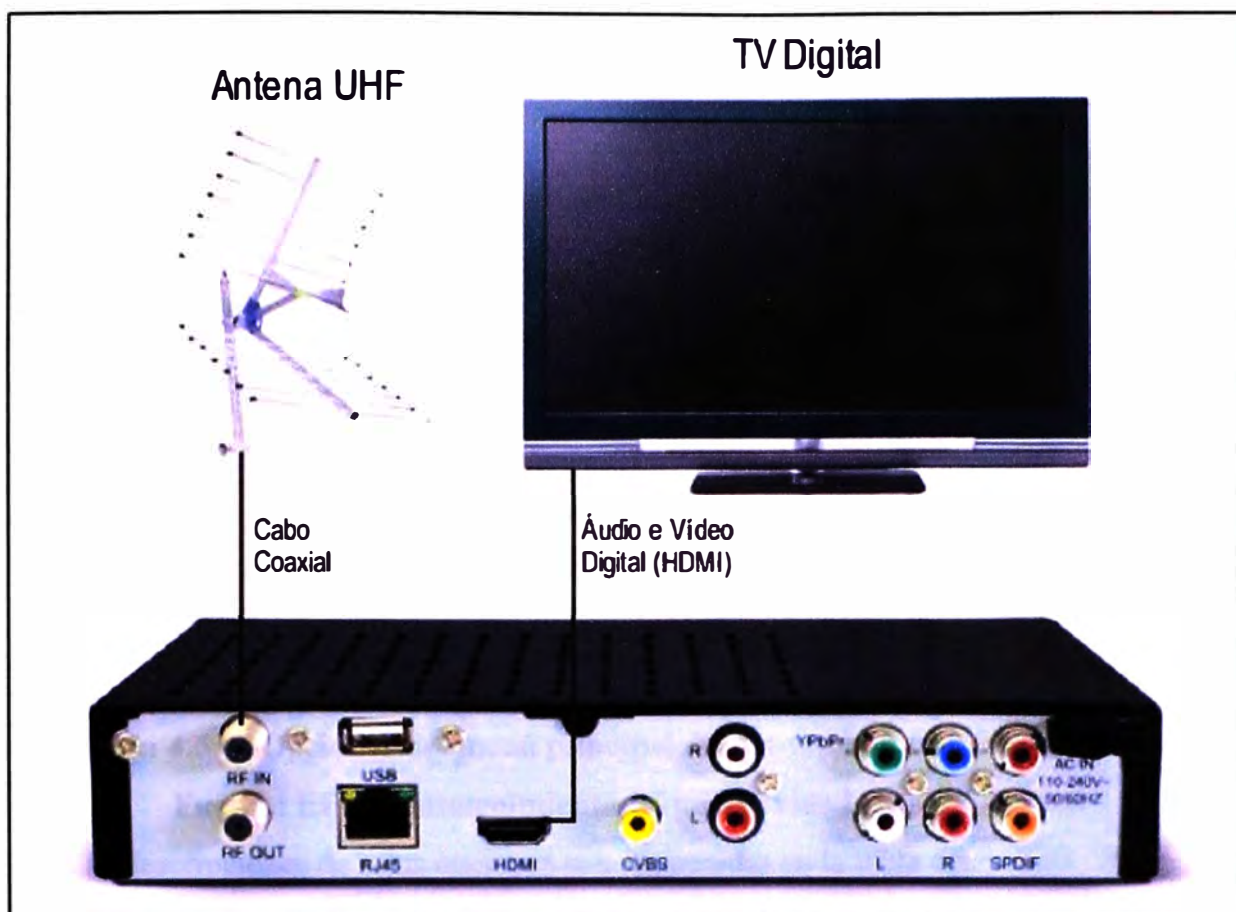


Figura 4.3 Escenario básico para el set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital



Figura 4.4 Boot del set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

El menú principal de EITV DeveloperBox ofrece 6 ítems que ofrecen acceso a submenús que permiten la configuración o los ajustes del equipo. El menú principal es el mostrado en la figura 4.5, exhibido en su idioma original el cual es el portugués.



Figura 4.5 Opciones del menú principal del set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Las descripciones de estas opciones son enunciadas en la tabla 4.3.

A continuación revisaremos las opciones que son de interés para la ejecución de la aplicación interactiva.

4.2 Opción de Interactividad Ginga

Esta opción muestra los aplicativos disponibles cargados en el equipo vía aire (A), red (N) o embebido (E) como muestra la figura 4.6. Esta lista puede ser ejecutada.

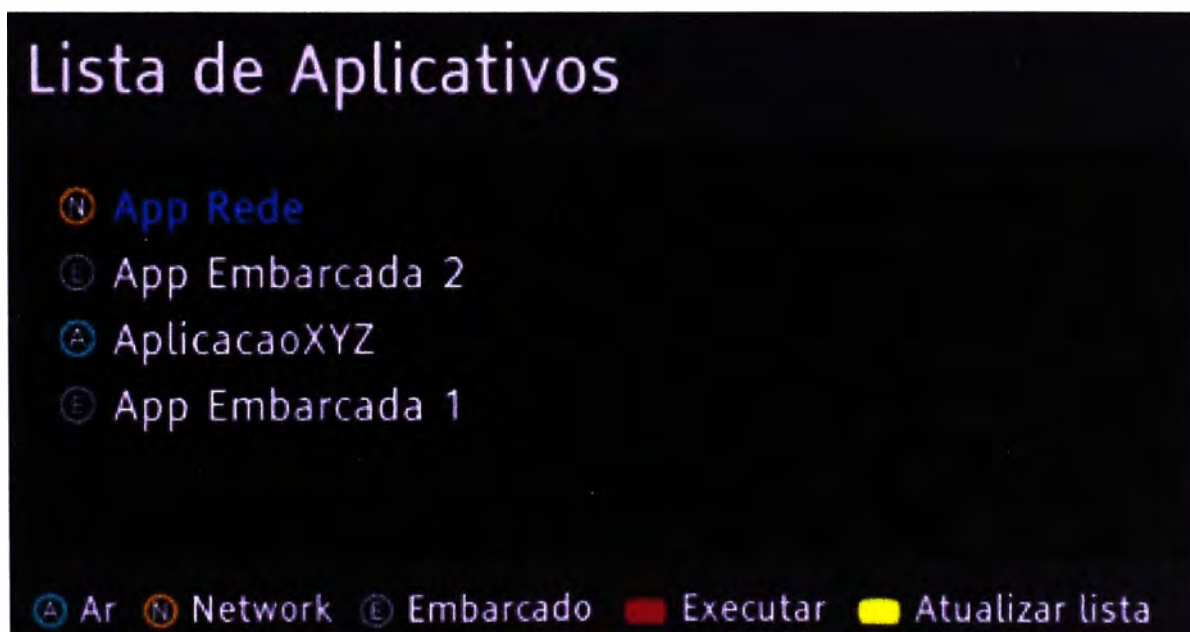


Figura 4.6 Aplicativos registrados en el set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Tabla 4.3 Descripción del menú principal del set-top box EITV DeveloperBox**Fuente:** EITV Entretenimiento e Interactividad para TV Digital

Ítem	Descripción
Canales	Opciones para la exhibición de la lista de canales y de la Guía de programación, Sintonía automática y Bloqueo de Canales.
Imagen	Exhibe opciones para configuración de Formato de video y de la transparencia.
Ginga	Muestra los aplicativos disponibles, por el aire, via red y aplicativos embebidos en el equipo.
Red	Exhibe opciones para la configuración de la interfaz de red Ethernet.
Multimedia	Permite acceder al contenido de una memoria USB conectada a la puerta USB.
Configuraciones	Exhibe opciones para configuración de idioma, cambio de contraseña, bloqueo de canales, de Ginga, Fecha y Hora, actualización, reiniciación y exhibición de la información del producto.

Para ejecutar uno de los aplicativos de la lista mostrada utilice las flechas direccionales del control remoto y a continuación, presione el botón [OK]. O por defecto, el botón rojo del control remoto. Para actualizar la lista presione el botón amarillo en el control remoto.

La ejecución de aplicativos vienen acompañados de mensajes que enuncian su estado, mensajes tales como “Iniciando...” o “Terminado!” son exhibidas a lo largo de ejecución de los aplicativos.

Para ejecutar el aplicativo cargado, presione el botón [OK] Algunos aplicativos son automáticamente cargados durante la selección del canal y la lista de aplicativos exhibida en la figura anterior es actualizada para reflejar tal cambio.

Es posible también utilizar la interfaz usb con el que cuenta el equipo, con la opción [MULTIMEDIA], para este caso, los aplicativos interactivos, están disponibles directamente sin necesidad de estar empaquetados en archivos zip, como ya se ha indicado.

4.3 Opción de Red

Esta opción permite la configuración de la Interfaz de red Ethernet del equipo para comunicación usando el canal de retorno, o sea, vía red y el acceso a canales IP. Aquí es posible configurar la interfaz – dirección IP – que debe ser del tipo DHCP o IP estático. Seleccione la primera opción utilizando las flechas direccionales del control remoto.

Observaremos una pantalla como la mostrada en la figura 4.7.

Si se desea configurar la interfaz de red Ethernet con un IP Fijo, seleccione y rellene los campos Dirección IP, Mascara de red, Gateway y DNS, usando los botones del control remoto, y deje el campo DHCP como desactivado.

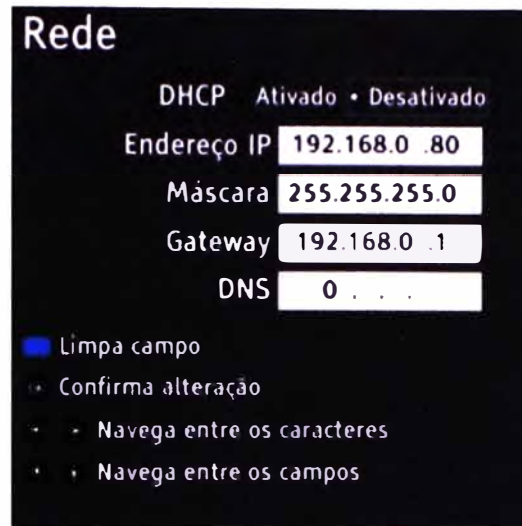


Figura 4.7 Configuración Ethernet del set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Después de colocar las opciones de acuerdo a lo requerido, presione [Confirma alteração] para confirmar el cambio. Un mensaje “Aplicando cambio, por favor espere...” es exhibida mientras el equipo realiza tales alteraciones.

4.4 Servidor Web del EITV DeveloperBox

El equipo posee un servidor web disponible para el acceso de usuarios a través del puerto Ethernet. Para acceder al servidor web, conectamos mediante cable UTP con conectores RJ-45 a la red del usuario, en la cual tenemos a nuestra computadora de desarrollo, comúnmente esta conexión se realiza al switch de la red.

Para acceder al servidor web, digite la dirección configurada en el set-top box en el web browser de su computadora (preferentemente Firefox o Chrome) la pantalla mostrada en la figura 4.8 da conformidad de una configuración correctamente hecha.

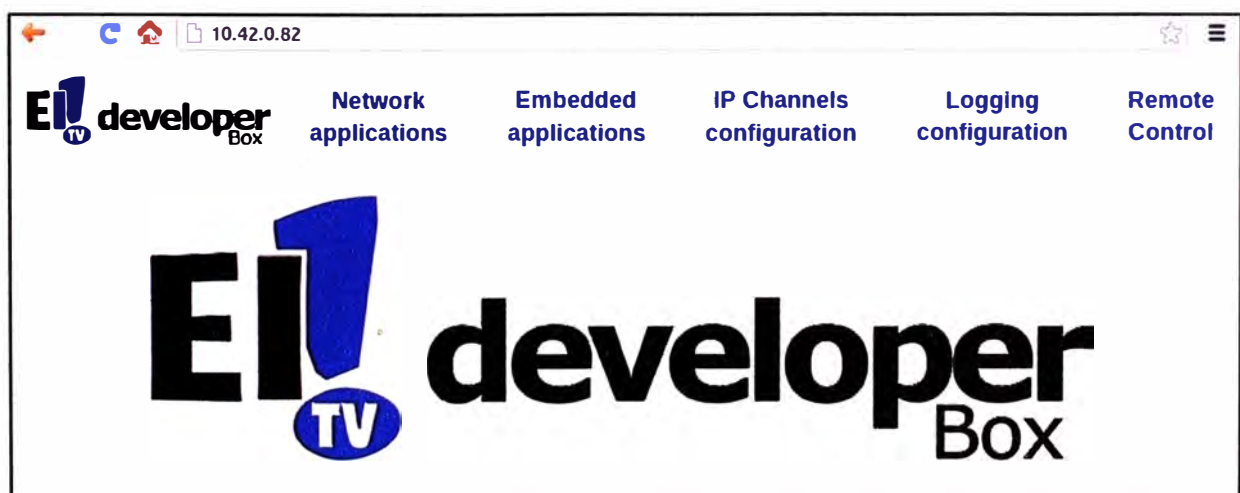


Figura 4.8 Servidor web del set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Según la figura 4.8 las opciones disponibles en esta pantalla son: Aplicaciones de Red, Aplicaciones Embebidas y Configuración del Registro.

4.5 Aplicaciones de Red

En las aplicaciones de red es posible registrar aplicativos en el EITV DeveloperBox que serán reflejados en la pantalla de la figura 4.6 que pueden ser ejecutadas por el equipo.

Una pantalla inicial de las aplicaciones de Red es exhibida en la figura 4.9.

Network applications on the set-top box

Name	Application Type	Base Directory	Initial Class	Location	Conexão
Add New Application					
Name	<input type="text"/>				
Application Type	<input type="text" value="v"/>				
Base Directory	<input type="text"/>				
Initial Class	<input type="text"/>				
Location	<input type="text" value="v"/>				
Address	<input type="text"/>				
File	<input type="text"/>				
Login	<input type="text"/>				
Password	<input type="text"/>				
Add					

Figura 4.9 Registro de aplicaciones de red del set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Todos estos campos deben ser rellenados para que el aplicativo pueda ser cargado en el EITV DeveloperBox. Los aplicativos registrados aquí, aparecen en la lista de aplicativos mostrado en la figura 4.6 con un índice (N) a la izquierda del aplicativo.

Luego de registrar un aplicativo, existe la posibilidad de borrarlo, opción que aparece en la columna “Conexión” de esta pantalla; coloque atención al momento de eliminar un aplicativo del EITV DeveloperBox, pues sin realizar una pregunta de conformidad eliminará el aplicativo seleccionado sin poder recuperarlo de alguna otra manera.

4.6 Aplicaciones embebidas

En las aplicaciones embebidas es posible registrar aplicativos en el EITV DeveloperBox que serán reflejados en la pantalla mostrada en la figura 4.6 y que pueden ser ejecutados por el equipo. La pantalla correspondiente es exhibida en la figura 4.10.

Los aplicativos ya registrados están en la parte superior como se observa en la figura 4.10. Todos los campos que aparecen en esta pantalla deben ser rellenados para que el aplicativo pueda ser registrado en el EITV DeveloperBox. Los aplicativos listados aquí,

aparecen con un indicativo (E) Embebido, a la izquierda del nombre del aplicativo en la pantalla que se muestra en la figura 4.6. Si todos los campos no fuesen rellenos, un mensaje de error se muestra y el aplicativo no será exhibido.

Name	Application Type	Base Directory	Initial Class	File
teste	Ginga-J		AmorEternoAmorXlet	Remove

Add New Application

Name:

Application Type:

Base Directory:

Initial Class:

File: No file chosen

[Add](#)

Available space 15684 kB.

Figura 4.10 Registro de aplicaciones embebidas del set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Un ejemplo de cómo deben ser rellenos los campos para registrar un aplicativo en el EITV DeveloperBox se muestra a continuación, en la figura 4.11. El campo [Base Directory] es la dirección relativa del archivo .ncl principal dentro de la estructura del archivo contenedor zip.

Name	Application Type	Base Directory	Initial Class	File
teste	Ginga-J		AmorEternoAmorXlet	Remove

Add New Application

Name:

Application Type:

Base Directory:

Initial Class:

File:

[Add](#)

Available space 15656 kB.

Figura 4.11 Registro de una nueva aplicación en el set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

El campo “Directorio Base” al no ser rellenado, indica que el aplicativo se ejecutará desde el directorio raíz dentro de la estructura de directorios conformada en el archivo zip, seleccionado en el campo “File”. Finalmente, presionamos sobre el botón [Add], y el aplicativo queda registrado como lo muestra la figura 4.12.

The screenshot displays the 'Embedded applications on the set-top box' interface. At the top, it states 'Application "Aplicacion" added.' Below this is a table listing registered applications:

Name	Application Type	Base Directory	Initial Class	File
teste	Ginga-J		AmorEtemoAmorXlet	Remove
Aplicacion	Ginga-NCL		AplicacionSalir.ncl	Remove

Below the table is a form titled 'Add New Application' with the following fields:

- Name:
- Application Type:
- Base Directory:
- Initial Class:
- File: No file chosen

An 'Add' button is located below the form. At the bottom of the interface, it shows 'Available space 15441 kB.'

Figura 4.12 Aplicación registrada en el set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

4.7 Configuración de registro

El EITV DeveloperBox posee un cliente syslog que envía el log de aplicativos JAVA y el log de los objetos Lua dentro de los aplicativos NCL vía UDP. En esta opción es posible configurar el IP de la máquina que recibirá estos logs, de tal forma que desde la computadora de desarrollo podamos visualizar en depuración los mensajes durante la ejecución.

Aún si no se registra la dirección IP de la computadora que recibirá los mensajes vía UDP, este log siempre es visible en la opción del cliente web, durante la ejecución de cada aplicativo cargado en el EITV DeveloperBox, como podemos ver en la figura 4.13.

Luego de registrada la aplicación mediante este cliente web de nuestra computadora de desarrollo. Podemos pasar a ejecutar en una pantalla de un televisor con resolución HD y con la ayuda del control remoto del EITV DeveloperBox seleccionamos el aplicativo interactivo que hemos registrado, según cómo podemos ver en la figura 4.6.

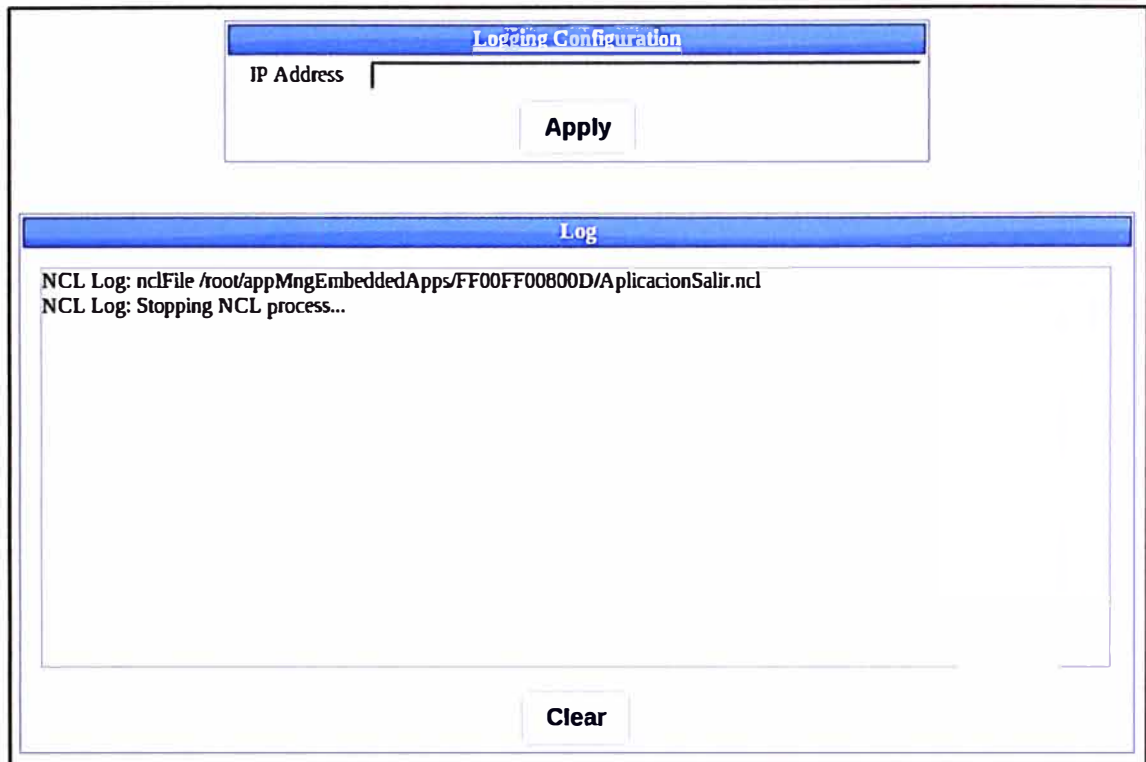


Figura 4.13 Log de ejecución de aplicaciones del set-top box EITV DeveloperBox

Fuente: EITV Entretenimiento e Interactividad para TV Digital

4.8 Empaquetamiento del aplicativo para ser registrado en el set-top box

Teniendo la estructura del aplicativo, tanto archivos ncl como archivos de media usado por el aplicativo en un directorio, la siguiente línea de comando, empaqueta el archivo app.zip necesario para ser registrado en el set-top box, así:

```
$ zip -R app.zip "*.ncl" "*.png"
```

4.9 Implementación y ejecución del aplicativo interactivo

4.9.1 Descripción del aplicativo

Este trabajo presenta un aplicativo interactivo demostrativo cuya intención es demostrar la ejecución de los conceptos revisados en este trabajo; aquí no se pretende decir que el aplicativo interactivo presentado es un producto final o profesional, el aplicativo demuestra como veremos, la veracidad de nuestro estudio y he ahí su trascendencia.

El demostrativo dispone de un menú de 4 botones laterales, uno de ellos, el denominado [Salir] el cual termina la aplicación, También, en la parte inferior, muestra dos cajas de texto, una destinada a la invocación de un logotipo, y la otra, exhibe un texto informativo que es manipulado visualmente mediante los botones laterales mencionados.

El demostrativo es ejecutado directamente sobre la señal de TDT, de tal manera que seguiremos viendo la señal en vivo con el redimensionamiento del campo visual en la pantalla, y el logotipo en la otra caja es obtenido mediante el canal de retorno.

Como se observa, este demostrativo no posee complejidad de menús u otros componentes, sin embargo, aquí se describen una secuencia de pasos a seguir durante su ejecución:

1. Mediante el control remoto del EITV DeveloperBox, se oprime la tecla [Menú]
2. Mediante las teclas direccionales elegimos la opción [Ginga] en pantalla, y se oprime la tecla del control remoto [OK]
3. Tal y como lo hemos visto anteriormente, estamos ante la figura 4.6, donde podemos ver el aplicativo Demo precedido de la letra [e], el cual es seleccionado también con las teclas direccionales, y finalmente ejecutado con la tecla [OK]
4. Se puede observar la aparición de la caja de texto inferior, la cual será manipulada visualmente mediante los botones laterales mostrados, se observa como el texto se visualiza en pantalla y a continuación seleccionamos un primer botón de la barra lateral que puede ser el botón [VERDE], se presiona la tecla [OK] del control remoto, tomando atención del aspecto visual de la caja de texto.
5. Se realiza un segundo desplazamiento con las teclas direccionales del control remoto hacia el botón lateral [ROJO] en pantalla, como se puede observar también, una línea azul en el borde del botón indica que botón estamos seleccionando con las teclas direccionales del control remoto; finalmente, presionamos la tecla [OK] del control remoto para ver el efecto visual que toma el texto de la caja inferior.
6. Un tercer y último movimiento hacia el botón [AMARILLO] podemos realizar y como ya hemos realizado anteriormente vemos el aspecto visual que toma la caja de texto, cuando presionamos la tecla [OK] del control remoto.
7. En todo momento hemos visto el logotipo de la otra caja de texto que muestra la imagen que por el canal de retorno de nuestro set-top box se está recibiendo, esta funcionalidad indica la comunicación con el mundo exterior y el acceso a datos mediante la interfaz Ethernet por parte del set-top box; para terminar, el botón [SALIR] redimensiona el campo visual a su tamaño original mostrando el mensaje Terminado! por parte del set-top box.

Lo aquí mostrado revela una secuencia de interactividad existente entre el espectador y su televisor mediante el set-top box, el efecto de las decisiones tomados por el espectador, se revelan visualmente en pantalla confirmadas al presionar los botones del control remoto, este es el concepto de interactividad que en el caso de la TDT se quiere mostrar, y mediante este aplicativo demostrativo hemos ejemplificado.

4.9.2 Implementación del código fuente

Ahora veremos por dentro del demostrativo, en el archivo demoFinal.ncl del proyecto, donde y como se realizan todos los comportamientos descritos anteriormente.

En la figura 4.14 podemos ver el punto de inicio del aplicativo, el puerto es quien indica que componente de media inicia el aplicativo, según se indica en la línea 40, el componente de video; para el caso de señal de tv real denominada TS (Transport Stream), la cual se recibe por el decodificador y se exhibe en la pantalla, la línea 41 es la indicada, y finalmente donde tenemos la capacidad de redimensionar el campo visual en el momento de ejecutar el aplicativo, podemos conseguir ello, manipulando las propiedades de esta media, según lo indica la línea 42.

```

39 <body id="myBodyID">
40   <port component="tv" id="entrada"/>
41   <media descriptor="dTv" id="tv" src="sbtvd-ts://0">
42     <property name="bounds"/>
43   </media>

```

Figura 4.14 Código NCL – Punto de Inicio del aplicativo

Fuente: Software NCL Composer

En la figura 4.15 se puede observar la distribución por tamaño que se ha dado a las regiones para todas las medias involucradas en el aplicativo; el ancho y el largo de la región Base que corresponde a toda la pantalla es el 100%, y cada componente de media ocupa un porcentaje del mismo. El redimensionamiento de la región de video se realiza al 80% en los atributos [height] y [width] según se indica en la línea 5, tanto las regiones de media de texto como botones están ubicados alrededor de la región de video lo cual es señalado por los atributos de posición [left] y [top] presentes en las líneas desde la 5 a la 12. El atributo [zindex] indica que todas las regiones están a la misma altura o distancia del espectador, esto tiene que ver con la profundidad de los objetos media contenidos en el aplicativo.

```

3 <head>
4   <regionBase id="regionBase">
5     <region height="80.00%" id="rgTv" left="0.00%" top="0.00%" width="80.00%" zIndex="0">
6     </region>
7     <region height="20.00%" id="rgT1" left="0.00%" top="80.00%" width="20.00%" zIndex="0"/>
8     <region height="20.00%" id="rgB1" left="80.00%" top="0.00%" width="20.00%" zIndex="0"/>
9     <region height="20.00%" id="rgB2" left="80.00%" top="20.00%" width="20.00%" zIndex="0"/>
10    <region height="20.00%" id="rgB3" left="80.00%" top="40.00%" width="20.00%" zIndex="0"/>
11    <region height="20.00%" id="rgB4" left="80.00%" top="60.00%" width="20.00%" zIndex="0"/>
12    <region height="20.00%" id="rgT2" left="20.00%" top="80.00%" width="80.00%" zIndex="0"/>
13  </regionBase>

```

Figura 4.15 Código NCL – Definición de regiones de media

Fuente: Software NCL Composer

En la figura 4.16, se observa que mediante los descriptores de media, podemos configurar el comportamiento de los objetos contenidos en el aplicativo, el atributo [focusIndex] indicado en las líneas desde la 17 a la 20, configura que objeto y en que secuencia toma el enfoque del aplicativo, los atributos [moveUp] y [moveDown] se corresponden con las teclas direccionales arriba y abajo del control remoto con los cuales navegamos entre los objetos a efectos de seleccionarlos, y que sigue el orden de enfoque establecido en focusIndex. Además, aquellos objetos que no han sido modificados en estos atributos como en las líneas 15, 16 y 17, no son seleccionables en el aplicativo.

```

14 <descriptorBase id="descriptorBase">
15   <descriptor id="dTv" region="rgTv"/>
16   <descriptor id="dT1" region="rgT1"/>
17   <descriptor focusIndex="1" id="dB1" moveDown="2" moveUp="4" region="rgB1"/>
18   <descriptor focusIndex="2" id="dB2" moveDown="3" moveUp="1" region="rgB2"/>
19   <descriptor focusIndex="3" id="dB3" moveDown="4" moveUp="2" region="rgB3"/>
20   <descriptor focusIndex="4" id="dB4" moveDown="1" moveUp="3" region="rgB4"/>
21   <descriptor id="dT2" region="rgT2"/>
22 </descriptorBase>

```

Figura 4.16 Código NCL –enfoco y desplazamiento de botones

Fuente: Software NCL Composer

La figura 4.17, describe el comportamiento de los conectores entre las medias; al ejecutar el aplicativo, como ya se ha mencionado, los enlaces existentes entre las medias tienen un comportamiento asociado (causalidad) que es indicado en los conectores causales establecidos en las líneas 24, 28 y 32. En el inicio, todas las medias son iniciadas mediante el conector causal [onBeginStart], todas las medias son terminadas según el conector causal [onSelectionStop], y los botones que sean seleccionados para realizar cambios en propiedades visuales de una media lo podrán hacer mediante el conector causal [onSelectionSetVar]. Esto será más comprensible en la figura 4.18, 4.19 y 4.20.

```

23 <connectorBase id="connectorBase">
24   <causalConnector id="onBeginStart">
25     <simpleCondition role="onBegin"/>
26     <simpleAction max="unbounded" qualifier="par" role="start"/>
27   </causalConnector>
28   <causalConnector id="onSelectionStop">
29     <simpleCondition role="onSelection"/>
30     <simpleAction max="unbounded" qualifier="par" role="stop"/>
31   </causalConnector>
32   <causalConnector id="onSelectionSetVar">
33     <connectorParam name="var"/>
34     <simpleCondition role="onSelection"/>
35     <simpleAction max="unbounded" qualifier="seq" role="set" value="$var"/>
36   </causalConnector>
37 </connectorBase>

```

Figura 4.17 Código NCL – Conectores causales

Fuente: Software NCL Composer

La figura 4.18 muestra el comportamiento propiamente dicho entre los objetos media en el momento inicial y final del aplicativo, descrito en los enlaces establecidos según las líneas 54 y 63. Para el enlace [linkInicio], se describe que cuando inicie el componente de TV en la línea 55, entonces se inicie también los componentes desde la línea 56 a la 61, o sea, todas las otras medias. Para el enlace [linkFin], se interpreta que cuando se seleccione la media, el componente b4, el cual es el botón [Salir], las medias entre las líneas 65 y 71 terminen. O sea, todos los otros objetos.

```

54      <link id="linkInicio" xconnector="onBeginStart">
55          <bind component="tv" role="onBegin"/>
56          <bind component="b1" role="start"/>
57          <bind component="b2" role="start"/>
58          <bind component="b3" role="start"/>
59          <bind component="b4" role="start"/>
60          <bind component="t1" role="start"/>
61          <bind component="t2" role="start"/>
62      </link>
63      <link id="linkFin" xconnector="onSelectionStop">
64          <bind component="b4" role="onSelection"/>
65          <bind component="tv" role="stop"/>
66          <bind component="b1" role="stop"/>
67          <bind component="b2" role="stop"/>
68          <bind component="b3" role="stop"/>
69          <bind component="b4" role="stop"/>
70          <bind component="t1" role="stop"/>
71          <bind component="t2" role="stop"/>
72      </link>

```

Figura 4.18 Código NCL - Inicio, término del aplicativo

Fuente: Software NCL Composer

La figura 4.19 muestra como al término del aplicativo se retorna al tamaño original el componente de video de la pantalla. Como se ve en la figura 4.15, el componente de video inicia con un tamaño del 80%, el enlace definido en la línea 73, toma el conector causal [onSelectionSetVar] para modificar la propiedad que define las dimensiones del componente de video [bounds] (posición y tamaño). Cuando se seleccione el botón [Salir] en la línea 74, entonces, la propiedad bounds del componente de tv en la línea 75, se dimensionen al 100% según lo indica las línea 76.

```

73      <link id="linkEndScreen" xconnector="onSelectionSetVar">
74          <bind component="b4" role="onSelection"/>
75          <bind component="tv" interface="bounds" role="set">
76              <bindParam name="var" value="0%,0%,100%,100%"/>
77          </bind>
78      </link>

```

Figura 4.19 Código NCL – Redimensionamiento del componente de video

Fuente: Software NCL Composer

La figura 4.20, describe la funcionalidad que tiene uno de los botones laterales de nuestro aplicativo cuando es seleccionado. Para el caso del enlace definido en la línea 79. El enlace [linkB1] correspondiente al botón [ROJO] en nuestra pantalla, que usa el conector causal [onSelectionSetVar], modificará la propiedad de otro componente de media al ser seleccionado [b1], según la línea 80. El componente a ser modificado en sus propiedades es el [t2] según lo indican las líneas 81 y 84, y como también se observa, las propiedades modificadas son [fontColor] y [fontSize] en las mismas líneas. Y los valores modificados son designados en las líneas 82 y 85. Donde se indica que el nuevo valor de color será el color rojo, y el nuevo valor de tamaño de fuente será el 16. Lo cual representa lo que finalmente ocurre al seleccionar el botón [ROJO] en nuestra pantalla.

```

79 <link id="linkB1" xconnector="onSelectionSetVar">
80 <bind component="b1" role="onSelection"/>
81 ⊖ <bind component="t2" interface="fontColor" role="set">
82 <bindParam name="var" value="red"/>
83 </bind>
84 ⊖ <bind component="t2" interface="fontSize" role="set">
85 <bindParam name="var" value="16"/>
86 </bind>
87 </link>

```

Figura 4. 20 Código NCL – Seleccionar botón

Fuente: Software NCL Composer

4.10 Simulación en la máquina virtual

La Figura 4.21 muestra la simulación del aplicativo demoFinal.ncl.

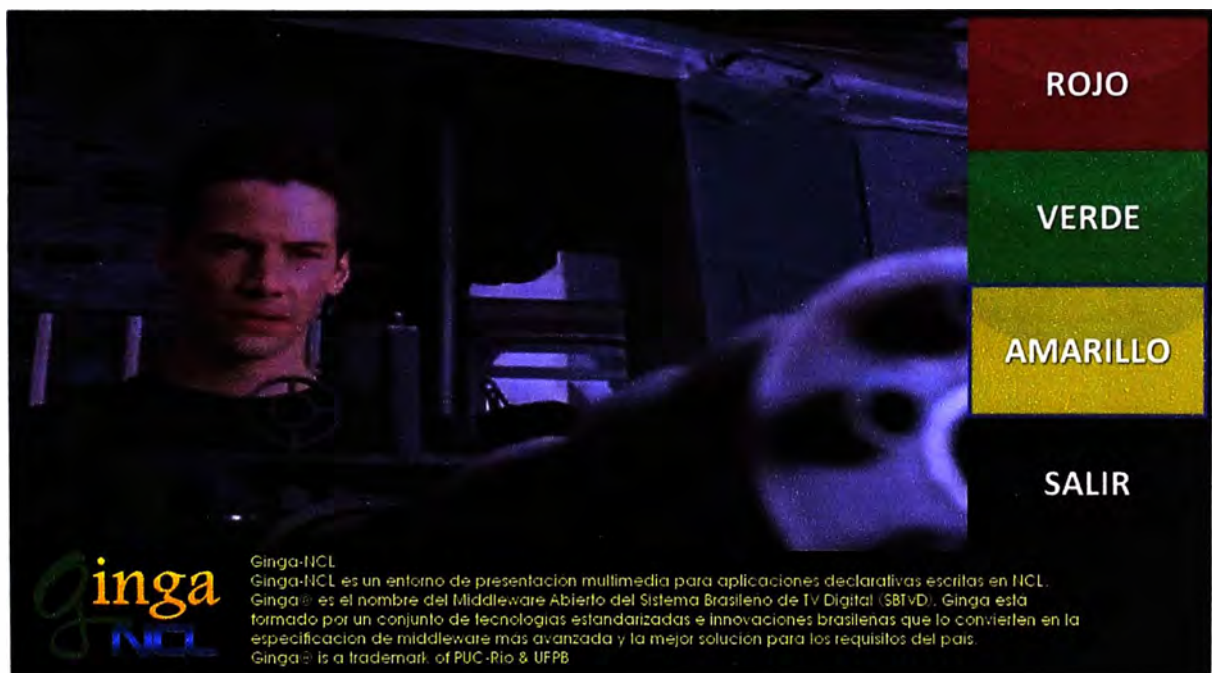


Figura 4.21 Simulación del aplicativo demoFinal.ncl

Fuente: Software VMware Player

Luego de revisados los conceptos, diseños, codificación y procedimientos implicados en la creación, pruebas y carga del aplicativo interactivo propuesto en este trabajo, es momento de ejecutarlo, y para esto según lo indicado en el procedimiento usamos el control remoto; se selecciona la opción [MENU] y se desplaza hasta la opción [Ginga] según se muestra en la figura 4.5, si todo es correcto en la lista de aplicaciones, según muestra la figura 4.6.

Podemos visualizar el aplicativo anticipado con la letra (E) llamado Demo, seleccionamos con el botón [OK], donde se muestra una pantalla como la vista en la figura 4.22 a continuación.



Figura 4.22 Aplicativo NCL ejecutado en el set-top box

Fuente: Cámara digital propia

Es visible como podemos usar el aplicativo Demo con señal real de TDT tal y como se ve en la Figura 4.22; para terminar, como ya se ha indicado oportunamente, el botón [SALIR] terminará la ejecución del aplicativo demostrativo.

Hay que mencionar que el televisor usado para este trabajo es un televisor LCD de 32" de resolución HD (1280 x 720) pixeles. Resolución que hemos usado en el diseño de la interfaz gráfica del aplicativo, así como también, fue la resolución elegida para la simulación en la máquina virtual gingavm.

4.11 Estimación de costos para la implementación de un Laboratorio de TV Digital

La tabla 4.4 muestra los costos estimados para la implementación del escenario descrito – de recepción – bajo el cual se desarrolla este trabajo, así como un presupuesto necesario para los proveedores de servicios de datos necesario.

Tabla 4.4 Equipos para el Laboratorio de Recepción de TV Digital

Fuente: Varias

Cantidad	Equipo	Marca	Modelo	Características	Costo
1	Stb	EITV	DeveloperBox	Soporte Ginga	S/.747
1	Tv	Sony	KDL-22EX355	LED 32" HD 2 Puertos HDMI 1 puerto USB	S/.1300
1	Antena	Miray	AA-M2	UHF, aérea Giratoria + cable	S/.80

También se muestra en la tablas 4.5 y 4.6, una estimación de servicios a proveedores de almacenamiento y datos, necesarias en aplicaciones interactivas que usan el canal de retorno.

Tabla 4.5 Estimación para un proveedor de almacenamiento

Fuente: AQPHost

Tipo	Almacenamiento	Transferencia mensual	Precio anual
Proveedor de almacenamiento	1500 MB	10 GB	S/.127.17

Tabla 4.6 Estimación para un proveedor de datos

Fuente: Claro

Tipo	Velocidad máxima	overbooking	Precio mensual
Proveedor de datos	2000 Kbps	200 Kbps	S/.115.00

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. El uso del middleware GINGA-NCL en la Televisión Digital Terrestre, facilita en gran medida el desarrollo de aplicaciones interactivas, no solo debido a las prestaciones que absorbe del lenguaje declarativo NCL, sino también por tener la condición de ser Software libre.
2. Comprobamos que el set-top box EITV DeveloperBox utilizado como decodificador digital, permite la proyección del video y sonido digital enviado por radiodifusión en señal digital que permite su reproducción en un televisor convencional “analógico”.
3. El set-top box EITV DeveloperBox utilizado en este trabajo, permite el uso de aplicaciones interactivas debido a que según las especificaciones del mismo cuenta con un middleware “Ginga” como parte de su sistema embebido que interpreta la comunicación de la aplicación interactiva y el sistema operativo del set-top box.
4. La naturaleza del lenguaje NCL, permite que desarrolladores sin experiencia previa, y con el conocimiento básico de lenguajes tales como el XML, puedan incursionar en nuevas tecnologías tales como la Televisión Digital Interactiva.
5. El Modelo de Contextos anidados (NCM), facilita al desarrollador, un punto de vista de abstracción, lo que permite no solo la comprensión y utilización de data convencional como son los videos, audios, gráficos y textos, sino que permite entender cómo interactúan entre ellos bajo la relación existente entre ellos en un escenario de interactividad, para el caso de la Televisión Digital.

RECOMENDACIONES

1. Este trabajo ha sido realizado íntegramente con Software libre, evitando así, los costos económicos relativos al copyright por el uso de las tecnologías y herramientas usadas en este trabajo. Se recomienda por tanto, motivar el desarrollo de televisión digital interactiva sin tener la necesidad de comprar algún producto de software que los proveedores de soluciones en esta área ofrecen.

2. La existencia de otras herramientas de desarrollo ofrece también ventajas – que no son expuestas aquí – por las cuales deben ser abordadas a partir de lo enunciado en este trabajo. La recomendación empieza por continuar investigando las potencialidades que estas otras herramientas de software libre ofrecen en el desarrollo de estas aplicaciones interactivas para la Televisión Digital.
3. Se recomienda que a nivel social se haga uso prioritario de esta reciente tecnología – que en el mediano plazo será parte de nuestra vida – en la atención de las necesidades básicas de la sociedad, y que no solo obedezca a la demanda de inversionistas en productos masivos, la interactividad a nivel de salud, educación e información pueden marcar un hito social para el País.
4. Al ser una tecnología novedosa en la actualidad, se recomienda a la Universidad la dirección de las investigaciones y desarrollo en esta área de la tecnología, sin necesidad de disponer de grandes inversiones, pueden realizarse trabajos respaldados por la Universidad integrando así que la comunidad de investigadores ofrezca productos tangibles, asesorando y realizando consultorías a las empresas televisoras de nuestro País.
5. Se propone el estudio técnico y económico que corresponda a la adquisición de soluciones tecnológicas que cierran todo el circuito que contempla la televisión digital. Promoviendo así, la implementación del primer canal de TV Digital terrestre universitario, panorama ya implementado en nuestros países vecinos, tal y como en el Brasil.
6. Se propone el desarrollo de aplicaciones interactivas usando Ginga-NCL para dispositivos móviles.

ANEXO A
EITV DeveloperBox

Característica	Especificação
Front-end (Tuner + Demodulador)	
Frequência de entrada	UHF: 470 MHz (CH14) a 806 MHz (CH69) VHF: 174 MHz (CH07) a 216 MHz (CH13)
Largura de banda	5,6 MHz
Nível de sinal	85 dBm a -20 dBm
Sinal	Compatível com o sistema ISDB-T
Impedância de entrada	75 Ohms (nominal)
Conexão de entrada/saída	Conector F
Unidade de processamento	
Processador	STi 7105 (CPU 450 MHz)
Memória RAM	256 MBytes
Memória Flash	128 MBytes
Decodificação de vídeo	
Padrão	Rec. ITU-T H.264 (MPEG-4 AVC)
Profile	HP@L4.0
Formatos	480i e 1080i
Frame rate	25, 30, 50 e 60 MHz
Frequência de vídeo	50 e 60 Hz
Decodificação de áudio	
Padrão	ISO/IEC 14496-3 (MPEG-4 AAC) (SBR)
Profile	AAC@L4 e HE-AAC@L4
Painel Frontal	
Botões	Troca de canais e liga/desliga (Standby)
	Sensor Infravermelho para controle remoto

Figura A.1 Especificaciones técnicas del Equipo EITV DeveloperBox I

Fuente: EITV Entretenimiento e Interactividad para TV Digital

Interfaces de comunicação¹	
USB	High Speed 2.0
Saída de Vídeo Componente	100 Mbps (RJ45)
Interfaces de comunicação¹	
Saída Digital de Áudio e Vídeo	HDMI
Saída de Vídeo Componente	YPbPr (Verde + Azul + Vermelho)
Saída de Áudio Estéreo 1 (Direito + Esquerdo)	2 RCA (Branco + Vermelho)
Saída de Vídeo Composto (CVBS – AV)	1 RCA (Amarelo)
Saída de Áudio Estéreo 2 (Direito + Esquerdo)	2 RCA (Branco + Amarelo)
Saída de Áudio Digital	SPDIF Coaxial (Laranja)
Fonte de energia	
Tensão de entrada	100 a 240 VAC, 50 a 60 Hz
Consumo	8 Watts
Dimensões	
Altura	37 mm
Largura	220 mm
Profundidade	187 mm
Peso	960 gramas

Figura A.2 Especificaciones técnicas del Equipo EITV DeveloperBox II

Fuente: EITV Entretenimiento e Interactividad para TV Digital

ANEXO B
GLOSARIO

API (Application Programming Interface): Interface de programación de aplicaciones, es una abstracción que describe una interface para la interacción con un conjunto de funciones utilizadas por los componentes del sistema de software.

Broadcasting: Es la distribución de contenidos de audio y video a una amplia audiencia vía radio, televisión u otro, a menudo medios de transmisión digital.

Códec: Es un dispositivo o programa de computador que tiene la capacidad de codificar y/o decodificar una señal o flujo de datos digital.

GINGA-J: Provee una infraestructura de ejecución de aplicaciones Java y extensiones específicamente adaptadas al ambiente de la TV.

GINGA-NCL: Es un entorno de presentación multimedia para aplicaciones declarativas escritas en NCL y su lenguaje de scripting Lua.

HDTV (High-Definition Television): Televisión en alta definición, se refiere al video que posee una resolución considerablemente más alta que la que se tiene en los sistemas de televisión tradicional.

SDTV (Standard-Definition Television): Televisión en definición estándar, este término se utiliza generalmente para referirse a la televisión digital, particularmente cuando la resolución de la transmisión es la misma o similar que la de los sistemas análogos.

MPEG-2 (Moving Pictures Experts Group): Es una extensión del estándar internacional MPEG-1 para la compresión digital de señales de audio y video.

MPEG-4: Compresión de audio AAC y video H.264.

NCL: Lenguaje de Contextos Anidados.

NCM: Modelo de contextos anidados.

Player: Cualquier programa, máquina o herramienta que permita reproducir sonido, animación o video.

TS: Flujo de transporte

XML: Lenguaje de marcas extensible

UDP: Protocolo de datagrama de Usuario.

ANEXO C
CÓDIGO FUENTE DEL APLICATIVO demoFinal.ncl


```

<ncl id="myNCLDocID">
  <head>
    <regionBase id="regionBase">
      <region height="80.00%" id="rgTv" left="0.00%"
top="0.00%" width="80.00%" zIndex="0">
        </region>
      <region height="20.00%" id="rgT1" left="0.00%"
top="80.00%" width="20.00%" zIndex="0">
        </region>
      <region height="20.00%" id="rgB1" left="80.00%"
top="0.00%" width="20.00%" zIndex="0">
        </region>
      <region height="20.00%" id="rgB2" left="80.00%"
top="20.00%" width="20.00%" zIndex="0">
        </region>
      <region height="20.00%" id="rgB3" left="80.00%"
top="40.00%" width="20.00%" zIndex="0">
        </region>
      <region height="20.00%" id="rgB4" left="80.00%"
top="60.00%" width="20.00%" zIndex="0">
        </region>
      <region height="20.00%" id="rgT2" left="20.00%"
top="80.00%" width="80.00%" zIndex="0">
        </region>
    </regionBase>
    <descriptorBase id="descriptorBase">
      <descriptor id="dTv" region="rgTv">
        </descriptor>
      <descriptor id="dT1" region="rgT1">
        </descriptor>
      <descriptor focusIndex="1" id="dB1" moveDown="2" mo-
veUp="4" region="rgB1">
        </descriptor>
      <descriptor focusIndex="2" id="dB2" moveDown="3" mo-
veUp="1" region="rgB2">
        </descriptor>
      <descriptor focusIndex="3" id="dB3" moveDown="4" mo-
veUp="2" region="rgB3">
        </descriptor>
      <descriptor focusIndex="4" id="dB4" moveDown="1" mo-
veUp="3" region="rgB4">
        </descriptor>
      <descriptor id="dT2" region="rgT2">
        </descriptor>
    </descriptorBase>
    <connectorBase id="connectorBase">
      <causalConnector id="onBeginStart">
        <simpleCondition role="onBegin">
          </simpleCondition>
          <simpleAction max="unbounded" qualifier="par"
role="start">
            </simpleAction>
          </causalConnector>
          <causalConnector id="onSelectionStop">
            <simpleCondition role="onSelection">
              </simpleCondition>
              <simpleAction max="unbounded" qualifier="par"
role="stop">
                </simpleAction>
              </causalConnector>

```

```

        <causalConnector id="onSelectionSetVar">
            <connectorParam name="var">
            </connectorParam>
            <simpleCondition role="onSelection">
            </simpleCondition>
            <simpleAction max="unbounded" qualifier="seq"
role="set" value="$var">
            </simpleAction>
        </causalConnector>
    </connectorBase>
</head>
<body id="myBodyID">
    <port component="tv" id="entrada">
    </port>
    <media descriptor="dTv" id="tv" src="sbtvd-ts://0">
        <property name="bounds">
        </property>
    </media>
    <media descriptor="dB1" id="b1" src="media/btnRed.png" ty-
pe="image/png">
    </media>
    <media descriptor="dB2" id="b2" src="media/btnGreen.png" ty-
pe="image/png">
    </media>
    <media descriptor="dB3" id="b3" src="media/btnYellow.png"
type="image/png">
    </media>
    <media descriptor="dB4" id="b4" src="media/btnBlack.png" ty-
pe="image/png">
    </media>
    <media descriptor="dT1" id="t1" src="media/textFig.html" ty-
pe="text/html">
    </media>
    <media descriptor="dT2" id="t2" src="media/text.txt"
type="text/plain">
        <property name="fontColor">
        </property>
        <property name="fontSize">
        </property>
    </media>
    <link id="linkInicio" xconnector="onBeginStart">
        <bind component="tv" role="onBegin">
        </bind>
        <bind component="b1" role="start">
        </bind>
        <bind component="b2" role="start">
        </bind>
        <bind component="b3" role="start">
        </bind>
        <bind component="b4" role="start">
        </bind>
        <bind component="t1" role="start">
        </bind>
        <bind component="t2" role="start">
        </bind>
    </link>
    <link id="linkFin" xconnector="onSelectionStop">
        <bind component="b4" role="onSelection">
        </bind>
        <bind component="tv" role="stop">

```

```

        </bind>
        <bind component="b1" role="stop">
        </bind>
        <bind component="b2" role="stop">
        </bind>
        <bind component="b3" role="stop">
        </bind>
        <bind component="b4" role="stop">
        </bind>
        <bind component="t1" role="stop">
        </bind>
        <bind component="t2" role="stop">
        </bind>
    </link>
    <link id="linkEndScreen" xconnector="onSelectionSetVar">
        <bind component="b4" role="onSelection">
        </bind>
        <bind component="tv" interface="bounds" role="set">
            <bindParam name="var" value="0%,0%,100%,100%">
            </bindParam>
        </bind>
    </link>
    <link id="linkB1" xconnector="onSelectionSetVar">
        <bind component="b1" role="onSelection">
        </bind>
        <bind component="t2" interface="fontColor" role="set">
            <bindParam name="var" value="red">
            </bindParam>
        </bind>
        <bind component="t2" interface="fontSize" role="set">
            <bindParam name="var" value="16">
            </bindParam>
        </bind>
    </link>
    <link id="linkB2" xconnector="onSelectionSetVar">
        <bind component="b2" role="onSelection">
        </bind>
        <bind component="t2" interface="fontColor" role="set">
            <bindParam name="var" value="green">
            </bindParam>
        </bind>
        <bind component="t2" interface="fontSize" role="set">
            <bindParam name="var" value="17">
            </bindParam>
        </bind>
    </link>
    <link id="linkB3" xconnector="onSelectionSetVar">
        <bind component="b3" role="onSelection">
        </bind>
        <bind component="t2" interface="fontColor" role="set">
            <bindParam name="var" value="yellow">
            </bindParam>
        </bind>
        <bind component="t2" interface="fontSize" role="set">
            <bindParam name="var" value="18">
            </bindParam>
        </bind>
    </link>
</body>
</ncl>

```

BIBLIOGRAFÍA

- [1]. Sergio Losa, Victor Pañeda, y José Alejandro Suárez, Carlos Bermejo, "La Televisión Digital Interactiva". Asturias. España. Universidad de Oviedo. 2011.
<http://156.35.151.9/~smi/5tm/09trabajos-sistemas/2/Memoria.pdf>
- [2]. Ministerio de Transportes y comunicaciones, "Informe anual de TDT en el Perú 2012" Lima Perú. 2012.
http://www.mtc.gob.pe/portal/comunicacion/concesion/radiodifusion/tv_digital/Informe%20anual%20de%20evaluacion%20prueba%20de%20extracto%202012.pdf
- [3]. Luiz Fernando Gomes Soares. "TV Interactiva se hace con Ginga". Fórum do sistema brasileiro de tv digital terrestre. 2009
http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/2009_04_soares.pdf
- [4]. Luiz Fernando Gomes Soares, Simone Diniz Junqueira Barbosa. "Programando em NCL 3.0". Rio de Janeiro. Brasil. Elsevier Editora Ltda. 2009. ISBN 978-85-352-3457-2.
<http://www.gingancl.org.br/pt-br/livrosecapitulosdelivros>
- [5]. Luis Fernando Gomes Soares, Simone Diniz Junqueira Barbosa. "TV digital interativa no Brasil se faz com Ginga. Fundamentos, Padrões, Auditoria Declarativa e Usabilidade". Río de Janeiro. Brasil. Departamento de Informática PUC-Rio, 2008.
<http://www.gingancl.org.br/pt-br/livrosecapitulosdelivros>
- [6]. Carlos de Salles Soares Neto, Luiz Fernando Gomes Soares, Rogério Ferreira Rodrigues, Simone Diniz Junqueira Barbosa, "Construindo Programas Audiovisuais Interativos Utilizando a NCL 3.0 e a Ferramenta Composer". Río de Janeiro. Brasil. Laboratorio de Sistemas Multimedia TeleMídia PUC-Rio, 2007.
<http://www.ncl.org.br/documentos/ TutorialNCL3.0-2ed.pdf>
- [7]. EITV Entretenimento e Interatividade para TV Digital, "EITV DeveloperBox - Manual do Usuário", Campinas-SP. Brasil. 2011
<http://www.eitv.com.br/>