



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE: INGENIERÍA EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA:

**DISEÑO DE UNA APLICACIÓN INTERACTIVA T-VOTING UTILIZANDO LA
PLATAFORMA GINGA/NCL PARA LA TELEVISIÓN DIGITAL**

AUTOR:

FREDDY OSWALDO MALLA MORA

TUTORES

ING. LUIS MONTOYA, MG.

ING. MAURO BOLAGAY, MG

Quito, Ecuador

2019

DECLARACIÓN

Yo Freddy Oswaldo Malla Mora, estudiante de la Universidad Tecnológica Israel de la carrera de Electrónica y Telecomunicaciones, declaro que el contenido en este Proyecto de Titulación requisito previo a la obtención del Grado de Ingeniería en Electrónica Digital y Telecomunicaciones, son originales, y de exclusiva responsabilidad legal y académica del autor.

FREDDY OSWALDO MALLA MORA

C.I.: 171811793-8

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR TÉCNICO

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación” **DISEÑO DE UNA APLICACIÓN INTERACTIVA T-VOTING UTILIZANDO LA PLATAFORMA GINGA/NCL PARA LA TELEVISIÓN DIGITAL**”, presentado por el Sr. Freddy Oswaldo Malla Mora, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se despliegue para su correspondiente estudio y calificación.

Quito D.M. 04 de junio del 2019

TUTOR TÉCNICO

Ing. Luis Montoya, Mg.

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR METODOLÓGICO

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación ” **DISEÑO DE UNA APLICACIÓN INTERACTIVA T-VOTING UTILIZANDO LA PLATAFORMA GINGA/NCL PARA LA TELEVISIÓN DIGITAL**”, presentado por el Sr. Freddy Oswaldo Malla Mora, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se despliegue para su correspondiente estudio y calificación.

Quito D.M. 04 de junio del 2019

TUTOR METODOLÓGICO

Ing. Mauro Bolagay, Mg.

AGRADECIMIENTOS

Primeramente, a Dios por darme la vida, la salud y las fuerzas necesarias para alcanzar mis sueños y poder culminar esta etapa de mi vida, a mis padres Isidro y Rosa y a mis hermanas que siempre estuvieron con su apoyo constante dándome el aliento moral.

A mi esposa Stefanía que día a día me ha acompañado y ha sabido alentarme en mis momentos de cansancio y al motivo fundamental de mi familia que es mi hijo Sebastián, él es el inicio de mis fuerzas cuando las mías acaban.

A mis tutores, Ing. Luis Montoya e Ing. Mauro Bolagay, quienes fueron la guía para que este proyecto sea una realidad.

DEDICATORIA

Esta dedicatoria es para plasmar mi eterno agradecimiento a mis padres que con sus palabras y ejemplos me han inculcado la humildad, la honestidad y la fortaleza para salir adelante y que ningún obstáculo es impedimento cuando se tiene la determinación, también a mis hermanas que son el apoyo cuando mis padres no están cerca de mí. A mi esposa por apoyarme y jamás dejar que desmaye y a mi hermoso hijo que con orgullo y felicidad acaricio cada día.

ÍNDICE GENERAL

Universidad Tecnológica Israel	ii
Aprobación del tutor técnico.....	ii
Aprobación del tutor metodológico	iii
Agradecimientos	iv
Dedicatoria.....	v
Índice general.....	vi
Índice de figuras.....	ix
Índice de tablas	xi
Resumen.....	1
Abstract.....	2
Introducción	3
Planteamiento del problema.....	4
Justificación	4
Objetivo general.....	4
Objetivos específicos	5
Alcance	5
Descripción de los capítulos	6
1 CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	7
1.1 Televisión digital terrestre	7
1.2 ISDB-T.....	8
1.2.1 Transmisión en ISDBT	9
1.2.2 Recepción.....	10
1.2.3 Estructura de ISDBT-T	11
1.2.4 Esquema del sistema de transmisión	11
1.2.5 Información técnica de ISDB-TB.	13
1.2.6 Parámetros del ISDB-T.....	14
1.2.7 ISDB-TB en el Ecuador.....	15

1.2.8 Bandas de frecuencia para televisión digital.....	16
1.2.9 Multiprogramación.	17
1.3 Ginga.....	18
1.3.1 Definición.	18
1.3.2 Set Top Box.	19
1.3.3 Midleware	19
1.3.4 Tipos de interacción.....	25
2 CAPÍTULO II: MARCO METODOLÓGICO	27
3 CAPÍTULO III: PROPUESTA.....	30
3.1 Descripción general del proyecto	30
3.1.1 NCL (Nested Context Language)	31
3.1.2 Aplicación en la <i>web</i>	34
3.1.3 Canal de retorno.....	35
3.2 Herramientas para el desarrollo	37
3.3 Funcionamiento del software.....	38
3.4 Diagrama del diseño	39
3.4.1 Diagrama de flujo	40
3.4.2 Diagrama de contexto	42
3.4.3 Presupuesto del proyecto	43
3.4.4 Ventajas de la aplicación	43
4 CAPITULO IV: IMPLEMENTACIÓN Y PRUEBAS	45
4.1 Desarrollo.....	45
4.2 Pruebas virtuales	50
4.3 Pruebas reales	51
CONCLUSIONES	53
RECOMENDACIONES.....	55
BIBLIOGRAFÍA	56
ANEXOS	57
ANEXO 1: INSTALACIÓN DE GINGA	57

ANEXO 2: INTALACIÓN DE ECLIPSE NCL.....	60
ANEXO 3: DIAGRAMA DE TIEMPO	64
ANEXO 4: DATASHEET SET TOP BOX.....	65
ANEXO 5: MANUAL DE USO DE LA APLICACIÓN.....	69
ANEXO 6: CÓDIGO DE LA APLICACIÓN INTERACTIVA	71

ÍNDICE DE FIGURAS

Figura 1.1. Cadena de transmisión de televisión digital terrestre	7
Figura 1.2. Distribución de canales, segmentos y programas ISDB-T	9
Figura 1.3. Estructura del sistema de radiodifusión digital	11
Figura 1.4. Esquema del sistema de transmisión	12
Figura 1.5. Flujo de transporte MPEG-2 TS.....	13
Figura 1.6. Set Top Box.....	19
Figura 1.7. Ginga Middleware	20
Figura 1.8. Componentes de Ginga common core.....	20
Figura 1.9. Subsistema Ginga-NCL.....	22
Figura 1.10. APIs de Ginga Java	25
Figura 1.11. Interacción local	25
Figura 1.12. Televisión interactiva remota	26
Figura 3.1. Estructura general NCL.....	31
Figura 3.2. Partes del documento NCL.....	31
Figura 3.3. Relación entre las entidades	33
Figura 3.4. Objetos media.....	34
Figura 3.5. Servidor Web.....	35
Figura 3.6. Televisión con canal de retorno.....	36
Figura 3.7. Funcionamiento tcp.lua	36
Figura 3.8. Selección de la resolución en la máquina virtual de Ginga.....	38
Figura 3.9. IP generada automáticamente en la máquina virtual.....	39
Figura 3.10. Esquema general de un proyecto en Eclipse NCL	39
Figura 3.11. IP generada automáticamente en la máquina virtual.....	40
Figura 3.12. Diagrama de flujo de la aplicación T-Voting.....	41
Figura 3.13. Diagrama de contexto general	42
Figura 3.14. Detalle de conectores.....	42
Figura 4.1. Teclas a usar del control remoto.....	45
Figura 4.2. Candidatos a la alcaldía de Quito	46
Figura 4.3. Botón de inicio de la aplicación	46
Figura 4.4. Botón de ingreso a la votación	47
Figura 4.5. Botón de salir de la aplicación	47
Figura 4.6. Barra de advertencia del contenido interactivo	47

Figura 4.7. Script acceso al servidor web remoto.....	48
Figura 4.8. Candidatos alcaldía de Quito.....	48
Figura 4.9. Mensaje de registro del voto	49
Figura 4.10. Archivos y base de datos cargados en el servidor web	49
Figura 4.11. Resultados visualizados a través de la web	50
Figura 4.12. Carga de la aplicación en GINGA.....	50
Figura 4.13. Ingreso al menú interactivo	50
Figura 4.14. Visualización de los candidatos	51
Figura 4.15. Registro de voto realizado por el televidente	51
Figura 4.16. Mensaje de registro del voto	51
Figura 4.17. Ingresar a votar y seleccionar a una de las 4 opciones	52
Figura 4.18. Escaneo con el celular del código QR y visualización de los resultados ..	52

ÍNDICE DE TABLAS

Tabla 1.1. Tasa útil de datos en 13 segmentos.....	10
Tabla 1.2. Parámetros ISDB-T	14
Tabla 1.3. Banda de frecuencias principales.....	16
Tabla 1.4. Canalización de frecuencias principales	17
Tabla 1.5. Multiprogramación	18
Tabla 2.1. Tesis de universidades relacionadas con Televisión Digital	27
Tabla 2.2. Tesis de análisis para el proyecto	28
Tabla 2.3. Curso de televisión digital interactiva	29
Tabla 3.1. Requisitos del computador para instalar Ginga.	37
Tabla 3.2. Requerimientos de software para emular Ginga.....	37
Tabla 4.1. Teclas de función.....	45

RESUMEN

El siguiente proyecto consiste en diseñar una aplicación interactiva *T-Voting*, utilizando como software libre Ginga/NCL para generar el código y como medio de prueba la televisión digital terrestre (TDT). El estándar ISDB-Tb (*Integrated Services Digital Broadcasting – Terrestrial Brasil*) brinda un camino tecnológico para poder interactuar con el televidente y genera nuevas formas de transmitir información.

Tiene un enfoque gubernamental (*T-Voting*) y tomando en consideración las elecciones de dignidades a la alcaldía de Quito, se usó como referencia a cuatro candidatos al azar para mostrar en pantalla. Las pruebas de funcionamiento se realizan cargando la aplicación en un equipo que cuenta con soporte total en interactividad en el estándar de televisión digital brasileño, denominado Set Top Box con Ginga modelo ET-SBX03.

La aplicación se puede visualizar en pantalla con un ícono de advertencia (informativo) y puede interactuar utilizando los botones de colores (rojo, verde, azul y amarillo) del control remoto del *smartbox*. Toda esta información es enviada a través de un canal de retorno (*Internet*) la cual es alojada en el hosting de un dominio determinado; estos datos son públicos y se visualizan en los teléfonos usando un lector de códigos QR o el *link* de la página.

Palabras claves: Televisión, aplicación, TDT, T-Voting, Ginga

ABSTRACT

The next project consists of designing an interactive T-Voting application, using as free software Ginga / NCL to generate the code and as a test medium digital terrestrial television (DTT). The ISDB-Tb (Integrated Services Digital Broadcasting - Terrestrial Brazil) standard provides a technological path to interact with the viewer and generates new ways of transmitting information.

It has a governmental focus (T-Voting) and taking into consideration the dignity elections to the mayor of Quito, it was used as a reference to four candidates at random to show on screen. The functional tests are carried out by loading the application in a team that has total support in interactivity in the Brazilian digital television standard, called Set Top Box with Ginga model ET-SBX03.

The application can be displayed on the screen with a warning icon (informative) and can interact using the colored buttons (red, green, blue and yellow) of the smartbox

Remote control. All this information is sent through a return channel (Internet) which is hosted in the hosting of a given domain; these data are public and are displayed on the phones using a QR code reader or the link on the page.

Keywords: Television, application, TDT, T-Voting, Ginga

INTRODUCCIÓN

Desde sus inicios, en el año 2007 cuando se empezó el estudio de los estándares: europeo DVB-T (Digital Video Broadcasting-Terrestrial), chino DTMB (*Digital Terrestrial Multimedia Broadcast*) y japonés con variaciones brasileñas ISDB-Tb (*Integrated Services Digital Broadcasting – Terrestrial Brasil*), se buscó el sistema que ofrezca las mayores prestaciones en robustez, calidad, nitidez, potencia e interactividad. Con el sistema norteamericano no se realizaron pruebas con equipos técnicos por falta de coordinación en aquella época.

En el 2010 bajo resolución N° 084-05-CONATEL-2010, el estándar que se decidió adoptar fue el japonés con modificaciones brasileñas, resaltando de los demás sistemas la televisión interactiva, que brinda la posibilidad de cambiar el escenario pasivo de los televidentes a activos desde la comunidad de los hogares, abriendo la posibilidad de personalizar los servicios brindados por la televisión digital. La inmersión de la tecnología interactiva brinda nuevas oportunidades de negocio en campos tales como la educación, gobierno, salud, comercio, etc.

En el documento presentado, se utiliza software Ginga (*software* para diseño de aplicaciones interactivas para la televisión digital terrestre) para el diseño de una aplicación interactiva *T-government* con canal de retorno unidireccional a través del internet y utilizando un lenguaje declarativo. Dentro de los objetivos esta diseñar una aplicación interactiva con una interfaz de manejo amigable e intuitiva y aprovechando la televisión digital terrestre para poder difundirla de forma abierta, fomentando la inclusión social con acceso a la información.

Planteamiento del problema

La principal función de las encuestadoras tales como Cedatos, Informe Confidencial, Blasco Peñaherrera & Asociados, entre otras; es presentar resultados de intención de voto de los encuestados, dar a conocer la opinión de la ciudadanía. La forma de evaluar en el caso de elecciones seccionales o presidenciales son encuestas manuales a través de “boca de urna”, para lo cual se requiere de personal con previa capacitación, personal para procesar esa información obtenida en las calles y equipos técnicos. Todo esto conlleva a inversión de tiempo y dinero.

Justificación

La televisión es un medio de comunicación masivo y de acceso libre, a través del cual se transmite una variada cantidad de información, de acuerdo al interés social y personal. En ese sentido el auge de la tecnología de televisión digital terrestre está revolucionando la forma de ver televisión con la inmersión de la televisión interactiva (por sus siglas en inglés *iTv*). En este proyecto se pretende que la *iTv* se masifique como alternativa de negocio aprovechando su capacidad de comunicación recíproca entre los involucrados.

Desde este punto de vista la información no solo implica leer o visualizar, sino también la posibilidad de que los televidentes puedan expresar su postura a través de una encuesta digital y basado en estos conceptos, se plantea como tema las elecciones a la alcaldía de Quito, en donde las encuestadoras pueden tener la facilidad de difundir la aplicación con contenido en el área gubernamental utilizando como medio de difusión la TDT, en la que la tele-audiencia realice la encuesta a través del televisor sin la necesidad de salir de la comunidad de sus hogares o trabajos y los resultados sean publicados en tiempo real a través del Internet. Es una alternativa de negocio que permite abrir un abanico de oportunidades, agilizar el tiempo de respuesta para procesar y mostrar resultados.

Objetivo general

Diseñar una aplicación interactiva *T-Voting* para cuatro candidatos a la alcaldía de Quito utilizando la plataforma GINGA/NCL para televisión digital terrestre (TDT).

Objetivos específicos

1. Definir un esquema de trabajo y parámetros que se usarán.
2. Investigar el funcionamiento de una trama MPEG-2 TS (*Moving Picture Experts Group 2 Transport Stream*) y los requisitos necesarios para enviar una señal multiplexada con audio y video codificados.
3. Diseñar una interfaz amigable para los usuarios, que les permita relacionarse de forma intuitiva.
4. Diseñar la aplicación interactiva para cuatro candidatos con canal de retorno que permita al televidente manifestar su inclinación por un candidato para la alcaldía de Quito a través del control remoto.
5. Mostrar en pantalla las opciones de los candidatos con los que el usuario podrá elegir a su favorito a través de los botones del control remoto.
6. Realizar la programación de la aplicación interactiva en lenguaje Eclipse NCL.
7. Configurar un canal de retorno a través de internet utilizando un software que sea compatible con NCL.
8. Registrar los votos en una base de datos plana, esta información el televidente podrá visualizar a través de un lector de código QR o ingresando a través de un navegador.
9. Visualizar los resultados a través un teléfono inteligente o a través de una página web.

Alcance

Con la investigación del proyecto *T-Voting* se diseñó una aplicación para cuatro candidatos para la televisión digital enfocada al sector gubernamental que permitió a cada ciudadano realizar la votación por los candidatos a la alcaldía de Quito y conocer los resultados de la encuesta en tiempo real mediante la lectura de un código QR

Las pruebas se realizaron de forma virtual con el desarrollo de la aplicación en Eclipse-NCL, para la forma visual su ejecución en el *middleware* GINGA y la presentación en tiempo real utilizando un STB (*Set Top Box*) proyectada a un televisor con señal de televisión digital.

DESCRIPCIÓN DE LOS CAPÍTULOS

El proyecto presentado se divide en cuatro capítulos:

En el capítulo 1 se detalla la información del marco teórico, información relevante de los temas que intervienen en el desarrollo, la definición de temas que son aportes al desarrollo de la aplicación como son televisión digital terrestre, la televisión digital en el Ecuador, aplicaciones interactivas en televisión digital y plataformas que se usan para el desarrollo de la aplicación como lo es GINGA NCL/LUA.

En el capítulo 2 se habla de la metodología empleada para la investigación del proyecto, las estrategias que se utilizan para cumplir los objetivos del proyecto y su alcance.

En el capítulo 3 se describe la propuesta que se planteó para el diseño de la aplicación, se analizan herramientas de hardware y software que se requieren para su desarrollo, diagramas generales que permiten aterrizar las ideas, los costos que implican llevar a cabo el proyecto y los tiempos para la cadena de actividades que se realizó.

En el capítulo 4 se registra y describe el producto final, las pruebas que se realizaron para obtener el diseño de la aplicación, la ubicación de cada uno de los botones, las imágenes de los candidatos, su ubicación en la pantalla del televisor y se prueba la comunicación entre la aplicación con la base de datos a través del canal de retorno.

1 CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1 Televisión digital terrestre

En la televisión analógica, los parámetros de la imagen y del sonido se representan por las magnitudes analógicas de señales eléctricas, es decir, su recorrido hasta los hogares ocupa un ancho de banda muy alto. En resumen se obtiene provecho de las señales de video que varían muy poco al pasar de un elemento de imagen o sonido a los contiguos (Millán R, 2008).

En código digital esos valores se sustituyen por números, 1 y 0. La conversión de la señal análoga se la realiza a través de un conversor analógico/digital. Esta representación en bits, da paso a procesos complejos de conversión, sin pérdida de calidad que brindan una serie de ventajas y abren un abanico de posibilidades de nuevos servicios en el hogar, una potencial fuente de difusión de noticias que los televidentes pueden consumir, como se indica en la figura 1.1.

Sin embargo, la señal digital entregada a la salida del conversor analógico a digital contiene un gran número de bits que demandan un alto consumo de medios como transporte y almacenaje. Por fortuna las señales de televisión contienen más datos de los requeridos por el ojo humano para visualizar de forma correcta las imágenes y los sonidos, y esta redundancia es aprovechada por los métodos de compresión digital, para minimizar la información producida en la digitalización hasta valores considerables que faciliten su transportación con una gran calidad y economía de recursos. Estas y una serie de métodos han facilitado el progreso de la televisión digital, permitiendo el almacenamiento y transporte de la señal de televisión con un mínimo uso de recursos (Millán R, 2008).

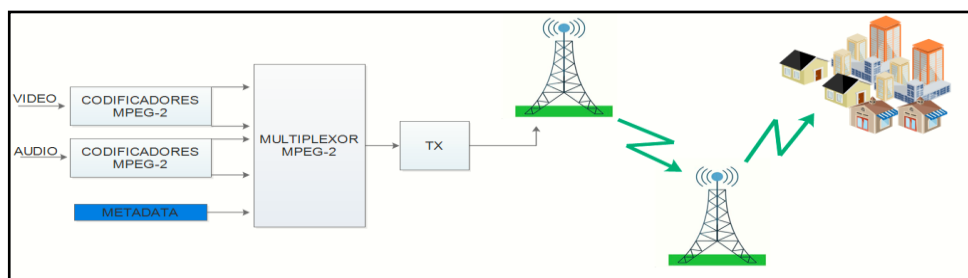


Figura 1.1. Cadena de transmisión de televisión digital terrestre

Fuente: Elaborado por el autor

1.2 ISDB-T

ISDB-Tb (Radiodifusión digital de servicios integrados terrestres de Brasil) es el estándar usado para la transmisión digital terrestre de Brasil. Se modificó el estándar digital ISDB-T TV de Japón y se adaptó como ISDB-Tb (radiodifusión digital de servicios integrados terrestres de Brasil). En el lado del transmisor, los cambios realizados en el estándar de TV digital de Japón hacen referencia a las plantillas que se utilizan para transmitir y a los canales de radiodifusión que se usaron en el mercado brasileño (International, s. f.).

En comparación con los demás estándares, ISDB-Tb abarca televisión, radio y servicios de datos. Como todos estos servicios se pueden transmitir de forma independiente y en muchas combinaciones distintas, es posible incluir HDTV (*High Definition Television*), SDTV (*Standard Definition Television*), audio y texto. A la par, los datos que se transmiten pueden llevar la misma programación con menor calidad y usar menor ancho de banda para los dispositivos móviles (International, s. f.).

Como en DVB-T, la modulación se hace usando un método de multiportadora con multiplicación codificada por división de frecuencias ortogonales (COFDM) en modo 2k, 4k y 8k. El canal de ancho de banda de 6 MHz se divide en 13 segmentos. Estos 13 segmentos del espectro OFDM (*Orthogonal Frequency Division Multiplexing*) se pueden combinar para crear grupos de segmentos que se denominan capas. Estas capas hacen posible la transmisión jerárquica con ISDB-Tb (International, s. f.).

La división de un canal de UHF (*Ultra High Frequency*) de 6MHz de ancho de banda para televisión digital es en 13 sub-bandas de 428 kHz cada uno, más un segmento de 430 kHz para evitar la interferencia de canal adyacente, como se indica en la figura 1.2.

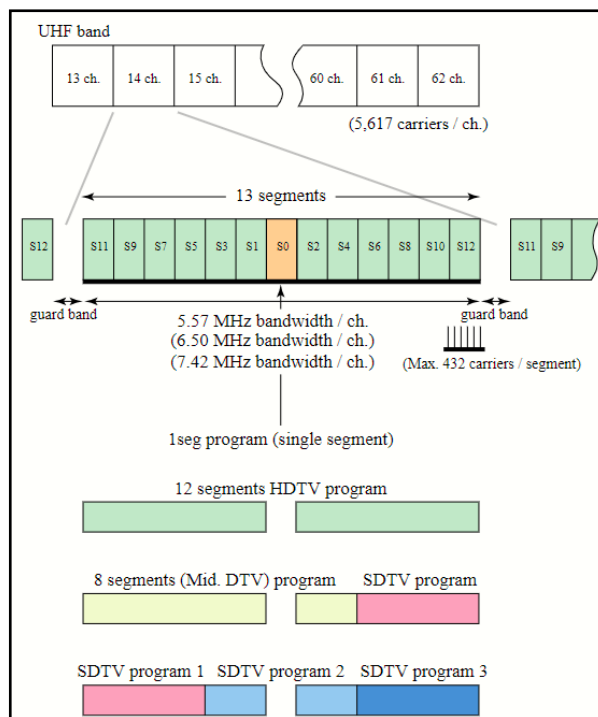


Figura 1.2. Distribución de canales, segmentos y programas ISDB-T
Fuente: (Namazu-tron,2008)

El segmento o sub-banda del centro, denominado S0 (Por segmento cero), se usa para transmitir la señal destinada a terminales móviles; a este sub-canal o segmento se lo denomina *one seg*, por ocupar sólo una sub-banda del canal. Para transmitir una mejor calidad en televisores de mejor resolución se utilizan más segmentos. En el ejemplo de la imagen 1.2, se muestra un sub-canal HD que ocupa 12 segmentos, más un canal de 1 segmento para transmitir TV a celulares. En la figura 1.2 se observan dos sub-canales en un solo canal; el primero en calidad HD de 720i, que requiere 8 segmentos y en calidad SD, se requiere 4 segmentos (YouBioit.com, 2012). Cada segmento puede requerir su esquema de transmisión (como: velocidades de codificación del código interno, profundidad del entrelazado temporal) y tipo de modulación. Los segmentos pueden ser combinados de la mejor forma para proporcionar un servicio de banda amplia. La transmisión jerárquica se obtiene transmitiendo grupos de segmentos OFDM (*Orthogonal Frequency Division Multiplexing*) con diferentes valores de transmisión (Urbina, Fuentes, Miranda, 2015).

1.2.1 Transmisión en ISDBT

El sistema ISDBT permite la transmisión de audio, video y datos que consisten en texto, diagramas, imágenes fijas y con movimiento, así como imágenes de alta calidad, a

dispositivos fijos como televisores y dispositivos móviles. Este estándar utiliza multiplexación codificada por división de frecuencias ortogonales (COFDM, por sus siglas en inglés). El sistema en su capa física es flexible debido a que puede operar en diferentes configuraciones, tiene corrección de errores, utiliza entrelazado (*interleaving*) en el dominio del tiempo y la frecuencia. El espectro de la radiodifusión de televisión digital debe consistir en 13 bloques OFDM sucesivos, con cada segmento ocupando $\frac{1}{4}$ del ancho de canal de televisión. Las señales de audio, video y datos generan flujos elementales (ES, siglas en inglés), en la siguiente fase se ordenan en forma de tramas de datos (PES, por sus siglas en inglés), y posteriormente se multiplexan los datos de cada señal en paquetes TS de 188 bytes. A continuación, se detalla en la tabla 1.1 los parámetros generales para los tipos de modulación en una transmisión digital terrestre.

Tabla 1.1. Tasa útil de datos en 13 segmentos

MODULACIÓN	CÓDIGO CONVOLUCIONAL	NÚMERO DE PORTADORAS (MODO 1/2/3)	TASA DE INFORMACIÓN (Mbps)			
			Tiempo de guarda (1/4)	Tiempo de guarda (1/8)	Tiempo de guarda (1/16)	Tiempo de guarda (1/32)
DQPSK QPSK	1/2	156/312/624	3.651	4.056	4.295	4.425
	2/3	208/216/832	4.868	5.409	5.727	5.900
	3/4	234/468/936	4.476	6.085	6.443	6.638
	5/6	260/520/1040	6.085	6.761	7.159	7.376
	7/8	273/546/1092	6.389	7.099	7.517	7.744
16QAM	1/2	312/624/1248	7.302	8.113	8.590	8.851
	2/3	416/832/1664	9.736	10.818	11.454	11.801
	3/4	468/936/1812	10.953	12.170	12.886	13.276
	5/6	520/1040/2080	12.170	13.522	14.318	14.752
	7/8	546/1092/2184	12.779	14.198	15.034	15.489
64QAM	1/2	468/936/1812	10.953	12.170	12.886	13.276
	2/3	624/1248/2496	14.604	16.277	17.181	17.702
	3/4	702/1404/2808	16.430	18.255	19.329	19.915
	5/6	780/1560/3120	18.255	20.284	21.477	22.128
	7/8	819/1638/3276	19.168	21.298	22.551	23.234

Fuente: (ARIB STD-B31 Version 1.6, 2005)

1.2.2 Recepción

La recepción se encarga de ejecutar los contenidos de audio, video, datos y mostrarlos en un televisor. Sus etapas son: el sintonizador encargado de captar la señal digital, el decodificador separa el contenido multimedia que llega en un flujo de transporte TS y en el televisor se visualiza la señal digital junto con el contenido interactivo.

Los receptores de un segmento (dispositivos móviles) o trece segmentos (dispositivos fijos) tienen que cumplir con las siguientes características:

- El nivel de señal tiene que ser de -20 dBm o más.
- El nivel de recepción de la antena tiene que ser de -77 dBm o menos.
- En los receptores *one seg* el nivel de entrada tiene que ser de -11 dB equivalente al ancho de banda medido en términos de potencia eléctrica.

1.2.3 Estructura de ISDBT-T

En general el sistema de transmisión digital está compuesto de 3 bloques como se indica en la figura 1.3, bloque de codificación de fuente, bloque de multiplicación, y bloque de codificación para la transmisión. En el esquema del sistema se tienen en consideración los contenidos de radiodifusión, la configuración del servicio de transmisión (recepción fija/móvil y recepción portátil), la estructura del sistema, las tecnologías empleadas y las especificaciones para la radiodifusión (Morales A, 2010).

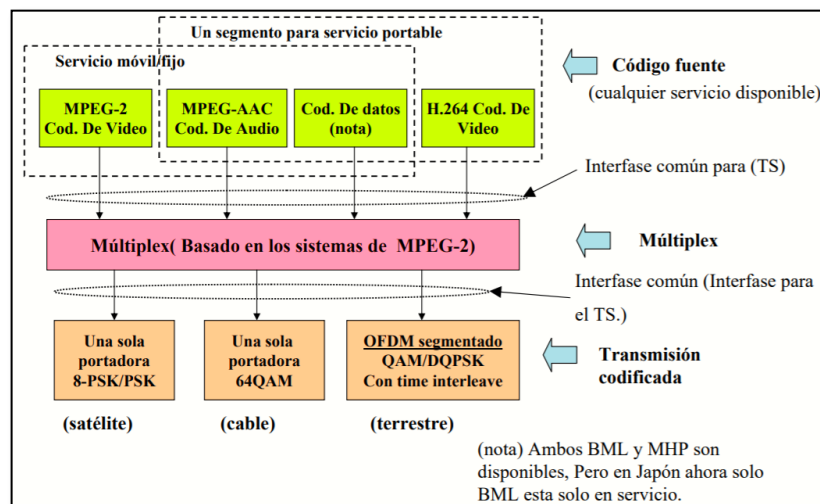


Figura 1.3. Estructura del sistema de radiodifusión digital

Fuente: (ISDB-T estándar japonés, 2006)

1.2.4 Esquema del sistema de transmisión

El sistema de transmisión, BST-OFDM (*Band Segmented Transmission Orthogonal Frequency Division Multiplexing*), configura una banda de transmisión compuesta por segmentos OFDM, cada uno tiene un ancho de banda de 6/14 MHz. Los parámetros de transmisión pueden establecer individualmente para cada segmento, logrando así una composición flexible del canal (Morales A, 2010).

Para conseguir una interfaz entre múltiples flujos de transporte MPEG-2 (*Moving Picture Experts Group 2*) y el sistema de transmisión BST-OFDM, estos flujos de transporte (TSs) son remultiplexados en un único TS como se indica en la figura 1.4. Adicional, la información acerca del control de transmisión como configuración del segmento del canal, parámetros de transmisión, etc. son enviados al receptor en forma de una señal TMCC (*Transmission Multiplexing Configuration Control*) (Morales A, 2010).

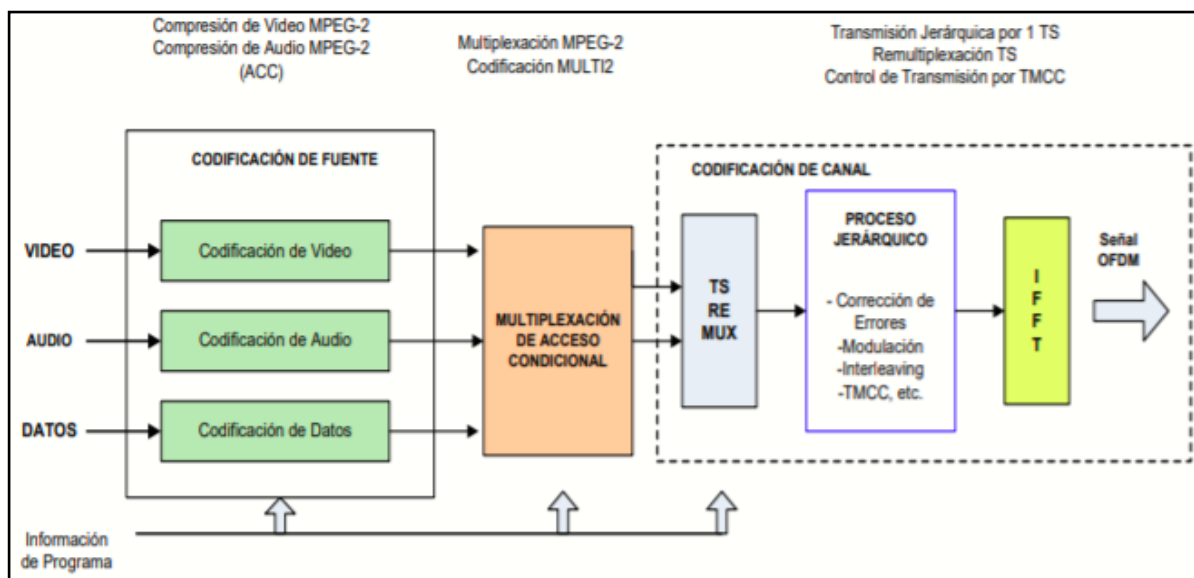


Figura 1.4. Esquema del sistema de transmisión

Fuente: (ISDB-T estándar japonés, 2006: 9)

Durante el proceso de envío de información, la señal se somete a las siguientes etapas:

- **Codificación y empaquetamiento:** para las señales analógicas de audio y video se requiere de equipos de codificación que cumplan con el estándar ISDBT. Para el video es H264/MPEG-2 y para el audio en MPEG-2. Estos requisitos pueden variar dependiendo del esquema que se utilice. Adicional se debe realizar el empaquetamiento y generación de flujos de transporte del audio y video codificado, donde los paquetes de audio y video se les asigna un PID (*Packet Identifier*) que es un identificador en notación hexadecimal (S. Gutiérrez, 2013).
- **Servidor de aplicaciones:** es el equipo que gestiona los contenidos, donde se alojan las aplicaciones interactivas y a través de un software genera el flujo de transporte de acuerdo al estándar MPEG-2, este servicio envía de forma aleatoria los paquetes.

- **Generación de tablas PSI/SI:** en el centro de emisión se debe contar con la posibilidad de generar y gestionar las tablas de información de programa (PSI) y las tablas de información de servicio (SI). Las tablas PSI/SI permiten enviar información adicional en el *Transport Stream* como: el nombre de la empresa que presta el servicio de televisión, la red de difusión, hora entre otros parámetros. La información que llevan los PSI/SI permiten al set top box sincronizar, reconstruir los servicios que se transportan y discriminar para proyectar al televisor (S. Gutiérrez, 2013).
- **Multiplexor:** es el equipo que permite organizar de forma secuencial el flujo del transporte MPEG-2 TS (*Moving Picture Experts Group 2 Transport Stream*) que se indica en la figura 1.5. A partir de la información de entrada que ya han sido procesados, en la salida del multiplexor se obtiene un flujo de transporte listo para modular y transmitir a través de RF (*Radio Frequency*) (S. Gutiérrez, 2013).

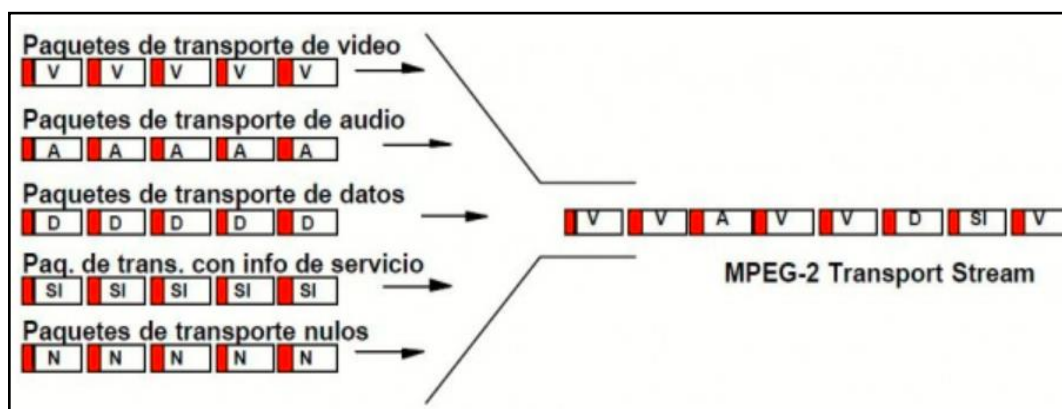


Figura 1.5. Flujo de transporte MPEG-2 TS

Fuente: (Ru Torres, 2009)

1.2.5 Información técnica de ISDB-Tb.

Los siguientes parámetros son necesarios conocer al momento de realizar una transmisión digital terrestre:

- **Modulación:** BST-OFMD (*Band Segmented Transmission-Orthogonal Frequency Division Multiplexing*).
- **Frecuencia:** VHF o UHF.
- **Arquitectura de transmisión:** 13 sub-bandas para receptores fijos y móviles y 1 segmento o sub-banda para terminales portables (celulares).

- **Cuadros por segundo:** 30 FPS (*Frame Per Second*) para servicio estándar y móvil, 30 FPS para servicio móvil (15 FPS en Japón).
- **Ancho de banda del canal:** 6 MHz en el caso de Ecuador, o dependiendo del país hasta 8 MHz.
- **Compresión de audio:** 5.1 MPEG-4 AAC@L4 para dispositivos fijos y móviles, MPEG-4 AAC@L2 para dispositivos portables.
- **Compresión de video:** MPEG-4 AVC HP@L4 para dispositivos fijos y móviles, MPEG-4 AVC BP@L1.3 para dispositivos portables.
- **Resolución video en dispositivos fijos:** SD 720x480i/p, SD720x576i/p, HD 1280x720p 16:9, Full HD 1920x1080i 16:9
- **Resolución video en disp. portables:** SQVGA (160x120 / 160x90), QVGA (320x240 / 320x180), CIF (352x288)
- **Multiplexado:** Sistema de multiplexado MPEG-2
- **Multiprograma:** El formato permite 1 programa Full HD por canal; o 1 programa HD+1 programa SD por canal; o 3 programas SD por canal (Estándar ISDB-T para Televisión Digital Terrestre - DivXLand.org, s. f.).

1.2.6 Parámetros del ISDB-T

En la siguiente tabla 1.2 se muestra los parámetros de operación del estándar ISDB-T para un ancho de banda de 6MHz:

Tabla 1.2. Parámetros ISDB-T

MODO ISDB-T	MODO 1 (2k)	MODO 2 (4k)	MODO 3 (8k)
Numero de segmentos en OFDM	13		
Ancho de banda útil	5.575 MHz	5.573 MHz	5.572 MHz
Espacio entre portadoras	3.968 kHz	1.984 kHz	0.992 kHz
Total de portadoras	1405	2809	4992
Modulación	QPSK, 16QAM, 64QAM, DQPSK		
Número de símbolos /cuadro	204		
Longitud del símbolo activo	252 us	504 us	1.008 ms
Longitud del intervalo de guarda	1/4, 1/8, 1/16, 1/32 de la duración del símbolo activo		
Código interior	Código de convolución (1/2, 2/3, 3/4, 5/6, 7/8)		
Código de salida	RS (204,188)		
Time interleave	0 ~ 0.5 s		
Velocidad de transferencia útil	3.651 Mbps ~ 23.234 Mbps		

Fuente: Elaborado por el Autor

1.2.7 ISDB-Tb en el Ecuador

De acuerdo a decreto ejecutivo 681 del 18 de octubre del 2007, se decretó a la SUPERTEL (Superintendencia de Telecomunicaciones) encargada de evaluar los estándares de Japón (ISDB-T) y la Comunidad Europea (DVB-T) con préstamos de equipos de transmisión. Lamentablemente por falta de coordinación no se llevó a cabo pruebas con equipos del estándar Norte Americano (ATSC).

De acuerdo a investigaciones realizadas por este organismo ecuatoriano como preferencias de los televidentes, los hábitos, uso y los estudios técnicos de cobertura, potencia de los equipos, calidad de señal, mayor número de canales e interacción de usuarios y operadores, se presentó el informe a la CONARTEL (Consejo Nacional de Radiodifusión y Televisión) en el 2010. Este organismo aceptó la recomendación del estándar japonés con variaciones brasileñas

Del informe presentado por la SUPERTEL, se obtienen las siguientes consideraciones:

- ISDB-T, presenta ventajas de rendimiento en condiciones de ruido impulsional y proporciona mayor flexibilidad en receptores móviles.
- ATSC, no es adecuado debido a su bajo rendimiento bajo condiciones de multitrayectoria y en receptores móviles.
- ISDB-T y DVB-T, tienen el mismo rendimiento en áreas bastante pobladas.
- En las pruebas también se concluyó que la reducción de potencia es de 10 a 1 en comparación con la televisión analógica, para cubrir áreas similares.

En el Ecuador el apagón analógico se realizará de forma progresiva desde el año 2020 hasta el año 2023 según lo programado por el MINTEL (Ministerio de Telecomunicaciones). Ha existido un retraso desde sus inicios en el 2010 debido a que más del 90% de la población no ha migrado sus equipos hacia los televisores digitales.

De acuerdo al plan del apagón analógico, este se llevaría a cabo por etapas, iniciando su despliegue en las principales ciudades como son Guayaquil, Quito y Cuenca; posteriormente en las capitales de cada provincia hasta finalizar en los lugares más alejados del país.

Actualmente existe un cronograma que se llevará a cabo desde mayo del 2020 hasta finales del 2023 de acuerdo a lo ordenado por el Ministerio de Telecomunicaciones. Pero desde el año 2012 algunos canales ya iniciaron su cambio de equipos para generar señal en HD, pero hasta el momento no han explotado al 100% lo que ofrece la televisión digital. La banda de frecuencia que se usará para el servicio de radiodifusión con emisiones de televisión es la UHF del espectro radioeléctrico.

Por otro lado, durante el período de *simulcast* (Transmisión simultánea), se utilizarán los canales adyacentes y los canales principales del servicio de TV Abierta Analógica, en la banda de canales 14, 15 y desde el 21 al 51, dependiendo de la disponibilidad existente. La asignación de canales virtuales será de acuerdo a la normativa correspondiente que se emita para el efecto. Los canales 14 y 15 serán considerados para la operación de la TDT en las zonas que la ARCOTEL determine (Alulema, 2012).

1.2.8 Bandas de frecuencia para televisión digital

Para ofrecer los servicios de televisión digital se establecen las siguientes bandas de frecuencias sugeridas en la tabla 1.3.

Tabla 1.3. Banda de frecuencias principales

UHF	
BANDA IV	de 470 a 482 MHz
	de 512 a 608 MHz
	de 614 a 644 MHz
BANDA V	de 644 a 698 MHz

Fuente: Elaborado por el Autor

Las bandas de frecuencias principales se dividen en 32 canales físicos de 6 MHz de ancho de banda cada uno, la frecuencia de la portadora central debe ser desplazada positivamente 1/7 MHz (142,857 kHz) con respecto a la frecuencia del centro, o también conocido como offset de frecuencia central del canal, de acuerdo a la tabla 1.4.

Tabla 1.4. Canalización de frecuencias principales

CANAL FISICO No.	FREC. INICIAL (MHz)	FREC. FINAL (MHz)	FREC. CENTRAL (MHz)
14	470	476	473 + 1/7
15	476	482	479 + 1/7
21	512	518	515 + 1/7
22	518	524	521 + 1/7
23	524	530	527 + 1/7
24	530	536	533 + 1/7
25	536	542	539 + 1/7
26	542	548	545 + 1/7
27	548	554	551 + 1/7
28	554	560	557 + 1/7
29	560	566	563 + 1/7
30	566	572	569 + 1/7
31	572	578	575 + 1/7
32	578	584	581 + 1/7
33	584	590	587 + 1/7
34	590	596	593 + 1/7
35	596	602	599 + 1/7
36	602	608	605 + 1/7
38	614	620	617 + 1/7
39	620	626	623 + 1/7
40	626	632	629 + 1/7
41	632	638	635 + 1/7
42	638	644	641 + 1/7
43	644	650	647 + 1/7
44	650	656	653 + 1/7
45	656	662	659 + 1/7
46	662	668	665 + 1/7
47	668	674	671 + 1/7
48	674	680	677 + 1/7
49	680	686	683 + 1/7
50	686	692	689 + 1/7
51	692	698	695 + 1/7

Fuente: Elaborado por el Autor

1.2.9 Multiprogramación.

En Ecuador la multiprogramación para televisión digital se encuentra regulada por la Agencia de Regulación y Control de las Telecomunicaciones y señala que los concesionarios que se hayan beneficiado de un canal lógico de 6MHz, pueden utilizarlo para la transmisión de la programación regular de acuerdo a los formatos del estándar ISDB-Tb, como se indica en la tabla 1.5.

Tabla 1.5. Multiprogramación

HDTV (1080i)				OS (<i>One Seg</i>)
HDTV (720P)		HDTV (720P)		OS (<i>One Seg</i>)
HDTV (720P)		SDTV	SDTV	OS (<i>One Seg</i>)
SDTV	SDTV	SDTV	SDTV	OS (<i>One Seg</i>)

Fuente: Elaborado por el Autor

ARCOTEL indica que la programación que se transmita por *one seg* deberá ser la misma que la que se transmita por SD o HD, y que este servicio de programación es obligatorio para los canales físicos.

1.3 Ginga

1.3.1 Definición.

Es el *middleware* abierto del SBTVD (Sistema Brasileño de Televisión Digital) conformado por un grupo de tecnologías estandarizadas e innovaciones desarrolladas en Brasil. Se divide en dos subsistemas interrelacionados que siguen dos paradigmas de programación diferente (Ginga, s. f.).

- GINGA-NCL (para aplicaciones declarativas NCL, *Nested Context Language*)
- GINGA-J (para aplicaciones con procedimientos Java).

Ginga-Brasil (ginga.org.br), indica que “La arquitectura Ginga facilita extensiones opcionales. Por ejemplo, el ambiente de ejecución Ginga-J, es el responsable de la ejecución de aplicaciones Java.

Los servicios de IPTV (*Internet Protocol Television*) específicos, tales como video bajo demanda, difusión de datos, etc., son ejemplos de otras posibles extensiones. Ginga ofrece servicios NCL (*Nested Context Language*) a todas las extensiones a través de una API (*Application Program Interfaces*) bien definida”.

1.3.2 Set Top Box.

El Set Top Box que se observa en la figura 1.6 es un decodificador que interviene en la recepción de la señal digital, es usado en los terminales de la recepción de la señal, como un dispositivo de acople en los televisores analógicos. Este dispositivo permite recibir los servicios interactivos que se transmiten en la señal BTS (*Broadcast Transport Stream*).

El dispositivo es el encargado de captar las señales digitales en los estándares de TV digital que hay en existencia, verifica que se tenga los permisos para abrir la señal y posteriormente la muestra en el televisor. Los decodificadores son una solución inmediata hasta que en los hogares se disponga de un televisor digital (García, 2013).



Figura 1.6. Set Top Box
Fuente: (EITV, 2017)

1.3.3 Middleware

Es una capa de software como se visualiza en la figura 1.7, cuya misión es facilitar el desarrollo y ejecución de aplicaciones interactivas en TV. Un middleware puede tener máquinas virtuales diferentes para manejar entornos de desarrollo diferentes (*HTML/JavaScript, Java, etc.*)

La máquina virtual describe unas especificaciones que hay que seguir para poder desarrollar y hacer uso de las funciones que brinda un Set-Top-Box, como, por ejemplo, dibujar en la pantalla del televisor, capturar las señales de un mando a distancia o utilizar el canal de retorno. Estas especificaciones también se llaman API (Interfaz Programador de Aplicaciones) (Torres, 2011).



Figura 1.7. Ginga Middleware

Fuente: (Comunidad Ginga Ecuador, 2011)

1.3.3.1 Arquitectura.

La arquitectura Ginga está dividida en tres grandes módulos:

- Ginga Common Core
- Ginga NCL (ambiente de presentación)
- Ginga Java (ambiente de ejecución)

1.3.3.2 Ginga common core.

El núcleo común de Ginga concentra servicios necesarios tanto para Motor de Presentación (declarativo) como para el Motor de Ejecución (de procedimiento). Este subsistema es la interfaz directa con el sistema operativo, haciendo un puente estrecho con el hardware. Aquí es donde se accede al sintonizador de canales, sistema de archivos, terminal gráfico, entre otros (Torres, 2011). Los componentes básicos del Núcleo Común (ver figura 1.8)

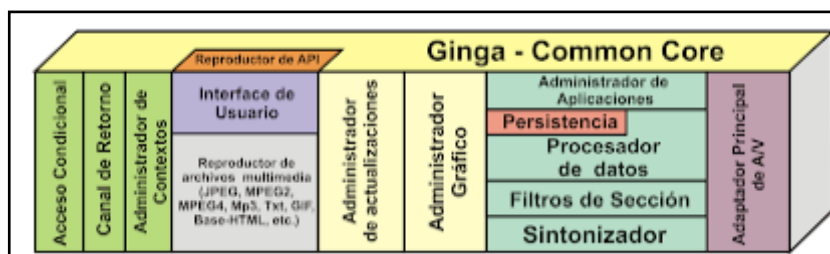


Figura 1.8. Componentes de Ginga common core

Fuente: (Comunidad Ginga Ecuador, 2011)

El sintonizador: es el módulo responsable de sintonizar un canal, seleccionando un canal físico y los flujos de transporte que están siendo enviados por este canal.

Filtros de selección: una vez sintonizado el canal, el *middleware* debe ser capaz de acceder a partes específicas del flujo de transporte. Para esto, existe un filtro de selección, el mismo que es capaz de buscar en el flujo, la parte exacta que las APIs necesitan para su ejecución, funcionando exactamente como un filtro, dejando pasar apenas la información requerida por la API.

Procesador de datos: es el elemento responsable de acceder, procesar y transferir los datos recibidos por la capa física. También es responsable de notificar a los otros componentes, sobre cualquier evento que se ha recibido.

Administrador de aplicaciones: es el módulo responsable de cargar, configurar, inicializar y ejecutar cualquier aplicación en cualquier entorno ya sea declarativo o de procedimiento. También es responsable de controlar el ciclo de vida de las aplicaciones, eliminarlas cuando sea necesario, además de controlar los recursos utilizados por esas APIs.

Adaptador principal de A/V: con el adaptador principal de A/V (audio y video), las aplicaciones consiguen ver el flujo de audio y vídeo. Esto es necesario cuando una aplicación necesita controlar sus acciones, de acuerdo con lo que se está transmitiendo.

Administrador de gráficos: las normas del *middleware* definen como se presentan al usuario las imágenes, videos, datos, etc., administrando las presentaciones de la misma manera que está definida en el estándar ARIB (*Association of Radio Industries and Businesses*) (ARIB B-24, 2004).

Administrador de actualizaciones: es el componente que gestiona las actualizaciones del sistema controlando, descargando las actualizaciones del *middleware* siempre que sea necesario, para corregir los errores encontrados en versiones anteriores. Esto debe ser hecho en tiempo de ejecución, sin perturbar el uso normal de la TV por parte del usuario.

Reproductor de archivos multimedia: son las herramientas necesarias para presentar los archivos multimedia recibidos, como por ejemplo archivos de tipo MPEG, JPEG, TXT, MP3, GIF, HTML, etc.

Interface de usuario: este módulo es responsable de captar e interpretar los eventos generados por los usuarios, tales como, comandos del control remoto y notificar a los otros módulos interesados.

Administrador de contextos: es el responsable de captar las preferencias del usuario, notificando a los otros componentes interesados esas preferencias. Esta información puede ser por ejemplo el horario en que el usuario mira la TV, o el bloquear y desbloquear canales, entre otros.

Canal de retorno: proporciona la interfaz de las capas superiores con el canal de interacción (o canal de retorno). Además, debe gestionar el canal de retorno de modo que los datos sean transmitidos cuando el canal esté disponible o forzar la transmisión en caso de que el usuario o una aplicación tengan definido un horario exacto.

Acceso condicional: este componente está encargado de restringir contenidos inapropiados recibidos por los canales de programación, proporcionando así seguridad para el middleware (Torres, 2011).

1.3.3.3 Ginga NCL.

La comunidad GINGA Ecuador indica que “El Ginga-NCL fue desarrollado por la Pontificia Universidad Católica de Rio de Janeiro – PUC- Rio, provee una infraestructura de presentación para aplicaciones declarativas escritas en el lenguaje NCL (Nested Context Language). NCL es una aplicación XML (eXtensible Markup Language) con facilidades para los aspectos de interactividad, sincronismo, espacio-temporal entre objetos de mídia, adaptabilidad, soporte a múltiples dispositivos y soporte a la producción de programas interactivos en vivo no-lineales”. Sus componentes se muestran en la figura 1.9.

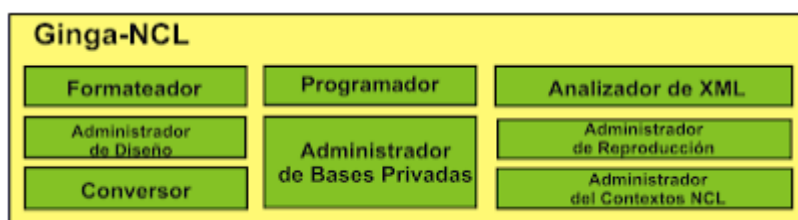


Figura 1.9. Subsistema Ginga-NCL
Fuente: (Comunidad Ginga Ecuador, 2011)

Formateador (*Formatter*): se encarga de recibir y controlar las aplicaciones multimedia escritas en NCL. Dichas aplicaciones son entregadas al Formateador por el Ginga-CC.

Analizador de XML (*XML Parser*), Convertidor (*Converter*): realizan la traducción de la aplicación NCL en la estructura interna de datos de Ginga-NCL para controlar la aplicación. Estos componentes son solicitados por el Formateador.

Programador (*Scheduler*): es iniciado para organizar el orden de la presentación del documento NCL (antes que inicie los objetos de media, se evalúan las condiciones de los enlaces y la programación correspondiente a las relaciones de las acciones que guiaran el flujo de la presentación). El componente Programador es responsable para dar la orden al componente Administrador de la Reproducción (*Player Manager*) para iniciar la reproducción apropiada del tipo de contenido de media para exhibirlo en el momento indicado.

Base Privada (*Private Base*): el Motor de Presentación (*Presentation Engine*) lidia con un conjunto de aplicaciones NCL que están dentro de una estructura conocida como Base Privada.

Administrador de la Base Privada (*Private Base Manager*): este componente está a cargo de recibir los comandos de edición de los documentos NCL y el darles mantenimiento a los documentos NCL presentados. Estos comandos de edición están divididos en tres subgrupos:

- 1er grupo de comandos, responsable por la activación y desactivación de una base privada, o sea, la habilitación de una determinada aplicación NCL.
- 2do grupo de comandos, responsable de iniciar, pausar, resumir, detener, remover las aplicaciones NCL.
- 3er grupo de comandos, responsable de la actualización de aplicaciones en tiempo real, permitiendo el agregar o remover elementos NCL y permite que se asignen valores a las propiedades de los objetos de media.

Administrador del diseño (*Layout Manager*): el Motor de Presentación soporta múltiples dispositivos de presentaciones a través del componente Administrador del Diseño,

el cual es responsable de mapear todas las regiones definidas en una aplicación NCL (Torres, 2011).

1.3.3.4 Ginga Java.

Como se ha mencionado, el ambiente imperativo Ginga-J ofrece soporte a las aplicaciones desarrolladas usando el lenguaje Java.

El Ginga-J es dividido en tres módulos, según ilustra la Figura 1.10: la máquina virtual Java; el núcleo y sus APIs, también llamadas APIs verde del Ginga-J; y el módulo responsable por el soporte a las APIs específicas del Ginga-J, llamadas de APIs amarilla y roja del Ginga-J. EL Ginga-J seguirá la especificación de la Norma ABNT NBR 15606-4 (Asociación Brasileña de Normas Técnicas). Las APIs verdes del núcleo son las responsables por mantener el sistema compatible lo máximo posible, con los sistemas americano y europeo.

Las APIs específicas del Ginga que pueden ser exportadas para otros sistemas son llamadas amarillas. Entre ellas están aquellas que dan soporte a múltiples usuarios, a múltiples dispositivos y a múltiples redes. También existen aquellas que ofrecen soporte a las aplicaciones que pueden ser recibidas, almacenadas y ejecutadas en un tiempo futuro.

El soporte para las necesidades específicas de aplicaciones para Brasil, en especial aplicaciones de inclusión social, son direccionadas por la API roja del Ginga-J. El Ginga-J tiene por base un conjunto de paquetes Java, comunes a diversos *middlewares* imperativos, según especificado en la Norma ABNT NBR 15606-4.

Entre las APIs específicas, sobresalen aquellas para la comunicación con el ambiente declarativo Ginga- NCL. Una aplicación Java puede actuar como entidad hija de una aplicación declarativa, o como una entidad inicial controlando el ciclo de vida de una entidad hija declarativa. Cuando la entidad Java es la entidad inicial, ella puede crear, modificar y destruir documentos declarativos NCL a través de las APIs de comandos de edición Ginga, según especificado en la Norma ABNT NBR 15606-2.

Cuando la entidad Java es una entidad hija, actúa como un objeto de media NCL, pudiendo ser registrada para recibir eventos NCL. Eventos NCL podrán, entonces, accionar métodos de las

clases Java del objeto. Objetos de media NCL con código imperativo Java pueden también comandar condiciones de disparos de relacionamientos NCL, usados en el sincronismo temporal y espacial de la presentación de contenidos. Pueden también manipular variables globales de aplicaciones declarativas, responsables por la determinación de la adaptación de contenidos o de la forma como los contenidos son presentados.

Las APIs amarillas son innovaciones brasileñas del ambiente imperativo Ginga-J, que lo distingue de los demás ambientes imperativos de los middlewares del sistema europeo y americano (Soares, s. f.).

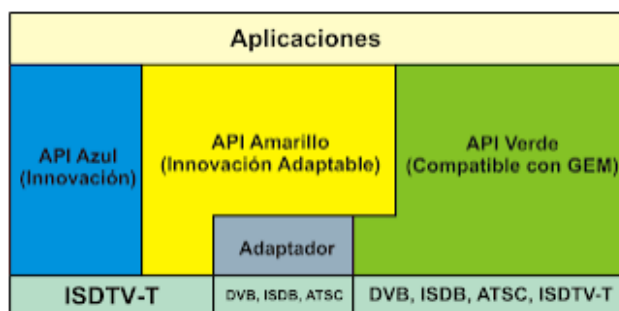


Figura 1.10. APIs de Ginga Java
Fuente: (Comunidad Ginga Ecuador, 2011)

1.3.4 Tipos de interacción.

1.3.4.1 Interacción local:

Estas aplicaciones son procesadas localmente, brindando la posibilidad de acceder a contenidos interactivos (ver figura 1.11), pero no pueden enviar datos o realizar consultas personalizadas usando el canal de retorno como el acceso a internet, que tendría un televidente en su hogar (Valencia, 2013)



Figura 1.11. Interacción local
Fuente: (Carlos García, 2013)

1.3.4.2 Interacción remota

Este tipo de aplicaciones brinda prestaciones más sofisticadas, como consulta personalizadas, enviar respuestas, votar, participar, etc. Para el correcto funcionamiento de este tipo de aplicaciones es necesario que el receptor tenga el soporte a un canal de retorno como se indica en la figura 1.12 (Valencia, 2013).



Figura 1.12. Televisión interactiva remota

Fuente: (Ronald Paucar,2010)

2 CAPÍTULO II: MARCO METODOLÓGICO

El proyecto planteado se lo divido en cuatro fases:

- Fase 1: Investigación
- Fase 2: Diseño, utilizando diagramas de flujo, diagrama esquemático y diagrama de casos
- Fase 3: Desarrollo de código
- Fase 4: Pruebas

En la fase 1, se utilizó la investigación bibliográfica - documental, que consiste en recopilar información de libros, artículos bibliográficos, tesis publicadas referente al tema de investigación. A continuación, se detallan los cuadros de información de universidades donde se ha desarrollado proyectos similares, en la tabla 2.1 se detalla la cantidad de documentos publicados por las universidades:

Tabla 2.1. Tesis de universidades relacionadas con Televisión Digital

UNIVERSIDADES	Coincidencias con "Televisión Digital"
Escuela Politécnica Nacional	7801
Escuela Politécnica del Ejercito	2417
Escuela Politécnica del Litoral	3579
Universidad de Cuenca	6281
Universidad del Azuay	304
Escuela Politécnica del Chimborazo	379
Universidad Técnica Particular de Loja	892
Universidad Politécnica Salesiana	10

Fuente: Elaborado por el autor

Recopilada la información necesaria se analizó la información utilizando el método deductivo, que consiste en analizar la información general adquirida y separar los temas de tesis y artículos publicados con temas similares al desarrollo de aplicaciones interactivas para la televisión digital terrestre como se indica en la tabla 2.2.

Tabla 2.2. Tesis de análisis para el proyecto

UNIVERSIDAD	TEMAS ANALIZADOS
Universidad de las Fuerzas Armadas	<i>Ginga studio</i> , software para la creación y edición de aplicaciones interactivas <i>Ginga-NCL</i> de televisión digital mediante una interfaz gráfica de usuario
	Diseño y desarrollo de una aplicación de contenidos interactivos para tv digital basada en el middleware <i>ginga</i> del sistema brasileño
Universidad Politécnica Salesiana	Diseño de una plataforma de software para televisión digital interactiva de un canal de deportes utilizando <i>Ginga-NCL Lua</i>
	Manejo del <i>software Ginga</i> para el desarrollo de aplicaciones interactivas para televisión digital, basado en el estándar Brasileño <i>ISDB-Tb</i>
Universidad de Cuenca	Implementación de un laboratorio de televisión digital que cumpla el estándar <i>ISD-TB</i>
	TV Digital fija utilizando Middleware <i>GINGA-NCL</i> aplicado a un noticiero digital II

Fuente: Elaborado por el autor

Con la información analizada de los diferentes temas donde intervienen los temas de televisión digital y aplicaciones interactivas, se aplicó el método analítico que consiste en extraer las partes relevantes de un todo y optimizar para el proyecto planteado. Se obtuvo información de la introducción de la televisión digital terrestre en el Ecuador y de algunas tesis de “Implementación de un Transmisor de Pruebas de TV Digital Terrestre ISDB-Tb para la Emisión de Aplicaciones Interactivas” desarrollada por el Ingeniero Diego Villamarin, que utilizan herramientas para la transmisión multiplexada de aplicaciones interactivas y artículos Técnicos como el del “Desarrollo de Aplicaciones Interactivas para TV Digital orientadas a informar a la Población en Desastres Naturales” publicada por el Ing. Bernal I. y Valencia J. que describe en su publicación la utilización de software GINGA – NCL, para la desarrollo de las aplicaciones interactivas.

Para tener mayor conocimiento de las herramientas que son necesarias para diseñar una aplicación, se tomó un curso particular de 40 horas en televisión digital y contenido interactivo el cual se indica en la tabla 2.3, con la ayuda de un experto en el tema.

Tabla 2.3. Curso de televisión digital interactiva

CURSO	DUARACIÓN	TUTOR
Diseño y Desarrollo de Aplicaciones Interactivas para TV Digital mediante el Middleware <i>GINGA NCL</i>	40 horas	Mg. Diego Villamarin

Fuente: Elaborado por el autor

En la fase 2, con los conocimientos necesarios se procedió a realizar un diagrama de flujo vertical para representar el proceso de planificación que se va a seguir para desarrollar la aplicación. Esto será sustentado con diagramas esquemáticos y diagramas de casos que se presentan en el capítulo 3.

La fase 3, comprende del desarrollo de la aplicación, se utilizará herramientas de *software* y *hardware* analizadas en el Capítulo 1, y utilizando un código que permita cumplir con los objetivos del proyecto, mencionados en la introducción.

En la fase 4, se realizó las pruebas de verificación del funcionamiento de la aplicación, poder determinar si las posiciones de las imágenes ocupan un espacio adecuado en el televisor. Las pruebas se realizaron utilizando un simulador para pruebas virtuales y pruebas reales cargando la aplicación al STB, esto será argumentado en el Capítulo 4 del proyecto.

3 CAPÍTULO III: PROPUESTA

3.1 Descripción general del proyecto

La televisión digital no comprende solo una mejor nitidez en la señal, o dar un mejor aprovechamiento del espectro radioeléctrico, comprende un nuevo campo de oportunidades que permitirá una mayor interactividad entre el usuario y su televisor. Si bien la inserción de la televisión digital inició en el 2010, hasta ahora no hay un canal de televisión que esté haciendo uso de aplicaciones como *T-voting*, *T-Learning*, *T-government*, *T-Health*, etc.; para explotar las ventajas que ofrece la televisión digital.

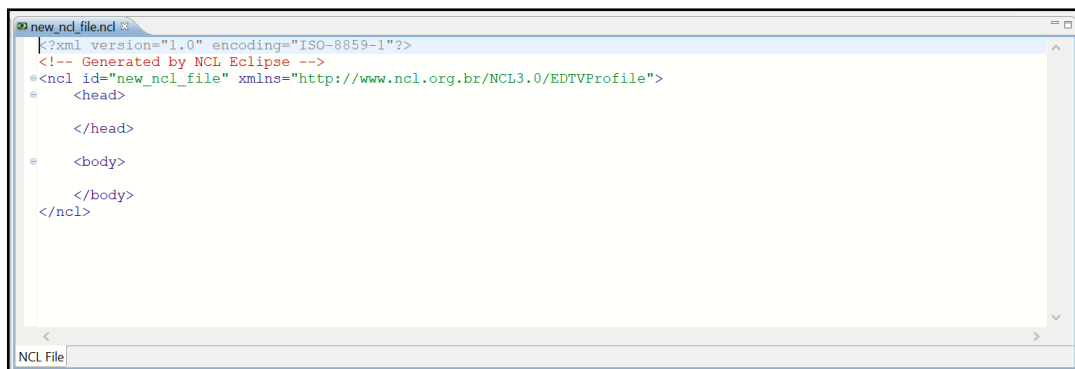
Las aplicaciones son un complemento a la información de la programación que maneja un medio de comunicación, y la función principal es poder brindar un servicio adicional al televidente y partiendo de este concepto se buscó una necesidad de poder suplir; que son las encuestas. Enfocando más el tema, el tipo de encuestas que se pretende dar un giro es el de encuestas electorales, pueden ser alcaldes o presidentes de una nación. Actualmente para realizar este tipo de encuestas que por lo general van preguntas generales y cerradas como: ¿Por qué candidato votaría en las elecciones a la alcaldía?, ¿Está de acuerdo con la administración del nuevo presidente de la república? o ¿Cuál es su partido político de su preferencia?, etc. La cantidad de tiempo y dinero que se invierten es bastante, porque no solo basta con realizar las encuestas manuales, hay que procesar la información para dar un resultado general de ser el caso.

El proyecto planteado se llama *T-Voting*, que tiene por objetivo general saber la opinión de la ciudadanía sobre cuatro candidatos a la alcaldía de Quito. ¿Cómo se lo va a realizar?, es a través del diseño de una aplicación interactiva que se ejecutará en el televisor de los hogares, cumpliendo este equipo con ciertas características de tecnología. La aplicación permitirá mostrar en pantalla una pregunta con los cuatro candidatos y el televidente puede votar por el de su preferencia y constatar los resultados ingresando a un servidor web con un link o utilizando un lector de códigos QR desde el teléfono celular.

Para poder diseñar el proyecto se ha identificado 3 partes fundamentales que la constituyen: NCL, aplicación en la *web* y Canal de retorno, los cuales se describen a continuación:

3.1.1 NCL (*Nested Context Language*)

Es donde se genera el código para la aplicación interactiva. Ofrece soporte de sincronismo, un canal de retorno, adaptación, presentación de contenidos y edición en vivo, es un lenguaje declarativo sencillo y ligero y lo más importante es *open source*, esto es un gran beneficio para el desarrollador. La estructura general de un documento NCL es como se visualiza en la figura 3.1.



```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="new_ncl_file" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
  </head>
  <body>
  </body>
</ncl>

```

Figura 3.1. Estructura general NCL
Fuente: Elaborado por el Freddy Malla

Su estructura se encuentra conformada por la cabecera y el cuerpo distribuida como se muestra en la figura 3.2.



Figura 3.2. Partes del documento NCL
Fuente: (D. Villamarin,2012)

A continuación, se describe cada una de sus partes:

1. **Región:** Es el lugar donde se muestra un elemento multimedia, representa una región de la pantalla. Su tamaño es definido en pixeles o porcentajes. Sus atributos son:

- a. **Id:** Identificador de región.
- b. **Height:** Altura de la región.
- c. **Width:** Ancho de la región.
- d. **Left, right, top, bottom:** Indica posición y dimension de la region.
- e. **Zindex:** Especifica la precedencia de superposición de la región.

Ejemplo:

```
<regionBase>
<region id=" rg1" height="100%" width="100%" zindex="1"/>
<region id="rg2" width="20%" height="20%" left="10%" right="10%"
zIndex="2"/>
<region id="rg3" width="20%" height="20%" left="10%" top="30%"
zIndex="2"/>
```

2. **Descriptor:** Especifica las propiedades de la presentación de un objeto multimedia. Sus atributos son:

- a. **Id:** Identificador del descriptor.
- b. **Región:** Identifica la región asociada.
- c. **explicitDur:** Define la duración del objeto multimedia osciado.

Ejemplo:

```
<descriptorBase>
<descriptor id="desc1" region="rg1"/>
<descriptor id="desc2" region="rg2" transIn="tran1" explicitDur="10s"
transOut="tran1">
<descriptorParam name="transparency" value="60%"/>
</descriptor>
<descriptor id="desc3" region="rg3">
<descriptorParam name="fontSize" value="20"/>
</descriptor>
</descriptorBase>
```

3. **Elemento multimedia:** Son el contenido multimedia como sonido, video, texto, imagen, *html* y *lua*. Sus atributos son:

- a. Id: Identificador del elemento multimedia.
- b. Descriptor: Identifica el descriptor asociado.
- c. Src: *Path* donde se ubica el elemento multimedia.
- d. *Type*: Define el tipo de objeto multimedia, de uso opcional.

Ejemplo:

```
<media id="video" src="media/video.avi" descriptor="desc1"/>
<media id="img" src="media/imagen.png" descriptor="desc2"/>
<media id="texto" src="media/texto.txt" descriptor="desc3"/>
```

4. **Puerta:** Para presentar elementos multimedia en la pantalla, para entrar a la interactividad.

Ejemplo:

```
<port id="puerta1" component="video"/>
<port id="p2" component="imagen"/>
<port id="p3" component="texto"/>
```

La estructura del código de la aplicación se lo define en 5 preguntas resumidas como se indica en la figura 3.3.



Figura 3.3. Relación entre las entidades

Fuente: (Diego Villamarin, 2012)

Los elementos que se van a mostrar con el desarrollo de una aplicación pueden ser los que se muestran en la figura 3.4.

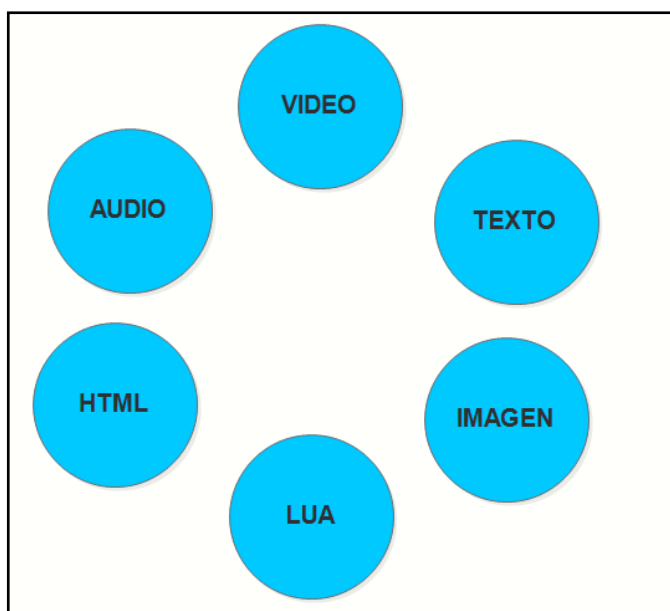


Figura 3.4. Objetos media
Fuente: Elaborado por el autor

3.1.2 Aplicación en la *web*

Definido así por su conformación entre el servidor web con su respectiva base de datos incorporada, que tiene tres categorías de acceso como son: el módulo de administración, visualización y de registro. El servidor web que se indica en la figura 3.5, permite alojar la base de datos y los archivos del programa, la base de datos que se sube al *hosting* (alojamiento *web*) es con extensión *php*.

Sus funciones que va a realizar respecto a la aplicación de diseño son:

- Recibir los datos generados por los televidentes.
- Facilitar la visualización de los datos en texto plano.
- Modificar la base de datos con su herramienta *php*.

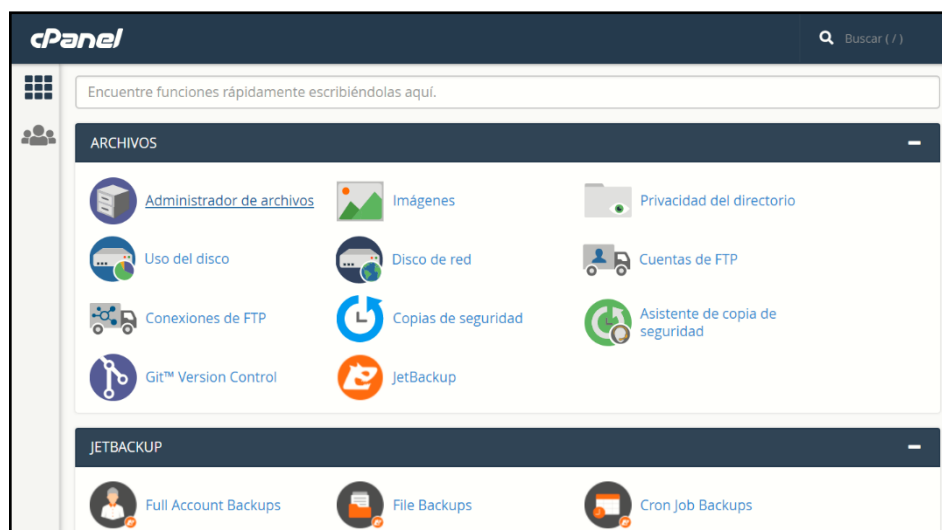


Figura 3.5. Servidor Web
Fuente: Elaborado por el autor

3.1.3 Canal de retorno

El canal de retorno es el medio por donde el televidente realizará la interacción con el proveedor de servicios de interactividad, como se visualiza en la figura 3.6.

En un canal interactivo su funcionalidad es llevar el audio, video debidamente codificados y multiplexados con los datos, inclusive la aplicación interactiva, de esta manera generar un solo flujo para luego ser modulado y posteriormente transmitido, luego de su transmisión estas señales son receptadas y *demultiplexadas* en el set top box o en el televisor si este ya viene integrado en el mismo.

Las tecnologías de acceso respecto al canal de retorno se indican en la norma brasileña ABNT NBR 15607-1 Las tecnologías definidas son: *módems* discados, *Ethernet*, ISDN (*Integrated Services Digital Network*), GSM (*Global System for Mobile*), CDMA (*Code Division Multiple Access*), *WiMAX* (*Worldwide Interoperability for Microwave Access*) y Wi-Fi.

Las aplicaciones interactivas con canal de retorno pueden ser unidireccionales o bidireccionales, como se indicó anteriormente la aplicación es bidireccional, el televidente envía una calificación y puede obtener los resultados de su voto.

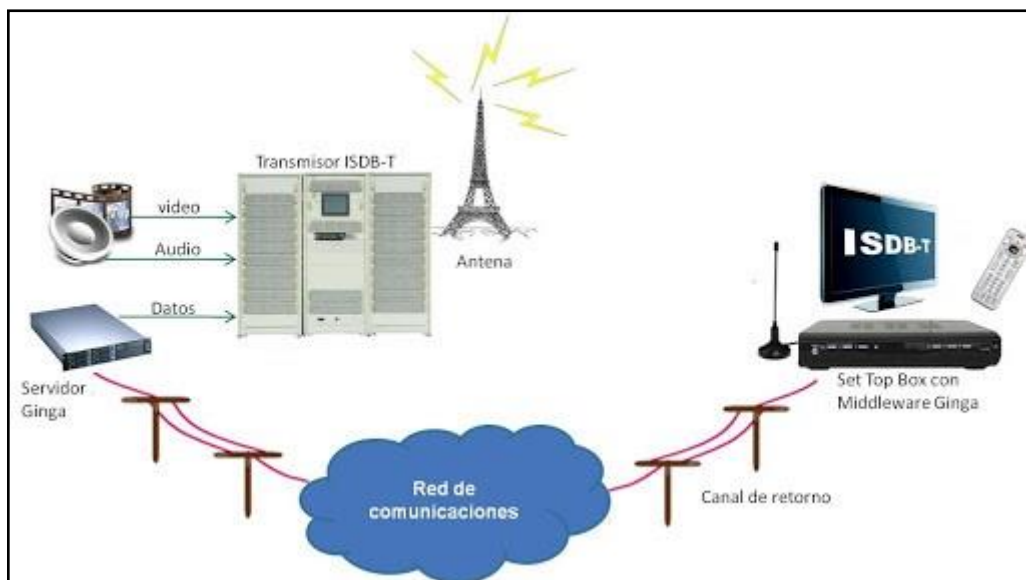


Figura 3.6. Televisión con canal de retorno

Fuente: (Ronald Paucar,2010)

La librería tcp.lua es un lenguaje imperativo usado para el desarrollo de aplicaciones con interacción remota que especifica cómo hacerlo, tiene un mayor poder de expresión, es un lenguaje de más bajo nivel, y más complejo de ser entendido. Desde 1993 se lo ha usado para el desarrollo de juegos.

A través de la codificación de la librería tcp.lua, se genera el enlace para la comunicación entre la aplicación y servidor web, donde se encuentran alojada la base de datos para el canal de retorno de la *app*. El medio de comunicación que se utiliza es el internet y la transferencia de información es unidireccional utilizando el puerto 80 como se indica en la figura 3.7.

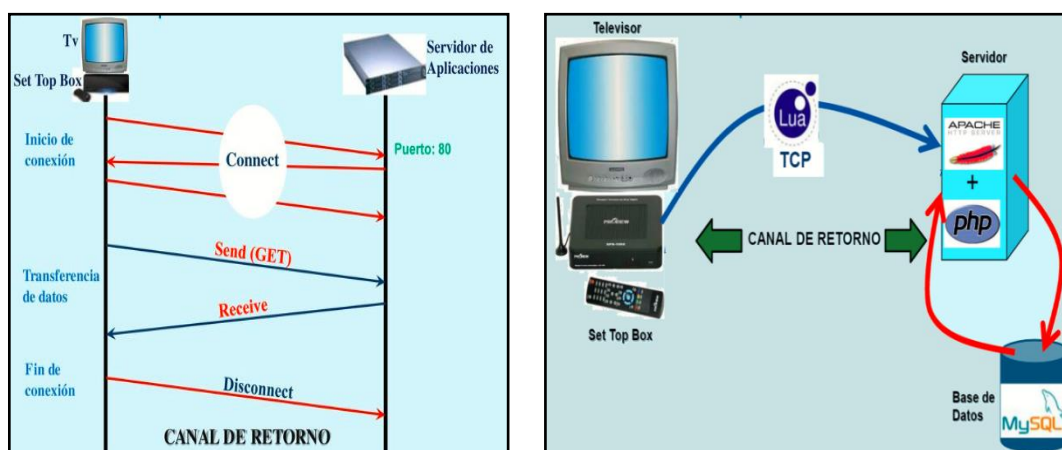


Figura 3.7. Funcionamiento tcp.lua

Fuente: (Ronald Paucar,2010)

3.2 Herramientas para el desarrollo

Para el desarrollo del lenguaje se ha utilizado herramientas que hayan sido probadas anteriormente como son Eclipse con sus respectivas librerías NCL instaladas y máquina de *Ginga* preinstalada, para que permita el enlace entre estos dos *softwares*.

El software utilizado para el desarrollo de la aplicación es el *Ginga* orientado al desarrollo de aplicaciones interactivas para televisión digital terrestre, permite el desarrollo de aplicaciones declarativas como imperativas, son dos ambientes que funcionan a la par, para el lenguaje declarativo NCL y para el lenguaje imperativo LUA.

A simple análisis hay que tener un conocimiento básico de cómo empezar a configurar y que es necesario para el desarrollo, por ello viendo esta necesidad se tomó un curso de 40 horas orientado a la creación de aplicaciones y los requerimientos básicos para el desarrollo de una app interactiva, de las clases recibidas se plantea los siguientes requisitos de hardware donde correrá *Ginga*, que se indica en las tablas 3.1 y 3.2.

Tabla 3.1. Requisitos del computador para instalar *Ginga*.

REQUERIMIENTOS DE HARDWARE	
PC	windows 7
Arquitectura	32 o 64 bits
RAM	min. 512 MB

Fuente: Elaborado por el autor

Tabla 3.2. Requerimientos de *software* para emular *Ginga*

REQUERIMIENTOS DE SOFTWARE
VMWare Player 3.1 o superior
Máquina Virtual de GINGA
JRE 1.6 o superior
Eclipse 3.2 o superior
NCL Eclipse

Fuente: Elaborado por el autor

3.3 Funcionamiento del software

La máquina virtual *Ginga* es cargada en el virtualizador *VMware Workstation* y es la que permitirá visualizar los videos que se creen en el software Eclipse, estas dos herramientas de software funcionan en sincronismo para lo cual *Eclipse-NCL* se comunica con *Ginga* a través de una IP que por defecto entrega la máquina virtual.

El software Eclipse requiere de librerías afines al propósito que se tiene del proyecto, la librería que se usará es *Eclipse NCL* que permite utilizar un lenguaje robusto y fácil de usar.

- **Instalación de la máquina virtual de GINGA:**

En el anexo 1 se adjunta los pasos a seguir para la instalación en el virtualizador de *Ginga*. A continuación, se mostrará solo la resolución que se va a utilizar para mostrar los videos con la aplicación que se vaya desarrollando (ver figura 3.8).



Figura 3.8. Selección de la resolución en la máquina virtual de GINGA

Fuente: Elaborado por el autor

En la interfaz que se muestra en la figura 3.9, se puede apreciar una IP, que se asocia con el programa de desarrollo del código que se genera en *Eclipse NCL*.

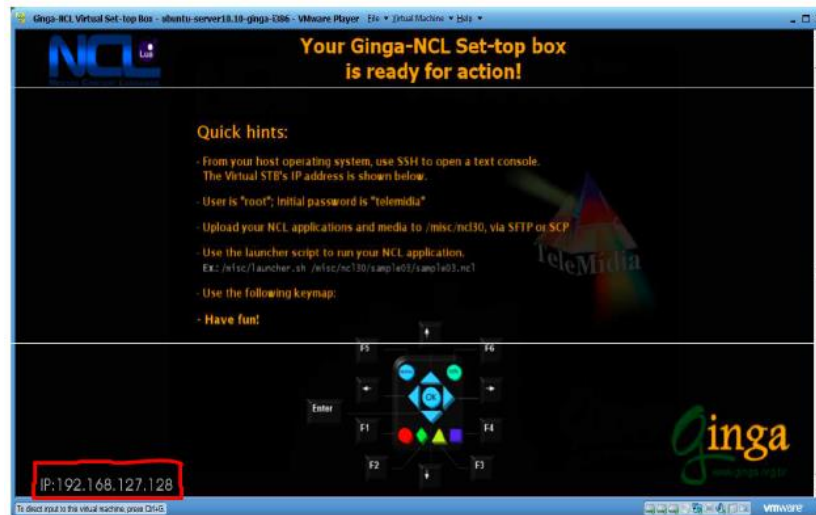


Figura 3.9. IP generada automáticamente en la máquina virtual
Fuente: Elaborado por el autor

Eclipse NCL es la herramienta que permitirá generar el código de la aplicación, un proyecto se encuentra configurado por una cabecera y un cuerpo, como se acota en la figura 3.10.

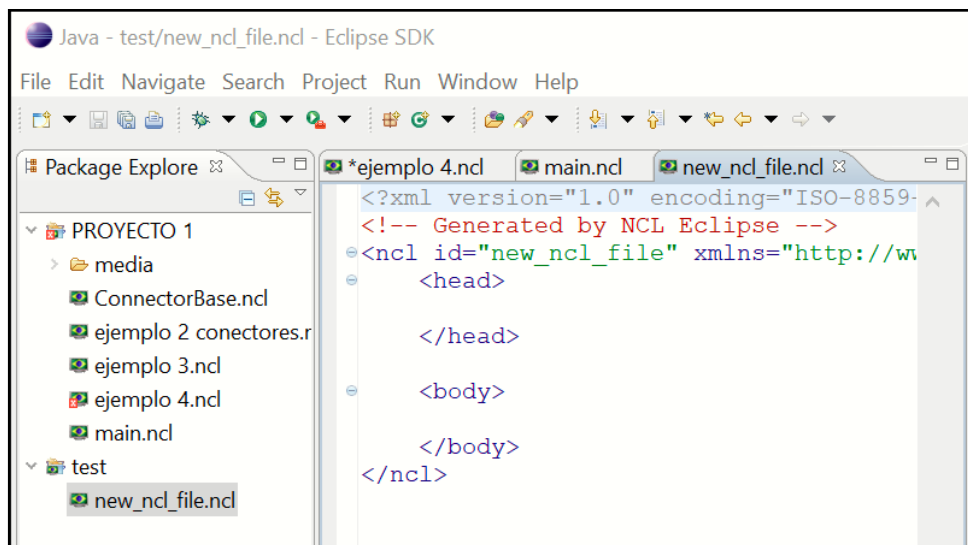


Figura 3.10. Esquema general de un proyecto en Eclipse NCL
Fuente: Elaborado por el autor

3.4 Diagrama del diseño

Los caminos para transmitir una aplicación básicamente son dos: el uno sería desde un canal de televisión y siguiendo una serie de etapas el video, audio y datos son codificados y multiplexados, estos generan tramas conocidas como BTS (*Broadcast Transport Streaming*). La señal es irradiada hacia la población y es captada por los receptores en los

hogares; el requisito para la recepción es a través de un televisor que contenga un decodificador integrado, regulado bajo el estándar ISDB-Tb o a su vez utilizando un decodificador externo, el cual debe ir adaptado al televisor a través de un cable HDMI. La característica principal del STB es que tiene que tener un *middleware Ginga* que permita abrir las aplicaciones al igual que el televisor tiene que estar acoplado al estándar de televisión digital usado en el país. El segundo camino, es el que se va a implementar en el proyecto que es cargar la aplicación directo al STB a través de un *pendrive*, y el canal de comunicación hacia el servidor web es utilizando el internet desde el STB que tiene un puerto de red RJ-45. El diagrama se presenta en la figura 3.11.

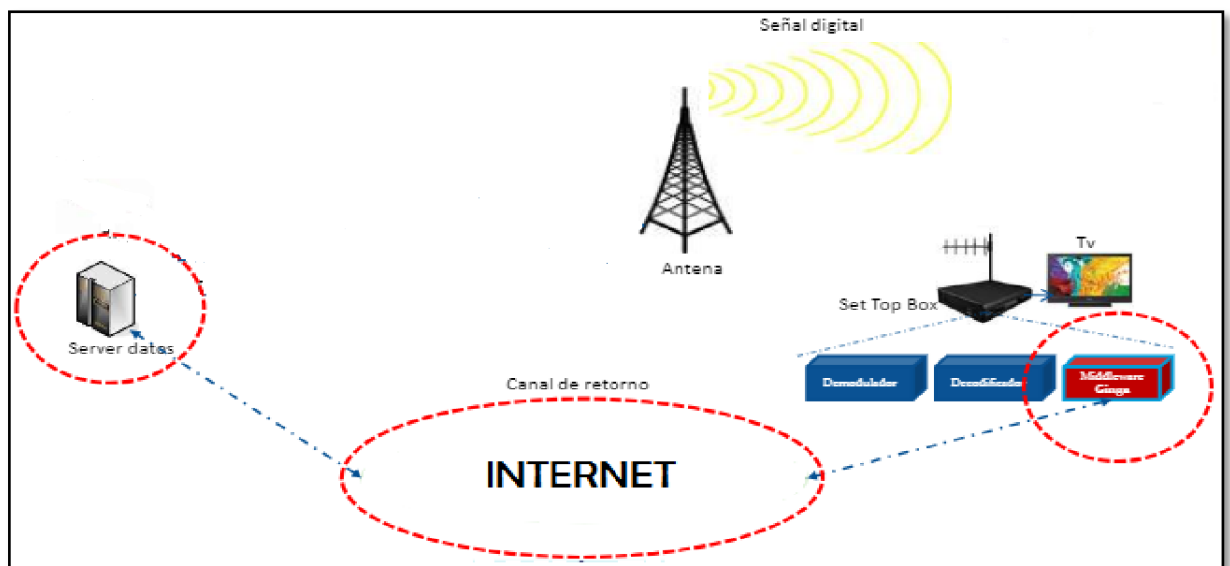


Figura 3.11. IP generada automáticamente en la máquina virtual
Fuente: Elaborado por el autor

3.4.1 Diagrama de flujo

El diagrama de flujo permite tener un camino de guía de lo que se pretende desarrollar, es la secuencia de etapas que tendrá la aplicación, para el caso de este proyecto en la figura 3.12 se describe la secuencia que tiene la aplicación *T-Voting*:

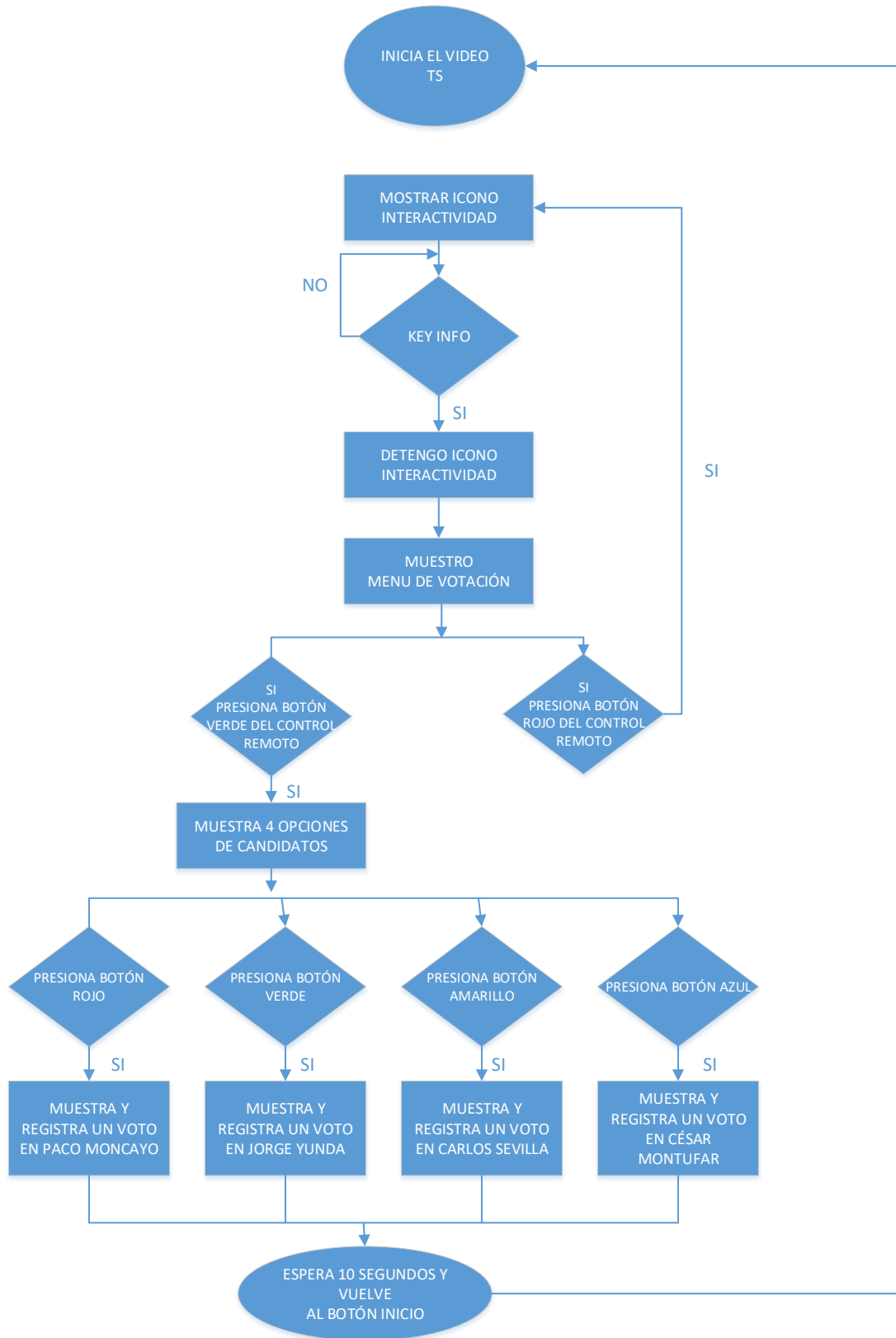


Figura 3.12. Diagrama de flujo de la aplicación T-Voting
 Fuente: Elaborado por el autor

3.4.2 Diagrama de Contexto

Es la representación gráfica de cuantos campos o menús se va a desarrollar en el diseño de la aplicación. Permite determinar y hacer un análisis de cuál es el alcance que se va a tener con su diseño como se indica en la figura 3.13.

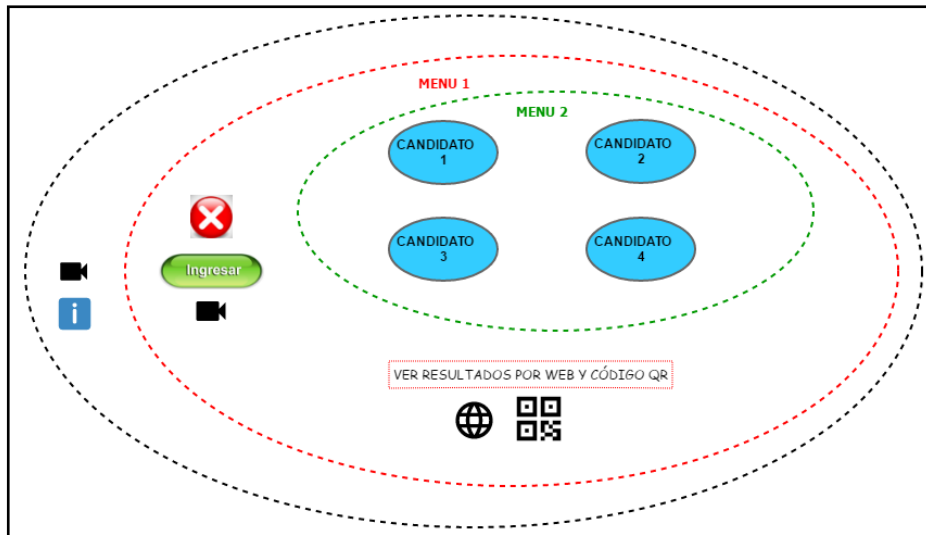


Figura 3.13. Diagrama de contexto general

Fuente: Elaborado por el autor

En la gráfica 3.14 se detallan los conectores que se tienen que crear para tener la secuencia de la aplicación, esto ayudará en el desarrollo de las funciones que tendrá cada conector.

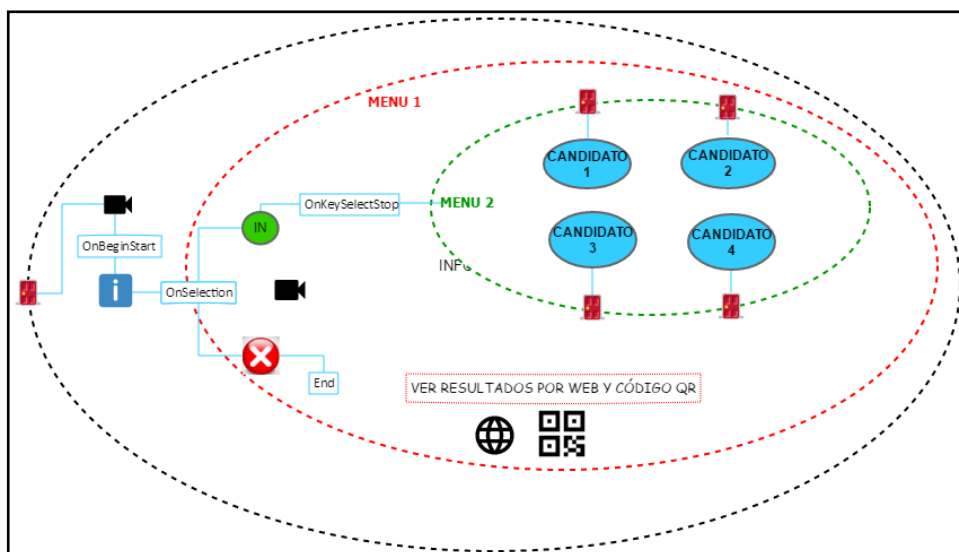


Figura 3.14. Detalle de conectores

Fuente: Elaborado por el autor

3.4.3 Presupuesto del proyecto

Para el diseño del proyecto no solo se generaron gastos de equipos, con anticipación se realizó un curso de 40 horas en Televisión digital interactiva.

El equipo que se utilizó para las pruebas de funcionamiento tiene que permitir interactividad en el estándar establecido para el Sistema Brasileño de TV Digital, que sea compatible con las normas declarativas Gínga NCL/Lua, cuyas características específicas se detallan en el anexo 5. El equipo tiene un costo elevado que supera los \$1000, por este motivo se optó por alquilar cuyo valor se detalla en la tabla 3.3.

Los gastos que se van a presentar para el desarrollo del proyecto son los que se detallan en la tabla 3.3:

Tabla 3.3. Presupuesto para el proyecto

PRESUPUESTO	\$
Curso de television digital	500
Alquiler de STB	80
Antena VHF	10
Hosting	80
Televisor	280
Total	950

Fuente: Elaborado por el autor

3.4.4 Ventajas de la aplicación

Con el diseño de la aplicación se trata de aprovechar los beneficios que ofrece la televisión digital. Como se ha mencionado anteriormente la televisión digital no solo se resume en una manera de optimizar el espectro radioeléctrico, sino de ver todos los beneficios que se pueden ofrecer, en la actualidad con el auge de las tecnologías y donde la mayoría de los hogares cuentan con un televisor en sus hogares las aplicaciones interactivas son un campo a explotar.

Las aplicaciones interactivas permiten llevar información útil para los televidentes, y además es una herramienta útil para las encuestadoras quienes son las encargadas de analizar los resultados que se generan.

En comparación con el tipo de encuestas que se realiza actualmente a través de una encuesta en papel o llamada telefónica, el usuario podrá realizarlo desde la comodidad de su hogar y en el tiempo de esparcimiento de cada uno, lo cual es beneficioso porque no interrumpe las actividades diarias de los usuarios sino se trata de aprovechar su tiempo de relax.

La aplicación permite conocer en tiempo real los resultados de las votaciones que realice a través de un código QR desde su Smartphone.

4 CAPITULO IV: IMPLEMENTACIÓN Y PRUEBAS

4.1 Desarrollo

El objetivo del proyecto es diseñar una aplicación interactiva con canal de retorno, que permita a los televidentes realizar una votación en forma de encuesta por un candidato de su preferencia utilizando los botones de colores (rojo, verde, amarillo y azul) del control remoto, como se visualiza en la figura 4.1.

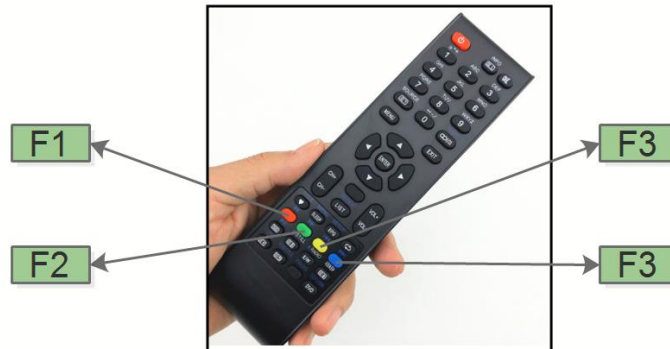


Figura 4.1. Teclas a usar del control remoto

Fuente: Elaborado por el autor

Los colores del control remoto están relacionados a las teclas función de una computadora, en la tabla 3.3 se detallan los principales.

Tabla 4.1. Teclas de función

TECLA	KERBOARD	NCL CODE
Rojo	F1	RED
Verde	F2	GREEN
Amarillo	F3	YELLOW
Azul	F4	BLUE
Menu	F5	MENU
Info	F6	INFO
Guide	F7	GUIDE
Ok	Enter	ENTER
Exit	Escape	EXIT
Izquierda	Izquierda	CURSOR_LEFT
Derecha	Derecha	CURSOR_RIGHT
Arriba	Arriba	CURSOR_UP
Abajo	Abajo	CURSOR_DOWN

Fuente: Elaborado por el autor

Las herramientas que se utilizaron para diseño y pruebas de la aplicación son las siguientes:

- Máquina virtual *Ginga* donde corre la aplicación de forma virtual.
- Eclipse NCL es donde se genera el código.
- *Set Top Box* programable modelo *eitv developer box*, equipo al que se carga la aplicación para las pruebas reales.

Lo primero que se va a realizar es determinar los candidatos, los cuales se han seleccionado al azar y que serán objeto de nuestra aplicación.

La aplicación está diseñada para determinar el nivel de interés de los televidentes por un candidato a la alcaldía de la ciudad Quito, para las elecciones de marzo del 2019, hay 16 candidatos de los cuales se ha seleccionado solo a cuatro como prueba, como se indica en la figura 4.2.



Figura 4.2. Candidatos a la alcaldía de Quito

Fuente: Elaborado por el autor

Una vez determinados cuantos candidatos van a ser encuestados a través de la aplicación mostrada en el televisor, se determina la ubicación en pantalla y el tipo de contenido que se va a mostrar al inicio de la aplicación. Se plantea la entrada con un botón que indique contenido interactivo, como se indica en la figura 4.3.

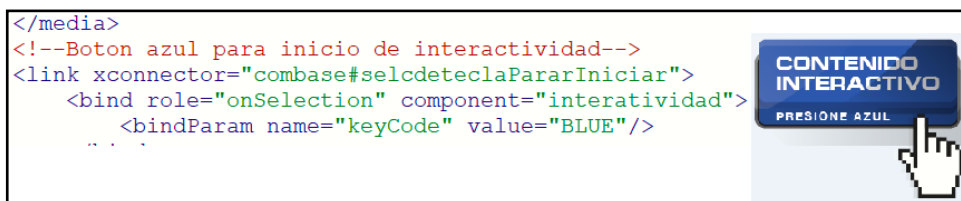


Figura 4.3. Botón de inicio de la aplicación

Fuente: Elaborado por el autor

El color seleccionado para el ícono de inicio de la aplicación es el color azul, además para que sea intuitivo que botón presionar los botones de entrada se configuraron con los colores del control remoto, como en este caso el ícono es azul, debe presionar el botón azul.

Una vez que presione el botón donde muestra el contenido interactivo. La intención es que aparezca un icono que indique que desea realizar la votación presionando el botón verde del control remoto como se indica en la figura 4.4, y se visualice en pantalla también un botón que indique salir, marcando el botón de color rojo como se muestra en la figura 4.5. Se han generado dos *links connectors* que permiten inicializar una acción en las figuras 4.4 y 4.5 se especifica los dos botones de entrada y salida.

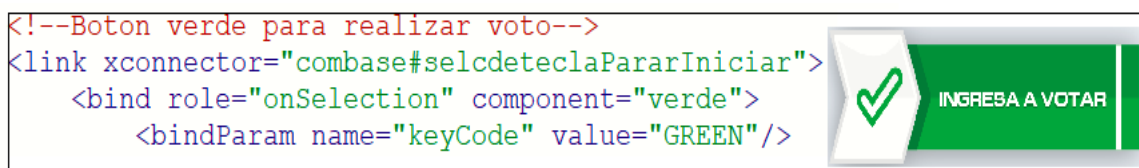


Figura 4.4. Botón de ingreso a la votación

Fuente: Elaborado por el autor

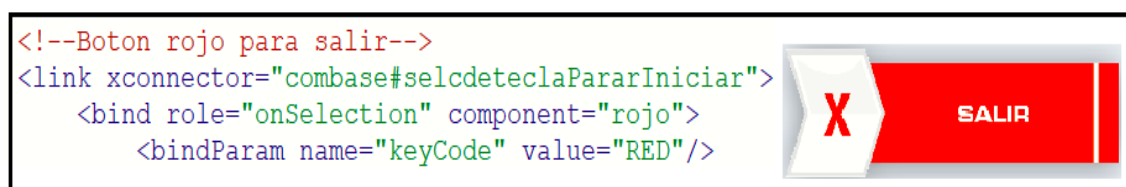


Figura 4.5. Botón de salir de la aplicación

Fuente: Elaborado por el autor

Al ingresar a votar aparecerá una barra en la parte inferior de televisor, que indica una dirección url donde están alojados los resultados de las encuestas o también puede visualizar los resultados a través de un lector de códigos QR, como se indica en la figura 4.6.



Figura 4.6. Barra de advertencia del contenido interactivo

Fuente: Elaborado por el autor

El enlace entre la aplicación y la base de datos se lo realiza a través de un script de acceso al servidor web, el código generado se muestra en la figura 4.7.

```
<!--Script lua para acceso al servidor web remoto para voto y consulta-->
<media id="lua" src="votacion2.lua" descriptor="dLua">
  <property name="voto"/>
  <property name="result"/>
</media>
```

VOTACIÓN2.LUA:

```
local host = "www.hst.com.ec"
print(evt.name, evt.value)

tcp.execute(
  function ()
    writeText("Registrando su voto...")
    tcp.connect(host, 80)
    print("Conectado a "..host)
    local url = "GET http://"..host.."/quito/votosc.php?
voto="..evt.value.."&n"
    print("URL: "..url)
    tcp.send(url)
    local result = tcp.receive("*a")
    if result then
      print("Datos de conexion TCP recibidos")
      f = loadstring(result)
      if f then
        f()
        writeResult(votosc)
        setLuaPropertie("result", 1)
      end
    else
      print("Error al recibir datos de la conexion")
    end
  end)
end
```

Figura 4.7. Script acceso al servidor web remoto

Fuente: Elaborado por el autor

Para votar por los candidatos se realizó la siguiente configuración de acuerdo a los colores del control remoto

- Color rojo: Para votar por el candidato Paco Moncayo
- Color verde: Para votar por el candidato Jorge Yunda
- Color amarillo: Para votar por el candidato Carlos Sevilla
- Color azul: Para votar por el candidato Cesar Montufar

Y como se puede evidenciar en la figura 4.8, los colores del marco de cada candidato están resaltados con los colores del control remoto que debe presionar.



Figura 4.8. Candidatos alcaldía de Quito

Fuente: Elaborado por el autor

Una vez realizada la votación por uno de los candidatos aparecerá un mensaje que indica un mensaje que se está registrando su voto como se indica en la figura 4.9.

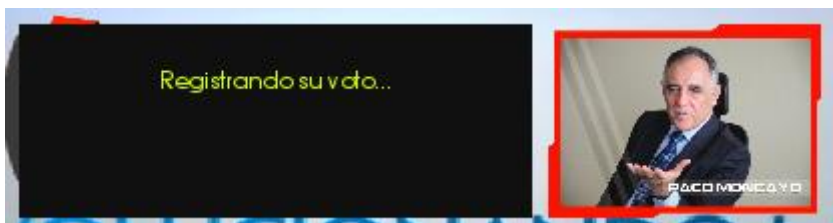


Figura 4.9. Mensaje de registro del voto

Fuente: Elaborado por el autor

Al responder la encuesta y registrado el voto habrá un tiempo de 10 segundos hasta que se registre el voto y automáticamente se cerrarán los menús y la aplicación volverá a su estado inicial.

Para poder ver los resultados se creó un script utilizando tcp.lua, que abre un canal de comunicación entre NCL y la base de datos con los archivos de las variables., en este caso existen cuatro variables, una por cada candidato, como se indica en la figura 4.10.

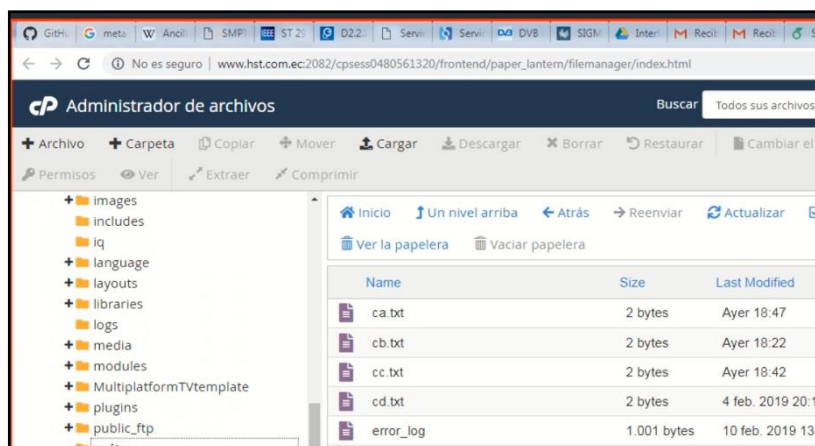


Figura 4.10. Archivos y base de datos cargados en el servidor web

Fuente: Elaborado por el autor

Los resultados se los puede ver de dos maneras, una es a través de un código QR que se visualiza en la pantalla (figura 4.11) y la otra es a través de la página web un link sé que se indica en la barra de advertencia de contenido como se indica en la misma figura. Sea cualquiera de los casos el resultado de los votos registrados se evidencia de la siguiente manera:

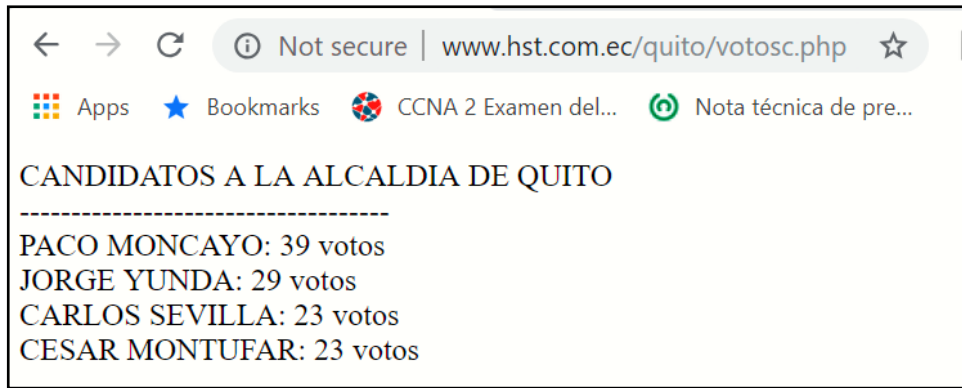


Figura 4.11. Resultados visualizados a través de la web

Fuente: Elaborado por el autor

Ese sería todo el proceso del diseño de la aplicación y de forma virtual se muestra como se indica en las siguientes figuras:

4.2 Pruebas virtuales

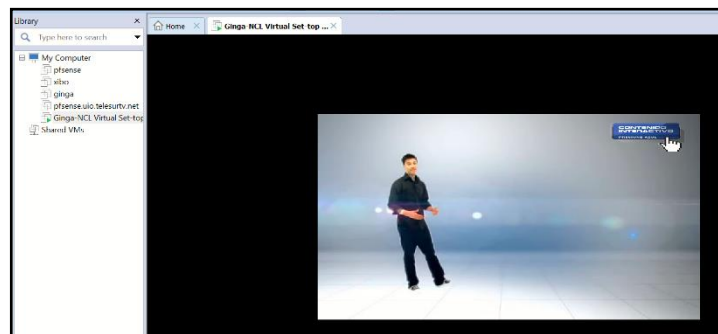


Figura 4.12. Carga de la aplicación en GINGA

Fuente: Elaborado por el autor

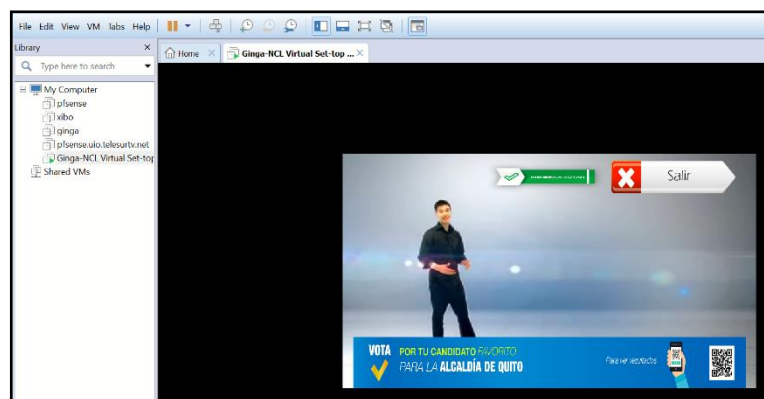


Figura 4.13. Ingreso al menú interactivo

Fuente: Elaborado por el autor

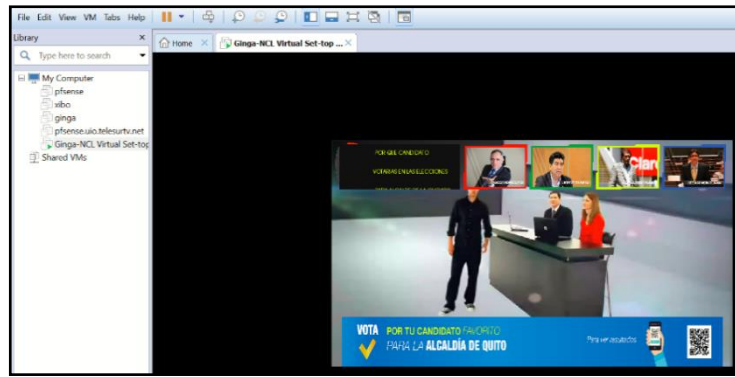


Figura 4.14. Visualización de los candidatos

Fuente: Elaborado por el autor

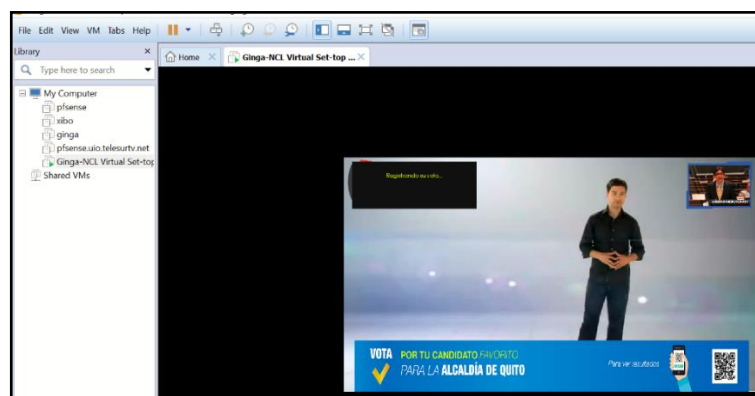


Figura 4.15. Registro de voto realizado por el televidente

Fuente: Elaborado por el autor

4.3 Pruebas reales

En la figura 4.16 se puede ver, mientras el televidente está observando la programación regular de un canal de televisión, en la parte superior derecha de la pantalla se muestra un botón que indica contenido interactivo.

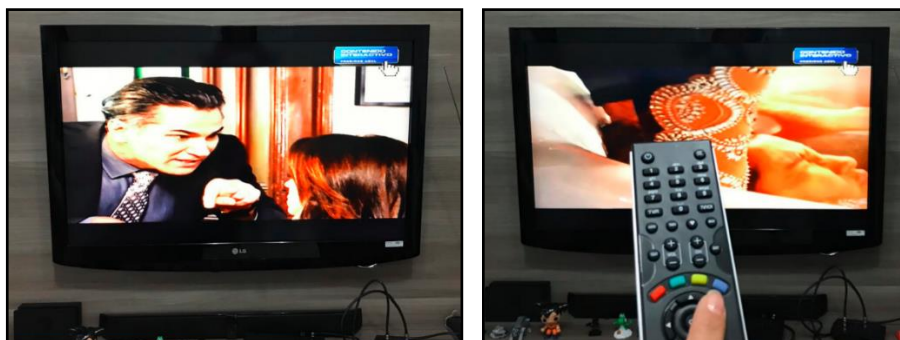


Figura 4.16. Mensaje de registro del voto

Fuente: Elaborado por el autor

Si selecciona la opción votar con el botón verde del control remoto, se despliega un submenú con una pregunta y cuatro opciones para escoger (Figura 4.17).

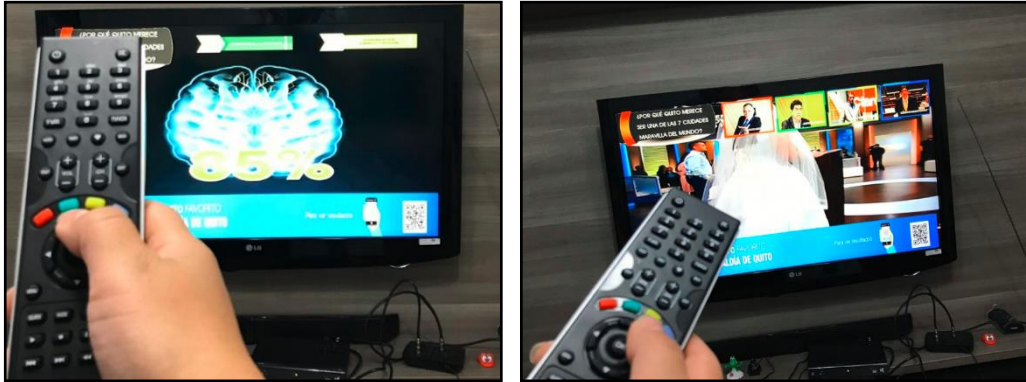


Figura 4.17. Ingresar a votar y seleccionar a una de las 4 opciones
Fuente: Elaborado por el autor

Registrado el voto por un candidato, los resultados se almacenan en un servidor *web*, al cual se puede acceder a ver los resultados a través de un lector de códigos QR o a través de la url (Figura 4.18). Estas dos opciones están ubicadas en una barra horizontal que se muestra en la parte inferior de la pantalla.

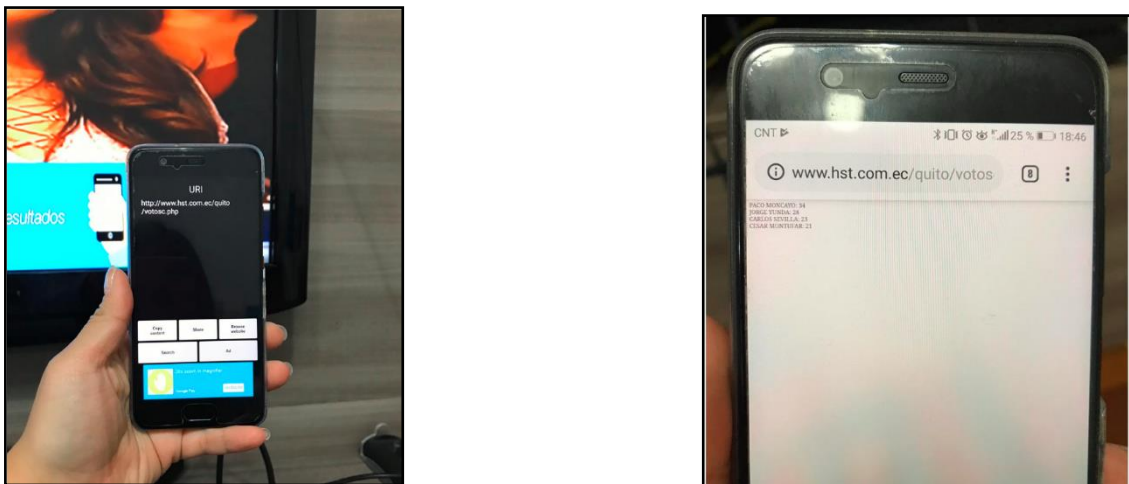


Figura 4.18. Escaneo con el celular del código QR y visualización de los resultados
Fuente: Elaborado por el autor

CONCLUSIONES

- Se realizó una investigación donde se pudo entender las etapas de procesamiento de la señal. El transporte de la información audio, video y datos son divididos en paquetes que son multiplexados para formar un flujo de transporte TS, el sistema de procesamiento no solo permite enviar un servicio sino algunos como: televisión en HD, televisión en SD y *one seg* para dispositivos móviles.
- Se desarrolló una aplicación interactiva para la televisión digital terrestre usando las herramientas de desarrollo reguladas por el estándar ISDB-Tb. *Software* como *Ginga-NCL* permitieron visualizar de forma virtual el contenido generado y la simulación del control remoto se lo realizó con las teclas de función (F1, F2, F3 y F4) de la computadora.
- La herramienta que se utilizó para el desarrollo del contenido fue *Ginga-NCL*, debido a que utiliza un lenguaje de programación de alto nivel para contenido interactivo, que se acopla a los requerimientos de la aplicación.
- La utilización de *plugins* de Eclipse NCL brindó una gran ventaja respecto a otros programas de desarrollo de contenido interactivo, debido a que estos complementos facilitaron librerías de comunicación entre la aplicación y el servidor web donde se alojaron los archivos de votación.
- Para la programación del código en el lado del servidor web se contempló la idea de generar gráficos de pastel para indicar los resultados, pero por falta de conocimientos en lenguajes de programación de bajo nivel se optó por utilizar un lenguaje amigable para únicamente hacer un script en php que permita actualizar los archivos en texto plano cada vez que se realice una votación.
- Se comprobó que la manera más rápida de acceder a los resultados es a través de un lector de códigos QR instalado en el celular, debido a que al capturar con la App QR, lleva directo a la página de la información de la encuesta.

- En las pruebas de funcionamiento realizadas a un grupo de 10 personas, se obtuvo un promedio de 8 sobre 10. Resaltando por los participantes la facilidad para acceder a la aplicación y la manera rápida de obtener los resultados.
- La resolución de las imágenes es un dato a tener en cuenta, estas deben tener una resolución de 1920x1080 para que al momento de mostrar en el televisor estas no se vean degradadas. Cabe señalar que el tamaño máximo de la aplicación es de 6 MB.
- Permite tener un resultado de forma rápida de la aceptación de un candidato, incluso en eventos futuros se podría anexar preguntas generales cerradas que permitan recabar más información.

RECOMENDACIONES

- Para el desarrollo del proyecto es necesario tener los conocimientos básicos de televisión digital terrestre, como se envían los datos junto con el audio y video y los equipos que están involucrados en la transmisión.
- Antes de generar una aplicación es necesario tener claro el alcance que se quiere dar a la aplicación, el tipo de contenido que se va a mostrar, debe tener un conocimiento del manejo de lenguaje de programación. En este caso que se utilizó Eclipse NCL para su desarrollo.
- Es indispensable que al momento de adquirir los equipos para las pruebas de la aplicación se adquiera un Set Top Box que sea configurable, para poder cargar la aplicación.
- Para el caso de que se requiera transmitir la aplicación multiplexada se deberá generar un TS, y configurar las tablas en el modulador con ciertos PID (identificador de paquete) para los paquetes de las diferentes fuentes. Cuando se decodifiquen los paquetes lleguen etiquetados y pueda recuperarse la señal (audio, video y datos).
- El desarrollo de la aplicación debe tener un contenido visual, que sea entendible para el televidente, no demasiado largo porque confundiría al usuario, tiene que ser conciso.
- Los botones con la información tienen que ser intuitivos, deben ser relacionados con los botones que trae el control remoto, para que el televidente se familiarice y no se vuelva compleja al momento de usar.

BIBLIOGRAFÍA

Alulema, D. (1970). La Televisión Digital Terrestre en el Ecuador es interactiva. *Eidos*, (5), 12. <https://doi.org/10.29019/eidos.v0i5.89>

División de canal UHF ISDB-T en segmentos | YouBioit.com. (2012). Recuperado 10 de enero de 2019, de <https://www.youbioit.com/es/article/12389/division-de-canal-uhf-isdb-t-en-segmentos>

Estandar ISDB-T para Television Digital Terrestre – DivXLand.org (s.f.). Recuperado 9 de enero 2019 de https://www.divxland.org/es/article/23/estandar_isdb_t_para_television_digital_terrestre#.XDKtu1xKg2x

Ginga. (s. f.). Recuperado 9 de enero de 2019, de <http://www.ginga.org.br/>

International, R. & S. (s. f.). Tecnología ISDB-TB. Recuperado 9 de enero de 2019, de https://www.rohde-schwarz.com/es/tecnologias/difusion-terrestre/isdb-tb/tecnologia-isdb-tb/isdb-tb_55800.html

S. Gutierrez. (2013). Guía para el desarrollo de aplicaciones interactivas en TDT para colombia. Recuperado de <http://bdigital.unal.edu.co/11869/13/300333.2013.pdf>

Soares, L. F. G. (s. f.). TV Interactiva se hace con Ginga, 4.

Tejedor, R. J. M. (s. f.). Televisión Digital Terrestre (TDT), 5.

Torres, J. (2011, junio 28). Comunidad Ginga Ecuador: Middleware Ginga. Recuperado 9 de enero de 2019, de <http://comunidadgingaec.blogspot.com/2011/06/middleware-ginga.html>

Urbina, I. S., Fuentes, I. O., & Miranda, I. D. (2015). Analisis del estandar ISDB-T para ser utilizado en una posible implementación de la televisión digital terrestre en Nicaragua, 89.

Valencia, B. (2013, julio). Desarrollo de Aplicaciones Interactivas para TV Digital orientadas a formar a la Población en Desastres Naturales.

ANEXOS

ANEXO 1: INSTALACIÓN DE GINGA

A continuación, se detalla la instalación de *Ginga*:

1. Primeramente, se necesita tener a la mano el *software virtualizador* de máquinas *vmware player* y se realiza los siguientes pasos:
 - Ejecutar el instalador, marcar el visto en *Desktop* para que se genere un acceso directo y se presiona *next*.
 - En la siguiente ventana se presiona *continue*.
 - Aceptar los términos de licencia.
 - Finalmente se resetea la computadora.

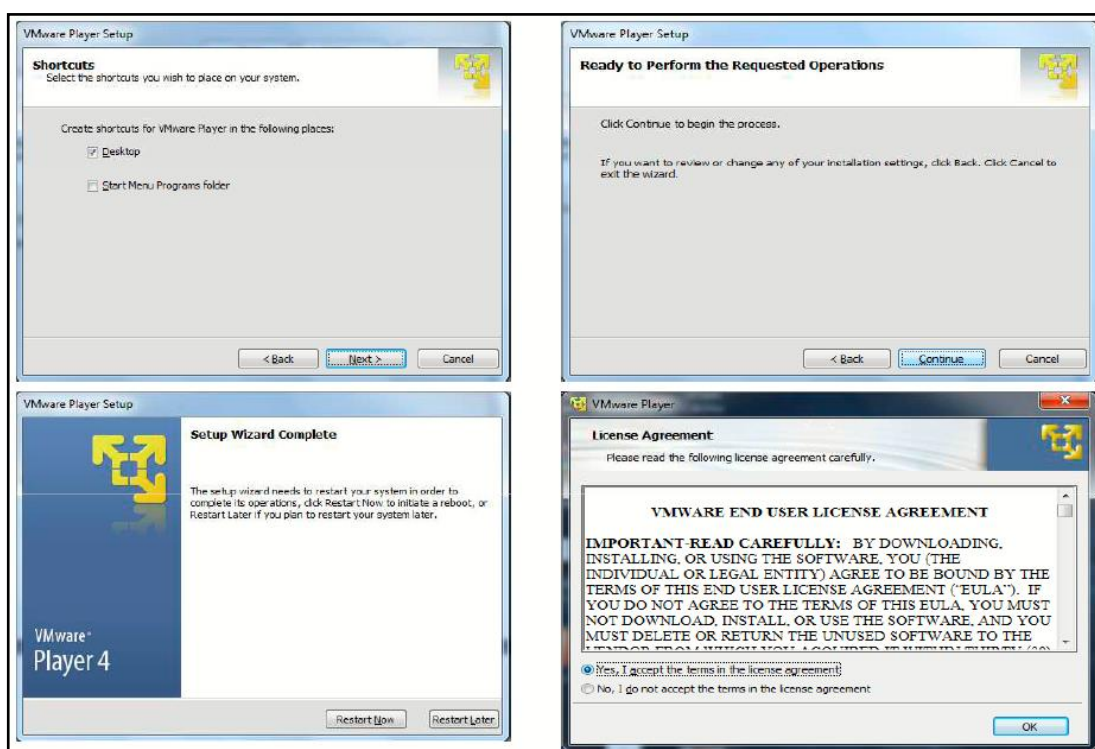


Figura 1. Instalación Vmware player

2. Instalado el programa que permite ejecutar las máquinas virtuales, se carga el *software ginga*, el formato del archivo es *.vmx*, como se indica en la figura 2.

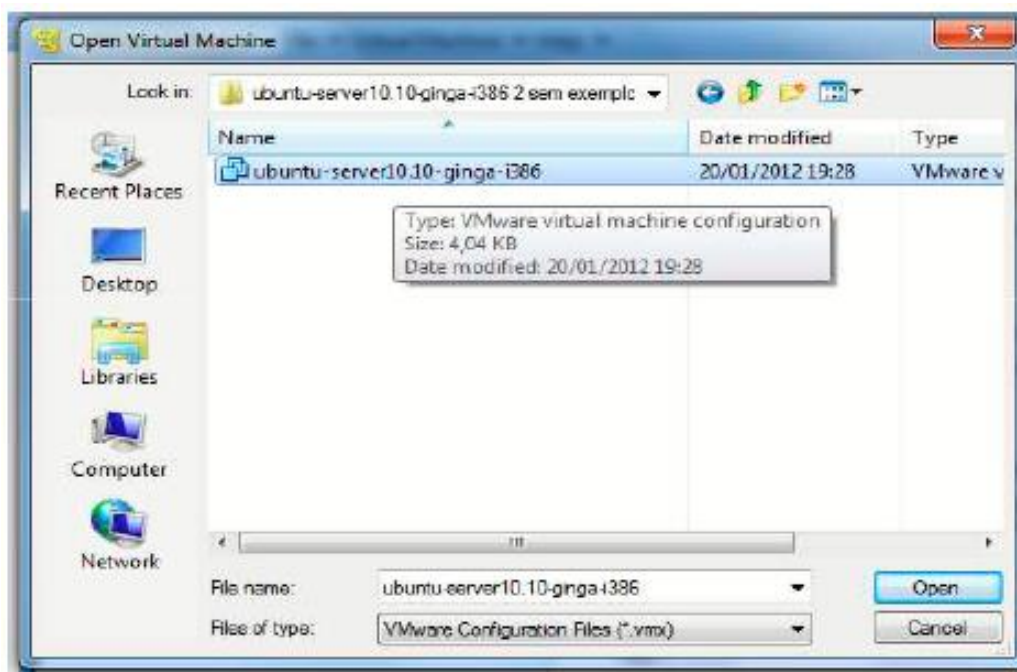


Figura 2. Selección del software *Ginga*

3. En la figura 3 se indica la máquina virtual instalada.

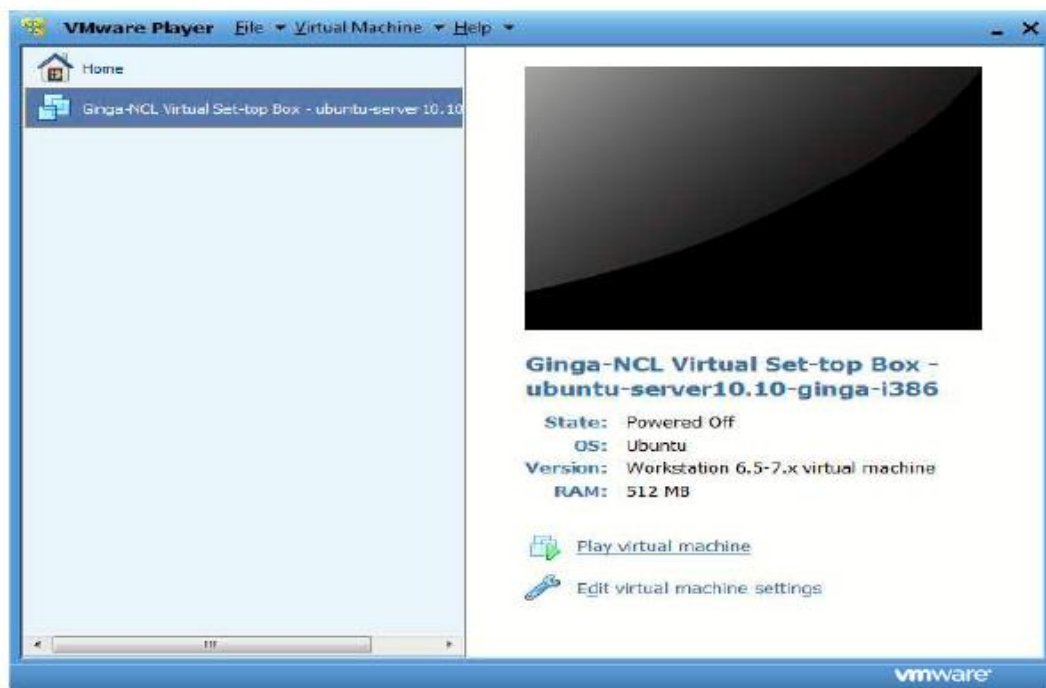


Figura 3. Máquina virtual *Ginga* cargada al *vmware*

4. Al encender la máquina de *Ginga* se muestra como en la figura 4, aquí se debe seleccionar la resolución que se desea utilizar; lo recomendable es 1280x720.

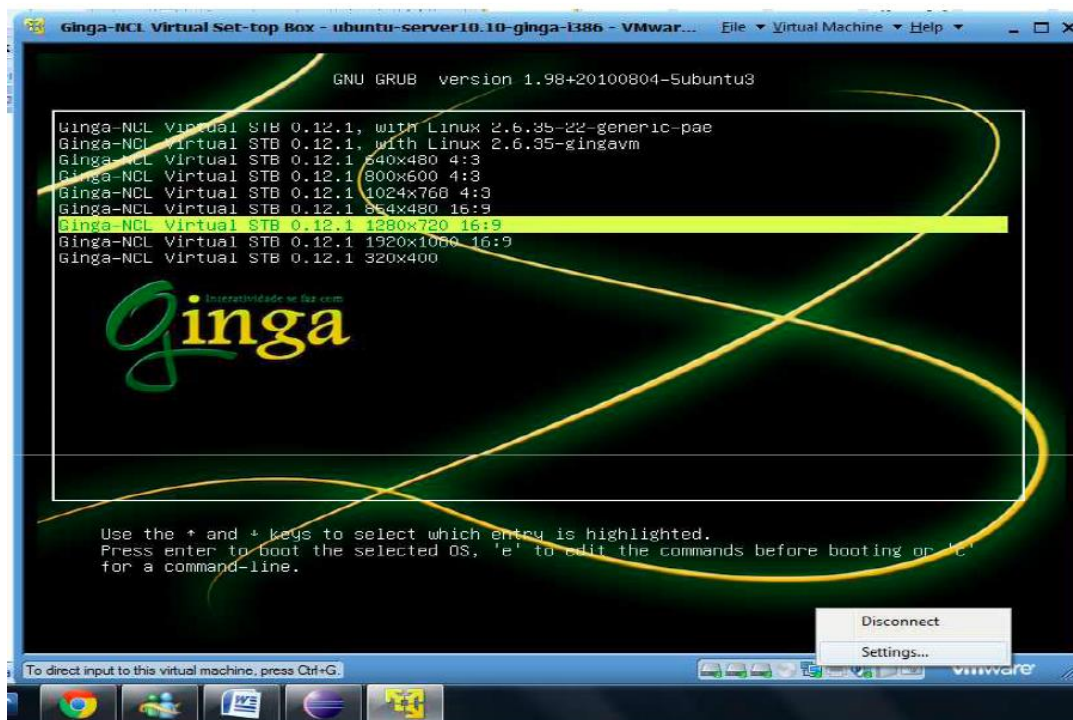


Figura 4. Resoluciones disponibles para usar *Ginga NCL*

5. Finalmente en la figura 5, se indica la interfaz final de *Ginga NCL*

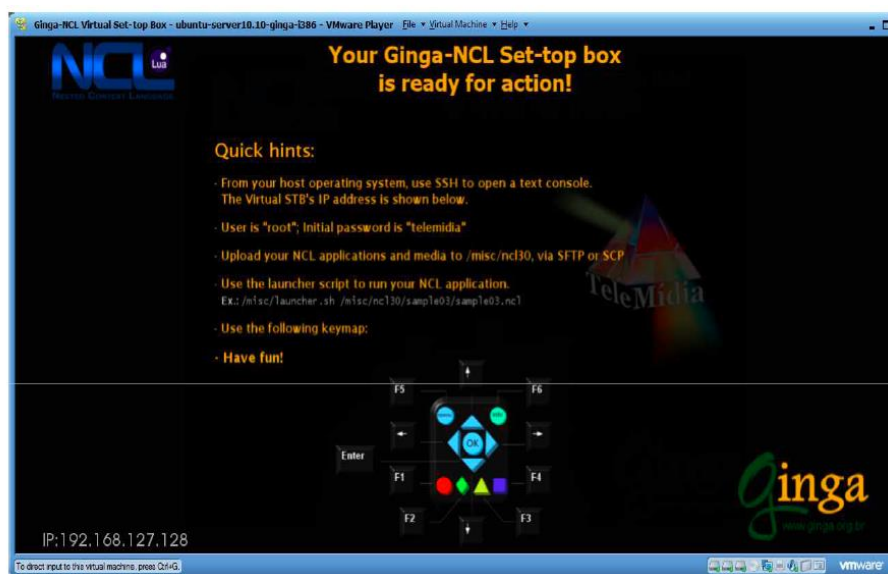


Figura 5. Interfaz de *Ginga NCL*

ANEXO 2: INTALACIÓN DE ECLIPSE NCL

Acontinuación se detallan la secuencia de pasos que se requieren para la instalación de Eclipse NCL:

1. el archivo eclipse SDK-3.7.1 viene comprimido, para proceder a ejecutar el archivo *.exe*, se necesita descomprimir como se lo realiza en la figura 1.



Figura 1. Descomprimir el achivo eclipse

2. Previo a ejecutar el archivo *eclipse.exe* que se indica en la figura 2, se requiere tener instalado *java JRE (Java Runtime Environment)* 1.6 o superior

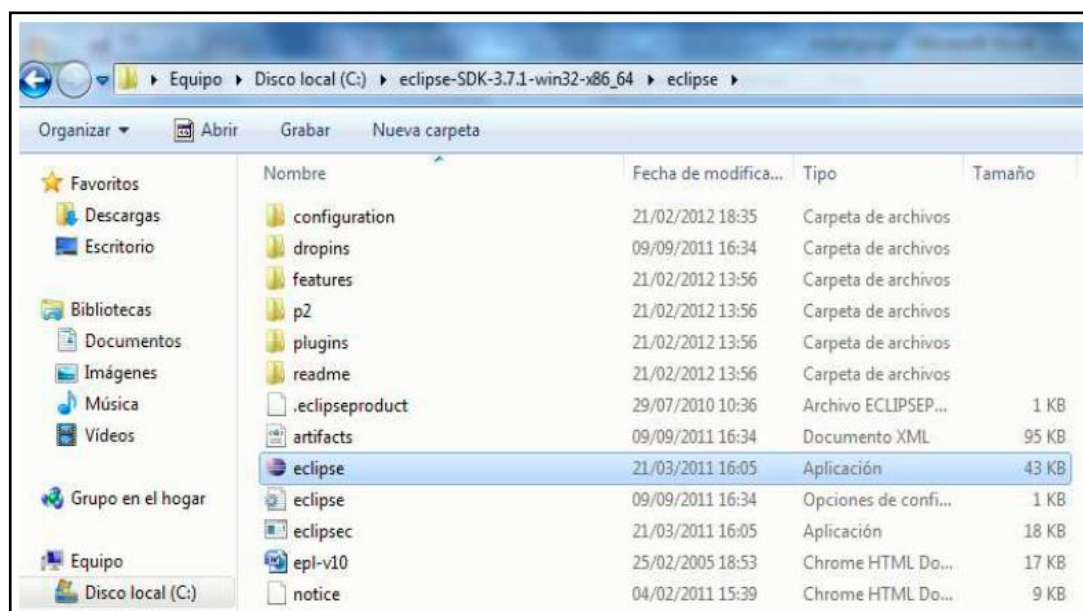


Figura 2. Ejecutar el archivo eclipse.exe

3. Al ejecutar el instalador, se despliega una ventana como se indica en la figura 3. En esta ventana, en la pestaña "Help" se selecciona la opción "Install New Software".

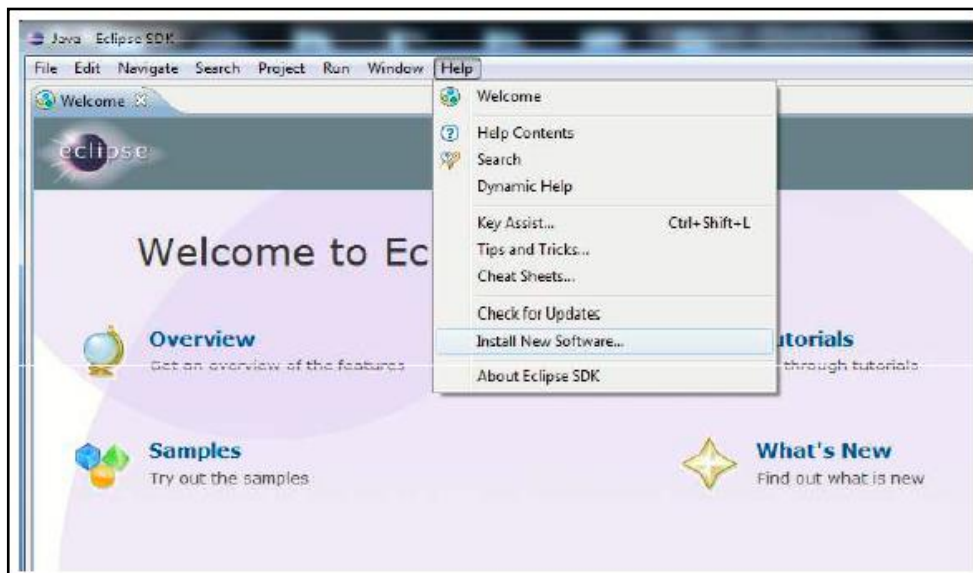


Figura 3. Instalación de nuevo software

4. En la siguiente ventana, se ubica un nombre para el *software* a instalar y una dirección del repositorio de donde se va a descargar, presionamos "OK" y "Next", tal como se indica en la figura 4.

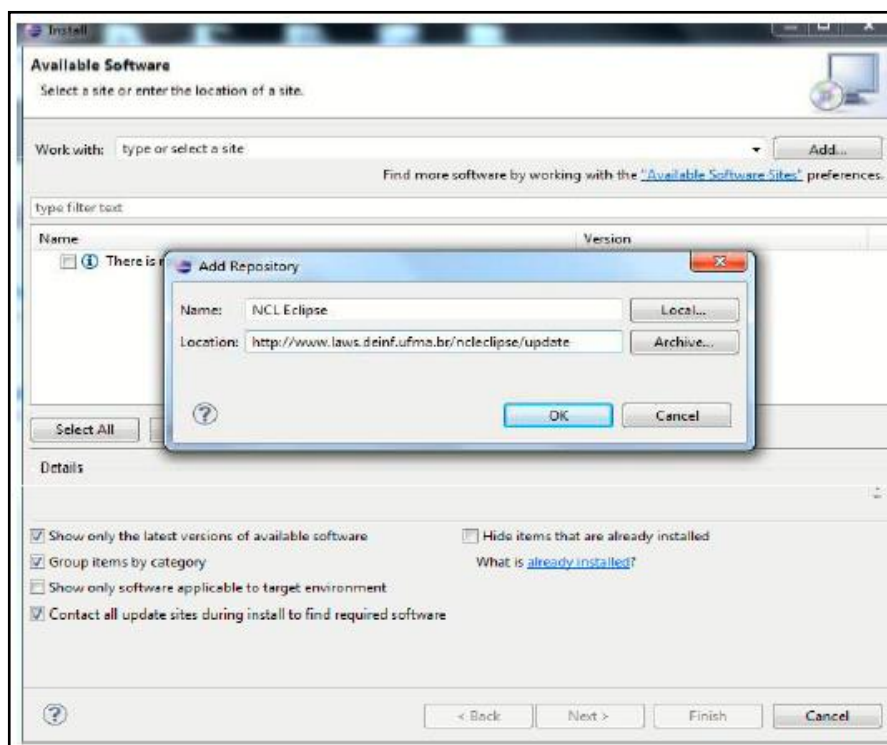


Figura 4. Repositorio de descarga del *software* NCL

5. En la ventana anterior al presionar next, muestra las versiones de las librerías disponibles. Seleccionamos la versión mas actual y presionamos "Next", como se visualiza en la figura 5.

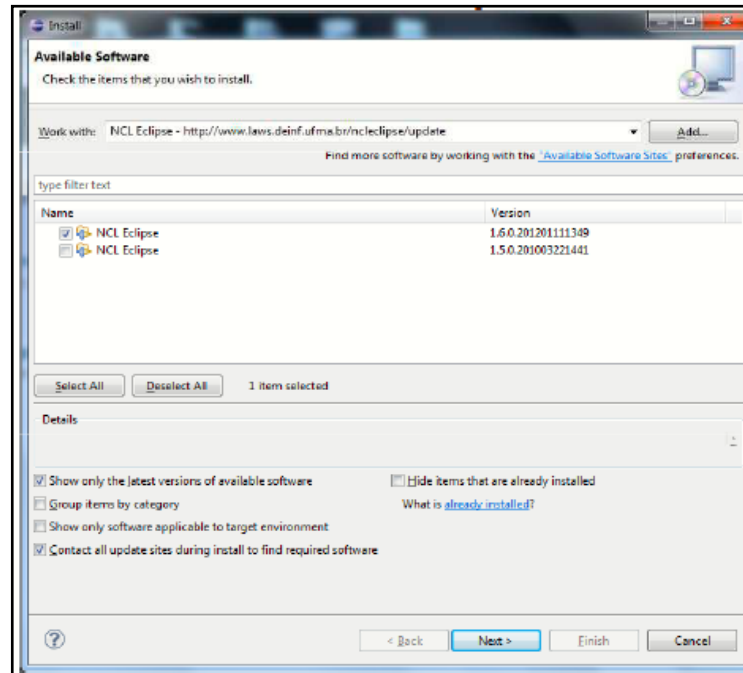


Figura 5. Librerías disponibles de NCL eclipse

6. Se quita el visto de "always run in background", y se presiona en "run in background". Posteriormente se muestra una ventana de advertencia se presiona "ok" y luego se reinicia la aplicación Eclipse, como se indica en la figura 6.



Figura 6. Instalación de software

7. Una vez instalada la librería *NCL eclipse*, en la pestaña *windows* se despliegan una lista de opciones, se selecciona *preferences*, tal como se indica en la figura 7.

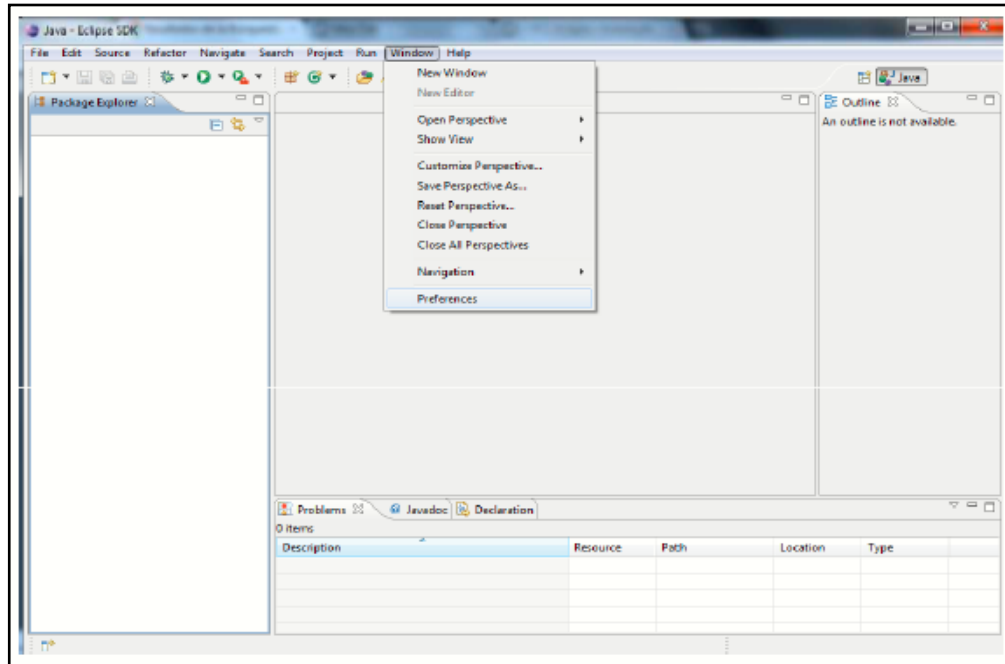


Figura 7. Selección de ventana de configuración

8. En la ventana *preferences* solo se modifica el *hostname*, *el username* y *password* (estos datos son obtenidos de la máquina virtual *Ginga*).

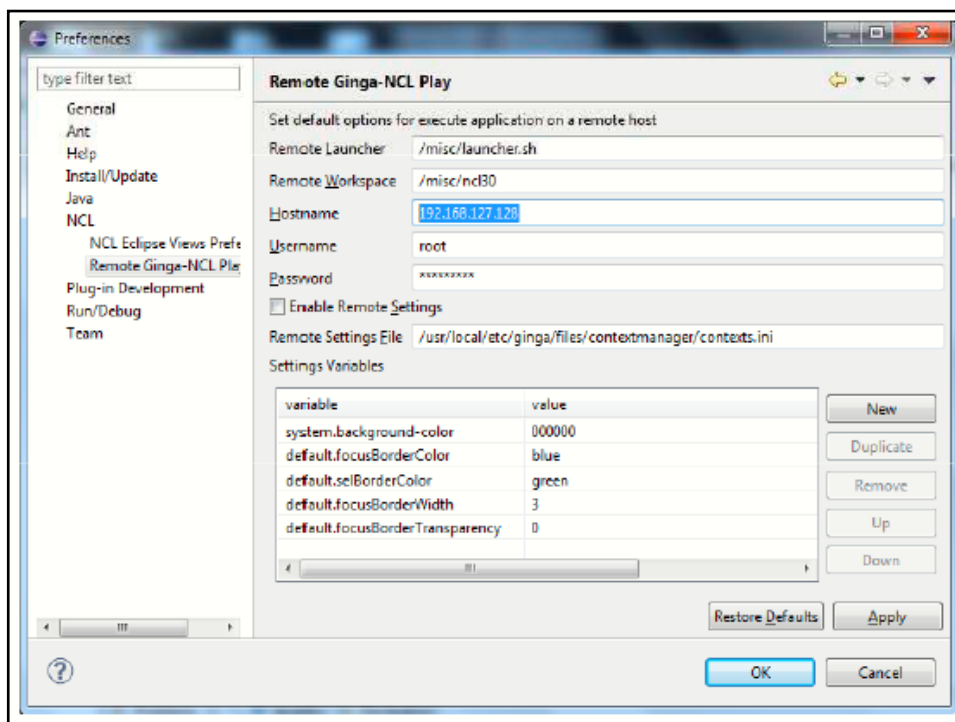
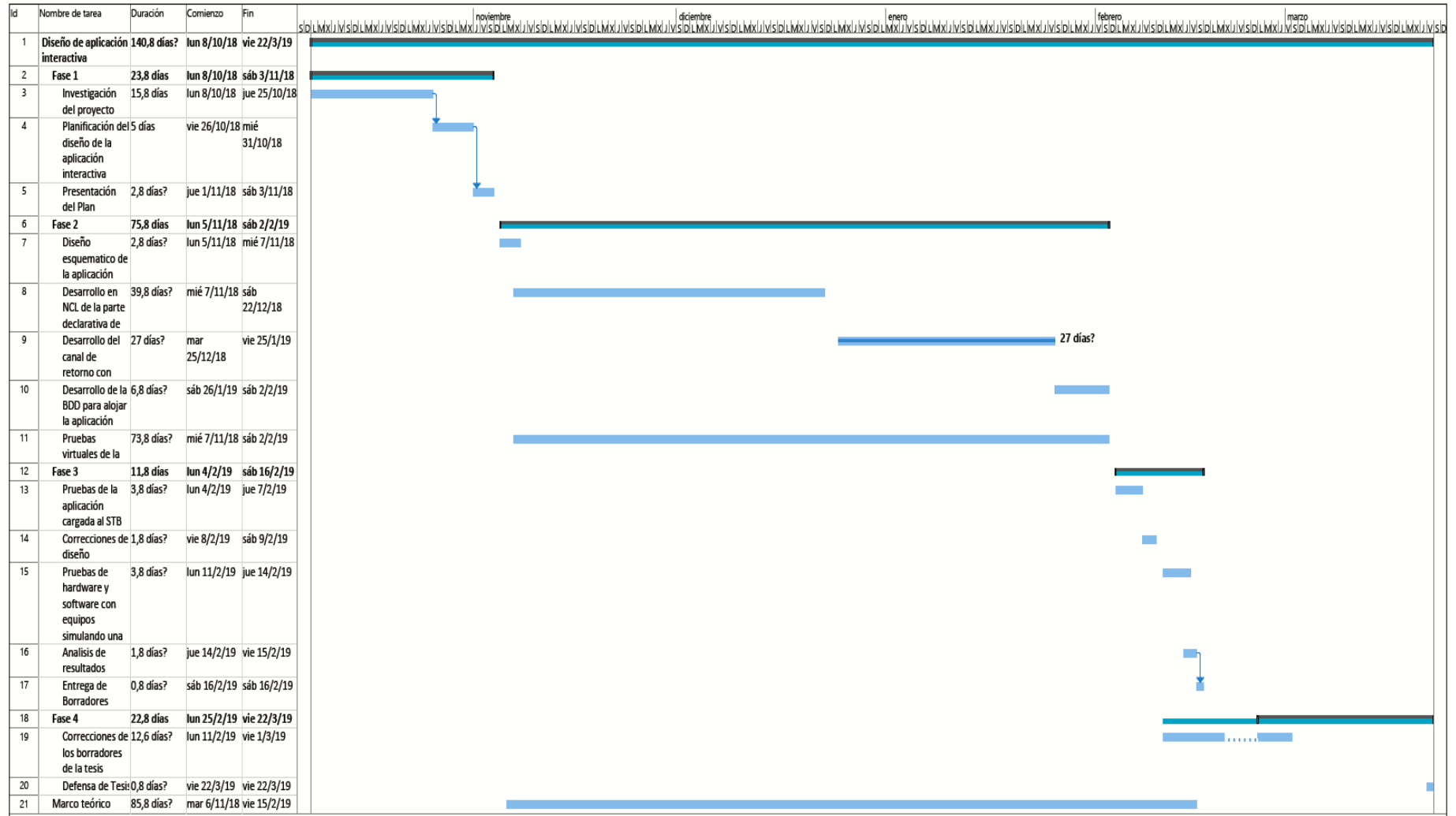


Figura 8. Configuración de parámetros para conexión con la máquina virtual *Ginga*

ANEXO 3: DIAGRAMA DE TIEMPO



ANEXO 4: DATASHEET SET TOP BOX



EiTV smartBox es un producto de EiTV, con características que lo diferencian de los demás conversores digitales existentes en el mercado. Fuera de sintonizar **canales digitales de TV abierta en alta definición**, **EiTV smartBox** tiene soporte total a **interactividad en el estándar definido para el Sistema Brasileño de TV Digital (DTV)** que permite acceso a diversos aplicativos interactivos transmitidos por los canales de TV.

EiTV smartBox transforma su receptor de TV en una Smart TV. Fuera de tener **sonido e imagen digitales** con calidad espectacular. **EiTV smartBox** tiene funciones interactivas de acceso a vídeos. Es la evolución de la TV Digital en la palma de su mano.



RECURSOS

IMAGEN

- ▶ Resolución en Full HD

- ▼ Señal sin interferencias y ruidos
- ▼ Guía de Programación en la pantalla
- ▼ Lista de canales favoritos

INTERACTIVIDAD

- ▼ Ginga (DTV): Ejecución de aplicativos interactivos transmitidos por los canales, almacenados en un pendrive externo via puerta USB o via Red. Para todos los casos es posible coleccionar informaciones del aplicativo, auxiliando en el desarrollo de estos.

SONIDO

- ▼ Alta calidad de sonido, 6 canales de audio (surround 5.1)
- ▼ Closed Caption
- ▼ Áudio Secundário (SAP)
- ▼ Áudio Descripción

Características Técnicas

- ▼ HDTV: TV Digital de Alta Definición en ISDB-T
- ▼ Formatos de Vídeo: MPEG-2, MP@/ H.264-AVC HP@L4.0
- ▼ Formatos de Áudio: MPEG-1/2 layer I/II, HE-AAC
- ▼ GNU/LINUX OS
- ▼ EITV Zapper + Middleware Ginga
- ▼ Chipset: STiH206
- ▼ FR output: loop through
- ▼ Faja de Frecuencia: UHF 470 - 806 MHz (canales 14 até 69)
- ▼ Full seg 5.7Mhz, One seg 0.43 Mhz
- ▼ Sensibilidad: -75 dBm nivel de entrada mínima
- ▼ RAM 512MB DDR3
- ▼ Flash 4 GB eMMC
- ▼ Salida HDMI
- ▼ Salida modulada en RF (canal 3/4)
- ▼ Salida RCA (VIDEO / R-audio-L)
- ▼ 2 puertas USB externas
- ▼ 1 puerta Ethernet: RJ45 - 10/100 base
- ▼ Guía Electrónica de TV (EPG)
- ▼ Control Parental protegido por contraseña
- ▼ Tecnología Closed Caption (subtítulo)
- ▼ Interactividad (DTV – Ginga)
- ▼ Sistema de aviso de emergencia (EWBS)
- ▼ Permite actualización de software via puerta USB
- ▼ Conexión en red via puerta Ethernet
- ▼ Fuente 12V Externa de Alimentación de Energía: AC 100 - 240V~ 50/60Hz
- ▼ Peso: 1 Kg
- ▼ Dimensiones: 170 x 110 x 35 mm

CONTENIDO DEL EMBALAJE



Convertor digital, Guia de Instalación



Control Remoto, Pilas / baterias AAA.



Cable RF

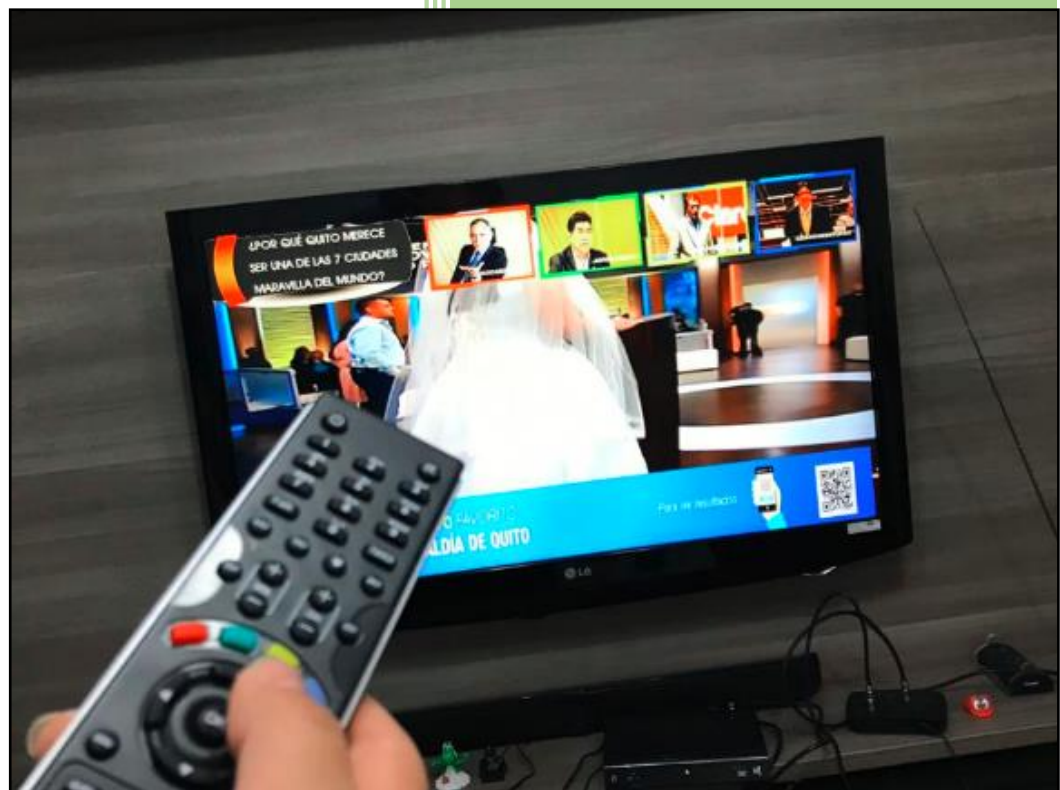


Fuente de energía



EITV – Entretenimento e Interatividade para TV Digital
 Rua Rafael Andrade Duarte, 600 – 6º Andar
 Jardim Paraíso - CEP 13.100-011 - Campinas - SP - Brasil
 Fone/Fax: (19) 3579-0744
 E-mail: atendimento@eitv.com.br
 WEB Site: www.eitv.com.br

MANUAL DE USO DE LA APLICACIÓN



APP T-VOTING

ANEXO 5: MANUAL DE USO DE LA APLICACIÓN

Requerimientos

Los requerimientos para hacer uso de la aplicación en los televisores de los hogares es el siguiente:

- Set Top Box "Conversor TV Digital", modelo TS 2017.
- Antena de recepción de señal VHF/UHF.
- Flash memory con la App.

Instalación

1. Cargue la aplicación en un pendrive y conectelo a la entrada USB del *set top box*, el espacio requerido para almacenar la aplicación es de 6MB
2. Para el uso del equipo receptor de aplicaciones se debe tener acceso a internet, este equipo requiere de acceso a la red para que pueda enviar la información hacia el internet. Por defecto cuando se enciende el equipo STB, este viene con una ip predeterminada de forma estatica, y para que se acople a la red del usuario esta opción debe ser cambiada a DHCP, como se puede verificar en la figura 1.



Figura 1. Configuración de la tarjeta de red del STB

Hay que tener en cuenta que el decodificador debe estar conectado al televisor a través de un cable HDMI.

3. Una vez cargada la aplicación en el STB este mostrará con un "ok" cuando haya sido cargada correctamente la aplicación. Realizado los pasos anteriores, el ícono de la aplicación se mostrará en el televisor, como se indica en la figura 2.



Figura 2. Ícono de inicio de la aplicación

4. Al hacer clic en el ícono este muestra un menú con dos íconos que indican "Realizar votación" y "Salir", si selecciona realizar votación se abre un sub-menú que indica los 4 candidatos cargados en la aplicación, se seleccionó 4 candidatos por facilidad de votación aprovechando los 4 colores que vienen por defecto en el control remoto (rojo, verde, amarillo y azul).



Figura 3. Ingreso al sub-menú donde se visualizan los candidatos

5. Una vez realizada la votación, puede visualizar los resultados generales ingresando al *link* donde se encuentra alojada la base de datos, o a través de un lector de códigos desde el celular.

ANEXO 6: CÓDIGO DE LA APLICACIÓN INTERACTIVA

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--Sistema de encuesta para TV Digital-->
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <regionBase>
      <region id="rgVideo" width="100%" height="100%"
zIndex="1"/>
      <region id="rgrojo" height="14%" width="30%" top="3%"
left="65%" zIndex="2"/>
      <region id="rgamarillo" height="14%" width="30%"
top="3%" left="65%" zIndex="2"/>
      <region id="rgverde" height="14%" width="30%" top="3%"
left="34%" zIndex="2"/>
      <region id="rgFondo" height="21%" width="64%" top="2%"
left="34%" zIndex="2"/>
      <region id="rgFondoB" height="22%" width="95%"
bottom="0%" left="2.6%" zIndex="2"/>
      <region id="rgFondo2" height="21%" width="30%" top="1%"
left="1%" zIndex="2"/>
      <region id="rgLua" height="20%" width="30%" top="2%"
left="2%" zIndex="3"/>
      <region id="rgima" height="20%" width="16%" top="2%"
left="33%" zIndex="3"/>
      <region id="rgimb" height="20%" width="16%" top="2%"
left="49.5%" zIndex="3"/>
      <region id="rgimc" height="20%" width="16%" top="2%"
left="66%" zIndex="3"/>
      <region id="rgimd" height="20%" width="16%" top="2%"
left="82.5%" zIndex="3"/>
      <region id="rgInt" width="19.5%" height="14.4%"
top="4.0%" right="4.0%" zIndex="3"/>
    </regionBase>
    <descriptorBase>
      <descriptor id="dVideo" region="rgVideo"/>
      <descriptor id="damarillo" region="rgamarillo"/>
      <descriptor id="drojo" region="rgrojo"/>
      <descriptor id="dverde" region="rgverde"/>
      <descriptor id="dima" region="rgima"/>
      <descriptor id="dimb" region="rgimb"/>
      <descriptor id="dimc" region="rgimc"/>
      <descriptor id="dimd" region="rgimd"/>
      <descriptor id="desInt" region="rgInt"/>
      <descriptor id="dFondo" region="rgFondo"/>
      <descriptor id="dFondoB" region="rgFondoB"/>
      <descriptor id="dFondo2" region="rgFondo2"/>
      <descriptor id="dLua" region="rgLua"/>
    </descriptorBase>
    <connectorBase>
      <importBase documentURI="connectorBase.ncl"
alias="combase"/>
    </connectorBase>
  </head>
  <body>
    <port id="portaInteratividad" component="interatividad"/>
    <port id="videoinicial" component="video"/>
    <media id="video" src="media/video.mp4" descriptor="dVideo"/>
    <media id="interatividad" src="media/int.png"
descriptor="desInt"/>
  </body>
</ncl>

```

```

        <media id="fondo2" src="media/fondo.png"
descriptor="dFondo2"/>
        <media id="fondoB" src="media/quitoQR.png"
descriptor="dFondoB"/>
        <media id="ima" src="media/ima.png" descriptor="dima"/>
        <media id="imb" src="media/imb.png" descriptor="dimb"/>
        <media id="imc" src="media/imc.png" descriptor="dimc"/>
        <media id="imd" src="media/imd.png" descriptor="dimd"/>
        <media id="verde" src="media/verde.png" descriptor="dverde"/>
        <media id="rojo" src="media/rojo.png" descriptor="drojo"/>
        <!--Script lua para acceso al servidor web remoto para voto y
consulta-->
        <media id="lua" src="votacion2.lua" descriptor="dLua">
            <property name="voto"/>
            <property name="result"/>
        </media>
        <!--Boton azul para inicio de interactividad-->
        <link xconnector="combase#selcdeteclaPararIniciar">
            <bind role="onSelection" component="interatividad">
                <bindParam name="keyCode" value="BLUE"/>
            </bind>
            <bind component="interatividad" role="stop"/>
            <bind component="rojo" role="start"/>
            <bind component="verde" role="start"/>
            <!--<bind component="lua" role="start"/>-->
            <!--<bind component="fondo2" role="start"/>-->
            <bind component="fondoB" role="start"/>
        </link>
        <!--Boton verde para realizar voto-->
        <link xconnector="combase#selcdeteclaPararIniciar">
            <bind role="onSelection" component="verde">
                <bindParam name="keyCode" value="GREEN"/>
            </bind>
            <bind component="rojo" role="stop"/>
            <bind component="verde" role="stop"/>
            <bind component="fondo2" role="start"/>
            <bind component="ima" role="start"/>
            <bind component="imb" role="start"/>
            <bind component="imc" role="start"/>
            <bind component="imd" role="start"/>
            <bind component="lua" role="start"/>
        </link>
        <!--Boton rojo para salir-->
        <link xconnector="combase#selcdeteclaPararIniciar">
            <bind role="onSelection" component="rojo">
                <bindParam name="keyCode" value="RED"/>
            </bind>
            <bind component="verde" role="stop"/>
            <bind component="rojo" role="stop"/>
            <bind component="fondo2" role="stop"/>
            <bind component="fondoB" role="stop"/>
            <bind component="interatividad" role="start"/>
        </link>
        <!--Boton rojo para votar PACO MONCAYO-->
        <link xconnector="combase#onKeySelecionSetStop">
            <bind component="ima" role="onSelection">
                <bindParam name="keyCode" value="RED"/>
            </bind>
            <bind component="lua" role="set" interface="voto">
                <bindParam name="var" value="ca"/>
            </bind>
        </link>

```

```

        <bind component="imb" role="stop"/>
        <bind component="imc" role="stop"/>
        <bind component="imd" role="stop"/>
        <bind component="ima" role="start"/>
</link>
<!--Boton verde para votar JORGE YUNDA-->
<link xconnector="combase#onKeySelecionSetStop">
    <bind component="imb" role="onSelection">
        <bindParam name="keyCode" value="GREEN"/>
    </bind>
    <bind component="lua" role="set" interface="voto">
        <bindParam name="var" value="cb"/>
    </bind>
    <bind component="ima" role="stop"/>
    <bind component="imc" role="stop"/>
    <bind component="imd" role="stop"/>
    <bind component="imb" role="start"/>
</link>
<!--Boton amarillo para votar CARLOS SEVILLA-->
<link xconnector="combase#onKeySelecionSetStop">
    <bind component="imc" role="onSelection">
        <bindParam name="keyCode" value="YELLOW"/>
    </bind>
    <bind component="lua" role="set" interface="voto">
        <bindParam name="var" value="cc"/>
    </bind>
    <bind component="ima" role="stop"/>
    <bind component="imb" role="stop"/>
    <bind component="imd" role="stop"/>
    <bind component="imc" role="start"/>
</link>
<!--Boton azul para votar CESAR MONTUFAR-->
<link xconnector="combase#onKeySelecionSetStop">
    <bind component="imd" role="onSelection">
        <bindParam name="keyCode" value="BLUE"/>
    </bind>
    <bind component="lua" role="set" interface="voto">
        <bindParam name="var" value="cd"/>
    </bind>
    <bind component="ima" role="stop"/>
    <bind component="imb" role="stop"/>
    <bind component="imc" role="stop"/>
    <bind component="imd" role="start"/>
</link>
<!--Cerrar voto-->
<link xconnector="combase#onEndAttributionDelayedStop">
    <bind component="lua" role="onEndAttribution"
interface="voto"/>
    <bind component="lua" role="stop"/>
</link>
<link xconnector="combase#onEndStop">
    <bind role="onEnd" component="lua"/>
    <bind role="stop" component="ima"/>
    <bind role="stop" component="imb"/>
    <bind role="stop" component="imc"/>
    <bind role="stop" component="imd"/>
    <bind role="stop" component="fondo2"/>
    <bind role="stop" component="fondoB"/>
    <bind role="stop" component="lua"/>
</link>
<!--Iniciar nuevamente interactividad-->

```

```
<link xconnector="combase#onEndStart">  
  <bind role="onEnd" component="lua"/>  
  <bind role="start" component="interatividad"/>  
</link>  
</body>  
</ncl>
```