

ESCUELA POLITÉCNICA DEL EJÉRCITO

INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES

PROYECTO DE GRADO

“ANÁLISIS DE DESEMPEÑO DEL CANAL DE RETORNO
BASADO EN EL DESARROLLO Y TRANSMISIÓN DE
APLICACIONES INTERACTIVAS DE TV DIGITAL PARA
EL SISTEMA ISDB-TB”

Autores

ANA CAROLINA PAREDES TERÁN NANCY

MARICELA TONGUINO ALCOSER

SANGOLQUÍ - ECUADOR

2012

CERTIFICACIÓN

Certifican, que el proyecto de grado titulado “ANÁLISIS DE DESEMPEÑO DEL CANAL DE RETORNO BASADO EN EL DESARROLLO Y TRANSMISIÓN DE APLICACIONES INTERACTIVAS DE TV DIGITAL PARA EL SISTEMA ISDB-TB”, realizado por las señoritas Ana Carolina Paredes Terán con CI:180410223-2 y Nancy Maricela Tonguino Alcoser con CI:171999919-3, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Ing. Gonzalo Olmedo Ph.D
DIRECTOR

Ing. Freddy Acosta
CODIRECTOR

RESUMEN

El presente proyecto de grado busca analizar el canal de retorno para la televisión digital terrestre (TDT), a través del desarrollo de la aplicación interactiva de compra y venta denominada ShoppingCar.

Inicialmente se realizó una descripción del estándar ISDB-Tb, adoptado por el Ecuador, el 26 de marzo del 2010 para TDT.

La plataforma para la ejecución de la aplicación interactiva, se basó en el middleware Ginga-Ncl y en Lua para la comunicación a través del canal de retorno.

Para el diseño de la aplicación ShoppingCar, se procedió a realizar la base de datos en MySQL, usando como administrador a phpMyAdmin. Una vez estructurada la base de datos se procede a realizar la comunicación con el Web Service a través de Lua.

Fueron realizadas pruebas de la aplicación interactiva tanto dentro de la intranet donde se encuentra el servidor, como también de forma externa a través de Internet.

Adicionalmente se obtuvo la tasa de transmisión, el ancho de banda requerido, el análisis de los paquetes enviados y recibidos utilizando wireshark. Además, con el software JMeter, se inyectó tráfico en el servidor para poder determinar la capacidad de rendimiento del mismo.

DEDICATORIA

Quiero dedicar este proyecto a los seres más importantes en mi vida, “mis viejitos”, Nancy y Germán, quienes me enseñaron a luchar y ser perseverante, además, quiero dedicar a mis hermanas y a toda mi familia por estar junto a mi y por creer en mi.

Finalmente quiero dedicar a todas las personas que tienen una meta por cumplir, para que nunca pierdan la fe y lo alcancen, ya que todo es posible.

Anita Paredes

DEDICATORIA

El presente proyecto dedico en primer lugar a Dios que durante todo este tiempo de estudio y preparación, me dio la fortaleza, sabiduría y amor, para culminar con éxito mi carrera profesional.

A mis padres Juan y Judith, quienes creyeron en mi y nunca dejaron de apoyarme, dándome ejemplos dignos de superación y entrega, porque gracias a ellos hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y el orgullo y amor que sienten por mi, fue lo que me hizo llegar hasta el final.

A mi hermano Juan David por su amor y apoyo incondicional desde niño, por todos los momentos que hemos compartido de alegrías y tristezas, por el interés y ayuda que me brindó en la etapa final de este proyecto. A mi querida familia tíos, tías, primos y primas, por darme siempre palabras de aliento durante el largo camino que recorrí para llegar a ser una buena profesional.

A mis grandes amigos del “Chulla Vida”, con quienes compartí momentos inolvidables en la etapa universitaria, gracias por haber sido como mi segunda familia y por dejarme ser parte de la vida de cada uno de ustedes.

A mi buen amigo bebé Ricardito, porque siempre estuvo pendiente del proceso de este proyecto, brindándome su ayuda incondicional y desinteresada, motivándome entre risas y lamentos, estoy segura que hoy que estas junto a Dios, podrás ver que tu miguis al fin alcanz tan anhelada meta.

Nancy Tonguino

AGRADECIMIENTOS

Quiero agradecer primero a Dios, por todas las bendiciones que me ha dado. A mis padres y a toda mi familia que siempre estuvieron a mi lado.

Agradezco a mi director y codirector de Tesis, Gonza y Freddy, por creer en mi y darme la oportunidad de aprender en este proyecto de grado.

Agradezco especialmente Ing. Sanvador, que me ofreció su ayuda compartiendo sus conocimientos y colaborandome como lo hacen los grandes amigos, y a los inges de sistemas Ing. Delgado e Ing. Marcillo, que de manera desinteresada me han brindado su colaborado en todo. Gracias a Jeni y todos sus amigos de la UPS por la colaboración y ayuda que me han brindado.

A mis amigos los “Chullas”, mis amigos de la U, mis amigos del cole y los que dejaron huella en mi corazón partiendo de éste mundo, a todos ustedes por ser grandes personas y por creer en mi simplemente gracias. Antes de finalizar quiero extender mi agradecimiento a Javi por confiar siempre en mi y darme tu apoyo incondicional.

A todos simplemente gracias.

Anita Paredes

AGRADECIMIENTOS

Quiero extender mis agradecimientos, al ingeniero Santiago Salvador por la dedicación, sabiduría, paciencia y entusiasmo, por su aporte invaluable para la realización de este proyecto.

Al Ingeniero Gonzalo Olmedo y al Ingeniero Freddy Acosta, por habernos brindado su tiempo y sabiduría, siendo un apoyo constante para culminar con éxito este proyecto.

A los ingenieros Ramiro Delgado y Diego Marcillo, quienes en todo momento nos brindaron su valiosa ayuda.

Nancy Tinguino

PRÓLOGO

En la actualidad se vive una transformación en el servicio de televisión terrestre radiodifundida, donde la televisión digital representa la revolución en la transmisión de programas junto a una gran flexibilidad en los contenidos emitidos. Ésta nueva modalidad permite al televidente interactuar con su televisión ya sea en equipos fijos, portátiles o móviles, con la característica que el televidente ahora puede disfrutar de una señal mejorada siendo esta más robusta sin ruidos, interferencias, doble imagen, de ésta forma, se perfecciona la calidad de los contenidos presentados al telespectador.

El Ecuador adoptó el estándar ISDB-Tb en el 2010, y al entrar en ésta nueva era, con nuevos retos por impulsar. El plan Nacional del Buen Vivir como lo estipula el Consejo Nacional de Planificación, se debe transformar la educación superior en base a la ciencia, tecnología e innovación; es por esto que la Escuela Politécnica del Ejército se impone metas para que sus alumnos den respuesta a las necesidades más urgentes y una de las razones más urgentes a desarrollarse en el país es el tema de la educación, que la población tenga acceso a fuentes de información de una manera rápida.

Uno de los desafíos que tenemos que enfrentar es la aceptación de la sociedad a esta nueva etapa que viene cargada con tecnología y equipos nuevos, los cuales deben cumplir con todos los requerimientos y necesidades de la sociedad tanto para su entretenimiento, educación, información, salud, redes sociales, etc., pero dicha tecnología debe de hacerse de forma sencilla y dinámica para que todos los ecuatorianos puedan acceder de forma fácil y ágilmente.

Actualmente, la Escuela Politécnica del Ejército, en un esfuerzo sin precedente y siendo pioneros en la investigación de la televisión digital terrestre en el Ecuador, se ha esmerado por impulsar nuevos proyectos que vayan de la mano con el estándar para televisión digital Brasileño - Japonés (ISDB-Tb), adoptado por el Ecuador, por medio de pruebas, en las que intervienen equipos adquiridos por la Institución.

El objetivo del presente proyecto de grado se basa en un análisis referente a un canal de retorno utilizando una red de datos y el middleware Ginga NCL. El principal aporte del canal de retorno es lograr una comunicación entre el usuario y el middleware, haciendo que la experiencia de ver televisión sea mucho más interactiva. Además contempla un adecuado manejo de base de datos y pruebas de transmisión en el proceso de comunicación, lo cual justifica la importancia del mismo en el desarrollo y progreso del país en materia de televisión digital.

Índice

Glosario	XIX
1. INTRODUCCIÓN	1
1.1. Introduccion	1
1.2. Alcance	2
1.3. Objetivos	2
1.3.1. Objetivo General	3
1.3.2. Objetivos Específicos	3
1.4. Organizacion del Documento	3
2. INTERACTIVIDAD EN LA TELEVISION DIGITAL TERRESTRE	5
2.1. Digital Video Broadcasting Terrestrial (DVB-T)	5
2.2. Advanced Television System Commite (ATSC)	6
2.3. Digital Terrestrial Multimedia Broadcast (DTMB)	6
2.4. Integrated Service Digital Broadcasting Terrestrial (ISDB-T)	7
2.4.1. Sistema de transmision en modo jerárquico	8
2.4.2. Principales Características del estandar ISDB-T	8
2.5. Televisión digital en el Ecuador	9

3. INTERACTIVIDAD EN LA TELEVISIÓN DIGITAL	11
3.1. Definición	11
3.2. Tipos de Interactividad	11
3.2.1. Interactividad Local	11
3.2.2. Interactividad Remota o con Canal de Retorno	12
3.3. Middleware	13
3.3.1. Definición	13
3.3.1.1. Middleware para Interactividad	14
3.4. GINGA	15
3.4.1. Tipos de GINGA	15
3.4.2. GINGA J	16
3.4.2.1. Definición	16
3.4.2.2. Arquitectura	16
3.4.2.3. Requerimientos	16
3.4.3. GINGA NCL	17
3.4.3.1. Definición	17
3.4.3.2. Arquitectura	18
3.4.3.3. Requerimientos del Software	18
3.4.3.4. Lenguaje de Programación en GINGA NCL	18
3.4.4. Cuadro comparativo entre GINGA J y GINGA NCL	20
3.4.5. LUA	20
3.4.5.1. Cómo usar LUA	21
3.4.5.2. Elementos de LUA	22

4. CANAL DE RETORNO EN TV DIGITAL	26
4.1. ¿Qué es el Canal de Retorno?	26
4.2. Tipos de Canales de Retorno	27
4.2.1. Satélite	27
4.2.2. Asymmetric Digital Subscriber LineADSL	28
4.2.3. Cable	29
4.2.4. Universal Mobile Telecommunications System(UMTS)	31
4.2.5. Local Multipoint Distribution Service(LMDS)	32
4.3. Funcionamiento del Canal de Retorno	33
4.3.1. Simulación en Virtual Set Top Box	33
4.3.2. Implementacion en el Set Top Box para TV Digital	33
5. BASE DE DATOS PARA APLICACIONES INTERACTIVAS	36
5.1. Bases de Datos	36
5.2. Bases de Datos	37
5.2.1. Bases de Datos Estaticas	37
5.2.2. Bases de Datos Dinámicas	37
5.3. Funcionamiento de la Bases de Datos y la Interactividad	37
5.3.1. Creación de la Base de Datos	38
5.3.2. Diseño de la Base de Datos	39
5.4. Conexion a la Base de Datos a través de un Web Service	41
5.5. Conexion del Servidor PHP con la Base de Datos	43
5.5.1. Estructura de Programacion	44

5.6. Conexion a la Base de Datos desde Lua a través de un Web Service	46
5.7. Programacion en Lua para consumir un Web Service	49
5.8. Entorno de Programacion en GINGA-NCL	51
5.9. Entorno de Programacion en LUA	56
5.10. Corrida de la Aplicacion ShoppingCar	60
6. ANÁLISIS DE RESULTADOS DEL CANAL DE RETORNO	62
6.1. Analisis del Canal de Retorno	62
6.2. Analisis del Rendimiento del Web Service	77
7. CONCLUSIONES Y RECOMENDACIONES	79
Bibliografía	81

Índice de Figuras

2.1. Estándares Adoptados por los países	5
2.2. Sistema de Transmisión en Modo Jerárquico	8
3.1. Tipos de Interactividad	12
3.2. Perfiles DVB-MHP	13
3.3. Middleware	14
3.4. Middleware	16
3.5. Arquitectura de GINGA-NCL	18
3.6. Palabras reservadas en LUA	22
3.7. Listado de strings	23
4.1. Arquitectura de la Red de Acceso xDLS	29
4.2. Distribución de Servicios	30
4.3. Esquema de una Red con Cable Bidireccional	31
4.4. Arquitectura del UMTS	32
4.5. Arquitectura del LMDS	32
4.6. Escenario de prueba (simulación) para la interactividad con canal de retorno	33
4.7. Escenario de prueba real para la interactividad con canal de retorno	34
4.8. Emisión del programa de compras para TDT	34

5.1. Proceso jerárquico para instalar phpMyAdmin	39
5.2. Tablas creadas para ShoppingCar	41
5.3. Funcionamiento del protocolo SOAP	42
5.4. Funcionamiento del protocolo HTTP	43
5.5. Esquema para establecer la conexión entre NCL-LUA y la base de datos MySQL a través del consumo un Web Service	44
5.6. Contenido del Proyecto appEjemplo.php y visualización de la clase clsAccesoDatos.php.	46
5.7. Web Service wsCar.php	47
5.8. Código de Programa. Librerías y Clases requeridas por el Web Service wsCar.php	48
5.9. Código de Programa. Instanciación de objetos, namespace y variables en el Web Service.	49
5.10. Código de Programa. Librerías y clases requeridas por el Web Service wsCar.php Función insertarPedido del Web Service.	50
5.11. Código de Programa. Añadiendo ncluasoap al programa de Lua.	50
5.12. Código de Programa. Variables locales dentro de la programación de Lua.	51
5.13. Código de Programa. Llamada a ncluasoap.	51
5.14. Código de Programa. Conexión con el servidor desde Lua	51
5.15. Proyecto NCL TESIS COMPRA	52
5.16. Código de Programa. Regiones creadas en main.ncl.	53
5.17. Código de Programa. Descriptores	53
5.18. Código de Programa. Conectores que serán utilizados para reproducir los nodos multimedia.	54
5.19. Código de Programa. Ports definidos en main.ncl	54
5.20. Código de Programa. Media definidos en main.ncl	55

5.21. Código de Programa. Programación asociada al evento Presionar Botón Rojo, donde se inicializará el archivo lua1.lua.	56
5.22. Código de Programa. Programación asociada al evento Presionar Botón Azul, donde se inicializar el archivo lua2.lua.	56
5.23. Código de Programa. Programación asociada al evento Presionar Botón Amarillo, donde se inicializar el archivo lua3.lua.	57
5.24. Código de Programa. Programación para salir de la aplicación interactiva.	57
5.25. Código de Programa. Parámetros correspondientes al primer producto, declaradas en el archivo lua1.lua.	58
5.26. Código de Programa. Parámetros correspondientes al segundo producto, declaradas en el archivo lua2.lua.	59
5.27. Código de Programa. Parámetros correspondientes al tercer producto, declaradas en el archivo lua3.lua.	59
5.28. Aplicación ShoppingCar, con las dos opciones	60
5.29. Aplicación ShoppingCar, con los tres productos de compra	61
6.1. Estructura de red para el uso del Canal de Retorno.	62
6.2. Estructura de red para el uso de Canal de Retorno, dentro de la Escuela Politécnica del Ejército.	63
6.3. Captura del Protocolo IPv4, en el tráfico de red durante la ejecución de la aplicacion.	64
6.4. Captura del Protocolo TCP, en el tráfico de red durante la ejecución de la aplicacion.	64
6.5. Captura de solicitud HTTP, en el trafico de red durante la ejecucion de la aplicacion.	65
6.6. Captura del Protocolo XML, en el tráfico de red durante la solicitud al servidor.	65
6.7. Captura de respuesta HTTP, en el tráfico de red durante la ejecución de la aplicacion.	66

6.8. Captura del protocolo XML, en el tráfico de red durante la respuesta del servidor.	67
6.9. Captura del Trafico de Red.	67
6.10. Captura del Protocolo HTTP en la corrida de la aplicacion.	68
6.11. Captura de la transmision de datos desde el host hasta el servidor.	69
6.12. Captura de la transmision de datos desde el servidor hasta el host.	69
6.13. Captura del envío y recepcion de paquetes.	70
6.14. Captura de la tasa de transmision para el caso A.	70
6.15. Análisis del Throughput Caso A.	71
6.16. Estructura de red para el uso de Canal de Retorno, desde un domicilio ubicado en Quito.	71
6.17. Captura del Tráfico de Red con la red 192.168.1.3	72
6.18. Captura del Protocolo http en la corrida de la aplicacion.	73
6.19. Captura de la transmisión de datos desde el host (192.168.1.3) hasta el servidor	74
6.20. Captura de la transmisión de datos desde el servidor hasta el host (192.168.1.3).	74
6.21. Captura del envío y recepcion de paquetes para el caso B	75
6.22. Captura de la tasa de transmision para el caso B.	75
6.23. Análisis del Throughput Caso B.	76

Índice de Tablas

3.1. Cuadro Comparativo entre Ginga NCL vs Ginga J	20
5.1. Diseño de la Base de Datos para ShoppingCar	40
5.2. Pila de Protocolos de los Web Service	47
6.1. Resultados obtenidos con 10000 peticiones	77
6.2. Resultados obtenidos con 40000 peticiones	78

Glosario

A

- ACAP Advanced Common Application Platform , p. 14.
- ACK ACKNOWLEDGEMENT , p. 72.
- ADSL Asymmetric Digital Subscriber Line, p. 26.
- API Application Programing Interface , p. 16.
- ARIB Association of Radio Industries Businesses , p. 7.
- ATM Asynchronous Transfer Mode, p. 28.
- ATSC Advanced Television System Commite, p. 6.

C

- COFDM Coded Orthogonal Frecuency Division Multiplexing, p. 5.
- Conatel Consejo Nacional de Telecomunicaciones , p. 9.

D

- DTMB Digital Terrestrial Multimedia Broadcast, p. 7.
- DVB-T Digital Video Broadcasting Terrestrial, p. 5.

G

- GEM Globally Executable MHP , p. 15.
- GSM Global System for Mobile Communications , p. 31.

H

- HD High Definition , p. 9.
- HDTV High Definition Television, p. 6.
- HTML HyperText Markup Language, p. 13.

I

- IDTV Interactive Digital Television, p. 1.
- IP Internet Protocol , p. 28.
- ISDB-T Integrated Service Digital Broadcasting Terrestrial, p. 7.
- ISDB-Tb Integrated Services Digital Broadcasting, p. 1.

L

- LMDS Local Multipoint Distribution Service, p. xii.

M

- MPEG Moving Pictures Experts Group, p. 5.
- MySQL Lenguaje Estructurado de Consulta , p. 45.

P

- PHP Hypertext Pre-processor, p. 43.

Q

- QAM Quadrature amplitude modulation, p. 6.

R

- RDSI Red Digital de Servicios Integrados , p. 28.

S

SBTVD Sistema Brasileño de TV Digital , p. 15.

SD Standar Definition , p. 9.

SOAP Simple Object Access Protocol , p. 42.

SYN Selective Synchronous Recording , p. 72.

T

TCP Protocolo de control de transmision/Protocolo de Internet, p. 72.

TDS-OFDM Time Domain Synchronuous Orthogonal Frequency División Multiplexing,
p. 7.

TDT Television Digital Terrestre, p. 1.

U

UDDI Universal Description, Discovery and Integration Resgistry , p. 48.

UMTS Universal Mobile Telecommunications System, p. 31.

UTRAN Terrestrial Radio Access Network, p. 31.

V

VSF Vestigial Sideband. Modulation, p. 6.

W

WSDL Web Services Description Language, p. 42.

X

XML eXtensible Markup Language, p. 17.

Capítulo 1

INTRODUCCIÓN

1.1. Introducción

La televisión es una fuente indispensable de información y entretenimiento en nuestros días. A pesar que el avance de internet como sustituto de estas características está creciendo considerablemente, la televisión continúa evolucionando para conseguir adecuarse a las necesidades emergentes. Como resultado disponemos cada vez de mayor calidad y usabilidad, resultando la combinación de ambas en lo que se conoce como televisión digital interactiva (IDTV).

La Televisión Digital Terrestre (TDT) implica una forma distinta de transmitir la señal televisiva, en referencia a la televisión analógica actual. El 26 de marzo del 2010, el Ecuador adoptó oficialmente el estándar japonés-brasileño, Integrated Services Digital Broadcasting (ISDB-Tb) para la televisión digital terrestre [1]. Paulatinamente, el país migrará sus transmisiones desde analógico a digital siguiendo un plan de migración que definir el tiempo en que los transmisores analógicos serán apagados.

La televisión digital representa el cambio tecnológico más radical en la industria televisiva, después de la aparición de la TV a color. La digitalización de la televisión permitirá la mejora de la calidad de la imagen y del sonido, el aumento de la oferta de canales de televisión, alta definición y otras características como guía de programa e interactividad haciendo del aparato televisor una terminal multimedia de mejores características [2].

Uno de los beneficios de la introducción de la TV digital es un mejor aprovechamiento del espectro radioeléctrico debido a las técnicas de transmisión digital empleadas, que permiten aumentar el número de programas transmitidos. Por otro lado, se elimina la

vulnerabilidad a interferencias que tiene la señal analógica con los canales adyacentes. El empleo de técnicas digitales de transmisión ofrece mayor robustez a los sistemas frente a interferencias. Aumenta así la posibilidad de emitir con mejores calidades de imagen y de sonido y con prestaciones más avanzadas.

En la televisión digital existen menores costos de transmisión, aunque en una fase inicial los canales de televisión deberán asumir el costo de modernizar sus equipos, a la larga la transmisión de programas empleando tecnología digital resulta menos costosa, ya que, entre otras razones, permite un uso más eficiente de la potencia de emisión de los transmisores. No obstante, la recepción de la televisión digital implicará costos por parte de los ciudadanos, como es la compra de un decodificador o de un televisor integrado.

Además permite la prestación de servicios interactivos, la interactividad aprovecha las capacidades tecnológicas disponibles de la televisión digital y, muy especialmente la comunicación bidireccional, para ofrecer una experiencia distinta al telespectador, quien desempeña un rol mucho más activo. Esto, obviamente enriquece enormemente el potencial de entretenimiento de la televisión. En un mercado como el ecuatoriano, basado fundamentalmente en los ingresos publicitarios, la aparición de nuevos agentes y nuevos canales, provocará una mayor competencia por la publicidad e impulsará nuevos modelos de negocio.

1.2. Alcance

El presente proyecto contempla el análisis del canal de retorno para el desarrollo de aplicaciones interactivas en tiempo real. Este proyecto permitirá profundizar el estudio de herramientas fundamentales sobre bases de datos, middleware para Ginga en plataformas NCL y Java, que servirán como soporte para producir dichas aplicaciones bajo el estándar ISDB-Tb, las cuales se realizarán pruebas en el Set-Top-Box. Se realizará el respectivo análisis de retardos en la comunicación.

1.3. Objetivos

Para cumplir con el alcance propuesto se han planteado los siguientes objetivos.

1.3.1. Objetivo General

Analizar el desempeño del canal de retorno para el desarrollo y transmisión de aplicaciones interactivas en televisión digital.

1.3.2. Objetivos Específicos

Los objetivos específicos son:

1. Integrar la información sobre bases de datos con middleware, para realizar aplicaciones las cuales sean una base en la investigación y desarrollo de la televisión digital.
2. Analizar el tráfico de red en el proceso de comunicación para poder dar soluciones en la ejecución de programas.
3. Profundizar la investigación del middleware Ginga para realizar aplicaciones interactivas en la plataforma NCL.
4. Investigar y profundizar TCP - LUA para realizar programas aplicativos y el respectivo análisis para la comunicación.
5. Realizar pruebas con los parámetros que establece el estándar ISDB-Tb para el Transport Stream aplicables en el canal de retorno y la red.
6. Desarrollar aplicaciones interactivas en GINGA NCL para las pruebas de canal de retorno en la TV Digital.
7. Realizar la carga de procesamiento de la aplicación.
8. Pruebas de ancho de bandas.

1.4. Organización del Documento

El presente proyecto de grado se encuentra distribuido en siete capítulos, a continuación se detalla una explicación de cada capítulo:

- En el Capítulo 2, se detallan los fundamentos teóricos de los estándares de la televisión digital; se realiza un análisis del estándar Brasileño-Japonés ISDB-Tb. El

estudio incluye aspectos como las características del estándar, las características y la modulación que maneja.

- En el Capítulo 3, se detalla los fundamentos teóricos sobre la interactividad en la televisión digital, además se detalla el estudio del middleware Ginga con su plataforma NCL, CC y Java. Se especifica el estudio en LUA, con sus parámetros y su estructura en la programación.
- En el Capítulo 4, se detalla los fundamentos teóricos del canal de retorno, además se realiza un análisis de las tecnologías existentes para la implementación del canal de retorno, como satélite, ADLS, cable, UMTS, entre otras.
- En el Capítulo 5, se detalla los fundamentos teóricos de base de datos, además de la estructura de como realizarla, los protocolos que interviene para la conexión con la base de datos, la conexión con el servidor y el desarrollo de la aplicación en NCL, LUA y MySQL.
- En el Capítulo 6, se realizan las pruebas en la máquina virtual utilizando el software Wireshark, para el análisis de la transmisión de la aplicación y examinar parámetros como paquetes caídos, el envío de paquetes, la tasa de transmisión y el ancho de banda que necesita la aplicación.
- En el Capítulo 7, se detallan las conclusiones del trabajo completado en el proyecto y se formulan las respectivas recomendaciones para la futura implementación de la televisión digital.

Capítulo 2

INTERACTIVIDAD EN LA TELEVISION DIGITAL TERRESTRE

Los estándares de TDT existentes han sido aprobados y adoptados en los diferentes países de acuerdo a sus características técnicas y servicios que ofrecen, dependiendo también de la ubicación geográfica. En la Figura 2.1, se observa los estándares adoptados en el mundo.

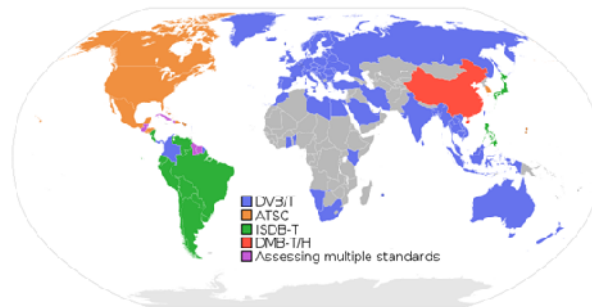


Figura 2.1: Estándares Adoptados por los países

A continuación se presentará una breve introducción a los estándares para televisión digital terrestre.

2.1. Digital Video Broadcasting Terrestrial (DVB-T)

Es el estándar para la transmisión de TDT, creado por la organización europea DVB. Éste sistema transmite audio, video y otros datos a través de un flujo MPEG-2, usando una modulación COFDM (Coded Orthogonal Frequency Division Multiplexing), en el cual se desarrolló para transmitir por cable y/o por satélite. El DVB-T fue adoptado por

la comunidad Europea, Oceanía, Colombia, Panamá, Trinidad y Tobago, Groenlandia algunos países sudafricanos y países del sur de África.

Como se habló anteriormente la modulación que utiliza es COFDM, que consta de dos partes, la primera consiste en dividir o multiplexar la frecuencia pasa banda del canal en muchas sub frecuencias. A continuación, en la segunda parte modula cada sub-frecuencia por un método tradicional, concretamente empleando modulación de fase en cuadratura QPSK, o modulación mixta de amplitud y fase en cuadratura QAM [3]; los parámetros que emplea pueden ser ajustables, lo cual permite tener el control sobre las señales y los posibles problemas que se presenten.

Es un estándar robusto con respecto al ruido y al multitrayecto. Es muy versátil y eficiente ya que la transmisión es muy flexible.

2.2. Advanced Television System Commite(ATSC)

Fue desarrollado por la Gran Alianza, un consorcio de compañías electrónicas y de telecomunicaciones, las cuales se reunieron para desarrollar un High Definition Television HDTV. Los formatos de ATSC también incluyen definición estándar, aunque los servicios de televisión de alta definición al principio sólo se pusieron en marcha en el formato digital. ATSC es un estándar norteamericano, el mismo que fue adoptado en varios países como USA, Canadá, Corea del Sur y parte de Centro América como México, Honduras, República Dominicana y El Salvador. ATSC emplea una modulación 8-VSB, Banda Lateral Vestigial, la cual se basa en la modulación 8-QAM, y 64-QAM con una codificación de Trellis, que permite veracidad y velocidad en las transmisiones a realizar. Se emplea MPEG-2, para la compresión de imagen y sonido, lo que significa que realiza una comparación entre los fotogramas actuales y los anteriores almacenando solo los cambios existentes. En 2012, ATSC 2.0 puede ser un estándar. Está destinado a ser usado con el estándar actual, pero permitirá que los servicios interactivos, incluyendo el video bajo demanda de servicios, la publicidad dirigida, MPEG de compresión, más guías de programación, y la capacidad de almacenar información sobre los nuevos receptores [4].

2.3. Digital Terrestrial Multimedia Broadcast(DTMB)

Es una tecnología de transmisión digital tanto de video, audio y datos para comunicaciones móviles y portátiles, está diseñada para diferentes tipos de aplicaciones como pueden

ser los teléfonos móviles, PDAs o sistemas de instrumentación para el automóvil. Éste estándar es utilizado en países como República Popular de China, Hong Kong y Macao, la modulación que utiliza es la TDS-OFDM (Time Domain Synchronous Orthogonal Frequency Division Multiplexing) lo que permite transmitir señales para receptores HDTV y para la multiplexación de audio y video utiliza el MPEG-2 o MPEG-4. Para obtener el estándar DTMB se unió algunas normas como ADTB-T, la misma que fue desarrollada por la universidad de Shanghai Jiao Tong, de Shanghai, DMB-T, la que fue desarrollada por la Universidad de Tsinghua, Beijing y finalmente TIMI [5].

2.4. Integrated Service Digital Broadcasting Terrestrial (ISDB-T)

Los inicios del estándar se formaron en ARIB (Asociación de Industrias y Empresas de Radiodifusión), el cual es un organismo de normalización en Japón, designado al estudio del uso eficiente del espectro radioeléctrico, en donde, alguna de sus actividades es la investigación y desarrollo de sistemas de radio y de radiodifusión.

El ARIB, es la entidad encargada de crear y mantener el estándar ISDB-T. Este estándar es adoptado en Japón en el 2003 y realiza transmisiones de radio y televisión digital. La norma ISDB está conformada por una familia de componentes como son la TDT (ISDB-T o ISDB-Tb), televisión satelital, televisión por cable, servicios de multimedia y radio digital.

La modulación que utiliza el estándar es OFDM en bandas segmentadas, logrando una transmisión jerárquica en la cual se emplean distintos esquemas de modulación (QPSK, DQPSK, 16QPSK, 64QPSK), además este estándar utiliza la multiplexación MPEG-2 los cuales son capaces de entregar televisión de alta definición. Tanto ISDB-T como ISDB-Tb permiten recepción de móviles en bandas de TV. 1seg es el nombre de un servicio ISDB-T para recepción en telefonía móvil, computadores portátiles y vehículos. En Junio del 2006, Brasil adoptó el estándar ISDB-T, pero con algunas modificaciones tales como H.264 o MPEG-4 (es una norma que define un códec de vídeo de alta compresión) para codificación de SDTV y HDTV, así como también utiliza tecnologías de middleware como es la plataforma GINGA [6].

2.4.1. Sistema de transmisión en modo jerárquico

El principio de utilizar la modulación OFDM es de dividir en multi-portadoras y enviar datos digitales, mientras que la transmisión OFDM segmentada, permite la transmisión de diferentes señales en un mismo ancho de banda de 6MHz a lo que se le conoce como transmisión en modo jerárquico, con lo cual las portadoras están agrupadas en 13 segmentos, como se observa en la Figura 2.2, donde uno de los segmentos es utilizado específicamente para mandar la señal en baja deficiencia para equipos portátiles o teléfonos celulares, sin que la transmisión sea perjudicada.

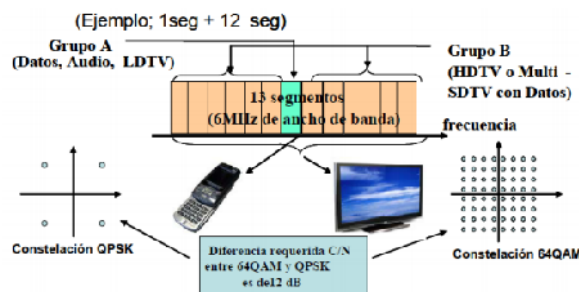


Figura 2.2: Sistema de Transmisión en Modo Jerárquico

Como se puede observar, para la transmisión de 1 grupo se necesita parámetros de transmisión más fuertes por lo que se usa QPSK, mientras para el grupo de 12, se utiliza una recepción fija con una mayor velocidad por lo que se utiliza una modulación de 64QAM.

2.4.2. Principales Características del estándar ISDB-T

Permite transmitir un canal HDTV y un canal para equipos portátiles dentro de un ancho de banda de 6 MHz o seleccionar la transmisión entre dos y tres canales de televisión en definición estándar (SDTV) en lugar de uno solo en HDTV, mediante el multiplexado de canales SDTV.

Proporciona servicios interactivos con transmisión de datos, como juegos o compras, vía línea telefónica o Internet de banda ancha. Además soporta acceso a Internet como un canal de retorno. El acceso a Internet también es provisto en teléfonos móviles. Esto se presentará con detalle en el Capítulo 3.

Presenta una buena robustez contra multi-pfad, ruidos urbanos, desvanecimiento y cualquier otra interferencia, utilizando efectivamente el uso de frecuencias y permite

interactuar con el usuario.

2.5. Televisión digital en el Ecuador

El Gobierno Ecuatoriano, el 26 de marzo del 2010 anunció que el Consejo Nacional de Telecomunicaciones (Conatel) aceptó la recomendación de la Superintendencia de Telecomunicaciones que se inclinó por la norma japonesa-brasileña de televisión digital (ISDB-Tb) siendo en consecuencia adoptada oficialmente como norma de televisión digital terrestre en nuestro país, ofreciendo televisión abierta gratuita con mejores prestaciones.

Algunas de las razones para la selección del estándar de televisión digital terrestre ISDB-Tb, adoptado en el Ecuador:

- Por la posibilidad de que su aplicación pueda masificarse aceleradamente y, por lo tanto, los equipos que la produzcan tengan menores costos.
- Por la alta calidad de sus modalidades de alta definición (HD) por las siglas del inglés de High Definition y definición estándar (SD) Standar Definition, así como por su eficiencia en el uso del espectro (portabilidad) y su gran capacidad para la convergencia de servicios (movilidad).
- Por sus menores costos de adopción con mayores beneficios netos para la sociedad, resultado de la suma de beneficios de los consumidores y radiodifusores.
- Por permitir la coexistencia de las señales HD y SD, posibilitando que los radiodifusores puedan transmitir simultáneamente a través un canal de HD y/o varios canales de SD.
- Por su fácil interconexión con computadoras personales y portátiles (Laptop y Network) y, por consiguiente, con Internet. De este modo el sistema ISDB-Tb es de doble vía o interactiva, facilidad que hace que el televidente pueda desempeñar un papel más activo[7].
- Mejor imagen y sonido.
- Mayor oferta de programas, pues cada canal podrá transmitir hasta ocho canales digitales de televisión.
- Nuevos servicios interactivos y multimedia, como correo electrónico, información de cotizaciones de bolsa, guías electrónicas de programas, vídeo pagado, etc.

- Recepción de televisión en dispositivos móviles como teléfonos celulares y receptores en vehículos en movimiento, en un solo canal.
- Oferta de múltiples contenidos de programación y servicios multimedia añadidos.
- Uso de canales adyacentes por óptimo manejo del espectro radioeléctrico.
- Oportunidad productiva y comercial para la industria por demanda de decodificadores y receptores por mejora y renovación del parque televisivo.
- Desarrollo de nuevas capacidades en nuevas tecnologías de producción, registro de programas regulares e interactivos, edición, post producción, transmisión y recepción de la televisión digital.
- Promoción de nuevos negocios para la producción de contenidos de entretenimiento, servicios informativos y de publicidad.
- Fomento de la Sociedad de la Información, lo que implica mayor inclusión económica y social, reducción de la brecha digital e integración nacional.
- Promoción de la universalidad del acceso al conocimiento (Educación, ciencia y cultura) [8].

Capítulo 3

INTERACTIVIDAD EN LA TELEVISIÓN DIGITAL

3.1. Definición

La interactividad es la capacidad de ofrecer al telespectador la posibilidad de personalizar contenidos adicionales a los programas de televisión, permitiendo al usuario ver informaciones asociadas al contenido audiovisual, la programación de los canales, participar en concursos, votaciones, comprar productos o servicios, e incluso participar en los propios programas de televisión con el mando a distancia. La interactividad es posible gracias a aplicaciones que complementan la programación, la información puede ser enviada durante el proceso de emisión pero que sólo se hace visible si el espectador lo desea, o bien se puede intercambiar información accediendo a servidores, a través de un canal de retorno utilizando el televisor como interfaz de salida.

3.2. Tipos de Interactividad

La interactividad puede ser de dos tipos, en la Figura 3.1, se observa la diferencia entre la interactividad local que es la que permite que el espectador interactúe con la información en el set-top-box, y la interactividad remota o con canal de retorno además de permitir la interactividad permite tener canal de retorno por medio de internet.

3.2.1. Interactividad Local

Permite que el espectador interactúe con la información que fue enviada por el Broadcast y que está almacenada en el receptor, la cual se actualice cada cierto tiempo. Con la interactividad local, el usuario puede acceder a contenidos interactivos pero no puede

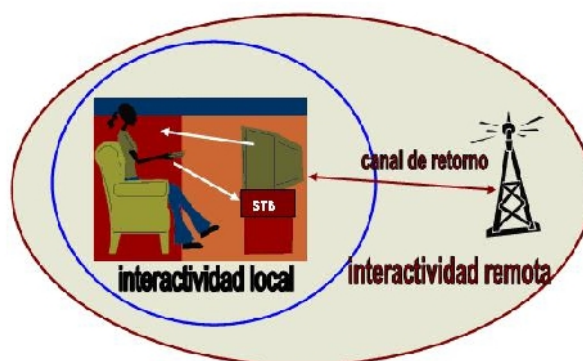


Figura 3.1: Tipos de Interactividad

enviar datos de vuelta.

Ejemplos de aplicaciones interactivas locales son las Guías de Programación, el Teletexto Digital, o información sobre los participantes en un programa

3.2.2. Interactividad Remota o con Canal de Retorno

La interactividad con canal de retorno permite no solo ver contenidos adicionales a la programación y navegar por ellos, sino también enviar respuestas por parte de los usuarios, e incluso comunicarse con otros usuarios. La interactividad con canal de retorno es la que permite a los usuarios participar en concursos, votar, o enviar mensajes a otros usuarios.

El espectador interactúa con un proveedor de servicios exterior, al que se conecta mediante un canal de retorno. Para ello precisa de un decodificador con la especificación MHP “Interactive Broadcasting”, en la que se pueden proporcionar servicios interactivos que pueden o no estar asociados a los servicios de vídeo y audio ofrecidos por el operador, y que permiten una interactividad completa. En este caso, la plataforma está dotada de un canal de retorno, con objeto de realizar la comunicación entre los decodificadores y la cabecera. Los servicios asociados a la interactividad remota se resumen a continuación:

- EPG Avanzada, Teletexto mejorado, Información Básica de Meteorología, Bursatil, Aeropuertos, tráfico.
- Información tipo carrusel emitida toda ella por el operador Información Detallada y particularizada de Meteorología, Bursatil, Aeropuertos, tráfico.
- Telecompra, TeleAdministración, Votaciones, encuestas.

- Información selectiva recibida en todo o en parte por el canal de retorno

Finalmente se tiene la interactividad plena, en la cual el espectador tiene acceso a Internet, cuyo objetivo es permitir proveer a los usuarios de servicios Internet, y que como nota destacada contiene un elemento HTML opcional denominado DVB-HTML. En la Figura 3.2, se muestra los componentes para tener un acceso a Internet y los protocolos que se necesita para brindar al usuario de los servicios de Internet.

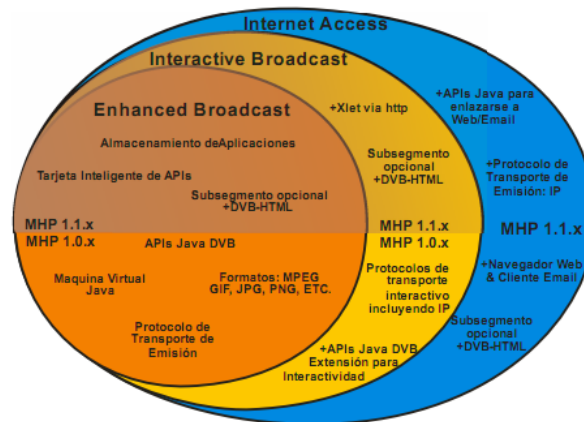


Figura 3.2: Perfiles DVB-MHP

3.3. Middleware

3.3.1. Definición

Middleware es un conjunto de software ubicado entre el código de las aplicaciones y la infraestructura de ejecución (plataforma de hardware y sistema operativo), tal y como se muestra en la Figura 3.3.

El middleware esconde las peculiaridades y complejidades del hardware, del sistema operativo, de los drivers de los dispositivos, del software y hardware encargados de la decodificación de la señal. Es una capa de comunicación entre la capa de aplicación y los servicios ofrecidos por las capas inferiores.

Un middleware para aplicaciones de TV digital consta de máquinas de ejecución de los lenguajes ofrecidos y librerías de funciones, que permiten un desarrollo fácil y rápido de aplicaciones para la TV, como se observa en la Figura 3.3.

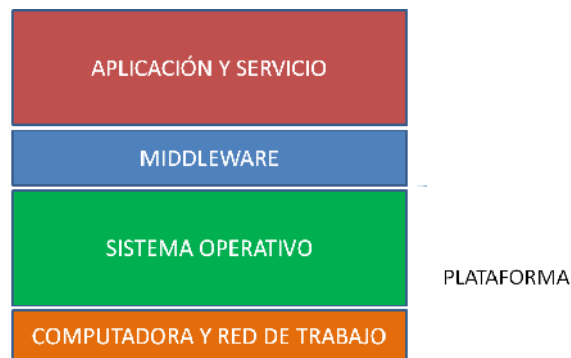


Figura 3.3: Middleware

3.3.1.1. Middleware para Interactividad

En el mercado existen varios middlewares. Se destacan los cinco patrones de middleware desarrollados en función del beneficio de la adopción de esta capa.

- Multimedia Home Platform (MHP)

MHP define una plataforma común para las aplicaciones interactivas de la televisión digital, independiente tanto del proveedor de servicios interactivos como del receptor de televisión utilizado. De este modo, MHP favorece la creación de un mercado horizontal donde aplicaciones, red de transmisión y terminales MHP pueden ser suministrados por proveedores o fabricantes independientes [9].

- Association of Radio Industries Businesses (ARIB)

La ARIB, organización voluntaria que congrega una multitud de empresas dedicadas al negocio de producir, financiar, fabricar, importar y exportar productos relacionados con la radiodifusión, se encarga de regir, crear y mantener los estándares de emisión de la señal audiovisual en el área japonesa [10].

- Advanced Common Application Platform (ACAP)

Esta plataforma fue creada por el Comité de Sistemas Avanzados de Televisión (ATSC), como base común para todos los sistemas de TV interactivos en USA, ya sean por cable,

terrestres o por satélite. Está también basada en GEM y añade algunos elementos de OCAP que son adecuados para el mercado de USA [11].

- Ginga

Ginga es el nombre del middleware abierto del Sistema Brasileño de TV Digital (SBTVD). Permite la posibilidad de poder presentar los contenidos en distintos receptores.

- Globally Executable MHP (GEM)

Globally Executable Multimedia Home Platform (GEM) permite que otros cuerpos de estandarización u organizaciones pudieran definir unas especificaciones basadas en el estándar MHP. GEM, basada en MHP versión 1.0.2, elimina los elementos específicos orientados a DVB, permitiendo así su sustitución por otros más adecuados según la aplicación [12].

3.4. GINGA

Ginga es el fruto del desarrollo de proyectos de investigación coordinados por los laboratorios Telemídia de la PUC-Rio y LAViD de la UFPB.

Ginga, es el nombre del Middleware Abierto del Sistema Nipo-Brasileño de TV Digital (ISDB-Tb), el mismo que está formado por un conjunto de tecnologías estandarizadas e innovaciones brasileñas que lo convierten en la especificación de middleware más avanzada, el cual ofrece una serie de facilidades para el desenvolvimiento de contenidos y aplicaciones para TV Digital, permitiendo presentar los contenidos en distintos receptores independientes de la plataforma de hardware y el tipo de receptor.

3.4.1. Tipos de GINGA

Las aplicaciones realizadas en Ginga, se pueden clasificar dependiendo de la forma en la que son escritas, esto puede ser el lenguaje Java llamado Ginga-J y las Aplicaciones Declarativas son escritas en el lenguaje NCL llamado Ginga NCL.

Una arquitectura de implementación de referencia del middleware Ginga puede ser dividida en tres grandes módulos: Ginga-CC (Common Core), el ambiente de presentación Ginga-NCL (declarativo) y el ambiente de ejecución Ginga-J (de procedimiento).

3.4.2. GINGA J

3.4.2.1. Definición

Ginga-J es parte del subsistema lógico del sistema Ginga, el mismo que procesa el contenido de los objetos Xlet, en el cual el componente clave del ambiente de aplicaciones de procedimiento es el motor de ejecución de contenidos de procedimientos, compuesta por la máquina virtual de Java.

3.4.2.2. Arquitectura

Ginga-J está basado en tres grupos de API's (Application Programming Interface) llamados Verde, Amarillo y Azul, a continuación se muestra la gráfica donde se observa los API's.

En la siguiente Figura 3.4, se puede observar los tres grupos de API's con el cual funciona, por ejemplo el API's Verdes es compatible con los API's GEM, en el API's Amarillo se especifican todos los requisitos específicos de Brasil y finalmente el API's Azul no es compatible con los API's GEM.

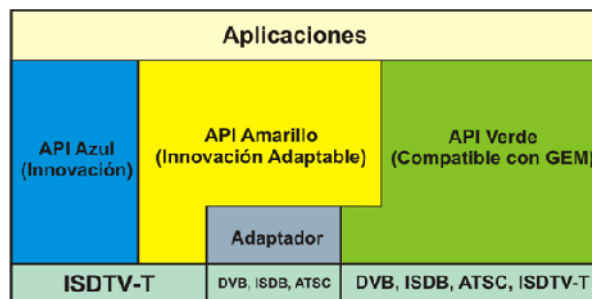


Figura 3.4: Middleware

3.4.2.3. Requerimientos

Se necesita contar un computador con las siguientes especificaciones:

- Procesador Intel o AMD a 1 Ghz
- Memoria RAM de 512 MB
- Tarjeta grafica de 512 Mb o superior
- Disco duro de 20 GB

Open Ginga: Es una implementación que permite trabajar en ambiente Ginga-J, basado en código libre que actualmente está siendo desarrollado por el laboratorio LAVID de la Universidad Federal de Parana.

Para crear las aplicaciones se las realizarán en el emulador Open Ginga, el mismo que debe tener las siguientes especificaciones:

- Linux Ubuntu 10.0.4
- Máquina Virtual Java J2SDK 1.4.0.2
- XML

3.4.3. GINGA NCL

3.4.3.1. Definición

El Ginga-NCL fue desarrollado por la Pontificia Universidad Católica de Rio de Janeiro PUC-Rio, proporciona una infraestructura de presentación para aplicaciones interactivas de tipo declarativas escritas en el lenguaje NCL (Nested Context Language), en donde NCL es una aplicación de XML (extensible Markup Language) la cual permite facilitar 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.

El NCL se basa en la estructura que separa el contenido y la estructura de un aplicativo, permitiendo definir objetos de media estructurados y relacionados tanto en tiempo y espacio [13].

3.4.3.2. Arquitectura

Para la arquitectura del Ginga-NCL está formado de un formateador, un administrador de diseño el mismo que se encarga de mantener la excelencia en las necesidades de recursos para el programa, el conversor, un programador, un administrador de bases privadas, un analizador de XML, un administrador de reproducción y uno de contextos NCL, todo lo expuesto se define en la Figura 3.5.

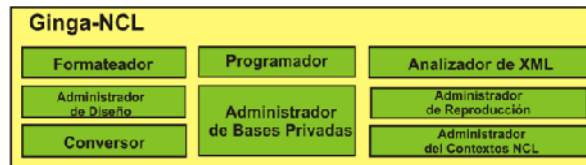


Figura 3.5: Arquitectura de GINGA-NCL

3.4.3.3. Requerimientos del Software

Los requerimientos que se necesitan para la instalación del Ginga-NCL, tenemos que contar con una computadora previamente instalado Linux, posteriormente se instala la maquina virtual. (Se podrá observar mejor en los anexos) **Requerimientos del Sistema**

- Maquina Virtual VMWare
- Memoria Ram no menor a un 1Gb
- Espacio de Disco no menor a 5Gb

Ginga-NCL STB Virtual es una aplicación virtual para los productos de VMWare. Por lo tanto, VMWare Player. La versión Ginga-NCL incluidos en Virtual STB es una implementación de referencia, con una mayor adhesión a las normas pertinentes. En una implementación de referencia, este software no debe ser tomado directamente para su uso y distribución de equipos de consumo.

3.4.3.4. Lenguaje de Programacion en GINGA NCL

NCL, es un lenguaje de aplicación XML que permite escribir presentaciones multimedia interactivas, además, es parte de las especificaciones de codificación de datos de lo Sistema Brasileño de Televisión Digital (SBTVD). El Modelo de Contextos Anidados NCM, modelo

subyacente del lenguaje NCL, los elementos multimedia pueden ser anidados, permitiendo segmentar y estructura el documento hipermedia según sea necesario.

El NCM al igual que todo lenguaje declarativo está basado en un modelo conceptual de datos, en el que se representa los conceptos estructurales de los datos así como los eventos relacionados a los mismos. Además en él, se definen las reglas estructurales y las operaciones sobre los datos de modo que se pueda manipular y actualizar las estructuras.

Los elementos que se utilizan para la programación en NCL son: **Regiones** Se define como el área de presentación de elemento multimedia, son definidas en el encabezado de programa (head), en la sección de las regiones base (regionBase).

Todo documento NCL posee por lo menos una región, que define la dimensión y las características del dispositivo donde uno o mas nodos multimedia serán presentados.

Descriptor Es el elemento encargado de definir como será presentado un nodo multimedia, asociándolo a una región, están definidos en el encabezado del programa (head), en la sección llamada base de descriptor (descriptorBase).

Contextos Los Contextos o nodos de composición son utilizados para estructurar un documento hipermedia, los mismos que pueden ser anidados, con el objetivo de reflejar la estructura del documento y ayudarle al autor a organizar de mejor manera los segmentos del programa audiovisual interactivo.

Puertas Una puerta es un punto de interfaz de un contexto, que ofrecen acceso externo al contenido de un contexto, es decir, para que un enlace apunte a un nodo interno, este contexto debe poseer una puerta que lo dirija hacia el nodo interno deseado.

Conectores El conector define los papeles que los nodos de origen y de destino ejercen en los enlaces que utiliza el conector.

Enlaces Los enlaces (links) asocian nodos a través de conectores, que definen la semántica de la asociación entre los nodos.

Anclas Las anclas son puntos de entrada para los nodos multimedia o contextos, las anclas emplean segmentos de un nodo multimedia o contexto, sea como origen o destino de los enlaces.

Switch Un switch es un contexto con nodos alternativos, es decir, dentro de los cuales apenas uno será activado. La decisión acerca de que nodo será activado es dado por reglas.

3.4.4. Cuadro comparativo entre GINGA J y GINGA NCL

Tabla 3.1: Cuadro Comparativo entre Ginga NCL vs Ginga J

Ginga NCL	Ginga J
Software libre	Necesita incluir bibliotecas extras
Menor capacidad de Procesamiento	Necesita mayor capacidad de procesamiento
Lenguaje con poca difusion	Lenguaje mas popular
La norma especifica su presencia en todo tipo de dispositivos	Se especifica en dispositivos fijos y moviles
Lenguaje maduro	Lenguaje en desarrollo

En este proyecto nos centraremos a realizar la aplicación en Ginga-NCL, debido a que cumple la condición de software totalmente libre y que actualmente es recomendación de ITU-T H.761, para IPTV y TDT.

3.4.5. LUA

Lua fue creado en 1993, es un lenguaje de programación ampliamente utilizado en diversas áreas, desde aplicaciones de escritorio, como, Photoshop de Adobe. Después de su creación, en gran parte responsable del crecimiento y la alta visibilidad de Lua fue su uso en el desarrollo de juegos, un hecho que aumentó el nivel de Lua en un buen número de programadores. Hoy en día, uno de los principales juegos que utiliza Lua como lenguaje de scripting es World of Warcraft.

Lua es un language de programación extensible diseñado para una programación procedimental general con utilidades para la descripción de datos. También ofrece un buen soporte para la programación orientada a objetos, programación funcional y programación orientada a datos. Se pretende que Lua sea usado como un lenguaje de script potente y ligero para cualquier programa que lo necesite. Lua está implementado como una biblioteca escrita en C limpio (esto es, en el subconjunto común de ANSI C y C++).

Lua es un lenguaje de programación imperativa, que es un modelo de programación que la describe en términos del estado del programa y sentencias que cambian dicho

estado. Los programadores imperativos son un conjunto de instrucciones que le indican a la computadora cómo realizar una tarea.

La programación en Lua no es muy diferente de la programación en otros lenguajes dinámicos, pero diferente. Cada idioma (lenguaje) tiene sus propias características, y un buen programador sabe cómo explotar las características particulares de cada lengua [14].

3.4.5.1. Cómo usar LUA

Los programas en Lua no son interpretados directamente, sino compilados a código bytecode, que es ejecutado en la máquina virtual de Lua. El proceso de compilación es normalmente transparente al usuario y se realiza en tiempo de ejecución, pero puede hacerse con anticipación para aumentar el rendimiento y reducir el uso de la memoria al prescindir del compilador.

Los detalles de cómo utilizar Lua dependen de la aplicación, éstos son lo que el editor debe usar, donde y como almacenar los programas, como ejecutar un programa, etc.

Siendo un lenguaje de extensión, Lua no tiene noción de programa principal (main): sólo funciona embebido en un cliente anfitrión, denominado programa contenedor o simplemente anfitrión (host). Éste puede invocar funciones para ejecutar un trozo de código Lua, puede escribir y leer variables de Lua y puede registrar funciones C para que sean llamadas por el código Lua. A través del uso de funciones C, Lua puede ser aumentado para abarcar un amplio rango de diferentes dominios, creando entonces lenguajes de programación personalizados que comparten el mismo marco sintáctico. La distribución de Lua incluye un programa anfitrión de muestra denominado lua, que usa la biblioteca de Lua para ofrecer un intérprete de Lua completo e independiente.

Para las máquinas Linux no es una receta preparada, dada la diversidad de las distribuciones de Linux. La compilación de Lua en una máquina Linux es muy sencilla y rápida. Algunas distribuciones vienen con Lua instalado por defecto, otras por ejemplo, Ubuntu y Debian, sólo se tiene que instalar el paquete lua5.1, que es el intérprete con las bibliotecas estándar. Varias bibliotecas externas también se ofrecen como paquetes adicionales. En cualquier caso, el intérprete de Lua es un programa de línea de comandos independiente que se ejecuta a través de un terminal.

Una vez instalado, es muy fácil de usar el intérprete. Lua no tiene ningún concepto de

una función main, cualquier comando que se pasa al intérprete se ejecuta inmediatamente.

3.4.5.2. Elementos de LUA

Esta sección describe el léxico, la sintaxis y la semántica de Lua. En otras palabras, esta sección describe qué elementos (tokens) son válidos, cómo deben combinarse y qué significa su combinación.

Las construcciones del lenguaje se explicarán usando la notación BNF extendida usual, en la que ϵ significa 0 o más veces, y $[a]$ significa una a opcional. Los símbolos no terminales se muestran en *italica*, las palabras clave (keywords) se muestran en **negrita**, y los otros símbolos terminales se muestran en un tipo de letra de paso fijo (typewriter), encerrada entre comillas simples. La sintaxis completa de Lua se encuentra al final de este manual.

Convecciones Léxicas Los nombres (también llamados identificadores) en Lua pueden ser cualquier cadena de caracteres sólo con letras, dígitos y caracteres de subrayado, no comenzando por un dígito. Esto coincide con la definición de los nombres en la mayoría de los lenguajes. (La definición de letra depende de la implementación local actual a través del sistema local: cualquier carácter considerado alfabético en el sistema local puede ser usado en un identificador.) Los identificadores se usan para nombrar variables y campos de tablas.

Las siguientes palabras clave (keywords) están reservadas y no pueden usarse como nombres, como se observa en la Figura 3.6:

and	break	Do	Else	elseif
end	false	For	Function	if
in	local	Nil	Not	or
repeat	return	Then	True	until
while				

Figura 3.6: Palabras reservadas en LUA

En Lua las letras mayúsculas y las minúsculas se consideran diferentes: **and** es una palabra reservada, pero **And** y **AND** son dos nombres diferentes válidos. Como convención, los nombres que comienzan por un subrayado seguido por letras en mayúsculas están reservados para uso como variables globales internas de Lua. Los siguientes strings denotan otros elementos, como se observa en la Figura 3.7:

```

+   -   *   /   %   ^   #
==  ~=  <=  >=  <   >   =
(   )   {   }   [   ]
;   :   ,   - -  - - |

```

Figura 3.7: Listado de strings

Los strings literales pueden ser delimitados por comillas simples (apóstrofes) o dobles, y pueden contener las siguientes secuencias de escape de C: `á`'(pitido, bell) `'b`'(retroceso, `backspace`), `'f`'(salto de página, `form feed`), `ñ`'(nueva línea, `newline`), `'r`'(retorno de carro, `carriage return`), `'t`'(tabulador horizontal, `horizontal tab`), `'v`'(tabulador vertical, `vertical tab`), `'` (barra inversa, `backslash`), `""`(comilla doble, `double quote`) y `'` (apóstrofe, `apostrophe`). Además, una `ñewline`'(esto es, una barra inversa seguida por un salto de línea real) produce un salto de línea en el string. Un carácter en un string puede también especificarse por su valor numérico usando la secuencia de escape `'ddd'`, donde `ddd` es una secuencia de tres dígitos decimales. (Tenga presente que si la secuencia numérica de escape está seguida de un dígito debe ser expresada usando exactamente tres dígitos.) Los strings en Lua pueden contener cualquier valor de 8 bits, incluyendo el carácter cero, el cual puede ser especificado mediante `'0'`. Para poner una comilla (simple) doble, una barra inversa, un retorno de carro o un carácter cero dentro de un string literal encerrado por comillas (simples) dobles se debe usar una secuencia de escape. Cualquier otro carácter puede ser incluido en el literal. (Algunos caracteres de control pueden causar problemas con el sistema de ficheros, pero Lua no tiene problemas con ellos.) Las constantes numéricas pueden contener una parte decimal opcional y también un exponente opcional. Lua también acepta constantes enteras hexadecimales, escritas anteponiendo el prefijo `0x`. Los comentarios comienzan con un doble guión (`-`) en cualquier lugar fuera de un string. Si el texto que sigue inmediatamente después de `-` no es un corchete largo de abrir el comentario es corto y llega hasta el final de línea. En otro caso tenemos un comentario largo, que alcanza hasta el correspondiente corchete largo de cerrar. Los comentarios largos se usan frecuentemente para deshabilitar temporalmente trozos de código.

Valores y Tipos Lua es un lenguaje dinámicamente tipado. Esto significa que las variables no tienen tipos; sólo tienen tipo los valores. No existen definiciones de tipo en el lenguaje. Todos los valores almacenan su propio tipo. Todos los valores en Lua son valores de primera clase. Esto significa que todos ellos pueden ser almacenados en variables, pueden ser pasados como argumentos de funciones, y también ser devueltos

como resultados. Existen ocho tipos básicos en Lua: nil, boolean, number, string, function, userdata, thread y table. Nil es el tipo del valor nil, cuya principal propiedad es ser diferente de cualquier otro valor; normalmente representa la ausencia de un valor útil. Boolean es el tipo de los valores false (falso) y true (verdadero). Tanto nil como false hacen una condición falsa; cualquier otro valor la hace verdadera. Number representa números reales (en coma flotante y doble precisión). Lua puede llamar (y manejar) funciones escritas en Lua y funciones escritas en C. El tipo userdata se incluye para permitir guardar en variables de Lua datos arbitrarios en C. Este tipo corresponde a bloques de memoria y no tienen asociadas operaciones predefinidas en Lua, excepto la asignación y el test de identidad. Sin embargo, usando metatablas, el programador puede definir operaciones asociadas a valores de tipo userdata. Los valores de este tipo no pueden ser creados o modificados en Lua, sino sólo a través de la API de C. Esto garantiza la integridad de los datos propiedad del programa anfitrión. El tipo thread representa procesos de ejecución y es usado para implementar co-rutinas. No deben confundirse los procesos de Lua con los del sistema operativo. Lua soporta co-rutinas en todos los sistemas, incluso en aquéllos que no soporten procesos. El tipo table (tabla) implementa arrays asociativos, esto es, arrays que pueden ser indexados no sólo con números, sino también con cualquier valor (excepto nil). Las tablas pueden ser heterogéneas, ya que pueden contener valores de todos los tipos (excepto nil). Las tablas son el único mecanismo de estructuración de datos en Lua; pueden ser usadas para representar arrays ordinarios, tablas de símbolos, conjuntos, registros, grafos, árboles, etc. Para representar registros Lua usa el nombre del campo como índice. El lenguaje soporta esta representación haciendo la notación b.nombre equivalente a b[“nombre”]. Como índices, también los valores de los campos de una tabla pueden ser de cualquier tipo (excepto nil). En particular, debido a que las funciones son valores de primera clase, los campos de las tablas pueden contener funciones. Entonces las tablas pueden contener también métodos. Los valores de las tablas, las funciones, los procesos y los userdata (completos) son objetos: las variables no contienen realmente esos valores, sino que sólo los referencian. La asignación, el paso de argumentos y el retorno de las funciones siempre manejan referencias a esos valores; esas operaciones no implican ningún tipo de copia. La función de biblioteca type retorna un string que describe el tipo de un valor dado.

Conversion Lua puede convertir automáticamente entre valores string y valores numéricos en tiempo de ejecución. Cualquier operación aritmética aplicada a un string intenta convertir el mismo en un número, siguiendo las reglas normales de conversión. Y viceversa, cuando un número se usa donde se espera un string el número se convierte a

string, con un formato razonable. Para un control completo en la conversión de números en strings debe usarse la función `format` de la biblioteca de manejo de strings.

Variabes Las variables son lugares donde se almacenan valores. Existen tres tipos de variables en Lua: globales, locales y campos de tabla. Un único nombre puede denotar una variable local o una global (o un argumento formal de una función, el cual es una forma particular de una variable local):

`var ::= nombre`

Lua asume que las variables son globales, a no ser que sean declaradas explícitamente como locales. Las variables locales tienen un ámbito (scope) definido léxicamente: pueden ser accedidas libremente desde dentro de las funciones definidas en su mismo ámbito. Antes de la primera asignación el valor de una variable es nil. Los corchetes se usan para indexar una tabla:

`var ::= prefixexp '[exp]'`

La primera expresión (prefixexp) debe dar como resultado un valor tabla; la segunda expresión (exp) identifica una entrada específica en esta tabla. La expresión que denota la tabla que es indexada tienen una sintaxis restringida [15].

Capítulo 4

CANAL DE RETORNO EN TV DIGITAL

4.1. ¿Qué es el Canal de Retorno?

El canal de retorno es un medio de transmisión que permite la comunicación entre el receptor de TV con el operador de servicio interactivo, y permite el envío y recepción de datos que el usuario intercambia con el operador de televisión, para poder hacer uso de ciertos servicios interactivos. Este canal de retorno puede ser implementado mediante varias tecnologías existentes en el mercado, cada una de las cuales presentan características que pueden ser mejores para determinadas zonas, regiones o países. Como se ha mencionado anteriormente, una de las ventajas más importantes en la televisión digital, es la posibilidad de que el usuario pueda interactuar con la televisión a través del control remoto, permitiéndole ver información adicional, manipular la programación de los canales de televisión, comprar servicios o productos, participar en concursos, votaciones, etc. Sin embargo este tipo de aplicaciones además de necesitar de una plataforma para aplicaciones interactivas, requieren de una comunicación bidireccional, es decir necesitan añadir un canal de retorno para que contemple al canal de bajada existente, para que así el usuario esté en la capacidad de interactuar directamente con el servidor exterior (Interactividad Remota) o en la que tenga acceso a internet (Interactividad Plena).

La interactividad con Canal de Retorno, son aplicaciones para el envío de datos por parte del telespectador a un proveedor de servicio. Este tipo de aplicaciones necesitan de una conexión a Internet por parte del decodificador para poder enviar los datos. El canal de retorno, que puede ser la línea telefónica, ADSL o el Cable Modem de los operadores de cable siempre que se tenga acceso a estos servicios.

Algunas de las aplicaciones interactivas con canal de retorno en la TV Digital podrán

ser usadas para diversos fines, siempre con el objetivo de brindar un beneficio a la sociedad y la inclusión digital, éste trae consigo diferentes topologías de comunicación y modelos de negocio, tales como; T-Voting o encuesta por televisión; donde los datos votados son almacenados en un servidor de base de datos, T-Learning ; aprendizaje interactivo a través de un televisor, personalización de contenidos educativos y recreativos para la consecución de cursos atractivos y efectivos para cada usuario, T-Commerce ; comprar a través de las pantallas del televisor, incluyendo anuncios y demás servicios de comercio, Telebanca; con acceso a bases de datos (mercados financieros, bolsa, etc), T-Government.

4.2. Tipos de Canales de Retorno

Existen ciertas aplicaciones que necesitan incrementar el grado de interactividad, para esto se necesita implementar un canal de retorno, a continuación se realizará un análisis de las tecnologías existentes para la implementación del canal del retorno.

4.2.1. Satélite

La Televisión Digital vía Satélite es el resultado de la aplicación de la tecnología digital a la señal de televisión, para luego transmitirla a una amplia zona geográfica por medio de satélites de comunicaciones, en contraste con la televisión terrestre, en el cual las ondas no salen de la atmósfera, siendo el principal elemento los satélites, los cuales ayudan al desarrollo de la difusión de la TV.

La televisión satelital ofrece una rápida instalación de los equipos, brinda una gran cobertura la cual no depende de cables ni distancias, además, el ancho de banda que tiene es muy alto llegando al orden de los Mbps.

Los servicios satelitales son muy útiles tanto para dispositivos móviles como para fijos, lo cual permite una comunicación más eficiente y de mejor capacidad ya que no dependen de ninguna otra red para el canal de retorno.

Una de las principales ventajas de utilizar los servicios satelitales es el de ofrecer una gran cobertura, pero a pesar de las ventajas mencionadas el utilizar estas redes solamente como un canal de retorno sería muy costoso para el operador, ya que el tiempo que demora en llegar la información transmitida desde el usuario al operador es mayor con lo cual puede provocar una reducción en la calidad de los servicios y problemas en las aplicaciones de tiempo real, por lo que el operador podría optar por ofrecer otro tipo de servicios como

el internet.

4.2.2. Asymmetric Digital Subscriber Line ADSL

Es una técnica de modulación de datos a altas velocidades sobre las existentes líneas telefónicas de par trenzado de cobre, la misma puede llegar a conseguir tradicionalmente velocidades de transmisión de datos descendentes (de la central hasta el usuario) de hasta 1,5 Mbps sobre distancias de 6 Km, y de hasta 8 Mbps para distancias de 3 km. Las velocidades máximas descendentes (desde el usuario a la central), iban de 16-640 Kbps, sobre los mismos tramos.

La velocidad real depende de las limitaciones puestas por la operadora según la tarifa contratada por el abonado. Las distancias alcanzadas resultan adecuadas para cubrir el 90 % de los abonados de la RTB.

El sistema de acceso ADSL se compone de dos módems en cada extremo de la línea telefónica, creándose tres canales de información: uno descendente a alta velocidad, otro ascendente dúplex a velocidad media, y el del servicio telefónico básico, el cual es separado del módem digital mediante filtros, garantizando así la continuidad del servicio telefónico ante una caída o fallo de dicho módem, lo cual es una ventaja respecto a RDSI. Al igual que en la RDSI y que el resto de tecnologías de acceso a Internet de banda ancha, los usuarios de ADSL pueden utilizar simultáneamente el teléfono e Internet. En la central del operador los módems suelen estar dispuestos en bastidores, y se conectan a la red Internet o a otras redes de datos mediante un enlace Ethernet, un gigarouter IP o un switch ATM. En la Figura 4.1, se mostrará cómo está conformada la arquitectura de un sistema xDSL, en el cual, desde la central de abonados o línea telefónica se le da varios usos como de telefonía, internet, televisión para pasar a la central telefónica la misma que conmuta nuestras señales para ir a la red troncal es decir al origen de las señales deseadas.

ADSL emplea la modulación DMT (Discrete Multi Tone), el cual consiste en la utilización de varias portadoras simultáneas para la transmisión de la señal de datos. Cada una de las portadoras que usa el DMT se denominan subportadoras y una vez moduladas ocupan un ancho de banda de 4KHz, donde la cantidad de información que va a transportar cada una de las subportadoras, depende de la proporción señal ruido que se estima al principio de la comunicación entre el modem ADSL del abonado y el de la compañía.

Una de las características más notables del ADSL es que nos permite compartir servicios

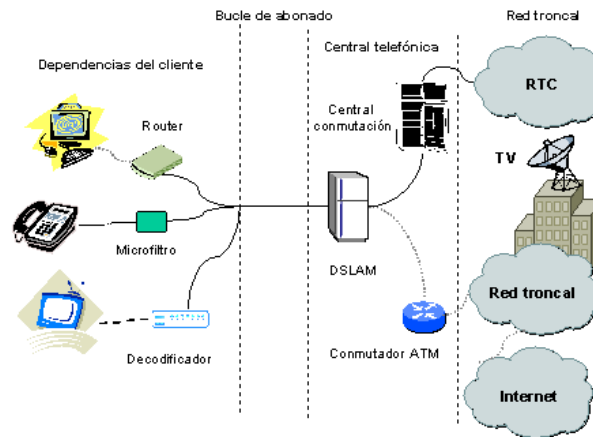


Figura 4.1: Arquitectura de la Red de Acceso xDSL

de telefonía como otros servicios que ofrezca el ADSL como IPTV o triple play, también tenemos que la ventaja del uso de esta tecnología es un medio flexible y relativamente barato para ser utilizado como canal de retorno para la televisión digital. Además, las aplicaciones multimedia y la necesidad de una mayor velocidad, hicieron que ADSL sea el sistema más desplegado por todo el mundo.

Actualmente, en diversos países donde las empresas de telefonía están implantando versiones mejoradas de esta tecnología como ADSL2 y ADSL2+ con capacidad de suministro de televisión y video de alta calidad por el par telefónico, lo cual supone una dura competencia entre los operadores telefónicos y los de cable, y la aparición de ofertas integradas de voz, datos y televisión, a partir de una misma línea y dentro de una sola empresa, que ofrezca estos tres servicios de comunicación. El uso de un mayor ancho de banda para estos servicios limita aún más la distancia a la que pueden funcionar, por el par de hilos. ADSL2 y ADSL2+ incorporan mecanismos de modulación y gestión de los recursos físicos avanzados, de modo que no sólo aumentan la capacidad del ADSL convencional de 8 mbps a 12 y 24 Mbps respectivamente, sino que introducen mejoras para evitar las interferencias o ruido y disminuir los efectos de la atenuación de ahí que se alcancen distancias de hasta 9 Km.

4.2.3. Cable

El cable coaxial ha sido muy utilizado para transportar datos y señales de televisión, sin embargo las características de sus componentes dan origen a pérdidas o perturbaciones como el ruido térmico, conectores, blindaje, etc., reflexiones de línea por desadaptaciones

de impedancia.

La TV por Cable es el resultado de la aplicación de la tecnología digital a la señal de televisión, para luego distribuirla por medio de redes híbridas (fibra óptica y cable coaxial) las cuales son introducidas a la red de acceso, para la transmisión por fibra óptica. Esto permite reducir las perturbaciones del sistema por cable coaxial y cubrir mayores distancias, hasta 100Km sin amplificadores. Junto con la señal de Televisión Digital, a través de estas redes se proporcionan otros servicios como radio, telefonía fija y acceso a Internet.

En la Figura 4.2 se muestra la distribución de este tipo de servicios se dividen en cuatro secciones:

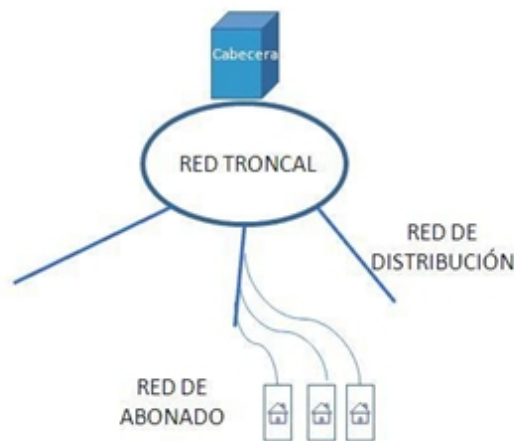


Figura 4.2: Distribucion de Servicios

La Cabecera es el órgano central, en el cual se combina las distintas fuentes de programación como canales de televisión, telefonía, datos, etc., en ésta cubre la demanda de los servicios de datos y de interactividad.

La Red Troncal, es la encargada de repartir la señal generada en la cabecera a todas las zonas de distribución donde lleve cable, generalmente la red troncal son redes ópticas.

La Red de Distribución, es la encargada de suministrar la señal hasta que llegue a los usuarios, su topología es de tipo bus el mismo que lleva la señal en forma descendente hasta el cliente.

A continuación, en la Figura 4.3, se mostrará el funcionamiento de una red híbrida,

la misma que está compuesta de cable coaxial y fibra óptica, en el cual se observa una transmisión de datos por lo cual utiliza la fibra óptica pero para sus terminales se utiliza el cable coaxial.

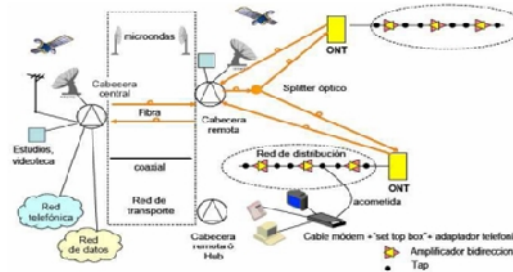


Figura 4.3: Esquema de una Red con Cable Bidireccional

Según la figura anterior, se puede concluir que la red HFC es un medio óptimo de transmisión el cual puede proveer de varios servicios como son telefonía, datos, televisión e inclusive puede soportar la interactividad en la TDT, gracias a su ancho de banda que ofrece tanto en sentido ascendente como descendente.

4.2.4. Universal Mobile Telecommunications System(UMTS)

Es una de las tecnologías de tercera generación sucesora del GSM; aunque inicialmente su uso estaba destinado para teléfonos móviles la red UMTS no está limitada a estos dispositivos, pudiendo ocupar otros.

Entre las características principales del UMTS es la capacidad multimedia, la velocidad de acceso a Internet, permite la transmisión de audio y video en tiempo real, así como también, la transmisión de voz en alta calidad.

Presenta una arquitectura en la cual se describen tres elementos como son el equipo de usuario, el mismo que es el equipo del usuario trae consigo para lograr la comunicación con la estación base; UTRAN (Terrestrial Radio Access Network), es la encargada de conectar con la red central y con el equipo de usuario a través de dos interfaces como son la Iu y Uu, la primera conecta a la red central con la red de acceso de radio mientras que la Uu se encuentra entre el equipo del usuario y la red; y la red central está formada por recursos de conmutación y transmisión necesarios para completar el trayecto de la comunicación hacia el abonado remoto, a continuación en la Figura ??, se identificará la arquitectura del UMTS.

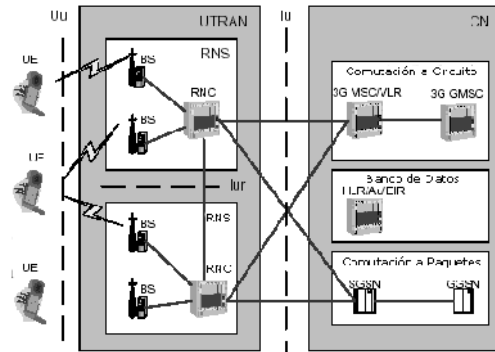


Figura 4.4: Arquitectura del UMTS

En la figura anterior se puede observar la arquitectura del UMTS, identificando sus elementos como el equipo de usuario, la UTRAN con sus estaciones bases, subsistemas de radio y controladores de radio y finalmente la red central.

4.2.5. Local Multipoint Distribution Service(LMDS)

Es una tecnología inalámbrica de conexión vía radio, ofreciendo servicios multimedia gracias a su ancho de banda. Los sistemas de LDMS brindan la cobertura del servicio por zonas, las cuales son distribuidas por estaciones base para cubrir con el número de usuarios, el abonado al sistema recibe la señal mediante una de tres vías: desde el emisor principal de la célula, si existe visibilidad directa entre éste y el receptor; desde un repetidor, en zonas de sombra; mediante un rayo reflejado en alguna superficie plana (paredes de edificios, reflectores / repetidores pasivos, etc.).

En la Figura 4.5 se puede observar que la torre de la estación base LMDS cubre a todos los abonados, brindando un servicio óptimo a sus usuarios.

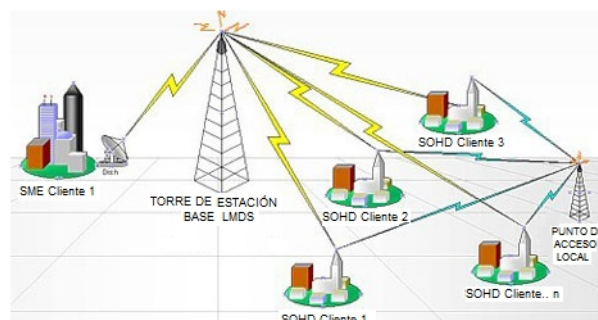


Figura 4.5: Arquitectura del LMDS

Una de las ventajas de utilizar LMDS es la alta velocidad de acceso a Internet, el costo de instalación es bajo, se posibilita la integración de servicios sobre el mismo medio, un mismo usuario puede recibir tales como acceso a Internet, telefonía, información multimedia bajo demanda, datos, etc., permitir la bidireccionalidad, se pueden ofrecer servicios como la TV multicanal, la telefonía ó el acceso a Internet conjuntamente mediante una plataforma única, además con el LMDS la calidad de la señal no se ve afectada por las defectuosas redes de comunicaciones locales existentes en muchos lugares, ya que todo el bucle local se realiza independiente de las mismas, vía radio.

4.3. Funcionamiento del Canal de Retorno

4.3.1. Simulación en Virtual Set Top Box

Las pruebas se realizan en el virtual Set Top Box Ginga NCL instalado en una Laptop y con sistema operativo Linux Ubuntu, el cual estará conectado a una red LAN Ethernet de 10/100 Mbps. En la Figura 4.6, se ilustra el escenario de pruebas para la interactividad con canal de retorno a través de internet.



Figura 4.6: Escenario de prueba (simulación) para la interactividad con canal de retorno

4.3.2. Implementación en el Set Top Box para TV Digital

También se puede realizar pruebas reales en un escenario implementado para la interactividad bidireccional. Para la implementación se utiliza un Set Top Box con soporte middleware Ginga NCL, MPEG-2, H.264 HD/SD, de 64 MB Flash Memory y 256 MB SDRAM, un Televisor Digital y un dispositivo de memoria externa (USB). El escenario de pruebas se muestra en la figura 4.7. A través de la interface de red (LAN Ethernet de 10/100 Mbps) del STB se conecta al internet, que se utiliza como canal de retorno para la

aplicacion interactiva.

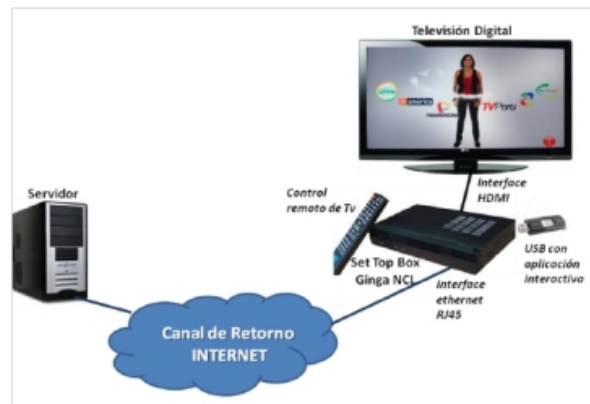


Figura 4.7: Escenario de prueba real para la interactividad con canal de retorno

La carga de la aplicación interactiva al Set Top Box se realizó mediante el puerto USB, a través de una memoria externa USB, al servidor. Como se puede observar en la Figura 4.8 la transmisión del programa de compras por TV.



Figura 4.8: Emisión del programa de compras para TDT

En la TDT se puede realizar programas de interactividad con canal de retorno, para lo cual es necesario que el usuario tenga el Set Top Box, acceso a Internet ya que aquí en una de las formas que se pueden hacer para cargar y bajar programas de la elección del telespectador. El Set Top Box va enlazado con el punto de conexión a la red, con lo cual por medio de un broadcast se llegarían a cargar aplicaciones desde el punto de nuestra central, lo que significaría que el usuario puede navegar desde su televisor y formar parte de aplicaciones en tiempo real. Las aplicaciones pueden ser: educativas, lo que significaría que se podrían descargar información de un buscador o dar pruebas en línea; compras, ya que se puede tener una tienda virtual, restaurantes, joyerías, y gracias al acceso del

internet se puede realizar transferencias del banco de preferencia, así también como revisar el estado de cuenta con el que contamos; entretenimiento, con lo que se puede realizar desarrollos de preguntas en línea, juegos, etc.

Capítulo 5

BASE DE DATOS PARA APLICACIONES INTERACTIVAS

5.1. Bases de Datos

Una base de datos es un conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su posterior uso. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

Cuando una base de datos es creada, ésta es asignada a un administrador, quien ejecutó la sentencia de creación. Usualmente, únicamente el administrador (o un superusuario) puede hacer cualquier cosa con los objetos de esa base de datos, y para que otros usuarios puedan usarla, deben otorgarse privilegios.

Se puede crear diferentes usuarios de la base de datos para cada aspecto de su aplicación, con derechos muy limitados sobre los objetos de la base de datos. Tan solo deben otorgarse los privilegios estrictamente necesarios y evitar que el mismo usuario pueda interactuar con la base de datos en diferentes casos de uso. Esto quiere decir, que si un intruso gana acceso a su base de datos usando las credenciales de sus aplicaciones, él solo puede efectuar tantos cambios como su aplicación se lo permita.

Una buena opción es no implementar toda la lógica del asunto en la aplicación web, (es decir, en su script). En su lugar, se debe hacerlo en el esquema de la base de datos usando vistas, disparadores o reglas[16].

5.2. Bases de Datos

Las bases de datos pueden ser clasificadas de acuerdo a la variabilidad de los datos almacenados, de esta manera encontramos:

5.2.1. Bases de Datos Estáticas

Son aquellas bases de datos de sólo lectura. Su propósito esencial consiste en el almacenamiento de información de carácter histórico, con el fin de que luego sea usada para analizar la conducta de una serie de datos a lo largo del tiempo. De esta manera, es posible efectuar proyecciones y actuar en consecuencia.

5.2.2. Bases de Datos Dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el correr del tiempo. De esta manera se realizan acciones tales como actualizaciones, eliminaciones o agregado de datos, más allá de las acciones básicas de consulta [17].

5.3. Funcionamiento de la Bases de Datos y la Interactividad

Para que funcione la interactividad con la televisión es necesario la existencia de una Plataforma de Televisión Digital. Dentro de esa Plataforma existirán varios componentes como:

- Un Sistema Operativo de gestión de Servicios Interactivos

El Sistema operativo de Gestión, es lo que hace posible la definición y realización de servicios interactivos.

- Contenidos Interactivos

De nada sirve toda la infraestructura tecnológica si no se apoya en un buen contenido capaz de atraer y aportar un cierto valor añadido al espectador.

- Un servidor de Comunicaciones

El Servidor de Comunicaciones controla el acceso de las llamadas y el flujo de datos bidireccional que ha de establecerse entre usuario y acceso de las llamadas, así como y el flujo de datos bidireccional que ha de establecerse entre usuario y plataforma para completar un determinado servicio. En definitiva, es quien gestiona el canal de retorno de información que llega desde los usuarios.

- Un sistema de Gestión de Usuarios

Una vez establecemos contacto con el usuario, tenemos que gestionar la información que éste proporciona. En definitiva se necesita un sistema de gestión de bases de datos que permita atender a los usuarios que ya no son simples espectadores, sino clientes de una plataforma digital.

- Un sistema de Gestión de Servicios

Es un sistema que garantiza la comunicación entre los elementos de la plataforma y las Bases de datos y sistemas del proveedor externo.

5.3.1. Creación de la Base de Datos

Como anteriormente se expuso, una base de datos bien estructurada permite manejar mayor información de forma organizada, y acceder a los datos adecuadamente para que puedan ser utilizados según la necesidad de la aplicación. Nuestra base de datos fue creada en MySQL, el mismo que es software libre que permite gestionar una o más bases de datos de forma relacional, multihilo y multiusuario [18], usando como administrador de la misma a phpMyAdmin.

phpMyAdmin, es una herramienta de software libre escrito en PHP que es un lenguaje diseñado para crear páginas web dinámicas, con lo que permite interactuar con una base de datos en éste caso MySQL de forma muy sencilla y desde una interfaz web [19], las mismas que presentan la siguientes características:

- Presenta una interfaz web intuitiva, es decir fácil de usar para los usuarios, independientemente de la capacidad técnica o experiencia previa.

- Permite buscar, crear, copiar, borrar, renombrar y modificar bases de datos, tablas, campos e índices.
- Permite ejecutar, editar y marcar cualquier SQL, incluso por lotes de consultas.
- Permite administrar usuarios y privilegios de MySQL
- Permite exportar datos de varios formatos: CSV, SQL, XML, PDF, etc

Para acceder a phpMyAdmin se necesita instalar previamente MySQL, Aapache2 y Php5, los cuales serán detallados en el diseño de base de datos. Apache es un servidor de http de código abierto, cuyo objetivo es de proporcionar un servicio seguro, eficiente y extensible que proporcione sincronía con los estándares http actuales y el servicio que ofrece

En la Figura5.1 se muestra el proceso jerárquico en la instalación de las diferentes herramientas informaticas que se utilizan para la utilizacion del lenguaje phpMyAdmin.

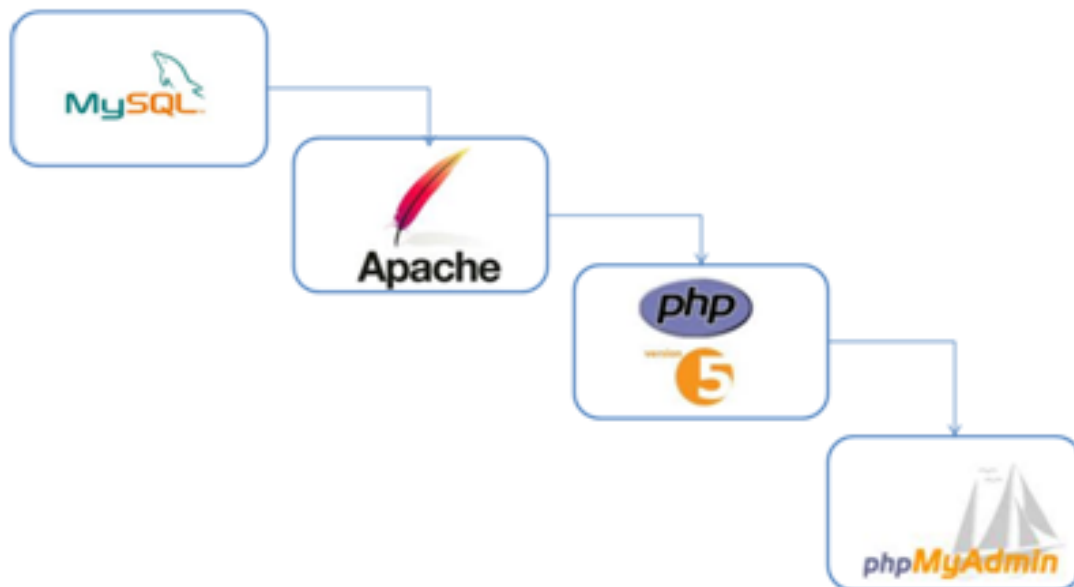


Figura 5.1: Proceso jerárquico para instalar phpMyAdmin

5.3.2. Diseño de la Base de Datos

Para el presente proyecto de tesis se ha diseñado la base de datos llamada ShoppingCar, la misma que tiene como objetivo realizar la compra de tres productos utilizando la TDTi.

Para el diseño de ShoppingCar se utiliza 6 tablas, para la base de datos, las mismas que se detalla el contenido en la Tabla 5.1.

Tabla 5.1: Diseño de la Base de Datos para ShoppingCar

Nombre de la Tabla	Descripción	Parametros de la Tabla
Cliente	Establece los datos del cliente	Identificacion del Cliente Nombre Direccion Teléfono Email Twitter
Cabecera de Pedido	Establece una relación con otras Tablas	Codigo de Pedido Identificacion del Cliente Fecha Total
Detalle del pedido	Ésta es una tabla muy importante ya que es donde posteriormente se puede verificar si el registro o almacenamiento de datos se hizo de forma correcta	Codigo de Pedido Codigo de Producto Cantidad Precio Total
Categoría	Establece un listado de identificacion para llevar un registro de los productos en stock	Codigo de categoría Descripcion de Categoría
Producto	En esta tabla se encuentra la descripción del Producto	Codigo de Producto Descripcion del Producto Codigo de Categoría Stock
Usuario	Ésta tabla fue creada para registrar a los nuevos usuarios y en aplicaciones futuras se puedan crear un login para llevar una base de datos que administre tambien la informacion de los nuevos clientes	Nick Contraseña Identificacion del Cliente Fecha Estado

Vale recalcar que esta tabla Usuario no se la utiliza porque no se efectúa un login en la aplicacion.

Con las tablas realizadas para la base de datos, se procedió a esquematizar en el phpMyAdmin, es decir, se identifican el tipo de variable, la relación entre tablas con primary key, el mismo que es un identificador que permite la asociación de los parámetros entre tablas. A continuación en la Figura 5.2, se observa la relación entre las diferentes tablas creadas en phpMyAdmin, además se observa los tipos de variables que se utilizó para crear la base de datos.

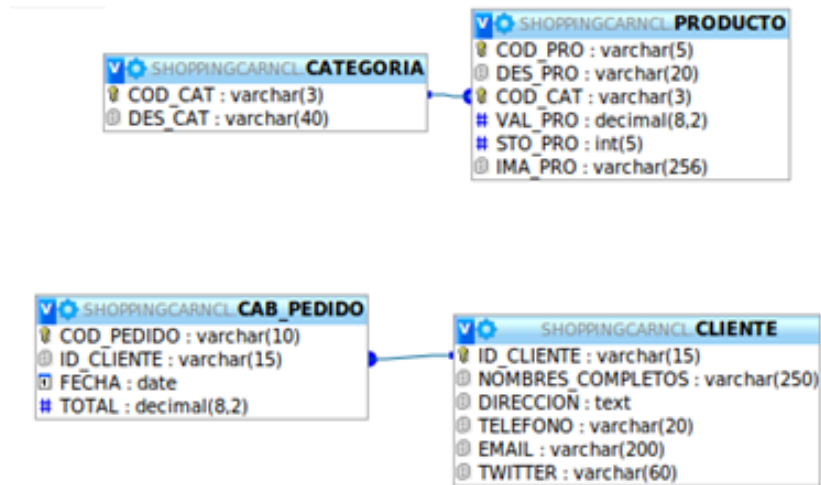


Figura 5.2: Tablas creadas para ShoppingCar

5.4. Conexión a la Base de Datos a través de un Web Service

Para establecer la conexión entre la aplicación generada en NCL-LUA y la base de datos en MySQL, se utilizaron protocolos que están enfocados en la conexión a internet, como el Simple Object Access Protocol (SOAP), que es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML, el mismo que es un lenguaje de marcado extensible, permitiendo el intercambio de información de las aplicaciones a través de HTTP.

SOAP, no se encuentra asociado a ningún protocolo de transporte; pero permite que, un mensaje de SOAP no es más que un documento XML, lo cual lo aprovecha para la codificación de los mensajes, en lugar de utilizar su propio sistema, los mismos que se encuentran especificados en el esquema de XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.

En la siguiente Figura 5.3 se observa el funcionamiento del protocolo SOAP, cuando el

Web Service es una aplicación modular accesible a través de Internet, su objetivo es la interoperabilidad entre aplicaciones, con un mínimo acoplamiento e independientemente de la tecnología con la que hayan sido desarrolladas, el Web Service se comunica mediante mensajes XML utilizando el protocolo SOAP.

La interfaz de un Web Service se describe en un documento XML utilizando el estándar Web Services Description Language (WSDL), en este fichero, que recibe el mismo nombre WSDL, se definen los detalles de cómo usar un Web Service, como las operaciones, el formato de los mensajes, el tipo y el protocolo de transporte del Web Service. El WSDL es, por tanto, básico a la hora de crear un cliente para un Web Service [20].



Figura 5.3: Funcionamiento del protocolo SOAP

El protocolo Hyper Text Transfer Protocol Secure (HTTP), permite la transferencia de archivos principalmente, en formato HTML, entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. En la Figura 5.4 se observa la comunicación entre el navegador y el servidor se lleva a cabo en dos etapas, las mismas que comprenden la parte de solicitud de HTTP y la segunda parte el servidor procesa la solicitud y después envía una respuesta HTTP, todo el proceso de comunicación se complementa con el llamado al servidor [21].

Y finalmente el protocolo Simple Mail Transfer Protocol (SMTP), que es un protocolo de la capa de aplicación, el mismo que es utilizado para el intercambio de mensajes. El protocolo SMTP funciona con comandos de textos enviados al servidor SMTP. A cada comando enviado por el cliente le sigue una respuesta del servidor SMTP compuesta por un número y un mensaje descriptivo [22].

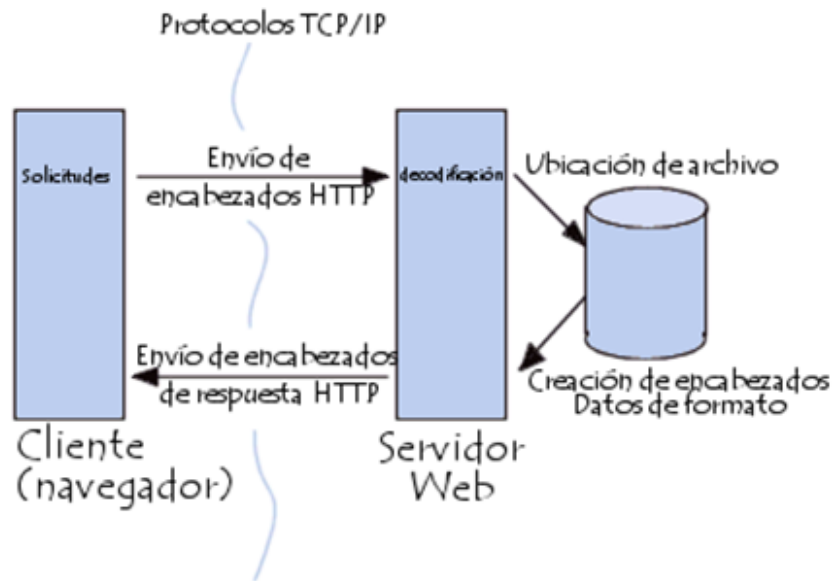


Figura 5.4: Funcionamiento del protocolo HTTP

Para crear la comunicación desde NCL-LUA con la base de datos MySQL se extrae la información del Web Service, comunicándose por la capa de aplicación a través de http y una vez establecida la conexión entre los dos se pudo interactuar de forma exitosa con la base de datos. A continuación se presentará un esquema de conexión en la Figura 5.5, que básicamente es de tipo cliente-servidor.

Como se puede observar en la Figura [?], el [Web Service??] tiene conexión con la base de datos por medio de PHP, el mismo que es un lenguaje de código abierto utilizado por los desarrolladores de aplicaciones web dinámicas y que puede ser incrustado en HTML. Vale mencionar que éste forma parte de appEjemplo.php, conteniendo por default la librería nusoap, que permite desarrollar Web Services bajo el lenguaje PHP, se empleó ésta librería ya que está en una fase de desarrollo mucho más avanzada. Además, PHP a partir de su versión 5 comienza a dar soporte para SOAP. En los siguientes ítems se describirá el proceso para lograr la comunicación entre NCL-LUA y MySQL.

5.5. Conexión del Servidor PHP con la Base de Datos

Como se describió anteriormente en PHP se desarrollan aplicaciones web dinámicas, además utiliza en los principales sistemas operativos, incluyendo Linux, además soporta la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, etc., lo cual le permite elegir con libertad el sistema operativo y el servidor web.

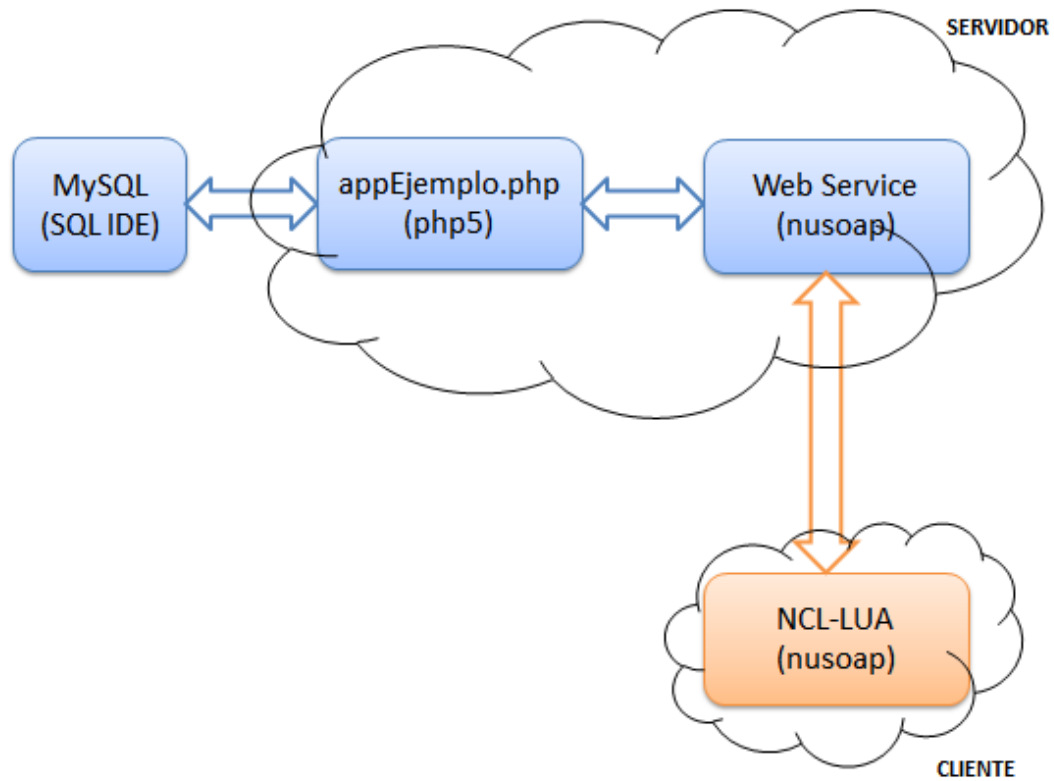


Figura 5.5: Esquema para establecer la conexión entre NCL-LUA y la base de datos MySQL a través del consumo un Web Service

En cuanto a la programación, ésta está orientada a objetos; incluye también, creación de imágenes, archivos PDF e incluso películas Flash, los resultados son presentados como XHTML y cualquier otro tipo de ficheros XML. Una de las características más potentes de PHP es su soporte a la gran cantidad de base de datos [23].

5.5.1. Estructura de Programacion

La programación es muy similar a la de HTML, en donde se tiene la cabecera y el cuerpo, a continuación se dará una declaración de la estructura de programación:

- **Variables:** Es un contenedor de información, en el cual contiene números enteros, números decimales, caracteres, etc, todas las variables comienzan con el símbolo del dólar (\$) y no es necesario definir una variable antes de usarla. Tampoco tienen tipos, es decir que una misma variable puede contener un número y luego puede contener caracteres.

- **Operadores:** Es algo que toma uno más valores o expresiones y produce otro valor de modo que la construcción en si misma se convierte en una expresión. Los operadores se dividen en aritméticos, comparación y lógicos, de donde los aritméticos se pueden aplicar a las variables y constantes numéricas, mientras que los de comparación son usados para comparar valores y así poder tomar decisiones, y por último los lógicos son usados para evaluar varias comparaciones, combinando los posibles valores de estas.
- **Instrucciones de Control:** Es algo que toma uno más valores o expresiones y produce otro valor de modo que la construcción en si misma se convierte en una expresión. Los operadores se dividen en aritméticos, comparación y lógicos, de donde los aritméticos se pueden aplicar a las variables y constantes numéricas, mientras que los de comparación son usados para comparar valores y así poder tomar decisiones, y por último los lógicos son usados para evaluar varias comparaciones, combinando los posibles valores de estas.
- **Clases:** Sirven para que el código sea más legible, al escribir una clase es más largo que escribir el código directamente, pero es más rentable por su portabilidad a otras aplicaciones y su mantenimiento. Las clases son una serie de variables y funciones que describen y actúan sobre algo, por ejemplo la clase Detalle Pedido, que nos sirve para realizar consultas a las bases de datos MySQL [24].

La ejecución de PHP se realizó utilizando el software Eclipse-Galileo, el mismo que es un entorno de desarrollo integrado de código abierto multiplataforma para aplicaciones basada en navegadores, dispone de un editor de texto con resaltado de sintaxis y la compilación es en tiempo real [25].

En el Eclipse Galileo se añadió el plugin de PHP que es un lenguaje de programación interpretado, el mismo que es usado principalmente en interpretación del lado del servidor. En éste se generaron clases para acceder a ciertas tablas de la base de datos ShoppingCar generada en phpMyAdmin. La idea de este código es para tener una clase asociada con cada uno de los campos de las tablas de una base de datos, de forma que la propia clase contenga todo el código necesario para conectar a la base de datos, así como poder crear nuevos registros, eliminarlos, actualizarlos e incluso buscar en la tabla. También con la clase generada se puede acceder a cada uno de los campos mediante un índice, el cual puede ser numérico (al estilo de un array) o bien alfanumérico, es decir, se puede acceder al contenido de un campo (columna) indicando el nombre de dicha columna.

La clase generada por esta utilidad tendrá una serie de propiedades y métodos que permitirán trabajar con los datos de cada fila de la tabla, algunos serán de instancia y otros serán estáticos o compartidos, de forma que se acceda a ellos sin necesidad de crear un nuevo objeto.

El proyecto PHP creado para la aplicación fue llamado appEjemplo, éste va relacionado con la base de datos anteriormente expuesta, donde las tablas generadas en ShoppingCart fueron creadas como clases, respetando los mismos parámetros que se establecieron en la base de datos. En la Figura 5.6 se puede observar en el lado izquierdo el contenido desplegado de appEjemplo.php y en el lado derecho la clase clsAccesoDatos.php, que es una clase exclusiva para la conexión, ésta permite la comunicación con el servidor y por ende ingresar exitosamente a la base de datos.

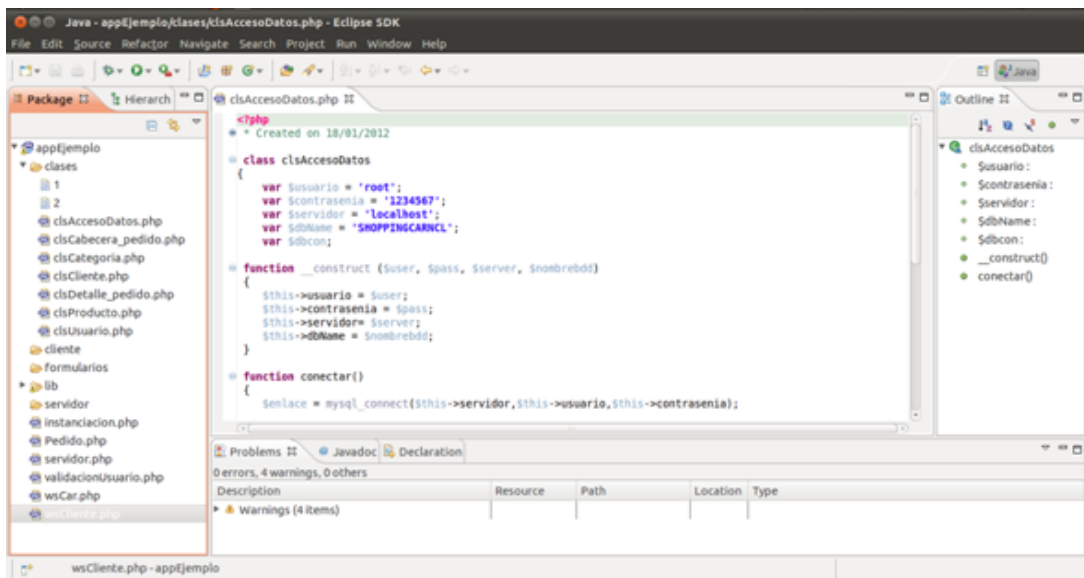


Figura 5.6: Contenido del Proyecto appEjemplo.php y visualización de la clase clsAccesoDatos.php.

5.6. Conexión a la Base de Datos desde Lua a través de un Web Service

Para la comunicación entre lua y la base de datos, se genera en appEjemplo un documento de tipo php llamado wsCar que viene hacer el Web Service a consumir, el mismo que contiene los parámetros necesarios para realizar el registro de compra, que son: cliente, producto y cantidad.

En la Figura 5.7, se puede observar a wsCar.php dentro de appEjemplo.php, el mismo que será detallado posteriormente.

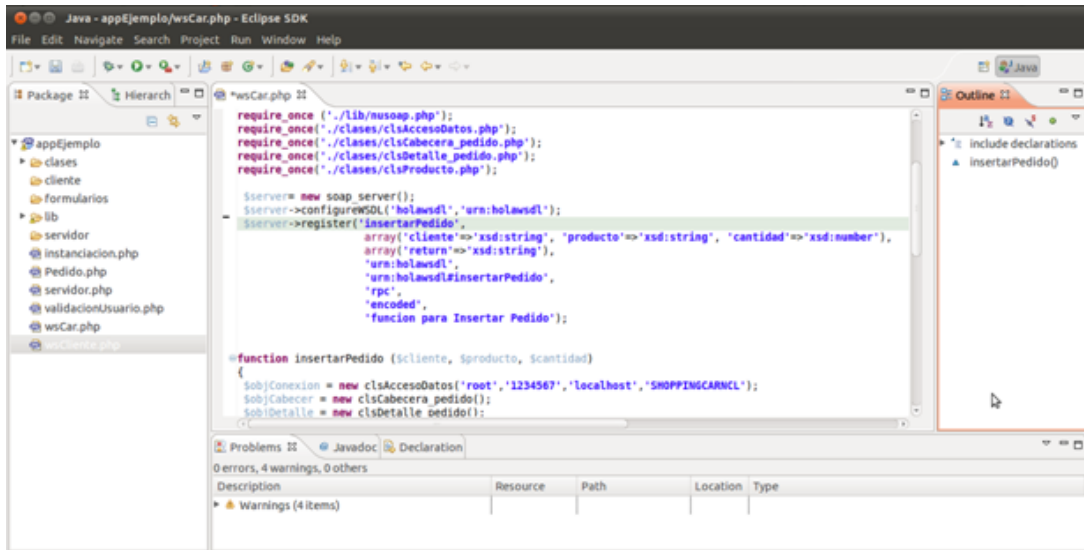


Figura 5.7: Web Service wsCar.php

Web Service: Es una aplicación que puede ser descrita, publicada, localizada e invocada a través de una red, generalmente Internet, utilizando un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

En la tabla 5.2, se especifican los protocolos que maneja un Web Service.

Tabla 5.2: Pila de Protocolos de los Web Service

Protocolo	Descripcion
UDDI	Ofrece un directorio de servicios de Internet
WSDL	Ofrece un modo de definir los servicios
SOAP	Permite invocar métodos de los servicios
XML	Permite enviar y recibir mensajes
HTTP, SMTP	Dan soporte a las capas superiores

Una vez creado el servicio, para conseguir que sea accesible por los consumidores, es necesario describirlo utilizando un lenguaje estándar llamado WSDL (Web Service Description Language) [26].

Los clientes del servicio podrán estar creados en cualquier lenguaje y ejecutarse sobre cualquier sistema operativo y hardware, lo único necesario es que sean capaces de obtener y entender la descripción WSDL de un servicio.

Un archivo WSDL es, en realidad, un archivo XML en el que se identifica el servicio y se indica el esquema para poder utilizarlo, así como el protocolo o protocolos que es posible utilizar.

Una vez que se dispone de esta información, el cliente puede comunicarse con el servicio utilizando en nuestro caso el protocolo SOAP.

Además de describir un servicio para que pueda ser utilizado por los clientes es importante publicar el servicio de modo que pueda ser encontrado por clientes que no conozcan necesariamente el componente que ofrece el servicio, pero que busquen un servicio de sus características. Esto se logra mediante el estándar UDDI (Universal Description, Discovery and Integration Registry) [27]. Realmente se trata de un servicio mundial en el que los proveedores de servicios pueden registrarlos de modo gratuito.

A continuación, en la Figura del Código de Programa 5.8, se mostrarán las librerías y las clases que requiere el Web Service para poder conectarse con la base de datos ShoppingCar.

```
require_once ('./lib/nusoap.php');
require_once ('./clases/clsAccesoDatos.php');
require_once ('./clases/clsCabecera_pedido.php');
require_once ('./clases/clsDetalle_pedido.php');
require_once ('./clases/clsProducto.php');
```

Figura 5.8: Código de Programa. Librerías y Clases requeridas por el Web Service wsCar.php

Como se puede observar en el Código de Programa 5.9, instanciamos la variable \$server como un objeto soap_server e iniciamos el soporte WSDL, bautizando a nuestro servicio con un namespace de holawsdl.

Y se establece los parámetros necesarios en wsCar.php como: el método insertarPedido, además de las variables cliente, producto como tipo string y la variable cantidad como tipo number y una variable de retorno tipo string.

En el Código de Programa 5.10 se muestra la función insertarPedido, que recibe tres parámetros; cliente, producto, cantidad. Y las operaciones correspondientes para realizar un registro de compra correcto en nuestra base de datos, retornando un mensaje que nos indique el estado del mismo.

El servidor permite abrir un canal para recuperar información, hay que tener en cuenta

```
$server= new soap_server();
$server->configureWSDL('holawsdl','urn:holawsdl');
$server->register('insertarPedido',
                array('cliente'=>'xsd:string',
'producto'=>'xsd:string', 'cantidad'=>'xsd:number'),
                array('return'=>'xsd:string'),
                'urn:holawsdl',
                'urn:holawsdl#insertarPedido',
                'rpc',
                'encoded',
                'funcion para Insertar Pedido');
```

Figura 5.9: Código de Programa. Instanciación de objetos, namespace y variables en el Web Service.

que el servidor con el que trabajemos tenga compatibilidad con el PHP y el Simple Object Access Protocol (Soap), el mismo que es un protocolo que permite el intercambio de información estructurada en la implementación de servicios Web en redes informáticas, utiliza el lenguaje XML para su formato de mensaje, basándose en la capa de aplicación en particular con los protocolos de HTTP y Simple Mail Transfer Protocol (SMTP), permite actualizar desde el encendido, para la negociación y transmisión de mensajes.

5.7. Programación en Lua para consumir un Web Service

Para utilizar el NCLua SOAP, sólo se tiene que añadir la línea `ncluasoap` al programa lua como se muestra en el Código de Programa 5.11.

Se declararon las variables `lcliente`, `lproducto` y `lcantidad`, como se muestra en el Código del Programa 5.12, donde `lcliente` es el usuario de la aplicación de compra ó telespectador, `lproducto` es el código del producto registrado en `ShoppingCar` y `lcantidad` la cantidad de productos en venta, que en nuestro caso es sólo una. Estas variables serán posteriormente enviadas al Web Service a través de una tabla en lua.

La función principal del módulo es `call (ncluasoap.call)`, que genera y envía una solicitud SOAP y consigue el retorno XML, que se convierte en una tabla de lua (con el módulo `LuaXML`) para facilitar el acceso a los datos de retorno con una llamada a un método remoto.

En el Código de Programa 5.13 se realiza la llamada a `ncluasoap` enviando los parámetros

```

function insertarPedido ($cliente, $producto, $cantidad)
(
    $objConexion = new clsAccesoDatos
('root', '1234567', 'localhost', 'SHOPPINGCARNCL');
    $objCabecer = new clsCabecera_pedido();
    $objDetalle = new clsDetalle_pedido();
    $objProducto = new clsProducto();
    $precio_unitario=0;
    $objConexion->conectar();
    $cod_pedido = $objCabecer->generaNumeroCabecera(8, 'EC');
    $id_cliente = $cliente;
    $fecha = $objCabecer->generarFecha();
    $cod_pro = $producto;
    //$cantidad=1;
    $precio_unitario = $objProducto->retornaPrecioUnitario
($cod_pro);
    $total = $cantidad * $precio_unitario;
    $mensaje = "";
    if ($objCabecer->insertar($objConexion->dbcon, $cod_pedido,
$id_cliente, $fecha, $total))
    (
        if ($objDetalle->insertar($cod_pedido, $cod_pro, $cantidad,
$precio_unitario))
        (
            $mensaje = "EL PEDIDO:". $cod_pedido. "se ha ingresado
exitosamente";
        )
        else
        (
            $mensaje = 'EL PEDIDO - DET:'. $cod_pedido. ' ha sido
rechazado: error:'. mysql_error() ;
        )
    )
)

```

Figura 5.10: Código de Programa. Librerías y clases requeridas por el Web Service ws-Car.php Funcion insertarPedido del Web Service.

```

package.path = package.path .. ';lib/?..lua'
require "ncluasoap"

```

Figura 5.11: Código de Programa. Añadiendo ncluasoap al programa de Lua.

ros, msg(Tabla de Lua), getResponse (función que imprime en formato XML el mensaje que retorna el Web Service) y soapVersion (Version del soap).

El parámetro principal de la función ncluasoap.call es msgTable, que debe ser una tabla en lua que contiene los datos para acceder al método Web Service. En el Código del Programa 5.14 se observa la estructura de msg.


```
local lcliente="17162955330"
local lproducto="ZP001"
local lcantidad="1"
```

Figura 5.12: Código de Programa. Variables locales dentro de la programación de Lua.

```
local soapVersion = "1.1"
ncluasoap.call(msg, getResponse, soapVersion)
```

Figura 5.13: Código de Programa. Llamada a ncluasoap.

```
local msg = {
  address = "http://10.1.30.197/AppEjemplo/wsCar.php",
  namespace = "http://10.1.30.197/holawsdl",
  operationName = "insertarPedido",
  params = {
    cliente = lcliente,
    cantidad = lproducto,
    producto = lcantidad
  }
}
```

La dirección IP es la de la máquina donde se halla el servidor

Figura 5.14: Código de Programa. Conexión con el servidor desde Lua

Como se puede apreciar en address se coloca el url del servidor, en namespace va el nombre como está en WSDL, en operationName se sitúa el nombre del método remoto al que se desea acceder en este caso insertarPedido. Y por último en params, se ubican los parámetros de entrada tal como se definen en el Web Service, los cuales se detallaron anteriormente. En principio se trabajó con un servidor local, por ello se colocó en address la dirección IP de la máquina que funcionaba como servidor local.

5.8. Entorno de Programación en GINGA-NCL

Para empezar se estructura un documento NCL, que permitirá representar elementos multimedia, sincronizarlos en tiempo y espacio, para crear nuestra aplicación interactiva.

Para ello en el software Eclipse HELIOS, se creó un proyecto NCL cuyo nombre es

TESIS_COMPRA y dentro de éste se generó un NCL document, que llevará por nombre main.ncl, en el que se desarrollará la aplicación. En la Figura 5.15, en el lado izquierdo, se observa el proyecto creado y su contenido desplegado, mientras que en el lado derecho est parte del main.ncl.



Figura 5.15: Proyecto NCL TESIS_COMPRA

El lugar donde se van a reproducir los objetos multimedia se denomina región, la cual permite inicializar la posición que utilizarán los nodos multimedia. En el Código de Programa 5.16 se puede observar las regiones que fueron creadas en la aplicación de compra.

Cómo se reproducen éstos objetos multimedia, simple a través de descriptores que indican en que región se mostrará el objeto multimedia y con que propiedades. En el Código de Programa 5.17 se muestra los descriptores asociados a las regiones anteriormente creadas.

Con la ayuda de links y conectores se establece el sincronismo entre los nodos, para determinar cuando se va a reproducir los nodos multimedia. En el Código de Programa 5.18, se presentan los conectores que se utilizaron en la aplicación.

En el body un Port que apunta a un objeto multimedia, indica que comenzará la presentación del mismo al iniciar la aplicación. En nuestro caso se instanciaron cuatro Ports, que mostrarán el menú principal de la aplicación, como se indica en el Código del Programa 5.19.

A continuación se declararon los nodos multimedia que fueron utilizados, con sus respectivos atributos, como se muestra en el Código de Programa 5.20.

```

<!-- Generated by NCL Eclipse -->
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
  <regionBase>

    <region id="rop1" left="2%" bottom="1%" height="25%" width="32%" zIndex="2"/>
    <region id="rop2" left="34%" bottom="1%" height="25%" width="32%" zIndex="2"/>
    <region id="rop3" left="66%" bottom="1%" height="25%" width="32%" zIndex="2"/>
    <region id="rvideo" left="0%" top="0%" height="100%" width="100%" zIndex="1"/>

    <region id="rimal" left="0%" top="0%" height="22%" width="35%" zIndex="3"/>
    <region id="rima2" left="0%" top="22%" height="22%" width="35%" zIndex="3"/>
    <region id="rima3" left="0%" top="44%" height="22%" width="35%" zIndex="3"/>
    <region id="rsa" left="9%" top="83%" height="15%" width="25%" zIndex="3"/>

    <region id="rbver" left="65%" top="60%" height="15%" width="15%" zIndex="3"/>

    <region id="rsall" left="9%" top="83%" height="15%" width="25%" zIndex="5"/>
    <region id="rlua" right="70%" top="75%" height="20%" width="40%" zIndex="5"/>
    <!-- Boton Informacion -->
    <region id="rvinf" left="0%" top="0%" height="100%" width="100%" zIndex="3"/>
    <region id="rsal2" left="82%" top="82%" height="17%" width="17%" zIndex="4"/>

  </regionBase>

```

Figura 5.16: Código de Programa. Regiones creadas en main.ncl.

```

<descriptorBase>
<descriptor id="dop1" region="rop1" focusIndex="1" moveLeft="3" moveRight="2" focusBorderColor="red":
  <descriptorParam name="transparency" value="70%"/>
</descriptor>
<descriptor id="dop2" region="rop2" focusIndex="2" moveLeft="1" moveRight="3" focusBorderColor="red":
  <descriptorParam name="transparency" value="70%"/>
</descriptor>
<descriptor id="dop3" region="rop3" focusIndex="3" moveLeft="2" moveRight="1" focusBorderColor="red":
  <descriptorParam name="transparency" value="70%"/>
</descriptor>

<descriptor id="dvideo" region="rvideo"/>
<descriptor id="dimal" region="rimal"/>
<descriptor id="dima2" region="rima2"/>
<descriptor id="dima3" region="rima3"/>
<descriptor id="dsa" region="rsa"/>

<descriptor id="dlua" region="rlua"/>
<descriptor id="dsall" region="rsall"/>

<descriptor id="dinfo" region="rvinf"/>
<descriptor id="dsal2" region="rsal2"/>
<descriptor id="dbver" region="rbver"/>

</descriptorBase>

```

Figura 5.17: Código de Programa. Descriptores

Al seleccionar la opción de **COMPRA** en el menú principal, se presentará un submenú donde se encontrarán los tres productos en venta. En el Código de Programa

Al elegir el primer producto (Cámara Sony), con el botón Rojo del Control Remoto, se realizará la compra. Por ello al ocurrir éste evento, debe ser iniciado el archivo.lua asociado

```

<connectorBase>
<causalConnector id="selcdeteclaPararIniciar">
<connectorParam name="keyCode"/>
<simpleCondition role="onSelection" key="$keyCode"/>

<compoundAction operator="seq">
<simpleAction role="stop" max="unbounded" qualifier="par"/>
<simpleAction role="start" max="unbounded" qualifier="par"/>
</compoundAction>
</causalConnector>

<causalConnector id="onBeginStart">
<simpleCondition role="onBegin"/>
<simpleAction role="start" max="unbounded" qualifier="par"/>
</causalConnector>

<causalConnector id="onBeginStartN">
<simpleCondition role="onBegin"/>
<simpleAction role="start" max="unbounded" qualifier="par"/>
</causalConnector>

<causalConnector id="onKeySelecionStop">
<connectorParam name="keyCode"/>
<simpleCondition role="onSelection" key="$keyCode"/>
<simpleAction role="stop" max="unbounded" qualifier="par"/>
</causalConnector>

<causalConnector id="onEndStop">
<simpleCondition role="onEnd"/>
<simpleAction role="stop" max="unbounded"/>
</causalConnector>

```

Figura 5.18: Código de Programa. Conectores que serán utilizados para reproducir los nodos multimedia.

```

<port id="comp1" component="comp"/>
<port id="info1" component="info"/>
<port id="sal11" component="sal"/>
<port id="s1" component="video"/>

```

Figura 5.19: Código de Programa. Ports definidos en main.ncl

al producto para establecer la conexión con la base de datos ShoppingCar y así registrar la compra, como se indica en el Código de Programa 5.21.

Al elegir el segundo producto (Samsung tablet), con el botón Azul del Control Remoto,

```

<media id="video" src="media/COMP.png" descriptor="dvideo">
  <property name="bounds"/>
</media>

<media id="comp" src="media/compra.png" descriptor="dop1"/>
<media id="info" src="media/informacion.png" descriptor="dop2"/>
<media id="sal" src="media/salir.png" descriptor="dop3"/>

<media id="ima1" src="media/brojo.png" descriptor="dima1"/>
<media id="ima2" src="media/bazul.png" descriptor="dima2"/>
<media id="ima3" src="media/bamarillo.png" descriptor="dima3"/>
<media id="sa" src="media/volver1.png" descriptor="dsa"/>

<media id="lua1" src="luaUno.lua" descriptor="dlua"/>
<media id="lua2" src="luaDos.lua" descriptor="dlua"/>
<media id="lua3" src="luaTres.lua" descriptor="dlua"/>
<media id="sali1" src="media/volver1.png" descriptor="dsali"/>
<media id="sali2" src="media/volver1.png" descriptor="dsali"/>
<media id="sali3" src="media/volver1.png" descriptor="dsali"/>

<media id="infor" src="media/COMP.png" descriptor="dinfo"/>
<media id="sal2" src="media/volver.png" descriptor="dsal2"/>

<media id="bver" src="media/kit.png" descriptor="dbver">
  <area id="areal" begin="10s" end="15s"/>
  <area id="area2" begin="8s" end="10s"/>
  <property name="bounds"/>
</media>

```

Figura 5.20: Código de Programa. Media definidos en main.ncl

se realizará la compra. Por ello al ocurrir éste evento, debe ser iniciado el archivo.lua asociado al producto en éste caso el archivo lua2.lua para establecer la conexión con la base de datos ShoppingCar y así registrar la compra, como se indica en el Código de Programa 5.22.

Al elegir el tercer producto (Macbook Air), con el botón Amarillo del Control Remoto, se realizará la compra. Por ello al ocurrir éste evento, debe ser iniciado el archivo.lua asociado al producto en éste caso el archivo lua3.lua, como se indica en el Código de Programa 5.23.

Al seleccionar en el menú principal de la aplicación, la opción SALIR, se ejecutará las siguientes líneas de programación como se muestra en el Código de Programa 5.24, para salir de la aplicación.

```

<!-- Primer Producto boton rojo -->
<link xconnector="selcdeteclaPararIniciar">

<bind role="onSelection" component="ima1">
  <bindParam name="keyCode" value="RED"/>
</bind>
  <bind component="ima2" role="stop"/>
  <bind component="ima1" role="start"/>
  <bind component="ima3" role="stop"/>
  <bind component="video" role="start"/>
  <bind role="start" component="lua1"/>
  <bind role="stop" component="lua2"/>
  <bind role="stop" component="lua3"/>

<bind role="start" component="sal11">
  </bind>
  <bind role="start" component="bver"/>
</link>

```

Figura 5.21: Código de Programa. Programación asociada al evento Presionar Botón Rojo, donde se inicializará el archivo lua1.lua.

```

<!-- Segundo Producto boton azul -->
<link xconnector="selcdeteclaPararIniciar">

<bind role="onSelection" component="ima2">
  <bindParam name="keyCode" value="BLUE"/>
</bind>
  <bind component="ima1" role="stop"/>
  <bind component="ima3" role="stop"/>
  <bind component="ima2" role="start"/>
  <bind component="video" role="start"/>
  <bind component="lua2" role="start"/>
  <bind role="stop" component="lua1"/>
  <bind role="stop" component="lua3"/>
  <bind role="start" component="sal12"/>
  <bind role="start" component="bver"/>
</link>

```

Figura 5.22: Código de Programa. Programación asociada al evento Presionar Boton Azul, donde se inicializará el archivo lua2.lua.

5.9. Entorno de Programacion en LUA

Como anteriormente se describió el código de programación que debería ir en el lenguaje Lua, en ésta sección se hará hincapié en los parámetros que deben modificarse en los

```

<!-- Tercer Producto boton amarillo -->
<link xconnector="selcdeteclaPararIniciar">

<bind role="onSelection" component="ima3">
<bindParam name="keyCode" value="YELLOW"/>
</bind>
<bind component="ima2" role="stop"/>
<bind component="ima1" role="stop"/>
<bind component="ima3" role="start"/>
<bind component="video" role="start"/>
<bind component="lua3" role="start"/>
<bind role="stop" component="lua1"/>
<bind role="stop" component="lua2"/>
<bind role="start" component="sal13"/>
<bind role="start" component="bver"/>
</link>

```

Figura 5.23: Código de Programa. Programación asociada al evento Presionar Botón Amarillo, donde se inicializará el archivo lua3.lua.

```

<!-- OPCION SALIR-->

<link xconnector="onSelectionSetStartStop">

<bind role="onSelection" component="sa">
<bindParam name="key" value="CURSOR_LEFT"/>
</bind>
<bind component="sal" role="start"/>
<bind component="sa" role="stop"/>
<bind component="ima1" role="stop"/>
<bind component="ima2" role="stop"/>
<bind component="ima3" role="stop"/>
<bind component="comp" role="start"/>
<bind component="info" role="start"/>
<bind component="video" interface="bounds" role="set">
<bindParam name="var" value="0%,0%,150%,150%"/>
</bind>
</link>

```

Figura 5.24: Código de Programa. Programación para salir de la aplicación interactiva.

diferentes archivos.lua, que fueron creados para realizar la compra de cada uno de los productos. Además en address y namespace se ha cambiado al url del servidor que fue publicado, es decir se dejó a un lado el uso de un servidor local.

En el Código de Programa 5.25 se puede observar las variables declaradas en el archivo lua1.lua, que corresponden al primer producto que es la Cámara Sony, que posteriormente serán enviadas dentro de local msg, como parámetros de ésta tabla, hacia el Web Service.

```
local lcliente="17162955330"
local lproducto="2P001"
local lcantidad="1"
local msg = {
  address = "http://190.15.140.12/appEjemplo/wsCar.php",
  namespace = "http://190.15.140.12/holawsdl",
  operationName = "insertarPedido",
  params = {
    cliente = lcliente,
    cantidad = lproducto,
    producto = lcantidad
  }
}

local soapVersion = "1.1"
ncluasop.call(msg, getResponse, soapVersion)
```

Figura 5.25: Código de Programa. Parámetros correspondientes al primer producto, declaradas en el archivo lua1.lua.

En el Código de Programa 5.26 se indica las variables declaradas en el archivo lua2.lua, que corresponden al segundo producto que es la Samsung tablet, que posteriormente serán enviadas dentro de local msg, como parámetros de ésta tabla, hacia el Web Service.

En el Código de Programa 5.27 se indica las variables declaradas en el archivo lua3.lua, que corresponden al tercer producto que es la Macbook Air, que posteriormente serán enviadas dentro de local msg, como parámetros de ésta tabla, hacia el Web Service.


```
local lcliente="17162955330"
local lproducto="CM001"
local lcantidad="1"
local msg = {
  address = "http://190.15.140.12/appEjemplo/wsCar.php",
  namespace = "http://190.15.140.12/holawsdl",
  operationName = "insertarPedido",
  params = {
    cliente = lcliente,
    cantidad = lproducto,
    producto = lcantidad
  }
}

local soapVersion = "1.1"
ncluasoap.call(msg, getResponse, soapVersion)
```

Figura 5.26: Código de Programa. Parámetros correspondientes al segundo producto, declaradas en el archivo lua2.lua.

```
local lcliente="17162955330"
local lproducto="HG001"
local lcantidad="1"
local msg = {
  address = "http://190.15.140.12/appEjemplo/wsCar.php",
  namespace = "http://190.15.140.12/holawsdl",
  operationName = "insertarPedido",
  params = {
    cliente = lcliente,
    cantidad = lproducto,
    producto = lcantidad
  }
}

local soapVersion = "1.1"
ncluasoap.call(msg, getResponse, soapVersion)
```

Figura 5.27: Código de Programa. Parámetros correspondientes al tercer producto, declaradas en el archivo lua3.lua.

5.10. Corrida de la Aplicacion ShoppingCar

Para el presente proyecto de tesis, se ha desarrollado una aplicación interactiva con canal de retorno, que tiene como objetivo la inclusión de la sociedad en la TDT, para nuestro caso en particular, la compra de tres productos diferentes, siendo registrada en una base de datos, enfocándose al T-Commerce ; que es la compra a través de las pantallas del televisor. A continuación se va a dar una breve explicación de la forma correcta de navegar en la aplicación ShoppingCar. Primero para cargar la aplicación, se utiliza el botón Menú. Posteriormente se selecciona la opción Ginga. Finalmente, se selecciona la aplicación ShoppingCar presionando el botón OK del control remoto. Una vez cargada la aplicación aparece el video de presentación de los productos a vender y en la parte inferior de la pantalla tres opciones: Compra, Información y Salir, tal y como se muestra en la Figura 5.28. Para cambiar entre opciones utilizamos las teclas de izquierda a derecha y para seleccionarlas el botón OK.

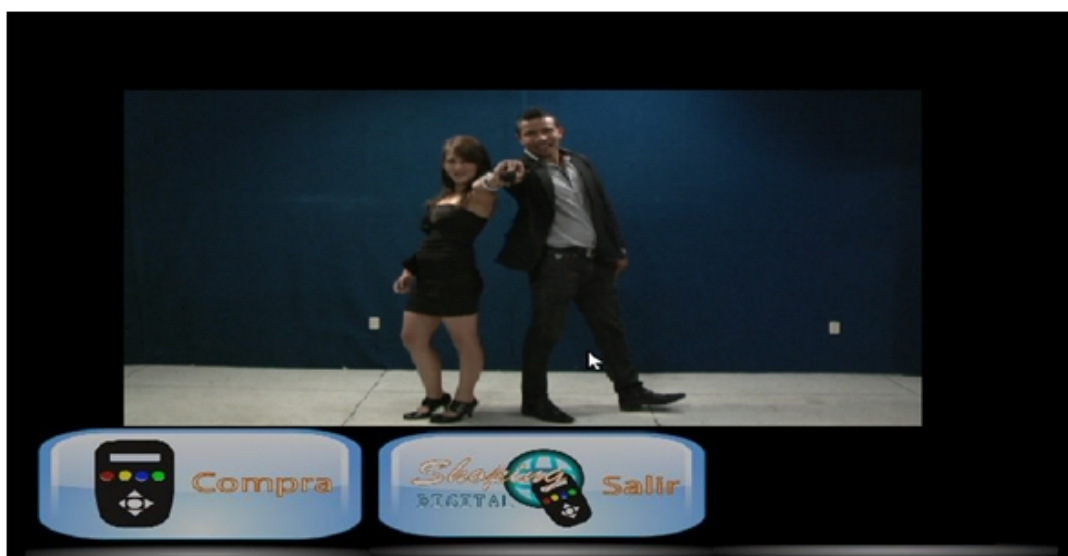


Figura 5.28: Aplicacion ShoppingCar, con las dos opciones

Seleccionando la opción Compras, aparecen tres opciones en la parte izquierda de la pantalla: Cámara Sony, Samsung tablet y una Macbook Air. Para realizar la compra, cada uno de los artículos está asociado a un indicador de diferente color. La cámara Sony está asociada al color rojo, la Samsung al azul y la Macbook al amarillo; así cuando se desee realizar la compra se debe presionar los colores del control remoto que corresponda al producto que se va a realizar la compra. En la Figura 5.29 se muestra las opciones de compras en la aplicación ShoppingCar. Una vez realizada la compra aparecerá un mensaje de confirmación en la pantalla, en la parte inferior izquierda. Con el botón volver se regresa

al menú principal.

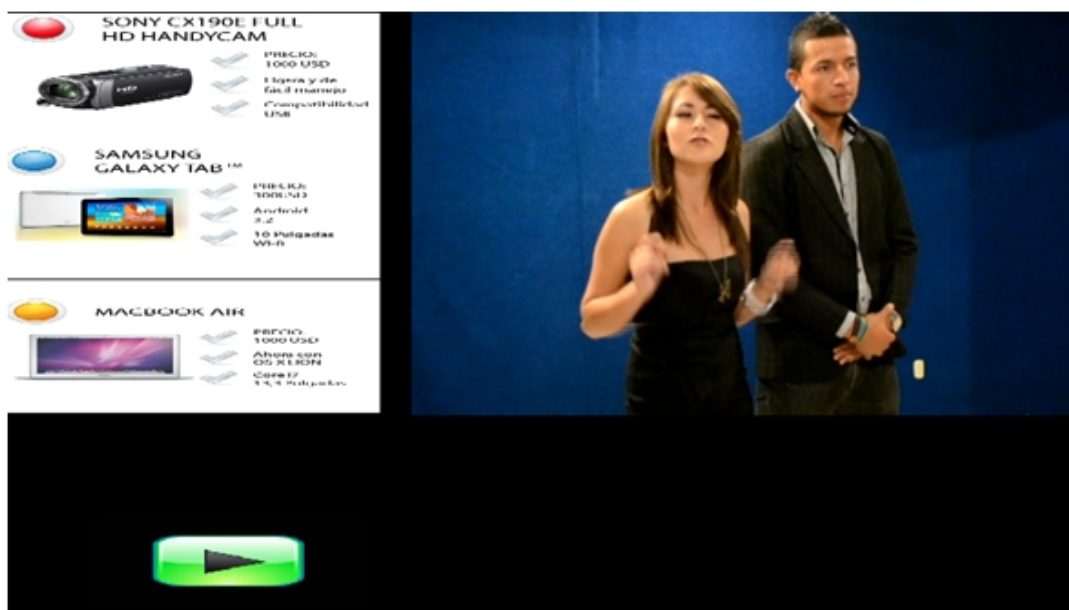


Figura 5.29: Aplicacion ShoppingCar, con los tres productos de compra

Cuando se elige la opción salir, se termina la aplicación.

Capítulo 6

ANÁLISIS DE RESULTADOS DEL CANAL DE RETORNO

6.1. Análisis del Canal de Retorno

En el presente capítulo se realizará el análisis del canal de retorno, de la aplicación ShoppingCar descrita en el capítulo 4, en la Figura 6.1, se muestra el esquema de red, donde el Set-Top-Box está representado por la máquina virtual de Ginga NCL, el servidor wsCar, que está levantado con una IP pública 190.15.140.12, en el laboratorio de sistemas de la Escuela Politécnica del Ejército, y el medio de conexión entre el servidor y la máquina virtual, es el internet a través de la interfaz Ethernet. Se debe enfatizar que en la máquina virtual previamente se cargó la aplicación ShoppingCar.

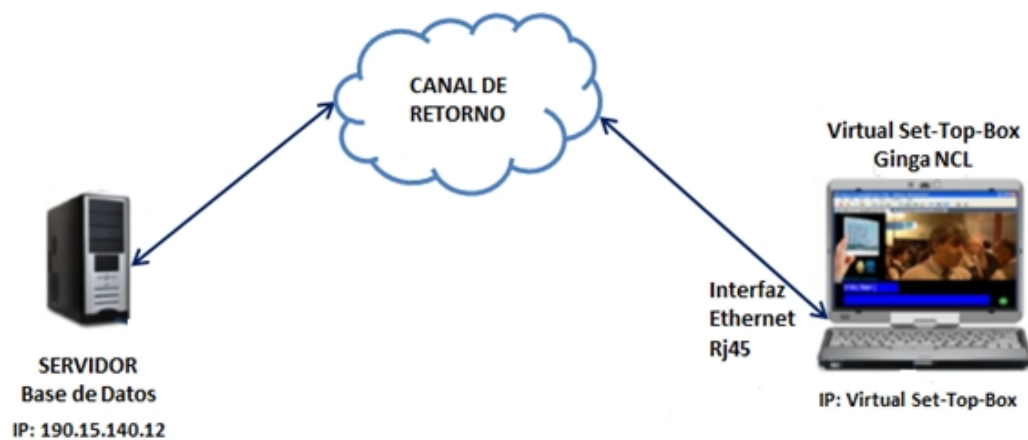


Figura 6.1: Estructura de red para el uso del Canal de Retorno.

Se tomaron en consideración dos entornos para el análisis del canal de retorno: el caso A, es ejecutando la aplicación dentro de la red donde se encuentra levantado el servidor, y

el caso B es al ejecutando la aplicación desde un domicilio ubicado en la ciudad de Quito. En la Figura 6.2, se puede observar el caso A, donde el Virtual Set-Top-Box Ginga NCL, tiene como IP 192.168.15.123.

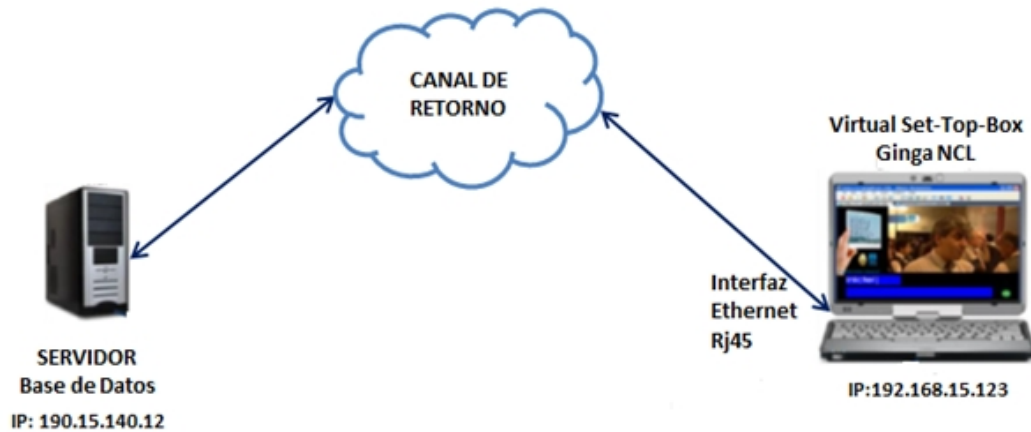


Figura 6.2: Estructura de red para el uso de Canal de Retorno, dentro de la Escuela Politécnica del Ejército.

La herramienta que fue utilizada para el respectivo análisis de red en cada uno de los entornos planteados, es Wireshark, éste analizador intenta capturar paquetes de red, para poder mostrarlos en detalle, como sea posible. Muestra los datos a través de un entorno gráfico y de forma amigable y entendible [28].

En principio se analizaron los parámetros de los protocolos de red que intervienen al momento de enviar la solicitud al Servidor, para realizar la compra de uno de los tres productos, es decir, cuando se ejecutó la llamada al Servidor desde Lua.

En la Figura 6.3 se muestra el ruteo del protocolo IPv4, que es un protocolo usado para la comunicación de datos a través de una red [29]. Dentro del datagrama de éste protocolo se tiene la dirección IP de origen (src: 192.168.15.123), que representa la dirección IP del equipo remitente y permite que el destinatario responda y la dirección IP de destino (190.15.140.12), dirección del destinatario del mensaje.

En la Figura 6.4 se muestra el formato de trama del protocolo TCP, que posibilita la administración de los datos que vienen del nivel más bajo, o que van hacia él, es decir, el protocolo IP [30]. Dentro de los campos de TCP, está el puerto de origen (40684), puerto relacionado con la aplicación en curso en la máquina origen y el puerto de destino (80), puerto relacionado con la aplicación en curso en la máquina destino.

```

Frame 236: 717 bytes on wire (5736 bits), 717 bytes captured (5736 bits)
Ethernet II, Src: Vmware_8f:6b:5d (00:0c:29:8f:6b:5d), Dst: Cisco-Li_00:de:f6 (68:7f:74:d0:de:f6)
Internet Protocol Version 4, Src: 192.168.15.123 (192.168.15.123), Dst: 190.15.140.12 (190.15.140.12)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 703
  Identification: 0x6e32 (28210)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (6)
  Header checksum: 0xafc7 [correct]
  Source: 192.168.15.123 (192.168.15.123)
  Destination: 190.15.140.12 (190.15.140.12)

```

Figura 6.3: Captura del Protocolo IPv4, en el tráfico de red durante la ejecución de la aplicación.

```

Transmission Control Protocol, Src Port: 40684 (40684), Dst Port: http (80), Seq: 1, Ack: 1, Len: 651
  Source port: 40684 (40684)
  Destination port: http (80)
  [Stream index: 1]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 652 (relative sequence number)]
  Acknowledgement number: 1 (relative ack number)
  Header length: 32 bytes
  Flags: 0x018 (PSH, ACK)
  window size value: 183
  [Calculated window size: 5856]
  [window size scaling factor: 32]
  Checksum: 0xe4fd [validation disabled]
  Options: (12 bytes)
  [SEQ/ACK analysis]

```

Figura 6.4: Captura del Protocolo TCP, en el tráfico de red durante la ejecución de la aplicación.

En la Figura 6.5, se puede observar la solicitud HTTP, que describe lo que el navegador o cliente envía al servidor. Se envían los datos a través del comando POST, el mismo que se encarga de abrir la conexión entre la petición del cliente y el servidor. Otros parámetros existentes en la solicitud son: el User-Agent, que es la información sobre el cliente, el nombre (ncluasoap) y la versión (0.5.6.6). El ncluasoap es una librería dentro de la aplicación que se encarga de generar y enviar una solicitud SOAP hacia el servidor, Content-Type, que es el tipo de contenido del cuerpo de la solicitud, en éste caso es text/xml, y el SOAPAction, que define un URI desde donde se localizará el servicio, para nuestra aplicación es "http://190.15.140.12/holawsdl/insertarPedido".

En la Figura 6.6, se observa el formato de trama del protocolo XML, donde la versión utilizada es la 1.0, y su cuerpo describe la función insertarPedido, que contiene el namespace establecido en WSDL (http://190.15.140.12/holawsdl) y los tres parámetros que son enviados al servidor.

Los parámetros analizados en los siguientes protocolos, son los capturados al momento en que el Servidor da respuesta a la solicitud anteriormente descrita. En la Figura 6.7,

```

Hypertext Transfer Protocol
POST http://190.15.140.12/appEjemplo/wsCar.php HTTP/1.0\n
[Expert Info (Chat/Sequence): POST http://190.15.140.12/appEjemplo/wsCar.php HTTP/1.0\n]
Request Method: POST
Request URI: http://190.15.140.12/appEjemplo/wsCar.php
Request Version: HTTP/1.0
User-Agent: ncluasoap/0.5.6.6\n
Content-Type: text/xml; charset=utf-8\n
SOAPAction: "http://190.15.140.12/holawsdl/insertarPedido"\n
Host: 190.15.140.12\n
Content-Length: 425\n
[Content length: 425]
\n
[Full request URI: http://190.15.140.12http://190.15.140.12/appEjemplo/wsCar.php]

```

Figura 6.5: Captura de solicitud HTTP, en el tráfico de red durante la ejecución de la aplicación.

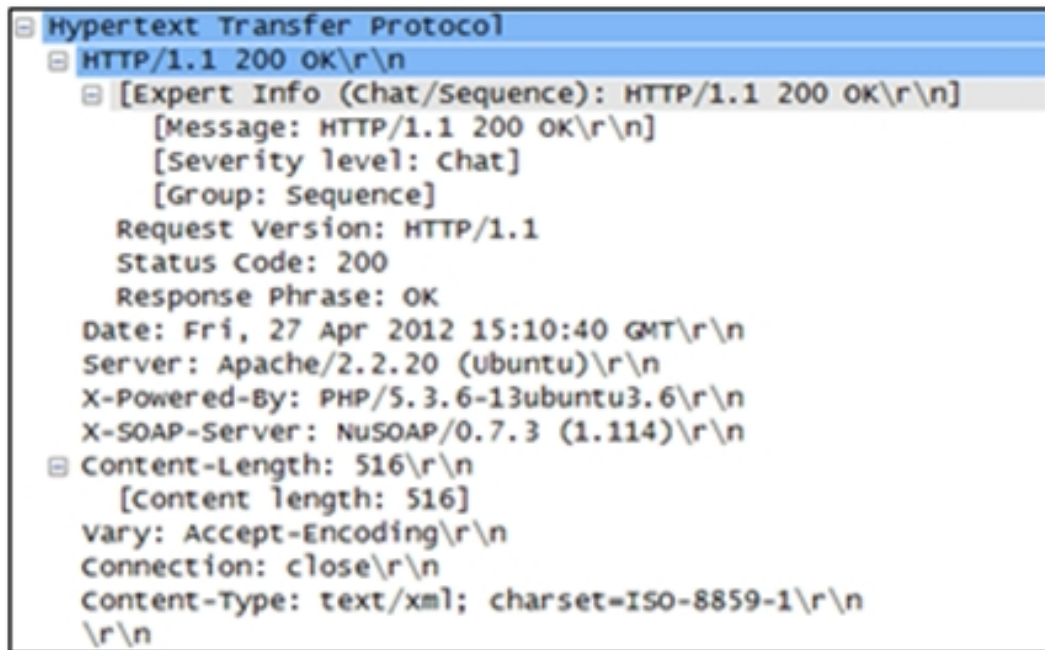
```

extensible Markup Language
<?xml
  version="1.0"
  ?>
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <insertarPedido
      xmlns="http://190.15.140.12/holawsdl">
      <cliente>
        17162955330
      </cliente>
      <cantidad>
        ZP001
      </cantidad>
      <producto>
        1
      </producto>
    </insertarPedido>
  </soap:Body>
</soap:Envelope>

```

Figura 6.6: Captura del Protocolo XML, en el tráfico de red durante la solicitud al servidor.

se presenta la respuesta HTTP, que el servidor envía al navegador, incluye una línea de estado, compuesta por la versión del protocolo utilizada, el estado de la solicitud en proceso mediante un texto explicativo y un código. Dentro del encabezado tenemos Status Code (200) y Response Phrase (OK), que indican que la solicitud se llevó a cabo de manera correcta. Server, indica las características del servidor que envió la respuesta, en éste caso, Apache (2.2.20). Content-Length, es la extensión del cuerpo de la respuesta y Content-Type, es el tipo de contenido del cuerpo de la respuesta, text/xml.



```
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
[Message: HTTP/1.1 200 OK\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Version: HTTP/1.1
Status Code: 200
Response Phrase: OK
Date: Fri, 27 Apr 2012 15:10:40 GMT\r\n
Server: Apache/2.2.20 (Ubuntu)\r\n
X-Powered-By: PHP/5.3.6-13ubuntu3.6\r\n
X-SOAP-Server: NuSOAP/0.7.3 (1.114)\r\n
Content-Length: 516\r\n
[Content length: 516]
Vary: Accept-Encoding\r\n
Connection: close\r\n
Content-Type: text/xml; charset=ISO-8859-1\r\n
\r\n
```

Figura 6.7: Captura de respuesta HTTP, en el tráfico de red durante la ejecución de la aplicación.

En la Figura 6.8, se observa el formato de trama del protocolo XML, cuando el Servidor envía la respuesta, donde la versión utilizada es la 1.0, y dentro de su cuerpo, se describe la función `insertarPedido`, que contiene el namespace (`http://190.15.140.12/holawsdl`), y la variable de retorno tipo string, es decir, retorna un mensaje que se ha ingresado exitosamente.

Una vez analizado los protocolos, se procedió a analizar el tráfico de red, determinando el ancho de banda máximo que requiere la aplicación `ShoppingCar`, su tasa de transmisión, los paquetes enviados y recibidos del servidor y el throughput. En la Figura 6.9, se observa el tráfico de red que se produjo al realizar las tres compras de la aplicación `ShoppingCar`, como se observa en la gráfica, existe varias oscilaciones en la transmisión, ya que como la aplicación está conectada al internet existen protocolos que siempre se están ejecutando, además, se observa que el tiempo que se demora al realizar las tres compras es aproximadamente de 120 segundos (2 min) y el ancho de banda máximo que ocupa la aplicación es 100kbps. Vale recalcar que en el tráfico de red, intervienen todos los recursos y protocolos existentes para poder conectarse a la red, a través del internet.

En la Figura 6.10, se observa el tráfico del protocolo de HTTP de la aplicación `ShoppingCar`, el mismo que permite enviar y recibir paquetes, al momento de realizar la


```

extensible Markup Language
  <?xml
    version="1.0"
    encoding="ISO-8859-1"
  ?>
  <SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns1:insertarPedidoResponse
      xmlns:ns1="http://190.15.140.12/holawsdl">
      <return
        xsi:type="xsd:string">
        EL PEDIDO:EC00000056se ha ingresado exitosamente
      </return>
    </ns1:insertarPedidoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Figura 6.8: Captura del protocolo XML, en el tráfico de red durante la respuesta del servidor.

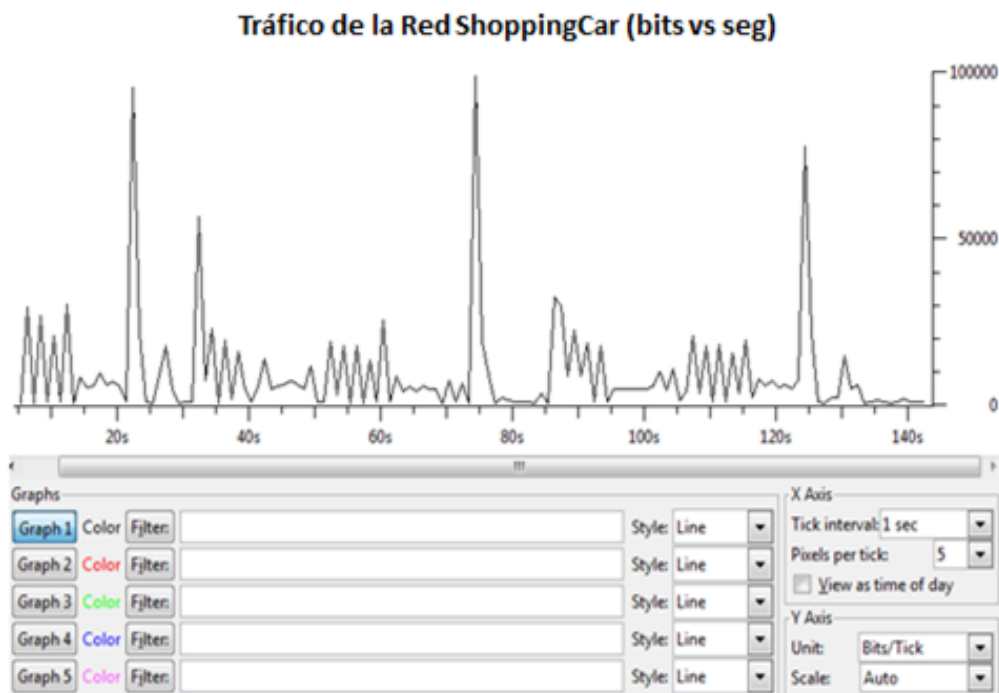


Figura 6.9: Captura del Tráfico de Red.

comprar por ello se obtiene en la gráfica tres oscilaciones, las que simbolizan la compra de los tres artículos, es decir, cada pico representa el envío de la petición de compra y la

recepción de la misma. Además se observa que el ancho de banda es de aproximadamente 20kpbs, ya que se captura sólo el tráfico generado por HTTP, durante la transmisión de los datos. Entre los factores que influyen en el ancho de banda son: las características del computador del servidor y del cliente.

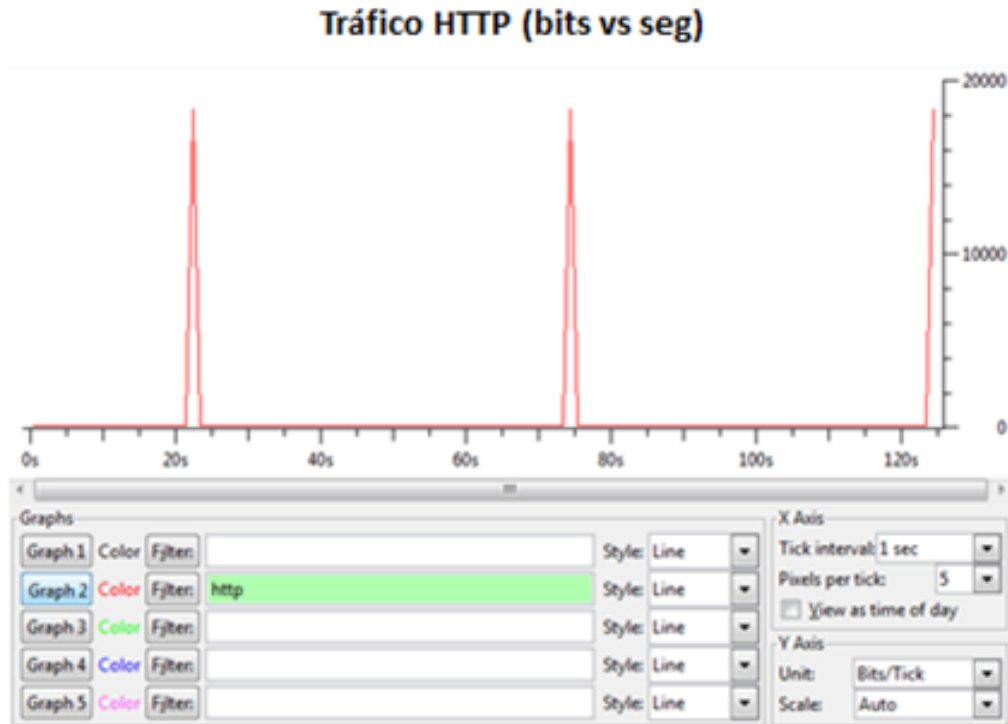


Figura 6.10: Captura del Protocolo HTTP en la corrida de la aplicación.

Con la Figura 6.11, se capturó la transmisión desde el host hasta el servidor, con lo cual se observa varias oscilaciones las que representan los protocolos de internet que se están llamando para la comunicación con el servidor.

Posteriormente, en la Figura 6.12, se observa la transmisión de la respuesta desde el servidor al host, por lo que en las gráficas se ven tres oscilaciones las cuales corresponden a las tres compras realizadas.

En la aplicación, cada llamado al servidor representa 12 paquetes enviados como realizamos tres compras equivale a 36 paquetes enviados al servidor, mientras que desde el servidor atender a cada solicitud de compra representa 5 paquetes, en total 15 paquetes, como se muestra en la Figura 6.13.

Como se puede observar en la Figura 6.14, la tasa de transmisión define un número de

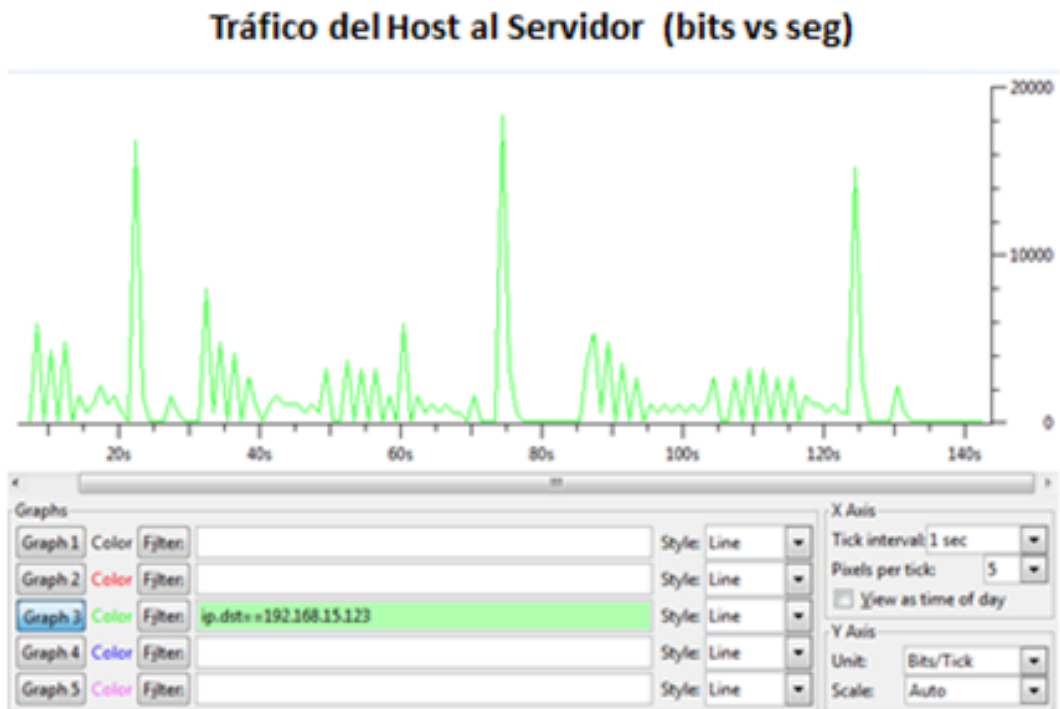


Figura 6.11: Captura de la transmisión de datos desde el host hasta el servidor.

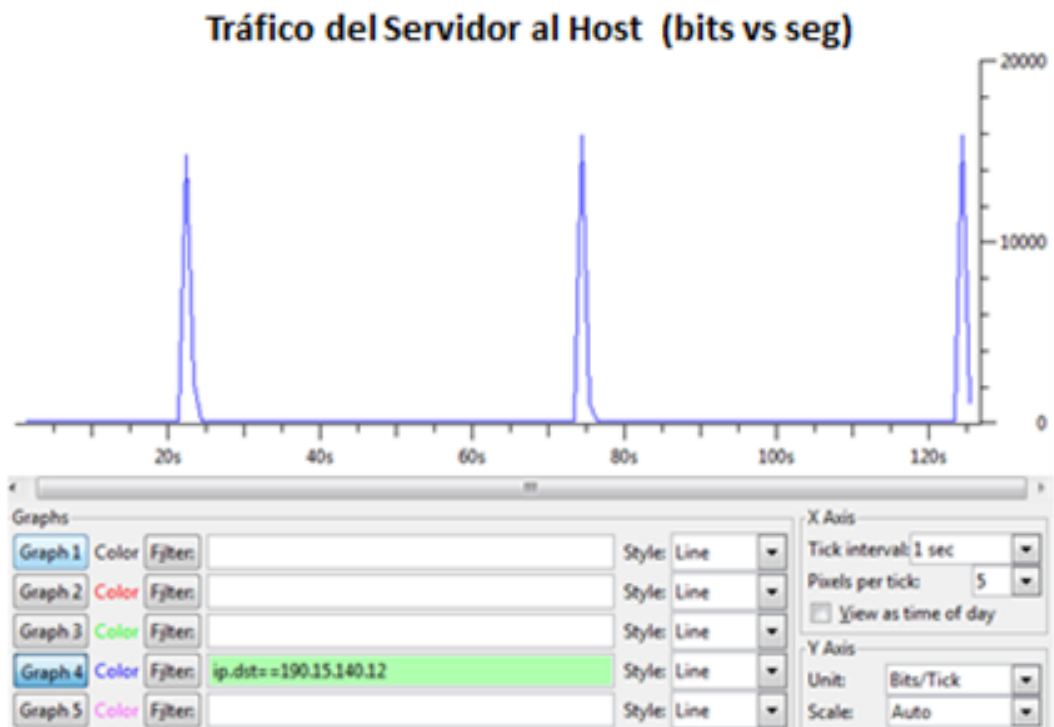


Figura 6.12: Captura de la transmisión de datos desde el servidor hasta el host.

TCP Endpoints									
Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
192.168.15.123	22	713	131614	365	108426	348	23188	-	-
192.168.15.122	53372	713	131614	348	23188	365	108426	-	-
190.15.140.12	80	51	9699	15	3369	36	6330	-	-
192.168.15.123	40684	17	3233	12	2110	5	1123	-	-
192.168.15.123	40685	17	3233	12	2110	5	1123	-	-
192.168.15.123	60671	17	3233	12	2110	5	1123	-	-

Figura 6.13: Captura del envío y recepción de paquetes.

bits que se transmite por unidad de tiempo, es decir, es la velocidad de transferencia de datos que existe en la aplicación ShoppingCar, que es de 9kbps.

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00 %	977	100,00 %	161535	0,009	0	0	0,000
Ethernet	99,49 %	972	99,72 %	161085	0,009	0	0	0,000
Internet Protocol Version 4	86,39 %	844	94,01 %	151855	0,009	0	0	0,000
Internet Group Management Protocol	1,23 %	12	0,35 %	566	0,000	12	566	0,000
Transmission Control Protocol	78,20 %	764	87,48 %	141313	0,008	385	25482	0,001
SSH Protocol	3,87 %	370	67,46 %	108976	0,006	370	108976	0,006
Hypertext Transfer Protocol	0,92 %	9	4,24 %	6855	0,000	0	0	0,000
eXtensible Markup Language	0,92 %	9	4,24 %	6855	0,000	3	2553	0,000
Hypertext Transfer Protocol	0,61 %	6	2,66 %	4302	0,000	0	0	0,000
Malformed Packet	0,61 %	6	2,66 %	4302	0,000	6	4302	0,000
User Datagram Protocol	6,96 %	68	6,18 %	9976	0,001	0	0	0,000
Dropbox LAN sync Discovery Protocol	3,38 %	33	3,36 %	5422	0,000	33	5422	0,000
Domain Name Service	1,02 %	10	1,01 %	1632	0,000	10	1632	0,000
NetBIOS Name Service	2,15 %	21	1,20 %	1932	0,000	21	1932	0,000
Bootstrap Protocol	0,10 %	1	0,21 %	342	0,000	1	342	0,000
NetBIOS Datagram Service	0,31 %	3	0,40 %	648	0,000	0	0	0,000
SMB (Server Message Block Protocol)	0,31 %	3	0,40 %	648	0,000	0	0	0,000
SMB MailSlot Protocol	0,31 %	3	0,40 %	648	0,000	0	0	0,000
Microsoft Windows Browser Protocol	0,31 %	3	0,40 %	648	0,000	3	648	0,000
Logical-Link Control	8,30 %	82	3,43 %	5544	0,000	0	0	0,000

Figura 6.14: Captura de la tasa de transmisión para el caso A.

El throughput es el nivel de utilización real del enlace, es decir, la capacidad de información que un elemento de red puede mover en un periodo de tiempo, en la siguiente Figura 6.15, observamos el throughput que viene dado en escala de Bytes vs Segundos, el mismo que es aproximadamente 1100 B/s, cambiando de escala a bits tenemos 8.8 kbps, como se demuestra es muy bajo el throughput ya que la aplicación solo se encarga de enviar y recibir un XML.

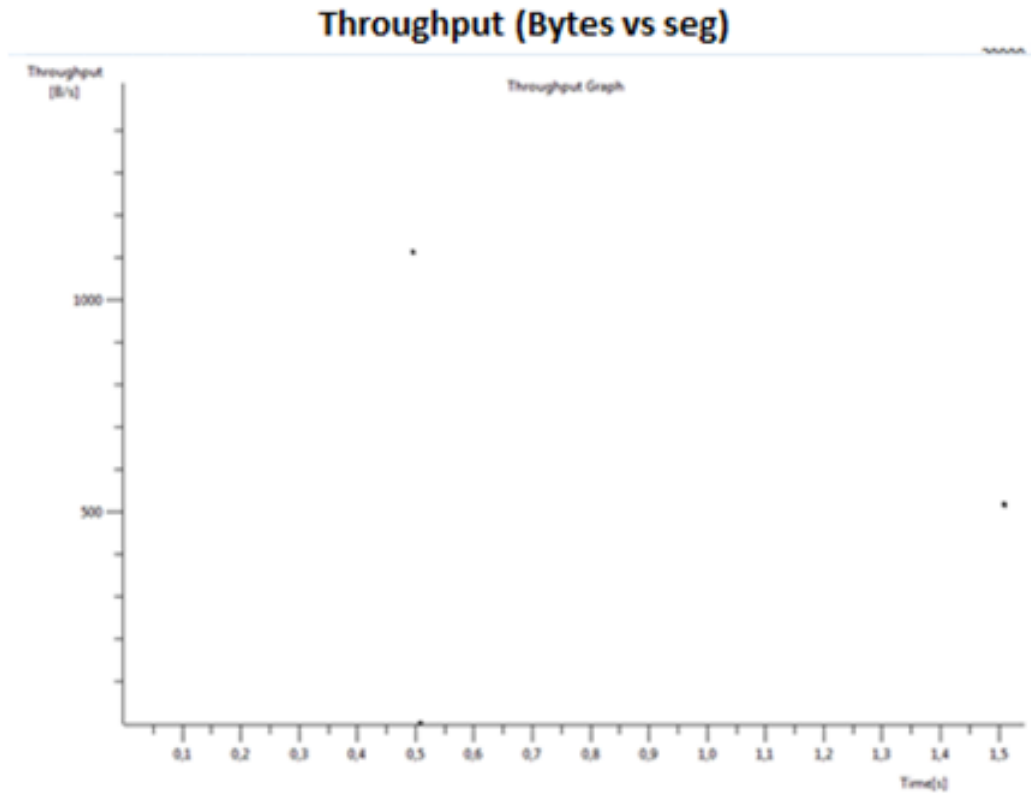


Figura 6.15: Análisis del Throughput Caso A.

En la Figura 6.16, se puede observar el caso B para nuestro análisis, donde el Virtual Set-Top-Box Ginga NCL, tiene como IP 192.168.1.3.



Figura 6.16: Estructura de red para el uso de Canal de Retorno, desde un domicilio ubicado en Quito.

El análisis de los parámetros de los protocolos de red que intervienen al momento de enviar la solicitud y recibir la respuesta del Servidor, al realizar la compra de uno de los

tres productos, es decir, cuando se ejecutó la llamada al Servidor desde Lua, es el mismo que en el caso A, con la única diferencia que cambiaría la IP del Virtual Set-Top-Box a la de 192.168.1.3.

En la Figura 6.17, se observa el tráfico de red que se produjo al realizar las tres compras de la aplicación ShoppingCar, como se observa en la gráfica, existen oscilaciones en la transmisión las mismas que son producidas porque la aplicación está conectada a internet por lo que existen protocolos que siempre se están ejecutando, además, a parte de la aplicación fluyen paquetes de señalización basados en TCP, paquetes ACK, paquetes SYN, etc. Se observa también, que el tiempo que se demora al realizar las tres compras es aproximadamente de 110 segundos y el ancho de banda máximo que ocupa la aplicación es 80kbps. Vale recalcar que la escala con la que se trabaja en las figuras son de bits vs seg.

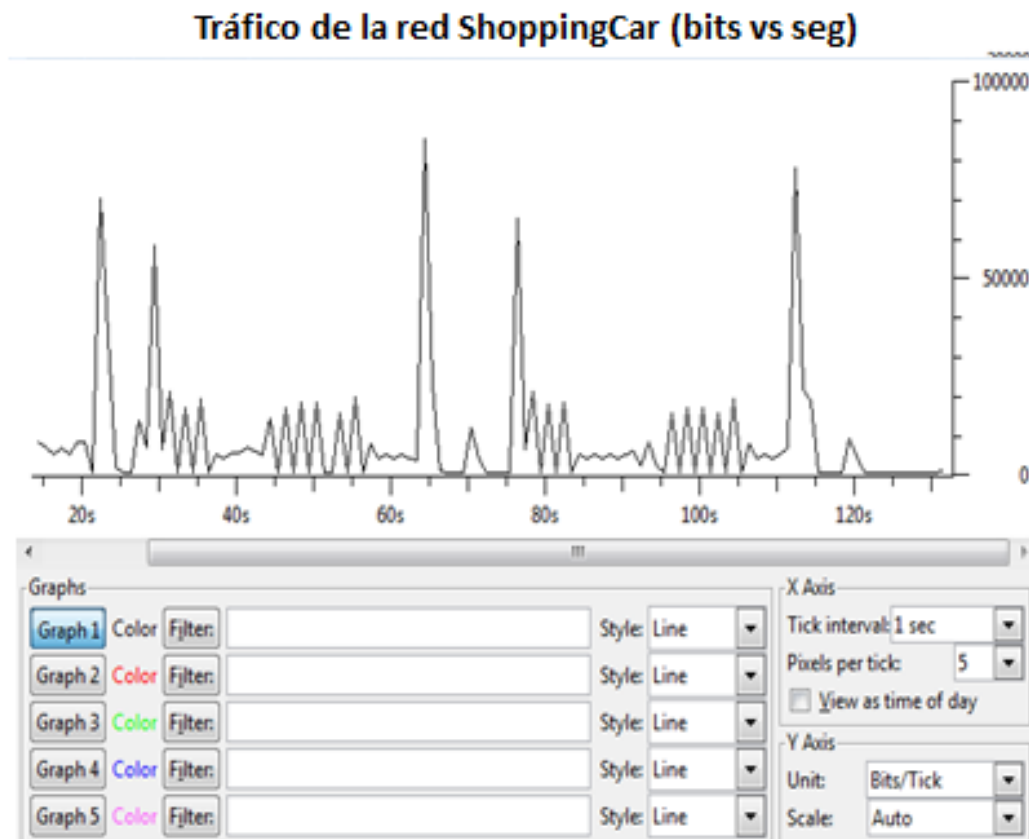


Figura 6.17: Captura del Tráfico de Red con la red 192.168.1.3

En la Figura 6.18, se observa el tráfico del protocolo de HTTP de la aplicación ShoppingCar, el mismo que permite enviar y recibir paquetes, al momento de realizar la compra por ello se obtiene en la gráfica tres oscilaciones, las que simbolizan la compra

de los tres artículos, es decir, cada pico representa el envío de la petición de compra y la recepción de la misma. Además se observa que el ancho de banda es de aproximadamente 20kpbs, ya que se captura sólo el tráfico generado por HTTP, durante la transmisión de los datos. Entre los factores que influyen en el ancho de banda son: las características del computador del servidor y del cliente.

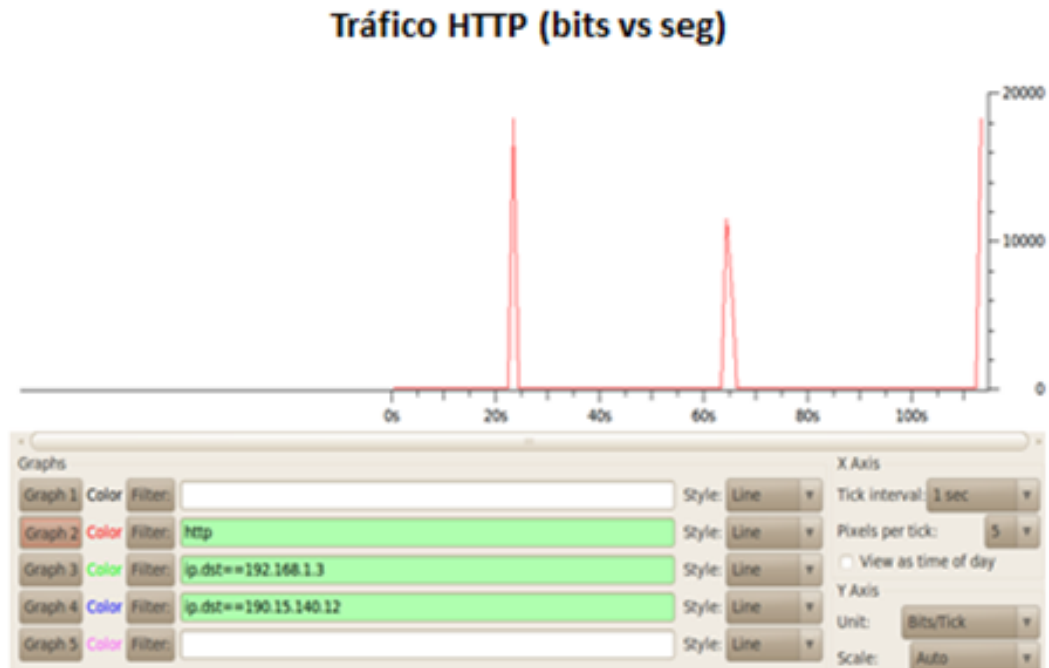


Figura 6.18: Captura del Protocolo http en la corrida de la aplicación.

Con la Figura 6.19, se capturó la transmisión desde el host hasta el servidor, con lo cual se observa varias oscilaciones, que representan los protocolos de internet que se están llamando para la comunicación con el servidor.

Posteriormente, en la Figura 6.20, se observa la transmisión de la respuesta solicitada por el host desde el servidor, por lo que en las gráficas se ven tres oscilaciones las cuales corresponden a las tres compras realizadas.

En la aplicación, cada llamado al servidor representa 12 paquetes enviados como realizamos tres compras equivale a 36 paquetes enviados al servidor, mientras que desde el servidor atender a cada solicitud de compra representa 5 paquetes, en total 15 paquetes, como se muestra en la Figura 6.21.

Como se puede observar en la Figura 6.22, la tasa de transmisión define un número de

Tráfico del Host al Servidor (bits vs seg)

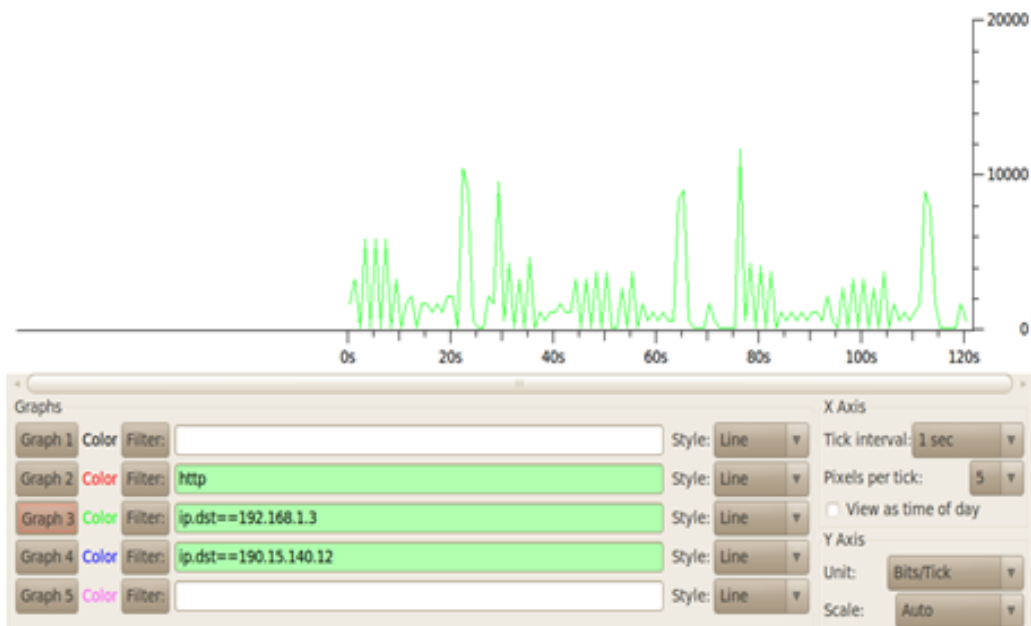


Figura 6.19: Captura de la transmisión de datos desde el host (192.168.1.3) hasta el servidor

Tráfico del Servidor al Host (bits vs seg)

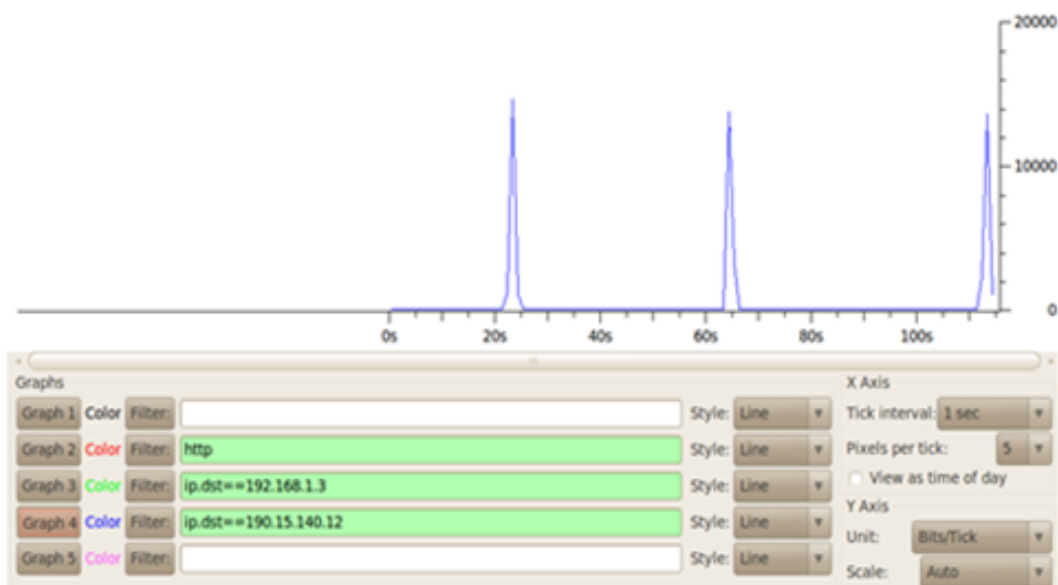


Figura 6.20: Captura de la transmisión de datos desde el servidor hasta el host (192.168.1.3).

bits que se transmite por unidad de tiempo, es decir, es la velocidad de transferencia de

TCP Endpoints: 6									
Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
192.168.1.3	ssh	726	133560	373	110042	353	23518	-	-
192.168.1.2	50172	726	133560	353	23518	373	110042	-	-
190.15.140.12	http	51	9693	15	3363	36	6330	-	-
192.168.1.3	45215	17	3231	12	2110	5	1121	-	-
192.168.1.3	45216	17	3231	12	2110	5	1121	-	-
192.168.1.3	45217	17	3231	12	2110	5	1121	-	-

Figura 6.21: Captura del envío y recepción de paquetes para el caso B

datos que existe en la aplicación ShoppingCar, que es de 9kbps.

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	En
Frame	100,00 %	787	100,00 %	143747	0,009	0	0	
Ethernet	100,00 %	787	100,00 %	143747	0,009	0	0	
Internet Protocol Version 4	99,75 %	785	99,94 %	143663	0,009	0	0	
Transmission Control Protocol	98,73 %	777	99,66 %	143253	0,009	390	25812	
SSH Protocol	48,43 %	378	76,94 %	110592	0,007	378	110592	
Hypertext Transfer Protocol	1,14 %	9	4,76 %	6849	0,000	0	0	
eXtensible Markup Language	1,14 %	9	4,76 %	6849	0,000	3	2547	
Hypertext Transfer Protocol Malformed Packet	0,76 %	6	2,99 %	4302	0,000	0	0	
Internet Group Management Protocol	1,02 %	8	0,29 %	410	0,000	8	410	
Address Resolution Protocol	0,25 %	2	0,06 %	84	0,000	2	84	

Figura 6.22: Captura de la tasa de transmisión para el caso B.

El throughput es el nivel de utilización real del enlace, es decir, la capacidad de información que un elemento de red puede mover en un periodo de tiempo, en la siguiente Figura 6.23, observamos el throughput que viene dado en escala de Bytes vs Segundos, el mismo que es aproximadamente 1100 B/s, cambiando de escala a bits tenemos 8.8 kbps, como se demuestra es muy bajo el throughput ya que la aplicación solo se encarga de enviar y recibir un XML.

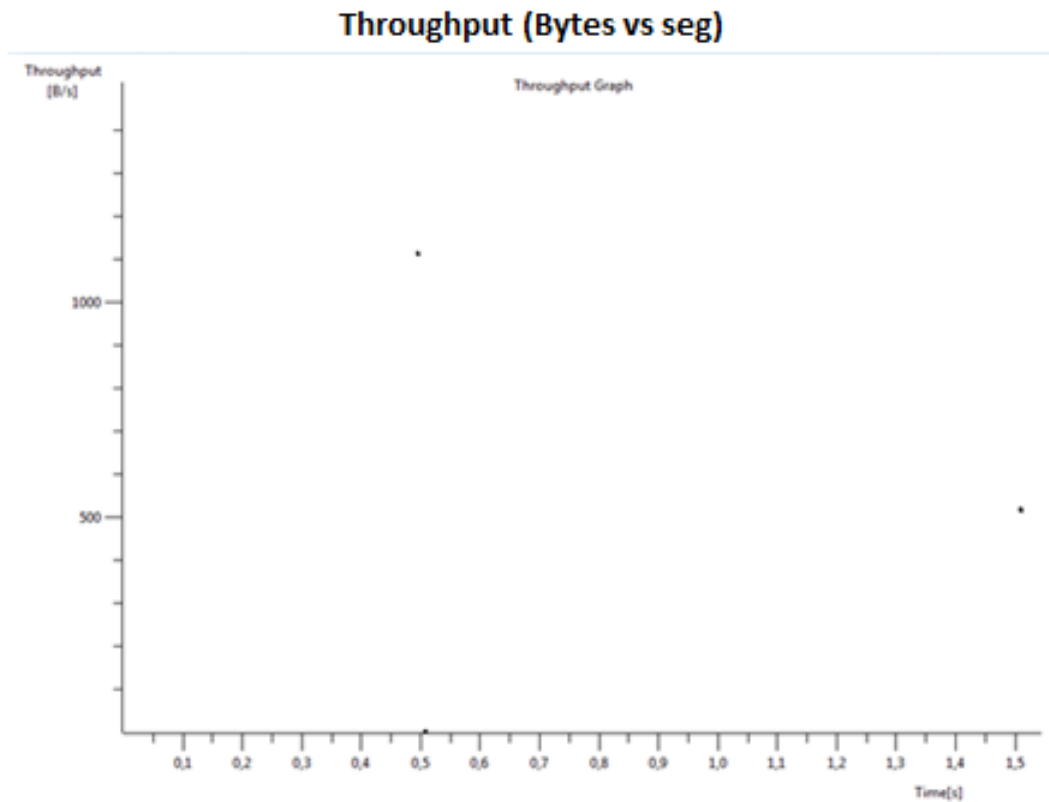


Figura 6.23: Análisis del Throughput Caso B.

Con los resultados obtenidos en el análisis de red para el caso A y para el caso B, se puede determinar el número de usuarios que soportaría el canal de retorno, se formulan varias consideraciones:

- Hay que tener en consideración que el servidor tiene que tener un excelente ancho de banda ya que en éste debe de ser capaz de solventar todas las solicitudes que realicen los clientes, el servidor que se utilizó cuenta con un procesador de 40G en disco, memoria de 1G y presenta un servicio de internet de 3Mbps tanto en uplink como donwlink.
- Hay que tener en cuenta el ancho de banda que posee el usuario, por ejemplo, si un usuario tiene un servicio de internet con 756kbps entonces en esa red pueden tener un máximo de usuarios de 7, es decir en esa red pueden usar 7 usuarios al mismo tiempo.

6.2. Análisis del Rendimiento del Web Service

Para realizar el análisis en el servidor wsCar, se utilizó el software Apache JMeter, el mismo que es una aplicación de código abierto, diseñado en Java, simulando una serie de peticiones las cuales miden la resistencia del servidor, para analizar el rendimiento global del sistema, es decir, permite mandar un número concurrentes de usuarios con el fin de obtener información de cómo se comporta el servidor wsCar bajo esas situaciones.

Las pruebas realizadas al Web Service se efectuaron fuera de la red de la ESPE, ya que nos interesa evaluar el comportamiento del servidor, desde el domicilio del cliente.

Cada segundo fueron enviadas un conjunto de peticiones de: 2500, 5000, 10000, 35000 y 40000 usuarios. A continuación se mostrarán las tablas que contienen los resultados con 10000 y 40000 usuarios.

En la tabla 6.1 se puede apreciar los resultados que se obtuvieron al enviar 10000 peticiones por segundo, donde la Media toma el valor de 1, siendo de tipo aritmética y representando el promedio de la cantidad de peticiones o muestras enviadas. En nuestro caso los elementos en el espacio muestral aumentan aritméticamente, por lo tanto la media aritmética y la mediana son iguales. Además se observa un rendimiento de 291,1 por ciento en cada segundo, al igual se obtuvo un error del 0 por ciento.

Tabla 6.1: Resultados obtenidos con 10000 peticiones

Etiqueta	Muestras	Media	Mediana	Min	Max	%Error	Rendimiento
Pet. Web Ser.	10000	1	1	0	40	0,00 %	291,1/sec
Total	10000	1	1	0	40	0,00 %	291,1/sec

En la tabla 6.2 se puede apreciar los resultados que se obtuvieron al enviar 40000 peticiones por segundo, donde la Media toma el valor de 1, siendo de tipo aritmética y representando el promedio de la cantidad de peticiones o muestras enviadas. En nuestro caso los elementos en el espacio muestral aumentan aritméticamente, por lo tanto la media aritmética y la mediana son iguales. Además se observa que el rendimiento disminuye en relación al aumento de peticiones por segundo, ya que fue de 190,2 por ciento en cada segundo, es decir, el servidor trabajó en su capacidad máxima, lo que puede ocasionar que el servidor ya no responda de forma adecuada a las peticiones solicitada

Tabla 6.2: Resultados obtenidos con 40000 peticiones

Etiqueta	Muestras	Media	Mediana	Min	Max	%Error	Rendimiento
Pet. Web Ser.	40000	1	1	0	43	0,00 %	190,2/sec
Total	40000	1	1	0	43	0,00 %	190,2/sec

CONCLUSIONES Y RECOMENDACIONES

- A través de los resultados obtenidos en las pruebas realizadas para la aplicación ShoppingCar, se obtuvo un tiempo de respuesta inmediato del Servidor wsCar, comprobando que existe eficiencia en la transmisión del canal de retorno, además, el ancho de banda máximo utilizado por la aplicación es de 100kbps, cumpliendo con los rangos establecidos para trabajar con Web Services, logrando alcanzar el objetivo propuesto.
- Como se pudo observar en las gráficas obtenidas del tráfico de red de ShoppingCar, la tasa de transmisión alcanza los 9kbps comparada con el valor del Throughput que se aproxima a 9kbps, se puede concluir que existe un buen desempeño de la red, ya que el valor del Throughput no sobrepasa al valor de la tasa de transmisión.
- La interactividad con canal de retorno, facilita la convergencia entre Internet y la televisión digital, lo que permite que el usuario pueda consumir servicios, diseñados en diferentes lenguajes de programación.
- El consumo de los recursos del Web Service, que es el medio de conexión entre la base de datos y el middleware, se realizó a través del protocolo estándar SOAP, para la invocación de servicios remotos, basado en protocolos estándares de Internet: HTTP para la transmisión y XML para la codificación de datos, garantizando que cualquier cliente con un navegador estándar pueda conectarse al servidor remoto.
- Con las características de infraestructura que cuenta el servidor de aplicaciones, se puede tener un máximo de 40000 usuarios por segundo, por lo cual se recomienda que al salir a producción la aplicación ShoppingCar, se considere mejorar las características de infraestructura del servidor.

Capítulo 7

- Se recomienda que el ancho de banda en downlink, por parte del usuario sea óptimo, para que la respuesta del servidor sea más efectiva y se evitaría oscilaciones innecesarias, que perjudicarían la transmisión.
- La base de datos ShoppingCar creada para la aplicación interactiva, es de tipo dinámica, es decir, la información almacenada puede ser modificada con el tiempo, permitiendo actualizar, añadir datos, y realizar operaciones fundamentales de consulta, mientras que, phpMyAdmin ayuda a proteger los datos evitando su alteración, mediante la gestión de usuarios, el mismo que asigna una contraseña al administrador.
- La televisión interactiva, ofrece diferentes servicios al usuario: administrativos, de asistencia, comercio, educación, entretenimiento, y mejorando con ello su calidad de vida, permitiendo realizar gestiones desde casa, lo que reduce desplazamientos, amplía y extiende la información a la sociedad, cumple funciones sociales, desarrollados en Ginga-NCL que es un entorno de presentación multimedia para aplicaciones declarativas escritas en NCL y su lenguaje de scripting Lua.

Bibliografía

- [1] Superintendencia de Telecomunicaciones Ecuador, “Adopción del estándar para TDT”, <http://www.supertel.gob.ec/tdt-ecuador/>, 2010, 03 de agosto del 2011.
- [2] Gobierno de Chile, Subsecretaría de Telecomunicaciones, “TVD”, http://www.subtel.gob.cl/prontus_vd/site/artic/20070315/pags/20070315163130.html, 03deagostodel2011.
- [3] DVB-T, “DVB-T”, <http://igorfun.com/dvb-t/>,04 de agosto del 2011.
- [4] ATSC, “Standards”, <http://http://www.atsc.org/cms/index.php/standards>,04 de agosto del 2011
- [5] Digital terrestrial television multimedia broadcasting (DTMB), “Introducción”, <http://ebooks.cambridge.org/chapter.jsf?bid=CBO9780511609732&cid=CBO9780511609732A070>,04 de agosto del 2011.
- [6] Standar ISDB-T, “Requerimientos originales de Digitalización”, <http://www.cicomra.org.ar/cicomra2/asp/Present.%20.%20Kawakita%20-%20NEC%20-ISDBT-6-Oct.pdf>,04 de agosto del 2011.
- [7] Informador, “Escoge Ecuador estándar japonés-brasileño para televisión digital”, <http://www.informador.com.mx/tecnologia/2010/188845/6/escoge-ecuador-estandar-japones-brasileno-para-television-digital.htm>,04 de agosto del 2011.
- [8] El ISDB-T, “Ventajas del ISDB-Tb”, <http://www.gingaperu.org/tvd-y-ginga/estandar-isdb-t.html>,04 de agosto del 2011.

- [9] Multimedia Home Platform (MHP), “Sistema Central”, <http://www.televisiondigital.electronicafacil.net/Sections-article21-p1.html>, 21 de agosto del 2011.
- [10] Informe Preliminar: Estado del Arte deReceptores Set-Top-Box Aplicaciones, “Association of Radio Industries and Businesses”, [http://es.scribd.com/doc/70106948/13/ARIB- %E2 %80 %93-Association-of- Radio-Industries-and-Businesses](http://es.scribd.com/doc/70106948/13/ARIB-%E2%80%93-Association-of-Radio-Industries-and-Businesses),21 de agosto del 2011.
- [11] ATSC Standard:Advanced Common Application Platform (ACAP), “ACAP”, <http://www.atsc.org/cms/standards/a101a.pdf>, 22deagostodel2011.
- [12] Globally Executable MHP (GEM) Specification 1.1.1, “GEM”, <http://www.mhp.org/specs/a103r1.tm3567r1.gem1.1.1.pdf>,23 de agosto del 2011.
- [13] Qué es Ginga, “Ginga-NCL (Nested Context Language)”, <http://www.gingaperu.org/tvd-y-ginga/ginga.html>,23 de agosto del 2011.
- [14] Ginga, Enlaces, Código Y Más, “Lenguaje Lua Y El Desarrollo De Aplicaciones Para Televisión Digital”, <http://alejandroayala.solmedia.ec/?tag=tdt-ecuador>,01 de octubre del 2011.
- [15] Manual de Referencia de Lua 5.1, “Convecciones léxicas”, <http://www.lua.org/manual/5.1/es/manual.html>,01 de octubre del 2011.
- [16] Base de datos, “1. Base de datos”, [http://www.monografias.com /trabajos11 /basda/basda.shtml](http://www.monografias.com/trabajos11/basda/basda.shtml),05 de enero del 2012.
- [17] Base de datos, “TIPOS DE BASES DE DATOS”, <http://www.lcc.uma.es/~galvez/ftp/bdst/Tema2.pdf>,05 de enero del 2012.
- [18] API MySQL Original, “Funciones de MySQL”, [http://www.php.net/ manu- al/es/ref.mysql.php](http://www.php.net/manual/es/ref.mysql.php),03 de abril del 2012.

- [19] Manual de PHP, “Conceptos Básicos”, [http://www.php.net/manual /es/index.php](http://www.php.net/manual/es/index.php), 18 de abril del 2012.
- [20] Protocolo simple de acceso a objetos (SOAP), “Protocolo Simple de Acceso a Objetos”, [http://www.ecured.cu/index.php/Protocolo_simple_de_acceso_a_objetos_\(SOAP\)](http://www.ecured.cu/index.php/Protocolo_simple_de_acceso_a_objetos_(SOAP)), 18 de abril del 2012.
- [21] El protocolo HTTP, “Introducción al protocolo HTTP”, <http://es.kioskea.net/contents/internet/http.php3>, 18 de abril del 2012.
- [22] Protocolos de mensajería (SMTP, POP3 e IMAP4), “El protocolo SMTP”, <http://es.kioskea.net/contents/internet/smtp.php3>, 18 de abril del 2012.
- [23] PHP: Acceso a bases de datos, “Acceso a bases de datos MySQL desde PHP”, <http://www.arsys.info/programacion/php-acceso-a-bases-de-datos/>, 19 de abril del 2012.
- [24] Manual de PHP, “Clases”, <http://www.manualdephp.com/manualphp/conexion-mysql.html>, 19 de abril del 2012.
- [25] Introducción al software libre, “Eclipse”, http://www.atenas.cult.cu/rl/informatica/manuales/sl/introduccion_al_SL/eclipse.html, 19 de abril del 2012.
- [26] WEB SERVICES, “¿Qué es un web service?”, [http://www.gxtechnical.com/gxdlsp/pub/genexus/devenv/docum/releasenotes/8.0/.. %5C.. %5C.. %5CInternet %5CTechnicalPapers %5CWeb_Services.htm](http://www.gxtechnical.com/gxdlsp/pub/genexus/devenv/docum/releasenotes/8.0/..%5C..%5C..%5CInternet%5CTechnicalPapers%5CWeb_Services.htm), 19 de abril del 2012.
- [27] UDDI, “¿Cómo funciona UDDI?”, <http://www.desarrolloweb.com/articulos/1589.php>, 22 de abril del 2012.
- [28] Alfon , “Seguridad y Redes”, <http://seguridadyredes.wordpress.com/2008/02/14/analisis-de-red-con-wireshark-interpretando-los-datos/>, 01 de mayo del 2012.

- [29] KIOSKEA.NET , “PROTOCOLO IP”, <http://es.kioskea.net/contents/internet/protip.php3>,
01 de mayo del 2012.
- [30] KIOSKEA.NET , “PROTOCOLO TCP”, <http://es.kioskea.net/contents/internet/tcp.php3>,
01 de mayo del 2012.

FECHA DE ENTREGA

El proyecto fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Escuela Politécnica del Ejército desde:

Sangolquí, _____ del 2012

ELABORADO POR:

Ana Carolina Paredes Teran
180410223-2

Nancy Maricela Tonguino Alcoser
171999919-3

AUTORIDAD

Ing. Edwin Chavez
Coordinador de la Carrera de Ingeniería en Electrónica
Telecomunicaciones