



ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
FACULTAD DE INFORMATICA Y ELECTRONICA
ESCUELA DE INGENIERIA EN ELECTRONICA
TELECOMUNICACIONES Y REDES

**IMPLEMENTACIÓN DE UN SERVIDOR LINUX PARA
ADMINISTRAR CONTENIDOS INTERACTIVOS BASADOS EN
GINGA Y EVALUAR EL DESEMPEÑO EN LA FACULTAD DE
INFORMÁTICA Y ELECTRÓNICA**

**Trabajo de titulación presentado para optar el grado académico de:
INGENIERO EN ELECTRONICA, TELECOMUNICACIONES Y
REDES**

**AUTORES: BYRON JAVIER INCA QUITO
VERÓNICA LORENA SÁNCHEZ SALAO
TUTOR: PhD. JEFFERSON RIVADENEIRA**

Riobamba-Ecuador

2016

ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
FACULTAD DE INFORMATICA Y ELECTRONICA
ESCUELA DE INGENIERIA EN TELECOMUNICACIONES Y REDES

El Tribunal de Trabajo de Titulación certifica que: El trabajo de investigación: IMPLEMENTACIÓN DE UN SERVIDOR LINUX PARA ADMINISTRAR CONTENIDOS INTERACTIVOS BASADOS EN GINGA Y EVALUAR EL DESEMPEÑO EN LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA, de responsabilidad de los señores: BYRON JAVIER INCA QUITO y VERÓNICA LORENA SÁNCHEZ SALAO, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Luna DECANO DE LA FACULTAD DE INFORMATICA Y ELECTRONICA	_____	_____
Ing. Franklin Moreno DIRECTOR DE LA ESCUELA DE ELECTRONICA TELECOMUNICACIONES Y REDES	_____	_____
PhD. Jefferson Ribadeneira DIRECTOR DE TRABAJO DE TITULACION	_____	_____
Ing. Jorge Yuquilema MIEMBRO DEL TRIBUNAL	_____	_____

Nosotros, BYRON JAVIER INCA QUITO y VERÓNICA LORENA SÁNCHEZ SALAO somos responsables de las ideas, doctrinas y resultados expuestos en este trabajo de titulación y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.

BYRON JAVIER INCA QUITO

VERÓNICA LORENA SÁNCHEZ SALAO

DEDICATORIA

A mis padres Rodrigo y Elsa

Por estar siempre apoyándome y quererme, a mi Polet por ser la personita que me brinda el amor más sincero. A Patricio, Tania, Pamela y Micaela Por hacerme feliz, a mis abuelitos y familiares por su cariño incondicional.

Verónica

Al concluir esta investigación que es el producto de un largo esfuerzo y dedicación expreso nuestros sinceros sentimientos de gratitud a mis padres quienes nos permitieron crecer y estudiar para ser profesionales ya que sus buenos ejemplos y sanos concejos han fortalecido mi espíritu, rindo también gratitud y admiración a mis hermanos y demás familiares que son el complemento de la vida, finalmente dedico este trabajo a los distinguidos maestros quienes ocupan un lugar muy especial en el desarrollo profesional.

Byron

AGRADECIMIENTO

En primer lugar quiero agradecer a Dios por darme la vida y estar junto a mí siempre. A mis padres Rodrigo Sánchez y Elsa Salao quienes con su esfuerzo logre una de mis metas que fue obtener el título de ingeniera en Electrónica en Telecomunicaciones y Redes, y además porque ellos me dieron el cariño y amor de una familia, a mi hermana Tania le agradezco el amor que siente por mi hija mismo amor que yo siento por ella, y a mi hermanas Pamela y Micaela por ser mi apoyo durante mi último semestre en la Politécnica. A mis abuelitos por siempre estar conmigo y por darme todo lo que me han dado, en especial a Sarita mi abuelita que espero siempre tenerla porque es la persona más buena q conozco. A mis tías y tíos por darme la fuerza de seguir adelante porque ellos son mi orgullo y un buen ejemplo a seguir. A el amor de mi vida por quereme tanto y darme una preciosa hija mi Polet, y sobre todo gracias por querer formar una familia juntos. A mis amigas por desearme lo mejor y siempre estar ahí cuando estaba desfalleciendo, decirles q son mi alegría y que espero volver a verlas. Finalmente quiero agradecer a todos las personas que de una u otra manera estuvieron conmigo y sé que me quieren como lo son mis primos y primas, sus hijos y parejas, gracias y que Dios les bendiga a todos los mencionados en esta hoja.

Verónica

Primero quiero agradecer a Dios por devolverme la salud y poder continuar con todos mis sueños que tengo para mí futuro. A mi abuela Mariana por transmitirme fuerza de voluntad con todas sus palabras, consejos y por brindarme su apoyo moral contagiándome de su carácter fuerte para poder afrontar mis problemas, a mis padres Fanny Quito y Heriberto Inca por haberme apoyado en todo el camino de mi carrera politécnica dándome ánimos para que continúe estudiando y culmine, a mis hermanos Geovanny, Francisco y Verónica por haberme acompañado todo este tiempo siendo mis amigos, a mis panas de la politécnica de mi generación por haberme acompañado en los momentos difíciles y estar pendientes a cada momento de cómo me va con este trabajo de titulación.

Byron

INDICE DE GENERAL

	Paginas
DEDICATORIA	iv
AGRADECIMIENTO	v
INDICE DE GENERAL	vi
INDICE DE TABLAS	ix
INDICE DE FIGURAS	x
INDICE DE GRAFICOS	xii
RESUMEN	xiii
SUMARY	xiv
INTRODUCCION	1
ANTECEDENTES	2
FORMULACIÓN DEL PROBLEMA	3
Sistematización del problema	3
JUSTIFICACION	4
Justificación teórica -----	4
Justificación aplicativa -----	5
OBJETIVOS	6
Objetivo general -----	6
Objetivos específicos -----	6
MÉTODOS Y TÉCNICAS	7
CAPITULO I	
1. MARCO TEORICO	8
1.1 Evolución de la televisión analógica a la digital -----	8
1.2 Generalidades de la norma ISDB-T -----	9
<i>1.2.1 Estándar ISDB-TB características</i> -----	<i>10</i>
<i>1.2.2 Tipos de comprensión, multiplexación y modulación</i> -----	<i>10</i>
1.3 Televisión Digital en el Ecuador -----	11
<i>1.3.1 Apagón analógico</i> -----	<i>12</i>
1.4 Ventajas y desventajas de la televisión digital -----	13

1.5	Infraestructura de una red de televisión digital-----	15
1.5.1	<i>Infraestructura física-----</i>	15
1.5.2	<i>Infraestructura de Software-----</i>	16
1.6	Servicios Interactivos-----	16
1.6.1	<i>Tipos de servicios interactivos-----</i>	16
1.6.2	<i>Aplicaciones Interactivas-----</i>	18
1.7	Servicios de aplicaciones en televisión digital-----	19
1.8	Introducción a middleware GINGA-NCL -----	20
1.9	Middleware GINGA-----	20
1.10	Eclipse NCL -----	21
1.11	Set top box virtual-----	24
1.11.1	<i>Creación de contenido interactivo en GINGA-NCL. -----</i>	26
1.12	LUA-----	26
1.12.1	<i>Extensiones de NCLua -----</i>	27
1.12.2	<i>Clase TCP LUA-----</i>	27
1.13	Interacción NCL-Lua.-----	29
1.14	Módulos NCLua. -----	29
1.14.1	<i>Módulo event-----</i>	29
1.14.2	<i>Módulo Canvas -----</i>	36
1.14.3	<i>Módulo settings -----</i>	39
1.14.4	<i>Módulo persistent-----</i>	39
1.15	Historia de GNU/LINUX -----	39
1.16	Que es GNU/LINUX y como funciona -----	40
1.17	Distribuciones libres de GNU/LINUX-----	41
1.18	Centos -----	43
1.18.1	<i>Centos minimal -----</i>	43
1.18.2	<i>Centos 6.4 -----</i>	44
1.18.3	<i>Tipo de licencia-----</i>	45
1.19	Distribuciones LINUX para servidores -----	45
1.20	Servidor LAMP-----	46
1.20.1	<i>Servidor web -----</i>	47
1.20.2	<i>Servidor de base de datos -----</i>	47
1.20.3	<i>PHP -----</i>	48
1.21	WIRESHARK -----	48
 CAPITULO II		
2.	MARCO METODOLOGICO.....	49
2.1	Instalación del sistema operativo Centos -----	49
2.2	Configuración de la tarjeta de red-----	53
2.3	Instalación y configuración de LAMP: apache mysql y php -----	54
2.3.1	<i>Instalación del servidor web apache-----</i>	54
2.3.2	<i>Instalación y configuración de mysql-----</i>	54

2.3.3	<i>Instalación y configuración de php</i>	55
2.4	Creación de una base de datos en PHPMYADMIN	58
2.5	Creación de un directorio virtual	60
2.6	Creación de la Página Web	61
2.7	Instalación de Software en el cliente	62
2.7.1	<i>Instalación de Eclipse</i>	62
2.8	Instalación de la middleware ginga ncl set top box virtual	65
2.8.1	<i>Instalación de la máquina virtual</i>	65
2.8.2	<i>Configuración de la máquina virtual de ginga</i>	66
2.9	Requisitos de Hardware y Software	67
2.10	Estructura cliente – servidor	69
 CAPITULO III		
3.	MARCO DE RESULTADOS	71
3.1	Ejecución y funcionamiento del servidor	71
3.1.1	<i>Comprobación de conexión</i>	71
3.1.2	<i>Ejecución de la aplicación en GINGA NCL</i>	73
3.2	Consulta de resultados del servidor de contenido interactivo	75
3.3	Evaluación de desempeño del servidor de aplicaciones y contenidos	76
3.3.1	<i>Evaluación del envío y recepción de la trama</i>	76
3.3.2	<i>Evaluación de Disponibilidad</i>	79
3.3.3	<i>Evaluación del tiempo de Respuesta</i>	80
CONCLUSIONES		87
RECOMENDACIONES		89
BIBLIOGRAFÍA		
ANEXOS		

INDICE DE TABLAS

Tabla 1-1	Tabla de cobertura	12
Tabla 2-1	Tipo de servicios de la Television Digital	17
Tabla 3-1	Elementos, atributos y contenidos de un documento NCL.....	23
Tabla 4-1	Tipos de media en Ginga-NCL	23
Tabla 5-1	Distribuciones de LINUX para servidores	46
Tabla 1-2	Preguntas de instalación de mysql.....	55
Tabla 2-2	Requisitos de Hardware del Cliente.....	67
Tabla 3-2	Requisitos de Hardware del Servidor	68
Tabla 4-2	Requisitos de Software del Cliente.....	68
Tabla 5-2	Software en el servidor	68
Tabla 6-2	Características Cliente - Servidor	70
Tabla 1-3	Características de la trama de envió.....	78
Tabla 2-3	Tiempos de envió del voto desde el cliente al servidor	80
Tabla 3-3	Tiempos de envió del voto desde el servidor al cliente	81
Tabla 4-3	Tiempos de envió del voto total.....	81
Tabla 5-3	Tiempos promedio, máximo y mínimo.....	81
Tabla 6-3	Comparación en el tiempo de respuesta.....	82
Tabla 7-3	Pregunta 1.....	83
Tabla 8-3	Pregunta 2.....	83
Tabla 9-3	Pregunta 3.....	84
Tabla10-3	Pregunta 4.....	84
Tabla11-3	Pregunta 5.....	85
Tabla12-3	Pregunta 6.....	85
Tabla13-3	Pregunta 7.....	86

INDICE DE FIGURAS

Figura 1-1	Transmisión de TDT	8
Figura 2-1	Estándar de TDT en el mundo.....	9
Figura 3-1	Infraestructura básica de una red de televisión Digital	15
Figura 4-1	Servicios Interactivos	17
Figura 5-1	GINGA NCL	20
Figura 6-1	Pantalla del Software de Eclipse	21
Figura 7-1	Elemento región	24
Figura 8-1	GINGA-NCL VSTB	24
Figura 9-1	Botones del control remoto.....	25
Figura 10-1	Programacion orientada a eventos.....	30
Figura 11-1	Logo de software linux.....	40
Figura 12-1	distribuciones de linux	42
Figura 13-1	logo de Centos.....	43
Figura 14-1	Servidor	46
Figura 15-1	Logo wireshark.....	48
Figura 1-2	Página de inicio de instalación Centos	49
Figura 2-2	Página de Configuración 1 VMWare	50
Figura 3-2	Configuración del idioma.....	50
Figura 4-2	Configuración dispositivos.....	50
Figura 5-2	Elección de la zona horaria	51
Figura 6-2	Configuración crear diseño personalizado	51
Figura 7-2	Partición de disco	52
Figura 8-2	Finalización Instalación Centos	52
Figura 9-2	Configuración de la IP del servidor.....	53
Figura 10-2	Dirección IP asignada al servidor.....	53
Figura 11-2	Instalación Mysql	54
Figura 12-2	Instalación PHP	55
Figura 13-2	Configuración de cortafuegos	56
Figura 14-2	Instalación PHPmyAdmin.....	56
Figura 15-2	Ejercitación del archivo .php.....	57
Figura 16-2	Página de Apache en Local host.....	57
Figura 17-2	Pantalla de configuración de phpmyadmin	58
Figura 18-2	Creación de tabla programa.....	59

Figura 19-2	Creación de tabla votos_programaprograma.....	59
Figura 20-2	Configuración de los campos de la tabla programa.....	59
Figura 21-2	Configuración de los campos de la tabla votos_programa.....	60
Figura 22-2	Configuración del directorio virtual	61
Figura 23-2	Código del archivo index.php	62
Figura 24-2	Inicio de eclipse.....	63
Figura 25-2	Descarga de plugins	63
Figura 26-2	Instalación de plugins.....	64
Figura 27-2	Finalización de instalación de Eclipse.....	64
Figura 28-2	Configuración de dirección IP en eclipse.....	65
Figura 29-2	VMWare opción abrir una máquina virtual	65
Figura 30-2	Inicio de la máquina virtual.....	66
Figura 31-2	Ingreso del usuario y contraseña	66
Figura 32-2	Configuración de dirección IP estática.....	67
Figura 33-2	Configuración del host Server Name	67
Figura 1-3	Ping desde la maquina servidor.....	71
Figura 2-3	Ping desde la maquina física al servidor	72
Figura 3-3	Ping del desde la máquina virtual al servidor.....	72
Figura 4-3	Ping desde la máquina física a la maquina virtual	73
Figura 5-3	Aplicación interactiva	73
Figura 6-3	Opción votada SI.....	74
Figura 7-3	Respuesta enviada al cliente.....	74
Figura 8-3	Página web del Servidor.....	75
Figura 9-3	Página de inicio de PHPmyadmin	75
Figura 10-3	Visualización votos en PHPmyadmin	76
Figura 11-3	Funcionamiento del sistema	77
Figura 12-3	Wireshark captura de tramas de envío	77
Figura 13-3	Información Detallada del paquete.....	78
Figura 14-3	Trama TCP desde servidor al usuario	79
Figura 15-3	Trama TCP de respuesta	79
Figura 16-3	Facultad de Informática y Electrónica.....	86

INDICE DE GRAFICOS

Grafico 1.3	Proyección de tiempo de respuesta	82
Grafico 2.3	Pregunta 1	83
Grafico 3.3	Pregunta 2	83
Grafico 4.3	Pregunta 3	84
Grafico 5.3	Pregunta 4	84
Grafico 6.3	Pregunta 5	85
Grafico 7.3	Pregunta 6	85
Grafico 8.3	Pregunta 7	86

RESUMEN

Se implementó un servidor Linux para administrar contenidos interactivos basados en GINGA y se evaluó el desempeño en la Facultad de Informática y Electrónica de la ESPOCH. Se realizó mediante el principio de cliente-servidor con canal de retorno de información, en el servidor de aplicaciones se utilizó el sistema operativo Linux distribución Centos que actuó como servidor LAMP, que incluye Apache como servidor web, MySQL como gestor de base de datos y PHP como lenguaje de programación de webs dinámicas, mediante estos programas se administra todo el contenido interactivo enviado desde el cliente. En el cliente se trabajó con la máquina virtual GINGA NCL que actuó como simulación de un Set Top Box (STB) físico, donde se utilizó la plataforma eclipse para la programación en el lenguaje NCL y LUA, que incluyen las funciones para la conexión con el servidor, se realizó una votación con opción Si o No y se envió los datos hacia el servidor de aplicaciones para que administre el conteo de votos. Los resultados se visualizaron en la página web www.servidoringa.com y en la pantalla del cliente, almacenando esta información en la base de datos, además se evaluó el servidor mediante el programa Wireshark donde se constató que las tramas que se envían y receptan se realizan correctamente en la comunicación. Se concluye que la implementación de este servidor de aplicaciones, que maneja datos de conteo de votos, es de aporte positivo porque no existe en la Facultad de Informática y Electrónica y su culminación permitió soportar la cantidad de infinidad de votos. Se recomienda que con este proyecto implementado los estudiantes desarrollen aplicaciones interactivas nuevas, para futuros avances complementando con el servidor de audio y video se podrá diseñar un canal de televisión digital con la arquitectura completa.

PALABRAS CLAVE: <TECNOLOGIA Y CIENCIAS DE LA INGENIERIA>, <TECNOLOGÍA DE COMUNICACIONES>, <NESTED CONTEXT LANGUAGE (NCL)> <SET TOP BOX (STB)>, <LENGUAJE DE PROGRAMACION (PHP)>, <MIDLEWARE (GINGA)>, <SERVIDOR DE APLICACIONES>, <LENGUAJE DE PROGRAMACION IMPERATIVO (LUA)>

SUMMARY

A Linux server was implemented in order to administer interactive contents base don GINGA and the performance was evaluated in School of Electronics & Computer Science at ESPOCH , it was done through principle of client-server with return channel of information, in the applications server was used the operative system Linux, Centos distribution that acted like LAMP server, it includes Apache like web server, MySQL like databases manager and PHP like programming language of dynamic webs, through these programmes all the interactive content is administered sent from the client. In the client was worked with the virtual machine GINGA NCL that acted like simulation of a Top Box Set (STB) physic, where the eclipse platform was used for the programming in the NCL and LUA language, that include the functions for the connection with the server, a voting with Yes or No option was done and the data were sent to the application server for the administration of vote counting. The results were displayed in the web page www.servidoringa.com and the client screen, storing this information in the data base, besides the server was evaluated through the program Wireshark where the frames that sent and received were confirmed, are done correctly in communication. It is concluded that the implementation of this applications server that handle data of vote counting, represents a positive contribution, because it does not exist in School of Electronics & Computer Science and its conclusión allowed to support the impressive quantity of votes. It is recommended that with the implementation of this Project, the students develop new and interactive applications, for future advances, complementing with the audio and the video server it will design a digital televisión channel with overall structure.

KEY WORDS: <TECHNOLOGY AND SCIENCES ENGINNERING>, <COMMUNICATION TECHNOLOGY>, <NESTED CONTEXT LANGUAGE (NCL)>, <SET TO BOX (STB)>, <PROGRAMMING LANGUAGE (PHP)>, <MIDDLEWARE (GINGA)>, <APPLICATION SERVER>, <IMPERATIVE LANGUAGE PROGRAMMING (LUA) >

INTRODUCCION

El Ecuador como la mayoría de países de Sudamérica, adopto el estándar ISDB-Tb (International System for Digital Broadcast, Terrestrial, Brazilian) para Televisión Digital. Entre las muchas ventajas que ofrece, una de las más importantes para el usuario, es la posibilidad de interactividad.

En este trabajo de titulación se presenta el tema Implementación de un servidor LINUX para administrar contenidos Interactivos basados en GINGA y Evaluar el desempeño en la Facultad de Informática y Electrónica, el cual permite que una aplicación interactiva se comunique con el servidor y el mismo pueda administrar la información que le envía la aplicación interactiva, para almacenar en una base de datos.

La facultad de Informática y Electrónica no cuenta con un servidor de estas características. Por un lado, el disponer de un servidor de contenido interactivo servirá para que los estudiantes puedan desarrollar sus propias aplicaciones y por otro lado constituye un elemento para la futura implementación de un canal de televisión completo en la FIE.

El presente trabajo se encuentra dividido en 3 capítulos, los mismos que se detallan a continuación:

El primer capítulo se presenta la información de los componentes del servidor y los componentes necesarios instalar en el cliente para la comunicación, conceptos sobre el middleware GINGA, Norma ISDB-T y sobre LINUX.

En el segundo capítulo se detalla los pasos para instalar el sistema operativo Centos y los programas como Eclipse, Wireshark, entre otros. Además sobre la instalación y configuración de LAMP, es decir, apache, php, Mysql que son fundamentales en un servidor y necesarias para el servidor de aplicaciones.

En el último capítulo se trata sobre la ejecución de la aplicación interactiva y pruebas de funcionamiento en el servidor para que se lleve a cabo el intercambio de información. Finalmente se hace la evaluación del desempeño del servidor con una análisis de trafico TCP con la ayuda del programa Wireshark, con unas encuestas a los estudiantes para evaluar el tiempo de respuesta del servidor y se analiza la disponibilidad del servidor.

ANTECEDENTES

Linux o GNU/LINUX es un Sistema Operativo como Windows, necesario para que el ordenador permita utilizar programas como: editores de texto, juegos, navegadores de Internet, etc. Puede usarse mediante un interfaz gráfico o mediante línea de comandos.

Linux hace su aparición a principios de la década de los noventa, era el año 1991 y por aquel entonces un estudiante de informática de la Universidad de Helsinki, llamado Linus Torvalds empezó, como una afición y sin poder imaginar a lo que llegaría este proyecto, a programar las primeras líneas de código de este sistema operativo al que llamaría más tarde Linux.

En 1981 IBM VM Machine fue el primer servidor y en 1991 NeXTCube fue el primer servidor WEB, entre otros, son importantes porque día a día en el mundo se crean más tipos de servidores. Se puede montar una instalación para un servidor basado en Linux y ponerlo en línea ya sea para montar un sitio web, una aplicación o cualquier otra necesidad relacionada.

Un servidor en Linux es muy usado en las universidades ya que es un software gratuito y se puede instalar en ordenadores personales, supercomputadores, consolas de última generación e incluso en el móvil. Ofrece información desde archivos de texto, video, audio, imágenes, emails, aplicaciones, programas, consultas a base de datos, etc.

El uso de servidores Linux en Universidades como la UPS de Cuenca y otras que manejan contenido interactivo se va desarrollando día a día, ya se ha realizado transmisiones de aplicaciones interactivas de tv digital con servidores en Ubuntu.

Existen varios servidores dentro de la ESPOCH, por ejemplo servidor web, de nombre de dominio, de correo electrónico y cada servidor tiene diferentes funciones. En la Facultad de Informática y Electrónica no cuenta con un servidor que almacene contenidos interactivos.

La televisión digital se va implementar en el Ecuador debido a que mediante Resolución N° 084-05-CONATEL-2010 el Consejo Nacional de Telecomunicaciones resolvió adoptar el estándar de televisión digital ISDBT Internacional para el Ecuador.

El Ministerio de Telecomunicaciones y de la Sociedad de la Información y entidades gubernamentales del sector de las telecomunicaciones trabajan en coordinación para implementar el sistema de TV digital en el país.

FORMULACIÓN DEL PROBLEMA

¿Cuál es la necesidad de implementar y evaluar un servidor Linux que administre contenido interactivo en la Facultad de Informática y Electrónica?

Sistematización del problema

¿Cuenta la escuela con un servidor Linux que administre contenido interactivo?

¿Qué versión de Linux será factible para que tenga un buen funcionamiento el proyecto?

¿Cómo se realizara la evaluación del desempeño del servidor?

¿Cuáles deberían ser los parámetros de requerimiento de Ginga para que se dé una conexión con el servidor?

JUSTIFICACION

Justificación teórica

Actualmente el diseño para la transmisión de televisión digital terrestre aún no se ha implantado en la Facultad de Informática y Electrónica, por lo cual con esta tesis se pretende estudiar un servidor Linux para administrar datos de la televisión digital interactiva.

Este tema es una réplica porque Ginga no es un tema nuevo, se tomara de referencia varios estudios de Ginga entre ellas el presente trabajo de la tesis de Miguel Dávila Sacoto de la Escuela Politecnica Salesiana con el tema “DISEÑO DE UNA PLATAFORMA DE SOFTWARE PARA TELEVISIÓN DIGITAL INTERACTIVA DE UN CANAL DE DEPORTES UTILIZANDO GINGA-NCL LUA”

Sin embargo el presente trabajo posee una variante porque se podrá almacenar la información que se dé entre la comunicación interactiva del usuario final y el servidor.

El presente trabajo de titulación se enfoca a instalar el servidor Linux y poder administrar los datos de la interactividad siendo solo una parte de la transmisión de la televisión digital.

La gran demanda de los usuarios finales y los estudiantes tienen la curiosidad en lo que se refiere al tema de la televisión digital, crear el servidor que administre contenidos interactivos, será un beneficio para los estudiantes en nuestra facultad porque se conocerá una parte básica de la infraestructura de la red de televisión digital, los estudiantes podrán crear aplicaciones interactivas. En el Ecuador se va a dar el cambio de tv analógica a tv digital y es muy importante que se conozca sobre uno de los equipos necesarios para el cambio a TV digital.

Se evaluará el desempeño del servidor, para saber el tiempo de respuesta que le proporciona al cliente que manda el voto, la disponibilidad del servidor para saber en qué momento falla el sistema.

En el futuro se podrá hacer pruebas de transmisión y recepción de tv digital con los equipos y proyectos que se den en nuestra Facultad.

Justificación aplicativa

Debido a que en la Facultad de Informática y Electrónica no existe un servidor para administrar contenidos interactivos se adquirirá un equipo con un procesador Intel Core i5.

El servidor de contenidos y aplicaciones es importante porque es parte de la infraestructura básica de una red de televisión digital que en complemento con el servidor play out y el set top box se podrá tener una infraestructura completa, para hacer pruebas de funcionamiento de la televisión digital en la FIE.

El servidor se lo realizara con software libre, CentOs versión 6.4. Se realizara en otro software como Ubuntu en caso de ser necesario. Se instalara los paquetes necesarios para la configuración para administrar y dar respuesta a las peticiones del cliente.

Los requisitos necesarios para implementar el servidor Linux son programas en software libre, una plataforma para poder programar eclipse y en el cliente el middleware Ginga para la interactividad con el usuario final.

Se evaluara el desempeño del servidor con una encuesta dirigida a los estudiantes que mediante la aplicación manden su voto al servidor, para hacer un análisis del tiempo de respuesta, y la disponibilidad del servidor ingresando varios votos a diferentes horarios para saber si algún rato deja de funcionar.

OBJETIVOS

Objetivo general

- Implementar un servidor Linux para administrar contenidos interactivos basados en Ginga y evaluar su desempeño en la Facultad de Informática y Electrónica

Objetivos específicos

- Investigar sobre la televisión digital y el estándar ISDB Tb en Ecuador.
- Determinar los requerimientos de software para implementar el servidor y el cliente.
- Implementar el servidor e instalar en el DTIC para hacer pruebas de funcionamiento de T-Voting en la Facultad de Informática y Electrónica.
- Evaluar el desempeño del servidor con el programa Wireshark, analizando el tiempo de respuesta y la disponibilidad.

MÉTODOS Y TÉCNICAS

METODOS

Tipo de investigación científica descriptiva, analítica y documental con el siguiente protocolo de investigación:

- 1.- Analizar los elementos que requerimos de software.
- 2.- Instalación y Configuración del software libre.
- 3.- Realización de la conexión cliente servidor.
- 4.- Configuración de GINGA y creación del contenido Interactivo
- 5.- Verificación de almacenamiento de información en el servidor.

El ciclo de vida adoptado para la ejecución del Proyecto de Tesis es:

- ✓ Recopilación de la información
- ✓ Clasificación de la información
- ✓ Implementación del proyecto
- ✓ Comprobación Descriptiva
- ✓ Evaluación de Resultados
- ✓ Conclusiones

TÉCNICAS

Las técnicas a utilizar en la elaboración de este proyecto investigativo, son las siguientes:

- Observación
- Recopilación
- Estudio
- Encuesta

CAPITULO I

1. MARCO TEORICO

1.1 Evolución de la televisión analógica a la digital

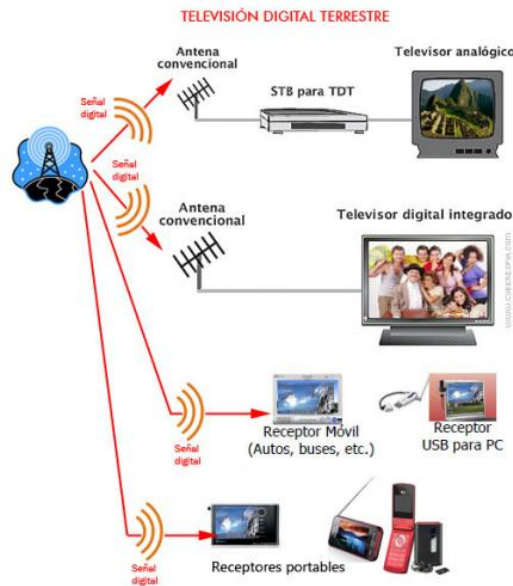


Figura 1.1: Transmisión de TDT

Fuente: <http://repositorio.usfq.edu.ec/bitstream/23000/2136/1/106548.pdf>

En el año de 1924 John Baird transmitió las primeras imágenes televisadas de objetos en movimiento. En el canal su totalidad fue ocupada por la señal de video, es por eso que la primera transmisión donde se tuvo audio y video simultáneamente fue en el año 1930.

Con el pasar de los años los sistemas electromecánicos en la televisión fueron cambiados por los sistemas electrónicos que eran más pequeños en tamaño y de mejor funcionamiento. En 1940, El Ing. González Camarena impacta al mundo al patentar la televisión a color mediante su sistema tricromático secuencial de campos.

Tiempo después emergen otros desarrollos tecnológicos para la televisión a color, con el fin de que los diferentes sistemas fueran compatibles, y que las señales en blanco y negro, fueran también recibidos en las televisiones a color, el inventor ruso Sworykin sugirió la idea de estandarizar los sistemas de Televisión que se estaban desarrollando en el mundo.

La TV digital inicia en 1994 con la introducción de los sistemas de televisión vía satélite DBS (Direct Broadcast Satellite), este hecho provoca el mayor cambio que ha sufrido la TV en los últimos 50 años. La televisión analógica tal y como la conocemos ahora, no había evolucionado en más de 60 años, hasta que aparece la televisión digital y la TV de alta definición.

Hace más de cincuenta años que se transmitió la primera señal de televisión en el Ecuador. En los años posteriores el mercado creció y también la cantidad de canales de televisión tradicional se multiplicaron. Todos estos canales transmiten su señal de manera analógica, pero para el 2020 tendrán que cambiar para poder ser transmitidos mediante televisión digital.

La Televisión Digital en el Ecuador empezara a partir del 2016, en la figura 1.1 se muestra un esquema de la televisión digital terrestre. (Robayo, 2013, p-8)

1.2 Generalidades de la norma ISDB-T

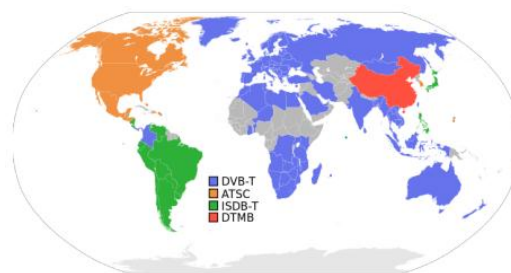


Figura 2.1: Estándar de TDT en el mundo

Fuente: <http://repositorio.usfq.edu.ec/bitstream/23000/2136/1/106548.pdf>

A nivel mundial se tiene cuatro estándares en la figura 2.1 se muestra las distintas normas que adopta cada país para televisión digital terrestre, por lo que cada país debe escoger entre:

1. ATSC norteamericano
2. DVB Europeo
3. ISDB-Tb Japonés-brasilero
4. DMBT-H Chino

En Ecuador el 25 de marzo del 2010 se adoptó la norma ISDB–Tb firmando un acuerdo con Japón y Brasil (“Televisión Digital en el Ecuador a partir del 2016”). El ISBD-Tb es un sistema de radiodifusión de televisión digital, conocido también como ISDB-T Internacional. Éste es una modificación brasileña del estándar japonés ISDB creado por Digital Broadcasting Expert Group (DiBEG).

1.2.1 *Estándar ISDB-TB características*

El estándar ISDBT-Tb tiene las siguientes características, a continuación las más sobresalientes:

- Interactividad en diversos niveles
- Movilidad y portabilidad
- Posee el middleware Ginga
- Soporta formatos en SDTV y HD
- Utiliza formato MPEG-4

1.2.2 *Tipos de comprensión, multiplexación y modulación*

En el sistema ISDB-Tb existen tres tipos de fuentes:

- Audio
- Video
- Datos

Estas fuentes tienen que ser comprimidas antes de ingresar al multiplexor de servicios, para que puedan ser comprimidas y codificadas estas dichas fuentes, ISDB-Tb utiliza el estándar MPEG-4, lo cual permite transmitir imágenes en alta definición, para esto requiere de una elevada tasa de bits de transmisión. (Davila, 2012, pp 8)

Modulación en ISDB en general utiliza 64 QAM-OFDM, 16QAM-OFDM, QPSK-OFDM y DQPSK-OFDM. Para la televisión digital se utiliza PSK y QAM con multiplexación COFDM:

La modulación de amplitud en cuadratura o QAM es una técnica que transporta dos señales independientes, mediante la modulación de una señal portadora, tanto en amplitud como en fase.

La modulación por desplazamiento de fase o PSK (Phase Shift Keying) es una forma de modulación angular que consiste en hacer variar la fase de la portadora entre un número de valores discretos.

COFDM (modulación por división de frecuencia ortogonal codificada) utiliza un gran número de portadoras para transmitir sobre cada una de ellas la información, esto hace la diferencia con los sistemas de modulación comunes que solo utilizan una señal portadora.

1.3 Televisión Digital en el Ecuador

El Ecuador se encuentra en un proceso tecnológico y la televisión comienza una nueva etapa de vida. Dentro de este proceso el MINTEL y de la Sociedad de la información, para cuidar y vigilar por los derechos de los consumidores puso en vigencia, a través del Registro Oficial No. 149, publicado el 23 de diciembre de 2013, el Reglamento Técnico Ecuatoriano “Televisores con sintonizador del estándar de televisión digital ISDB-T Internacional”.

Actualmente, ya existen señales con tecnología digital, estándar de televisión ISDB-T Internacional en el país, que están funcionando gracias al trabajo coordinado entre el Gobierno de la Revolución Ciudadana y la empresa privada.

Para iniciar con el proceso de transición se formó el Comité Interinstitucional Técnico de Implementación de la Televisión Digital Terrestre CITDT.

Al momento, 23 operadores de televisión abierta de los 89 autorizados a nivel nacional ya transmiten señales de Televisión en formato Digital en varias ciudades, tales como: Quito, Guayaquil, Cuenca, Santo Domingo, Manta, Latacunga y Ambato.

A continuación se detallan los operadores que al momento emiten señales digitales en diferentes ciudades del país: En la siguiente tabla 1.1 se muestra la Cobertura de TDT Fecha de actualización: 21 - 08 – 2014:

Tabla 3.1 Tabla de cobertura

No.	ESTACIÓN	CANAL VIRTUAL	AREA SERVIDA
1	ECUADOR TV	7	QUITO
2	GAMA TV	2	
3	TELEAMAZONAS	4	
4	TELESISTEMA	5	
5	ECUAVISA	8	
6	TELEVISIÓN SATELITAL (TVS)	25	
7	TELESUCESOS	29	
8	RTU	46	
9	CANAL UNO	12	
10	ECUADOR TV	7	GUAYAQUIL
11	ECUAVISA	2	
12	RED TELESISTEMA (R.T.S)	4	
13	TELEAMAZONAS GUAYAQUIL	5	
14	TC TELEVISIÓN	10	
15	CANAL UNO	12	
16	TV+ (TEVEMAS)	26	
17	TELEVISIÓN SATELITAL	36	
18	COSTANERA (RTU)	30	
19	ECUADOR TV	7	CUENCA
20	UNIMAX	34	AMBATO-LATACUNGA
21	COLOR TV	36	
22	OROMAR	41	MANTA-PORTOVIEJO
23	TELEATAHUALPA (RTU)	25	SANTO DOMINGO

Fuente: <http://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2014/09/Tabla-de-cobertura-TDT.pdf>

1.3.1 Apagón analógico

El apagón analógico es el nombre con el que se conoce el cese de las emisiones analógicas de los operadores de televisión alrededor del mundo.

En varios países se cambió, en el 2006 se dio en Luxemburgo y países bajos. En el 2007 a Finlandia, Andorra, Suecia y Suiza. En el 2009 a Dinamarca y noruega. En el 2010 a Bélgica,

España, Letonia, Estonia, Eslovenia y Croacia. Y así a varios países en el 2016 se dará en Tailandia, Panamá y Ecuador.

A finales de 2016 en el Ecuador, se prevé inicie el apagón analógico para dar paso a la Televisión Digital Terrestre (TDT), y la población debe prepararse.

Una manera de hacerlo es adquirir los televisores que tienen el sintonizador del estándar ISDBT-Tb o ISDB-T Internacional, que se encuentra disponible en nuestro país.

Para evitar que los ecuatorianos se vean afectados con la adquisición de televisores con otros estándares diferentes al ISDBT-Tb, el 23 de diciembre de 2013, se emitió el Reglamento Técnico RTE 83 para Televisores, fecha a partir de la que todos los televisores, que se importen, fabriquen, ensamblen o comercialicen en el Ecuador deben ser aptos para el estándar ISDB-Tb.

Cuando los ciudadanos deseen adquirir un nuevo televisor, en el Ecuador, pueden verificar si el receptor permite sintonizar o desplegar señales digitales con el estándar ISDB-T Internacional, observando una etiqueta en la parte posterior o frontal del mismo, la que indicará si disponen del sintonizador del estándar ISDB- T Internacional o ISDB-Tb.

Para las personas que quieran conservar su televisor convencional o para aquellos que adquirieron aparatos con tecnología plasma, LED y LCD, que no están acorde con el estándar de televisión digital ecuatoriano, es la adquisición de un decodificador (Set Top Box).

El apagón analógico en Ecuador se realizará de manera progresiva. En el 2016 iniciará en: Quito, Guayaquil, Cuenca. Posteriormente, en el 2017 se realizará el apagón analógico en varias capitales de provincia y, finalmente, en el 2018, todo el territorio nacional tendrá únicamente señales de televisión digital.

1.4 Ventajas y desventajas de la televisión digital

Ventajas

- Interacción: Si el operador brinda todas las posibilidades de interacción, con el control remoto o con el set to box, el televidente podrá realizar varias operaciones:

Consultas sobre la programación que se le está presentando, siempre y cuando el emisor coloque dicha información a su disposición.

Por ejemplo: en un partido de fútbol podrá indagar sobre la marca de los implementos utilizados y las posibles tiendas en donde se puedan adquirir, datos biográficos sobre los jugadores, información sobre la ciudad y el estadio en donde se realiza el juego, historia sobre los dos equipos y el torneo respectivo, etc. La única condición es que el operador incluya toda la información de utilidad para los televidentes.

Participación en programas: en la TV Digital el televidente podrá participar en programas, por ejemplo, concursos, ofertas de empleo, respuestas a preguntas formuladas etc.

- Una mejor definición en la imagen y el sonido Los ruidos e interferencias de la señal se reducirán sustancialmente lo mismo que la presencia de dobles imágenes, usuales en la TV. Analógica.
- Más cantidad, variedad y calidad de los contenidos La TV Digital permitirá más programas, que se escogerán de una guía Electrónica de Programación, con un menú sobre la programación disponible. El hecho de aumentar la oferta de programación, redundará en beneficio de la variedad e innovación y abrirá nuevas opciones de empleo.
- Posibilidades de crear nuevos canales Debido a la optimización en la utilización del espectro electromagnético, existirá la posibilidad de la entrada de nuevos operadores, lo que beneficiará la democratización del servicio.

Desventajas

- Decodificador: es necesario comprar un decodificador de TDT para poder decodificar la señal digital.
- Modificación costosa: el mantenimiento y los cambios que se realizan son más costosos que en la televisión analógica.
- Instalaciones: para poder recibir la señal de TDT correctamente es necesario llevar a cabo una instalación adicional que la analógica.

- Otra de las grandes desventajas será la desaparición de muchos canales locales.

1.5 Infraestructura de una red de televisión digital

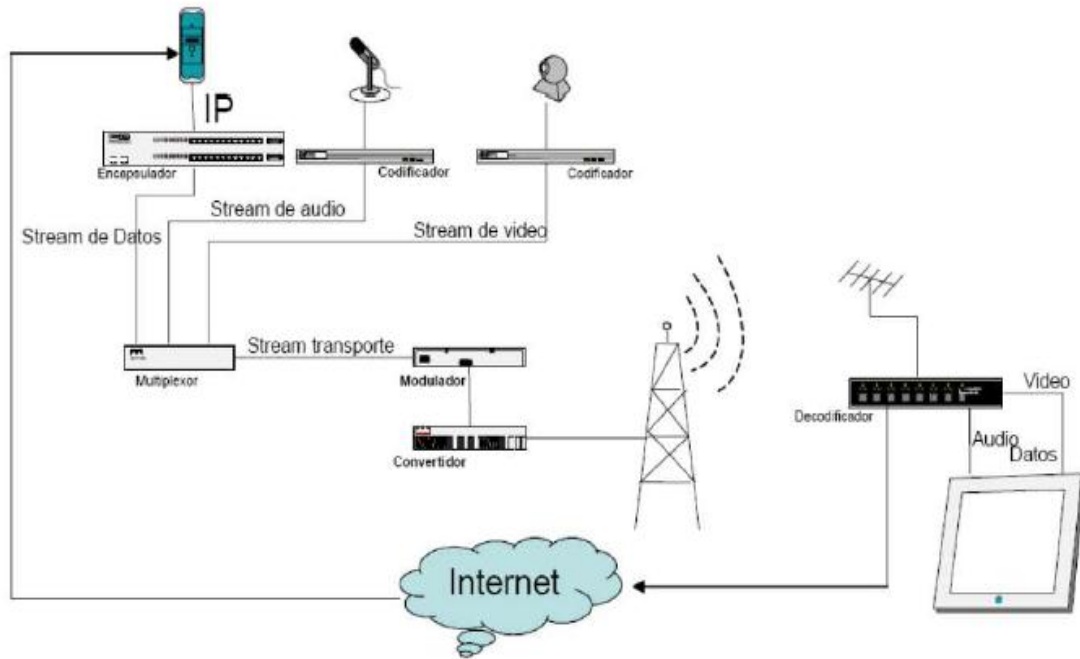


Figura 3.1: Infraestructura básica de una red de televisión Digital

Fuente: <http://dSPACE.UPS.edu.ec/bitstream/123456789/1736/13/UPS-CT02324.pdf>

Los streaming de audio, video y datos se multiplexan en un stream de transporte MPEG-2, el cual se modula a través de una placa física. En la figura 3.1 se muestra un esquema de la infraestructura básica de televisión digital.

1.5.1 Infraestructura física

Tiene como mínimo 3 componentes: el servidor de aplicaciones y contenidos, el servidor de playout y el set top box. En la infraestructura del proveedor están los 2 primeros y en la infraestructura del usuario está el Set To Box (STB). (Sacoto, 2012, pp-10-16)

1.5.2 *Infraestructura de Software*

Junto con la infraestructura física, es necesario el software que se encargara del control, ejecución y administración de los elementos de hardware, así tenemos: *El software para el servidor de aplicaciones y contenidos.*- Se encarga del almacenamiento, difusión y servicio de las aplicaciones NCL.

El software para el servidor de playout.-Este software se encarga de la codificación de los contenidos, la multiplexación y la generación del carrousel de objetos.

1.6 **Servicios Interactivos**

Son los contenidos ofrecidos por los diferentes sistemas que soporta la televisión interactiva. El contenido interactivo es aquel programa que nos da como resultado cuando las aplicaciones interactivas se ejecutan.

1.6.1 *Tipos de servicios interactivos*

Se pueden establecer tres servicios interactivos, en la figura 4.1 se muestran los 3 diferentes servicios interactivos:

- Servicios de información: Son aquellos que ofrecen una información independiente o aparte de la programación audiovisual que se está emitiendo en ese momento.
- Servicios ligados a la programación: Son aquellos que complementan con información suplementaria la programación audiovisual.
- Servicios transaccionales: Son aquellos que ofrecen la posibilidad de enviar y recibir información de forma personalizada y exclusiva.



Figura 4.1: Servicios Interactivos

Fuente: file:///C:/Users/verolore/Downloads/55-717-1-PB.pdf

Los servicios interactivos se complementan con aplicaciones interactivas ya sean de interactividad local o interactividad remota. En la tabla 1.1 se muestra los tipos de interactividad.

Tabla 2.1: Tipo de servicios de la Televisión Digital

TIPOS DE SERVICIO	INTERACTIVIDAD LOCAL	INTERACTIVIDAD REMOTA
Servicios de información	<ul style="list-style-type: none"> • Guía electrónica de Programación • Teletexto mejorado • juegos 	
Servicios ligados a la programación	<ul style="list-style-type: none"> • Estadísticas deportivas • Información ampliada sobre el programa • Noticias desarrolladas 	<ul style="list-style-type: none"> • Participación en concursos • Encuestas • Preguntas y entrevistas a invitados • Pago por visión
Servicios transaccionales		<ul style="list-style-type: none"> • Reserva de plazas • Consultas bancarias • T-administración • Compras

Fuente: http://televisiondigitalterrestredt.com/interactividad_tdt.htm

1.6.2 Aplicaciones Interactivas

Podemos definir las como aquellos programas adicionales a los contenidos de televisión que pueden verse a través de un decodificador interactivo.

El usuario es el que decide si quiere o no ver las aplicaciones interactivas mediante el mando a distancia por el control remoto. Con el fin de comunicar al usuario la posibilidad de acceso a aplicaciones interactivas los operadores o canales de televisión presentan un pequeño menú interactivo que indica al usuario que puede ver una aplicación o un grupo de aplicaciones interactivas.

Interactividad

La interactividad es la capacidad de ofrecer 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, siendo el usuario el que decide si desea o no verlos, y cuándo verlos.

Tipos de Aplicaciones Interactivas

- Servicios de información: Tales como: Estado del tráfico, tiempo, farmacias de guardia, teléfonos de interés, información de aeropuertos, trenes, autobuses, etc.

- Servicios de entretenimiento: Acceso a concursos, contenidos o programas, votaciones, publicidad interactiva, venta por impulso, compra de eventos (fútbol, cine, conciertos, etc.), informaciones sobre concursantes, actores, etc.

1.7 Servicios de aplicaciones en televisión digital

T-Voting

Se utilizan como método de medición de la percepción ciudadana en temas de interés público y se dan los datos necesarios para realizar cuadros estadísticos que ayudan a tomar decisiones contribuyendo en cierta medida a la opinión pública, ejemplo: encuestas.

T-Learning

Permite el aprendizaje interactivo a través del televisor. El rol activo del televidente provoca que los cursos de T-Learning sean nuevas herramientas de contenidos didácticos y recreativos para seguir una enseñanza atractiva y efectiva.

T-Health

Son servicios de asistencia e información médica a través del televisor, como por ejemplo información sobre enfermedades, dietas, solicitar citas en servicios sanitarios, recordatorio de toma de medicamentos, consejos para dejar de tomar.

T-Governmen

Son servicios gubernamentales ofrecidos por el televisor. Evitando el desplazamiento por parte de los telespectadores, hacia las oficinas de administración gubernamental, donde las consultas son realizadas a través del televisor.

Es el servicio de comercio por Televisión, el telespectador a través de este aplicativo tendrá la oportunidad de adquirir productos anunciados directamente por la televisión, sin la necesidad de ir al lugar donde se encuentra la empresa.

1.8 Introducción a middleware GINGA-NCL



Figura 5.1: GINGA NCL

Fuente: <http://trademark.markify.com/trademarks/ctm/GINGA+ncl/008932618>

El GINGA NCL fue desarrollado por la Pontificia Universidad Católica de Rio de Janeiro PUC Rio, provee una infraestructura de presentación para aplicaciones interactivas de tipo declarativas escritas en el lenguaje NCL (Nested Context Lenguaje), en la figura 5.1 se observa el logo de Ginga NCL.

NCL es una aplicación de XML (eXtensible Markup Language) con facilidades para los aspectos de interactividad, sincronismo, espacio temporal entre objetos de mídia, adaptabilidad, soporte a múltiples dispositivos y soporte a la producción de programas interactivos en vivo.

1.9 Middleware GINGA

El Middleware es un conjunto de software ubicado entre el código de las aplicaciones y la infraestructura de ejecución (plataforma de hardware y sistema operativo).

Un middleware para aplicaciones de TV digital consta de máquinas de ejecución de los lenguajes ofrecidos y librerías de funciones, que permiten el desarrollo rápido y fácil de aplicaciones.

GINGA es el nombre del middleware de la Recomendación ITU-T para servicios de IPTV y del Sistema Tipo-Brasileño de TV Digital Terrestre (ISDB-TB). Está formado por un conjunto de tecnologías estandarizadas e innovaciones brasileñas que lo convierten en la especificación de middleware más avanzada.

1.10 Eclipse NCL

Eclipse es un IDE de programación para desarrollar aplicaciones en varios lenguajes (C, java, php entre otros).

Esta plataforma es modular ya que por medio de la adición de plugins se puede extender la funcionalidad, es decir provee la infraestructura para edición de productos a partir de estos. En la figura 6.1 se muestra la pantalla principal de Eclipse Galileo.

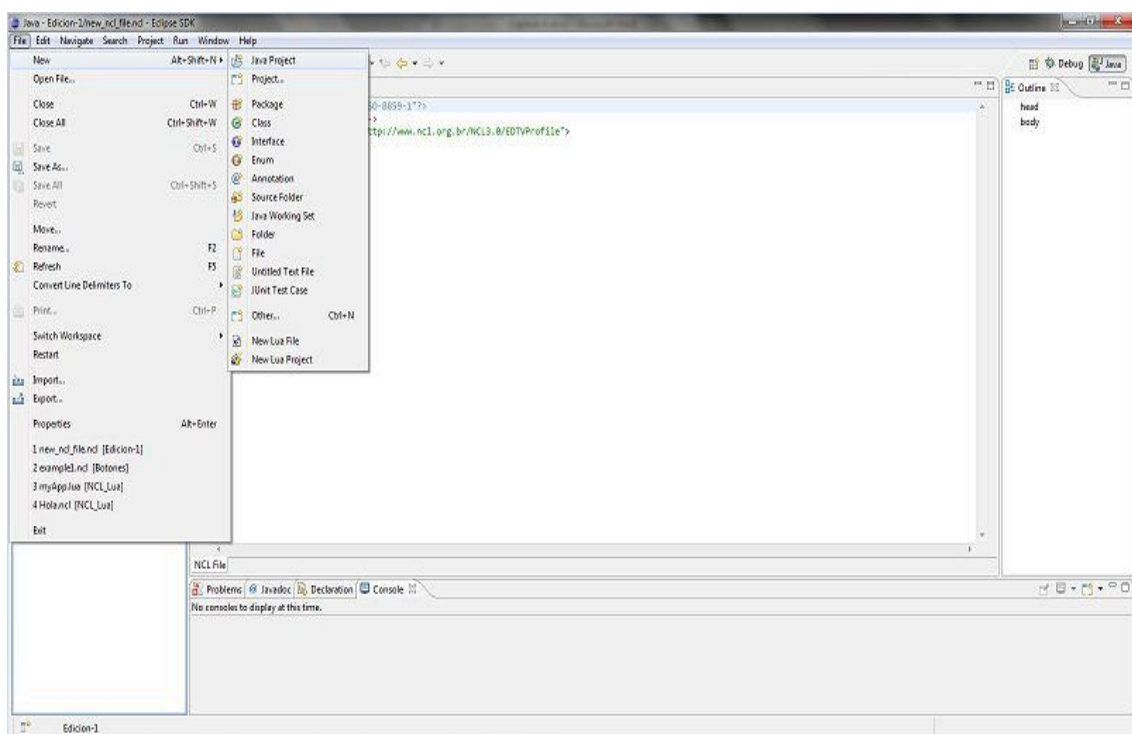


Figura 6.1: Pantalla del Software de Eclipse

Fuente: UPS-CT002411.pdf

La característica fundamental que impulsa la adaptación de ésta a nuevos entornos de desarrollo es que posee una licencia de tipo EPL de código abierto que permite, usar, modificar, copiar y distribuir nuevas versiones del producto.

1.8.1 Documento NCL

El documento NCL consta de 2 partes: Estructura básica: Encabezado y Cuerpo del programa

- Encabezado

Existen 2 encabezados: el archivo NCL y el encabezado del programa

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
```

```
<head>
```

Donde se encuentran: base de regiones, base de descriptores, base de conectores

```
</head>
```

- Cuerpo del programa

Denominada también como body

```
<body>
```

Donde está la puerta de interfaz del programa y el contenido del programa

```
</body>
```

```
</ncl>
```

NCL es un lenguaje desarrollado utilizando una estructura modular basada en web, por lo que permite una alta escalabilidad en aplicaciones. Es a su vez, un lenguaje declarativo, por lo que se especifica el funcionamiento multimedia, permitiendo la interactividad entre objetos de audio, video, y aplicaciones.

NCL es un lenguaje declarativo del middleware ginga, está basado en el modelo conceptual de datos NCM que significa Nested Context Model, este modelo se basa en la representación y

manipulación de documentos hipermedia, representa los datos, eventos y relaciones que existe entre ellos.

En la tabla 2.1 se muestra Los elementos, atributos y contenidos que definen la estructura básica de los documentos NCL.

Elementos	Atributos	Contenidos
ncl	id, title, xmlns	head, body
head	----	importedDocumentBase, ruleBase, transitionBase, regionBase, descriptorBase, descriptorSwitch, connectorBase, meta, metadata
body	----	port, media, area, property, context, switch, switchPort, link

Tabla 3-1: Elementos, atributos y contenidos de un documento NCL

Fuente: https://tvdigitaluach.files.wordpress.com/2012/10/curso_1.pdf

Las entidades básicas representadas en NCL son: Objeto media, Descriptor, Región, link y contexto.

Los objetos media son Imagen, audio, video, texto. En la tabla 3.1 se muestra que tipo de archivos soporta cada objeto media.

Tipo de Media	Tipo de Archivo
Imagen	BMP, PNG, GIF, JPG y JPEG
Audio	WAV, MP3, MP2, MPEG, MPG, MP4 y MPG4
Video	MPEG y MPEG4
Texto	HTM, HTML, TXT, CSS y XML

Tabla 4-1: Tipos de media en Ginga-NCL

Fuente: https://tvdigitaluach.files.wordpress.com/2012/10/curso_1.pdf

Regiones representa un área de un dispositivo en el cual ciertos objetos media serán visualizados, figura 7.1

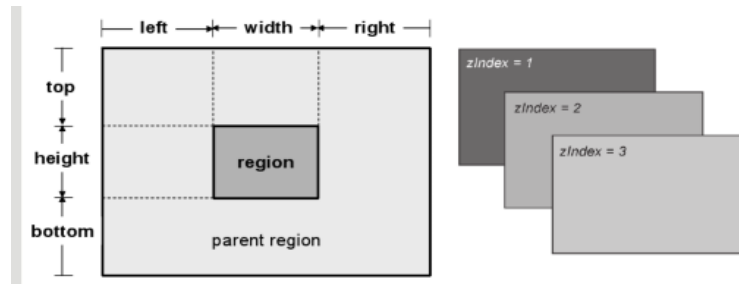


Figura 7.1: Elemento región

Fuente: https://tvdigitaluach.files.wordpress.com/2012/10/curso_1.pdf

Descriptores indica en que región se mostrara el objeto media y con qué propiedades.

Links y conectores especifican acciones sobre objetos media a partir de la ocurrencia de eventos.

Contextos: permite agrupar elementos NCL y reutilizarlos.

1.11 Set top box virtual

El Virtual SetTopBox (VSTB) es una implementación C++ del middleware GINGA emulada en una máquina virtual para VMWare que posee instalada una imagen de Fedora LINUX. El VSTB simula el ambiente de presentación de aplicaciones declarativas, tiene un mejor rendimiento y un entorno más parecido a una aplicación fijado en las STB, que el que es producido por el emulador.



Figura 8.1: GINGA-NCL VSTB

Fuente: Captura de la pantalla del software VSTB.

La máquina virtual de ubuntu-server10.10-GINGA-i386 fue creada y configurada por el personal del Laboratorio de la PUC-Río de software TELEMEDIA utilizando VMWare Workstation 7. En la figura 8.1 se muestra la página de inicio del Virtual Set Top Box.

La instalación ha sido realizada solamente para incluir los paquetes de software esenciales para el desarrollo del middleware GINGA y la ejecución de GINGA-NCL versión C++. Para ejecutar una aplicación sobre el VSTB, se debe abrir una conexión utilizando el protocolo SSH, por medio de un software de acceso remoto. En la figura 9.1 se muestra los botones del control remoto y su interpretación con el teclado.





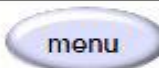






Valores para la propiedad key:	Correspondencia con botones del control remoto:
RED	F1 
GREEN	F2 
YELLOW	F3 
BLUE	F4 
MENU	F5 
INFO	F6 
ENTER	
CURSOR_LEFT	
CURSOR_UP	
CURSOR_RIGHT	
CURSOR_DOWN	

Figura 9.1: Botones del control remoto

Fuente: <http://dspace.ups.edu.ec/handle/123456789/2228>

1.11.1 Creación de contenido interactivo en GINGA-NCL.

Para la creación de contenido interactivo existe el lenguaje NCL, que se emplea para realizar ejemplos de documentos hipermedia. Las funciones del lenguaje son varias y se interpreta en un documento NCL el cual exhibe el programa audiovisual interactivo representado en un receptor de TVD por medio de un VSTB.

1.12 LUA

Desde sus inicios Lua fue diseñado para ser utilizado en conjunto con otros lenguajes, no es muy común que un programa sea escrito puramente en éste.

Este lenguaje permite que una aplicación principal pueda ser ampliada o adaptada a través de scripts. Lua es un lenguaje que combina sintaxis procedural con declarativa, con pocos comandos primitivos.

Por lo tanto, comparado con otros lenguajes, posee implementación ligera y extensible. Otra de las características es que posee un sistema dinámico y un alto grado de portabilidad, pudiendo ser ejecutada en diversas plataformas, tales como computadores personales, celulares, consolas de videojuego, etc. El nombre del lenguaje, Lua, se remite a la idea de un lenguaje satélite. (Galabay y Vivar, 2012, p-37)

Siendo un lenguaje de extensión, Lua no tiene noción del programa principal (main): sólo funciona como una extensión en un cliente anfitrión, denominado programa contenedor o simplemente anfitrión (host) que invoca funciones para ejecutar un segmento de código Lua, puede escribir y leer variables de Lua y puede registrar funciones C para que sean llamadas por el código Lua.

Las características de Lua aparte de su alto rendimiento, bajo consumo de recursos, simplicidad, eficiencia, portabilidad, además de su licencia libre que reduce el costo a cero de la adopción del interpretador por unidad producida, combinan a la perfección con el escenario de la TV Digital.

La portabilidad es importante cuando el middleware sea desarrollado para dispositivos con características contradictorias como la telefonía celular y SetTopBoxes.

1.12.1 Extensiones de NCLua

Lua es el lenguaje de script adoptado por el módulo Ginga-NCL para implementar objetos imperativos en documentos NCL que son puramente declarativos.

Para adecuar al ambiente de la televisión digital y que se integre a NCL, el lenguaje Lua se fue ampliando con nuevas funcionalidades generando así el plugin NCLua. Por ejemplo, este plugin necesita comunicarse con el documento NCL para saber cuándo su objeto <media> correspondiente es iniciado por un enlace.

Un NCLua también puede responder a las claves en el control remoto, y realizar las operaciones dibujo o escritura libremente dentro de la región NCL que le está destinado. Estas características no son específicas del idioma NCL, por lo que no son parte de la biblioteca patrón. Lo que diferencia un NCLua de un programa Lua puro es el hecho de que es controlada por el documento NCL en la cual se inserta, y utilizar las extensiones descritas a continuación.

Además de la biblioteca estándar de Lua, cinco nuevos módulos están disponibles para los scripts NCLua. :

1. Módulo NCLedit: permite que scripts Lua manipulen objetos declarativos de documentos NCL, adicionando, modificando e removiendo informaciones.
2. Módulo event: permite que objetos NCLua puedan comunicarse con el documento NCL y otras entidades externas (tales como control remoto y el canal de interactividad), a través de eventos de una forma asíncrona.
3. Módulo canvas: ofrece elementos (API) para diseñar objetos gráficos en la región de NCLua.
4. Módulo settings: exporta una tabla con variables detenidas por el autor del documento NCL en variables de ambiente reservadas, contenidas en el nodo application/x-ginga-settings.
5. Módulo persistent: exporta un cuadro con las variables persistentes entre ejecuciones de objetos imperativos, estos datos están guardados en un área restringida del middleware.

1.12.2 Clase TCP LUA

Es una librería que permite enviar solicitudes TCP a un servidor remoto, se necesitó la clase TCP LUA para permitir la conexión TCP entre el cliente y el servidor, porque facilito la interacción en las aplicaciones, como de T-Voting y T-learning.

Se utilizó las siguientes funciones:

Tcp.connect

Tcp.send

Tcp.receive

Tcp.disconnect

Tcp.execute

Tcp.handler

```
tcp.execute(  
  function ()
```

```
    writeText("Obteniendo resultado. Por favor, espere... ")
```

```
    tcp.connect(host, 80)
```

```
    --conecta no servidor
```

```
    print("Conectado a " .. host)
```

```
    local url = "GET http://www.servidoringa.com/luca_votos_ginga.php?voto=" .. evt.value .. "\n"
```

```
    print("URL: " .. url)
```

```
    tcp.send(url)
```

```
    local result = tcp.receive("*a")
```

```
    if result then
```

```
        print("Datos de conexión TCP recibidos")
```

```
            f = loadstring(result)
```

```
                if f then
```

```
                    f()
```

```
                    writeResult(votos)
```

```
                setLuaPropertie("result", 1)
```

```
                end
```

```
            else
```

```
                print("Error al recibir datos de conexión TCP")
```

```
                if evt.error ~= nil then
```

```
                    result = 'error: ' .. evt.error
```

```
                end
```

```
            end
```

```
        tcp.disconnect()
```

```
    end
```

```
)
```

1.13 Interacción NCL-Lua.

Un script Lua maneja la misma abstracción para objetos media utilizando por imagenes, videos u otro tipo de medias, algo que apenas lo hace NCL ya que este se refiere al objeto media y no a su contenido.

El lenguaje Lua posee adaptación para funcionar contenido dentro del lenguaje NCL, pero debe ser escrito en un documento con extensión .lua separado del documento mismo, que apenas lo referencia como cualquier otro nodo media.

La principal característica de NCLua está en que su ciclo de vida es controlado por el documento NCL que lo referencia, y es activado en el momento en que el programa Lua es inicializado. La función puede recibir un valor de tiempo como un parámetro opcional que puede ser usado para especificar el momento exacto que el documento debe ser ejecutado.

1.14 Módulos NCLua.

Las bibliotecas disponibles para NCLua están divididas en cuatro módulos esenciales donde cada uno expresa un conjunto de funciones.

1.14.1 Módulo event

El middleware Ginga posee un modelo propio de ejecución y comunicación de objetos imperativos incrustados en documentos NCL. En el caso de objetos NCLua, los mecanismos de integración con un documento NCL se realizan a través del paradigma de la programación orientada a eventos.

El paradigma, es realizado por la difusión y recepción de eventos que se efectúan para comunicarse con el documento NCL y toda la interacción con entidades externas a la aplicación, tales como el canal de interactividad, control remoto y temporizadores. El módulo event de NCLua es utilizado para este propósito y su entendimiento es esencial para desarrollar cualquier aplicación que utilice objetos NClua.

Para comunicarse con un NCLua, una entidad externa debe insertar un evento en la cola, que a continuación es re-direccionado a las funciones tratadoras de eventos, definidas por el

programador de scripts NCLua. Mientras cada tratador procesa un evento (uno a la vez), ningún otro evento de la cola es tratado.

Lua recibe todos los eventos que ocurren en la aplicación NCL, si la media tiene foco. En la figura 10.1 se muestra la conexión por medio de eventos entre NCL y Lua.

Un evento se capta mediante una función manejadora de eventos. El manejador de eventos tiene como parámetro un evento.

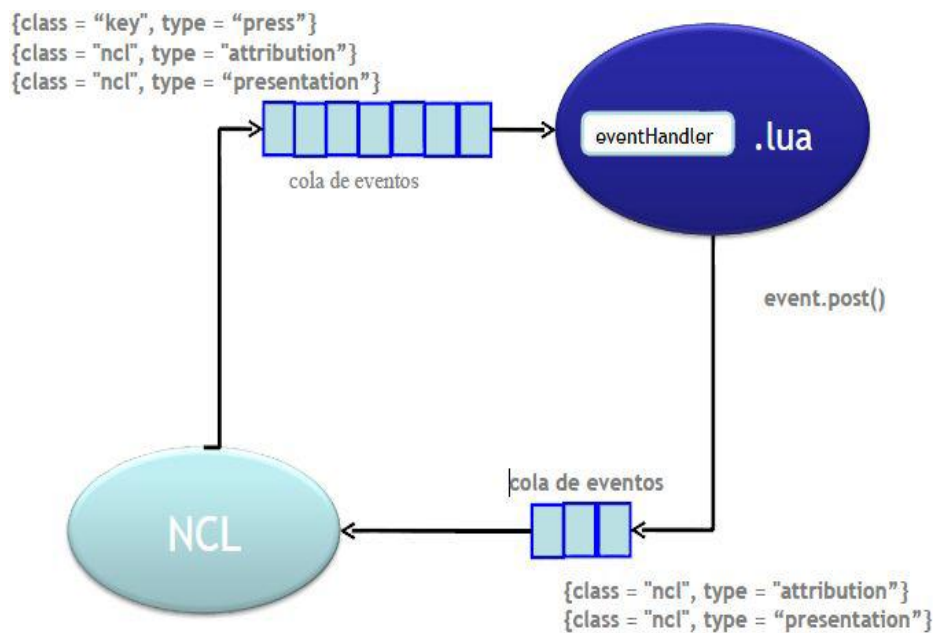


Figura 10.1: Programación Orientada a eventos

Fuente: <http://dspace.ups.edu.ec/handle/123456789/2228>

Ejemplo:

```
function handler(evt)
```

```
...
```

```
end
```

Además debemos indicar que el manejador de eventos reciba todos los eventos de la aplicación.

```
event.register(handler)
```

Para ser informado cuando eventos externos son recibidos, un NCLua debe registrar al menos una función de tratamiento en su cuerpo a través de una llamada a la función `event.register`. (El nombre de la función a ser registrada es cualquiera). El código de un NCLua sigue una estructura común a todos los scripts, tales como los siguientes:

```
... _ código de inicialización.
```

```
function handler (evt)
... _ código para tratar los eventos
End
event.register(handler) _ registro del tratador
```

El código de inicialización, la definición de los tratadores y su registro son ejecutados antes de que el documento NCL (o cualquier entidad externa a script) señale cualquier evento a NCLua, incluyendo el inicio de la presentación del objeto.

Después de que el proceso de carga del script, es efectuado por el sistema, solo el código del tratador es llamado cada vez que ocurre un evento externo.

Un NCLua también puede enviar eventos para comunicarse con la aplicación NCL, y por ejemplo, podría enviar datos por el canal de interactividad o señalar su estado a un documento NCL., para esto utiliza la función `event.post()` mostrado a continuación (figura 13.1).

```
event.post (dst; evt)
```

Como NCLua (a través de sus manejadores) debe correr rápidamente, nunca la función de envío de eventos espera el regreso de un valor. Si el destino necesita devolver una información a un NCLua, debe hacerlo mediante el envío de un nuevo evento. En el siguiente ejemplo, el NCLua señala a un documento su fin natural.

```
event.post {
class='ncl',
type='presentation',
action='stop',
}
```

Los eventos son tablas Lua simples siendo el campo `class` el responsable de identificar la clase de evento y separarlos por categorías. La clase identifica no solamente el origen de eventos pasados a los tratadores, sino también su destino, si el evento es generado y publicado por un script NCLua.

Las siguientes clases de eventos están definidas:

Clase ncl: Utilizada en la comunicación entre un NCLua y el documento NCL que contiene el objeto de medios.

Clase user: A través de esta clase, aplicaciones pueden extender su funcionalidad creando sus propios eventos. Como los eventos de esta clase son para uso interno, no tiene sentido publicar sus eventos con el destino igual a "out".

Clase key: Representa la presión de teclas del control remoto por el usuario. Para esta clase no tiene sentido que el NCLua genere eventos, ya que el control remoto es un dispositivo únicamente de entrada.

Clase tcp: Permite acceso al canal de interactividad por medio del protocolo TCP.

Clase sms: Usada para envío y recibimiento de mensajes SMS en dispositivos móviles.

Clase edit: Permite que los comandos de edición en vivo sean activados a partir de scripts NCLua.

Clase si: Proporciona acceso a un conjunto de informaciones multiplexadas en un flujo de transporte y transmitidas periódicamente por difusión.

Como se puede observar hay eventos sólo de entrada, sólo de salida, y eventos que se utilizan en ambas direcciones.

Clase ncl: Un documento NCLua se comunica con uno en el cual esta insertado a través de esta clase de eventos. En un documento NCL, las relaciones entre los nodos media son descritas a través de enlaces que relacionan condiciones de anclaje.

Un documento NCLua interactúa con el documento únicamente por medio de sus enlaces que están asociados en sus objetos media. Por lo tanto no es posible que un NCLua inter_era directamente en el comportamiento de otras medias presentadas.

Los enlaces que accionan el NCLua, con sus condiciones satisfechas hacen que el NCLua reciba un evento describiendo la acción a ser tomada. Por ejemplo el enlace que aparece a continuación:

```
<link xconnector="onBeginStart">
<bind role="onBegin" component="videoId"/>
<bind role="start" component="luaId"/>
</link>
```

Cuando el video inicia el NCLua recibe el evento:

```
{ class='ncl', type='presentation', action='start' }
```

Este evento será recibido por la función registrada durante la inicialización del script. Para los enlaces cuya condición depende de un NCLua, la acción será desencadenada cuando este señalice el evento que realiza la condición esperada.

Por ejemplo el enlace que aparece a continuación:

```
<link xconnector="onBeginStart">
<bind role="onEnd" component="luaId"/>
<bind role="start" component="imageId"/>
</link>
```

Tan pronto como el NCLua publicar el evento:

```
event.post { class='ncl', type='presentation', action='stop' }
```

El enlace mostrara la imagen que forma parte del enlace. Existen dos tipos de eventos para la clase ncl soportados por NCLua: presentación y atribución. El tipo es identificado en el campo type del evento y podrá asumir, por tanto, solamente los valores 'presentation' o 'attribution'.

Tipo 'presentation': Los eventos de presentación controlan la exhibición del nodo. Además pueden estar asociados con áreas (anclas de presentación) que definen a un nodo como un todo. Las áreas son especificadas por el campo área y equivalen a un nodo completo cuando no son especificadas se asume un valor 'nil'.

El campo action indica la acción a ser realizada o señalizada por NCLua, dependiendo si este está recibiendo o generando un evento. En resumen, un evento de presentación tiene la siguiente estructura:

- class: 'ncl'
- type: 'presentation'
- area: [string] Nombre del ancla (label) asociada al evento.
- action: [string] Puede asumir los siguientes

- valores: 'start', 'stop', 'abort', 'pause' e 'resume'.

Tipo 'attribution': Eventos de atribución controlan las propiedades de NCLua. El campo property del evento contiene el nombre de las propiedades que están afectadas. Los eventos de asignación son bastante similares a la de presentación, una vez que se rigen por el mismo tipo de máquina de estado.

Por lo tanto, la acción start en un evento de atribución corresponde al role=_set_ en un enlace NCL. El campo value se declara como el valor a ser atribuido y siempre es una string, una vez que proviene de un atributo XML. La acción para que se inicie en un evento de asignación corresponde al role= "set" en un enlace NCL.

Las propiedades de NCLua no tienen ninguna relación directa con las variables declaradas en el script. Una NCLua que intenta cambiar el valor de una propiedad debe publicar un evento para este propósito. Las propiedades de los nodos están controladas por el propio documento NCL.

Por ejemplo:

```
event.post {class = 'ncl', type = 'attribution', property = 'myProp', action
= 'start', value = '10', }
```

Un evento de atribución posee la siguiente estructura.

- class: 'ncl'
- type: 'attribution'
- property: [string] Nombre de la propiedad (name) asociada al evento.
- action: [string] Puede asumir los siguiente valores: 'start', 'stop', 'abort', 'pause' e 'resume'.
- value: [string] Nuevo valor a ser atribuido a la propiedad.

Clase 'key':

Representa la presión de teclas del control remoto por el usuario. Para esta clase no tiene sentido que el NCLua genere eventos, ya que el control remoto es un dispositivo únicamente de entrada.

Ejemplo:

```
{class='key', type='press', key='0' }
```

Eventos da clase key posee la siguiente estructura:

- class: 'key'

- type: [string] Puede asumir 'press' o 'release'.
- key: [string] Valor de la tecla en cuestión.

Clase 'user':

Las aplicaciones pueden extender su funcionalidad creando sus propios eventos desde esta clase. Los campos de la tabla que representa el evento se de_ne (además del, el campo clase). Como los eventos de esta clase son para uso interno, no tiene sentido publicar sus eventos con el destino igual a "out". Ejemplo:

```
{ class='user', data='mydata' }
```

Clase 'tcp':

El uso del canal de interactividad es llevada a cabo por medio de esta clase de eventos. Con el fin de enviar y recibir datos, la conexión debe ser preestablecida, registrando un evento como se indica a continuación.

```
{event.post
  class = 'tcp',
  type = 'connect', host = <addr>,
  port = <number>,
  [timeout = <number>,)
}
```

El resultado de la conexión es un tratador de eventos pre registrado. El evento de regreso tiene la siguiente estructura:

```
evt = {
  class = 'tcp',
  type = 'connect',
  host = <addr>,
  port = <number>,
  connection = identi_er,
  error = <err_msg>,
}
```

Los campos error y connection son mutuamente exclusivos. Cuando se trata de un problema de conexión, un mensaje de error es devuelto en el campo error. Cuando la conexión se realiza correctamente, un identificador único para la conexión es devuelto en el campo connection. Una NCLua envía datos a través del canal de retorno publicando eventos de la siguiente manera:

```

{event.post
class = 'tcp',
type = 'data',
connection = <identifier>,
value = <string>,
[timeout = number,]
}

```

De manera similar, un NCLua recibe datos del canal de retorno en eventos de la siguiente forma.

```

evt = {
class = 'tcp',
type = 'data',
value = <string>,
connection = <identifier>,
error = <err_msg>,
}

```

Una vez más, los campos error y connection son mutuamente excluyentes. Cuando se trata de un problema de conexión, un mensaje de error es devuelto en el campo de error. Cuando la conexión se realiza correctamente, un identificador único para la conexión se devuelve en el campo connection. Para cerrar la conexión, el siguiente suceso se debería publicar:

```

{event.post
class = 'tcp',
type = 'disconnect',
connection = <identifier>
}

```

1.14.2 Módulo Canvas

UNA NCLua tiene la posibilidad de realizar las operaciones gráficas durante la presentación de una aplicación, tales como el dibujo de líneas, círculos, imágenes, etc. Esto se realiza por medio de la utilización del módulo canvas que ofrece un API para ser utilizado por NCLua.

Cuando un objeto NCLua es iniciado, la región del elemento <media> correspondiente (del tipo application/x-ginga-NCLua) funciona como variable global canvas para el script Lua. Si la

variable <media> no posee ninguna región especificada (propiedad left, right, top y bottom) este valor de canvas es establecido como `_ nil _`. Si la región está asociado con, por ejemplo:

```
<region id="luaRegion" width="300" height="100" top="200" left="20"/>
```

La variable canvas de NCLua correspondiente estará asociado a la región lua- Region de tamaño 300x100 se localiza en la posición (20,200). Además de los primitivos gráficos que ya se mencionaron, también es posible instanciar nuevos canvas, a través de un constructor `canvas:new(...)`, y mediante esta manera se representan otros objetos gráficos (layers) que después pueden ser compuestos.

La variable canvas de NCLua correspondiente estará asociado a la región lua- Region de tamaño 300x100 se localiza en la posición (20,200).

Además de los primitivos gráficos que ya se mencionaron, también es posible instanciar nuevos canvas, a través de un constructor `canvas:new(...)`, y mediante esta manera se representan otros objetos gráficos (layers) que después pueden ser compuestos.

Un objeto canvas almacena en su estado atributos bajo los cuales las primitivas gráficas funcionan, por ejemplo, si el atributo de color es azul, una llamada al `canvas:drawLine(...)` dibujara una línea azul sobre canvas. Los atributos se accede a través de los métodos de previo `attr` y `su_jo` del nombre del atributo (por ejemplo `attrcolor`), los cuales sirven tanto para la lectura y la escritura (getter y setter).

El primer parámetro de todos los métodos del módulo siempre es `self`, es decir, una referencia al canvas en cuestión. Por lo tanto, se recomienda utilizar el método llamado `Lua` utilizando el operador: (color operator) como en: `myCanvas:drawRect('_ll', 10, 10, 100, 100)`

Las coordenadas pasadas a los métodos son siempre relativas al punto más a la izquierda y a la parte superior de canvas (0,0), como es común entre sistemas gráficos.

Atributos de Canvas:

- `canvas:attrSize()`: retorna las dimensiones del canvas. Ejemplo:
`local w,h = canvas:attrSize()`
- `canvas:attrColor(color)`: función que modifica el color del canvas. Ejemplo:
`canvas:attrColor('white'),`
`mcanvas:attrColor('red'),`

- `canvas:attrFont(fontFamily,fontSize,fontWeight)`: función que modifica atributos de la fuente. Ejemplo:
`canvas:attrFont('vera', 24, 'bold'),`
`canvas:attrFont('dejaVuSerif', 20, 'normal')`

Funciones:

- `canvas:new(image_path)` retorna un nuevo canvas cuyo contenido es la imagen pasada como parámetro. Ejemplo:
`local img = canvas:new('imagen.png')`

Las funciones ya vistas pueden ser aplicadas sobre el nuevo canvas. Ejemplo:

`local w,h = img:attrSize()`

- `canvas:compose(x,y,src)`: función que compone el canvas principal con el canvas especificado en `src` en la posición `x,y`. Ejemplo:
`canvas: compose(0,0,canvas:new('imagen.png'))`
- `canvas:drawLine(x1,y1,x2,y2)`: función que dibuja una línea con sus extremos en `(x1,y1)` y `(x2,y2)` . Ejemplo:
`canvas:drawLine(10, 10, 100, 100)`
- `canvas:drawRect(mode,x,y,width,height)`: función que dibuja un rectángulo. El parámetro `mode` puede tomar los valores 'frame' o 'fill'. Ejemplo:
`canvas:drawRect('_ll', 10, 10, 100, 100)`
`canvas:drawRect('frame', 50, 50, 200, 200)`
- `canvas:drawText(x,y,'texto')`: función que dibuja un 'texto' en la posición `x,y` . Ejemplo:
`canvas:drawText(10, 10, 'Hola Mundo!')`
`canvas:drawText(5, 40, 'Ejemplo Lua')`
- `canvas:flush()` función para actualizar la superficie del canvas. Ejemplo:
`canvas:flush()`

1.14.3 Módulo settings

Las propiedades de un nodo settings solo pueden ser modificadas por medio de los enlaces de NCL. No es posible atribuir valores a campos que representan variables en los nodos settings.

La tabla settings divide sus grupos en varias subtablas correspondiente a cada grupo del nodo application/x-ginga-settings. Por ejemplo en un objeto NCLua la variable del nodo settings “system.CPU” es referida como settings.system.CPU.

```
lang = settings.system.language
age = settings.user.age
val = settings.default.selBorderColor settings.user.
age = 18 _> ERRO
```

1.14.4 Módulo persistent

Aplicaciones NCLua permiten que datos sean guardados en una zona de acceso restringida del middleware y recuperados entre ejecución. Al exhibir Lua define un área reservada, inaccesible para objetos NCL no procedurales. No existe ninguna variable predefinida o reservada a estos objetos procedurales pudiendo atribuir valores a estas variables directamente. El uso de tablas persistentes es semejante a la utilización de tablas settings, excepto por el caso de que el código procedural pueda mudar los valores de los campos.

```
persistent.service.total = 10
color = persistent.shared.color
```

1.15 Historia de GNU/LINUX

GNU es un acrónimo este proyecto fue iniciado por Richard Stallman y anunciado el 27 de septiembre de 1983, con el objetivo de crear un sistema operativo que sea completamente libre.

En 1991 Linus Torvalds, siendo entonces un estudiante de la Universidad de Helsinki, Finlandia fue el creador del kernel de LINUX. En 1992, el núcleo LINUX fue combinado con el sistema GNU.

El Sistema Operativo formado por esta combinación se conoce como GNU/LINUX que es equipamiento lógico libre o Software Libre. Esto significa que el usuario tiene la libertad de redistribuir y modificar a de acuerdo a sus necesidades, siempre que se incluya el código fuente, como lo indica la Licencia Publica General GNU, que es el modo que ha dispuesto la Fundación de equipamiento lógico libre.

Esto también incluye el derecho a poder instalar el núcleo de GNU/LINUX en cualquier número de ordenadores o equipos de cómputo que el usuario desee o necesite.

1.16 Que es GNU/LINUX y como funciona



Figura 11.1: Logo Software LINUX

Fuente: <http://www.LINUX-es.org/distribuciones>

GNU/LINUX es un poderoso y versátil sistema operativo con licencia libre y que implementa el estándar POSIX (acrónimo de Portable Operating System Interface), que se traduce como Interfaz de Sistema Operativo Portable. En la figura 11.1 se muestra el logo de Linux

GNU/LINUX no es equipamiento lógico gratuito (comúnmente denominado como Freeware), se trata de equipamiento lógico libre. Cuando nos referimos a libre, lo hacemos en relación a la libertad y no al precio.

La GPL (General Public Licence), que se traduce como Licencia Publica General, a la cual Linus Torvalds incorporo a LINUX, está diseñada para asegurar que el usuario tenga siempre la libertad de distribuir copias del equipamiento lógico y cobrar por el servicio si así lo desea.

GNU/LINUX es también de la mejor alternativa de siglo XXI para los usuarios que no solo desean libertad, sino que también desean un sistema operativo estable, robusto y confiable. (Lozada, 2010, p.15)

Es un sistema operativo ideal para utilizar en Redes, como es el caso de servidores, estaciones de trabajo y también para computadoras personales. Las características de GNU/LINUX le permiten desempeñar múltiples tareas en forma simultánea de forma segura y confiable.

Tal ha sido el impacto, que muchas de las empresas de Software más importantes del mundo entre las cuales están IBM, Oracle y Sun Microsystems han encontrado en GNU/LINUX una plataforma con un muy amplio mercado y se han afanado al desarrollo de versiones para LINUX de sus más importantes aplicaciones.

Grandes corporaciones como COMPAQ, Dell, Hewlett Packard, IBM y muchos más, llevan varios años distribuyendo equipos con GNU/LINUX como sistema operativo.

LINUX ha dejado de ser un sistema operativo poco atractivo y complicado de utilizar para convertirse en una alternativa real para quienes buscan un sistema operativo confiable y poderoso, ya sea para un servidor, estación de trabajo o la computadora personal de un usuario valeroso.

1.17 Distribuciones libres de GNU/LINUX

Una distribución es una recopilación de programas y ficheros, organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet, o comprando los CDs de las mismas, los cuales almacenan todo lo necesario para instalar en un sistema LINUX.

Casi todos los principales distribuidores de LINUX, ofrecen la posibilidad de descargarse sus distribuciones, vía FTP. En la figura 12.1 se muestra todas las distribuciones a lo largo de los últimos años.

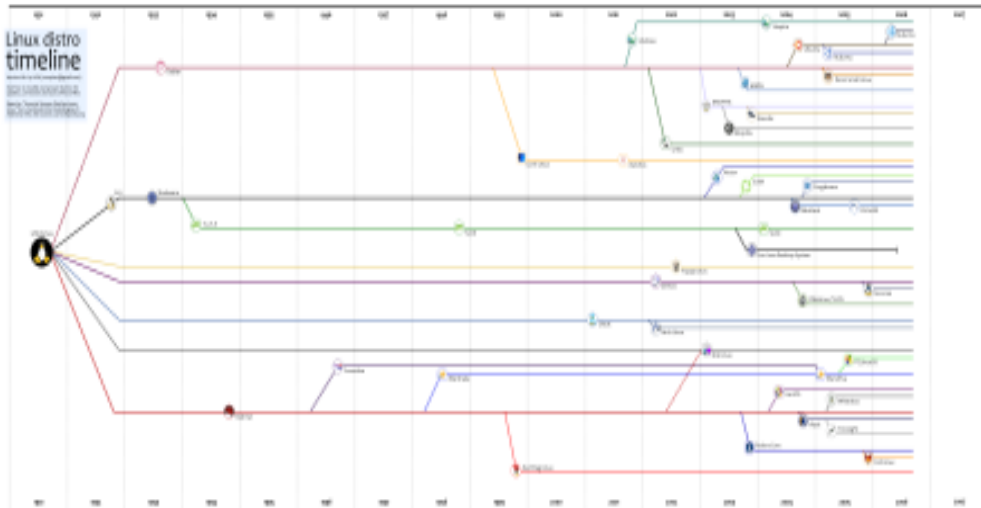


Figura 12.1: Distribuciones de LINUX

Fuente: <http://www.LINUX-es.org/distribuciones>

A continuación las distribuciones más importantes de LINUX:

UBUNTU: Distribución basada en Debian, centrada en el usuario final y facilidad de uso. Muy popular y con mucho soporte en la comunidad. El entorno de escritorio por defecto es GNOME.

REDHAT ENTERPRISE: Esta es una distribución que tiene muy buena calidad, contenidos y soporte a los usuarios por parte de la empresa que la distribuye. Es necesario el pago de una licencia de soporte, enfocada a empresas.

FEDORA: Esta es una distribución patrocinada por RedHat y soportada por la comunidad. Fácil de instalar y buena calidad.

DEBIAN: El proceso de instalación es quizás un poco más complicado, pero sin mayores problemas. Gran estabilidad antes que últimos avances.

OpenSuSE: Fácil de instalar. Versión libre de la distribución comercial SuSE.

MANDRIVA: Facilidad de uso para todos los usuarios. Entre otras distribuciones de LINUX las cuales se las pueden descargar de la página oficial de Centos o desde el repositorio de Mirror.

1.18 Centos



Figura 13.1: Logo de Centos

Fuente: <http://www.LINUX-es.org/distribuciones>

Abreviatura de Community Enterprise Operating System Es una distribución de LINUX gratuita que está basada en la distribución Red Hat Enterprise LINUX (RHEL), gratuito, aunque no es mantenido por Red Hat. Red Hat Enterprise LINUX se compone de software libre y código abierto, pero se publica en formato binario usable (CD-ROM o DVD-ROM) solamente a suscriptores pagados.

Como es requerido, Red Hat libera todo el código fuente del producto de forma pública bajo los términos de la Licencia pública general de GNU y otras licencias. Centos se encuentra como líder en las estadísticas de distribuciones GNU/LINUX en la parte de Servidores. En la figura 13.1 se muestra el logo de Centos.

1.18.1 Centos minimal

Es sin el entorno gráfico instalado. Esto causa que solo se ve una terminal de texto plano. Si se requiere instalar un escritorio gráfico en un Centos 6 Minimal seria de instalar los paquetes necesarios.

Centos es compilada a partir de las fuentes que publica Red Hat y es totalmente compatible con los paquetes diseñados para correr sobre la plataforma de Red Hat.

Al ser Centos compatible con REL (Red Hat Enterprise LINUX), provoca que sea una opción muy atractiva al momento de realizar despliegues de soluciones LINUX dado que es muy extensa la documentación y procedimientos que se encuentran libremente en internet para poder implementar cualquier solución.

El mayor inconveniente que tiene esta distribución es el tamaño de sus medios de instalación; son dos imágenes de DVD y entre las dos suman más de 4GB.

Si uno no ha sido precavido y ha pre-descargado dichas imágenes del internet podrá pasar un tiempo considerable antes de poder iniciar cualquier proceso de instalación si no se tienen dichos medios.

Por tal motivo el grupo de Centos ha encontrado que esta es una situación que sucede casualmente a las personas y han preparado un medio de instalación llamado Centos Minimal, el cual permite instalar un sistema LINUX base, el cual puede fácilmente instalar paquetes adicionales desde el internet utilizando el gestor de paquetes YUM.

La instalación de Centos minimal es realmente mínima; la imagen del disco de instalación tan solo ocupa 300MB, pero de la misma forma deja por fuera muchas funcionalidades que cualquier administrador puede considerar como básicas y esenciales.

Por eso lo primero que hay que hacer una vez de instales un equipo haciendo uso del este medio de instalación es ejecutar los comandos: Yum update, Yum groupinstall base.

Esto instalara aquellas herramientas básicas necesarias para la operación de un servidor en el día a día (tales como perl, crontab, postfis, wget, entre otras).

1.18.2 Centos 6.4

Las distribuciones para servidores no tienen grandes cambios, sin embargo, en Centos 6.4 se incluye las últimas versiones de samba que permite gestionar dominios con Directorios Activos, mejoras en los drivers para Microsoft Hyper-V para que Centos sea más eficiente cuando es instalado en estos servidores.

CentOS usa yum como paquete de gestión de las actualizaciones, herramienta también utilizada por la distribución Fedora.

Hardware recomendado para operar:

Sin entorno de escritorio:

- Memoria RAM: 64MB (mínimo).
- Espacio en Disco Duro: 1 GB (mínimo) - 2 GB (recomendado).

Con entorno de escritorio:

- Memoria RAM: 2 GB (mínimo)
- Espacio en Disco Duro: 20 GB (mínimo) - 40 GB (recomendado).

1.18.3 Tipo de licencia

Centos tiene la Licencia Pública General GNU es la licencia que acompaña los paquetes distribuidos por el Proyecto GNU, más una gran variedad de software que incluye el núcleo del sistema operativo LINUX.

La formulación de GPL es tal que en vez de limitar la distribución del software que protege, llega hasta impedir que este software sea integrado en software propietario.

La GPL se basa en la legislación internacional de copyright, lo que debe garantizar cobertura legal para el software licenciado con GPL.

1.19 Distribuciones LINUX para servidores

Debian: Fue iniciada por Ian Murdock. Es la única entre las variedades aquí mostradas en la que no hay ninguna compañía detrás, la distribución está controlada totalmente por voluntarios vinculados por el contrato social Debian. Un líder es elegido cada año desde y por los miembros del proyecto Debian. (Homero, 2010, p-1)

Ubuntu: La distribución que más auge ha tenido en los últimos años, con cada vez más usuarios y que más rápido se ha adaptado a las necesidades de los mismos. Ubuntu es una distribución basada en Debian.

Red Hat Enterprise: Red Hat Enterprise LINUX (RHEL) Es la distribución de LINUX más conocida y muy popular en cuanto a servidores, además de ser uno de los más antiguos.

Centos: Es una versión libre disposición de Red Hat Enterprise que no cobra por el acceso a actualizaciones de seguridad. Esto es posible debido a la licencia libre bajo la cual se libera Red Hat.

Dado que es una copia casi exacta de RHEL, con únicamente los logotipos y marcas registradas modificadas, los binarios son 100% compatibles, es decir, las aplicaciones diseñadas para aplicaciones comerciales de Red Hat se ejecutarán sin modificaciones y con total compatibilidad.

SuSE Enterprise: LINUX Enterprise Server es una distribución que toma prestado de Red Hat su gestión de paquetes, distribución y su modelo de negocio. Fue creada originalmente en Alemania por un grupo de consultores de UNIX, SuSE significa “Software- und System-Entwicklung” (Desarrollo de software y de sistemas).

En la tabla 4.1 se muestra las distribuciones de Linux para servidores, con su año de inicio.

Tabla 5.1: Distribuciones de LINUX para servidores

Distribución	Año de inicio	Basada en	Formato de paquetes	¿Apoyo del distribuidor?
Debian	1993	-	DEB	no
Ubuntu	2004	Debian	DEB	yes
Red Hat Enterprise	1994	-	RPM	yes
CentOS	2004	Red Hat Enterprise	RPM	no
SuSE	1994	Slackware/RedHat	RPM	yes

Linux Hispano - [Linuxhispano.net](http://linuxhispano.net)

Fuente: <http://www.LINUXhispano.net/2010/02/16/distribuciones-LINUX-para-servidores/>

1.20 Servidor LAMP



Figura 14.1: Servidor

Fuente: <http://aprenderaprogramar.com/>

Un servidor es un ordenador o máquina informática que está al “servicio” de otras máquinas, ordenadores o personas llamadas clientes, el servidor suministra a estos todo tipo de información. En la figura 14.1 se muestra la imagen de servidor.

Se trata de una combinación muy popular en los servidores de LINUX, que va con su acrónimo: LINUX, Apache, MySQL y PHP.

- LINUX es el sistema operativo sobre el que trabaja.
- Apache Es el servidor web.
- MySQL Es un gestor de bases de datos.
- PHP es un lenguaje de programación de webs dinámicas.

Se denomina "LAMP" a un grupo de software de código libre que se instala normalmente en conjunto para habilitar un servidor para alojar sitios y aplicaciones web dinámicas.

1.20.1 Servidor web

Almacena principalmente documentos HTML, que son documentos a modo de archivos con un formato especial para la visualización de páginas web en los navegadores de los clientes, imágenes, videos, texto, presentaciones, y todo tipo de información que se requiera. Además se encarga de enviar estas informaciones a los clientes.

APACHE: Es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores tengan su confianza en este programa.

1.20.2 Servidor de base de datos

Da servicios de almacenamiento y gestión de bases de datos a sus clientes. Una base de datos es un sistema que nos permite almacenar grandes cantidades de información.

My SQL: MySQL es el servidor de bases de datos relacionales más acogido en el mundo, desarrollado y proporcionado por MySQL AB que es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL. Es un sistema de administración de bases de datos.

Una base de datos: Es una colección estructurada de datos. La información que puede almacenar una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceso, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como MySQL.

1.20.3 PHP

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

Uso de scripts en php: Scripts del lado del servidor. Este es el campo más tradicional y el foco principal.

Es necesario ejecutar el servidor con una instalación de PHP. Se puede acceder al resultado del programa de PHP con un navegador, viendo la página de PHP a través del servidor.

1.21 WIRESHARK



Figura 15.1: Logo Wireshark

Fuente: <http://www.welivesecurity.com/>

Wireshark es una herramienta multiplataforma utilizada para realizar análisis sobre paquetes de red, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica.

Wireshark es software libre, y se ejecuta sobre la mayoría de sistemas operativos. En la figura 15.1 se muestra el logo de este programa.

La funcionalidad que provee es añade una interfaz gráfica y muchas opciones de organización y filtrado de información. Así, permite ver todo el tráfico que pasa a través de una red (usualmente una red Ethernet, aunque es compatible con algunas otras) estableciendo la configuración en modo promiscuo.

Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. Wireshark incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

CAPITULO II

2. MARCO METODOLOGICO

2.1 Instalación del sistema operativo Centos

Para la instalación primero se descarga la imagen iso desde el repositorio oficial de Centos en la URL: http://isoredirect.centos.org/centos/7/isos/x86_64/CentOS-7-x86_64-DVD-1511.iso. Se elige por el tipo de sistema de 32 o 64 bits.

Al comienzo de la instalación se visualiza la bienvenida del sistema Operativo Centos con diferentes opciones, se escoge la opción Install or upgrade an existing system y dar un enter de confirmación, como se muestra en la figura 1.2.



Figura 1.2: Página de inicio de instalación Centos

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se escoge skip (found local) como se visualiza en la figura 2.2 para continuar la instalación.

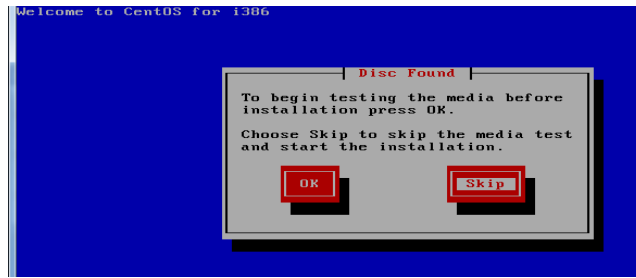


Figura 2.2: Pagina de Configuración 1 Centos

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se elige el idioma en este caso español, que se usa durante la instalación como se muestra en la figura 3.2.

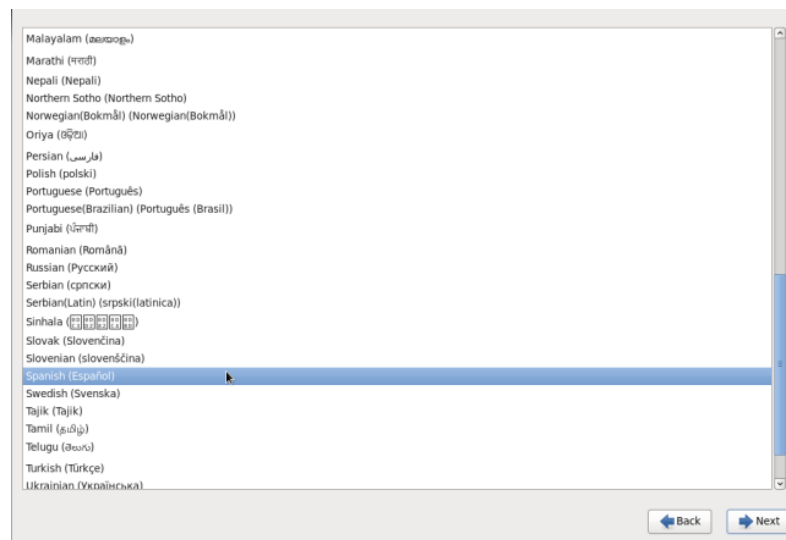


Figura 3.2: Configuración del idioma

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se señala los dispositivos de almacenamiento básicos y dar clic en siguiente, figura 4.2

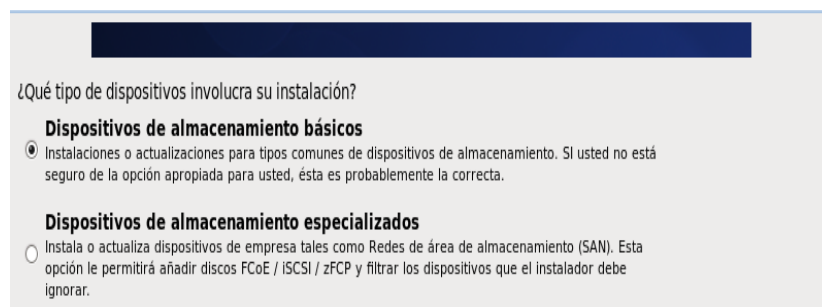


Figura 4.2: Configuración dispositivos

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Elegir la zona donde está ubicado el equipo. Dependiendo de esto será la zona horaria que se activara, ejemplo América/Bogotá, figura 5.2. Dar clic al botón Siguiente.

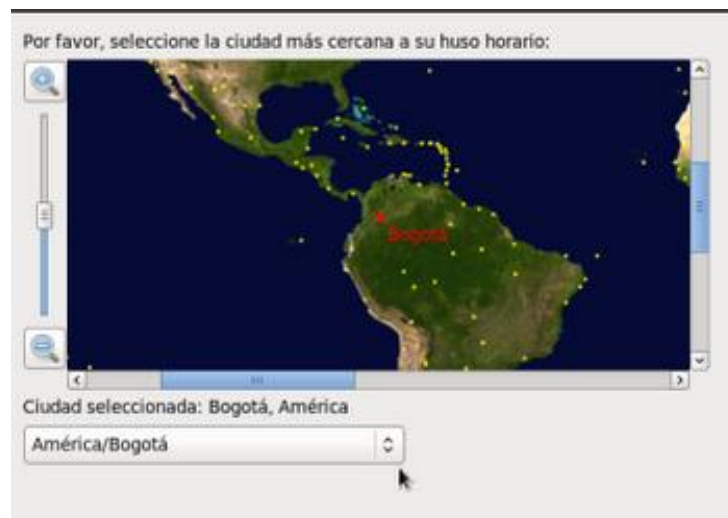


Figura 5.2: Elección de la zona horaria.

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Configurar una contraseña **espochginga** para el usuario **root** que es el administrador del sistema. En la siguiente figura 6.2 se escoge la opción crear un diseño personalizado y dar clic al botón siguiente.

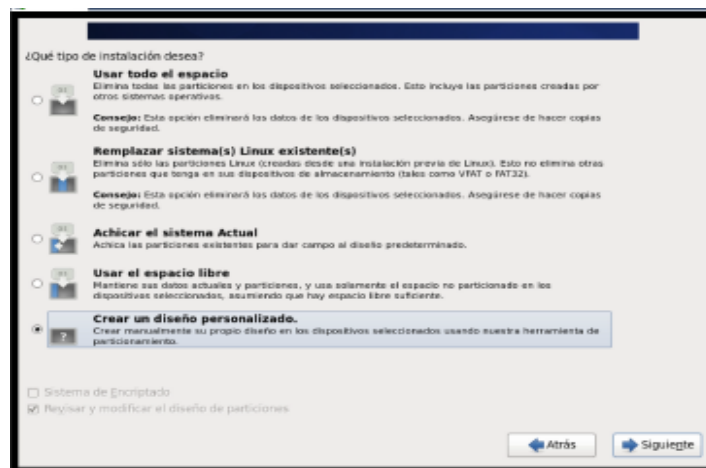


Figura 6.2: Configuración crear diseño personalizado

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Luego se realiza la partición del disco con el espacio recomendado como se muestran en la figura 7.2 con su nombre y tamaño en MB.

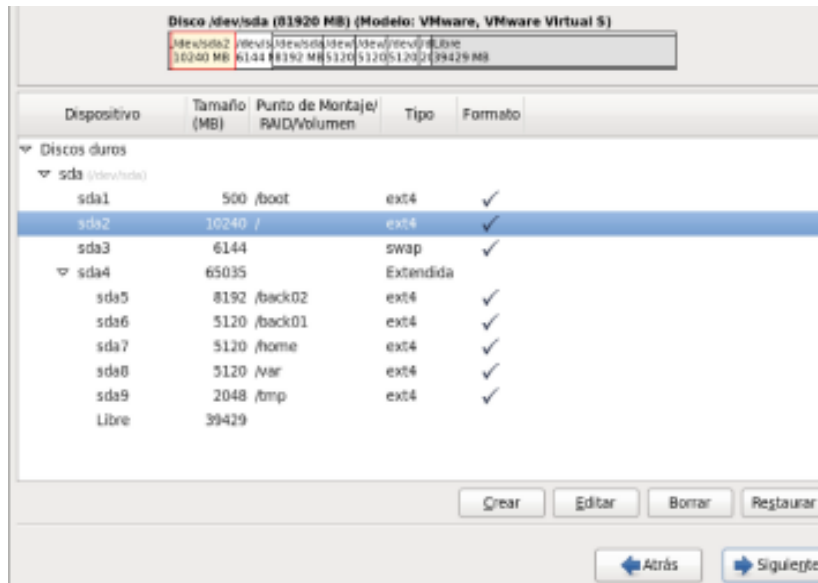


Figura 7.2: Partición de disco

Fuente: Inca, Byron., Sánchez, Verónica. 2016

El siguiente paso es esperar hasta que se instale el sistema operativo. Como se muestra en la figura 8.2

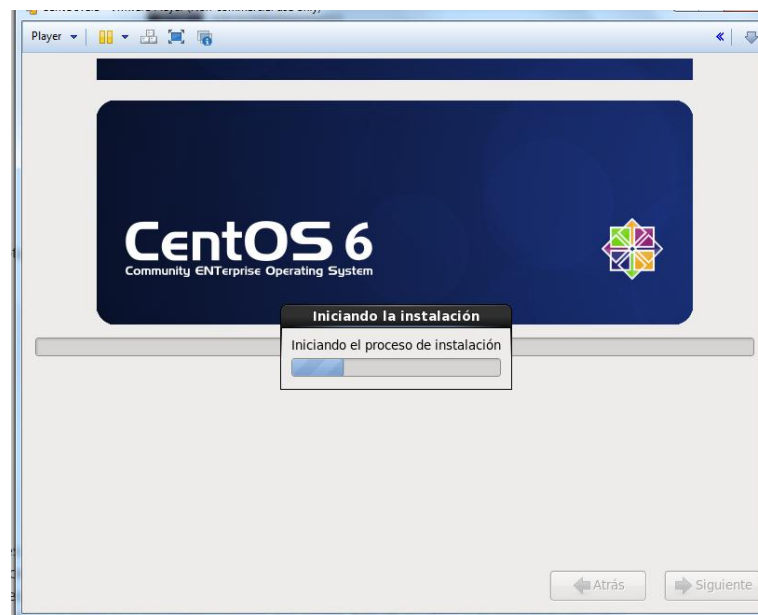


Figura 8.2: Finalización Instalación Centos

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se reinicia el equipo, al principio pide nombre del usuario: **root** y la contraseña configurada: **espochginga**.

2.2 Configuración de la tarjeta de red

Hay dos opciones para configurar la tarjeta de red: De modo consola o modo gráfico. Para mayor facilidad elegir la opción modo gráfico: Abrir la pestaña Sistema, luego en Preferencias y después conexiones de red, obsérvese en la figura 9.2 Aquí se configura la dirección IP estática 172.17.103.107 que se asignó el DTIC de la ESPOCH.

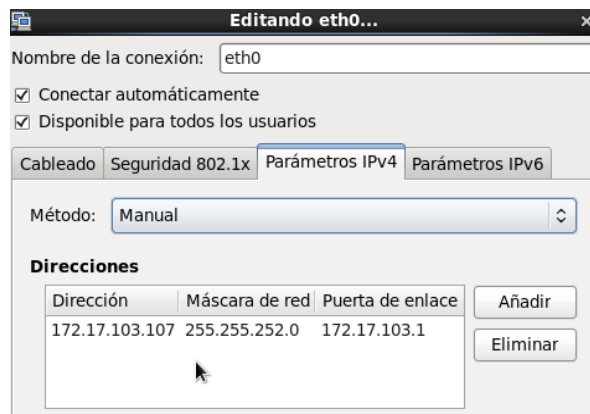


Figura 9.2: Configuración de la IP del servidor

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Luego poner el siguiente comando en modo consola `service network restart` para reiniciar los servicios de red y verificar con `ifconfig` la IP de la maquina ha sido configurada, como en la figura 10.2.

```
Interrupción de la interfaz de loopback:          [ OK ]
Activación de la interfaz de loopback:          [ OK ]
Activando interfaz eth0: Estado de conexión activa: activada
Ruta de conexión activa: /org/freedesktop/NetworkManager/ActiveConnection/2
[ OK ]

[root@vero ~]# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:0C:29:A5:DA:69
          inet addr:172.17.103.107  Bcast:172.17.103.255  Mask:255.255.252.0
          inet6 addr: fe80::20c:29ff:fea5:da69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1022 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:74665 (72.9 KiB)  TX bytes:510 (510.0 b)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:292 errors:0 dropped:0 overruns:0 frame:0
          TX packets:292 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22680 (22.1 KiB)  TX bytes:22680 (22.1 KiB)
```

Figura 10.2: Dirección IP asignada al servidor

Fuente: Inca, Byron., Sánchez, Verónica. 2016

2.3 Instalación y configuración de LAMP: apache mysql y php

Se instala en modo consola los softwares de LAMP: Apache, Mysql y Php de los repositorios oficiales de cada programa, para continuar luego con la programación del servidor:

2.3.1 Instalación del servidor web apache

Instalar el apache server con el siguiente comando en modo consola:

```
Yum -y install http
```

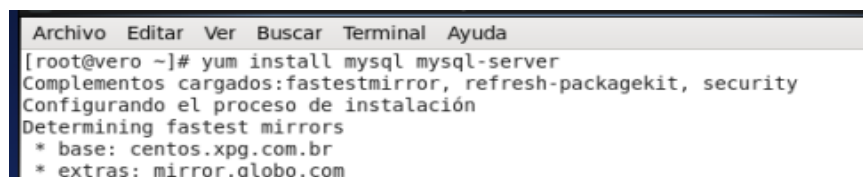
Luego iniciar el servicio de httpd con:

```
service httpd start
```

El comando para ver los canales con los que apache se ejecuta automáticamente al iniciar el sistema operativo es: **chkconfig - -list httpd**. Se habilita en modo encendido 2345 y después configurar con la instrucción **chkconfig httpd on**.o **service httpd start** para levantar apache así quedar activado los servidor web.

2.3.2 Instalación y configuración de mysql

Se descarga los paquetes del repositorio de Mysql con la instrucción: **yum install mysql mysql-server**. En la figura 11.2 se visualiza la ejecución del comando



```
Archivo Editar Ver Buscar Terminal Ayuda
[root@vero ~]# yum install mysql mysql-server
Complementos cargados:fastestmirror, refresh-packagekit, security
Configurando el proceso de instalación
Determining fastest mirrors
 * base: centos.xpg.com.br
 * extras: mirror.globo.com
```

Figura 11.2: Instalación Mysql

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Iniciar el servicio de mysql con el siguiente comando: **service mysqld start**.

Luego establecer la opción que el servicio se levante siempre al iniciar el sistema operativo con la instrucción: **chkconfig - - level 345 mysqld on**.

Después configurar el mysql con el comando: **mysql_secure_installation** para que elimina la base de datos de prueba y asegura el servidor de producción. Luego ingresar la contraseña de root, donde se visualiza la opción de reconfigurar la contraseña del root con la nueva contraseña **ginga**.

A continuación va aparecer varias preguntas donde se confirma con yes o se niega con un no, estas configuraciones se recomienda realizar como la tabla 1.2:

Tabla 1.2 Preguntas de instalación de mysql

PREGUNTA	OPCION
Remover usuarios anónimos	No
Deshabilitar inicio de root remoto	No
Remover base de datos de prueba	Yes
Recargar tabla de privilegios ahora	Yes

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Para finalizar ingresar a mysql con el siguiente comando: **mysql -u root -p** donde se visualiza la consola de mysql, aquí ingresar el siguiente comando para ver la versión que se ha instalada: **SELECT VERSION** y poder saber si es compatible en caso de conexión con otro software.

2.3.3 Instalación y configuración de php

Se instala con el siguiente comando en el terminal: **yum install php php-common php-cli php-mysql -y** que permite la conexión de php a mysql. En la siguiente figura 12.2 se visualiza la descarga de php. Luego agregar mysql a php

```

=====
Paquete      Arquitectura Versión      Repositorio  Tamaño
=====
Instalando:
php          i686         5.3.3-46.el6_7.1  updates     1.1 M
php-cli     i686         5.3.3-46.el6_7.1  updates     2.2 M
php-common  i686         5.3.3-46.el6_7.1  updates     530 k
php-mysql   i686         5.3.3-46.el6_7.1  updates     83 k
Instalando para las dependencias:
php-pdo     i686         5.3.3-46.el6_7.1  updates     78 k
=====
Resumen de la transacción
=====
Instalar      5 Paquete(s)

Tamaño total de la descarga: 4.0 M
Tamaño instalado: 13 M
Está de acuerdo [s/N]:s
Descargando paquetes:
[1/5]: php-5.3.3-46.e (10%) 100% [=====] 58 kB/s | 417 kB  --- ETA

```

Figura 12.2: Instalación PHP

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Instalación y configuración de phpmyadmin

Se conoce que en los repositorios de Centos no contiene a phpmyadmin, por lo que se agrega un repositorio de epel – fedora con la instrucción:

```
rpm -iUvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

Luego instalar phpmyadmi con el comando: yum install phpmyadmin, pero antes se deshabilita los cortafuego debido a la seguridad de Centos porque no están habilitadas las conexiones entrantes.

Para deshabitar el cortafuego ingresamos con el comando: setup y aparece lo siguiente en la figura 13.2. Para habilitar se ingresa en la configuración del cortafuego y se elige la opción desactivar.

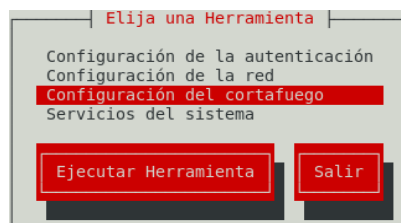


Figura 13.2: Instalación PHP

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Para probar las instalaciones se debe reiniciar el apache con: service httpd restart. Luego ingresar en el navegador la url: localhost/phpmyadmin como en la figura 14.2 y se visualizara la página de bienvenida de PHPMyadmin. Ingresar el usuario root y la contraseña ginga para acceder.



Figura 14.2: Instalación PHPmyAdmin

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Luego se configura y se asigna permisos a la carpeta para poder leer, ejecutar y escribir a los archivos dentro de esta, con el comando: chmod 777 /var/www/html. Se accede a la carpeta en el

sistema de archivos ubicada en el directorio `var/www/html` y está un archivo de prueba con extensión `php`. Se puede acceder a este archivo por el navegador con la Url: `localhost/prueba.php` esto como prueba de que funciona el servidor apache, figura 15.2:



Figura 15.2: Ejercitación del archivo .php

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Para probar abrir el navegador y escribir localhost en el url y se va a direccionar a la página de apache, figura 16.2:

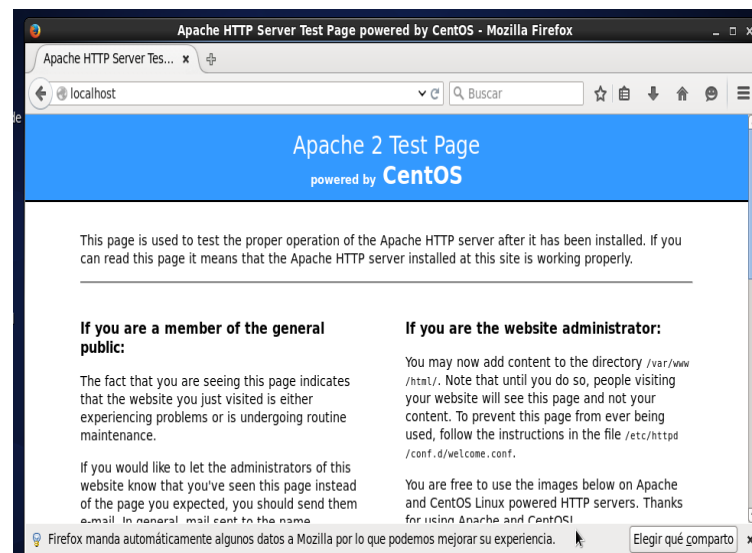


Figura 16.2: Página de Apache en Local host

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Para detener el servicio de apache se ingresa el comando `Service httpd stop` Levantar otra vez el servicio: `service httpd start`. Acceder a la ruta donde se encuentra para modificar el mensaje de bienvenido: `Cd /var/www/html`. Echo `“Centos vero”>>>index.html`

2.4 Creación de una base de datos en PHPMYADMIN

Para crear una base de datos en phpmyadmin se debe iniciar una sesión en el navegador web en www.servidoringa.com/phpmyadmin/, los pasos a seguir son ingresar como root y password, la contraseña es ginga.

Luego hacer clic en la pestaña de Base de datos en blanco, de la pantalla de bienvenida y poner el nombre de base datos denominado en este caso ginga. En la siguiente figura 17.2 se visualiza el nombre de la tabla ginga.



Figura 17.2: Pantalla de configuración de phpmyadmin

Fuente: Inca, Byron., Sánchez, Verónica. 2016

En la pestaña estructura de la figura 18.2 aparece un panel para crear las tablas, aquí es donde se coloca el nombre de la tabla ejemplo programa y se pone el número de columnas que se va trabajar en este caso 2

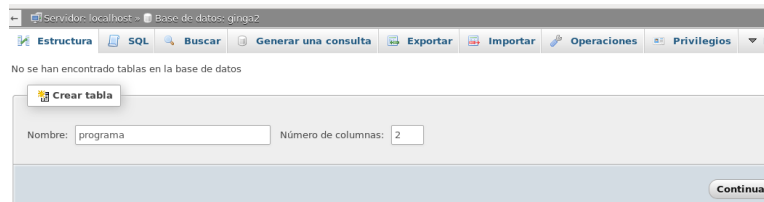


Figura 18.2: Creación de tabla programa

Fuente: Inca, Byron., Sánchez, Verónica. 2016

El panel de opciones se da clic en la base de datos ginga y se sigue el mismo proceso para crear la tabla llamada `votos_programa` con 4 columnas, figura 19.2

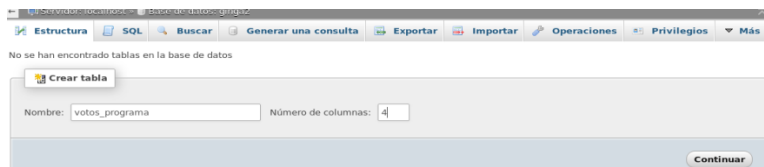


Figura 19.2: Creación de tabla votos_programa

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Luego el panel de opciones presentado permite establecer los parámetros de los campos de cada columna de la tabla programa quedando con los atributos de la siguiente manera en la figura 20.2, donde el campo `id` es la primary key para poder relacionar con la otra tabla `voto_programa`.

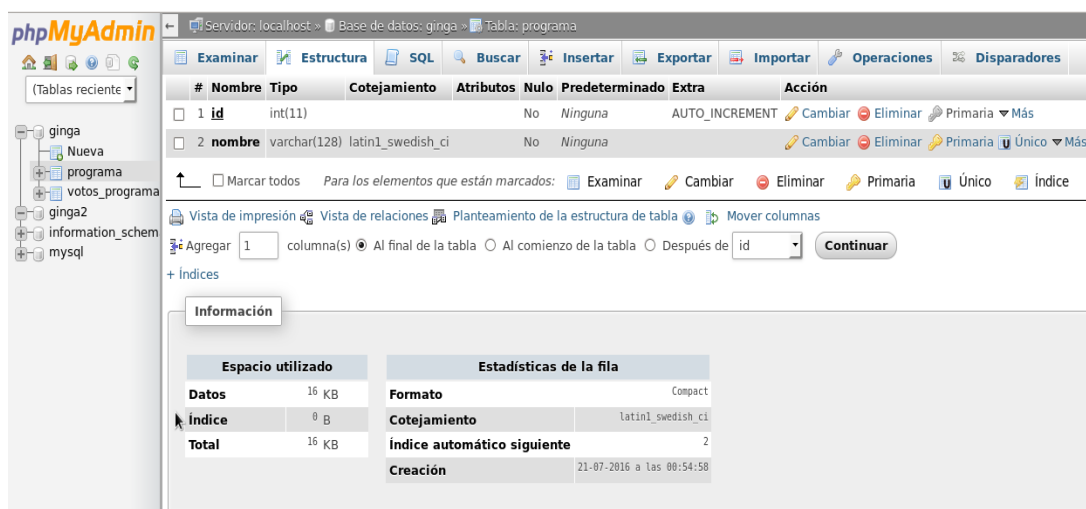


Figura 20.2: Configuración de los campos de la tabla programa

Fuente: Inca, Byron., Sánchez, Verónica. 2016

De igual forma se configura para la tabla `votos_programa` en la figura 21.2, los campos de cada columna siguiendo el mismo método que también se establece el `id` primary key para relacionar con la tabla `programa`



Figura 21.2: Configuración de los campos de la tabla `votos_programa`

Fuente: Inca, Byron., Sánchez, Verónica. 2016

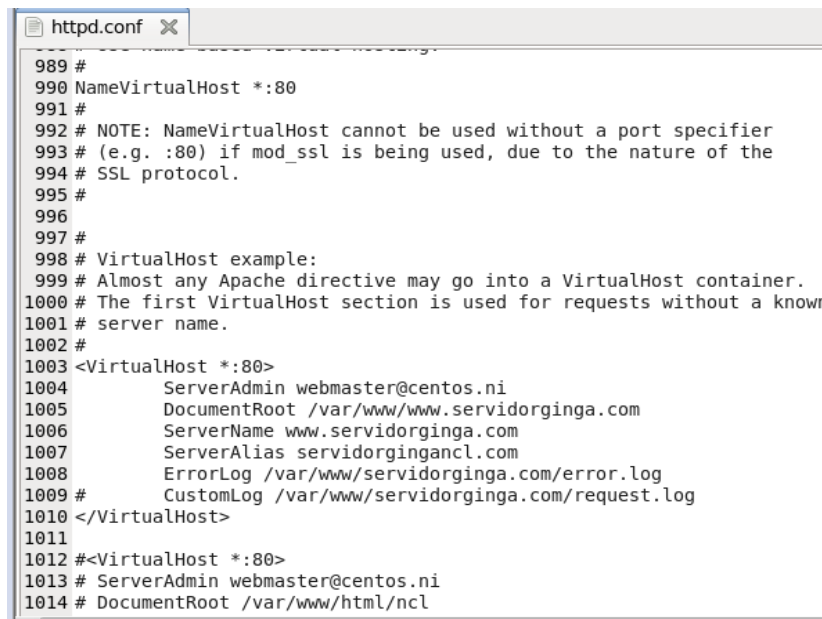
2.5 Creación de un directorio virtual

Se crea la carpeta `www.servidoringa.com` donde se encuentran todos los archivos `php` de la página web, con el siguiente comando: `mkdir /var/www/www.servidoringa.com`

En modo consola ingresar el comando `vi/etc/httpd/conf/httpd.conf` para configurar el archivo `httpd.conf`. Se edita la línea siguiente: `name virtual host` y se descomenta, con esto se activa los dominios virtuales.

```
NameVirtualHost *:80
```

En la siguiente figura 22.2 se muestra las configuraciones del `ServerName`, `Server Admin`, `Document Root` y `Serve Alias` para establecer los directorios virtuales.



```
989 #
990 NameVirtualHost *:80
991 #
992 # NOTE: NameVirtualHost cannot be used without a port specifier
993 # (e.g. :80) if mod_ssl is being used, due to the nature of the
994 # SSL protocol.
995 #
996 #
997 #
998 # VirtualHost example:
999 # Almost any Apache directive may go into a VirtualHost container.
1000 # The first VirtualHost section is used for requests without a known
1001 # server name.
1002 #
1003 <VirtualHost *:80>
1004     ServerAdmin webmaster@centos.ni
1005     DocumentRoot /var/www/www.servidoringa.com
1006     ServerName www.servidoringa.com
1007     ServerAlias servidoringancl.com
1008     ErrorLog /var/www/servidoringa.com/error.log
1009     CustomLog /var/www/servidoringa.com/request.log
1010 </VirtualHost>
1011
1012 #<VirtualHost *:80>
1013 # ServerAdmin webmaster@centos.ni
1014 # DocumentRoot /var/www/html/ncl
```

Figura 22.2: Configuración del directorio virtual

Fuente: Inca, Byron., Sánchez, Verónica. 2016

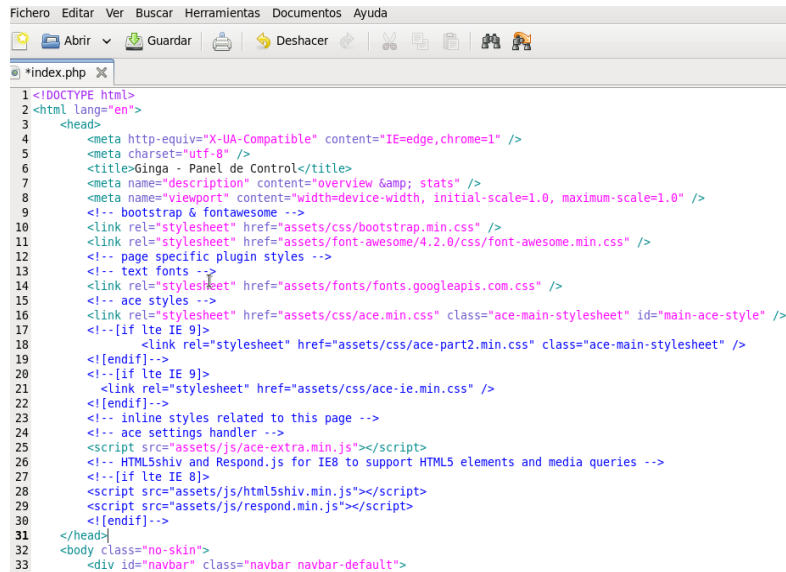
Se reinicia el servicio httpd con: `service httpd restart` para que verificar que el directorio virtual no tiene error.

Se asigna permisos con la instrucción: `Chown -R apache: apache /var/www/www.servidoringa.com`. Así ya se puede programar en el lenguaje php para crear la página web en el archivo `index.php`.

2.6 Creación de la Página Web

Se programa la página web mediante el lenguaje php en 4 archivos: `index.php`, `VotosGinga.php`, `lua_voto_ginga.php` y `Class_mysql.php`, los cuales su código fuente se encuentra en Anexos. En la figura 23.2 muestra el código del archivo `index.php`

Para estructurar la página Web se organiza su contenido para que cada elemento aparezca en el lugar y con el formato deseado. El formato (los estilos) los dejamos para CSS y la estructura la organizamos en HTML.



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
5     <meta charset="utf-8" />
6     <title>Ginga - Panel de Control</title>
7     <meta name="description" content="overview &amp; stats" />
8     <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0" />
9     <!-- bootstrap & fontawesome -->
10    <link rel="stylesheet" href="assets/css/bootstrap.min.css" />
11    <link rel="stylesheet" href="assets/font-awesome/4.2.0/css/font-awesome.min.css" />
12    <!-- page specific plugin styles -->
13    <!-- text fonts -->
14    <link rel="stylesheet" href="assets/fonts/fonts.googleapis.com.css" />
15    <!-- ace styles -->
16    <link rel="stylesheet" href="assets/css/ace.min.css" class="ace-main-stylesheet" id="main-ace-style" />
17    <!--[if lte IE 9]>
18      <link rel="stylesheet" href="assets/css/ace-part2.min.css" class="ace-main-stylesheet" />
19    <![endif]>
20    <!--[if lte IE 9]>
21      <link rel="stylesheet" href="assets/css/ace-ie.min.css" />
22    <![endif]>
23    <!-- inline styles related to this page -->
24    <!-- ace settings handler -->
25    <script src="assets/js/ace-extra.min.js"></script>
26    <!-- HTML5shiv and Respond.js for IE8 to support HTML5 elements and media queries -->
27    <!--[if lte IE 8]>
28      <script src="assets/js/html5shiv.min.js"></script>
29      <script src="assets/js/respond.min.js"></script>
30    <![endif]>
31  </head>
32  <body class="no-skin">
33    <div id="navbar" class="navbar navbar-default">
```

Figura 23.2: Código del archivo index.php

Fuente: Inca, Byron., Sánchez, Verónica. 2016

2.7 Instalación de Software en el cliente

Los softwares que se utilizan en el cliente son: Eclipse, la máquina virtual ginga y lua.

2.7.1 Instalación de Eclipse

Se descarga de la página oficial eclipse la versión galileo en: [www. eclipse.org/galileo/](http://www.eclipse.org/galileo/) pero previamente para que funcione esta plataforma de programación se necesita los archivos JDK de java que puede descargar en la siguiente url:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html?ssSourceSiteId=otnes>

Una vez descargado se abre el programa donde se visualiza la pantalla de inicio de bienvenida y se dirige a la pestaña Help y se selecciona Install New Software para configurar los plugins, Figura 24.2

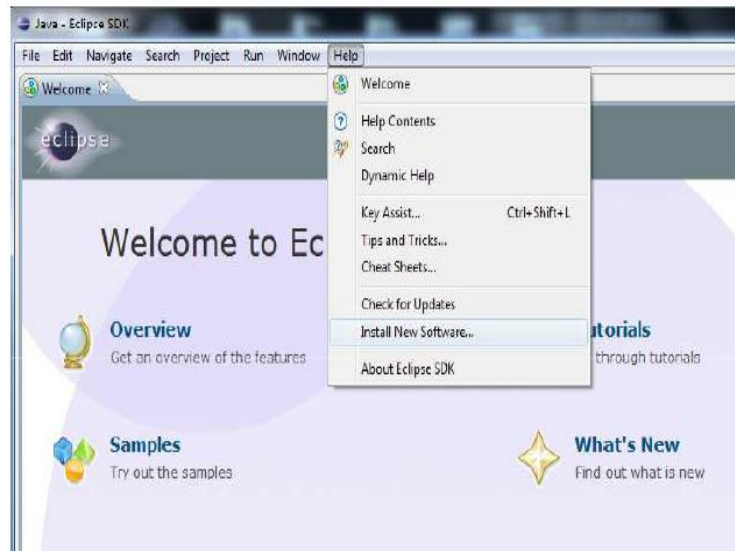


Figura 24.2: Inicio de eclipse

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Ingresa el nombre y la dirección url donde se encuentran el repositorio de los plugin para que se descargue figura 25.2

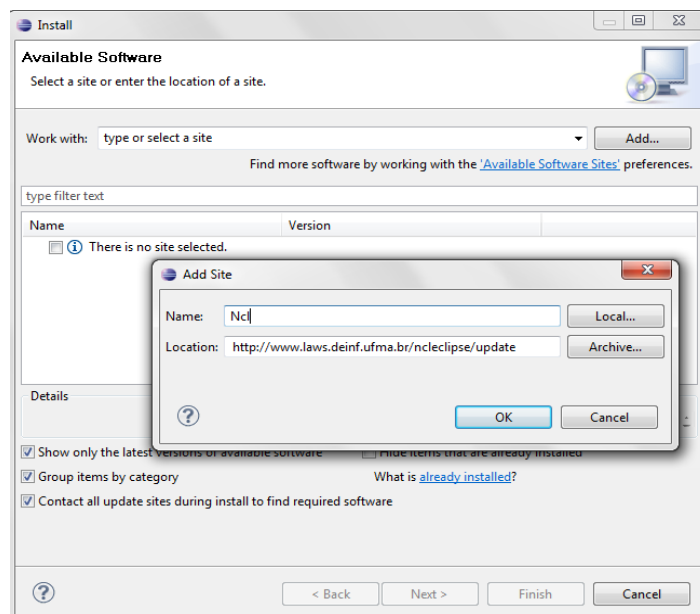


Figura 25.2: Descarga de plugins

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se añade el repositorio NCL Eclipse los plugin necesarios para que la plataforma pueda interpretar el lenguaje NCL, Figura 26.2

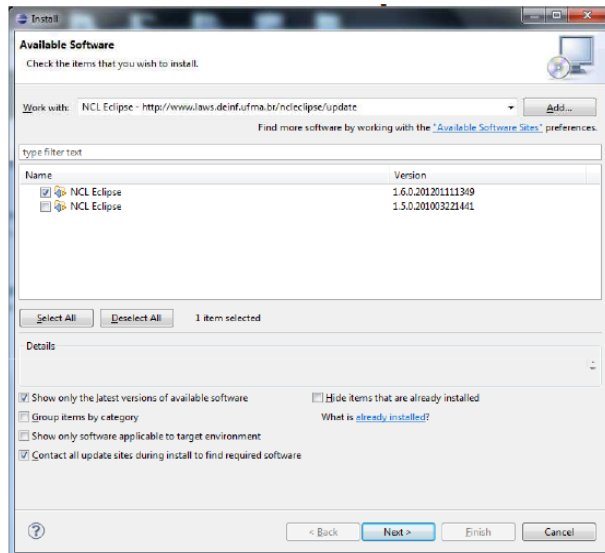


Figura 26.2: Instalación de plugins

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se concluye la instalación de los plugins, así en el eclipse ya se puede programar en el lenguaje declarativo de NCL, figura 27.2

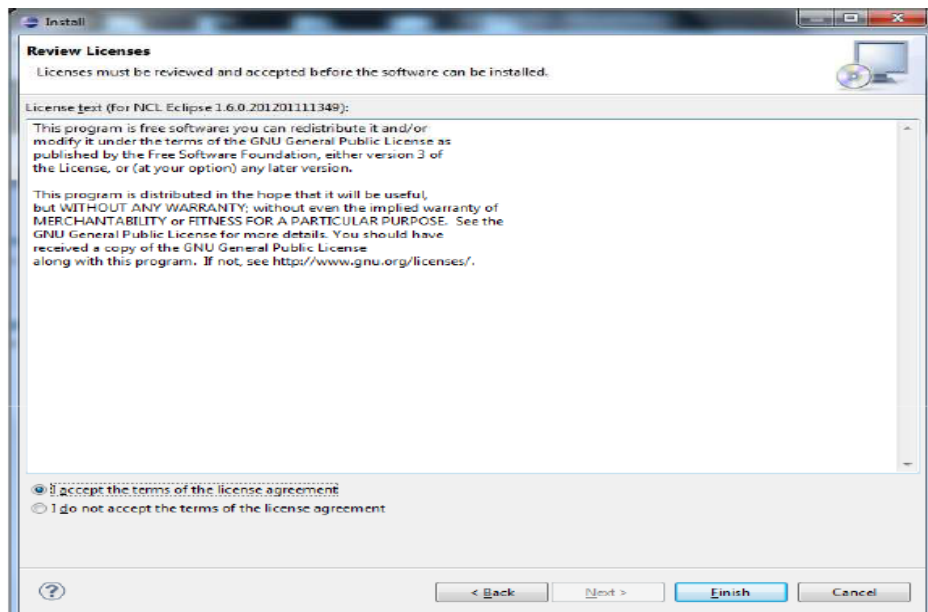


Figura 27.2: Finalización de instalación de Eclipse

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Luego se ingresa en la pestaña Windows y elegir preferencias para configurar la dirección IP de la maquina donde va correr los archivos de programación. La figura 28.2 muestra la configuración en remote ginga NCL play.

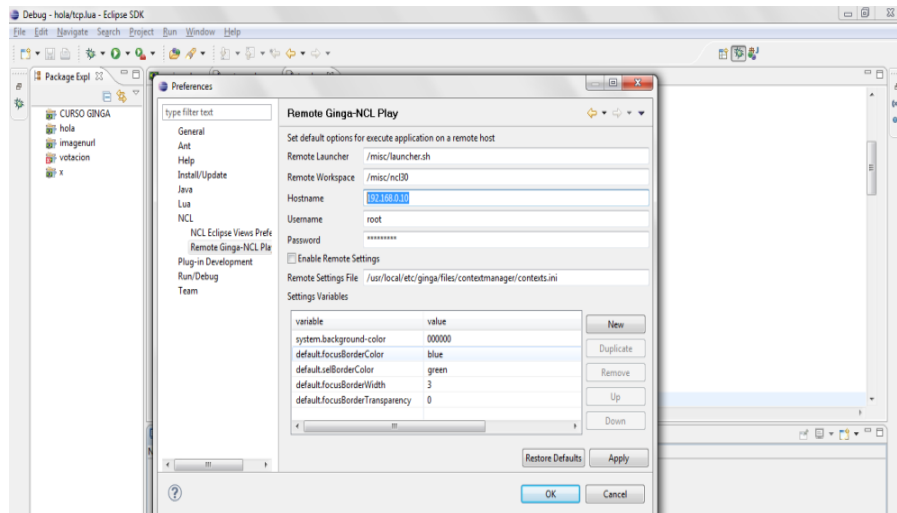


Figura 28.2: Configuración de dirección IP en eclipse

Fuente: Inca, Byron., Sánchez, Verónica. 2016

2.8 Instalación de la middleware ginga ncl set top box virtual

2.8.1 Instalación de la máquina virtual

Se descarga la maquina Ginga Ncl Virtual STB en la dirección: <http://www.gingancl.org.br/pt-br/ferramentas>. Se realizó la simulación del Set Top Box Ginga NCL en la máquina Virtual para ello se utilizó el software VMWare. A continuación se realiza los siguientes pasos:

Se abre la maquina dentro de VMWare, figura 29.2: Se elige la opción open a virtual machine que permite abrir la máquina virtual de ginga. En la figura 1.2 se muestra la página de inicio para abrir una máquina virtual.



Figura 29.2: VMWare opción abrir una máquina virtual

Fuente: Inca, Byron , Sánchez, Verónica. 2016

2.8.2 Configuración de la máquina virtual de ginga

Se abre en la opción GINGA-NCL Virtual STB 0.12.1 with LINUX 2.6.35-22-generic-para configurar en modo texto plano primera opción como la figura 30.2.



Figura 30.2: Inicio de la máquina virtual

Fuente: Inca, Byron., Sánchez, Verónica. 2016

En la siguiente figura 31.2 se visualiza el usuario: root y contraseña: telemidia que se debe ingresar.

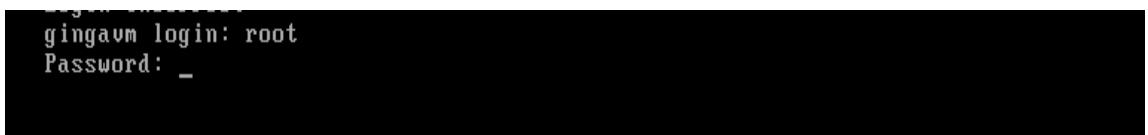


Figura 31.2: Ingreso del usuario y contraseña

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Digitar el comando: `nano /etc/network/interfaces` para la configuración de la dirección IP estática, guardar los cambios con control x , se restablece los servicios de red con el comando: `/etc/init.d/networking restart`


```

GNU nano 2.2.4 File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 172.25.200.65
netmask 255.255.252.0
gateway 172.25.200.1

```

Figura 32.2: Configuración de dirección IP estática

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Configuración del host del servidor con el comando: **vi /etc/hosts** y establecer la dirección IP de la máquina virtual guardando las nuevas configuraciones.

```

GNU nano 2.2.4 File: /etc/hosts

127.0.0.1 localhost
127.0.1.1 gingavm.localdomain gingavm
172.17.103.107 www.servidoringa.com

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

Figura 33.2: Configuración del host Server Name

Fuente: Inca, Byron., Sánchez, Verónica. 2016

2.9 Requisitos de Hardware y Software

En los equipos el hardware es importante para la implementación por lo que se debe cumplir las siguientes características en el cliente, como que muestra la tabla 1.2:

Tabla 2.2: Requisitos de hardware del cliente

Mínimos Computador Cliente	Actuales en el Cliente
Sistema Operativo: XP, Vista, WIN 7 de 32 o 64 bits, Mínimo P4, Mínimo 512MB en RAM, 5GB de espacio Libre en Disco Duro	Sistema Operativo de 64 bits Windows 7, Core i5, RAM tiene 8GB, Espacio en Disco Duro es de 572GB

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Las características de hardware del equipo para el servidor se detallan en la tabla 2.2.

Tabla 3.2: Requisitos de hardware del servidor

Mínimos Computador Servidor	Actuales en el Servidor
Mínimo Pentium 4, Mínimo 512MB en RAM, 5GB de espacio Libre en Disco Duro	Procesador Core i5, RAM tiene 8GB, Disco duro de un 1 TB

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Las características que deben cumplir los softwares se detallan en la tabla 3.2

Tabla 4.2: Requisitos de software del cliente

Mínimos Computador Cliente	Actuales en el Cliente
Vmware Player 4.1 o superior, Máquina Virtual GINGA, JRE 1.6 o superior. Eclipse 3.2 o superior, NCL Eclipse	Vmware Player 12.0, Máquina Virtual GINGA, JRE 1.6 o superior. Eclipse 3.5, NCL Eclipse

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Software Servidor en el equipo del servidor se instaló el sistema operativo Linux la distribución Centos 6.4. En la tabla 4.2 se muestra el nombre del software con su respectiva versión.

Tabla 5.2: Software en el servidor

Nombre del Software	Versión
Apache	2.2.15
PHP	5.3.3
MYsql	5.1.73

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se instaló las versiones mencionadas anteriormente para que exista compatibilidad entre software.

2.10 Estructura cliente – servidor

Se detalla las características de las configuraciones que se realiza en los equipos tanto del cliente y en el servidor como se muestra en la tabla 4.2.

El Esquema de funcionamiento de un Sistema Cliente/Servidor es:

- El cliente solicita una información al servidor.
- El servidor recibe la petición del cliente.
- El servidor procesa dicha solicitud.
- El servidor envía el resultado obtenido al cliente.
- El cliente recibe el resultado y lo muestra

Canal de retorno

El canal de retorno es el canal de transmisión desde el usuario final hasta el proveedor del servicio.

Permite que el usuario este en capacidad de interactuar con el servidor exterior teniendo Interactividad remota. Este tipo de aplicaciones necesita de una conexión a Internet por parte del decodificador para poder enviar los datos.

Se necesita el canal de retorno para este tipo de aplicaciones para tener una comunicación bidireccional.

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.

Este canal de retorno puede ser implementado mediante varias tecnologías existentes, por los diferentes tipos de canal de retorno como: línea telefónica, Ethernet, Bluetooth, Wifi. Pero en televisión digital usamos el Ethernet.

Tabla 6.2: Características Cliente - Servidor

Características del Servidor	Características del Cliente
<div data-bbox="424 394 668 795" data-label="Image"> </div> <p>SERVIDOR</p> <p>IP 172.17.103.107</p> <p>Instalado apache + mysql + php</p> <p>Creado un directorio virtual en /var/www/www.servidoringa.com</p> <p>Tiene cargado el archivo index.php</p> <p>Class_mysql.php lua_votos.php</p> <p>Tiene Centos versión 6.4</p> <p>Hace ping con la maquina cliente</p> <p>RAM 8Gb core i5 disco duro de 1TB</p>	<div data-bbox="850 394 1378 795" data-label="Image"> </div> <p>CLIENTE</p> <p>IP 172.17.200.66 Máquina virtual de GINGA</p> <p>IP 172.17.200.65 maquina física dirección por dhcp</p> <p>Instalado Eclipse</p> <p>Instalado LUA</p> <p>Sistema operativo: Windows 7</p> <p>Core i5 RAM 8GB</p> <p>STB virtual</p> <p>Se tiene el control remoto en un STB en la máquina virtual VMware. Este simula una televisión con el control remoto.</p>

Fuente: Inca, Byron., Sánchez, Verónica. 2016

CAPITULO III

3. MARCO DE RESULTADOS

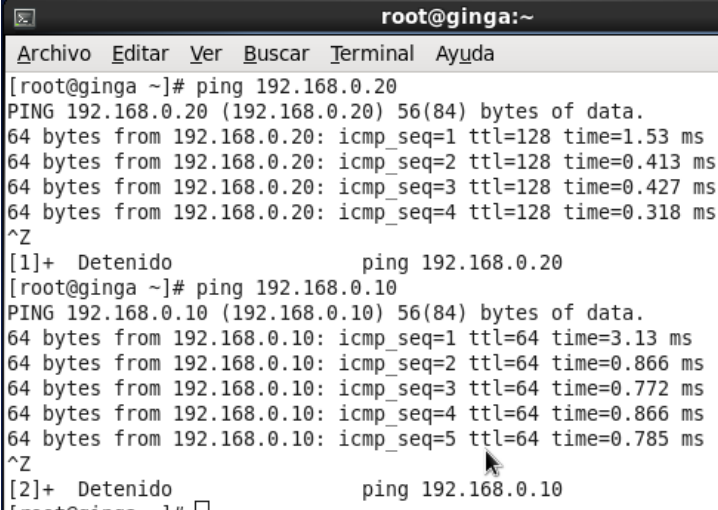
3.1 Ejecución y funcionamiento del servidor

El servidor Linux que almacena contenido interactivo basado en ginga, cuenta con las siguientes partes:

- Página web dinámica con una dirección IP estática
- Almacenamiento de votos en phpmyadmin
- Script en PHP para la actualización de los datos

3.1.1 Comprobación de conexión

Se realizó un diagnostico en la red que comprueba el estado de la comunicación del host local con un equipo remoto de una red IP por medio del envío de paquetes ICMP de solicitud y de respuesta. En la figura 1.3 se muestra el ping que se hace desde el servidor hacia el cliente, en una red local, antes de instalar el servidor en DTIC.



```
root@ginga:~
Archivo Editar Ver Buscar Terminal Ayuda
[root@ginga ~]# ping 192.168.0.20
PING 192.168.0.20 (192.168.0.20) 56(84) bytes of data.
64 bytes from 192.168.0.20: icmp_seq=1 ttl=128 time=1.53 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=128 time=0.413 ms
64 bytes from 192.168.0.20: icmp_seq=3 ttl=128 time=0.427 ms
64 bytes from 192.168.0.20: icmp_seq=4 ttl=128 time=0.318 ms
^Z
[1]+  Detenido                  ping 192.168.0.20
[root@ginga ~]# ping 192.168.0.10
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=64 time=3.13 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=64 time=0.866 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=64 time=0.772 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=64 time=0.866 ms
64 bytes from 192.168.0.10: icmp_seq=5 ttl=64 time=0.785 ms
^Z
[2]+  Detenido                  ping 192.168.0.10
```

Figura 1.3: Ping desde la maquina servidor al cliente.

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Luego se hace ping desde el cliente hacia el servidor, el cual tiene instalado la máquina virtual de ginga, razón por la cual tiene 2 direcciones IP, la una de la maquina física y la otra de la máquina virtual que simula el SET TO BOX. En la figura 2.3 y 3.3 se muestran el ping realizado por parte del cliente:

```
ca: Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Roberto Inca>ping 172.17.103.107

Haciendo ping a 172.17.103.107 con 32 bytes de datos:
Respuesta desde 172.17.103.107: bytes=32 tiempo<1m TTL=60
Respuesta desde 172.17.103.107: bytes=32 tiempo<1m TTL=60
Respuesta desde 172.17.103.107: bytes=32 tiempo<1m TTL=60
Respuesta desde 172.17.103.107: bytes=32 tiempo<1m TTL=60

Estadísticas de ping para 172.17.103.107:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\Roberto Inca>_
```

Figura 2.3: Ping desde la maquina física a la servidor

Fuente: Inca, Byron., Sánchez, Verónica. 2016

En la figura 3.3 también se encuentra el ping de la máquina virtual a la maquina física y desde la maquina física a la máquina virtual en la figura 4.3 determinando así la comunicación satisfactoria.

```
root@gingavn:~# ping 172.17.103.107
PING 172.17.103.107 (172.17.103.107) 56(84) bytes of data:
64 bytes from 172.17.103.107: icmp_req=1 ttl=60 time=0.977 ms
64 bytes from 172.17.103.107: icmp_req=2 ttl=60 time=1.00 ms
64 bytes from 172.17.103.107: icmp_req=3 ttl=60 time=1.12 ms
64 bytes from 172.17.103.107: icmp_req=4 ttl=60 time=1.20 ms
64 bytes from 172.17.103.107: icmp_req=5 ttl=60 time=1.06 ms
64 bytes from 172.17.103.107: icmp_req=6 ttl=60 time=0.955 ms
64 bytes from 172.17.103.107: icmp_req=7 ttl=60 time=0.940 ms
64 bytes from 172.17.103.107: icmp_req=8 ttl=60 time=0.970 ms
^Z
[1]+  Stopped                  ping 172.17.103.107
root@gingavn:~# ping 172.25.200.66
PING 172.25.200.66 (172.25.200.66) 56(84) bytes of data:
64 bytes from 172.25.200.66: icmp_req=1 ttl=128 time=1.28 ms
64 bytes from 172.25.200.66: icmp_req=2 ttl=128 time=1.65 ms
64 bytes from 172.25.200.66: icmp_req=3 ttl=128 time=0.502 ms
64 bytes from 172.25.200.66: icmp_req=4 ttl=128 time=0.511 ms
64 bytes from 172.25.200.66: icmp_req=5 ttl=128 time=0.521 ms
^Z
[2]+  Stopped                  ping 172.25.200.66
root@gingavn:~# _
```

Figura 3.3: Ping desde la máquina virtual al servidor

Fuente: Inca, Byron., Sánchez, Verónica. 20

```

C:\Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Roberto Inca>ping 172.25.200.65

Haciendo ping a 172.25.200.65 con 32 bytes de datos:
Respuesta desde 172.25.200.65: bytes=32 tiempo=1ms TTL=64
Respuesta desde 172.25.200.65: bytes=32 tiempo<1m TTL=64
Respuesta desde 172.25.200.65: bytes=32 tiempo=2ms TTL=64
Respuesta desde 172.25.200.65: bytes=32 tiempo=4ms TTL=64

Estadísticas de ping para 172.25.200.65:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 4ms, Media = 1ms

C:\Users\Roberto Inca>

```

Figura 4.3: Ping desde la máquina física a la máquina virtual

Fuente: Inca, Byron., Sánchez, Verónica. 2016

3.1.2 Ejecución de la aplicación en GINGA NCL

La programación en Ncl y Lua se corre en la plataforma eclipse, donde envía por medio del protocolo ssh a la máquina virtual Set Top Box Ginga, donde se visualiza la aplicación que trata de un video con una pregunta para el usuario, para que interactue en un sistema de votacion de una pregunta con dos opciones para contestar la pregunta con el SI boton verde F2 y NO boton rojo F1 del control remoto . En la figura 5.3 se muestra la ejecucion de la aplicación. c



Figura 5.3: Aplicación interactiva

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Al presionar cualquiera de las dos opciones se va enviar un voto hacia el servidor de contenidos mediante la funcion TCP.lua con lo que el usuario interactua. En este caso el cliente presiono F2

del teclado que significa el boton verde. En la figura 6.3 se visualiza el voto si enviado al servidor.El voto se recibe en el servidor por el canal de retorno y se almacena en la base de datos.

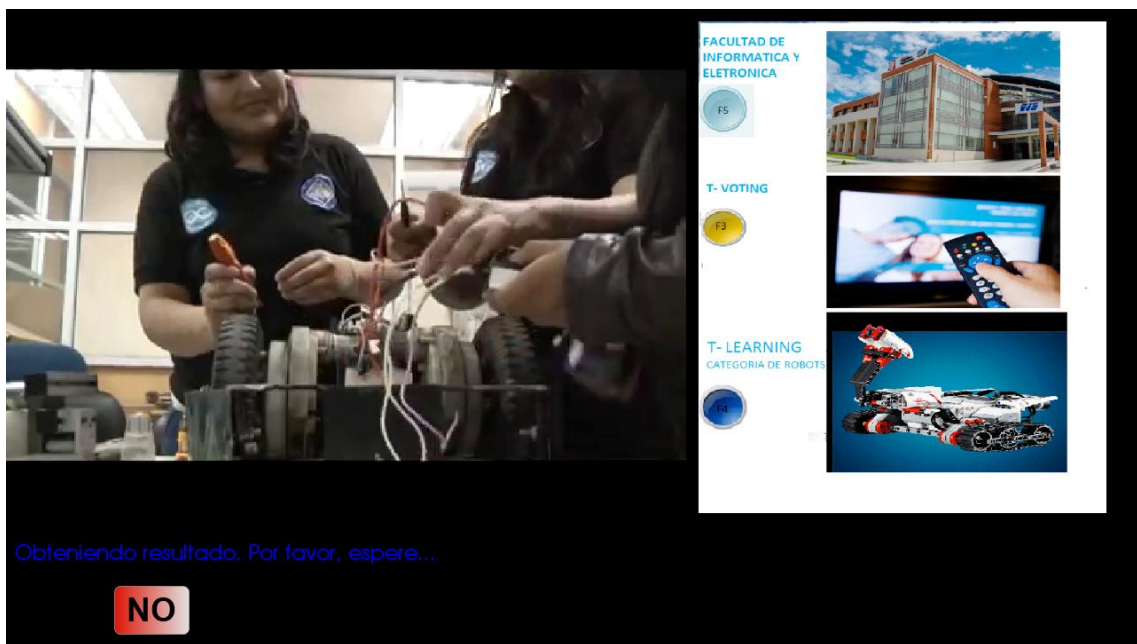


Figura 6.3: Opción votada SI

Fuente: Inca, Byron., Sánchez, Verónica. 2016

El cliente recibe los datos para mostrar en pantalla los resultados de la votación. En la figura 7.3 se visualizó el número de votos del SI y número de votos NO.

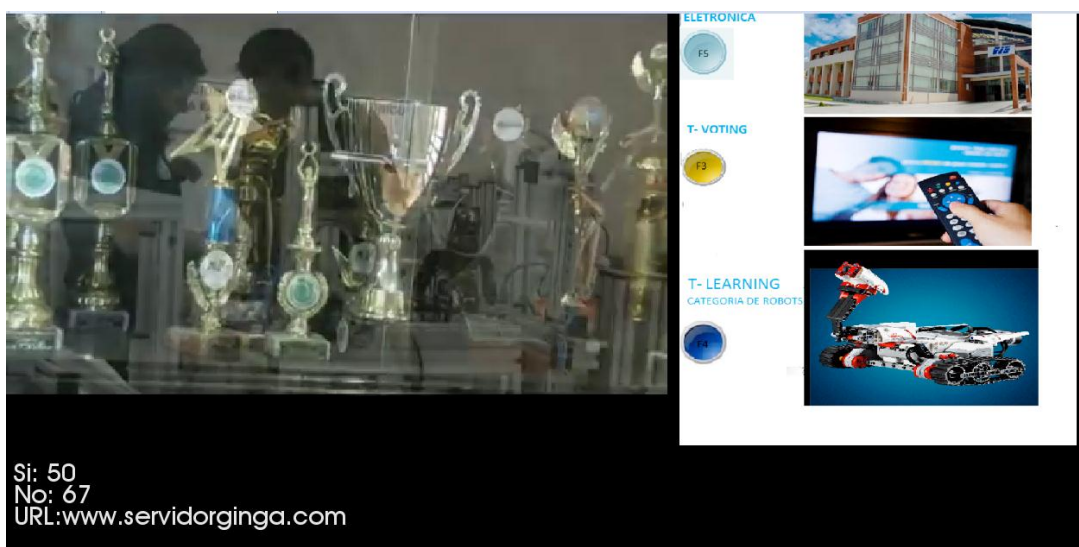


Figura 7.3: Respuesta enviada al cliente

Fuente: Inca, Byron., Sánchez, Verónica. 2016

3.2 Consulta de resultados del servidor de contenido interactivo

En el servidor puede administrar los datos que ha enviado el cliente, donde como resultado se puede ver en la página web www.servidoringa.com con el número de votos de la opción SI y NO. También muestra la página una gráfica con porcentajes del sistema de votación, figura 8.3

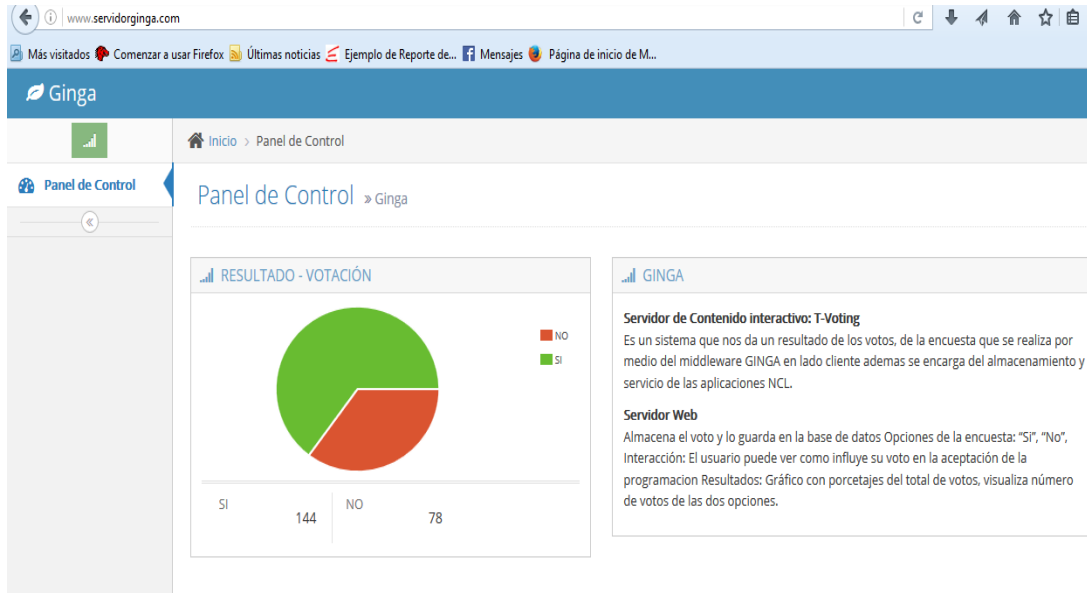


Figura 8.3: Página web del Servidor

Fuente: Inca, Byron., Sánchez, Verónica. 2016

El servidor guarda en la base de datos el conteo de votos, donde se almacena los campos con información más detallada del sistema de votación, para ver esto ingresar en el navegador www.servidoringa.com/phpmyadmin con Usuario: **root** y contraseña **ginga**. En la figura 9.3 se ve la página de administración phpmyadmin.



Figura 9.3: Página de inicio de PHPmyadmin

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Al ingresar a la base de datos llamada **ginga** se verifica los resultados obtenidos en la comunicación cliente-servidor. Se puede evidenciar mediante las consultas SQL ingresando en la tabla votos programa que el contenido interactivo del cliente queda almacenado., toda esta información para gestionar.

Los resultados después de realizar la pruebas funcionales se visualiza en la figura 10.3 con el conteo de número de votos del Sí y No con la respectiva fecha que se realizó el voto con lo que queda comprobado que existe la comunicación entre el cliente y el servidor al existir los registros del sistema de votación.

ID	status	type	date
12	1	n	2016-07-18
13	1	n	2016-07-18
14	1	n	2016-07-18
15	1	n	2016-07-18
16	1	n	2016-07-18
17	1	n	2016-07-18
18	1	n	2016-07-18
19	1	n	2016-07-18
20	1	n	2016-07-18
21	1	n	2016-07-18
22	1	n	2016-07-18
23	1	n	2016-07-18
24	1	n	2016-07-18
25	1	n	2016-07-18
26	1	n	2016-07-18
27	1	s	2016-07-18
28	1	s	2016-07-18
29	1	s	2016-07-18
30	1	s	2016-07-18

Figura 10.3: Visualización votos en PHPmyadmin

Fuente: Inca, Byron., Sánchez, Verónica. 2016

3.3 Evaluación de desempeño del servidor de aplicaciones y contenidos

3.3.1 Evaluación del envío y recepción de la trama

Evaluación del desempeño del servidor, en la figura 11.3 se muestra un esquema del servidor LAMP y el Cliente con los software utilizados.

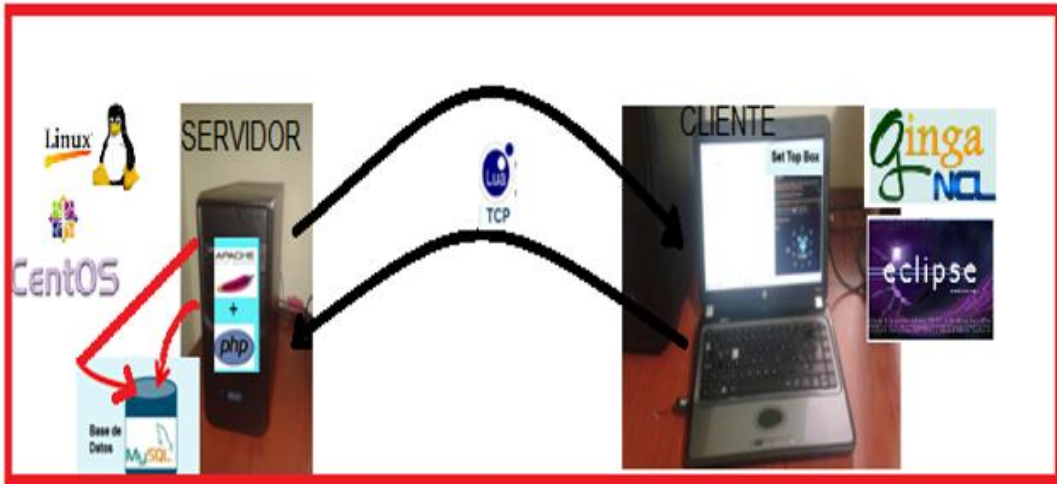


Figura 11.3: Funcionamiento del sistema

Fuente: Inca, Byron., Sánchez, Verónica. 2016

En la ejecución de la clase TCP dentro del archivo LUA, existen dos fases principales:

- El envío de datos el cual es una petición TCP ejecutada por el cliente
- La recepción de votos totales por cada opción, se hace con la opción TCP Get dentro de la librería TCP.

Luego de realizar la petición de una conexión TCP con el servidor, el equipo usuario realiza el envío de datos. Este paquete debe tener una longitud de 74 bytes, con su respectivo encabezado de red, encabezado TCP, y por lo tanto las direcciones IP de los equipos, incluido el número de puerto, todos estos parámetros se los configura en la clase TCPLua, la cual se encarga de anexar los datos del voto realizado en la trama TCP.

En la figura 12.3 podemos observar los datos capturados de una de las tramas TCP de envío de datos de la aplicación, y sus características en la tabla 1.3.

No.	Time	Source	Destination	Protocol	Len	Info
13842	79.070446	77.234.42.26	172.25.200.66	HTTP	234	HTTP/1.1 200 OK (application/octet-stream)
13843	79.073255	172.25.200.66	77.234.42.26	HTTP	364	GET /R/A34KIDVzZNY4ZTE10GY30TQxMmQ5ZTkzYmU4MjdMzA2YjVjEgQCJxAWGJABIGeEKgQIBRABH
13873	79.159138	77.234.42.26	172.25.200.66	TCP	60	80 → 59749 [ACK] Seq=181 Ack=311 Win=2 Len=0
14064	79.902889	172.25.200.65	172.17.103.107	TCP	74	55547 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=4294918329 TSecr=
14065	79.903704	172.17.103.107	172.25.200.65	TCP	74	80 → 55547 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1380 SACK_PERM=1 TSval=166
14066	79.904332	172.25.200.65	172.17.103.107	TCP	66	55547 → 80 [ACK] Seq=1 Ack=1 Win=5856 Len=0 TSval=4294918330 TSecr=1660930217
14138	80.411423	172.25.200.65	172.17.103.107	TCP	128	[TCP segment of a reassembled PDU]
14139	80.413590	172.17.103.107	172.25.200.65	TCP	66	80 → 55547 [ACK] Seq=1 Ack=63 Win=14528 Len=0 TSval=166030726 TSecr=4294918457
14152	80.476090	172.17.103.107	172.25.200.65	TCP	125	80 → 55547 [PSH, ACK] Seq=1 Ack=63 Win=14528 Len=59 TSval=166030789 TSecr=429491

Figura 12.3: Wireshark captura de tramas de envío

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Tabla 1.3: Características de la trama de envío

No. Posición del paquete en la captura.	523
Time. Muestra el Timestamp del paquete.	333.48996564
Source. Dirección origen del paquete.	192.168.0.10
Destination. Dirección destino del paquete.	192.168.0.100
Protocol. Nombre del protocolo del paquete.	TCP

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Una vez notificado el voto al servidor, la aplicación se encarga de publicar en pantalla los resultados hasta el momento en el que el usuario realizó su voto.

Este proceso se realiza mediante la subclase TCP Get de la clase TCPLua, en el cual el equipo de usuario realiza otra petición TCP al servidor, pero esta vez para leer los datos de la base de datos con el No o Si que se genera por el archivo PHP.

En la figura 13.3 se despliega información detallada del paquete seleccionado en el panel de paquetes:

```
Frame 14064: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
  Interface id: 0 (\Device\NPF_{280D3DB4-E927-44C6-831C-E97064C7EEBC})
  Encapsulation type: Ethernet (1)
  Arrival Time: Oct 28, 2016 10:01:52.686476000 Hora est. Pacífico, Sudamérica
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1477666912.686476000 seconds
  [Time delta from previous captured frame: 0.001221000 seconds]
  [Time delta from previous displayed frame: 0.743751000 seconds]
  [Time since reference or first frame: 79.902889000 seconds]
  Frame Number: 14064
  Frame Length: 74 bytes (592 bits)
  Capture Length: 74 bytes (592 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp]
  [Coloring Rule Name: HTTP]
  [Coloring Rule String: http || tcp.port == 80 || http2]
  Ethernet II, Src: Vmware_27:96:4d (00:50:56:27:96:4d), Dst: CiscoInc_75:6a:62 (00:57:d2:75:6a:62)
  Internet Protocol Version 4, Src: 172.25.200.65, Dst: 172.17.103.107
  Transmission Control Protocol, Src Port: 55547 (55547), Dst Port: 80 (80), Seq: 0, Len: 0
```

Figura 13.3: Información detallada del paquete

Fuente: Inca, Byron., Sánchez, Verónica. 2016

En la figura 14.3 se muestra que el segmento TCP de solicitud es de 66 bytes de longitud, por lo cual se nota el retardo al momento de realizar la respuesta desde el servidor. Además se muestra las características de la trama TCP de petición desde el equipo de usuario.

No.	Time	Source	Destination	Protocol	Len	Info
13842	79.070446	77.234.42.26	172.25.200.66	HTTP	234	HTTP/1.1 200 OK (application/octet-stream)
13843	79.073255	172.25.200.66	77.234.42.26	HTTP	364	GET /R/A34KIDYzZwY4ZTE10GY3OTQxMmQ5ZTkzYwU4MjdhMzA2YjVjEgQCJxAWGJABIGeEQIIBRABKcIBBD:
13873	79.159138	77.234.42.26	172.25.200.66	TCP	60	80 → 59749 [ACK] Seq=181 Ack=311 Win=2 Len=0
14064	79.902889	172.25.200.65	172.17.103.107	TCP	74	55547 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=4294918329 TSecr=0 WS=
14065	79.903704	172.17.103.107	172.25.200.65	TCP	74	80 → 55547 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1380 SACK_PERM=1 TSval=166030217
14066	79.904332	172.25.200.65	172.17.103.107	TCP	66	55547 → 80 [ACK] Seq=1 Ack=1 Win=5856 Len=0 TSval=4294918330 TSecr=166030217
14138	80.411243	172.25.200.65	172.17.103.107	TCP	128	[TCP segment of a reassembled PDU]
14139	80.413590	172.17.103.107	172.25.200.65	TCP	66	80 → 55547 [ACK] Seq=1 Ack=63 Win=14528 Len=0 TSval=166030726 TSecr=4294918457
14152	80.476090	172.17.103.107	172.25.200.65	TCP	125	80 → 55547 [PSH, ACK] Seq=1 Ack=63 Win=14528 Len=59 TSval=166030789 TSecr=4294918457
14153	80.476092	172.17.103.107	172.25.200.65	TCP	66	80 → 55547 [FIN, ACK] Seq=60 Ack=63 Win=14528 Len=0 TSval=166030789 TSecr=4294918457
14154	80.476515	172.25.200.65	172.17.103.107	TCP	66	55547 → 80 [ACK] Seq=63 Ack=60 Win=5856 Len=0 TSval=4294918473 TSecr=166030789
14158	80.516232	172.25.200.65	172.17.103.107	TCP	66	55547 → 80 [ACK] Seq=63 Ack=61 Win=5856 Len=0 TSval=4294918483 TSecr=166030789
14272	81.412634	172.25.200.65	172.17.103.107	TCP	66	55547 → 80 [FIN, ACK] Seq=63 Ack=61 Win=5856 Len=0 TSval=4294918707 TSecr=166030789

Figura 14.3: Trama TCP desde servidor al usuario

Fuente: Inca, Byron., Sánchez, Verónica. 2016

En la siguiente figura 15.3 se muestra un retardo que sufre con un length de 54 y 66 bytes, en la dirección IP de la maquina física hacia la máquina virtual, por lo que la respuesta del servidor al cliente también sufre este retardo.

```

# Frame 14139: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
  Interface id: 0 (\Device\NPF_{28803DB4-E927-44C6-831C-E97064C7EEBC})
  Encapsulation type: Ethernet (1)
  Arrival Time: Oct 28, 2016 10:01:53.197177000 Hora est. Pacífico, Sudamérica
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1477666913.197177000 seconds
  [Time delta from previous captured frame: 0.002167000 seconds]
  [Time delta from previous displayed frame: 0.002167000 seconds]
  [Time since reference or first frame: 80.413590000 seconds]
  Frame Number: 14139
  Frame Length: 66 bytes (528 bits)
  Capture Length: 66 bytes (528 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp]
  [Coloring Rule Name: HTTP]
  [Coloring Rule String: http || tcp.port == 80 || http2]
  > Ethernet II, Src: CiscoInc_75:6a:62 (00:57:d2:75:6a:62), Dst: Vmware_27:96:4d (00:50:56:27:96:4d)
  > Internet Protocol Version 4, Src: 172.17.103.107, Dst: 172.25.200.65
  > Transmission Control Protocol, Src Port: 80 (80), Dst Port: 55547 (55547), Seq: 1, Ack: 63, Len: 0

```

Figura 15.3: Información de trama TCP de respuesta

Fuente: Inca, Byron., Sánchez, Verónica. 2016

3.3.2 Evaluación de Disponibilidad

El servidor se encuentra en el departamento de Tecnologías de Información y Comunicación con una disponibilidad del 100%, es decir se le tiene encendido y funcionando todo el día y la noche. En caso de corte de energía eléctrica también se encontrara disponible porque cuenta la ESPOCH con el sistema de alimentación ininterrumpida (UPS).

Para saber la disponibilidad mensual de un servidor que recibe un mantenimiento de 4 hrs x mes. Se calcula con la siguiente formula

Dicho de otra forma más simplificada: $CA = (ATS - DT / ATS) \times 100$

El Agreed Time Service (Tiempo acordado del servicio): se determina tomando en cuenta que el servidor trabaja las 24/7 por mes, es decir: $24 \times 30 = 720 \text{ hrs}$ y a esto le reducimos las 4 hrs del mantenimiento del servidor que se considera, $720 - 4 = 716$

Downtime: Este también se determina tomando en cuenta el tiempo que tardaríamos en dar el mantenimiento al servidor

$CA = (716 - 4) / 716 \times 100 = 99\%$

3.3.3 Evaluación del tiempo de Respuesta

El rendimiento es la capacidad de un sistema para minimizar el tiempo de respuesta. Para tener el tiempo de respuesta nos basamos en la votación según el número de usuarios que envían datos al servidor.

Se evaluó el servidor con 10 usuarios que utilizaban la aplicación de t-voting al mismo tiempo. Se analizó la tramas de envío tcp a través del puerto 80 y como resultados se dieron diferentes tiempo al realizar todo este proceso. En la tabla 2.3 Se observa los tiempos de envío del voto desde el cliente al servidor.

Tabla 2.3 Tiempos de envío del voto desde el cliente al servidor

USUARIO	ORIGEN	DESTINO	PROTOCOLO	LONGITUD [bytes]	TIEMPO [s]
1	172.25.200.67	172.17.103.107	TCP	74	0,101568
2	172.25.200.232	172.17.103.107	TCP	74	0,131941
3	172.25.200.125	172.17.103.107	TCP	74	0,123342
4	172.25.200.181	172.17.103.107	TCP	74	0,110301
5	172.25.201.124	172.17.103.107	TCP	74	0,143912
6	172.25.201.51	172.17.103.107	TCP	74	0,172673
7	172.25.201.61	172.17.103.107	TCP	74	0,123982
8	172.25.201.208	172.17.103.107	TCP	74	0,131923
9	172.25.201.132	172.17.103.107	TCP	74	0,169191
10	172.25.201.134	172.17.103.107	TCP	74	0,197354

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se analizó la tramas de respuesta del servidor y los resultados de la votación a través del puerto 80 y como resultados se dieron diferentes tiempo al realizar todo este proceso. En la tabla 3.3 Se observa los tiempos de envió del voto desde el servidor al cliente

Tabla 3.3 Tiempos de envió del voto desde el servidor al cliente

USUARIO	ORIGEN	DESTINO	PROTOCOLO	LONGITUD [byte]	TIEMPO [s]
1	172.17.103.107	172.25.200.67	TCP	66	0,257475
2	172.17.103.107	172.25.200.232	TCP	66	0,290442
3	172.17.103.107	172.25.200.125	TCP	66	0,250391
4	172.17.103.107	172.25.200.181	TCP	66	0,272134
5	172.17.103.107	172.25.201.124	TCP	66	0,234443
6	172.17.103.107	172.25.201.51	TCP	66	0,297911
7	172.17.103.107	172.25.201.61	TCP	66	0,247501
8	172.17.103.107	172.25.201.208	TCP	66	0,252184
9	172.17.103.107	172.25.201.132	TCP	66	0,289822
10	172.17.103.107	172.25.201.134	TCP	66	0,206455

Fuente: Inca, Byron., Sánchez, Verónica. 2016

La suma del tiempo de envió de datos más el tiempo de respuesta de resultados será igual al tiempo total que tarda en mostrar los resultados de la votación en pantalla del usuario, tabla 4.3.

Tabla 4.3 Tiempos de envió del voto total

Usuario	t de envió	t de respuesta	t total
1	0,101568	0,257475	0,359043
2	0,131941	0,290442	0,422383
3	0,123342	0,250391	0,373733
4	0,110301	0,272134	0,382435
5	0,143912	0,234443	0,378355
6	0,172673	0,297911	0,470584
7	0,123982	0,247501	0,371483
8	0,131923	0,252184	0,384107
9	0,169191	0,289822	0,459013
10	0,197354	0,206455	0,403809

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Arrojando como resultado que el promedio de tiempo que tarda en mostrar los resultado de la votación, valor máximo y mínimo en realizar todo estos procesos, tabla 5.3.

Tabla 5.3 Tiempos promedio, máximo y mínimo.

PROMEDIO	0,4004945
VALOR MAX	0,470584
VALOR MIN	0,359043

Fuente: Inca, Byron., Sánchez, Verónica. 2016

En la tabla 6.3 se muestra que la mayor cantidad de votos hace que la respuesta del servidor tenga un retardo pero también muestra que los votos siempre se están almacenando en la base de datos.

Se ha realizado una proyección del tiempo que el servidor tarda en responder al incrementar el número de usuarios.

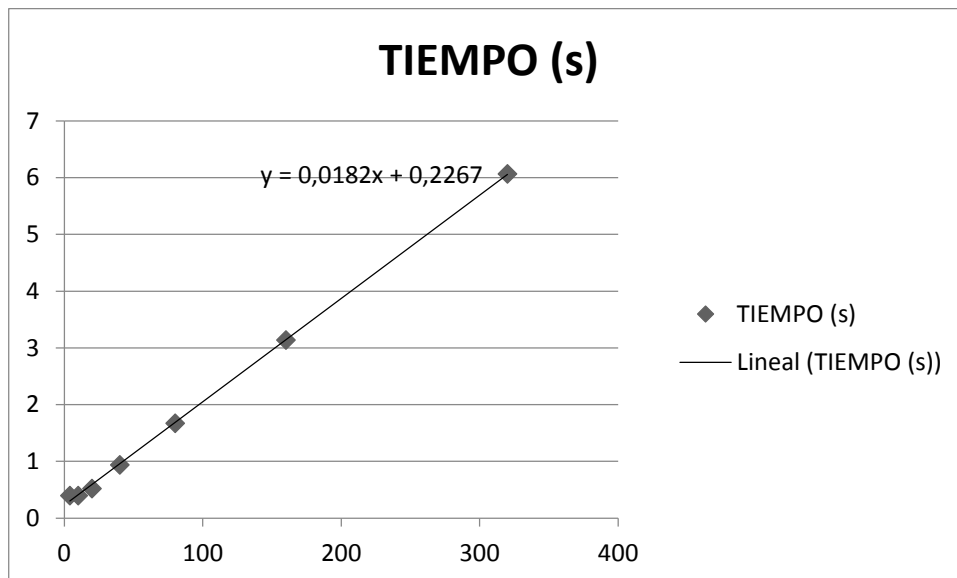
Tabla 6.3 Comparación en el tiempo de respuesta

USUARIOS	TIEMPO (s)
4	0,399081
10	0,400494
20	0,521532
40	0,940526
80	1,673406
160	3,139166
320	6,070686

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Se estima que al completar unos 200 usuarios el tiempo en responder al servidor es de 3s, figura 16.3

Grafico 1.3: Proyección de tiempo de respuesta



Fuente: Inca, Byron., Sánchez, Verónica. 2016

También se hizo una evaluación por parte de los estudiantes de la FIE, dando sus respuestas en una encuesta. Se ha realizado la encuesta luego de haber utilizado la aplicación de T-Voting y como resultado se muestra la tabulación siguiente:

- **Población:** los estudiantes de la Facultad de Informática y Electrónica
- **Muestra:** los estudiantes de las diferentes Escuelas de la FIE

Encuesta

Objetivos

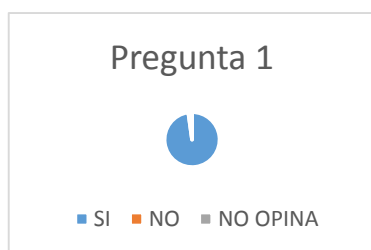
- Conocer la calidad de servicio que brinda el servidor de aplicaciones.4
- Conocer cuántos usuarios soporta el sistema de votación
- Conocer la respuesta del servidor en un tiempo específico

Tabla 7.3 Pregunta 1. En el servidor de aplicaciones se almacena su voto?

SI	45	97,83%
NO	1	2,17%
NO OPINA	0	0,00%

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Grafico 2. 3 Pregunta 1



Fuente: Inca, Byron., Sánchez, Verónica. 2016

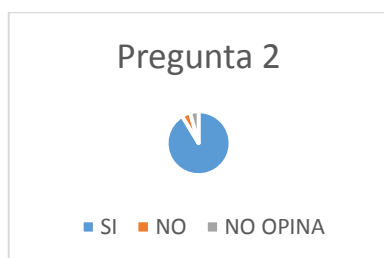
Interpretación: En el grafico 1.3 se muestra que los votos realizados por los estudiantes se almaceno un 98 % en el servidor.

Tabla 8.3 Pregunta 2. EL servidor muestra la cantidad de votos totales de las 2 opciones SI y NO en su página web. Verificar si se actualizó ingresando a la página web?

SI	42	91,30%
NO	2	4,35%
NO OPINA	2	4,35%

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Grafico 3. 3 Pregunta 2



Fuente: Inca, Byron., Sánchez, Verónica. 2016

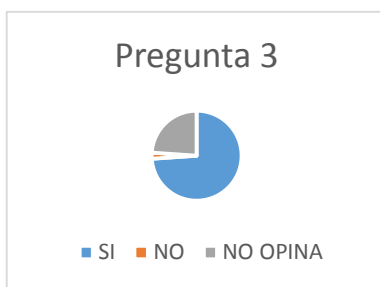
Interpretación: En el grafico 2.3 se visualiza que el servidor muestra los votos en su página web en un 92%.

Tabla 9.3 Pregunta 3. .- El servidor dio una respuesta inmediata de los resultados de la votación en un tiempo estimado de 1 minuto?

SI	34	73,91%
NO	1	2,17%
NO OPINA	11	23,91%

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Grafico 4. 3 Pregunta 3



Fuente: Inca, Byron., Sánchez, Verónica. 2016

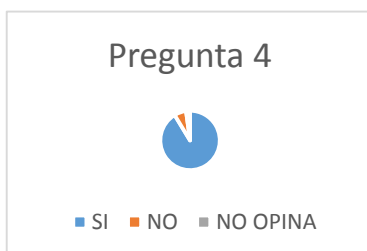
Interpretación: En el grafico 3.3 se muestra que la respuesta inmediata del servidor está en un 74% disponible para su ejecución.

Tabla 10.3 Pregunta 4. El tiempo de respuesta de resultados que se muestra en pantalla fue rápido?

SI	42	91,30%
NO	3	6,52%
NO OPINA	1	2,17%

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Grafico 5. 3 Pregunta 4



Fuente: Inca, Byron., Sánchez, Verónica. 2016

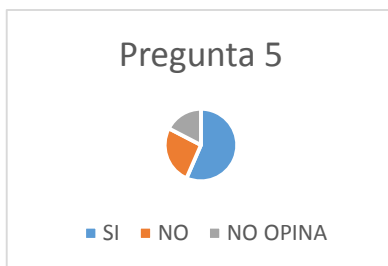
Interpretación: En el grafico 4.3 se muestra que el tiempo de respuesta del servidor es rápida en un 91%.

Tabla 11.3 Pregunta 5. Según su criterio le resultado fácil el manejo del sistema de votación T-voting.

SI	26	56,52%
NO	12	26,09%
NO OPINA	8	17,39%

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Grafico 6. 3 Pregunta 5



Fuente: Inca, Byron., Sánchez, Verónica. 2016

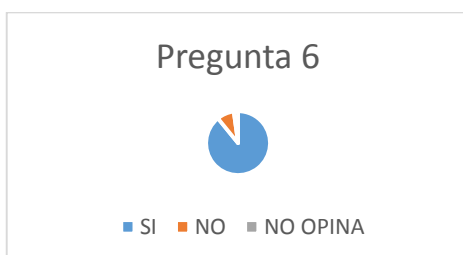
Interpretación: En el grafico 5.3 se muestra que el manejo del sistema de votación no es tan fácil de manejarlo por lo que un 26% de estudiantes no manejo a perfección este sistema de votación, pero un 56% si le pareció fácil el manejo.

Tabla 12.3 Pregunta 6. Le parece interesante la implementación de este servidor de aplicaciones y contenidos como parte de la infraestructura de red de televisión digital en la FIE?

SI	41	89,13%
NO	4	8,70%
NO OPINA	1	2,17%

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Grafico 7. 3 Pregunta 6



Fuente: Inca, Byron., Sánchez, Verónica. 2016

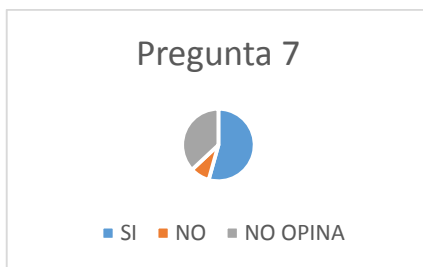
Interpretación: En la el grafico 6.3 se visualiza que la mayoría de estudiantes voto SI para la implementación de un servidor de contenidos interactivos con un total de 89%.

Tabla 13.3 Pregunta 7. Estaría Dispuesto a crear nuevas aplicaciones interactivas basadas en ginga NCL.

SI	25	54,35%
NO	4	8,70%
NO OPINA	17	36,96%

Fuente: Inca, Byron., Sánchez, Verónica. 2016

Grafico 8. 3 Pregunta 7



Fuente: Inca, Byron., Sánchez, Verónica. 2016

Interpretación: En el grafico 7.3 se visualiza que un 60 % de estudiantes desean saber programar en NCL para crear nuevas aplicaciones.

Con la terminación de este trabajo se resolvieron 5 preguntas importantes dentro de nuestra implementación:

¿Cuenta la escuela con un servidor Linux que administre contenido interactivo?

En nuestra Facultad no existe un servidor que administre contenido interactivo, por lo que se ha implementado este proyecto para que en un futuro los estudiantes desarrollen nuevas aplicaciones interactivas.



Figura 16.3: Facultad de Informática y Electrónica

Fuente: <http://fie.esPOCH.edu.ec/>

El servidor implementado en Centos será muy útil para la enseñanza de contenidos interactivos en la facultad de Informática y Electrónica. Ya que, el servidor estará a disposición para pruebas de interactividad y conexión. Temporalmente el servidor está alojado en el DTIC para hacer las pruebas de funcionamiento.

¿Qué versión de Linux será factible para que tenga un buen funcionamiento el proyecto?

Este proyecto se puede implementar en Centos 6.4 y Centos 7. Además este proyecto también se puede realizar en Ubuntu.

¿Cómo se realizara la evaluación del desempeño del servidor?

Con el programa wireshark se evaluó el flujo de transporte sobre la capa IP, se interpretó la trama de red involucrada, así como también el direccionamiento y las peticiones de la clase TCPLua.

También se pudo evaluar el desempeño del servidor, analizando el tiempo de respuesta mediante el sistema de votación t-voting ejecutando la aplicación por varios usuarios al mismo tiempo, la disponibilidad del servidor para ver si está activo todo el día, y determinar con estos parámetros si tiene un buen desempeño en la FIE.

¿Cuáles deberían ser los parámetros de requerimiento de Ginga para que se dé una conexión con el servidor?

Parámetros configurados en la clase TCPLua son los necesarios para que haya la conexión y programar en Eclipse NCL para crear las aplicaciones interactivas.

CONCLUSIONES

- Se concluye que es factible realizar un servidor Linux que maneja contenidos interactivos, el cual recibe los datos y los almacena en su base de datos quedando comprobado que se pudo implementar una parte de la infraestructura básica de la red de televisión digital.
- A través de la investigación se pudo conocer que se está emitiendo las señales de televisión digital en Quito, Guayaquil y Cuenca, con lo que progresivamente se da el cambio de la televisión analógica a televisión digital. A finales del 2016 el apagón analógico empezará en algunas ciudades terminando este proceso hasta el 2018 en todo el país.
- Se conoció que existe el Comité Interinstitucional Técnico de Implementación de la Televisión Digital Terrestre CITDT el cual se encarga desde del año 2013 que el estándar ISDBT, se debe tener en toda televisión que sea ensamblada, importada o comercializada dentro del país.
- Se determinó que las herramientas para el servidor en software fueron apache como servidor web, mysql para crear la base de datos, php para la creación de una web dinámica porque este conjunto de software libre tiene integración con base de datos. En el cliente el software que se utilizó fue la máquina virtual de ginga que simula un SET TO BOX físico y el programa eclipse para programar en NCL y LUA.
- Se implementó e instaló en el DTIC con la dirección IP 172.17.103.107 y mediante los datos enviados de la aplicación T voting desde el laboratorio ubicado en el tercer piso de la FIE, se comprobó que el servidor almacena en la base de datos el historial de votos y la fecha en que realizó.
- Se demostró que el desempeño del servidor es bueno porque mediante el análisis de tramas de red, el tiempo de envío y recepción de datos es rápido pero al existir varios usuarios al mismo tiempo haciendo el uso del servidor provoca un retardo en el tiempo de respuesta. Se realizó cálculos dando como resultado la disponibilidad del servidor de 99%.

RECOMENDACIONES

- Antes de empezar a crear programas interactivos se debe tener presente el tipo de contenido que se quiere mostrar, es de vital importancia desarrollar contenido interactivo amigable y agradable para el usuario.
- Se recomienda tener conocimiento de varios lenguajes de programación que son esenciales entre ellos: NCL para correr la aplicación interactiva, Lua para la conexión, PHP para la página web y también HTML. Además, realizar un análisis más profundo sobre LUA como lenguaje de programación porque en el Ecuador existen muy pocos programadores de este lenguaje.
- Tener conocimientos en base de datos y del lenguaje php. Es por eso que se recomienda que los estudiantes de sistemas mejoren la página web del servidor, y los estudiantes de telecomunicaciones para que creen nuevos contenidos interactivos de tv digital como la t – learning o t- comerse.
- El servidor de aplicaciones que se ha implementado constituye un paso importante para el futuro establecimiento de un laboratorio de televisión digital en la FIE, además adicionando el servidor de playout para que se pueda tener completa la infraestructura de red para interactividad hacia los televidentes.

BIBLIOGRAFÍA

1. **BARRIOS DUEÑAS, JOEL** . *Configuración de Servidores con GNU/LINUX*. Sitio de Red: Alcance Libre. Edición Junio 25,2014
[Consulta: 18 Julio 2016].
<http://www.alcancelibre.org/>
2. **CABRERA JARAMILLO, FERNANDO**. *CentOS Minimal* [blog]. *Febrero 2016*.
Disponible en: <https://notasdeit.wordpress.com/2013/05/15/instalando-centos-minimal/>
3. **DÁVILA SACOTO, MIGUEL**. *Diseño de una plataforma de software para televisión digital interactiva de un canal de deportes utilizando ginga-ncl-lua* [en línea].Cuenca: Sacoto [enero 2012].GINGA: middleware, pp 31-33. [23 junio 2016].
Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/1736/13/UPS-CT02324.pdf>
4. **ECUADOR. MINISTERIO DE TELECOMUNICACIONES**. *Tecnologías de la Información y Comunicación* [en línea]. Ecuador: Mintel,2014. [junio 2016].
<http://www.telecomunicaciones.gob.ec/wpcontent/uploads/downloads/2014/07/mes-Telecom-TIC.pdf>
5. **GALABAY PABLO**. *Manejo del software Ginga para el desarrollo de aplicaciones interactivas para televisión digital, basado en el estándar Brasileño ISDB-Tb* [en línea]. Quito: Vivar Freddy.21 de Junio 2012.
[Consulta: 6 de Julio 2016].
Disponible en: <http://dspace.ups.edu.ec/handle/123456789/2228>
6. **GOMES SOARES, L.M. ; JUNQUEIRA BARBOSA, S.**, *Programando en NCL 3.0*, 2da. ed., s.I., Elsevier, 2010, pp. 28, 60.
7. **HORNERO LUQUE, A.**, *Distribuciones Linux para servidores*[blog]. 16th febrero 2010. [10 de junio 2016].
Disponible en: <http://www.LINUXhispano.net/2010/02/16/distribuciones-LINUX-para-servidores/>

8. **INFORMÁTICA-HOY.** *Tipos de licencia de Software* [en línea]. Copyright © 2007-2012. Disponible en: <http://www.informatica-hoy.com.ar/software-libre-gnu/Tipos-de-licencia-de-Software.php>
9. **MOLINA, ENRIQUE.** *Servidor LAMP* [blog]. Ciudad **Elche** Comunidad Valenciana. 2016. Disponible en: <http://dksignmt.com/2015/04/08/que-es-un-servidor-lamp-y-como-instalar-uno/>
10. **OLEAS K. ; LÓPEZ, M.** *Estudio metodológico para el diseño de interfaces entre el pc y el usuario utilizando isdb.tb y middleware ging.* (Tesis pregrado). Escuela Superior Politécnica de Chimborazo. Facultad de Informática y Electrónica Riobamba Ecuador. Pp. 78-86.
11. **ROBAYO ALEXANDRA,** *Televisión Digital Terrestre: Contexto y efectos sobre la publicidad en el Ecuador* [en línea] (Tesis pregrado). Universidad San Francisco de Quito: (Consultado: Mayo 2016). Disponible en: <http://repositorio.usfq.edu.ec/bitstream/23000/2136/1/106548.pdf>
12. **SEN, J.A.,** *Análisis y diseño de sistemas de TDT.*, Traducido del inglés por Paul Lara., 2da. ed., México, McGraw Hill., 2010, pp 547-568.
13. **TELEVISIÓN DIGITAL TERRESTRE.** *Interactividad* [en línea]. Quito: valencia J. , Diciembre de 2014 [junio 2016]. Disponible en: http://televisiondigitalterrestretdt.com/interactividad_tdt.htm
14. **VALENCIA J. BERNAL I.** “*Desarrollo de Aplicaciones Interactivas para TV Digital orientadas a formar a la Población en Desastres Naturales*”. *Revista Politecnica*. Julio 2013, Vol. 32, No. 2, Páginas: 80–86

ANEXOS

ANEXO A. CODIGO NCL

```
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <transitionBase>
      <transition id="tFade" type="fade" dur="0.7s" />
    </transitionBase>

    <regionBase>
      <region id="rgTela" width="100%" height="100%">
        <region id="rgVideo" width="60%" height="80%"
zIndex="0" />
        <region id="rgVideo1" width="30%" height="70%"
top="7%" right="23%" zIndex="0" />
        <region id="rgSim" width="89" height="59" top="90%"
left="10" zIndex="1" />
        <region id="rgNao" width="89" height="58" top="90%"
left="120" zIndex="1" />
        <region id="rmenu1" width="25%" height="50%"
top="10%" right="23%" zIndex="2" />
        <region id="rcontrol" width="30%" height="70%"
top="10%" right="23%" zIndex="2" />
        <region id="rtele" width="30%" height="70%"
top="10%" right="23%" zIndex="2" />
        <region id="rsistemas" width="30%" height="70%"
top="10%" right="23%" zIndex="2" />
        <region id="rdiseno" width="30%" height="70%" top="10%"
right="23%" zIndex="2" />
        <region id="rgLua" width="100%" height="90" left="0"
top="80%" zIndex="1" />
        <region id="rgplan" width="30%" height="70%"
top="10%" right="23%" zIndex="2" />
        <region id="rgvolador" width="30%" height="70%"
top="10%" right="23%" zIndex="2" />
        <region id="rlaberinto" width="30%" height="70%"
top="10%" right="23%" zIndex="2" />
        <region id="rhumanoide" width="30%" height="70%"
top="10%" right="23%" zIndex="2" />
      </region>
    </regionBase>

    <descriptorBase>
      <descriptor id="dVideo" region="rgVideo" />
      <descriptor id="dSim" region="rgSim" transIn="tFade"
transOut="tFade" />
      <descriptor id="dVideo1" region="rgVideo1"/>
      <descriptor id="dmenu1" region="rmenu1"/>
      <descriptor id="dcontrol" region="rcontrol"/>
      <descriptor id="dtele" region="rtele"/>
      <descriptor id="dsistemas" region="rsistemas"/>
      <descriptor id="ddiseno" region="rdiseno"/>
      <descriptor id="dvolador" region="rgvolador"/>
      <descriptor id="dplan" region="rgplan"/>
      <descriptor id="dlaberinto" region="rlaberinto"/>
      <descriptor id="dhumanoide" region="rhumanoide"/>
      <descriptor id="dacuatico" region="rhumanoide"/>
      <descriptor id="dbailarin" region="rhumanoide"/>

      <descriptor id="dNao" region="rgNao" transIn="tFade"
transOut="tFade" />
      <descriptor id="dLua" region="rgLua" focusIndex="luaIdx">
        <descriptorParam name="transparency" value="10%"/>
      </descriptor>
    </descriptorBase>
  </head>
</ncl>
```

```

        <connectorBase>
            <importBase documentURI="ConnectorBase.ncl"
alias="conector"/>

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

            <causalConnector id="onBeginSet">
                <connectorParam name="var" />
                <simpleCondition role="onBegin" />
                <simpleAction role="set" value="$var" />
            </causalConnector>

            <causalConnector id="onKeySelecionSetStop">
                <connectorParam name="keyCode"/>
                <connectorParam name="var"/>

                <simpleCondition role="onSelection" key="$keyCode"/>
                <compoundAction operator="seq">
                    <simpleAction role="set" value="$var" />

                    <simpleAction role="stop" />
                </compoundAction>
            </causalConnector>

            <causalConnector id="onKeySelecionDelayedStop">
                <connectorParam name="keyCode"/>

                <simpleCondition role="onSelection" key="$keyCode"/>
                <simpleAction role="stop" delay="1s" />
            </causalConnector>

            <causalConnector id="onKeySelecionStop">
                <connectorParam name="keyCode"/>

                <simpleCondition role="onSelection" key="$keyCode"/>
                <simpleAction role="stop" />
            </causalConnector>

            <causalConnector id="onEndAttributionDelayedStop">
                <simpleCondition role="onEndAttribution"/>
                <simpleAction role="stop" delay="10s"/>
            </causalConnector>
        </connectorBase>
    </head>

    <body>
        <port id="pInicio" component="video"/>

        <media type="application/x-ginga-settings" id="programSettings">
            <property name="service.currentKeyMaster" value="luaIdx"/>
        </media>

        <media id="sim" src="media/sim.png" descriptor="dSim" />
        <media id="nao" src="media/nao.png" descriptor="dNao" />
        <media id="info" src="media/info.png" descriptor="dVideo1"/>

        <media id="menu1" src="media/menu1.png" descriptor="dmenu1"/>

        <media id="control" src="media/control.png"
descriptor="dcontrol"/>
        <media id="tele" src="media/tele.png" descriptor="dtele"/>
    </body>

```

```

        <media id="sistemas" src="media/sistemas.png"
descriptor="dacuatico"/>

        <media id="diseno1" src="media/dise.png" descriptor="ddiseno"/>
        <media id="volador" src="media/volador.png"
descriptor="dvolador"/>
        <media id="acuatico" src="media/acautico.png"
descriptor="dvolador"/>
        <media id="bailarin" src="media/bailarin.png"
descriptor="dbailarin"/>
        <media id="laberinto" src="media/laberinto.png"
descriptor="dlaberinto"/>
        <media id="zumo" src="media/zumo.png" descriptor="dplan"/>
        <media id="humanoide" src="media/humanoide.png"
descriptor="dhumanoide"/>
        <media id="lua" src="votacao.lua" descriptor="dLua">
            <property name="voto"/>
            <property name="result"/>
        </media>
        <media id="video" src="media/fie.mp4" descriptor="dVideo"/>
        <link xconnector="onBeginStartN">
            <bind component="video" role="onBegin" />
            <bind component="info" role="start" />
        </link>
        <link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="info">
<bindParam name="keyCode" value="YELLOW"/>
</bind>
            <bind role="start" component="sim"/>
            <bind role="start" component="nao"/>
            <bind role="start" component="lua"/>
            <bind role="stop" component="menu1"/>
        </link>
        <link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="info">
            <bindParam name="keyCode" value="MENU"/>
        </bind>
            <bind role="start" component="menu1"/>
            <bind role="stop" component="info"/>
        </link>
        <link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="info">
            <bindParam name="keyCode" value="BLUE"/>
        </bind>
            <bind role="start" component="zumo"/>
            <bind role="stop" component="info"/>
        </link>
        <link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="menu1">
            <bindParam name="keyCode" value="RED"/>
        </bind>
            <bind role="start" component="tele"/>
            <bind role="stop" component="menu1"/>
        </link>
        <link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="menu1">
            <bindParam name="keyCode" value="GREEN"/>
        </bind>
            <bind role="start" component="control"/>
            <bind role="stop" component="menu1"/>
        </link>
        <link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="menu1">
<bindParam name="keyCode" value="YELLOW"/>
</bind>
            <bind role="start" component="diseno1"/>
            <bind role="stop" component="menu1"/>
        </link>

```

```
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="menu1">
  <bindParam name="keyCode" value="BLUE"/>
</bind>
  <bind role="start" component="sistemas"/>
  <bind role="stop" component="menu1"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="menu1">
  <bindParam name="keyCode" value="MENU"/>
</bind>
  <bind role="start" component="info"/>
  <bind role="stop" component="menu1"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="tele">
  <bindParam name="keyCode" value="MENU"/>
</bind>
  <bind role="start" component="menu1"/>
  <bind role="stop" component="tele"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="control">
  <bindParam name="keyCode" value="MENU"/>
</bind>
  <bind role="start" component="menu1"/>
  <bind role="stop" component="control"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="diseno1">
  <bindParam name="keyCode" value="MENU"/>
</bind>
  <bind role="start" component="menu1"/>
  <bind role="stop" component="diseno1"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="sistemas">
  <bindParam name="keyCode" value="MENU"/>
</bind>
  <bind role="start" component="menu1"/>
  <bind role="stop" component="sistemas"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="laberinto">
  <bindParam name="keyCode" value="CURSOR_RIGHT"/>
</bind>
  <bind role="start" component="volador"/>
  <bind role="stop" component="laberinto"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="zumo">
  <bindParam name="keyCode" value="CURSOR_RIGHT"/>
</bind>
  <bind role="start" component="laberinto"/>
  <bind role="stop" component="zumo"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="volador">
  <bindParam name="keyCode" value="CURSOR_RIGHT"/>
</bind>
  <bind role="start" component="humanoide"/>
  <bind role="stop" component="volador"/>
</link>
<link xconnector="conector#onKeySelectionStartNStopN">
<bind role="onSelection" component="humanoide">
  <bindParam name="keyCode" value="CURSOR_RIGHT"/>
</bind>
</link>
```

```

        <bind role="start" component="acuatico"/>
        <bind role="stop" component="humanoide"/>
    </link>
    <link xconnector="conector#onKeySelectionStartNStopN">
    <bind role="onSelection" component="acuatico">
        <bindParam name="keyCode" value="CURSOR_RIGHT"/>
    </bind>
        <bind role="start" component="bailarin"/>
        <bind role="stop" component="acuatico"/>
    </link>
    <link xconnector="conector#onKeySelectionStartNStopN">
    <bind role="onSelection" component="bailarin">
        <bindParam name="keyCode" value="CURSOR_RIGHT"/>
    </bind>
        <bind role="start" component="info"/>
        <bind role="stop" component="bailarin"/>
    </link>
    <link xconnector="conector#onKeySelectionStartNStopN">
    <bind role="onSelection" component="zumo">
        <bindParam name="keyCode" value="INFO"/>
    </bind>
        <bind role="start" component="menu1"/>
        <bind role="stop" component="zumo"/>
    </link>
    <link xconnector="onKeySelecionSetStop">
        <bind component="sim" role="onSelection">
            <bindParam name="keyCode" value="GREEN"/>
        </bind>
        <bind component="lua" role="set" interface="voto">
            <bindParam name="var" value="sim"/>
        </bind>
        <bind component="nao" role="stop"/>
    </link>
    <link xconnector="onKeySelecionDelayedStop">
        <bind component="sim" role="onSelection">
            <bindParam name="keyCode" value="GREEN"/>
        </bind>
        <bind component="sim" role="stop" />
    </link>
    <link xconnector="onKeySelecionSetStop">
        <bind component="nao" role="onSelection">
            <bindParam name="keyCode" value="RED"/>
        </bind>
        <bind component="lua" role="set" interface="voto">
            <bindParam name="var" value="nao"/>
        </bind>
        <bind component="sim" role="stop" />
    </link>
    <link xconnector="onKeySelecionDelayedStop">
        <bind component="nao" role="onSelection">
            <bindParam name="keyCode" value="RED"/>
        </bind>
        <bind component="nao" role="stop" />
    </link>
    <link xconnector="onEndAttributionDelayedStop">
        <bind component="lua" role="onEndAttribution"
interface="result" />
        <bind component="lua" role="stop" />
    </link>
</body>
</ncl>

```

ANEXO B. CODIGO. LUA

```
require 'tcp'
function setLuaPropertie(propName, propValue)
    local evt = {
        class = 'ncl',
        type = 'attribution',
        name = propName,
        value = propValue,
    }
    evt.action = 'start'; event.post(evt)
    evt.action = 'stop'; event.post(evt)
end
function writeText(text)
    canvas:attrColor(0,0,255,0)
    canvas:clear()
    canvas:attrFont("vera", 36)
    canvas:drawText(10,20, text)
    canvas:flush()
end
function writeResult(votos)
    canvas:attrColor(255,255,255,0)
    canvas:clear()
    canvas:attrFont("vera", 36)
    canvas:drawText(10, 5, "SI: "..votos.sim)
    canvas:drawText(10, 30, "NO: "..votos.nao)
    canvas:drawText(10, 55, "URL:"votos.url)
    canvas:flush()
end
function handler (evt)
    if evt.class == 'ncl' and evt.type == 'presentation'
    and evt.action == 'start' then
        writeText("¿ Desea que el canal Gamatv pase a ser un canal estatal ?")
    end
    if evt.class ~= 'ncl' or evt.type ~= 'attribution'
    or evt.action ~= 'start' or evt.name ~= 'voto' then
        return
    end
    local host = "www.servidoringa.com"

    print(evt.name, evt.value)
    tcp.execute(
        function ()
```

```

writeText("Obteniendo resultado. Por favor, espere... ")
tcp.connect(host, 80)
--conecta no servidor
print("Conectado a "..host)

local url = "GET http://www.servidoringa.com/lua_votos_ginga.php?voto="..evt.value.."n"

print("URL: "..url)
tcp.send(url)

local result = tcp.receive("*a")
if result then
    print("Datos de conexion TCP recibidos")
        f = loadstring(result)
            if f then
                f()
                writeResult(votos)

                setLuaPropertie("result", 1)
            end
        else
            print("Error al recibir datos de conexion TCP")
            if evt.error ~= nil then
                result = 'error: ' .. evt.error
            end
        end

    tcp.disconnect()
end
)
end

event.register(handler)

```

ANEXO C. TCP.LUA

```

local _G, coroutine, event, assert, pairs, type, print
    = _G, coroutine, event, assert, pairs, type, print
local s_sub = string.sub
module 'tcp'
local CONNECTIONS = {}
local current = function ()
    return assert(CONNECTIONS[assert(coroutine.running())])
end
local resume = function (co, ...)

```



```

assert(coroutine.status(co) == 'suspended')
assert(coroutine.resume(co, ...))
if coroutine.status(co) == 'dead' then
    CONNECTIONS[co] = nil
end
end
function handler (evt)
    if evt.class ~= 'tcp' then return end
    if evt.type == 'connect' then
        for co, t in pairs(CONNECTIONS) do
            if (t.waiting == 'connect') and
                (t.host == evt.host) and (t.port == evt.port) then
                t.connection = evt.connection
                t.waiting = nil
                resume(co)
                break
            end
        end
    end
    return
end
if evt.type == 'disconnect' then
    for co, t in pairs(CONNECTIONS) do
        if t.waiting and
            (t.connection == evt.connection) then
            t.waiting = nil
            resume(co, nil, 'disconnected')
        end
    end
    return
end

if evt.type == 'data' then
    for co, t in pairs(CONNECTIONS) do
        if (t.waiting == 'data') and
            (t.connection == evt.connection) then
            resume(co, evt.value)
        end
    end
    return
end
end
event.register(handler)
function execute (f, ...)
    resume(coroutine.create(f), ...)
    print("finalizou tcp.execute")

```

```

end
function connect (host, port)
  local t = {
    host = host,
    port = port,
    waiting = 'connect'
  }
  CONNECTIONS[coroutine.running()] = t
  event.post {
    class = 'tcp',
    type = 'connect',
    host = host,
    port = port,
  }
  return coroutine.yield()
end
function disconnect ()
  local t = current()
  event.post {
    class = 'tcp',
    type = 'disconnect',
    connection = assert(t.connection),
  }
end
function send (value)
  local t = current()
  event.post {
    class = 'tcp',
    type = 'data',
    connection = assert(t.connection),
    value = value,
  }
end
function receive (pattern)
  pattern = pattern or "-- TODO: '*l'/number"
  local t = current()
  t.waiting = 'data'
  t.pattern = pattern
  if s_sub(pattern, 1, 2) ~= '*a' then
    return coroutine.yield()
  end
  local all = ""
  while true do
    local ret = coroutine.yield()
    if ret then

```

```

        all = all .. ret
    else
        return all
    end
end
end
end

```

ANEXO D: CÓDIGO PHP

CLASS_MYSQL.PHP

```

// MySQL Class v0.8.1
class MySQL
{
    // Base variables
    public $lastError;    // Holds the last error
    public $lastQuery;    // Holds the last query
    public $result;       // Holds the MySQL query result
    public $records;      // Holds the total number of records returned
    public $affected;     // Holds the total number of records affected
    public $rawResults;   // Holds raw 'arrayed' results
    public $arrayedResult; // Holds an array of the result
    private $hostname;    // MySQL Hostname
    private $username;    // MySQL Username
    private $password;    // MySQL Password
    private $database;    // MySQL Database
    private $databaseLink; // Database Connection Link
    /* * **** */
    * Class Constructor *
    * **** */
    function __construct($database, $username, $password, $hostname = 'localhost', $port = 3306, $persistant = false)
    {
        $this->database = $database;
        $this->username = $username;
        $this->password = $password;
        $this->hostname = $hostname . ':' . $port;
        $this->Connect($persistant);
    }
    /* * **** */
    * Class Destructor *
    * **** */
    function __destruct()
    {
        $this->closeConnection();
    }
    /* * **** */
    * Private Functions *

```

```

* ***** */
// Connects class to database
// $persistant (boolean) - Use persistant connection?
private function Connect($persistant = false)
{
    $this->CloseConnection();
    if ($persistant) {
        $this->databaseLink = mysql_pconnect($this->hostname, $this->username, $this->password);
    } else {
        $this->databaseLink = mysql_connect($this->hostname, $this->username, $this->password);
    }
    if (!$this->databaseLink) {
        $this->lastError = 'Could not connect to server: ' . mysql_error($this->databaseLink);
        return false;
    }
    if (!$this->UseDB()) {
        $this->lastError = 'Could not connect to database: ' . mysql_error($this->databaseLink);
        return false;
    }

    $this->setCharset(); // TODO: remove forced charset find out a specific management
    return true;
}
// Select database to use
private function UseDB()
{
    if (!mysql_select_db($this->database, $this->databaseLink)) {
        $this->lastError = 'Cannot select database: ' . mysql_error($this->databaseLink);
        return false;
    } else {
        return true;
    }
}
// Performs a 'mysql_real_escape_string' on the entire array/string
private function SecureData($data, $types = array())
{
    if (is_array($data)) {
        $i = 0;
        foreach ($data as $key => $val) {
            if (!is_array($data[$key])) {
                $data[$key] = $this->CleanData($data[$key], $types[$i]);
                $data[$key] = mysql_real_escape_string($data[$key], $this->databaseLink);
                $i++;
            }
        }
    }
}

```

```

} else {
    $data = $this->CleanData($data, $types);
    $data = mysql_real_escape_string($data, $this->databaseLink);
}
return $data;
}
// clean the variable with given types
// possible types: none, str, int, float, bool, datetime, ts2dt (given timestamp convert to mysql datetime)
// bonus types: hexcolor, email
private function CleanData($data, $type = "")
{
    switch ($type) {
        case 'none':
            // useless do not reffect just do nothing
            //$data = $data;
            break;
        case 'str':
        case 'string':
            settype($data, 'string');
            break;
        case 'int':
        case 'integer':
            settype($data, 'integer');
            break;
        case 'float':
            settype($data, 'float');
            break;
        case 'bool':
        case 'boolean':
            settype($data, 'boolean');
            break;
        // Y-m-d H:i:s
        // 2014-01-01 12:30:30
        case 'datetime':
            $data = trim($data);
            $data = preg_replace('/^[^d\:- ]/i', "", $data);
            preg_match('/^([\d]{4}-[\d]{2}-[\d]{2} [\d]{2}:[\d]{2}:[\d]{2})$/', $data, $matches);
            $data = $matches[1];
            break;
        case 'ts2dt':
            settype($data, 'integer');
            $data = date('Y-m-d H:i:s', $data);
            break;
        // bonus types
        case 'hexcolor':

```

```

        preg_match('/([0-9abcdef]{6})/i', $data, $matches);
        $data = $matches[1];
        break;
    case 'email':
        $data = filter_var($data, FILTER_VALIDATE_EMAIL);
        break;
    default:
        break;
    }
    return $data;
}
/* * **** */
* Public Functions *
* **** */
// Executes MySQL query
public function executeSQL($query)
{
    $this->lastQuery = $query;
    if ($this->result = mysql_query($query, $this->databaseLink)) {
        if (gettype($this->result) === 'resource') {
            $this->records = @mysql_num_rows($this->result);
        } else {
            $this->records = 0;
        }
        $this->affected = @mysql_affected_rows($this->databaseLink);
        if ($this->records > 0) {
            $this->arrayResults();
            return $this->arrayedResult;
        } else {
            return true;
        }
    } else {
        $this->lastError = mysql_error($this->databaseLink);
        return false;
    }
}
}
public function commit()
{
    return mysql_query("COMMIT", $this->databaseLink);
}
public function rollback()
{
    return mysql_query("ROLLBACK", $this->databaseLink);
}

```

```

public function setCharset($charset = 'UTF8')
{
    return mysql_set_charset($this->SecureData($charset, 'string'), $this->databaseLink);
}

// Adds a record to the database based on the array key names
public function insert($table, $vars, $exclude = "", $datatypes = array())
{
    // Catch Exclusions
    if ($exclude == "") {
        $exclude = array();
    }
    array_push($exclude, 'MAX_FILE_SIZE'); // Automatically exclude this one
    // Prepare Variables
    $vars = $this->SecureData($vars, $datatypes);
    $query = "INSERT INTO `{ $table }` SET ";
    foreach ($vars as $key => $value) {
        if (in_array($key, $exclude)) {
            continue;
        }
        $query .= "`{$key}` = '{$value}', ";
    }
    $query = trim($query, ', ');
    return $this->executeSQL($query);
}

// Deletes a record from the database
public function delete($table, $where = "", $limit = "", $like = false, $wheretypes = array())
{
    $query = "DELETE FROM `{ $table }` WHERE ";
    if (is_array($where) && $where != "") {
        // Prepare Variables
        $where = $this->SecureData($where, $wheretypes);
        foreach ($where as $key => $value) {
            if ($like) {
                $query .= "`{$key}` LIKE '%{$value}%' AND ";
            } else {
                $query .= "`{$key}` = '{$value}' AND ";
            }
        }
        $query = substr($query, 0, -5);
    }
    if ($limit != "") {
        $query .= ' LIMIT ' . $limit;
    }
    return $this->executeSQL($query);
}

```

```

// Gets a single row from $from where $where is true
public function select($from, $where = "", $orderBy = "", $limit = "", $like = false, $operand = 'AND', $cols = '*',
$wheretypes = array())
{
    // Catch Exceptions
    if (trim($from) == "") {
        return false;
    }

    $query = "SELECT {$cols} FROM `{$from}` WHERE ";

    if (is_array($where) && $where != "") {
        // Prepare Variables
        $where = $this->SecureData($where, $wheretypes);
        foreach ($where as $key => $value) {
            if ($like) {
                $query .= "`{$key}` LIKE '%{$value}%' {$operand} ";
            } else {
                $query .= "`{$key}` = '{$value}' {$operand} ";
            }
        }
        $query = substr($query, 0, -(strlen($operand) + 2));
    } else {
        $query = substr($query, 0, -6);
    }

    if ($orderBy != "") {
        $query .= ' ORDER BY ' . $orderBy;
    }

    if ($limit != "") {
        $query .= ' LIMIT ' . $limit;
    }

    $result = $this->executeSQL($query);
    if (is_array($result))
        return $result;

    return array();
}

// Updates a record in the database based on WHERE
public function update($table, $set, $where, $exclude = "", $datatypes = array(), $wheretypes = array())
{
    // Catch Exceptions
    if (trim($table) == "" || !is_array($set) || !is_array($where)) {

```



```

        return false;
    }
    if ($exclude == "") {
        $exclude = array();
    }
    array_push($exclude, 'MAX_FILE_SIZE'); // Automatically exclude this one
    $set = $this->SecureData($set, $datatypes);
    $where = $this->SecureData($where, $wheretypes);
    // SET
    $query = "UPDATE `{ $table }` SET ";
    foreach ($set as $key => $value) {
        if (in_array($key, $exclude)) {
            continue;
        }
        $query .= "`{$key}` = '{$value}', ";
    }
    $query = substr($query, 0, -2);
    // WHERE
    $query .= ' WHERE ';
    foreach ($where as $key => $value) {
        $query .= "`{$key}` = '{$value}' AND ";
    }
    $query = substr($query, 0, -5);
    return $this->executeSQL($query);
}
// 'Arrays' a single result
public function arrayResult()
{
    $this->arrayedResult = mysql_fetch_assoc($this->result) or die(mysql_error($this->databaseLink));
    return $this->arrayedResult;
}
// 'Arrays' multiple result
public function arrayResults()
{
    if ($this->records == 1) {
        return $this->arrayResult();
    }
    $this->arrayedResult = array();
    while ($data = mysql_fetch_assoc($this->result)) {
        $this->arrayedResult[] = $data;
    }
    return $this->arrayedResult;
}
// 'Arrays' multiple results with a key
public function arrayResultsWithKey($key = 'id')

```

```

{
    if (isset($this->arrayedResult)) {
        unset($this->arrayedResult);
    }
    $this->arrayedResult = array();
    while ($row = mysql_fetch_assoc($this->result)) {
        foreach ($row as $theKey => $theValue) {
            $this->arrayedResult[$row[$key]][$theKey] = $theValue;
        }
    }
    return $this->arrayedResult;
}
// Returns last insert ID
public function lastInsertID()
{
    return mysql_insert_id($this->databaseLink);
}
// Return number of rows
public function countRows($from, $where = "")
{
    $result = $this->select($from, $where, "", "", false, 'AND', 'count(*)');
    return $result["count(*)"];
}
// Closes the connections
public function closeConnection()
{
    if ($this->databaseLink) {
        // Commit before closing just in case :)
        $this->commit();
        mysql_close($this->databaseLink);
    }
}
}

```

LUA_VOTOS_GINGA.PHP

```
<?php
```

```

// Agrego la clase VotosGinga, la cual gestiona los votos que los usuarios realizan
include './VotosGinga.php';
// Creo el Objeto VotosGinga
$votosGinga = new VotosGinga();
function generaTablaVotos()
{
    $vg = new VotosGinga();
    print $vg->getVotoProgramaSinFormato();
}

```

```

/*    print ( "votos = { \n" );
    print ( "    sim = 3333, \n" );
    print ( "    nao = 4444, \n" );
    print ( "    url = 'www.servidoringa.com' \n" );
    print ( " } \n" );*/
}
function exhibeVotos()
{
    print( "Sim: 5555 <br/>" );
    print( "N&atilde;o: 6666 <br/>" );
}
// Registro Voto
if (isset($_REQUEST["voto"])) {
    $votosGinga->registrarVotoBaseDatos( '1', $_REQUEST["voto"] );
    generaTablaVotos();
}else{
    exhibeVotos();
}
?>

```

VOTOSGINGA.PHP

```

<?php
// Agrego la clase mysql, de acceso a datos
include './class_mysql.php';
class VotosGinga
{
    private $_mysql;
    private $_baseDatos;
    private $_usuario;
    private $_passwd;

    function __construct()
    {
        $this->_baseDatos = 'ginga';
        $this->_usuario = 'root';
        $this->_passwd = 'ginga';

        $this->_mysql = new MySQL( $this->_baseDatos,
                                $this->_usuario,
                                $this->_passwd );
    }
}
/*
Metodo que registra los votos en la BBDD - Ginga

```

```

idPrograma: Clave de Identificación del Programa que el usuario vota
voto: Opcion Si (S) ó No (N)
*/
function registrarVotoBaseDatos( $idPrograma = 1, $voto = 'S' )
{
    $voto = array( "idPrograma"=>$idPrograma ,
        "chrVoto"=>$voto,
        "dtFechaVoto"=>date("Y/m/d") );

    return $this->_mysql->insert( 'votos_programa', $voto );
}

function getVotosPrograma( $idPrograma = 1 )
{
    $colores = array( "#68BC31", "#2091CF", "#AF4E96", "#DA5430", "#FEE074" );
    // Sentencia SQL que Lista informacion de programa y sus votos
    $query = " SELECT t1.idPrograma,
        t2.nombre AS nombrePrograma,
        t1.chrVoto AS voto,
        COUNT(t1.chrVoto) AS sintonia
    FROM votos_programa t1
    JOIN programa t2 ON t1.idPrograma = t2.id
    WHERE t1.idPrograma = '". $idPrograma .' "
    GROUP BY t1.idPrograma,
        t2.nombre,
        t1.chrVoto";

    $dtaVotos = $this->_mysql->executeSQL( $query );
    $programaVoto = "[";
    foreach( $dtaVotos as $key => $voto ){

        $objVoto = new stdClass();
        $objVoto->label = ( strtoupper( $voto["voto"] ) == 'S' )? 'SI'
            : 'NO';

        $objVoto->data = $voto["sintonia"];
        $objVoto->color = ( strtoupper( $voto["voto"] ) == 'S' )? '#68BC31'
            : '#DA5430';

        $programaVoto .= json_encode( $objVoto ) . ',';
    }

    $programaVoto = trim( $programaVoto, "," ) . "]";
    return $programaVoto;
}

function getVotoProgramaSinFormato( $idPrograma = 1 )

```

```

{
// Sentencia SQL que Lista informacion de programa y sus votos
$query = " SELECT CASE  WHEN t1.chrVoto = 's'
                THEN 'sim'
                ELSE 'nao'
            END AS voto,
            COUNT(t1.chrVoto) AS sintonia
        FROM votos_programa t1
        JOIN programa t2 ON t1.idPrograma = t2.id
        WHERE t1.idPrograma = '". $idPrograma .' "
        GROUP BY  t1.idPrograma,
                t2.nombre,
                t1.chrVoto";

    $dataVotos = $this->_mysql->executeSQL( $query );
    $votoLua = 'votos = {';
    foreach( $dataVotos as $key => $voto ){
        $votoLua .= $voto["voto"] . ' = ' . $voto["sintonia"] . ' ,';
    }
    $votoLua .= "url = 'www.servidoringa.com'";
    $votoLua .= ' }';
    return $votoLua;
}
}

```

INDEX.PHP

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
    <meta charset="utf-8" />
    <title>Ginga - Panel de Control</title>
    <meta name="description" content="overview & stats" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0" />
    <!-- bootstrap & fontawesome -->
    <link rel="stylesheet" href="assets/css/bootstrap.min.css" />
    <link rel="stylesheet" href="assets/font-awesome/4.2.0/css/font-awesome.min.css" />
    <!-- page specific plugin styles -->
    <!-- text fonts -->
    <link rel="stylesheet" href="assets/fonts/fonts.googleapis.com.css" />
    <!-- ace styles -->
    <link rel="stylesheet" href="assets/css/ace.min.css" class="ace-main-stylesheet" id="main-ace-style" />

    <!--[if lte IE 9]>
        <link rel="stylesheet" href="assets/css/ace-part2.min.css" class="ace-main-stylesheet" />
    </if>

```

```

<![endif]-->
<!--[if lte IE 9]>
  <link rel="stylesheet" href="assets/css/ace-ie.min.css" />
<![endif]-->
<!-- inline styles related to this page -->
<!-- ace settings handler -->
<script src="assets/js/ace-extra.min.js"></script>
<!-- HTML5shiv and Respond.js for IE8 to support HTML5 elements and media queries -->
<!--[if lte IE 8]>
<script src="assets/js/html5shiv.min.js"></script>
<script src="assets/js/respond.min.js"></script>
<![endif]-->
</head>

<body class="no-skin">
  <div id="navbar" class="navbar navbar-default">
    <script type="text/javascript">
      try {
        ace.settings.check('navbar', 'fixed')
      } catch (e) {
      }
    </script>
    <div class="navbar-container" id="navbar-container">
      <button type="button" class="navbar-toggle menu-toggler pull-left" id="menu-toggler" data-
target="#sidebar">
        <span class="sr-only">Toggle sidebar</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <div class="navbar-header pull-left">
        <a href="index.html" class="navbar-brand">
          <small>
            <i class="fa fa-leaf"></i>
            Ginga
          </small>
        </a>
      </div>
      <div class="navbar-buttons navbar-header pull-right" role="navigation">
        <ul class="nav ace-nav">
          <li class="light-blue">
            <a data-toggle="dropdown" href="#" class="dropdown-toggle">
              
                <small>Saludos</small>

```

```

        </span>
      </a>
    </li>
  </ul>
</div>
</div><!-- /.navbar-container -->
</div>
<div class="main-container" id="main-container">
  <script type="text/javascript">
    try {
      ace.settings.check('main-container', 'fixed')
    } catch (e) {
    }
  </script>
  <div id="sidebar" class="sidebar responsive">
    <script type="text/javascript">
      try {
        ace.settings.check('sidebar', 'fixed')
      } catch (e) {
      }
    </script>
    <div class="sidebar-shortcuts" id="sidebar-shortcuts">
      <div class="sidebar-shortcuts-large" id="sidebar-shortcuts-large">
        <button class="btn btn-success">
          <i class="ace-icon fa fa-signal"></i>
        </button>
      </div>
      <div class="sidebar-shortcuts-mini" id="sidebar-shortcuts-mini">
        <span class="btn btn-success"></span>
        <span class="btn btn-info"></span>
        <span class="btn btn-warning"></span>
        <span class="btn btn-danger"></span>
      </div>
    </div><!-- /.sidebar-shortcuts -->
    <ul class="nav nav-list">
      <li class="active">
        <a href="index.html">
          <i class="menu-icon fa fa-tachometer"></i>
          <span class="menu-text"> Panel de Control </span>
        </a>
        <b class="arrow"></b>
      </li>
    </ul><!-- /.nav-list -->
    <div class="sidebar-toggle sidebar-collapse" id="sidebar-collapse">

```

```

        <i class="ace-icon fa fa-angle-double-left" data-icon1="ace-icon fa fa-angle-double-left" data-
icon2="ace-icon fa fa-angle-double-right"></i>
    </div>
    <script type="text/javascript">
        try {
            ace.settings.check('sidebar', 'collapsed')
        } catch (e) {
        }
    </script>
</div>
<div class="main-content">
    <div class="main-content-inner">
        <div class="breadcrumbs" id="breadcrumbs">
            <script type="text/javascript">
                try {
                    ace.settings.check('breadcrumbs', 'fixed')
                } catch (e) {
                }
            </script>
            <ul class="breadcrumb">
                <li>
                    <i class="ace-icon fa fa-home home-icon"></i>
                    <a href="#">Inicio</a>
                </li>
                <li class="active">Panel de Control</li>
            </ul><!-- /.breadcrumb -->
            <div class="nav-search" id="nav-search">
            </div><!-- /.nav-search -->
        </div>
        <div class="page-content">
            <div class="ace-settings-container" id="ace-settings-container">
                <div class="btn btn-app btn-xs btn-warning ace-settings-btn" id="ace-settings-btn">
                    <i class="ace-icon fa fa-cog bigger-130"></i>
                </div>
                <div class="ace-settings-box clearfix" id="ace-settings-box">
                    <div class="pull-left width-50">
                        <div class="ace-settings-item">
                            <div class="pull-left">
                                <select id="skin-colorpicker" class="hide">
                                    <option data-skin="no-skin" value="#438EB9">#438EB9</option>
                                    <option data-skin="skin-1" value="#222A2D">#222A2D</option>
                                    <option data-skin="skin-2" value="#C6487E">#C6487E</option>
                                    <option data-skin="skin-3" value="#D0D0D0">#D0D0D0</option>
                                </select>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```



```

        <span>&nbsp;Choose Skin</span>
    </div>
    <div class="ace-settings-item">
        <input type="checkbox" class="ace ace-checkbox-2" id="ace-settings-navbar" />
        <label class="lbl" for="ace-settings-navbar"> Fixed Navbar</label>
    </div>
    <div class="ace-settings-item">
        <input type="checkbox" class="ace ace-checkbox-2" id="ace-settings-sidebar" />
        <label class="lbl" for="ace-settings-sidebar"> Fixed Sidebar</label>
    </div>

    <div class="ace-settings-item">
        <input type="checkbox" class="ace ace-checkbox-2" id="ace-settings-breadcrumbs" />
        <label class="lbl" for="ace-settings-breadcrumbs"> Fixed Breadcrumbs</label>
    </div>

    <div class="ace-settings-item">
        <input type="checkbox" class="ace ace-checkbox-2" id="ace-settings-rtl" />
        <label class="lbl" for="ace-settings-rtl"> Right To Left (rtl)</label>
    </div>
    <div class="ace-settings-item">
        <input type="checkbox" class="ace ace-checkbox-2" id="ace-settings-add-container" />
        <label class="lbl" for="ace-settings-add-container">
            Inside
            <b>.container</b>
        </label>
    </div>
</div><!-- /.pull-left -->
<div class="pull-left width-50">
    <div class="ace-settings-item">
        <input type="checkbox" class="ace ace-checkbox-2" id="ace-settings-hover" />
        <label class="lbl" for="ace-settings-hover"> Submenu on Hover</label>
    </div>
    <div class="ace-settings-item">
        <input type="checkbox" class="ace ace-checkbox-2" id="ace-settings-compact" />
        <label class="lbl" for="ace-settings-compact"> Compact Sidebar</label>
    </div>
    <div class="ace-settings-item">
        <input type="checkbox" class="ace ace-checkbox-2" id="ace-settings-highlight" />
        <label class="lbl" for="ace-settings-highlight"> Alt. Active Item</label>
    </div>
</div><!-- /.pull-left -->
</div><!-- /.ace-settings-box -->
</div><!-- /.ace-settings-container -->
<div class="page-header">

```

```

<h1>
  Panel de Control
  <small>
    <i class="ace-icon fa fa-angle-double-right"></i>
    Ginga
  </small>
</h1>
</div><!-- /.page-header -->

<div class="row">
  <div class="col-xs-12">
    <!-- PAGE CONTENT BEGINS -->
    <div class="row">
      <div class="space-6"></div>
      <div class="col-sm-7 infobox-container">
        <div class="infobox infobox-green">
          <div class="infobox-icon">
            <i class="ace-icon fa fa-comments"></i>
          </div>
          <div class="infobox-data">
            <span class="infobox-data-number">32</span>
            <div class="infobox-content">comments + 2 reviews</div>
          </div>
          <div class="stat stat-success">8%</div>
        </div>
        <div class="infobox infobox-blue">
          <div class="infobox-icon">
            <i class="ace-icon fa fa-twitter"></i>
          </div>
          <div class="infobox-data">
            <span class="infobox-data-number">11</span>
            <div class="infobox-content">new followers</div>
          </div>
          <div class="badge badge-success">
            +32%
            <i class="ace-icon fa fa-arrow-up"></i>
          </div>
        </div>
      </div>
      <!-- GRAFICO -->
      <div class="vspace-12-sm"></div>

      <div class="col-sm-5">
        <div class="widget-box">
          <div class="widget-header widget-header-flat widget-header-small">

```

```

        <h5 class="widget-title">
            <i class="ace-icon fa fa-signal"></i>
            Traffic Sources
        </h5>
    </div>
    <div class="widget-body">
        <div class="widget-main">
            <div id="piechart-placeholder"></div>
            <div class="hr hr8 hr-double"></div>
            <div class="clearfix">
                <div class="grid3">
                    <span class="grey">
                        <i class="ace-icon fa fa-facebook-square fa-2x blue"></i>
                        &nbsp;&nbsp;&nbsp; likes
                    </span>
                    <h4 class="bigger pull-right">5555</h4>
                </div>
                <div class="grid3">
                    <span class="grey">
                        <i class="ace-icon fa fa-twitter-square fa-2x purple"></i>
                        &nbsp;&nbsp;&nbsp; tweets
                    </span>
                    <h4 class="bigger pull-right">777</h4>
                </div>
            </div>
        </div><!-- /.widget-main -->
    </div><!-- /.widget-body -->
</div><!-- /.widget-box -->
</div><!-- /.col -->
</div><!-- /.row -->
<!-- PAGE CONTENT ENDS -->
</div><!-- /.col -->
</div><!-- /.row -->
</div><!-- /.page-content -->
</div>
</div><!-- /.main-content -->
<div class="footer">
    <div class="footer-inner">
        <div class="footer-content">
            <span class="bigger-120">
                <span class="blue bolder">ESPOCH</span>
                Escuela Ingenier&iacute;a Electr&oacute;nica &copy; 2015-2016
            </span>
            &nbsp;&nbsp;&nbsp;
            <span class="action-buttons">

```

```
<a href="#">
  <i class="ace-icon fa fa-twitter-square light-blue bigger-150"></i>
</a>
<a href="#">
  <i class="ace-icon fa fa-facebook-square text-primary bigger-150"></i>
</a>
<a href="#">
  <i class="ace-icon fa fa-rss-square orange bigger-150"></i>
</a>
</span>
</div>
</div>
</div>
```

```
<a href="#" id="btn-scroll-up" class="btn-scroll-up btn btn-sm btn-inverse">
  <i class="ace-icon fa fa-angle-double-up icon-only bigger-110"></i>
</a>
</div><!-- /.main-container -->
<!-- basi scripts -->
<!--[if !IE]> -->
<script src="assets/js/jquery.2.1.1.min.js"></script>
<!-- <![endif]-->
<!--[if IE]>
  <script src="assets/js/jquery.1.11.1.min.js"></script>
<![endif]-->
<!--[if !IE]> -->
<script type="text/javascript">
  window.jQuery || document.write("<script src='assets/js/jquery.min.js'>" + "<" + "/script>");
</script>
<!-- <![endif]-->
<!--[if IE]>
  <script type="text/javascript">
  window.jQuery || document.write("<script src='assets/js/jquery1x.min.js'>" + "<" + "/script>");
  </script>
<![endif]-->
<script type="text/javascript">
  if ('ontouchstart' in document.documentElement)
    document.write("<script src='assets/js/jquery.mobile.custom.min.js'>" + "<" + "/script>");
</script>
<script src="assets/js/bootstrap.min.js"></script>
<!-- page specific plugin scripts -->
<!--[if lte IE 8]>
  <script src="assets/js/excanvas.min.js"></script>
<![endif]-->
<script src="assets/js/jquery-ui.custom.min.js"></script>
```

```

<script src="assets/js/jquery.ui.touch-punch.min.js"></script>
<script src="assets/js/jquery.easypiechart.min.js"></script>
<script src="assets/js/jquery.sparkline.min.js"></script>
<script src="assets/js/jquery.flot.min.js"></script>
<script src="assets/js/jquery.flot.pie.min.js"></script>
<script src="assets/js/jquery.flot.resize.min.js"></script>
<!-- ace scripts -->
<script src="assets/js/ace-elements.min.js"></script>
<script src="assets/js/ace.min.js"></script>
<!-- inline scripts related to this page
<script src="assets/js/graficos.js"></script> --
<?php
    // Agrego la clase VotosGinga, la cual gestiona los votos que los usuarios realizan
    include './VotosGinga.php'
    // Creo el Objeto VotosGinga
    $votosGinga = new VotosGinga();
    if (isset($_REQUEST["voto"])) {
        $votosGinga->registrarVotoBaseDatos( '1', $_REQUEST["voto"] );
    } else{
        echo "<script type=\"text/javascript\">
            jQuery(function ($) {
                $('.easy-pie-chart.percentage').each(function () {
                    var box = $(this).closest('.infobox');
                    var barColor = $(this).data('color') || (!box.hasClass('infobox-dark') ? box.css('color') :
'rgba(255,255,255,0.95)');
                    var trackColor = barColor == 'rgba(255,255,255,0.95)' ? 'rgba(255,255,255,0.25)' : '#E2E2E2';
                    var size = parseInt($(this).data('size')) || 50;
                    $(this).easyPieChart({
                        barColor: barColor,
                        trackColor: trackColor,
                        scaleColor: false,
                        lineCap: 'butt',
                        lineWidth: parseInt(size / 10),
                        animate: /msie\s*(8|7|6)/.test(navigator.userAgent.toLowerCase()) ? false : 1000,
                        size: size
                    });
                })
                $('.sparkline').each(function () {
                    var box = $(this).closest('.infobox');
                    var barColor = !box.hasClass('infobox-dark') ? box.css('color') : '#FFF';
                    $(this).sparkline('html',
                    {
                        tagValuesAttribute: 'data-values',
                        type: 'bar',
                        barColor: barColor,

```

```

        chartRangeMin: $(this).data('min') || 0
    });
});
//flot chart resize plugin, somehow manipulates default browser resize event to optimize it!
//but sometimes it brings up errors with normal resize event handlers
$.resize.throttleWindow = false;
var placeholder = $('#piechart-placeholder').css({'width': '90%', 'min-height': '150px'});"
echo "    var data = ". $votosGinga->getVotosPrograma() .";";
echo "    function drawPieChart(placeholder, data, position) {
        $.plot(placeholder, data, {
            series: {
                pie: {
                    show: true,
                    tilt: 0.8,
                    highlight: {
                        opacity: 0.25
                    },
                    stroke: {
                        color: '#fff',
                        width: 2
                    },
                    startAngle: 2
                }
            },
            legend: {
                show: true,
                position: position || "\"ne\"",
                labelBoxBorderColor: null,
                margin: [-30, 15]
            }
        },
        grid: {
            hoverable: true,
            clickable: true
        }
    })
}

```

```
drawPieChart(placeholder, data)";
```

```
echo "    /**
```

we saved the drawing function and the data to redraw with different position later when switching to RTL mode dynamically

```
so that's not needed actually.
```

```
*/
```

```

// placeholder.data('chart', data);
// placeholder.data('draw', drawPieChart);
//pie chart tooltip example
var tooltip = $("<div class='tooltip top in'><div class='tooltip-
inner'></div></div>").hide().appendTo('body');
var previousPoint = null;
placeholder.on('plothover', function (event, pos, item) {
    if (item) {
        if (previousPoint != item.seriesIndex) {
            previousPoint = item.seriesIndex;
            var tip = item.series['label'] + \" : \" + item.series['percent'] + '%';
            tooltip.show().children(0).text(tip);
        }
        tooltip.css({top: pos.pageY + 10, left: pos.pageX + 10});
    } else {
        tooltip.hide();
        previousPoint = null;
    }
});
////////////////////////////////////
$(document).one('ajaxloadstart.page', function (e) {
    tooltip.remove();
});
var d1 = [];
for (var i = 0; i < Math.PI * 2; i += 0.5) {
    d1.push([i, Math.sin(i)]);
}
var d2 = [];
for (var i = 0; i < Math.PI * 2; i += 0.5) {
    d2.push([i, Math.cos(i)]);
}
var d3 = [];
for (var i = 0; i < Math.PI * 2; i += 0.2) {
    d3.push([i, Math.tan(i)]);
}
});
echo "</script>";
}
?>
</body>
</html>

```

ANEXO . E

ENCUESTA

TEMA: Implementación de un servidor Linux para administrar contenido interactivo basado en Ginga y evaluar el desempeño en la Facultad de Informática y Electrónica

Nombre _____

Escuela _____

Objetivo:

- Conocer la calidad de servicio que brinda el servidor de aplicaciones.
- Conocer cuántos usuarios soporta el sistema de votación
- Conocer la respuesta del servidor en un tiempo específico

Instrucciones:

1.- ingrese a la página 172.17.103.107 o www.servidoringa.com

2.- ponga cuantos votos están. Y apúntelos en la hoja

3.- corra la aplicación y mande su voto. En el video aparece su voto anótelo

4.- vuelva a ingresar a la página 172.17.103.107 y verifique cuantos votos hay

Encierre en un circulo la respuesta correcta según su criterio.

Preguntas:

1. En el servidor de aplicaciones se almacena su voto. Verifique con lo que anoto en el paso 3
 - Si
 - No
 - No sabe/no opina
2. EL servidor muestra la cantidad de votos totales de las dos opciones SI y No en su página web. Verificar si se actualizó ingresando a la página web.
 - Si
 - No
 - No sabe/no opina
3. El servidor dio una respuesta inmediata de los resultados de la votación en un tiempo estimado de 1 minuto
 - Si
 - No
 - No sabe/no opina
4. El tiempo de respuesta de resultados que se muestra en pantalla fue rápido
 - Si
 - No
 - No sabe/no opina

5. Según su criterio le resulto fácil el manejo del sistema de votación T-voting.
 - Si
 - No
 - No sabe/no opina
6. Le parece interesante la implementación de este servidor de aplicaciones y contenidos como parte de la infraestructura de red de televisión digital en la FIE.
 - Si
 - No
 - No sabe/no opina
7. Estaría dispuesto a crear nuevas aplicaciones interactivas basadas en Ginga Ncl
 - Si
 - No
 - No sabe/no opina