



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

CARRERA: INGENIERÍA EN SISTEMAS

Tesis previa a la obtención del título de:

INGENIERO DE SISTEMAS

TEMA:

**“Desarrollo de software para datos estadísticos de fútbol para tv digital
en el canal XYZ en GINGA_NCL”**

AUTORES:

CARMEN MARÍA OSTAIZA VERA

CARLOS ANDRÉS BARRAGÁN GUALPA

DIRECTOR:

GALO VALVERDE LANDÍVAR

Guayaquil, Mayo del 2015

DECLARACIÓN DE RESPONSABILIDAD

Nosotros **Ostaiza Vera Carmen María** y **Barragán Gualpa Carlos Andrés**, autorizamos a la Universidad Politecnica Salesina la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Carmen María Ostaiza Vera

CC 0922675699

Carlos Andrés Barragán Gualpa

CC 0927257923

DEDICATORIA

En primer lugar a Dios por permitir que logre llegar hasta esta etapa de mi vida profesional y personal.

A mis padres, Manuel Agustín Barragán y Melba Gualpa Méndez, quienes han sabido guiarme por el camino del bien y también me han apoyado toda mi vida con sus consejos y cuidados.

A mis hermanos Rafael Agustín Barragán Gualpa y María Isabel Barragán Gualpa.

A mis profesores por todas sus valiosas enseñanzas y experiencias.

Carlos Andrés Barragán Gualpa

DEDICATORIA

Dedico primeramente mis esfuerzos al Creador del Universo, el Santo Bendito es, porque siempre ha estado a mi lado brindándome la fortaleza que he necesitado, para superar cada uno de los obstáculos que se me han presentado en la vida, a mi familia de manera especial a mi madre Deisy Vera Hidalgo y a mi padre Pedro Ostaiza Cedeño por su amor y apoyo, quienes a pesar de sus limitaciones económicas me enseñaron desde pequeña la importancia de estudiar y superarse en la vida, también dedico este trabajo y esfuerzo a mi esposo Ángel Lombeida Toledo, por estar siempre a mi lado dándome su amor y apoyo incondicional cuando lo necesito, gracias por ayudarme a levantar cuando desfallezco y darme fuerzas para que continúe adelante y no me rinda, a mi hija Ángela Lombeida Ostaiza, aquel regalo del cielo que Dios le dio a mi vida para ser el ancla y fortaleza que muchas veces necesito para seguir adelante y no rendirme, y también a todas aquellas personas que de una u otra manera han influido en mi vida para bien y me han ayudado cuando he necesitado una mano amiga.

Carmen María Ostaiza Vera

AGRADECIMIENTO

El presente proyecto de tesis es un esfuerzo mancomunado y realizado cuidadosamente mediante una revisión continua y exigente por parte de las personas involucradas en el mismo.

Una vez más agradezco a Dios por llegar a esta etapa final de mi aprendizaje universitario.

Y así mismo a todas las personas que estuvieron conmigo y supieron corregirme cuando era necesario, compartiendo conocimientos y facilitando recursos para lograr el objetivo común.

En especial, deseo agradecer al Ing. Jonathan Samaniego y al Ing. Galo Valverde, por su valiosa ayuda y guía en la elaboración de este trabajo de tesis que se convierte en algo innovador y con mucho futuro.

Carlos Andrés Barragán Gualpa

AGRADECIMIENTO

Expresó mi agradecimiento a la Universidad Politécnica Salesiana, por haberme abierto sus puertas, para poder adquirir los conocimientos y destrezas, que me ayudaran a desarrollar dentro del ámbito laboral.

A mis maestros aquellos ingenieros que siempre estuvieron dispuestos a impartir sus conocimientos teóricos y prácticos, con el único afán de que aprendamos a conocer el ámbito laboral con el que nos íbamos a enfrentar.

Al Ing. Jonathan Samaniego, por habernos ayudado en el trabajo de tesis, inicialmente y al Ing. Galo Valverde Landívar que fue nuestro tutor guía, que siempre estuvo dispuesto a enseñarnos, apoyarnos y ayudarnos.

Carmen María Ostaiza Vera

INDICE GENERAL

CAPÍTULO 1	2
DISEÑO DE LA INVESTIGACIÓN	2
1.1. Antecedentes de la Investigación	2
1.1.1. Planteamiento del Problema	3
1.1.2. Formulación del Problema de Investigación	3
1.1.3. Sistematización del Problema de Investigación.....	3
1.2. Objetivos de la Investigación	3
1.2.1. Objetivo General.....	3
1.2.2. Objetivos Específicos	4
1.3. Justificación de la Investigación.....	4
CAPÍTULO 2.....	5
MARCO REFERENCIAL DE LA INVESTIGACIÓN	5
2.1. Marco Teórico	5
2.1.1. Importancia de la Estadística en el Fútbol.....	5
2.1.2. Televisión Digital Terrestre	5
2.1.2.1. Definición.....	5
2.1.2.1.1. Ventajas de Cambio Tecnológico en el País	5
2.1.3. Televisión Digital Interactiva	6
2.1.3.1. Definición.....	6
2.1.4. Tipos de Interactividad	7
2.1.4.1. Interactividad Local	7
2.1.4.2. Interactividad Completa.....	8
2.1.4.3. Servicios Interactivos	8
2.1.5. Estándares Internacionales de Televisión Digital Terrestre	9
2.1.5.1. Estándar Americano ATSC (Advance Television System Committe) 10	
2.1.5.2. Estándar Japonés ISDB – T (Integrated Service Digital Broadcasting Terrestrial) 10	
2.1.5.3. Estándar SBTVD (Sistema Nipo – Brasileño de Televisión Digital Terrestre) 10	
2.1.5.4. Estándar Europeo DVB – T (Digital Video Broadcasting Terrestrial) 11	
2.1.5.5. Estándar Chino DTMB (Digital Terrestrial Multimedia Broadcasting) 11	
2.1.5.6. Estadísticas de Adopción de Estándares	13
2.1.6. Middleware	13
2.1.6.1. Definición.....	13

2.1.7.	Ginga.....	13
2.1.7.1.	Definición.....	13
2.1.7.1.1.	Ginga NCL	14
2.1.7.1.2.	El Lenguaje NCL.....	16
2.1.7.1.3.	Núcleo Común de Ginga (Comon - Core).....	19
2.1.7.1.4.	Requisitos de Software	21
2.1.8.	Lenguaje de Programación PHP	21
2.1.8.1.	Definición.....	21
2.1.8.2.	Codeigniter	22
2.1.8.3.	Arquitectura MVC Codeigniter.....	23
2.1.9.	Base de Datos.....	24
2.1.9.1.	Definición.....	24
2.1.9.2.	MySQL.....	24
2.1.10.	VillageFlow	24
2.1.10.1.	Definición.....	24
2.1.11.	Set-Top-Box	25
2.1.11.1.	Arquitectura de Set-Top-Box	26
2.1.11.2.	Software de Set – Top – Box	28
2.1.11.3.	Conclusiones de Set - Top – Box	29
2.2.	MARCO CONCEPTUAL.....	30
2.3.	FORMULACIÓN DE HIPÓTESIS	32
2.3.1.	Hipótesis General.....	32
2.3.2.	Hipótesis Particulares	32
2.4.	Matriz Causa – Efecto	32
2.5.	Variables.....	34
2.5.1.	Variables Independientes	34
2.5.2.	Variables Dependientes	34
CAPÍTULO 3		35
MARCO METODOLÓGICO DE LA INVESTIGACIÓN		35
3.1.	Tipo de Estudio	35
3.2.	Método de Investigación	35
3.2.1.	Método de Análisis	35
3.2.2.	Método Experimental	36
3.2.3.	Método Comparativo	36
3.3.	Fuentes y Técnicas para la recolección de la Información.....	37

3.3.1.	La entrevista.....	37
3.3.2.	Observación	37
3.3.3.	Recoleccion de datos y procesamiento de documentación	37
3.3.4.	Procesamiento de la información.....	38
CAPÍTULO 4		39
ANÁLISIS DE SISTEMA		39
4.1.	Resultado de la Investigación.....	39
4.2.	Arquitectura de Aplicación	39
4.3.	Diagramas de la Aplicación	41
4.4.	Diagrama de Clases	42
4.5.	Diagrama de Objetos	43
4.6.	Diagrama de Módulo.....	43
4.7.	Diagrama de Casos de Uso.....	44
4.8.	Diagrama de Escenarios	44
4.9.	Diagrama de Estados	45
4.10.	Diagrama de Interacción.....	46
4.11.	Capa de Datos	47
4.11.1.	Modelo Entidad Relación.....	47
4.12.	Capa de Presentación	48
4.13.	Aplicación Web	54
4.14.	Aplicación Ginga	54
4.15.	Capa de Negocios	55
4.16.	Especificación de Librerías Utilizadas	56
4.16.1.	Codeigniter Versión 2.01	56
4.16.2.	GroceryCrud Version 1.3.1	56
4.17.	Requisitos de Hardware y Software.....	57
4.18.	Pruebas y Resultados	57
CONCLUSIONES Y RECOMENDACIONES.....		60
5.1.	Conclusiones	60
5.2.	Recomendaciones	61
Bibliografía		62
ANEXOS		63
Anexo 1. Código del archivo Login.php		63
Anexo 2. Código del módulo de sincronización de la plataforma web.....		64
Anexo 3. Código del archivo mod_ConexionBD.php.....		67

Anexo 4. Código de la función en la aplicación WebClient	79
Anexo 5. Código de la aplicación interactiva, archivo main.ncl.....	79
Anexo 6. Código del archivo cuadroGoleadores.lua.....	85
Anexo 7. Código del archivo cuadroProxPar.lua	85
Anexo 8. Código del archivo cuadroResultados.lua	86
Anexo 9. Código del archivo cuadroTablaPos.lua.	86
Anexo 10. Diccionario de Datos de Tablas	87
Anexo 11. Manual de Usuario Plataforma Web.....	97

INDICE DE FIGURAS

Figura 2.1 Interactividad	7
Figura 2.2 Interactividad Local	7
Figura 2.3 Canal de Retorno	8
Figura 2.4 Interactividad Completa.....	8
Figura 2.5 Mapa Mundial de Televisión Digital Terrestre	12
Figura 2.6 Arquitectura Ginga	14
Figura 2.7 Subsistema Ginga-NCL.....	15
Figura 2.8 Componentes de Ginga - Common Core.....	19
Figura 2.9 Arquitectura MVC	23
Figura 2.10 Tipos de Archivos en Base de Datos	24
Figura 2.11 Arquitectura VillageFlow	25
Figura 2.12 Arquitectura General de un Set-Top-Box.....	26
Figura 2.13 Arquitectura de hardware de un set-top-box convencional	28
Figura 2.14 Software de Set-Top-Box	28
Figura 4.1 Diagrama de Arquitectura.....	39
Figura 4.2 Diagrama de bloques de la plataforma web.....	41
Figura 4.3 Diagrama de Clases de Aplicación Web	42
Figura 4.4 Diagrama de Objetos de la Aplicación Web.....	43
Figura 4.5 Diagrama de Módulos	43
Figura 4.6 Diagrama de Casos de Uso	44
Figura 4.7 Diagrama de Escenarios de la Aplicación Web.....	45
Figura 4.8 Diagrama de Estado Aplicación Web.....	45
Figura 4.9 Diagrama de Estado Aplicación Interactiva	46
Figura 4.10 Diagrama de Interacción Aplicación web	46
Figura 4.11 Entidad Relación de tablas de aplicación web.....	47
Figura 4.12 Módulo de Equipos.....	48
Figura 4.13 Módulo de Jugadores	49
Figura 4.14 Módulo de Usuarios.....	49
Figura 4.15 Módulo de Partidos.....	50
Figura 4.16 Módulo de Resultados	50
Figura 4.17 Módulo Tabla de Posiciones.....	51
Figura 4.18 Módulo de Calendario	51

Figura 4.19 Módulo de Goles y Tarjeta	52
Figura 4.20 Módulo de Configuraciones	52
Figura 4.21 Módulo de Sincronización	53
Figura 4.22 Módulo de Sincronización	53
Figura 4.23 Módulo acerca de.....	53
Figura 4.24 Diagrama Interfaces de Aplicación Web	54
Figura 4.25 Interfaces de Aplicación Ginga	54
Figura 4.26 Estructura de MVC de la Aplicación Web	55
Figura 4.27 Interfaz Ginga Menú Principal	57
Figura 4.28 Interfaz Ginga Tabla de Posiciones	58
Figura 4.29 Interfaz Ginga Tabla de Resultados.....	58
Figura 4.30 Interfaz Ginga Próximos Partidos.....	59
Figura 4.31 Interfaz Ginga Goleadores	59

INDICE DE TABLAS

Tabla 2.1 Resumen Estadístico de los Estándares de Televisión Digital Terrestre ... 13

RESUMEN

La televisión digital terrestre representa una nueva e innovadora forma de disfrutar y percibir contenido audiovisual y multimedia. Desde los inicios, la televisión ha sido un poderoso medio para difundir información y entretenimiento a las personas y ha permitido ser testigos de innumerables acontecimientos de la historia de la humanidad.

El 26 de Marzo del 2010, Ecuador adoptó oficialmente el estándar japonés-brasileño (ISDB-Tb) para la Televisión Digital Terrestre. Este estándar tiene sus orígenes en Japón, país que desarrollo su propio estándar para transmisiones de televisión en formato digital, descontinuando el uso de la tecnología analógica.

La principal ventaja de la Televisión Digital Terrestre es la optimización del uso espectro electromagnético, permitiendo crear nuevos servicios multimedia y audiovisuales, mejorando la calidad de los contenidos y lo más importante, otorgando interactividad y servicios varios al usuario combinado con una programación diversa y aplicaciones interactivas.

Por estos motivos, en el presente proyecto de tesis se ha propuesto demostrar la utilidad y características de la TDT mediante la creación de un prototipo de aplicación interactiva que permitirá al usuario visualizar información del campeonato ecuatoriano de futbol en cualquier momento simplemente utilizando el control remoto.

ABSTRACT

Digital terrestrial television represents a new and innovative way to enjoy and feel audiovisual and multimedia content. From the beginning, television has been a powerful medium for disseminating information and entertainment to people and allowed us to witness many events in the history of mankind.

On March 26, 2010, Ecuador officially adopted the Japanese-Brazilian standard for Digital Terrestrial Television. This standard has its origins in Japan, a country that developed its own standard for television broadcasting in digital format, discontinuing use of analog technology.

The main advantage of the Digital Terrestrial Television is the optimization of the electromagnetic spectrum use, allowing you to create new multimedia and audiovisual services, improving the quality of content and most importantly, providing interactivity and user several services combined with diverse programming and interactive applications.

For these reasons, in this thesis project we the authors have proposed to demonstrate the utility and characteristics of DTT by creating a prototype interactive application that will allow the user to view information of the Ecuadorian soccer championship at any time by using the remote control.

INTRODUCCIÓN

La televisión brinda mucho entretenimiento e información cada día. A pesar de los avances tecnológicos actuales, este medio sigue siendo una de las formas de entretenimiento más comunes en los hogares de los ecuatorianos, por este motivo sigue evolucionando con el objetivo de cumplir con necesidades y llenar expectativas. Como resultado de esto llega al Ecuador la Televisión Digital Terrestre (TDT).

Este trabajo de tesis pretende dar a conocer el uso que se puede dar a la nueva tecnología a implementarse en el país y mostrar los beneficios que esta brinda.

La propuesta de este prototipo de software ofrecerá al televidente la oportunidad de interactuar por medio del control remoto en la comodidad de su hogar y obtener información en tiempo real del campeonato nacional de fútbol.

El planteamiento y desarrollo de este proyecto de tesis está distribuido en 5 capítulos, el contenido de cada uno de estos se detalla a continuación.

Capítulo 1 presenta el planteamiento de la investigación, los objetivos, el problema, justificación y alcance.

Capítulo 2 en este se desarrollan los conceptos teóricos relacionados con TDT los cuales ayudarán a entender y desarrollar mejor el tema planteado.

Capítulo 3 se expone el tipo de estudio que se va a realizar, las metodologías que se utilizarán para lograr el cumplimiento de los objetivos, fuentes y técnicas empleadas para obtener información.

Capítulo 4 muestra el desarrollo del software prototipo propuesto, el cual permite lograr la interactividad del usuario y mostrarle en tiempo real los resultados o estadísticas del campeonato de fútbol ecuatoriano, además presenta las pruebas, con las cuales se demuestra el correcto funcionamiento.

Capítulo 5 aquí se detallan todas las conclusiones y recomendaciones basadas en los resultados obtenidos, que pueden servir de fundamento en implementaciones o investigaciones futuras de Televisión Digital.

CAPÍTULO 1

DISEÑO DE LA INVESTIGACIÓN

1.1. Antecedentes de la Investigación

La Televisión Digital Terrestre (TDT) es una forma diferente de transmisión de la señal televisiva, tomando como referencia la televisión analógica actual. Poco a poco, en el país cambiará la manera de transmitir de señal analógica a digital para esto se creará un plan para la migración con el cual se definirá el tiempo en que se apagarán definitivamente los transmisores analógicos.

Uno de los beneficios que brinda la TV digital es un mejor aprovechamiento del espectro radioeléctrico debido a las técnicas de transmisión digitales empleadas, que permiten aumentar el número de programas transmitidos. Por otro lado, se elimina la vulnerabilidad a interferencias que tiene la señal analógica con los canales adyacentes.

Para la implementación de televisión digital se requiere menos costos de transmisión, en la fase inicial de la implementación los canales de televisión deberán asumir el costo para poder modernizar sus equipos de transmisión. Además, el uso de la televisión digital implicará costos a los televidentes, debido a que tendrán que comprar televisores integrados o decodificadores.

Esta nueva forma de transmitir la señal televisiva, permite que se puedan brindar nuevos usos y servicios interactivos, la interactividad permite que se aprovechen las características y beneficios que ofrece la tecnología de la televisión digital.

Esta investigación servirá de base o apoyo para que en posteriores estudios y avances en la tecnología de hardware a implementar en el país se pueda lograr tener bidireccionalidad en este tipo de software.

El prototipo desarrollado permitirá mostrar las estadísticas generales del campeonato nacional de fútbol como son:

- Tabla de Posiciones
- Próximos partidos
- Resultados
- Goleadores

1.1.1. Planteamiento del Problema

Actualmente en el país no existen muchas aplicaciones desarrolladas para televisión digital, por este motivo al momento del cambio definitivo de transmisión analógica a digital, los televidentes no encontrarán cambios notorios en la manera de ver televisión, ya que no tendrán ningún nivel de interactividad con esta.

La falta de este tipo de software impide que los usos, beneficios y servicios que brinda esta tecnología no sean perceptibles ante los televidentes, debido a que no ven cambios significativos en la transición analógica a digital.

Además la tecnología no es aprovechada en su totalidad, desperdiciando de esta manera los recursos que esta brinda.

1.1.2. Formulación del Problema de Investigación

¿Cómo lograr brindarles información en tiempo real a los televidentes sobre los acontecimientos del campeonato nacional de fútbol, y agregar interactividad?

1.1.3. Sistematización del Problema de Investigación

¿Qué herramientas y tecnologías se deben utilizar e integrar para lograr por medio del software la interactividad del televidente?

¿Cuál es la arquitectura que se debe utilizar para lograr que el televidente tenga información en tiempo real?

¿Con qué dispositivos móviles se podrá acceder a la aplicación administrativa para ingresar la información?

1.2. Objetivos de la Investigación

1.2.1. Objetivo General

Diseñar e Implementar un software para televisión digital para un canal de televisión, con el fin de brindarles información en tiempo real a los televidentes sobre los

acontecimientos del campeonato nacional de fútbol y agregar interactividad por medio del control remoto cambiando de esta manera la forma de ver televisión.

1.2.2. Objetivos Específicos

Determinar que herramientas y tecnologías se deben utilizar e integrar para lograr por medio del software la interactividad del televidente.

Analizar que arquitectura se debe utilizar para lograr que el televidente tenga información en tiempo real.

Determinar mediante que dispositivos móviles se podrá acceder a la aplicación administrativa para ingresar la información.

1.3. Justificación de la Investigación

La propuesta de desarrollo e implementación del software planteado pretende dar a conocer los usos, beneficios, y servicios que se puede brindar por medio de la tecnología TDT, este software ofrecerá al televidente la oportunidad de interactuar por medio del control remoto en la comodidad de su hogar y obtener información en tiempo real, de todos los acontecimientos del campeonato nacional de fútbol. Permitirá que un usuario administrador acceda desde dispositivos móviles, para actualizar la información de los acontecimientos ocurridos, para que luego esta información sea reflejada a los televidentes.

De esta manera se logrará reflejar cambios notorios al televidente, para que diferencie lo nuevo de lo actual.

Por otra parte, en cuanto a su alcance, este proyecto abrirá nuevos caminos para canales televisivos que presenten situaciones similares a la que aquí se plantea, sirviendo como marco referencial a estas.

Por último, la implementación del proyecto propuesto permitirá profesionalmente poner en manifiesto los conocimientos que adquiridos en la carrera y sentar las bases para nuevos estudios que surjan a partir de la problemática planteada.

CAPÍTULO 2

MARCO REFERENCIAL DE LA INVESTIGACIÓN

2.1. Marco Teórico

2.1.1. Importancia de la Estadística en el Fútbol

La estadística en el fútbol ayuda en la toma de decisiones ya que se puede basar en el análisis y en la interpretación de los datos o resultados obtenidos.

Con el análisis e interpretación del comportamiento de una variable o varias se puede determinar o identificar posibles resultados en el campeonato de fútbol nacional.

2.1.2. Televisión Digital Terrestre

2.1.2.1. Definición

La Televisión Digital Terrestre “TDT” es el resultado de la aplicación de la tecnología digital a la señal de televisión analógica, proceso que permitirá optimizar el espectro radioeléctrico e implementar nuevos servicios audiovisuales e interactivos con una programación diversa, a través de este medio de comunicación. Es una oportunidad que posibilitará el desarrollo de múltiples programas y aplicaciones como: Telegobierno, Telesalud y Teleducación para el buen vivir de la población. (Ministerio de Comunicaciones y Sociedad de la Información, 2014).

2.1.2.1.1. Ventajas de Cambio Tecnológico en el País

Mediante el cambio tecnológico de la televisión analógica hacia la televisión digital se garantizará el derecho a la comunicación, inclusión, cohesión y equidad social a la población en general; así como la universalización del servicio de televisión abierta de manera libre y gratuita.

Se mejorará la calidad del servicio de televisión en cuanto al audio, video, nuevos servicios como interactividad; se promoverá la generación de contenidos de educación, salud y cultura que mejoren la calidad de programación actual fomentando los valores nacionales; se promoverá la generación de fuentes de empleo y capacitación a los diferentes actores que participan en el proceso.

Otra de las ventajas de la Televisión Digital que se quiere aprovechar es el sistema de alertas de emergencia, para lo que se requiere un trabajo conjunto entre el sector público y privado a fin de beneficiar a la población con una información instantánea y

sectorizada de alertas de emergencia, para que se actúe en forma adecuada en el caso de ocurrencia de desastres naturales.

El cambio de la televisión analógica a la televisión digital tiene una connotación especialmente social, puesto que se espera aprovechar las ventajas de la televisión digital como lo es la interactividad como medio para mejorar la calidad de vida de la población, puesto que, a través de la interactividad se prevé que los ciudadanos tengan acceso mediante la programación de la televisión, a servicios sociales como, por ejemplo a la obtención de turnos para atención médica en el IESS, se evaluará la satisfacción del usuario en la prestación de servicios de telecomunicaciones a fin de procurar su mejora en el caso de ser necesario, se dispondrá de acceso inmediato a información de gobierno en línea, entre otras opciones que brinda la tecnología. (Ministerio de Comunicaciones y Sociedad de la Información, 2014)

2.1.3. Televisión Digital Interactiva

2.1.3.1. Definición

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 quiere o no verlos, y cuando verlos.

La interactividad le muestra al espectador la posibilidad de personalizar el contenido que muestra su televisor, bien sea accediendo a información enviada durante el proceso de emisión, pero que solo se hace visible si el espectador lo desea, o bien accediendo a servidores con los que puede intercambiar información. A través de un canal de retorno utilizando el televisor como una interfaz de salida. (López & Oleas, 2014).

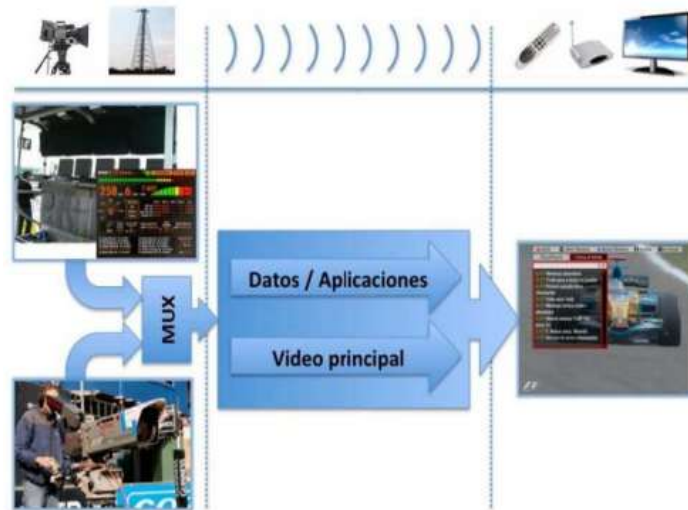


Figura 2.1 Interactividad
Fuente: (López & Oleas, 2014)

2.1.4. Tipos de Interactividad

2.1.4.1. Interactividad Local

El espectador interactúa con la información que está almacenada en el receptor, la cual se renueva con cierta periodicidad.

Con la interactividad local, el usuario puede acceder a contenidos interactivos pero no puede enviar datos de vuelta. Ejemplos de aplicaciones interactivas locales son las guías electrónicas de programación, el teletexto digital, o la información sobre los participantes en un programa.

Únicamente trabaja con aplicaciones transmitidas por Broadcast y descargadas en el Set-top-Box.



Figura 2.2 Interactividad Local
Fuente: (López & Oleas, 2014)

2.1.4.2. Interactividad Completa

El espectador interactúa con un proveedor de servicios exterior, al que se conecta mediante un canal de retorno.

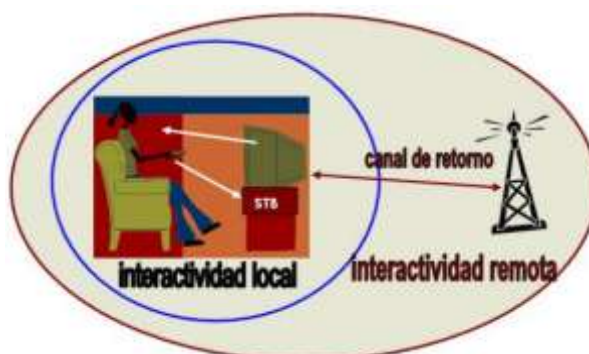


Figura 2.3 Canal de Retorno

Fuente: (López & Oleas, 2014)

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



Figura 2.4 Interactividad Completa

Fuente: (López & Oleas, 2014)

2.1.4.3. Servicios Interactivos

Se pueden establecer tres categorías de **servicios interactivos**:

- **Servicios de información:** Son aquellos que ofrecen una información independiente 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 emitida.
- **Servicios transaccionales:** Son aquellos que ofrecen la posibilidad de enviar y recibir información de forma personalizada y exclusiva.

Los servicios interactivos se implementan con aplicaciones interactivas. Se pueden definir como aquellos programas adicionales a los contenidos de televisión que pueden verse a través de un descodificador interactivo. El usuario es el que decide si quiere o no ver las aplicaciones interactivas mediante una acción simple con el mando a distancia. 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

Las aplicaciones interactivas pueden ser de muy diversos tipos, pero de manera general se pueden dividir en:

- **Servicios Públicos**
 - Servicios de información (estado del tráfico, tiempo, farmacias de guardia, teléfonos de interés, información de aeropuertos, trenes, autobuses, etc.).
 - Servicios avanzados como servicios sanitarios, gestión de impuestos, informaciones y servicios de ayuntamientos y administraciones públicas, servicios sociales a colectivos específicos (tercera edad, etc.).
 - Servicios de entretenimiento o comerciales.
 - Fidelización a contenidos o programas, concursos, votaciones, publicidad interactiva, venta por impulso, compra de eventos (fútbol, cine, conciertos, etc.)
 - Informaciones sobre concursantes, actores, etc., y pueden tanto independientes del contenido, como integradas y sincronizadas con el mismo (concurso interactivo al mismo tiempo que el programa). (López & Oleas, 2014)

2.1.5. Estándares Internacionales de Televisión Digital Terrestre

A continuación se detalla los estándares a ser evaluados para la definición del estándar de Televisión Digital Terrestre (TDT) en el Ecuador.

2.1.5.1. Estándar Americano ATSC (Advance Television System Committe)

El estándar ATSC ha sido diseñado en Estados Unidos para la transmisión de una señal de televisión digital de alta definición (HDTV, High Definition Television), en un ancho de banda de 6 MHz, utilizando codificación de video MPEG-2 (Moving Picture Expert Group). El comité ATSC fue fundado en 1982 con organizaciones privadas de radiodifusores, fabricantes de equipos profesionales y de equipos de consumo, empresas de las industrias de computación, TV por cable, satélite y fabricantes de semiconductores. Entre sus fortalezas se destaca la calidad de la señal en alta definición transmitida para puntos fijos.

El desarrollo del estándar ATSC empezó en el año de 1987 y culminó diez años más tarde en 1997.

Adoptado en: Estados Unidos, Canadá, México, Corea del Sur, Honduras y El Salvador.

2.1.5.2. Estándar Japonés ISDB – T (Integrated Service Digital Broadcasting Terrestrial)

La característica general de este sistema es que divide la banda de frecuencia de un canal en trece segmentos. El radiodifusor puede seleccionar la combinación de segmentos a utilizar.

El sistema ISDB-T, fue desarrollado por ARIB (Association of Radio Industries and Businesses) y adoptado en Japón en 1999, en diciembre de 2003 se puso en marcha en ciudades como Tokyo, Osaka y Nagoya, posteriormente se expandió a otras ciudades en el 2006.

La transmisión a terminales portátiles se la realiza mediante el concepto de recepción parcial de un segmento (“1seg”). Su principal fortaleza está relacionada con la recepción de la señal de televisión en terminales móviles.

El país que ha adoptado este estándar es Japón.

2.1.5.3. Estándar SBTVD (Sistema Nipo – Brasileño de Televisión Digital Terrestre)

El Sistema de Televisión Digital Terrestre Brasileño ha sido definido con base al estándar ISDB-T japonés, utilizando una codificación de video MPEG-4. Es el

resultado de investigaciones y aportes de varios sectores de gobierno, centros de investigación y universidades brasileñas, en acuerdo con el gobierno japonés.

La transmisión para dispositivos móviles es igual al estándar japonés. Entre sus fortalezas destaca la posibilidad de combinar transmisiones de alta definición con las de definición estándar en un mismo canal.

Los países que han adoptado este estándar son Brasil y Perú.

2.1.5.4. Estándar Europeo DVB – T (Digital Video Broadcasting Terrestrial)

Originalmente diseñado para canales de 8 MHz (aplicable también a 7 y 6 MHz), utilizando codificación de video MPEG-2, fue desarrollado con el objeto de optimizar su funcionamiento en cualquiera de los tres entornos de operación presentes en Europa:

- Transmisión en un canal actualmente libre o un canal adyacente
- Transmisión en pequeñas redes SFN (Signal Frequency Network)
- Transmisión en grandes redes SFN

La norma europea DVB se inició en 1993 desarrollada por la industria y las entidades gubernamentales de países principalmente de la Unión Europea. En 1995 finalizó con sus tres formatos DVB-S (Satélite), DVB-C (Cable) y DVB-T (Terrestre). Dentro de sus fortalezas se destaca la multiprogramación que permite ubicar en un mismo canal varias señales de definición estándar SD (Standard Definition).

Está diseñado para redes de frecuencia única y redes de multifrecuencia, y es adoptado en varios países de Europa, Asia, África, Oceanía y América (Guayana Francesa, Uruguay, Colombia y Panamá). (Superintendencia de Telecomunicaciones, 2010)

2.1.5.5. Estándar Chino DTMB (Digital Terrestrial Multimedia Broadcasting)

En el año de 1994 el gobierno chino fundó el grupo de Expertos Ejecutivos Técnicos de Televisión de Alta Definición, conformado por varias universidades e institutos de investigación. Después de tres años de esfuerzo, el grupo desarrollaría la primera televisión de alta definición/prototipo de DTTB (Digital Terrestrial Television Broadcasting).

La norma china fue definida en 2006 y recibió la aprobación final de la República Popular China en Agosto de 2007, comenzando transmisiones en Hong Kong el 31 de Diciembre 2007. DTMB es una fusión de varias tecnologías e incluye derivaciones de la norteamericana ATSC y la europea DVB-T.

Está diseñado para redes de frecuencia única y redes de multifrecuencia. Es un estándar que incluye desde sus inicios soporte para dispositivos móviles, como celulares y reproductores multimedia. Este estándar permite la transmisión bajo compresión MPEG-2 y MPEG-4; y es adoptado en: República Popular China junto a las regiones autónomas de Hong Kong y Macau. (Superintendencia de Telecomunicaciones, 2010)

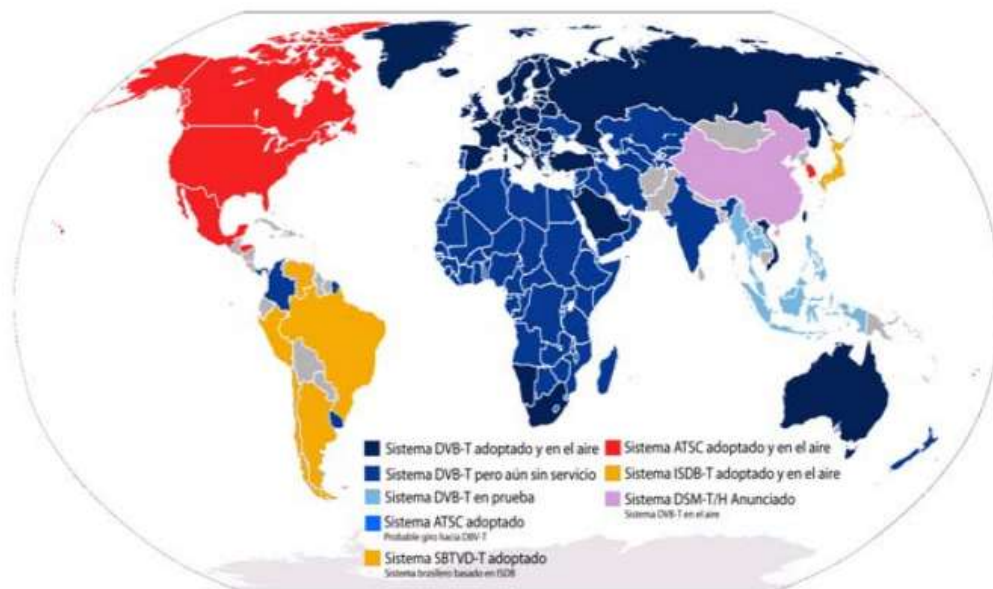


Figura 2.5 Mapa Mundial de Televisión Digital Terrestre

Fuente: (Superintendencia de Telecomunicaciones, 2010)

2.1.5.6. Estadísticas de Adopción de Estándares

	ATSC	ISDB-T/SBTVD	DVB-T	DTMB
Adopción en cantidad de países	6	6	110	1
Países con adopción recomendada	0	0	76	0
Población mundial cubierta	7.50%	4.80%	49.80%	20%
Regiones / Países	América del Norte, Honduras, El Salvador y Corea	Japón, Brasil, Perú, Argentina, Chile y Venezuela	Europa, Oceanía, Panamá, Francesa, Colombia y Uruguay	Asia, África, Guayana y China
Participación en la estandarización	Esencialmente Empresas y Estados de América del Norte	Empresas y Estados de Japón, Brasil, Perú, Argentina, Chile y Venezuela	Coalición de empresas mundiales, coordinadas por el DVB Project (270 miembros)	Empresas y Estado de China

Tabla 2.1 Resumen Estadístico de los Estándares de Televisión Digital Terrestre

Fuente: (Superintendencia de Telecomunicaciones, 2010)

2.1.6. Middleware

2.1.6.1. Definición

Middleware es un conjunto de software ubicado entre el código de las aplicaciones y la infraestructura de ejecución (plataforma de hardware y sistema operativo). 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, 2014)

2.1.7. Ginga

2.1.7.1. Definición

Ginga es el nombre del middleware abierto del Sistema Brasileño de TV Digital (SBTVD). El nombre fue escogido en reconocimiento a la cultura, arte y continua lucha por libertad e igualdad del pueblo brasileño.

Ginga es una capa de software intermedio (middleware), entre el hardware/Sistema Operativo y las aplicaciones, que ofrece una serie de facilidades para el desenvolvimiento de contenidos y aplicaciones para TV Digital, permitiendo la

posibilidad de poder presentar los contenidos en distintos receptores independientemente de la plataforma de hardware del fabricante y el tipo de receptor (TV, celular, PDAs, etc.).

Las aplicaciones ejecutadas sobre Ginga son clasificadas en dos categorías, dependiendo de la forma en la que son escritas. Las Aplicaciones de Procedimiento son escritas usando el lenguaje Java y las Aplicaciones Declarativas son escritas usando el lenguaje NCL.

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



Figura 2.6 Arquitectura Ginga

Fuente: (Comunidad Ginga Ecuador, 2011)

2.1.7.1.1. Ginga NCL

El Ginga-NCL fue desarrollado por la Pontificia Universidad Católica de Rio de Janeiro – PUC- Rio, provee una infraestructura de presentación para aplicaciones declarativas escritas en el lenguaje NCL (Nested Context Language). NCL es una aplicación XML (eXtensible Markup Language) con facilidades para los aspectos de interactividad, sincronismo, espacio-temporal entre objetos de mídia, adaptabilidad, soporte a múltiples dispositivos y soporte a la producción de programas interactivos en vivo no-lineares.

La especificación de este subsistema se base en las normas ABNT NBR 15606-2 y ABNT NBR 15606-5.

Los componentes de este subsistema se muestran en la siguiente Figura.

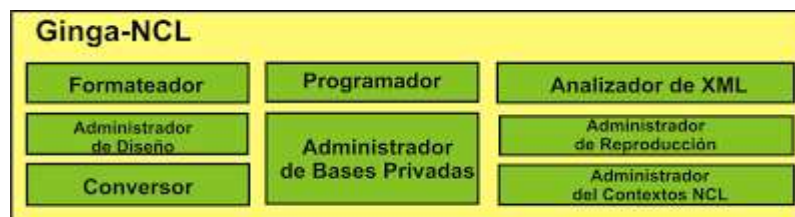


Figura 2.7 Subsistema Ginga-NCL

Fuente: (Comunidad Ginga Ecuador, 2011)

A continuación se define los elementos principales de Ginga-NCL:

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

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

Programador (Scheduler): Es iniciado para organizar el orden de la presentación del documento NCL (antes que inicie los objetos de media, se evalúan las condiciones de los enlaces y la programación correspondiente a las relaciones de las acciones que guiaran el flujo de la presentación).

El componente Programador es responsable para dar la orden al componente Administrador de la Reproducción (Player Manager) para iniciar la reproducción apropiada del tipo de contenido de media para exhibirlo en el momento indicado.

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

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

- 1er Grupo de Comandos, responsable por la activación y desactivación de una base privada, o sea, la habilitación de una determinada aplicación NCL.
- 2do Grupo de Comandos, responsable de iniciar, pausar, resumir, detener, remover las aplicaciones NCL.

- 3er Grupo de Comandos, responsable de la actualización de aplicaciones en tiempo real, permitiendo el agregar o remover elementos NCL y permite que se asignen valores a las propiedades de los objetos de media.

Administrador del Diseño (Layout Manager): El Motor de Presentación soporta múltiples dispositivos de presentaciones a través del componente Administrador del Diseño, el cual es responsable de mapear todas las regiones definidas en una aplicación NCL. (Comunidad Ginga Ecuador, 2011)

2.1.7.1.2. El Lenguaje NCL

NCL es una aplicación XML que permite un acceso por módulos. El acceso por módulos se ha usado en varias recomendaciones de lenguaje W3C. Un módulo es la colección de relaciones semánticas de elementos XML, atributos y valores de atributos que presentan una unidad de funcionalidad.

Los módulos son definidos en conjuntos coherentes. Un perfil del lenguaje es una combinación de módulos, para el estándar Brasileño de TVD, Ginga-NCL define dos perfiles de lenguaje: el Perfil EDTV (Enhanced Digital TV Profile) y el Perfil BDTV (Basic Digital TV Profile).

Aquí se describirán las principales definiciones hechas por Ginga-NCL. La estructura básica de un módulo NCL lo definen el elemento raíz, llamado <ncl>, y sus elementos hijo, el elemento <head> y el elemento <body>, siguiendo la terminología adoptada por otro estándar W3C.

El elemento <head> puede contener los siguientes elementos hijo: <importedDocumentBase>, <ruleBase>, <transitionBase>, <regionBase>, <descriptorBase>, <connectorBase>, <meta> y <metadata>.

El elemento <body> puede contener los elementos hijo <port>, <attribute>, <media>, <context>, <switch> y <link>. El elemento <body> es tratado como un nodo de contexto NCM. NCM, es el modelo conceptual de NCL, donde un nodo puede ser un contexto, un switch o un objeto multimedia. Los nodos de contexto pueden contener otros nodos NCM y enlaces (links). Un nodo switch contiene otros nodos NCM, los nodos NCM están representados por sus correspondiente elementos NCL.

El elemento <media> define un objeto multimedia específico, su tipo y su localización. Otros tipos de elementos <media> son las de tipo “application/x-ginga-settings”, que

especifica un objeto cuyos atributos son variables globales definidas por el documento original o son variables de ambiente reservadas que pueden ser manipuladas por el procesamiento del documento NCL; y las de tipo “application/x-ginga-time”, que especifican un elemento <media> especial cuyo contenido es el Tiempo del Meridiano de Greenwich (GTM).

El elemento <context> es el responsable de la definición de nodos de contexto. Un nodo de contexto NCM es un tipo particular de un nodo compuesto NCM y este define como será contenido un grupo de nodos y un grupo de enlaces como ya se mencionó. Al igual que el elemento <body>, un elemento <context> puede tener elementos hijo <port>, <attribute>, <media>, <context>, <switch>, y <link>.

El elemento <switch> permite la definición de nodos de documentos alternativos (representados por los elementos <media>, <context> y <switch>) para ser escogidos durante su tiempo de presentación. La reglas de prueba utilizadas en la selección del componente switch a ser presentadas se definen por el elemento <rule> o <compositeRule>, los que son agrupados por el elemento <ruleBase>, definido como un elemento hijo del elemento <head>.

Las interfaces NCL funcionalmente permiten la definición de nodos de interface a ser utilizado en relación con otras interfaces. El elemento <area> permite la definición de anclas de contenidos representando porciones espaciales, opciones temporales, u opciones temporales y espaciales de un objeto multimedia (<media>).

El elemento <port> especifica un nodo puerta compuesto (<context>, <body> o <switch>) con su respectivo mapeo a una interface de uno sus componentes hijo. El elemento <attribute> es usado para definir un nodo de atributo o un grupo de nodos de atributo como uno de los nodos de interface. El elemento <switchPort> permite la creación de interfaces de elementos <switch> que son mapeadas para un grupo de interfaces alternativas de nodos de switches internos.

El elemento <descriptor> especifica la información temporal y espacial necesaria para presentar cada componente del documento. El elemento puede referirse a un elemento <region> en el caso que se quiera definir su la posición inicial de la presentación de un elemento <media> en algún dispositivo de salida. Los elementos <descriptor> deben ser definidos dentro del elemento <head> del documento, el elemento <regionBase> define un grupo de elementos <region>, cada uno de los cuales puede

contener anidados otros elementos <region>, y así sucesivamente, las regiones definen las áreas de presentación que se utilizaran en el dispositivo y son referenciadas por los descriptores, como ya se mencionó.

Un elemento <causalConnector> representa una relación que puede ser usada para la creación de elementos <link> en el documento. En una relación causal, una condición debe ser satisfecha para activar una acción. Un elemento <link> enlaza (a través de los elementos <bind>) un nodo de interface con los roles del conector, definiendo una relación espacio-temporal entre los objetos NCL.

El elemento <descriptorSwitch> contiene un grupo de descriptores alternativos para ser asociados con un objeto NCL. Similar al elemento <switch>, un <descriptorSwitch> la selección se realiza durante la presentación del documento, utilizando las reglas de prueba definidas por el elemento <rule> o <compositeRule>.

Con el fin de permitir una entidad base para incorporar otra base ya definida, se puede utilizar el elemento <importBase>. Adicionalmente, un documento NCL puede ser importado a través del elemento <importNCL>. El elemento <importedDocumentBase> especifica un grupo de documento NCL importados, y debe también ser definido como un elemento hijo del elemento <head>.

Algunos atributos de elementos NCL importantes son definidos en otros módulos NCL. El módulo de entidad de reuso (EntityReuse) permite la reutilización de un documento NCL. Este módulo define al atributo refer, el que hace referencia a un elemento URI que puede ser reutilizado. Solo los elementos <media>, <context>, <body> y <switch> pueden ser reutilizados. El módulo de navegación por teclas (KeyNavigation) provee las extensiones necesarias para describir las operaciones de movimientos de foco utilizando un dispositivo de control como el control remoto. Básicamente, este módulo define atributos que pueden ser incorporados por elementos <descriptor>.

Algunas funcionalidades de SMIL son también incorporadas por NCL. El elemento <transition> y algunos atributos de transición son definidos en el módulo de Transiciones Básicas (BasicTransitions) y el módulo de Modificación de Transiciones (TransitionModifiers) del SMIL. El elemento <transitionBase> de NCL especifica un grupo de efectos de transición, definidos por el elemento <transition>, y debe ser definido como un elemento hijo del elemento <head>.

Finalmente, el módulo SMIL de Meta Información (MetaInformation) es también incorporado. Este módulo no contiene información que sea usada o mostrada durante la presentación. En cambio, este contiene información sobre contenidos que son usados o mostrados. El módulo de Meta Información posee dos elementos que permiten describir documentos NCL.

El elemento <meta> especifica un solo par de propiedades o valores. El elemento <metadata> contiene información que es también relacionado a la meta información del documento. Esto actúa como el elemento ruta de un árbol RDF: el elemento RDF y su subsistema de elementos. (Comunidad Ginga Ecuador, 2011).

2.1.7.1.3. Núcleo Común de Ginga (Comon - Core)

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

Está compuesto por los decodificadores de contenido común y por procedimientos para obtener contenidos transportados en flujos de transporte MPEG-2 y a través del canal de interactividad. Los Decodificadores de contenido común sirven tanto a las aplicaciones de procedimiento como a las declarativas que necesiten decodificar y presentar tipos comunes de contenidos como PNG, JPEG, MPEG y otros formatos. El núcleo común de Ginga debe obligatoriamente también ser compatible con el modelo conceptual de exhibición, como se describe en la norma ABNT NBR 15606-1.



Figura 2.8 Componentes de Ginga - Common Core

Fuente: (Comunidad Ginga Ecuador, 2011)

En la Figura anterior, se muestran los componentes básicos del Núcleo Común descritos a continuación:

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

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

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

Persistencia: Ginga es capaz de guardar archivos, incluso después que ha finalizado el proceso que los creó, para que este pueda ser abierto en otra ocasión.

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

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

Administrador de Gráficos: Las normas del middleware definen como se presentan al usuario las imágenes, videos, datos, etc., administrando las presentaciones de la misma manera que está definida en el estándar ARIB [ARIB B-24, 2004].

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

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

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

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

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

Acceso Condicional: Este componente está encargado de restringir contenidos inapropiados recibidos por los canales de programación, proporcionando así seguridad para el middleware. (Comunidad Ginga Ecuador, 2011).

2.1.7.1.4. Requisitos de Software

Los requerimientos necesarios para la instalación de Ginga-NCL son los siguientes:

1. Sistema Operativo Linux
2. Máquina Virtual VMWare
3. Memoria RAM no menor a un 1.5 Gb
4. Espacio de Disco no menor a 8Gb

2.1.8. Lenguaje de Programación PHP

2.1.8.1. Definición

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la web. Dentro de una página Web puede incrustar código PHP que se ejecutará cada vez que se visite una página. El código PHP es interpretado en el servidor Web y genera código HTML y otro contenido que el visitante verá.

PHP fue concebido en 1994 y es fruto del trabajo de un hombre, Rasmus Lerdorf.

Ha sido adoptado por otras personas de talento y ha experimentado cuatro importantes transformaciones hasta convertirse en el producto actual.

PHP es un producto de código abierto, lo que quiere decir que puede acceder a su código. Puede utilizarlo, modificarlo y redistribuirlo sin coste alguno. (Welling & Thomson, 2005, pág. 33).

2.1.8.2. Codeigniter

Codeigniter es un framework para desarrollo de aplicaciones - un conjunto de herramientas para gente que construye sitios web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido que lo que podría hacer si escribiera el código desde cero, proveyéndole un rico conjunto de bibliotecas para tareas comunes, así como una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. Codeigniter le permite enfocarse creativamente en su proyecto al minimizar la cantidad de código necesaria para una tarea dada (Velo, 2012, pág. 6).

Codeigniter es Libre.- Codeigniter está liberado bajo licencias open source del estilo Apache/BSD, así que puede usarlo donde desee.

Codeigniter es Liviano.- Es realmente liviano. El núcleo del sistema sólo requiere algunas bibliotecas muy pequeñas. Esto está en marcado contraste con muchos frameworks que requieren muchos más recursos. Las bibliotecas adicionales se cargan dinámicamente bajo pedido, basado en sus necesidades para un proceso dado, por lo que el sistema base es muy ligero y bastante rápido.

Codeigniter es Rápido

Es verdaderamente rápido. Utiliza un conjunto de librerías con funciones optimizadas, pensadas para un desarrollo rápido y fácil.

Codeigniter Usa M-V-C

Codeigniter usa el enfoque Modelo-Vista-Controlador, que permite una gran separación entre la lógica y la presentación. Es particularmente bueno para proyectos en los que los diseñadores trabajan en sus archivos de plantillas, ya que el código en estos archivos será mínimo (Velo, 2012, pág. 14).

2.1.8.3. Arquitectura MVC Codeigniter



Figura 2.9 Arquitectura MVC

Fuente: (CubeNube, 2011)

Codeigniter está basado en el patrón de desarrollo Modelo-Vista-Controlador. MVC es un enfoque de software que separa la lógica de la aplicación de la presentación. En la práctica, le permite a sus páginas web contener mínimo código ya que la presentación está separada del código PHP.

- **El Modelo** representa sus estructuras de datos. Típicamente sus clases del modelo contendrán funciones que los ayudarán a devolver, insertar y actualizar información de su base de datos.

- **La Vista** es la información que se presenta al usuario. Una vista será normalmente una página web, pero en Codeigniter, una vista también puede ser un fragmento de página como el encabezado o pie de página.

También puede ser una página RSS, o cualquier otro tipo de "página".

- **El Controlador** sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web.

Codeigniter tiene un enfoque bastante poco estricto de MVC, ya que los Modelos no son obligatorios. Si no necesita la separación añadida o encuentra que mantener modelos requiere más complejidad de la que quiere, puede ignorarlos y construir su aplicación mínimamente usando Controladores y Vistas. Codeigniter también le permite incorporar sus propios scripts, o inclusive desarrollar bibliotecas del núcleo para el sistema, permitiéndole trabajar en una forma que tenga más sentido para Ud (Velo, 2012, pág. 18)

2.1.9. Base de Datos

2.1.9.1. Definición

Una base de datos es un contenedor de una colección de datos de diferente tipo como: texto, imagen, sonido, video, etc. Que se encuentran internamente relacionados entre sí a los cuales se puede acceder y consultar en el momento que se requiera.



Figura 2.10 Tipos de Archivos en Base de Datos

Elaborado por: Autores

2.1.9.2. MySQL

MySQL es un sistema de administración de bases de datos relacional (RDBMS) rápido y sólido. Las bases de datos permiten almacenar, buscar, ordenar y recuperar datos de forma eficiente. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, para proporcionar acceso a dichos datos y para asegurarse que solo obtienen acceso a ellos los usuarios con autorización. Por lo tanto, MySQL es un servidor multiusuario y de subprocesamiento múltiple. Utiliza SQL (del inglés Structured Query Language, Lenguaje de consulta estructurado), el lenguaje estándar para la consulta de bases de datos utilizado en todo el mundo.

2.1.10. VillageFlow

2.1.10.1. Definición

VILLAGEFLOW (TM) es la plataforma de software de última generación para, funcionamiento, procesamiento y monitoreo de televisión de difusión digital de señales (Transport Stream). VILLAGEFLOW está optimizado para trabajar en tiempo real, con operación continua 24H / 7D y es compatible con una amplia gama de adaptadores de entrada / salida. (Dektec y otros proveedores de terceros).

Básicamente, se puede adecuar las instalaciones de radiodifusión más baratas, flexibles y altamente funcionales, al tiempo que permite diversas señales de radiodifusión experimental y servicios difíciles.

VILLAGEFLOW cumple con los estándares mundiales, tales como DVB, ISDB-T, DTMB, ATSC, DVB-S / S2, DVB-T / T2, DVB-C / C2, incluyendo módulos para la modulación de RF y demodulación, la emisión de datos, generación de EPG, subtítulo, seguimiento detallado y apoyo para todos los estándares de vídeo que van desde la televisión móvil a los sistemas de alta definición e incluso superior (por ejemplo, 4K). (VILLAGE island, 2014).

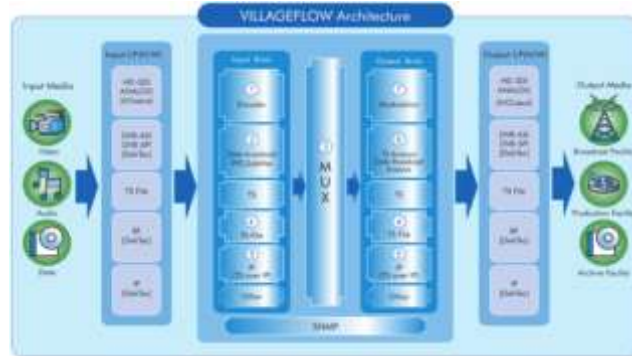


Figura 2.11 Arquitectura VillageFlow

Fuente: (VILLAGE island, 2014)

2.1.11. Set-Top-Box

El Set-Top-Box es un dispositivo externo, el cual permite visualizar que un televisor convencional (Televisión analógica) pueda reproducir programas de televisión emitidos con tecnología digital.

Los componentes físicos que constituyen un Set-Top-Box son:

- Placa del sistema
- Sintonizador
- Modulador/ demodulador
- Demultiplexador
- Decodificador
- Procesador gráfico
- CPU (Central Processing unit)
- Memoria
- Disco
- Interfaces físicas

Así también, se puede clasificar los Set-Top-Box en tres categorías: Broadcast TV, Enhanced TV y Advanced Services. Estas clasificaciones serán descritas a seguir:

a) **Broadcast TV:** Los set-top-box para difusión de TV (broadcast TV) son utilizados para los servicios tradicionales de TV, adicionado de un sistema pay-per-view e instrumentos básicos de navegación. Sin embargo, permiten la recepción de datos en formato digital. También disponen de una cantidad limitada de memoria, limitado número de puertos de interface y una capacidad de procesamiento limitada. Han sido proyectadas de forma que puedan soportar algunos sistemas avanzados, tales como el servicio de mensajes y “video-on-demand”.

b) **Enhanced TV:** La diferencia de los set-top-box de la categoría Enhanced TV consiste en incluir un canal de retorno. Los “Enhanced” set-top-box soportan comercio electrónico, video sobre demanda, y un navegador para 5 conexiones a Internet. La presencia del canal de retorno posibilita comunicaciones por e-mail y por “chat”. Estos set-top-box tienen una capacidad de procesamiento y una capacidad de memoria mayor en relación a los set-top-box “Broadcast TV”.

c) **Advanced Services:** Los set-top-box de servicios avanzados presentan una velocidad de procesamiento cerca de 10 veces superior a los set-top-box “Broadcast TV”.

Las capacidades mejoradas de este tipo de set-top-box en conjunto con el canal de retorno de elevada velocidad, permiten que este tipo de set-top-box tenga acceso a una variedad de servicios de Internet e interactividad. Generalmente, este tipo de set-top-box viene asociado a un disco duro incorporado.

2.1.11.1. Arquitectura de Set-Top-Box

Los set-top-box son sistemas embarcados proyectados para un propósito específico: recepción de señales de TV digital, recuperación del video y exhibición en una TV. En la Figura 2.12 se ilustra la arquitectura en capas de un set-top-box genérico.



Figura 2.12 Arquitectura General de un Set-Top-Box

Fuente: (Villanueva & Velasquez, 2010)

En la capa superior, se tienen los servicios y contenidos que pueden ser producidos en una transmisión de TV digital. Ejemplos de estos servicios son las guías de programación electrónica (EPG), sistema pay-per-view, juegos on-line, programas interactivos, etc.

La segunda capa, se tienen las aplicaciones. Estas aplicaciones son responsables por promover el tipo de servicio de la camada superior.

En la tercera capa, el Middleware, que básicamente tiene como función realizar una interface entre el hardware del set-top-box y las aplicaciones. De esta forma, las aplicaciones pueden ejecutarse de forma transparente sin la preocupación por la forma de acceso al hardware de un set-top-box específico. De esta forma el desarrollo y portabilidad de las aplicaciones se torna más simple, debido a que todas las aplicaciones usan una misma API en común.

En la cuarta capa, se tienen los componentes multimedia de decodificación y codificación, así como los otros módulos multimedia.

En la quinta capa, el sistema operacional, es responsable por el funcionamiento del hardware, proveyendo una camada de abstracción al hardware del set-top-box.

En la última capa se presenta el hardware de un set-top-box, que es constituido por un CPU, dispositivos de entrada y salida, almacenamiento, decodificación, sintonización, etc. En la Figura 2.13, se ilustra la arquitectura del hardware de un set-top-box convencional, el cual puede ser dividido en tres etapas:

- Etapa Inicial (Front End): interfaces de sintonización de las señales.

En esta primera etapa se realiza la recepción de una señal proveniente de tres posibles medios de transmisión: terrestre, cable o satélite. El sintonizador (tuner) selecciona la frecuencia de recepción y modula en banda base la señal de entrada. La señal es muestreada para crear la representación digital del mismo que es realizada por el conversor analógico/digital. Las próximas etapas son la demodulación y la corrección de error de la señal, que luego enseguida es encaminado a la etapa intermediaria.

- Etapa intermedia: interfaces de transporte de control de sistemas.

En la etapa intermediaria se tiene la demultiplexación del flujo MPEG-2, que separa audio, video y datos contenidos en el flujo de transporte. La selección de los paquetes

de audio, video o datos es realizada de acuerdo con las acciones del usuario, que son llevadas a la CPU por las interfaces de entrada y salida.

- Etapa final: decodificadores de audio y video y la etapa de salida de las señales para la TV.

En la etapa final es realizada la decodificación del flujo MPEG-2 que fue seleccionada en la etapa intermedia. Este flujo es convertido nuevamente para una señal analógica, modulada en banda base y enviada para exhibición en un equipo de TV convencional.

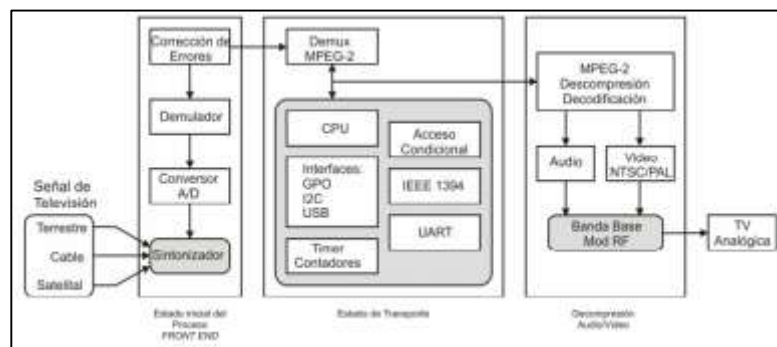


Figura 2.13 Arquitectura de hardware de un set-top-box convencional

Fuentes: (Villanueva & Velasquez, 2010)

2.1.11.2. Software de Set – Top – Box

Desde un punto de vista de hardware, los componentes de la arquitectura del Set-Top-Box que diferencian los diferentes estándares de Televisión Digital (ISDB-T, DBV, ATSC, etc.), son los circuitos integrados demodulador y decodificador de video.



Figura 2.14 Software de Set-Top-Box

Fuente: (Villanueva & Velasquez, 2010)

Para la arquitectura del hardware descrito en la Figura 2.13, existen herramientas de software divididas en varias capas, como se ilustra en la Figura 2.14. Estas herramientas de software hacen uso de las funcionalidades del hardware, para interactuar con el telespectador y entregarlo al contenido televisivo interactivo.

La camada inferior del software está formada por un conjunto de controladores de dispositivos necesarios, para que un sistema operativo de tiempo real ejecutado en la camada adyacente pueda controlar el hardware. A diferencia de los sistemas operativos de computadoras personales, el sistema operativo del set-top-box dispone de recursos más escasos que un micro-computador, como una memoria menor y procesadores con menor velocidad.

En la camada del sistema operativo, se encuentra el Middleware, que es la camada responsable por proporcionar una interface de programación de aplicación (API – Application Programming Interface) a los desarrolladores de aplicaciones.

Esta interface es independiente del hardware y de las tecnologías de comunicación, facilitando el desarrollo de aplicaciones de software.

Entre los estándares de middleware, están el estándar Europeo MHP (Multimedia Home Plataform) de la DVB (Digital Video Broadcasting), el americano DASE (DTV Application Software Environment) de la ATSC (Advanced Television Systems Committee), el estándar ARIB de la ISDB-T de Japón, y el GINGA del estándar brasileño ISDB-Tb.

Finalmente, la camada de aplicación, donde residen los aplicativos producidos que permiten realizar la interactividad con los diversos servicios, entregando así al telespectador el contenido digital en la televisión.

2.1.11.3. Conclusiones de Set - Top – Box

Los Set-Top-Box permiten la decodificación de señales obtenidas a partir de televisión digital terrestre, cable o satelital, para reproducirlo en una televisión analógica.

Los Set-Top-Box se encuentran asociados a un software denominado Middleware, que permite ejecutar aplicaciones interactivas en el set-top-box. Para el caso particular del estándar brasileño, ISDB-Tb es el Middleware Ginga y para el estándar Japonés ISDB-T es el Middleware BML.

Para los diferentes estándares de televisión digital en el mundo (ISDB-T, DBV, ATSC, etc.), las arquitecturas de hardware de los set-top-boxes se diferencian principalmente por el tipo de demodulador y el decodificador MPEG-2/MPEG-4 (Villanueva & Velasquez, 2010).

2.2. MARCO CONCEPTUAL

La televisión digital representa una nueva forma de experimentar y disfrutar contenido audiovisual. Desde sus inicios, ha revolucionado la forma en que las personas acceden a información en tiempo real y a medida que ha avanzado la tecnología esta ha servido justamente para potenciar y agrandar los límites de la misma.

Para efectos del presente proyecto de tesis, esta investigación ha llevado a comprender mejor lo que significa la televisión digital y la manera como esta nueva tecnología puede ayudar a innovar y mejorar la experiencia del usuario.

En el presente trabajo se requiere caracterizar todos aquellos elementos que intervienen en el proceso de la investigación para la comprensión del tema, luego se dará un marco teórico con lo que se encontró con respecto al tema en las referencias bibliográficas, el desarrollo de un prototipo, una integración del marco teórico, y finalmente una conclusión.

Lua. – Lenguaje de programación con sintaxis similar a JavaScript. Es un lenguaje de scripting, procedural, utilizado mayormente como un componente embebido que provee funcionalidad adicional a una aplicación Ginga-NCL.

TDT.- (Televisión Digital Terrestre). Transmisión de imágenes y sonidos en televisión utilizando codificación binaria a través de una red de repetidores terrestres.

Set Top Box.- Es el nombre anglosajón con el que se conoce al receptor digital que se coloca encima o debajo del televisor y que permite visualizar el contenido digital en el mismo.

EPG.- (Electronic Program Guide). Una de las prestaciones de la televisión digital, consiste en presentar al usuario televidente todos los canales organizados de manera rápida y sencilla, además de realizar búsqueda de diversas temáticas: deportes, personajes, series, etc.

MPEG. - (Moving Picture Experts Group). Es una familia de estándares utilizados para codificar información audio-visual, tales como música, videos y películas, en un formato digital comprimido. Una de las ventajas respecto a otros estándares, es que los archivos MPEG son mucho más pequeños pero conservan la misma calidad.

Standard Definition.- es el acrónimo que reciben las señales de televisión que no se pueden considerar señales de alta definición (HDTV) ni de señal de televisión de definición mejorada (EDTV). Es la resolución de vídeo dominante desde el origen de la televisión hasta la aparición de la alta definición.

ARIB. - (Association of Radio Industries and Businesses). Organización de origen japonesa, fue establecida para promover la investigación y desarrollo de nuevos sistemas de radio y servir como una organización de desarrollo de estándares para el avance y la unificación de estándares internacionales y actividades relacionadas en el campo de telecomunicaciones y transmisión (Broadcasting).

Java.- Lenguaje de programación orientada a objetos, basada en la sintaxis del lenguaje C.

XML.- (eXtensible Markup Language). Lenguaje de marcas desarrollado por la World Wide Web Consortium, es un formato utilizados para el almacenamiento y transmisión de datos por internet.

NCL.- (Nested Context Language). Lenguaje de contexto anidado, es un lenguaje de etiquetas similar a HTML desarrollado por la Pontificia Universidad Católica de Rio de Janeiro y ligada al estándar Ginga, con este lenguaje se pueden crear aplicaciones interactivas para la televisión digital terrestre.

PHP.- (HyperText Pre Processor). Lenguaje de programación para crear scripts del lado del servidor, los cuales se interpretan en un servidor web y muestran información en formato HTML en navegador web.

MYSQL.- Es sistema de administración de base de datos relacional, multiusuario, implementada en el modelo cliente-servidor. Alineada con el estándar SQL, el cual sirve como lenguaje para manipulación de datos.

SQL. - (Structured Query Language). Es el lenguaje que se utilizan en las bases de datos relacionales para extraer información de las mismas.

MVC.- (Modelo Vista Controlador). Es un patrón utilizado en el desarrollo de software, el cual consiste en separar en 3 capas los componentes principales de una aplicación, o sea los datos del programa (base de datos), la interfaz gráfica de usuario (vista) y la lógica de eventos de la aplicación (controlador).

ANSI.- (American National Standards Institute). Es una organización norteamericana encargada de supervisar el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos.

Multiplexado.- Proceso en el campo de la electrónica digital mediante el cual se puede conmutar varias señales digitales de entrada en un solo canal de salida.

Canal de retorno.- Medio de comunicación mediante el cual un receptor digital (set top box) es capaz de enviar y recibir datos desde un servidoro cualquier fuente de informacion relacionada a la television. Por lo general se refiere a una conexión a internet.

2.3. FORMULACIÓN DE HIPÓTESIS

2.3.1. Hipótesis General

El desarrollo e implementación de este software para televisión digital permitirá, mostrarle al televidente una manera diferente de ver televisión, debido a la interactividad agregada, que cambiará la forma de ver televisión en los hogares de los ecuatorianos.

2.3.2. Hipótesis Particulares

La falta de investigación y de apertura que se le ha brindado a este tema, ocasiona que no se conozca de qué manera se puede lograr la interactividad del televidente.

La falta de modelos de arquitectura y del conocimiento de las nuevas tecnologías impide que se les muestre a los televidentes información en tiempo real.

Los dispositivos móviles y la web, es lo que más se utiliza actualmente para acceder aplicaciones y brindar información.

2.4. Matriz Causa – Efecto

PROBLEMA GENERAL	OBJETIVO GENERAL	HIPÓTESIS GENERAL
¿Cómo lograr brindarles información en tiempo real a los televidentes sobre los acontecimientos del campeonato nacional de fútbol, y agregar	Diseñar e Implementar un software para televisión digital para un canal de televisión, con el fin de brindarles información en tiempo	El desarrollo e implementación de este software para televisión digital permitirá, mostrarle al televidente una

interactividad?	real a los televidentes sobre los acontecimientos del campeonato nacional de fútbol, y agregar interactividad por medio del control remoto cambiando de esta manera la forma de ver televisión.	manera diferente de ver televisión, debido a la interactividad agregada, que cambiará la forma de ver televisión en los hogares de los ecuatorianos.
PROBLEMAS ESPECÍFICOS	OBJETIVOS ESPECÍFICOS	HIPÓTESIS ESPECÍFICAS
<p>¿Qué herramientas y tecnologías se deben utilizar e integrar para lograr por medio del software la interactividad del televidente?</p> <p>¿Cuál es la arquitectura que se debe utilizar para lograr que el televidente tenga información en tiempo real?</p> <p>¿Con qué dispositivos móviles se podrá acceder a la aplicación administrativa para ingresar la información?</p>	<p>Determinar que herramientas y tecnologías se deben utilizar e integrar para lograr por medio del software la interactividad del televidente.</p> <p>Analizar que arquitectura se debe utilizar para lograr que el televidente tenga información en tiempo real.</p> <p>Determinar mediante que dispositivos móviles se podrá acceder a la</p>	<p>La falta de investigación y de apertura que se le ha brindado a este tema, ocasiona que no se conozca de qué manera se puede lograr la interactividad para el televidente.</p> <p>La falta de modelos de arquitectura y del conocimiento de las nuevas tecnologías impide que se les muestre a los televidentes información en tiempo real.</p>

	aplicación administrativa para ingresar la información.	Los dispositivos móviles y la web, es lo que más se utiliza actualmente para acceder aplicaciones y brindar información.
--	---	---

2.5. Variables

2.5.1. Variables Independientes

Estándar empleado para la comunicación digital: existen diversos estándares a nivel mundial que especifican normas técnicas para la transmisión de contenido digital. En este caso el Ecuador adopto el estándar ISDB-T originario de Japón.

Nivel socio económico del usuario: indica si el usuario está en capacidad adquisitiva suficiente para poder obtener un receptor digital, el cual es necesario para poder visualizar el contenido digital en su televisor, o en su defecto adquirir un nuevo televisor que tenga incorporado un receptor digital en su interior.

2.5.2. Variables Dependientes

Alcance de Señal Digital: el rango de alcance físico en el territorio que puede cubrir la transmisión de dicha señal digital.

Receptores para señal digital: son los set top boxes que traen instalado el middleware GINGA compatible con el estándar ISDB-T para televisión digital.

CAPÍTULO 3

MARCO METODOLÓGICO DE LA INVESTIGACIÓN

3.1. Tipo de Estudio

Con este estudio se desea determinar o establecer los niveles de interactividad que se pueden lograr con esta tecnología.

En segunda instancia este trabajo será explorativo debido a que este es un tema que ha sido poco analizado y desarrollado en el país. Además es un estudio flexible que por medio de un prototipo presentará las características de esta tecnología.

Se realizarán investigaciones en canales televisivos en las cuales se realizan procesos similares a los propuestos, permitiendo comprobar la efectividad para abarcar todas las necesidades requeridas por el sujeto de estudio.

Se describe las consecuencias de no contar con una manera que les brinde a los televidentes la manera de interactuar con los servicios que se ofrecen.

Se mostrará estudios y comentarios de investigaciones encontradas en foros y en el ministerio de telecomunicaciones del país para ver cómo avanza y desarrolla la TDT en el país y en canales televisivos.

3.2. Método de Investigación

Para llevar a cabo la ejecución de este proyecto se utilizará como método de investigación el método de análisis, el método experimental, el método comparativo los cuales sirven para encontrar soluciones a los problemas planteados creando una relación de causa-efecto.

3.2.1. Método de Análisis

Se ejecutará un proceso de conocimiento, comenzando por la identificación de los equipos de transmisión de TV Digital en los cuales van a funcionar la aplicación, reconocimientos de servidores y plataformas de SO, además se va analizar cada uno de los aspectos que tengan relación causa efecto para ver que variables que comprenden el objeto de estudio se pueden ver afectada.

3.2.2. Método Experimental

Desarrollo del prototipo de software, con la arquitectura que permitirá mostrar en tiempo real, los acontecimiento del campeonato nacional de fútbol.

3.2.3. Método Comparativo

La plataforma GINGA es la adecuada en este caso porque es compatible con el estándar ISDB-T para televisión digital, adoptado oficialmente por el Ecuador a través del Ministerio de Telecomunicaciones en el año 2010.

Es compatible porque en un principio GINGA fue desarrollado en Brasil, orientado al sistema de Televisión Digital Terrestre (SBTVD) de ese país y que a su vez está basado en la norma japonesa ISDB-T (Integrated Services Digital Broadcasting. La T es de Terrestre).

Brasil fue el primer país sudamericano en adoptar la norma japonesa y la modificó para adaptarla a su entorno, funcionando a partir del 2007. La norma modificada se llama ISDB-Tb (Built-in) y está abierta a la comunidad bajo licencia GNU. En pocas palabras GINGA es un sistema que funciona como intermediario entre el código de las aplicaciones interactivas y la plataforma de hardware (que sería a efectos prácticos los receptores digitales o set top boxes).

Ventajas:

- ✓ Contenido televisivo de alta definición y calidad
- ✓ Acceso a aplicaciones interactivas, que permiten al usuario realizar tareas como por ejemplo voto por televisión, compras, consultas de información bancaria, información actualizada de las estaciones que transmiten contenido digital, guías de programación electrónica, juegos, etc.
- ✓ Acceso a internet mediante el uso del receptor digital, visualizando la información en el televisor mismo.

Desventajas:

- ✓ Cabe recalcar que la conexión con el proveedor de servicios (interactividad) es posible si el receptor digital tiene disponible un canal de retorno mediante el cual puede comunicarse con dicho proveedor. Por lo general es una conexión a internet.

- ✓ La transición de la tecnología analógica a la transmisión digital aún está en proceso de estructuración, por lo cual el uso de las aplicaciones podría verse muy limitado.
- ✓ Los usuarios necesitarían un receptor digital de excelente calidad, pero hay que tomar en cuenta que no todos los receptores tienen instalado el middleware GINGA, el usuario fácilmente podría adquirir un receptor que no sea el adecuado e incurrir en un gasto innecesario.

3.3. Fuentes y Técnicas para la recolección de la Información

Teniendo en cuenta que es necesario recolectar información indispensable para el desarrollo del presente proyecto de tesis, se recurre a la entrevista, la cual se lleva a cabo con los encargados del departamento de sistemas del canal de televisión a fin de establecer los requerimientos para el diseño de la aplicación interactiva y de la aplicación Web.

3.3.1. La entrevista

Es un recurso importante ya que permite conocer de manera personal y más exacta todas las ideas de parte de los usuarios finales en cuanto al manejo de la aplicación web y la aplicación interactiva. Se recabaron diferentes opiniones que pueden servir para mejorar el rendimiento o cualquier otra característica relevante.

3.3.2. Observación

Mediante este proceso se puede percibir de mejor manera todo lo necesario para poder iniciar el desarrollo de la aplicación interactiva y la aplicación web también. Además se realizaron comparaciones con aplicaciones existentes para verificar funcionalidad requerida y estándar.

3.3.3. Recolección de datos y procesamiento de documentación

Aquí se procede a recabar toda la información necesaria, así como a la depuración de herramientas de recolección de datos, por ejemplo manuales físicos o digitales disponibles en Internet sobre los nuevos lenguajes de programación requeridos para el desarrollo del software, programas utilitarios para el diseño de diagramas y/o esquemas.

3.3.4. Procesamiento de la información

Una vez compilada toda la información, se procede a organizar los datos necesarios para obtener los resultados requeridos. Con estos resultados se procede a crear las respectivas aplicaciones (web e interactiva) para dar solución a los requerimientos planteados anteriormente.

El resultado es la creación de dos tipos de aplicaciones: primero una aplicación web usando el lenguaje PHP, la cual es necesaria para que el usuario pueda dar mantenimiento sobre toda la información del campeonato ecuatoriano de fútbol, es decir, jugadores, partidos, calendarios de partidos, equipos, tabla de posiciones y sincronización de datos.

Segundo es la creación de una aplicación interactiva usando el lenguaje GINGA-NCL, esta aplicación cliente esta destinada para los usuarios que sintonizan en sus casas el canal de televisión, mediante esta aplicación el usuario puede visualizar solamente información de próximos partidos, goleadores, resultados y tabla de posiciones.

CAPÍTULO 4

ANÁLISIS DE SISTEMA

4.1. Resultado de la Investigación

El principal propósito del prototipo desarrollado en el presente proyecto de tesis es poder mostrar el uso de una aplicación interactiva, como debería ser visualmente y el nivel intuitivo que debería tener para una mejor comprensión por parte del usuario.

El alcance de esta aplicación básicamente es mostrar información en tiempo real de los partidos del campeonato ecuatoriano de fútbol.

Para ello se complementa de una aplicación web de tipo servidor, en la cual el usuario de la estación televisora, previamente capacitado para el manejo de la misma, ingresa la información conforme avanzan los partidos, y desde la misma estación utilizando su respectiva infraestructura de hardware y software, envía información actualizada la cual llega a cada receptor.

4.2. Arquitectura de Aplicación



Figura 4.1 Diagrama de Arquitectura

Elaborado por: Autores

La aplicación interactiva desarrollada en Ginga-NCL, tiene una arquitectura cliente-servidor tal como se observa en la figura 4.1, el cual está basado en un modelo declarativo utilizando funciones procedurales escritas en lenguaje Lúa y obtiene la información mediante archivos de texto que son creados y descargados desde la plataforma web que viene a ser el sistema servidor.

La aplicación que realiza la descarga de los archivos de texto al sistema cliente se llama WebClient. Cada cierto tiempo, el cual es configurable para el usuario, realiza la descarga de los archivos desde un directorio llamado data/, que es creado por la plataforma web. Utilizando el programa VillageFlow el usuario puede transmitir la aplicación interactiva, junto con los datos actualizados dentro de la misma aplicación, a través del espacio aéreo mediante la señal digital hasta los receptores digitales de los televidentes.

En la capa de base de datos, se guarda toda la información correspondiente a los partidos, calendario, resultados, jugadores, equipos, goles y tarjetas, tabla de posiciones, parámetros de funcionamiento así como los usuarios registrados para poder ingresar datos al sistema.

La capa de negocio, que viene a ser el “modelo” en el patrón de diseño MVC, proporciona el vínculo entre la base de datos y la capa de presentación de la plataforma web. Esta capa de negocio contiene todas las funciones necesarias para poder mostrar información al usuario así como generar nueva información del campeonato conforme avanzan los partidos de acuerdo a la jornada actual.

A continuación se detallan cada uno de los componentes de la aplicación:

Aplicación Web

La aplicación web es la plataforma desarrollada en lenguaje PHP, cuya finalidad es permitir al usuario el ingreso y actualización de la información de los partidos.

Base de datos

En la base de datos es el repositorio en donde se almacena toda la información sobre equipos, jugadores, usuarios, partidos, resultados, tabla de posiciones y parámetros del sistema junto con los goles y tarjetas.

VillageFlow

Software diseñado para poder realizar Broadcasting de contenidos y programas para televisión digital. Tiene una estructura que permite generar, gestionar y controlar la emisión de señales digitales de manera continúa 24/7.

Set Top Box

Es un receptor o sintonizador para televisión digital, este dispositivo permite visualizar en el televisor los programas y contenidos interactivos que ofrecen las estaciones emisoras.

4.3. Diagramas de la Aplicación

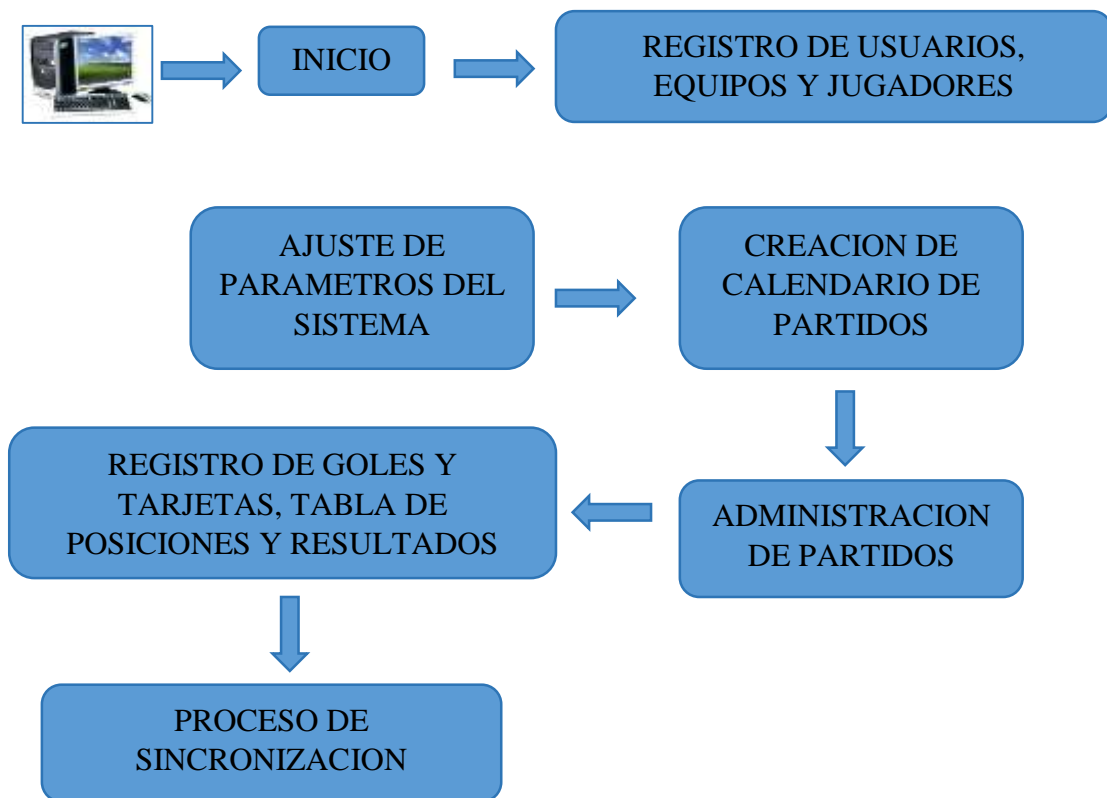


Figura 4.2 Diagrama de bloques de la plataforma web

Elaborado por: Autores

Se puede ver en la figura 4.2 que todo el proceso comienza cuando el usuario inicia sesión en la plataforma web. Procede a registrar los equipos junto con los jugadores así como los usuarios que también podrán acceder al sistema para ingresar información del campeonato. Adicional a esto se configuran los parámetros del sistema, son de importancia ya que se guardan el número de etapas del campeonato, número de fechas o jornadas de todo el campeonato actual así como el número de partidos por jornada.

4.5. Diagrama de Objetos

En la figura 4.4 se muestra el diagrama de objetos muestra una instancia de las clases del proyecto en un momento dado. Aquí hay un ejemplo, se crea una instancia de la clase Equipo, con sus atributos y acciones (métodos) y como está relacionada a su vez con los objetos (instancias) de la clase Jugador. Así mismo se ve la instancia de la clase partido con la instancia de la clase Resultado.



Figura 4.4 Diagrama de Objetos de la Aplicación Web

Elaborado por: Autores

4.6. Diagrama de Módulo

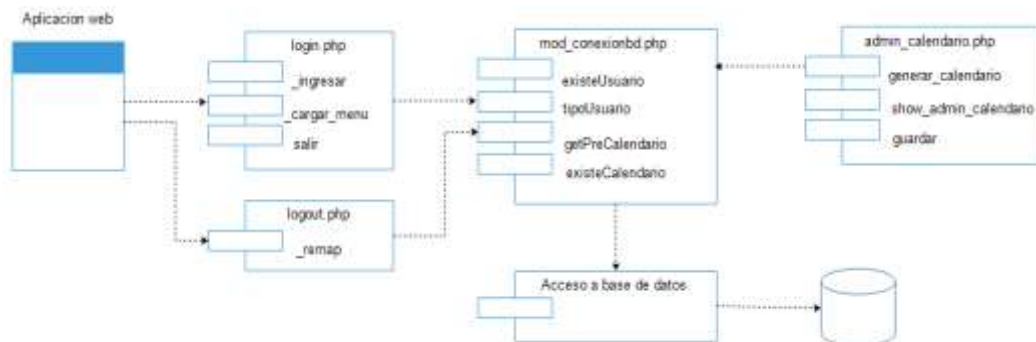


Figura 4.5 Diagrama de Módulos

Elaborado por: Autores

login.php: módulo que se encarga de validar las credenciales del usuario y permitir o denegar el acceso a las opciones del sistema.

logout.php: módulo que cierra la sesión del usuario.

mod_conexion.php: el archivo de capa de negocios, aquí se encuentran todos los métodos que sirven para que la plataforma web extraiga los datos de la capa de datos y muestre dicha información al usuario. Los métodos “existeUsuario” y “tipoUsuario”

se encargan de validar si existe el usuario y comprobar si es un usuario “Administrador” o “Publicador”. Este módulo está conectado a la capa de datos.

admin_calendario.php: archivo controlador que permite al usuario crear el calendario para el campeonato actual. El método “generar_calendario” guarda la información del calendario para el año en curso, “show_admin_calendario” muestra el calendario que ha sido creado por el usuario y el método “guardar” ingresa información de los partidos por fecha relacionados al actual calendario.

4.7. Diagrama de Casos de Uso



Figura 4.6 Diagrama de Casos de Uso

Elaborado por: Autores

En la figura 4.6 se puede visualizar las opciones de la plataforma web, el usuario tiene la opción de registrar nuevos usuarios, generar calendario así como guardar partidos y registrar goles y amonestaciones en los mismos. Por último está la opción de sincronización.

4.8. Diagrama de Escenarios

En la figura 4.7 se observa que el usuario inicia sesión en la plataforma web, realiza el registro y actualización de nuevos partidos así como de datos de equipos, jugadores o usuarios.

También puede registrar goles y amonestaciones, los cuales a su vez actualizan los resultados de los partidos y la tabla de posiciones. Por último está el proceso de sincronización de datos, los cuales son visualizados por el usuario mediante la aplicación interactiva.

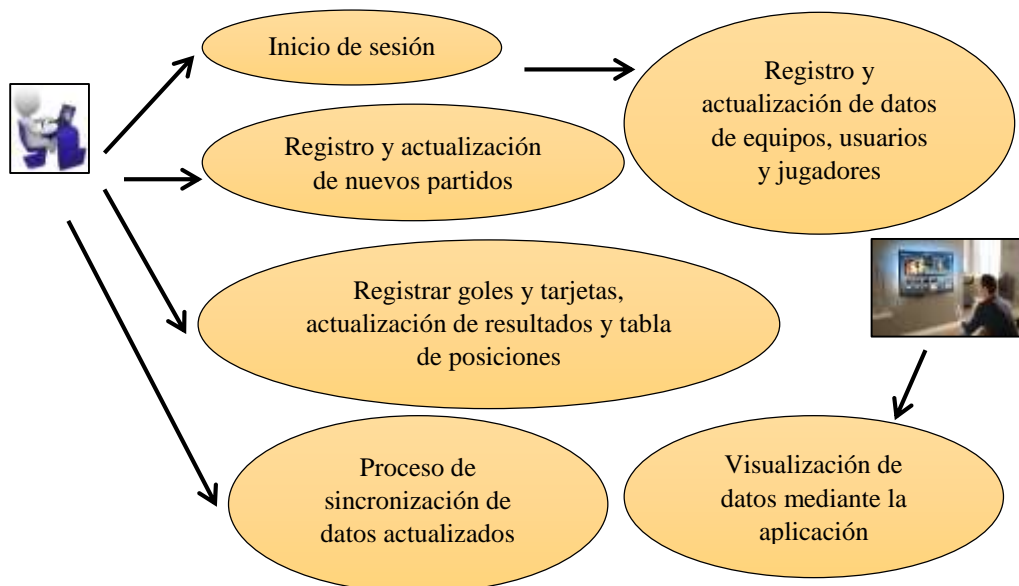


Figura 4.7 Diagrama de Escenarios de la Aplicación Web

Elaborado por: Autores

4.9. Diagrama de Estados

Un evento es un conjunto de acontecimientos que suceden en un determinado tiempo y que pueden ocurrir durante la manipulación de la plataforma web o de la aplicación interactiva.



Figura 4.8 Diagrama de Estado Aplicación Web

Elaborado por: Autores

Aplicación web

En la figura 4.8 se muestran los eventos que el usuario puede realizar en la plataforma web desde que inicia sesión hasta que cierra la sesión. El flujo comienza cuando el usuario accede a cada una de las opciones del menú principal, aquí se explica cómo cada acción del usuario (evento) genera una determinada acción dentro de la

plataforma. Si da click en la opción “Equipos” inmediatamente el usuario visualizara la Administración de Equipos, y así respectivamente con cada opción.

Aplicación interactiva



Figura 4.9 Diagrama de Estado Aplicación Interactiva

Elaborado por: Autores

Muestra todos los eventos que ocurren cuando el usuario utiliza la aplicación interactiva. Igual como se explica la figura 4.9, aquí se puede ver varios eventos que suceden cuando el usuario elige determinada opción.

4.10. Diagrama de Interacción

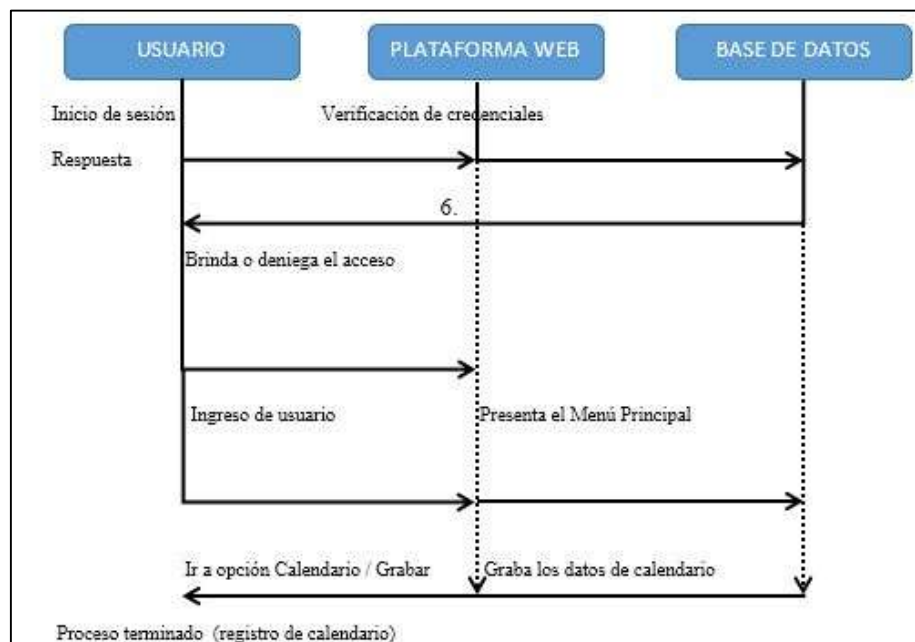


Figura 4.10 Diagrama de Interacción Aplicación web

Elaborado por: Autores

En la figura 4.11 se muestra el modelo entidad relación y estructura de las tablas de la base de datos de la plataforma web.

La entidad que representa a un jugador está asociada con las entidades de equipo, goleador y goles y tarjetas. Un equipo puede tener más de un jugador, un jugador solo está asociado a un equipo.

Un jugador puede tener asociado más de un gol y más de una tarjeta (amonestación)

La entidad partido está directamente relacionada con la entidad equipos, con la entidad de estado y con la entidad de resultado. Un partido solo puede tener un resultado.

La entidad usuario está asociada con la entidad roles, que determina los privilegios que el usuario tiene en la plataforma web. La entidad ci_sessions es una tabla auxiliar para el manejo de sesiones en la plataforma web.

4.12. Capa de Presentación

También llamada capa de usuario, básicamente es lo que el usuario ve al momento de acceder a la aplicación y aquella con la que interactúa directamente.

Módulo de Equipos

Equipo	Director Técnico	Estado	Estado	Escudo	Fecha de registro	Fecha de última modificación	Acciones
Barrabera Sporting Club	Carlos Incha	Estadio Monumental Banco Pichincha	Activo	No hay imagen	2014-01-21 00:37:02	2014-07-15 20:48:09	editar
Centro deportivo Océido	Gabriel Ferrero	Estadio Olímpico de Ruzumbá	Activo		2014-01-21 22:17:35	2014-07-15 20:46:40	editar
Club Deportivo El Nacional	Carlos Sevilla	Estadio Olímpico Atahualpa	Activo		2014-01-21 22:57:14	2014-07-13 30:47:10	editar
Club Sport Emelec	Gustavo Quinteros	Estadio George Capwell	Activo		2014-01-21 22:54:35	0000-00-00 00:00:00	editar
Deportivo Cuenca	Luis Solar	Estadio Alejandro Serrano Aguilar	Activo		2014-01-21 22:55:51	2014-01-25 17:01:40	editar
Deportivo Quito	Alex Carlos Garay	Estadio Olímpico Atahualpa	Activo		2014-01-21 22:13:49	0000-00-00 00:00:00	editar
Equipo A	Administrat	El Estadio	Inactivo	No hay imagen	2014-04-05 14:01:12	2014-07-16 21:23:54	editar
Independiente José Terán	Pablo Raposo	Estadio General Rumiñahui	Activo		2014-01-21 22:12:02	0000-00-00 00:00:00	editar
Independiente del Valle	Juan	Activo		2014-01-21 22:12:02	0000-00-00 00:00:00	editar	

Figura 4.12 Módulo de Equipos

Elaborado por: Autores

En el módulo de equipos se registran todos los equipos de la serie A que disputan el actual campeonato como se muestra en la figura 4.12. Una vez registrado se puede editar su información en caso que sea necesario.

Módulo de Jugadores



The screenshot shows a web application interface for managing players. At the top, there is a search bar and a 'Buscar' button. Below it, a table displays a list of players with columns for 'Nombres', 'Apellidos', 'Edad', 'Posición', '# dorsal', 'Equipo', and 'Acciones'. The table contains 15 rows of data, each representing a player with their respective details and an 'Editar' button in the 'Acciones' column.

Nombres	Apellidos	Edad	Posición	# dorsal	Equipo	Acciones
Adrian	Bola	25	Arquero	12	Club Deportivo El Nacional	Editar
Alberto	Quintero	27	Defensor		Maria Pabon Club	Editar
Alejandro	Pareja	22	Volante de ataque		Deportivo Guano	Editar
Alejandro	Prado	30	Volante central	9	Deportivo Guano	Editar
Alejandro	Vizcarra	24	Defensa lateral		Club Deportivo El Nacional	Editar
Alejandro	Espinoza	25	Defensa central	3	Universidad Católica	Editar
Alex	Baños	29	Volante de ataque	18	Barcelona Sporting Club	Editar
Alex	George	25	Defensor	20	Barcelona Sporting Club	Editar
Alex	Gonzalez	26	Volante de ataque		Deportivo Guano	Editar
Alexander	Dominguez	26	Arquero	22	Liga Deportiva Uruguiana	Editar
Alexis	Lemos	24	Arquero	12	Deportivo Guano	Editar
Alexis	De Jesus	25	Defensor		Deportivo Guano	Editar
Anderson	Ordóñez	20	Defensa central	3	Barcelona Sporting Club	Editar
Anderson	Blanco	18	Defensa lateral	30	Barcelona Sporting Club	Editar
Andrés	Ray	16	Defensa lateral	30	Barcelona Sporting Club	Editar

Figura 4.13 Módulo de Jugadores

Elaborado por: Autores

En la figura 4.13, se visualiza el módulo en donde se registra la información de los jugadores así como el equipo para el cual juegan. Se puede editar su información en caso de ser necesario.

Módulo de Usuarios



The screenshot shows a web application interface for managing users. At the top, there is a search bar and a 'Buscar' button. Below it, a table displays a list of users with columns for 'ID', 'Cédula', 'Nombres', 'Apellidos', 'Fecha y hora de registro', 'Fecha y hora de última modificación', 'Rol', and 'Acciones'. The table contains 3 rows of data, each representing a user with their respective details and 'Editar' and 'Borrar' buttons in the 'Acciones' column.

ID	Cédula	Nombres	Apellidos	Fecha y hora de registro	Fecha y hora de última modificación	Rol	Acciones
1	0999000000	adrian	adrian	2013-07-05 00:23:22	2014-03-01 17:56:52	Administrador	Editar Borrar
2	1101000000	Manuel	Sarragan	2014-02-10 22:19:26		Publicador	Editar Borrar
3	0807207003	Juan	Piguave	2014-02-10 22:18:25	2014-02-10 22:18:48	Publicador	Editar Borrar

Figura 4.14 Módulo de Usuarios

Elaborado por: Autores

Aquí en la figura 4.14, se muestra como se guarda información sobre los usuarios que podrán operar el sistema. Cabe recalcar que hay dos perfiles para los usuarios: Administrador y Publicador. Un usuario administrador tiene acceso a todos los módulos del sistema, en cambio un usuario publicador tiene acceso a todos los módulos menos al de usuarios. El Id de cada usuario es el número de cédula.

Módulo de Partidos

Aquí en la imagen 4.15 se puede visualizar el estado de los partidos actuales y de fechas pasadas, utilizando filtros de búsquedas ubicados en la parte inferior de la página (también disponibles en los demás módulos). Una vez que se finaliza un partido no se puede volver a editar su información.

Mostrar: 25 entradas										Buscar:	
Temporada	Etapas	Jornada	Equipo local	Goles equipo local	Equipo visitante	Goles equipo visitante	Estadio	Fecha y hora del partido	Estado del partido		
2014	1	Fecha # 1	Universidad Católica	1	Deportivo Quercia	1	Estadio Olímpico Abanipala	25/01/2014 - 12:00	Finalizado		
2014	1	Fecha # 1	Manita Fútbol Club	1	Liga Deportiva Universitaria	0	Estadio Jockey	25/01/2014 - 12:00	Finalizado		
2014	1	Fecha # 1	Liga de Liga	2	Club Deportivo El Nacional	1	Estadio Reina del Caire	25/01/2014 - 13:00	Finalizado		
2014	1	Fecha # 1	Independiente Jose Teran	2	Deportes Quito	0	Estadio General Rumiñahui	25/01/2014 - 15:00	Finalizado		
2014	1	Fecha # 1	Centro Deportivo Gimedo	2	Barcelona Sporting Club	1	Estadio Olímpico de Rostambá	26/01/2014 - 12:00	Finalizado		
2014	1	Fecha # 1	Club Sport Erenec	1	Mushuc Runa Sporting Club	0	Estadio George Capwell	26/01/2014 - 16:30	Finalizado		
2014	1	Fecha # 2	Club Deportivo El Nacional	0	Centro deportivo Gimedo	3	Estadio Olímpico Abanipala	29/01/2014 - 16:00	Finalizado		
2014	1	Fecha # 2	Mushuc Runa Sporting Club	1	Universidad Católica	2	Estadio General Rumiñahui	29/01/2014 - 16:00	Finalizado		
2014	1	Fecha # 2	Barcelona Sporting Club	1	Manita Fútbol Club	6	Estadio Monumental Banco Pichincha	29/01/2014 - 19:00	Finalizado		
2014	1	Fecha # 2	Liga Deportiva Universitaria	2	Liga de Liga	2	Estadio Casa Blanca	29/01/2014 - 19:00	Finalizado		
2014	1	Fecha # 2	Deportivo Quercia	1	Independiente Jose Teran	1	Estadio Alejandro Serrano Aguilar	29/01/2014 - 20:00	Finalizado		
2014	1	Fecha # 2	Deportivo Quito	0	Club Sport Erenec	0	Estadio Olímpico Abanipala	29/03/2014 - 16:00	Finalizado		

Figura 4.15 Módulo de Partidos

Elaborado por: Autores

Módulo de Resultados

Mostrar: 25 entradas										Buscar:	
Temporada	Etapas	Jornada	Equipo local	Goles equipo local	Equipo visitante	Goles equipo visitante	Estadio	Fecha y hora del partido	Estado del partido	Acciones	
2014	2	Fecha # 24	Barcelona Sporting Club	0	Centro deportivo Gimedo	0		18/06/2014 - 13:01	No jugado	Editar	
2014	2	Fecha # 24	Club Deportivo El Nacional	0	Club Sport Erenec	0		18/06/2014 - 13:01	No jugado	Editar	
2014	2	Fecha # 24	Deportivo Quercia	0	Deportes Quito	0		19/06/2014 - 13:01	No jugado	Editar	
2014	2	Fecha # 24	Independiente Jose Teran	0	Liga de Liga	0		19/06/2014 - 13:01	No jugado	Editar	
2014	2	Fecha # 24	Liga Deportiva Universitaria	0	Manita Fútbol Club	0		19/06/2014 - 13:01	No jugado	Editar	
2014	2	Fecha # 24	Universidad Católica	0	Mushuc Runa Sporting Club	0		19/06/2014 - 13:01	No jugado	Editar	
2014	1	Fecha # 1	Universidad Católica	1	Deportivo Quercia	1	Estadio Olímpico Abanipala	25/01/2014 - 12:00	Finalizado	Editar	
2014	1	Fecha # 1	Manita Fútbol Club	1	Liga Deportiva Universitaria	0	Estadio Jockey	25/01/2014 - 12:00	Finalizado	Editar	
2014	1	Fecha # 1	Liga de Liga	2	Club Deportivo El Nacional	1	Estadio Reina del Caire	25/01/2014 - 13:00	Finalizado	Editar	
2014	1	Fecha # 1	Independiente Jose Teran	2	Deportivo Quito	0	Estadio General Rumiñahui	25/01/2014 - 13:00	Finalizado	Editar	
2014	1	Fecha # 1	Centro Deportivo Gimedo	2	Barcelona Sporting Club	1	Estadio Olímpico de Rostambá	26/01/2014 - 12:00	Finalizado	Editar	

Figura 4.16 Módulo de Resultados

Elaborado por: Autores

Aquí en la figura 4.16 se visualizan todos los resultados de los partidos e información adicional tales como etapa, fecha, temporada, estadio y fecha y hora en que se disputo el partido.

Módulo de Tabla de Posiciones

Temporada	Etapa	Equipo	Partidos jugados	Goles a favor	Goles en contra	Gol diferencia	Puntos
2014	2	Manta Futbol Club	1	2	3	-1	3
2014	2	Club Deportivo El Nacional	1	1	2	-1	3
2014	2	Banana Sporting Club	1	0	0	0	1
2014	2	Universidad Católica	1	0	0	0	1
2014	2	Deportivo Guano	1	1	1	0	1
2014	2	Unión Rusa Sporting Club	1	1	1	0	1
2014	2	Centro deportivo Orosco	1	1	1	0	1
2014	2	Club Sport Esmes	1	1	1	0	1
2014	2	Liga Deportiva Universitaria	1	1	1	0	1
2014	2	Liga de Liga	1	1	1	0	1
2014	2	Deportivo Coenra	1	2	1	+1	3
2014	2	Independiente Jose Teran	1	3	2	+1	3
2014	1	Manta Futbol Club	22	21	26	-6	23
2014	1	Deportivo Coenra	22	21	31	-10	23
2014	1	Deportivo Guano	22	20	22	-2	26
2014	1	Centro deportivo Orosco	22	21	24	-3	26

Figura 4.17 Módulo Tabla de Posiciones

Elaborado por: Autores

En la figura 4.17, se muestra el módulo que permite visualizar la tabla de posiciones, goles a favor y en contra, partidos jugados. Así mismo cuenta con filtros de búsqueda en la parte inferior de la página.

Módulo de Calendario

Modulo de Generacion de Calendario

Generar Calendario

Volver al inicio principal

Cancelar

Fecha # 25

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Guardar

Fecha # 26

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Local: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Equipo Visitante: Banana Sporting Club

Guardar

Figura 4.18 Módulo de Calendario

Elaborado por: Autores

Aquí en la figura 4.18, se muestra como el usuario genera primero el calendario para la temporada actual, dando click en el botón “Generar Calendario”. Una vez generado, se muestra todas las jornadas del campeonato actual junto con listas de los equipos para que el usuario pueda guardar los partidos.

Módulo de Goles y Tarjetas

En la figura 4.19, se observa que una vez que los partidos son iniciados por el usuario en el Módulo de Partidos, en el módulo de Goles y Tarjetas se pueden ingresar los goles de cada equipo así como las amonestaciones que recibirá cada jugador dependiendo de lo que suceda en el transcurso del encuentro.

Fecha: # 34

Equipo Local	Jugadores Equipo Local	Amonestaciones	Resultado	Amonestaciones	Jugadores Equipo Visitante	Equipo Visitante	Minuto	Estado
Barcelona Sporting Club	Alex Graça		0 - 0		Andrés Vivas	Centro deportivo Cibeola	1	1er tiempo
Club Deportivo El Nacional	Adrian Bare		0 - 0		Angel Vera	Club Sport Emelec	1	1er tiempo
Deportivo Guayaquil	Alfredo Pizarro		0 - 0		Alfredo Parra	Deportivo Quito	1	1er tiempo
Independiente José Terán	Araken Lomas		0 - 0		Arnando Gomez	Liga de Loja	1	1er tiempo
Liga Deportiva Universitaria	Alexander Dominguez		0 - 0		Abelito Gomez	Manta Fútbol Club	1	1er tiempo
Universidad Católica	Alfredo Espinosa		0 - 0		Dylan Rodriguez	Mushuc Rana Sporting Club	1	1er tiempo

Sitio alojado en 200808

Figura 4.19 Módulo de Goles y Tarjeta

Elaborado por: Autores

Módulo de Configuraciones

El módulo mostrado en la figura 4.20 permite establecer 4 parámetros importantes: número de etapas del campeonato, número de fechas del campeonato, número de partidos por fecha del campeonato y el nombre del directorio local dentro del servidor que alojara los archivos de texto con información actualizada. Por defecto y para uso práctico, el valor es “data”.

Exportar Imprimir

Código	Parámetro	Valor numérico	Valor de texto	Acciones
LOCAL_DIR	Directorio local de archivos txt		data	<input type="button" value="Editar"/>
MEC	Número de etapas del campeonato	2		<input type="button" value="Editar"/>
MFC	Número de fechas del campeonato	44		<input type="button" value="Editar"/>
MPP	Número de partidos por fecha	6		<input type="button" value="Editar"/>

Sitio alojado en 200808

Figura 4.20 Módulo de Configuraciones

Elaborado por: Autores

Módulo de Sincronización

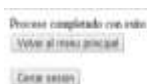


Figura 4.21 Módulo de Sincronización
Elaborado por: Los autores



Name	Last modified	Size	Description
Parent Directory			
goleadores.txt	02-Sep-2014 00:09	244	
proximos_partidos.txt	02-Sep-2014 00:09	368	
resultados.txt	02-Sep-2014 00:09	432	
tabla_posiciones.txt	02-Sep-2014 00:09	403	

Figura 4.22 Módulo de Sincronización
Elaborado por: Los autores

Este módulo visualizado en la Figura 4.22 se encarga de crear o re-crear los archivos de texto con información la última información actualizada de la plataforma web. En total son 4 archivos, ubicados en la carpeta data/ dentro del servidor de la plataforma:

Goleadores.txt: información con el top ten de los goleadores del actual campeonato.

Proximos_partidos.txt: información sobre los próximos encuentros a disputarse en el actual campeonato.

Resultados.txt: información sobre los resultados de los partidos finalizados en el actual campeonato.

Tabla_posiciones.txt: información sobre la tabla de posiciones del actual campeonato.

Módulo Acerca de



Figura 4.23 Módulo acerca de
Elaborado por: Autores

La figura 4.23 muestra información sobre los autores del presente proyecto de tesis.

4.13. Aplicación Web

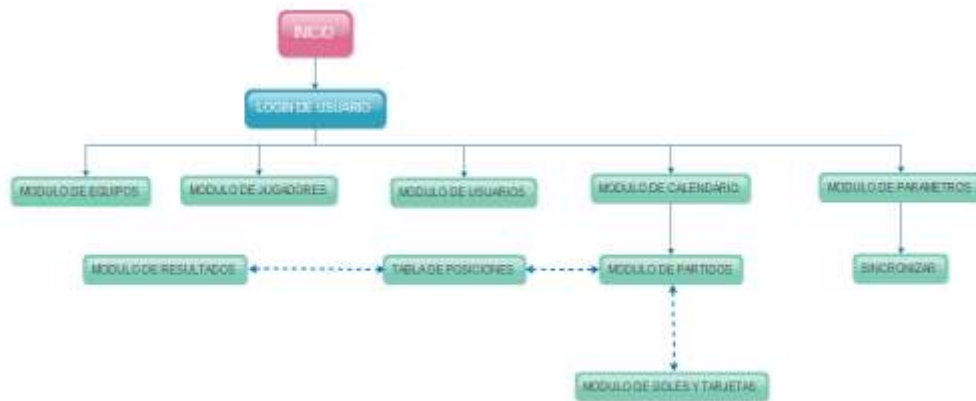


Figura 4.24 Diagrama Interfaces de Aplicación Web

Elaborado por: Autores

Al iniciar sesión el usuario visualiza el menú Principal, ahí tiene disponible las opciones:

- ✓ Módulo de equipos
- ✓ Módulo de Jugadores
- ✓ Módulo de Usuarios
- ✓ Módulo de Calendario
- ✓ Módulo de Parámetros
- ✓ Módulo de Tabla de Posiciones: relacionado con el módulo de Partidos y el Módulo de Resultados junto con el Módulo de Goles y tarjetas.
- ✓ Módulo Sincronizar

4.14. Aplicación Ginga

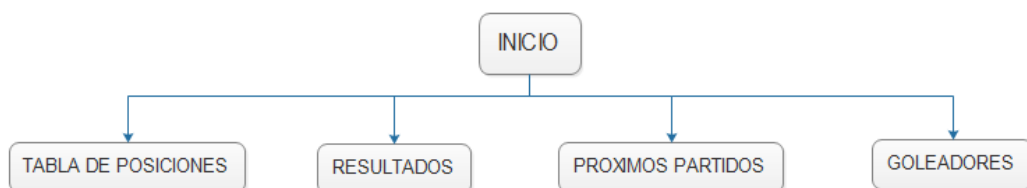


Figura 4.25 Interfaces de Aplicación Ginga

Elaborado por: Autores

La figura 4.25 muestra las opciones de la aplicación interactiva. Ofrece 4 opciones: Tabla de posiciones, Resultados, Próximos partidos y Goleadores. Inicialmente para mostrar el menú de estas opciones se debe oprimir el botón rojo.

4.15. Capa de Negocios

También llamada capa de lógica, aquí se establecen las reglas que deben cumplirse en el manejo de la plataforma web. Se comunica con la capa de presentación para poder recibir las solicitudes del usuario y validar la información que extrae de la capa de datos y mostrar esos resultados al usuario.

Para el presente trabajo de tesis, se utilizó el lenguaje de programación PHP junto con el framework Codeigniter, para el desarrollo de las clases que conforman la capa de negocio.

En resumen, el siguiente gráfico muestra la capa de negocio:

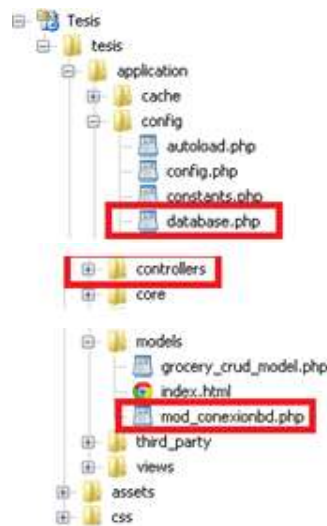


Figura 4.26 Estructura de MVC de la Aplicación Web

Elaborado por: Autores

En la figura 4.26, se observa que dentro de la carpeta /config existe el archivo database.php, el cual se encarga de establecer la conexión de la aplicación web a la base de datos MYSQL.

Dentro de la carpeta /controllers se encuentran los archivos “controladores”, es decir los archivos que gestionan la comunicación entre la capa de presentación y la capa de negocio. Se encargan de recibir las peticiones del usuario y redirigir los eventos en la plataforma web en base a dichas peticiones.

Dentro de la carpeta /models, está el archivo mod_conexionbd.php, aquí se encuentran todos los métodos que utiliza la capa de negocio para poder consultar, modificar e ingresar información en la base de datos.

4.16. Especificación de Librerías Utilizadas

Para el diseño y codificación de la plataforma web, se utilizó el lenguaje PHP y junto a este se utilizaron dos librerías de desarrollo:

4.16.1. Codeigniter Versión 2.01

Es un framework creado en PHP, proporciona una base mediante la cual se puede estructurar un proyecto web siguiendo el patrón MVC. Es un conjunto de librerías que contiene funciones que agilizan la tarea del programador, minimizando el tiempo de desarrollo y brindando una lógica de acceso a la base de datos.

4.16.2. GroceryCrud Version 1.3.1

Es una librería especializada en la creación de CRUD para proyectos web. Un CRUD es un sistema que permite realizar las 4 operaciones básicas sobre una tabla de una base de datos

C: Create (Crear o insertar registros)

R: Read (Leer, hacer un select)

U: Update (Actualizar un registro)

D: Delete (Borrar un registro)

La ventaja que proporciona es que además de acelerar la creación de la lógica de negocio, se pueden utilizar callbacks, funciones que sirven para realizar validaciones de datos, personalización de la vista de los registros almacenados, controles para subir imágenes, etc.

Para la creación de la aplicación interactiva, se utilizó el entorno de desarrollo Eclipse Luna versión 4.4, al cual se le agrego el plugin de Eclipse-NCL y el plugin para Eclipse-Lua.

Plugin NCL Eclipse: es un conjunto de librerías que sirven como un complemento para poder desarrollar programas interactivos en lenguaje NCL utilizando el entorno Eclipse. Agrega las funcionalidades necesarias para poder ejecutar y probar las aplicaciones.

Plugin Lua Eclipse: contiene las herramientas y complementos necesarios para poder crear scripts en lenguaje Lua, utilizando el entorno Eclipse.

4.17. Requisitos de Hardware y Software

Para las pruebas del prototipo de la aplicación interactiva se utilizó lo siguiente:

Hardware:

- ✓ Portatil AMD 4GB RAM, 1.3GHz 64 bits

Software:

- ✓ Máquina virtual VMWare Ginga NCL – Ubuntu Server 10.10
- ✓ Ambiente de desarrollo Eclipse Luna
- ✓ VMWare Player version 5.0.2

4.18. Pruebas y Resultados

Las pruebas de la aplicación interactiva se realizaron en GINGA-NCL utilizando un ambiente virtual, mediante el cual se emula un receptor digital que es capaz de reproducir este tipo de aplicaciones.

A continuación se muestra los resultados de la prueba:

Menu principal



Figura 4.27 Interfaz Ginga Menú Principal

Elaborado por: Autores

Tabla de posiciones



Figura 4.28 Interfaz Ginga Tabla de Posiciones

Elaborado por: Autores

Resultados



Figura 4.29 Interfaz Ginga Tabla de Resultados

Elaborado por: Autores

Proximos partidos

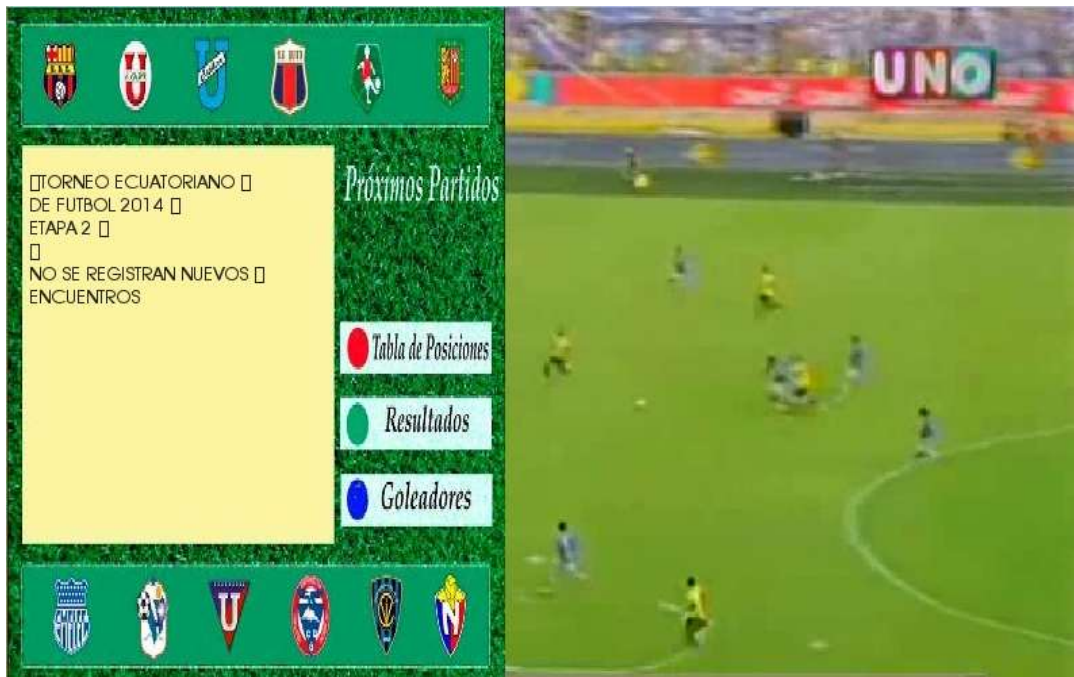


Figura 4.30 Interfaz Ginga Próximos Partidos
Elaborado por: Autores

Goleadores



Figura 4.31 Interfaz Ginga Goleadores
Elaborado por: Autores

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Con el desarrollo de la aplicación web, a la cual se puede acceder desde cualquier dispositivo móvil con Internet, tales como: Ordenador Portátil, Smart Phone, Ipad, un usuario administrador puede ingresar y actualizar datos referente a un partido en curso o planificado para la siguiente jornada. De esta manera se logra tener información consistente en tiempo real, la cual estará disponible para que el usuario televidente pueda visualizarla mediante la aplicación cliente. Esta información es transmitida por el aire desde la estación televisiva hasta el receptor digital del usuario.

Se pudo agregar interactividad por medio del lenguaje de programación Ginga NCL, ya que permite realizar y probar el diseño de las aplicaciones con menús interactivos, los cuales son accesibles por medio del control remoto. Cada botón (rojo, azul, amarillo, verde) muestra al usuario televidente las opciones que ofrece el programa: tabla de posiciones, goleadores, próximos partidos y resultados.

Esto es sin duda un valor agregado para el usuario televidente ya que la principal ventaja es que podrá estar informado no solo del partido actual que mira en su televisor sino que también podrá estar al tanto de las incidencias de otros partidos que se estén disputando en otra ciudad sin necesidad de consultar otra fuente.

En un principio fue necesario acoplarse a un paradigma diferente de programación, se requirió consultar manuales, tutoriales en línea, blogs, así como pequeños proyectos de ejemplo que sirvieron para poder comprender mejor el flujo del lenguaje al momento de empezar la fase de desarrollo.

Cabe resaltar también que se presentaron dificultades en la fase de testeo de la aplicación interactiva, al realizar pruebas en una máquina virtual no se obtuvo resultados más realistas que cuando se probó la aplicación directamente en un receptor digital. Se descubrió un error en el código de la aplicación que posteriormente fue corregido y se alcanzaron mejores resultados.

5.2. Recomendaciones

Una vez concluido este trabajo se recomienda a futuros estudiantes interesados que exploren mucho más sobre otros modelos de arquitectura que se puedan implementar y en el uso del sistema Ginga NCL, puesto que es una herramienta que aún está en desarrollo y evolución continua y puede ser mejorada en cuanto a sus funcionalidades, además se recomienda investigar con software de este tipo.

En base a lo aprendido, se puede establecer que uno de los puntos más importantes al momento de iniciar un proyecto de esta naturaleza es contar con herramientas de desarrollo adecuadas y que estén actualizadas a sus últimas versiones, así como en lo posible contar con equipos receptores de calidad que tengan instalados el middleware GINGA el cual es necesario ya que permite la ejecución de aplicaciones desarrolladas en lenguaje NCL.

También se destaca que algunos receptores digitales tienen incorporado un puerto Ethernet el cual sirve para conectar el dispositivo a una red LAN y tener acceso a internet. Con esto se pueden crear aplicaciones interactivas que envíen y reciban datos no solo desde una estación de televisión sino desde prácticamente cualquier plataforma orientada a servicios que utilice tecnología de transmisión de información usando diferentes medios (webservices por ejemplo) y mediante esto explotar un mercado de desarrollo de aplicaciones para diferente áreas:

Consultas de saldos y precios de productos, realizar encuestas y votaciones en línea, compras o reservaciones, obtener información sobre el clima, eventos, etc.

En el punto de vista social, se presenta los siguientes:

Campañas de prevención y educación sobre enfermedades información de relevancia (probabilidades de un próximo terremoto en zonas de peligro), estadísticas y novedades de último momento.

En resumen la interactividad permite ampliar la información que se transmite en el programa de televisión y brinda al usuario un panorama más completo sobre lo que necesita y desea saber.

Bibliografía

- Comunidad Ginga Ecuador. (28 de 06 de 2011). *Ginga Ecuador*. Obtenido de <http://goo.gl/nwgaqO>
- CubeNube. (07 de 11 de 2011). Obtenido de <http://blog.cubenube.com/2011/11/la-arquitectura-modelo-vista.html>
- Ginga. (29 de 11 de 2014). Obtenido de <http://goo.gl/ZMbjp8>
- Gobierno de España. (29 de 11 de 2014). *Ministerio de Industria, Energía y Turismo*. Obtenido de <http://goo.gl/oKQxCQ>
- López, M., & Oleas, K. (29 de 11 de 2014). *El diseño de interfaces entre el pc y el usuario utilizando isdb.tb y middleware ginga*. Obtenido de <http://goo.gl/KzS7fN>
- Secretaria Nacional de Telecomunicaciones. (2014). *Ministerio de Telecomunicaciones y Sociedad de la Información*. Obtenido de <http://goo.gl/aEiYRs>
- Superintendencia de Telecomunicaciones. (26 de 03 de 2010). *Supertel*. Obtenido de <http://goo.gl/NC28bh>
- Telecomunicaciones, Ministerio de. (29 de 11 de 2014). *Ministerio de Comunicaciones y Sociedad de la Información*. Obtenido de <http://goo.gl/DA3V31>
- Velo, F. (2012). *Guía del Usuario de CodeIgniter Versión 2.1.3*.
- VILLAGE island. (29 de 11 de 2014). *VILLAGE island*. Obtenido de <http://goo.gl/CKo8eq>
- Villanueva, J. M., & Velasquez, C. (2010). *Informe Preliminar: Estado del Arte de Receptores Set-Top-Box – Aplicaciones*. Lima.
- Welling, L., & Thomson, L. (2005). *Desarrollo Web con PHP y MySQL*. Madrid: Anaya.

ANEXOS

Anexo 1. Código del archivo Login.php

Este archivo contiene las funciones que sirven para validar las credenciales del usuario y permitir o denegar el acceso al menú principal

```
<?php
if(!defined('BASEPATH'))
    exit("No se puede ejecutar directamente el script");

class login extends CI_Controller {

    function login() {
        parent::__construct();
        $this->load->model('mod_ConexionBD','DB');
        $this->load->library('user_agent');
        $this->load->helper('date');
        date_default_timezone_set('America/Guayaquil');
    }

    function _remap($me) {
        switch($me) {
            case 'principal': $this->_ingresar(); break;
            case 'ver_menu': $this->_cargar_menu(); break;
            case 'salir': $this->salir(); break;
            //case 'logout': $this->
            default: $this->load->view('view_login');
        }
    }

    function _ingresar() {
        $usuario = $_POST['txtuser'];
        $password = $_POST['txtpass'];
        $data = $this->DB->existeUsuario($usuario,$password);
        $rol_codigo = $this->DB->tipoUsuario($usuario,$password);

        if($data || $this->session->userdata('logoneado')) {
            $ses_ip = $this->input->ip_address();
            $ses_agent = $this->agent->browser().'.'.$this->agent->version();
            $ses_user = $usuario;
            $ses_os = $this->agent->platform();

            // compruebo si el usuario ingresa desde un dispositivo movil
            if($this->agent->is_mobile()) {
                $ses_mobile_agent = $this->agent->mobile();
            }
            else {
                $ses_mobile_agent = null;
            }

            $ses_fecha_login = Lib_utils::getFecha(); //date('Y-m-d H:i:s'); //mdate('%Y-%m-%d %h:%i:%s');

            $data = array(
                'ses_ip' => $ses_ip,
                'ses_agent' => $ses_agent,
                'ses_os' => $ses_os,
                'ses_mobile_agent' => $ses_mobile_agent,
                'ses_user' => $ses_user,
                'ses_fecha_login' => $ses_fecha_login,
                'ses_fecha_logout' => null);
            $this->db->insert('tbl_sesiones',$data);
        }
    }
}
```

```

// aqui consulto el id del registro de sesion asociado al usuario que ingreso al sistema
$query_id = $this->db->query('select ses_id from tbl_sesiones where ses_ip="'. $ses_ip.'" and
ses_user="'. $ses_user.'" and ses_fecha_login="'. $ses_fecha_login.'"');
$ses_id = $query_id->row()->ses_id;

$session_data = array(
    'logoneado' => TRUE,
    'rol_codigo' => $rol_codigo,
    'ses_id' => $ses_id,
    'ses_ip' => $ses_ip,
    'ses_user' => $ses_user,
    'ses_fecha_login' => $ses_fecha_login
);
$this->session->set_userdata($session_data);

$this->load->view('view_menu_principal');
}
else {
    $this->lib_utils->mensaje('Usuario no existe');
    $this->lib_utils->adelante();
}
}

function _cargar_menu() {
    if($this->session->userdata('logoneado')) { // si el usuario ya esta logoneado
        $this->load->view('view_menu_principal');
    }
    else {
        redirect('login');
    }
}

function salir() {
    redirect('logout');
}
}
?>

```

Anexo 2. Código del módulo de sincronización de la plataforma web

```

<?php
if(!defined('BASEPATH'))
    exit('No se puede ejecutar el script directamente');

class admin_sync extends CI_Controller {

    public function admin_sync() {
        parent::__construct();
        $this->load->library('Lib_utils',null,'lu');
        $this->load->model('mod_ConexionBD','DB');
        $this->load->helper('file');
        $this->load->helper('date');
        $this->load->library('ftp');
        date_default_timezone_set('America/Guayaquil');
    }

    function _remap($me) {
        if(!$this->session->userdata('ses_user') && !$this->session->userdata('logoneado')) {
            redirect('login','refresh');
        }
    }
}

```

```

else {
    if($me == 'sincronizar') {
        $this->sincronizar();
    }
}

/** Envia los archivos de texto por ftp al servidor de aplicacion */
public function sincronizar() {
    try {
        $flag = null;
        $ldir = $this->DB->getLocalDirectory();
        delete_files($ldir);
        $flag = $this->createFile_Goleadores();
        $flag = $this->createFile_Resultados();
        $flag = $this->createFile_ProxPartidos();
        $flag = $this->createFile_TablaPos();

        if($flag == true) {
            $data['flag'] = true;
            $data['success'] = 'Proceso completado con exito';
            $this->load->view('view_admin_sync',$data);
        }
        else {
            $data['flag'] = false;
            $data['error'] = 'Error al intentar crear los archivos';
            $this->load->view('view_admin_sync',$data);
        }
    } catch(Exception $e) {
        show_error('ERROR: '.$e->getMessage().' --- '.$e->getTraceAsString());
    }
}

/** Crea el archivo txt de la tabla de posiciones */
public function createFile_TablaPos() {
    $data = "";
    $tabla = $this->DB->getTablaPos();
    $size = count($tabla);
    if($size == 0) {
        $data.= 'NO HAY INFORMACION DISPONIBLE';
    }
    else {
        $data.= "Equipo\t\t";
        $data.= " Ptos\t\t\n";
        //$data.= "GD.\t\t\n";
        for($i=0;$i<$size;$i++) {
            $cad = explode(".", $tabla[$i]);
            $data.= $cad[0]. " ";
            $data.= $cad[1]. " ";
            //$data.= $cad[2]. "\r\n";
            $data.= "\r\n";
        }
    }
    $ldir = $this->DB->getLocalDirectory();
    return $this->createTxt("", $data, $ldir. 'tabla_posiciones.txt');
}

/** Crea el archivo con el contenido txt (Proximos partidos) */
public function createFile_ProxPartidos() {
    $data = "";

```



```

$locales = $this->DB->getEquiposLocalesByFecha();
$visitantes = $this->DB->getEquiposVisitantesByFecha();
$size_l = count($locales);
$size_v = count($visitantes);
if($size_l == 0 || $size_v == 0) {
    $data.='NO SE REGISTRAN NUEVOS ENCUESTROS';
}
else {
    if($size_l == $size_v && ($size_l > 0 || $size_v > 0)) {
        for($i=0;$i<$size_l;$i++) {
            $data.=$locales[$i]."\r\nVS"." ".$visitantes[$i]."\r\n\r\n";
        }
    }
}
$ldir = $this->DB->getLocalDirectory();
return $this->createTxt("$data,$ldir.'proximos_partidos.txt');
}

/** Crea el archivo con el contenido txt (Resultados) */
public function createFile_Resultados() {
    $data = "";
    $locales = $this->DB->getEquiposLocalesResultado();
    $visitantes = $this->DB->getEquiposVisitantesResultado();
    $size_l = count($locales);
    $size_v = count($visitantes);
    if($size_l == 0 || $size_v == 0) {
        $data.='NO SE REGISTRAN RESULTADOS';
    }
    else {
        if($size_l == $size_v && ($size_l > 0 || $size_v > 0)) {
            for($i=0;$i<$size_l;$i++) {
                if(($i+1) == $size_l) {
                    $data.=$locales[$i]."\r\n".$visitantes[$i];
                }
                else {
                    $data.=$locales[$i]."\r\n".$visitantes[$i]."\r\n";
                }
            }
        }
    }
    $ldir = $this->DB->getLocalDirectory();
    return $this->createTxt("$data,$ldir.'resultados.txt');
}

/** Crea el archivo txt de Goleadores */
public function createFile_Goleadores() {
    $data = "";
    $goleadores = $this->DB->getGoleadores();
    $size = count($goleadores);
    if($size == 0) {
        $data.='NO HAY INFORMACION DE GOLEADORES';
    }
    else {
        for($i=0;$i<$size;$i++) {
            $cad = explode('.', $goleadores[$i]);
            $data.=$cad[0]."\t";
            $data.=$cad[1]."\r\n";
        }
    }
    $ldir = $this->DB->getLocalDirectory();
}

```

```

return $this->createTxt("$data,$ldir.'goleadores.txt');
}

/** Agrega el contenido al archivo txt */
public function createTxt($title,$data,$file) {
    $year = date('Y');
    try {
        $ldir = $this->DB->getLocalDirectory();
        $path = FCPATH.$ldir;
        if(!is_dir($path)) { //create the folder if it's not already exists
            mkdir($path,0777,TRUE);
        }
        else {
            //delete_files();
        }
        $cab = "TORNEO ECUATORIANO\r\nDE FUTBOL ".$year.chr(13).chr(10).'ETAPA
'.$this->DB->getEtapa().chr(13).chr(10)
            .$title.chr(13).chr(10).$data.'
            ';
        if(!write_file(FCPATH.$file,$cab,'w')) {
            echo 'Unable to write the file ';
            return false;
        }
        else {
            //echo 'Proceso ejecutado con exito';
            return true;
        }
    }catch(Exception $e) {
        echo 'Exception: '.$e->getMessage();
    }
}
}
?>

```

Anexo 3. Código del archivo mod_ConexionBD.php

Capa de acceso a la base datos en la plataforma web.

```

<?php
if(!defined('BASEPATH'))
    exit("No se puede ejecutar directamente el script");

class Mod_ConexionBD extends CI_Model {

    function Mod_ConexionBD() {
        parent::__construct();
        $this->load->helper('date');
        date_default_timezone_set('America/Guayaquil');
    }

    /** Funcion que retorna los minutos del partido (1 - 90) */
    function getMin() {
        $min = array();
        for($i=1;$i<=120;$i++)
            $min[] = $i;
        return $min;
    }

    /** Funcion que comprueba las credenciales del usuario */
    function existeUsuario($user,$password) {
        $query = $this->db->query("select usu_cedula,usu_password from tbl_usuarios ");
    }
}

```

```

        "where usu_cedula = '$user' and usu_password = '".base64_encode($password)."'");
    if ($query->num_rows() > 0) {
        return true;
    }
    return false;
}

/** Funcion que retorna el codigo asociado al privilegio del usuario */
function tipoUsuario($user,$password) {
    $sql = "select rol_codigo from tbl_usuarios ".
        "where usu_cedula = '$user' and usu_password = '".base64_encode($password)."'";
    $query = $this->db->query($sql);
    if ($query->num_rows() > 0) {
        return $query->row()->rol_codigo;
    }
}

/** Funcion que carga los nombres de los estadios */
function loadEstadios() {
    $estadios = $this->db->query('SELECT DISTINCT equ_estadio FROM tbl_equipos ORDER BY
equ_estadio');
    $eq = array();
    $cont = $estadios->num_rows();
    for($i = 0;$i < $cont;$i++)
        $eq[$estadios->row($i)->equ_estadio] = $estadios->row($i)->equ_estadio;
    return $eq;
}

/** Funcion que devuelve la ruta del escudo del equipo */
public function getEscudo($pk) {
    $sql = 'SELECT equ_imagen FROM tbl_equipos WHERE equ_id = '.$pk;
    $obj = $this->db->query($sql);
    $img = $obj->row()->equ_imagen;
    return $img;
}

/** Funcion que devuelve el pre- calendario */
public function getPreCalendario($temp) {
    $sql = 'SELECT pcl_generado FROM tbl_pre_calendario WHERE pcl_temporada = '.$temp;
    $res = $this->db->query($sql);
    return $res->row();
}

/** Funcion que devuelve el número de etapas del campeonato */
public function getNumEtapas() {
    $sql = 'SELECT con_valor_número AS ETAPAS FROM tbl_configuraciones WHERE con_param =
"NEC"';
    $nec = $this->db->query($sql);
    return $nec->row()->ETAPAS;
}

/** Funcion que devuelve el número de fechas del campeonato */
function getNumFechas() {
    $fechas = $this->db->query('SELECT con_valor_número AS NUM_FECHAS FROM
tbl_configuraciones WHERE con_param = "NFC"');
    return $fechas->row()->NUM_FECHAS;
}

/** Funcion que devuelve el número de partidos por fecha*/
function getNumParxFecha() {
    $par_fechas = $this->db->query("SELECT con_valor_número AS NUM_PAR_FECHAS FROM
tbl_configuraciones WHERE (con_param = 'NPF')");
    return $par_fechas->row()->NUM_PAR_FECHAS;
}

/** Devuelve el número de fecha desde el cual se genera el calendario */
function getFechaInicio() {

```

```

    $temporada = mdate('%Y');
    $sql = 'SELECT DISTINCT MAX(cal_num_fecha) AS FECHA_INICIO FROM tbl_calendario
WHERE cal_temporada = '.$temporada;
    $finicio = $this->db->query($sql);
    $val = $finicio->row()->FECHA_INICIO;
    if($val == null or empty($val))
        return 0;
    return $val;
}

/** Devuelve el estado del partido (1er tiempo, 2do tiempo, finalizado) */
public function getEstadoPartido($id) {
    $temporada = mdate('%Y');
    $sql = 'SELECT par_estado FROM tbl_partidos WHERE par_id = '.$id.' AND par_temporada =
'.$temporada;
    $state = $this->db->query($sql);
    if($state->num_rows() == 1)
        return $state->row()->par_estado;
}

/** Devuelve un arreglo con los equipos y su respectivo codigo */
public function loadEquipos() {
    $sql = 'SELECT DISTINCT equ_id AS ID, equ_nombre AS EQUIPO FROM tbl_equipos WHERE
equ_estado = "A" ORDER BY equ_nombre';
    $equ = array();
    $equipos = $this->db->query($sql);
    foreach($equipos->result() as $row)
        $equ[$row->ID] = $row->EQUIPO;
    return $equ;
}

/** Devuelve un arreglo con los equipos */
public function getEquipos() {
    $sql = 'SELECT DISTINCT equ_nombre AS EQUIPO FROM tbl_equipos WHERE equ_estado =
"A" ORDER BY equ_nombre ASC';
    $equ = array();
    $equipos = $this->db->query($sql);
    foreach($equipos->result() as $row)
        $equ[] = $row->EQUIPO;
    return $equ;
}

/** Carga el equipo local de acuerdo al partido y fecha correspondiente */
public function loadEquipoLocal($par_id,$par_temporada,$par_num_fecha) {
    $sql = 'SELECT DISTINCT e.equ_id AS ID, e.equ_nombre AS EQUIPO FROM tbl_equipos e,
tbl_partidos p
WHERE e.equ_estado = "A" AND p.par_id = '.$par_id.' AND p.par_temporada =
'.$par_temporada.' AND p.par_num_fecha = '.$par_num_fecha.' AND e.equ_id = p.par_equ_local';
    $data = array();
    $equipo = $this->db->query($sql);
    $cont = $equipo->num_rows();
    for($i = 0;$i < $cont;$i++)
        $data[$equipo->row($i)->ID] = $equipo->row($i)->EQUIPO;
    return $data;
}

/** Carga el equipo visitante de acuerdo al partido y fecha correspondiente */
public function loadEquipoVisit($par_id,$par_temporada,$par_num_fecha) {
    $sql = 'SELECT DISTINCT e.equ_id AS ID, e.equ_nombre AS EQUIPO FROM tbl_equipos e,
tbl_partidos p
WHERE e.equ_estado = "A" AND p.par_id = '.$par_id.' AND p.par_temporada =
'.$par_temporada.' AND p.par_num_fecha = '.$par_num_fecha.' AND e.equ_id = p.par_equ_visit';
    $data = array();
    $equipo = $this->db->query($sql);
    $cont = $equipo->num_rows();
    for($i = 0;$i < $cont;$i++)
        $data[$equipo->row($i)->ID] = $equipo->row($i)->EQUIPO;
}

```

```

return $data;
}

/** Devuelve un arreglo con los nombres de los jugadores */
public function getJugadores() {
    $sql = "SELECT jug_id as ID, jug_nombres AS NOMBRES, jug_apellidos AS APELLIDOS FROM
tbl_jugadores ORDER BY jug_nombres ASC";
    $jug = array();
    $jugadores = $this->db->query($sql);
    foreach($jugadores->result() as $row)
        $jug[$row->ID] = $row->NOMBRES.' '.$row->APELLIDOS;
    return $jug;
}

/** Devuelve un arreglo con los nombres de los jugadores del equipo local y si estan amonestados */
public function getJugadoresLocalesByEquipo($sequ_id,$fecha_id,$par_id) {
    $sql = 'SELECT DISTINCT j.jug_id as ID, j.jug_nombres AS NOMBRES, j.jug_apellidos AS
APELLIDOS FROM tbl_jugadores j
INNER JOIN tbl_goles_tarjetas glt
ON j.jug_id <> glt.glt_jug_local
AND ((glt.par_id = '.$par_id.' AND glt.glt_num_fecha = '.$fecha_id.' AND glt.glt_amon_local
<> "Tarjeta roja" AND glt.glt_num_ama_loc < 1)
OR (SELECT SUM(glt.glt_num_ama_loc) < 2))
WHERE j.equ_id = '.$sequ_id.' ORDER BY j.jug_nombres ASC';
    $jug = array();
    $jugadores = $this->db->query($sql);
    $cont = $jugadores->num_rows();
    for($i = 0; $i < $cont; $i++)
        $jug[$jugadores->row($i)->ID] = $jugadores->row($i)->NOMBRES.' '.$jugadores->row($i)-
>APELLIDOS;
    return $jug;
}

/** Devuelve un arreglo con los nombres de los jugadores del equipo visitante y si estan amonestados */
public function getJugadoresVisitantesByEquipo($sequ_id,$fecha_id,$par_id) {
    $sql = 'SELECT DISTINCT j.jug_id as ID, j.jug_nombres AS NOMBRES, j.jug_apellidos AS
APELLIDOS FROM tbl_jugadores j
INNER JOIN tbl_goles_tarjetas glt
ON j.jug_id <> glt.glt_jug_visit
AND ((glt.par_id = '.$par_id.' AND glt.glt_num_fecha = '.$fecha_id.' AND glt.glt_amon_local
<> "Tarjeta roja" AND glt.glt_num_ama_loc < 1)
OR (SELECT SUM(glt.glt_num_ama_loc) < 2))
WHERE j.equ_id = '.$sequ_id.' ORDER BY j.jug_nombres ASC';
    $jug = array();
    $jugadores = $this->db->query($sql);
    $cont = $jugadores->num_rows();
    for($i = 0; $i < $cont; $i++)
        $jug[$jugadores->row($i)->ID] = $jugadores->row($i)->NOMBRES.' '.$jugadores->row($i)-
>APELLIDOS;
    return $jug;
}

/** Devuelve la fechas del campeonato actual del módulo PARTIDOS */
public function getFechasPartidos() {
    $temporada = mdate('%Y');
    $sql = 'SELECT DISTINCT par_num_fecha AS FECHAS FROM tbl_partidos
WHERE par_temporada = '.$temporada.' AND par_estado <> 4 AND par_estado <> 1
ORDER BY par_num_fecha ASC';
    $data = array();
    $fechas = $this->db->query($sql);
    $size = $fechas->num_rows();
    for($i = 0; $i < $size; $i++)
        $data[] = $fechas->row($i)->FECHAS;
    return $data;
}

/** Devuelve los goles de la tabla de PARTIDOS */

```

```

public function getGolesByFecha($fecha,$array_id) {
    $str = "";
    $temporada = mdate('%Y');
    $cont = count($array_id);
    if($cont > 0) {
        for($i=0;$i<$cont;$i++) {
            if(($i+1) == $cont) {
                $str .= $array_id[$i];
                break;
            }
            $str .= $array_id[$i].',';
        }
    }
    $add = 'AND par_id IN (.'.$str.')';
    $sql = 'SELECT par_gol_local AS GLOCAL, par_gol_visit AS GVISIT FROM tbl_partidos
WHERE par_temporada = '.$temporada.' AND par_num_fecha = '.$fecha.'
AND par_estado <> 4 and par_estado <> 1 ';
    //print_r($sql);
    //if($cont > 0) $sql .= $add;
    $data = array();
    $par = $this->db->query($sql);
    $size = $par->num_rows();
    for($i = 0; $i < $size; $i++)
        $data[] = $par->row($i)->GLOCAL.'-'.$par->row($i)->GVISIT;
    return $data;
}

/** Devuelve el nombre del equipo de acuerdo al codigo */
public function getNombreEquipo($id) {
    $sql = 'SELECT equ_nombre as NOMBRE FROM tbl_equipos WHERE equ_estado = "A" AND
equ_id = '.$id;
    $val = $this->db->query($sql);
    return $val->row()->NOMBRE;
}

/** Devuelve los partidos de acuerdo a la fecha actual del campeonato */
public function getPartidosByFecha($fecha,$array_id) {
    $temporada = mdate('%Y');
    $cont = count($array_id);
    $str = "";
    for($i=0;$i<$cont;$i++) {
        if(($i+1) == $cont) {
            $str .= $array_id[$i];
            break;
        }
        $str .= $array_id[$i].',';
    }
    $add = 'AND par_id IN(.'.$str.')';
    $sql = 'SELECT par_equ_local AS ELOCAL, par_equ_visit AS EVISIT FROM tbl_partidos
WHERE par_temporada = '.$temporada.' AND par_num_fecha = '.$fecha.'
AND par_estado <> 4 AND par_estado <> 1 ';
    //if($cont > 0) $sql.=$add;
    $data = array();
    $par = $this->db->query($sql);
    $size = $par->num_rows();
    for($i = 0; $i < $size; $i++)
        $data[] = $par->row($i)->ELOCAL.'-'.$par->row($i)->EVISIT;
    return $data;
}

/** Devuelve el número de etapa correspondiente al partido de acuerdo a la temporada */
public function getEtapaPartido($par_id) {
    $par_temporada = mdate('%Y');
    $sql = 'SELECT par_etapa FROM tbl_partidos WHERE par_id = '.$par_id.' AND par_temporada =
'.$par_temporada;
    $set = $this->db->query($sql);
    return $set->row()->par_etapa;
}

```

```

}

/** Devuelve el estado del partido de acuerdo a la fecha del campeonato */
public function getEstadoByFecha($fecha,$sid) {
    $temporada = mdate('%Y');
    $estado = "";
    //$sql = 'SELECT est_estado as ESTADO FROM tbl_estados
//     WHERE est_id = (SELECT par_estado FROM tbl_partidos
//     WHERE par_temporada = '.$temporada.'
//     AND par_num_fecha = '.$fecha.'
//     AND par_id = '.$sid.'
//     AND par_estado <> 4 AND par_estado <> 1)';
    $sql = 'SELECT e.est_estado AS ESTADO FROM tbl_estados e
INNER JOIN tbl_partidos p
ON e.est_id = p.par_estado AND par_temporada = '.$temporada.'
AND par_num_fecha = '.$fecha.'
AND par_id = '.$sid.'
AND par_estado <> 4 AND par_estado <> 1';

    //print_r($sql);
    $val = $this->db->query($sql);
    if($val->num_rows() > 0)
        $estado = $val->row()->ESTADO;
    return $estado;
}

/** Devuelve un arreglo con los id de cada partido */
public function getIdByFecha($fecha) {
    $temporada = mdate('%Y');
    $sql = 'SELECT par_id AS ID FROM tbl_partidos WHERE par_temporada = '.$temporada.' AND
par_num_fecha = '.$fecha.'
AND par_estado <> 4 AND par_estado <> 1';
    $data = array();
    $id = $this->db->query($sql);
    $cont = $id->num_rows();
    if($cont > 0) {
        for($i = 0; $i < $cont; $i++)
            $data[] = $id->row($i)->ID;
    }
    return $data;
}

/** Devuelve los equipos locales del partido de acuerdo a la fecha */
public function getEquiposLocalesByFecha() {
    $temporada = mdate('%Y');
    $num_fecha = '(SELECT MAX(par_num_fecha) FROM tbl_partidos)';
    $sql = 'SELECT e.equ_nombre AS EQUIPO_LOCAL
FROM tbl_partidos p, tbl_equipos e
WHERE e.equ_estado = "A" AND p.par_temporada = '.$temporada.' AND p.par_num_fecha =
'.$num_fecha.' AND p.par_equ_local IN (SELECT e.equ_id FROM tbl_equipos ) AND p.par_estado = 1
AND p.par_etapa = (SELECT MAX(par_etapa) FROM tbl_partidos where par_temporada =
'.$temporada.')';
    $data = array();
    $equipos = $this->db->query($sql);
    foreach($equipos->result() as $row)
        $data[] = $row->EQUIPO_LOCAL;
    return $data;
}

/** Devuelve los equipos visitantes del partido de acuerdo a la fecha */
public function getEquiposVisitantesByFecha() {
    $temporada = mdate('%Y');
    $num_fecha = '(SELECT MAX(par_num_fecha) FROM tbl_partidos)';
    $sql = 'SELECT e.equ_nombre AS EQUIPO_VISITANTE
FROM tbl_partidos p, tbl_equipos e
WHERE e.equ_estado = "A" AND p.par_temporada = '.$temporada.' AND p.par_num_fecha =
'.$num_fecha.' AND p.par_equ_visit IN (SELECT e.equ_id FROM tbl_equipos ) AND p.par_estado = 1

```

```

        AND p.par_etapa = (SELECT MAX(par_etapa) FROM tbl_partidos where par_temporada =
        '$temporada.>');
        $data = array();
        $equipos = $this->db->query($sql);
        foreach($equipos->result() as $row)
            $data[] = $row->EQUIPO_VISITANTE;
        return $data;
    }

    /** Devuelve los equipos locales por temporada y por fecha para la página de resultados */
    public function getEquiposLocalesResultado() {
        $temporada = mdate('%Y');
        $sql = 'SELECT e.equ_nombre AS E_LOCAL, r.`res_gol_local` AS GLOCAL FROM tbl_equipos e,
tbl_resultados r, tbl_partidos p
        WHERE e.equ_estado = "A" AND e.`equ_id` = r.`res_equ_local` AND p.par_id = r.`par_id`
        AND r.res_temporada = '$temporada.' AND
        r.`par_num_fecha` = (SELECT MAX(par_num_fecha) FROM tbl_resultados WHERE
        res_temporada = '$temporada.>')
        AND r.res_etapa = (SELECT MAX(res_etapa) FROM tbl_resultados WHERE res_temporada
        = '$temporada.>');
        $data = array();
        $equ = $this->db->query($sql);
        $cont = $equ->num_rows();
        for($i = 0; $i < $cont; $i++)
            $data[] = $equ->row($i)->E_LOCAL.' '.$equ->row($i)->GLOCAL;
        return $data;
    }

    /** Devuelve los equipos visitantes por temporada y por fecha para la página de resultados */
    public function getEquiposVisitantesResultado() {
        $temporada = mdate('%Y');
        $sql = 'SELECT equ_nombre AS E_VISIT, r.`res_gol_visit` AS GVISIT, es.est_estado AS ESTADO
        FROM tbl_equipos e, tbl_resultados r, tbl_partidos p, tbl_estados es
        WHERE e.equ_estado = "A" AND e.`equ_id` = r.`res_equ_visit` AND p.par_id = r.`par_id`
        AND r.res_temporada = '$temporada.' AND
        r.`par_num_fecha` = (SELECT MAX(par_num_fecha) FROM tbl_resultados WHERE
        res_temporada = '$temporada.>') AND r.res_estado = es.est_id
        AND r.res_etapa = (SELECT MAX(res_etapa) FROM tbl_resultados WHERE res_temporada
        = '$temporada.>');
        $data = array();
        $equ = $this->db->query($sql);
        $cont = $equ->num_rows();
        for($i = 0; $i < $cont; $i++)
            $data[] = $equ->row($i)->E_VISIT.' '.$equ->row($i)->GVISIT."\\r\\n".$equ->row($i)->ESTADO;
        return $data;
    }

    /** Funcion que devuelve los goleadores del campeonato */
    public function getGoleadores() {
        $temporada = mdate('%Y');
        $sql = 'SELECT CONCAT(j.jug_nombres," ",j.`jug_apellidos`) AS JUGADOR, COUNT(*) AS
        GOLES
        FROM tbl_jugadores j, tbl_goleadores g
        WHERE g.gol_temporada = '$temporada.' AND g.`jug_id` = j.`jug_id`
        GROUP BY g.`jug_id` ORDER BY GOLES DESC LIMIT 10';
        $data = array();
        $gol = $this->db->query($sql);
        $cont = $gol->num_rows();
        for($i = 0; $i < $cont; $i++)
            $data[] = $gol->row($i)->JUGADOR.' '.$gol->row($i)->GOLES;
        return $data;
    }

    /** Funcion que devuelve la tabla de posiciones del campeonato de acuerdo a la temporada y etapa */
    public function getTablaPos() {
        $temporada = mdate('%Y');
    }

```



```

    $sql = 'SELECT e.`equ_nombre` AS EQUIPO, p.`pos_pts` AS PUNTOS, p.`pos_gd` AS
GOL_DIFERENCIA
    FROM tbl_equipos e, tbl_posiciones p
    WHERE e.equ_estado = "A" AND (p.`pos_equ_id` = e.`equ_id`) AND p.`pos_temporada` =
'.$temporada.' and p.pos_etapa = (SELECT DISTINCT MAX(pos_etapa) AS pos_etapa FROM
tbl_posiciones WHERE pos_temporada = '.$temporada.')
    ORDER BY p.`pos_pts` DESC, p.`pos_gd` DESC';
    //echo $sql;
    $tabla = array();
    $data = $this->db->query($sql);
    $cont = $data->num_rows();
    for($i = 0; $i < $cont; $i++)
        $tabla[] = $data->row($i)->EQUIPO.'.'. $data->row($i)->PUNTOS; //.'. $data->row($i)-
>GOL_DIFERENCIA;
    return $tabla;
}

/** Devuelve el conteo de partidos por fecha de la tabla PARTIDOS */
public function getCountPartidosByFecha($fecha) {
    $sql = 'SELECT COUNT(*) AS CONT FROM tbl_partidos WHERE par_num_fecha = '.$fecha;
    $cont = $this->db->query($sql);
    return $cont->row()->CONT;
}

/** Devuelve el nombre del host servidor de aplicaciones a donde se enviarian los txt actualizados */
public function getHost() {
    $sql = "SELECT con_valor_cadena as HOST FROM tbl_configuraciones WHERE con_param =
'HOST' ";
    $host = $this->db->query($sql);
    return $host->row()->HOST;
}

/** Devuelve el nombre de usuario y password del servidor de aplicaciones */
public function getUserAndPass() {
    $data = array();
    $sql = "SELECT con_valor_cadena AS VAL FROM tbl_configuraciones WHERE con_param =
'USER' OR con_param = 'PASS' ";
    $rst = $this->db->query($sql);
    foreach($rst->result() as $row)
        $data[] = $row->VAL;
    return $data;
}

/** Devuelve el directorio local en donde se guardan los archivos txt */
public function getLocalDirectory() {
    $sql = "SELECT con_valor_cadena AS LDIR FROM tbl_configuraciones WHERE con_param =
'LOCAL_DIR' ";
    $rdir = $this->db->query($sql);
    return $rdir->row()->LDIR;
}

/** Devuelve el directorio remoto a donde se enviara por FTP los archivos txt */
public function getRemoteDirectory() {
    $sql = "SELECT con_valor_cadena AS RDIR FROM tbl_configuraciones WHERE con_param =
'REMOTE_DIR' ";
    $rdir = $this->db->query($sql);
    return $rdir->row()->RDIR;
}

/** Devuelve true si el calendario de la temporada actual ya ha sido generado */
public function existeCalendario($temporada) {
    $sql = 'SELECT COUNT(*) AS COUNT FROM tbl_pre_calendario WHERE pcl_temporada =
'.$temporada.' AND pcl_generado = 1';
    $res = $this->db->query($sql);
    $cont = $res->row()->COUNT;
    $flag = false;
    if($cont == 1)

```

```

    $flag = true;
    return $flag;
}

/** Devuelve el número de partidos jugados de acuerdo a la temporada y a la etapa correspondiente */
public function getPJ($temporada,$equ_id,$etapa) {
    $sql = 'SELECT pos_pj FROM tbl_posiciones WHERE pos_temporada = '.$temporada.' AND
pos_etapa = '.$etapa.' AND pos_equ_id = '.$equ_id;
    $set = $this->db->query($sql);
    return $set->row()->pos_pj;
}

/** Actualiza el número de partidos jugados */
public function updatePJ($temporada,$equ_id,$pj) {
    $sql = 'UPDATE tbl_posiciones SET pos_pj = '.$pj.' WHERE pos_temporada = '.$temporada.' AND
pos_equ_id = '.$equ_id;
    return $this->db->query($sql);
}

/** Devuelve los datos de los equipos de acuerdo al partido */
public function getEquipoData($par_id,$par_temporada,$par_num_fecha) {
    $data = "";
    $sql = 'SELECT par_equ_local AS ELOCAL, par_equ_visit AS EVISIT,',
'par_fecha AS FECHA, par_estadio AS ESTADIO, par_arbitro as ARBITRO, par_estado as
ESTADO, par_incidencias as INCIDENCIAS '
'FROM tbl_partidos '
'WHERE par_id = '.$par_id.' AND par_temporada = '.$par_temporada.' AND par_num_fecha =
'.$par_num_fecha;
    $set = $this->db->query($sql);
    $size = $set->num_rows();
    for($i = 0; $i < $size; $i++)
        $data = $set->row($i)->ELOCAL.'!'.$set->row($i)->EVISIT.'!'.$set->row($i)->FECHA.'!'.$set-
->row($i)->ESTADIO.'!'.$set->row($i)->ARBITRO.'!'.$set->row($i)->ESTADO.'!'.$set->row($i)-
->INCIDENCIAS;
    return $data;
}

/** Devuelve los goles(gf,gc,gd) y los puntos para actualizar la tabla de posiciones de acuerdo a la
temporada y la etapa correspondiente dentro del campeonato */
public function getGolesyPuntos($pos_temporada,$pos_equ_id,$pos_etapa) {
    $data = "";
    $sql = 'SELECT pos_gf,pos_gc,pos_gd,pos_pts FROM tbl_posiciones
WHERE pos_temporada = '.$pos_temporada.' AND pos_etapa = '.$pos_etapa.' AND
pos_equ_id = '.$pos_equ_id;
    $set = $this->db->query($sql);
    $cont = $set->num_rows();
    for($i = 0;$i < $cont;$i++)
        $data = $set->row($i)->pos_gf.'!'.$set->row($i)->pos_gc.'!'.$set->row($i)->pos_gd.'!'.$set-
->row($i)->pos_pts;
    return $data;
}

/** Retorna la etapa actual de acuerdo a la temporada */
public function getEtapa() {
    $temporada = mdate('%Y');
    $etapa = 0;
    $sql = 'SELECT DISTINCT MAX(pos_etapa) AS pos_etapa FROM tbl_posiciones WHERE
pos_temporada = '.$temporada;
    $set = $this->db->query($sql);
    return $set->row()->pos_etapa;
}

/** ESTO REEMPLAZA AL TRIGGER par_res_insert (AFTER UPDATE ON tbl_partidos) */
public function parResInsert($post,$pk) {
    $temporada = mdate('%Y');
    $etapa = $post['par_etapa'];
    $pts_local = 0;

```

```

$pts_visit = 0;
$scout = 0;
$sql = 'SELECT COUNT(*) AS COUNT FROM tbl_resultados r WHERE r.par_id = '.$pk.
      ' AND r.res_temporada = '.$temporada.
      ' AND r.res_etapa = '.$etapa.
      ' AND r.par_num_fecha = '.$post['par_num_fecha'].
      ' AND r.res_equ_local = '.$post['par_equ_local'].
      ' AND r.res_equ_visit = '.$post['par_equ_visit'];
$res = $this->db->query($sql);
$scout = $res->row()->COUNT;
if(($post['par_estado'] == 4 || $post['par_estado'] == 3 || $post['par_estado'] == 2) && ($scout == 0)) {
    if($post['par_gol_local'] > $post['par_gol_visit'])
        $pts_local = 3;
    if($post['par_gol_visit'] > $post['par_gol_local'])
        $pts_visit = 3;
    if($post['par_gol_local'] == $post['par_gol_visit']) {
        $pts_local = 1;
        $pts_visit = 1;
    }
    $data = array('res_temporada' => $temporada,
                  'res_etapa' => $etapa,
                  'par_id' => $pk,
                  'par_num_fecha' => $post['par_num_fecha'],
                  'res_equ_local' => $post['par_equ_local'],
                  'res_equ_visit' => $post['par_equ_visit'],
                  'res_gol_local' => $post['par_gol_local'],
                  'res_gol_visit' => $post['par_gol_visit'],
                  'res_pts_local' => $pts_local,
                  'res_pts_visit' => $pts_visit,
                  'res_fecha' => date('Y-m-d H:i:s',strtotime(str_replace('/', '-', $post['par_fecha']))),
                  'res_estadio' => $post['par_estadio'],
                  'res_estado' => $post['par_estado'],
                  'res_incidencias' => $post['par_incidencias'],
                  'res_fecha_insert' => Lib_utils::getFecha());
    $this->db->insert('tbl_resultados', $data);
    if($post['par_estado'] == 4) // si el partido ya finaliza
        $this->insertOrUpdateTablaPos($data);
}
$pts_local = 0;
$pts_visit = 0;
if(($post['par_estado'] == 4 || $post['par_estado'] == 3 || $post['par_estado'] == 2) && ($scout == 1)) {
    if($post['par_gol_local'] > $post['par_gol_visit'])
        $pts_local = 3;
    if($post['par_gol_visit'] > $post['par_gol_local'])
        $pts_visit = 3;
    if($post['par_gol_local'] == $post['par_gol_visit']) {
        $pts_local = 1;
        $pts_visit = 1;
    }
}
$where = array('res_temporada' => $temporada,
               'par_id' => $pk,
               'par_num_fecha' => $post['par_num_fecha'],
               'res_equ_local' => $post['par_equ_local'],
               'res_equ_visit' => $post['par_equ_visit']);
$data = array('res_gol_local' => $post['par_gol_local'],
              'res_gol_visit' => $post['par_gol_visit'],
              'res_pts_local' => $pts_local,
              'res_pts_visit' => $pts_visit,
              'res_fecha' => date('Y-m-d H:i:s',strtotime(str_replace('/', '-', $post['par_fecha']))),
              'res_estadio' => $post['par_estadio'],
              'res_estado' => $post['par_estado'],
              'res_incidencias' => $post['par_incidencias'],
              'res_fecha_update' => Lib_utils::getFecha());
$this->db->update('tbl_resultados', $data, $where);
$data['res_equ_local'] = $post['par_equ_local'];
$data['res_equ_visit'] = $post['par_equ_visit'];
$data['pts_local'] = $pts_local;

```

```

        $data['pts_visit'] = $pts_visit;
        $data['res_etapa'] = $etapa;
        if($post['par_estado'] == 4) // si el partido ya finaliza
            $this->insertOrUpdateTablaPos($data);
    }
}

/** Funcion que inserta o actualiza la tabla de posiciones de acuerdo a los resultados
una vez que finalizan los partidos */
public function insertOrUpdateTablaPos($post) {
    $user = $this->session->userdata('ses_user');
    $temporada = mdate('%Y');
    $etapa = $post['res_etapa'];
    $cont = 0;
    $sql = 'SELECT COUNT(*) AS COUNT FROM tbl_posiciones p
          WHERE p.pos_temporada = '.$temporada.' AND p.pos_etapa = '.$etapa.' AND
          ((p.pos_equ_id = '.$post['res_equ_local'].'.) OR (p.pos_equ_id = '.$post['res_equ_visit'].'))';
    $res = $this->db->query($sql);
    $cont = $res->row()->COUNT;
    $pj_local = $this->DB->getPJ($temporada,$post['res_equ_local'],$etapa);
    $pj_visit = $this->DB->getPJ($temporada,$post['res_equ_visit'],$etapa);
    if($cont == 0) { // si no existe el registro se inserta
        $data1 = array('pos_temporada' => $temporada,
            'pos_etapa' => $etapa,
            'pos_equ_id' => $post['res_equ_local'],
            'pos_pj' => ++$pj_local,
            'pos_gf' => $post['res_gol_local'],
            'pos_gc' => $post['res_gol_visit'],
            'pos_gd' => ($post['res_gol_local'] - $post['res_gol_visit']),
            'pos_pts' => $post['res_pts_local'],
            'pos_user_insert' => $user,
            'pos_fecha_insert' => Lib_utils::getFecha());
        $data2 = array('pos_temporada' => $temporada,
            'pos_etapa' => $etapa,
            'pos_equ_id' => $post['res_equ_visit'],
            'pos_pj' => ++$pj_visit,
            'pos_gf' => $post['res_gol_visit'],
            'pos_gc' => $post['res_gol_local'],
            'pos_gd' => ($post['res_gol_visit'] - $post['res_gol_local']),
            'pos_pts' => $post['res_pts_visit'],
            'pos_user_insert' => $user,
            'pos_fecha_insert' => Lib_utils::getFecha());
        $this->db->insert('tbl_posiciones',$data1); // equipo local
        $this->db->insert('tbl_posiciones',$data2); // equipo visitante
    }
    else { // caso contrario solo se actualiza los pj, gf, gc, gd, y puntos
        $gyp = $this->getGolesyPuntos($temporada,$post['res_equ_local'],$etapa);
        $ext = explode(',',$gyp);
        $pos_gf = $ext[0];
        $pos_gc = $ext[1];
        $pos_gd = $ext[2];
        $pos_pts = $ext[3];
        $data = array('pos_pj' => ++$pj_local,
            'pos_gf' => ($post['res_gol_local'] + $pos_gf),
            'pos_gc' => ($post['res_gol_visit'] + $pos_gc),
            'pos_gd' => (($post['res_gol_local'] - $post['res_gol_visit']) + $pos_gd),
            'pos_pts' => ($post['pts_local'] + $pos_pts),
            'pos_user_update' => $user,
            'pos_fecha_update' => Lib_utils::getFecha());
        $where = array('pos_temporada' => $temporada,
            'pos_equ_id' => $post['res_equ_local']);
        $this->db->update('tbl_posiciones',$data,$where);
        ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
        $gyp = $this->getGolesyPuntos($temporada,$post['res_equ_visit'],$etapa);
        $ext = explode(',',$gyp);
        $pos_gf = $ext[0];
        $pos_gc = $ext[1];
    }
}

```

```

$pos_gd = $ext[2];
$pos_pts = $ext[3];
$data = array('pos_pj' => ++$pj_visit,
             'pos_gf' => ($post['res_gol_visit'] + $pos_gf),
             'pos_gc' => ($post['res_gol_local'] + $pos_gc),
             'pos_gd' => (($post['res_gol_visit'] - $post['res_gol_local']) + $pos_gd),
             'pos_pts' => ($post['pts_visit'] + $pos_pts),
             'pos_user_update' => $user,
             'pos_fecha_update' => Lib_utils::getFecha());
$where = array('pos_temporada' => $temporada,
             'pos_equ_id' => $post['res_equ_visit']);
$this->db->update('tbl_posiciones',$data,$where);
}
}
/** Carga las opciones de busqueda de informacion para las estadisticas */
public function getOpciones() {
    $data = array();
    $msg = array(
        0 => 'Goleadores',
        1 => 'Resultados',
        2 => 'Tabla de posiciones'
    );
    $opt = count($msg);
    for($i = 0; $i < $opt; $i++)
        $data[$i] = $msg[$i];
    return $data;
}

/** Retorna todas las temporadas para las estadisticas */
public function getTemporadas() {
    $data = array();
    $sql = 'SELECT pcl_temporada as TEMPORADA FROM tbl_pre_calendario';
    $res = $this->db->query($sql);
    $count = $res->num_rows();
    for($i = 0; $i < $count; $i++)
        $data[$res->row($i)->TEMPORADA] = $res->row($i)->TEMPORADA;
    return $data;
}

/** Retorna los goleadores de acuerdo a la temporada (para las estadisticas) */
public function getGoleadoresByTemp($temporada) {
    $sql = 'SELECT CONCAT(j.jug_nombres," ",j.jug_apellidos) AS JUGADOR, COUNT(*) AS
GOLES
FROM tbl_jugadores j, tbl_goleadores g
WHERE g.gol_temporada = '.$temporada.' AND g.jug_id = j.jug_id
GROUP BY g.jug_id` ORDER BY GOLES DESC LIMIT 10';
    $data = array();
    $gol = $this->db->query($sql);
    $count = $gol->num_rows();
    for($i = 0; $i < $count; $i++)
        $data[] = $gol->row($i)->JUGADOR.'!'.$gol->row($i)->GOLES;
    return $data;
}

public function getEquipoFinalista($etapa, $temporada) {
    $sql = 'SELECT e.equ_nombre AS equipo, p.pos_pts AS puntos, p.pos_gd AS gol_diferencia,
p.pos_etapa AS etapa
FROM tbl_posiciones p
INNER JOIN tblEquipos e ON e.equ_id = p.pos_equ_id
WHERE p.pos_temporada = '.$temporada.' AND p.pos_etapa = '.$etapa.'
ORDER BY p.pos_pts` DESC LIMIT 1';
    $set = $this->db->query($sql);
    return $set->row();
}

/** Funcion que retorna los equipos finalistas de la 1era y 2da etapa, que disputan la final */
public function getFinalistas() {

```

```

$equipos = array();
$temporada = mdate('%Y');
$etapas = $this->getEtapa();
for($i = 0; $i < $etapas; $i++) {
    $equipos[$i] = $this->getEquipoFinalista(($etapas+1),$temporada);
}
return $equipos;
}
}
?>

```

Anexo 4. Código de la función en la aplicación WebClient

Inicia la descarga de los archivos de texto desde el servidor web de la plataforma de manejo de información de los partidos del campeonato, al servidor donde se encuentra instalado VillageFlow y donde está cargada la aplicación interactiva.

```

private void initService()
{
    try
    {
        if (txtServerFolder.Text.Trim() == "" || txtLocalFolder.Text.Trim() == "" ||
            val(txtInterval.Text.Trim()) == 0)
        {
            MessageBox.Show("No se puede iniciar programa, faltan datos, asegure de ingresar toda la
            información requerida en todos los campos");
        }
        else
        {
            //borrar los archivos existentes antes de la descarga
            deleteFiles();

            WebClient client = new WebClient();
            string[] filenames = new string[4];
            filenames[0] = "goleadores.txt";
            filenames[1] = "proximos_partidos.txt";
            filenames[2] = "resultados.txt";
            filenames[3] = "tabla_posiciones.txt";
            for (int i = 0; i < filenames.Length; i++)
            {
                string address = txtServerFolder.Text.Trim() + "/" + filenames[i];
                string filename = txtLocalFolder.Text.Trim() + "\\\" + filenames[i];
                client.DownloadFile(address, filename);
            }
            client.Dispose();
            date = DateTime.Now.ToString();
            success = true;
            setMessage(date);
            saveData();
        }
    }
    catch (Exception e)
    {
        MessageBox.Show("Error al iniciar la descarga: " + e.Message);
        success = false;
    }
}

```

Anexo 5. Código de la aplicación interactiva, archivo main.ncl

Aquí se declaran las acciones y elementos que se mostrara en la pantalla dependiendo el botón que oprima el usuario.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="new_ncl_file" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <transitionBase>
      <transition id="tsFade" type="fade" dur="5s"/>
      <transition id="tsMenu" type="clockWipe" dur="5s" />
    </transitionBase>
    <regionBase>
      <region id="rgVideo" width="1%" height="1%" />
      <region id="rg1" width="100%" height="100%" />
      <region id="rgStyle" width="30%" height="80%" top="18%" />
      <region id="rgPage" width="50%" height="70%" top="23%" left="5%" />
      <region id="rgProxPar" width="50%" height="50%" top="31%" left="5%" />
    </regionBase>
    <region id="rgGoleadores" width="35%" height="50%" top="35%" left="5%" />
    <region id="rgResultados" width="90%" height="70%" top="27%" left="5%" />
    <region id="rgMenu" width="236px" height="109px" top="80%" />
  </regionBase>
  <descriptorBase>
    <descriptor id="dsVideo" region="rgVideo" />
    <descriptor id="dsStyle" region="rgStyle" />
    <descriptor id="ds1" region="rg1" />
    <descriptor id="dsPage" region="rgPage" />
    <descriptor id="dsProxPar" region="rgProxPar" />
    <descriptor id="dsGoleadores" region="rgGoleadores" />
    <descriptor id="dsResultados" region="rgResultados" />
    <descriptor id="dsMenu" region="rgMenu" explicitDur="30s" />
  </descriptorBase>
  <connectorBase>
    <causalConnector id="showMenu"> <!-- enunciar la consecuencia de esta accion previa -->
      <connectorParam name="keyCode" />
      <simpleCondition role="onSelection" key="$keyCode" />
      <simpleAction role="start" />
    </causalConnector>
    <causalConnector id="quitMenu"> <!-- enunciar la consecuencia de esta accion previa -->
      <connectorParam name="keyCode" />
      <simpleCondition role="onSelection" key="$keyCode" />
      <simpleAction role="stop" />
    </causalConnector>
    <causalConnector id="showTablaPos"> <!-- enunciar la consecuencia de esta accion previa -->
      <connectorParam name="keyCode" />
      <connectorParam name="var" />
      <simpleCondition role="onSelection" key="$keyCode" />
      <simpleAction role="start" />
    </causalConnector>
    <causalConnector id="quitTablaPos"> <!-- enunciar la consecuencia de esta accion previa -->
      <connectorParam name="keyCode" />
      <connectorParam name="var" />
    </causalConnector>
  </connectorBase>
</ncl>

```

```

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

esta accion previa -->
    <causalConnector id="showProxPartidos"> <!-- enunciar la consecuencia de
        <connectorParam name="keyCode" />
        <simpleCondition role="onSelection" key="$keyCode" />
        <simpleAction role="start" />
    </causalConnector>

esta accion previa -->
    <causalConnector id="quitProxPartidos"> <!-- enunciar la consecuencia de
        <connectorParam name="keyCode" />
        <simpleCondition role="onSelection" key="$keyCode" />
        <simpleAction role="stop" />
    </causalConnector>

esta accion previa -->
    <causalConnector id="showGoleadores"> <!-- enunciar la consecuencia de
        <connectorParam name="keyCode" />
        <simpleCondition role="onSelection" key="$keyCode" />
        <simpleAction role="start" />
    </causalConnector>

esta accion previa -->
    <causalConnector id="quitGoleadores"> <!-- enunciar la consecuencia de
        <connectorParam name="keyCode" />
        <simpleCondition role="onSelection" key="$keyCode" />
        <simpleAction role="stop" />
    </causalConnector>

esta accion previa -->
    <causalConnector id="showResultados"> <!-- enunciar la consecuencia de
        <connectorParam name="keyCode" />
        <simpleCondition role="onSelection" key="$keyCode" />
        <simpleAction role="start" />
    </causalConnector>

esta accion previa -->
    <causalConnector id="quitResultados"> <!-- enunciar la consecuencia de
        <connectorParam name="keyCode" />
        <simpleCondition role="onSelection" key="$keyCode" />
        <simpleAction role="stop" />
    </causalConnector>

    <causalConnector id="showStyle">
    <connectorParam name="keyCode"/>
    <connectorParam name="var"/>
    <simpleCondition role="onSelection" key="$keyCode"/>
    <simpleAction role="start"/>
    </causalConnector>

    <causalConnector id="quitStyle">
    <connectorParam name="keyCode"/>
    <connectorParam name="var"/>
    <simpleCondition role="onSelection" key="$keyCode"/>
    <simpleAction role="stop"/>
    </causalConnector>

    <causalConnector id="changeStyle">
    <connectorParam name="keyCode"/>
    <simpleCondition role="onSelection" key="$keyCode" />
    <compoundAction operator="seq">
        <simpleAction role="stop" />
        <simpleAction role="start" />
    </compoundAction>
    </causalConnector>

```



```

</causalConnector>

<causalConnector id="onEndMenu"> <!-- enunciar la consecuencia de esta accion previa --
>
    <simpleCondition role="onEnd" />
    <simpleAction role="stop" />
</causalConnector>

</connectorBase>

</head>
<body>

<port id="pws" component="ws"/>

<media id="ws" src="scripts/mensaje.lua" descriptor="ds1"/>

<media id="menu" src="img/menu6.jpg" descriptor="dsMenu" type="image/jpeg" />

<media id="mCuadroTablajos" src="img/Menu1-ROJO.jpg" descriptor="dsStyle" />
<media id="mCuadroResultados" src="img/Menu2-VERDE.jpg" descriptor="dsStyle"
/>
<media id="mCuadroProxPar" src="img/Menu3-AMARILLO.jpg"
descriptor="dsStyle" />
<media id="mCuadroGoleadores" src="img/Menu4-AZUL.jpg" descriptor="dsStyle"
/>

<media id="pageTablaPos" src="data/tabla_posiciones.txt" descriptor="dsPage" >
    <property name="fontSize" value="13"/>
    <property name="fontColor" value="black"/>
    <property name="fontWeight" value="bold"/>
</media>

<media id="pageProxPar" src="data/proximos_partidos.txt" descriptor="dsProxPar">
    <property name="fontSize" value="13"/>
    <property name="fontColor" value="black"/>
    <property name="fontWeight" value="bold"/>
</media>

<media id="pageGoleadores" src="data/goleadores.txt" descriptor="dsGoleadores" >
    <property name="fontSize" value="15"/>
    <property name="fontColor" value="black"/>
    <property name="fontWeight" value="bold"/>
</media>

<media id="pageResultados" src="data/resultados.txt" descriptor="dsResultados" >
    <property name="fontSize" value="12"/>
    <property name="fontColor" value="black"/>
    <property name="fontWeight" value="bold"/>
</media>

<link xconnector="showMenu">
    <bind role="onSelection" component="ws"/>
    <linkParam name="keyCode" value="RED"/>
    <bind role="start" component="menu" >
    </bind>
</link>

<!--<link xconnector="quitMenu">
    <bind role="onSelection" component="menu"/>
    <linkParam name="keyCode" value="EXIT"/>
    <bind role="stop" component="menu" >
    </bind>
</link-->

```

```

<link xconnector="showTablaPos">
  <bind role="onSelection" component="menu"/>
<linkParam name="keyCode" value="RED"/>
  <bind role="start" component="pageTablaPos"/>
</link>

<link xconnector="quitTablaPos">
  <bind role="onSelection" component="pageTablaPos"/>
<linkParam name="keyCode" value="RED"/>
  <bind role="stop" component="pageTablaPos"/>
</link>

<link xconnector="showProxPartidos">
  <bind role="onSelection" component="menu"/>
<linkParam name="keyCode" value="YELLOW"/>
  <bind role="start" component="pageProxPar"/>
</link>

<link xconnector="quitProxPartidos">
  <bind role="onSelection" component="pageProxPar"/>
<linkParam name="keyCode" value="YELLOW"/>
  <bind role="stop" component="pageProxPar"/>
</link>

<link xconnector="showResultados">
  <bind role="onSelection" component="menu"/>
<linkParam name="keyCode" value="GREEN"/>
  <bind role="start" component="pageResultados"/>
</link>

<link xconnector="quitResultados">
  <bind role="onSelection" component="pageResultados"/>
<linkParam name="keyCode" value="GREEN"/>
  <bind role="stop" component="pageResultados"/>
</link>

<link xconnector="showGoleadores">
  <bind role="onSelection" component="menu"/>
<linkParam name="keyCode" value="BLUE"/>
  <bind role="start" component="pageGoleadores"/>
</link>

<link xconnector="quitGoleadores">
  <bind role="onSelection" component="pageGoleadores"/>
<linkParam name="keyCode" value="BLUE"/>
  <bind role="stop" component="pageGoleadores"/>
</link>

<!-- AQUI COMIENZAN LOS ENLACES A LOS CONECTORES DE LOS
CUADROS HECHOS EN LUA -->

<link xconnector="showStyle">
  <bind role="onSelection" component="menu"/>
<linkParam name="keyCode" value="RED"/>
  <bind role="start" component="mcuadroTablapos"/>
</link>

<link xconnector="quitStyle">
  <bind role="onSelection" component="pageTablaPos"/>
<linkParam name="keyCode" value="RED"/>
  <bind role="stop" component="mcuadroTablapos"/>
</link>

<link xconnector="showStyle">
  <bind role="onSelection" component="menu"/>
<linkParam name="keyCode" value="GREEN"/>
  <bind role="start" component="mcuadroResultados"/>

```

```

</link>

<link xconnector="quitStyle">
  <bind role="onSelection" component="pageResultados"/>
</link>
<linkParam name="keyCode" value="GREEN"/>
  <bind role="stop" component="mcuadroResultados"/>
</link>-->

<link xconnector="showStyle">
  <bind role="onSelection" component="menu"/>
</link>
<linkParam name="keyCode" value="YELLOW"/>
  <bind role="start" component="mcuadroProxPar"/>
</link>

<link xconnector="quitStyle">
  <bind role="onSelection" component="pageProxPar"/>
</link>
<linkParam name="keyCode" value="YELLOW"/>
  <bind role="stop" component="mcuadroProxPar"/>
</link>-->

<link xconnector="showStyle">
  <bind role="onSelection" component="menu"/>
</link>
<linkParam name="keyCode" value="BLUE"/>
  <bind role="start" component="mcuadroGoleadores"/>
</link>

<link xconnector="quitStyle">
  <bind role="onSelection" component="pageGoleadores"/>
</link>
<linkParam name="keyCode" value="BLUE"/>
  <bind role="stop" component="mcuadroGoleadores"/>
</link>

<!-- Cuando pasan los 30 segundo si interaccion, se quita el menu y los cuadros -->
<link xconnector="onEndMenu">
  <bind role="onEnd" component="menu"/>
  <bind role="stop" component="pageTablaPos"/>
</link>

<link xconnector="onEndMenu">
  <bind role="onEnd" component="menu"/>
  <bind role="stop" component="pageResultados"/>
</link>

<link xconnector="onEndMenu">
  <bind role="onEnd" component="menu"/>
  <bind role="stop" component="pageProxPar"/>
</link>

<link xconnector="onEndMenu">
  <bind role="onEnd" component="menu"/>
  <bind role="stop" component="pageGoleadores"/>
</link>

<link xconnector="onEndMenu">
  <bind role="onEnd" component="menu"/>
  <bind role="stop" component="mcuadroTablapos"/>
</link>

<link xconnector="onEndMenu">
  <bind role="onEnd" component="menu"/>
  <bind role="stop" component="mcuadroResultados"/>
</link>

<link xconnector="onEndMenu">
  <bind role="onEnd" component="menu"/>
  <bind role="stop" component="mcuadroProxPar"/>
</link>

```

```
<link xconnector="onEndMenu">
  <bind role="onEnd" component="menu"/>
  <bind role="stop" component="mcuadroGoleadores"/>
</link>

<!-- ////////////////////////////////////////////////// -->

<!-- Tabla de posiciones - Resultados -->
<!-- <link xconnector="changeStyle">
  <bind role="onSelection" component="pageTablaPos"/>
  <linkParam name="keyCode" value="GREEN"/>
  <bind role="stop" component="pageTablaPos"/>
  <bind role="start" component="pageResultados"/>
</link> -->

<!--<link xconnector="changeStyle">
  <bind role="onSelection" component="mcuadroTablajos"/>
  <linkParam name="keyCode" value="GREEN"/>
  <bind role="stop" component="mcuadroTablajos"/>
  <bind role="start" component="mcuadroResultados"/>
</link-->

</body>
</ncl>
```

Anexo 6. Codigo del archivo cuadroGoleadores.lua

Dibuja en pantalla el cuadro en donde se muestra la tabla de goleadores.

```
function mostrarDatos(x,y,s,datos,color)
  canvas:attrColor(color)
  canvas:attrFont('Tiresias', s, 'bold')
  canvas:drawText(x, y, datos)
  canvas:flush()
end

function cuadroGoleadores()
  canvas:attrColor('black')
  canvas:drawRect('fill',5,5,320,350)
  mostrarDatos(40,8,16,"Aplicacion TDT",'white')
  canvas:attrColor('white')
  canvas:drawRect('fill',25,25,300,330)
  canvas:attrColor('gray')
  canvas:drawRect('fill',25,25,300,30)
  mostrarDatos(30,30,15,"Goleadores",'blue')
  canvas:flush()
end

cuadroGoleadores()
```

Anexo 7. Codigo del archivo cuadroProxPar.lua

Dibuja en pantalla el cuadro en donde se muestran los próximos partidos.

```
function mostrarDatos(x,y,s,datos,color)
  canvas:attrColor(color)
  canvas:attrFont('Tiresias', s, 'bold')
  canvas:drawText(x, y, datos)
  canvas:flush()
end

function cuadroProximosPartidos()
```

```

        canvas:attrColor('black')
        canvas:drawRect('fill',5,5,320,350)
        mostrarDatos(40,8,16,"Aplicacion TDT",'white')
        canvas:attrColor('white')
        canvas:drawRect('fill',25,25,300,330)
        canvas:attrColor('gray')
        canvas:drawRect('fill',25,25,300,30)
        mostrarDatos(30,30,15,"Proximos Partidos",'yellow')
        canvas:flush()
    end
cuadroProximosPartidos()

```

Anexo 8. Codigo del archivo cuadroResultados.lua

Dibuja en pantalla el cuadro en donde se muestran los resultados.

```

function mostrarDatos(x,y,s,datos,color)
    canvas:attrColor(color)
    canvas:attrFont("Tiresias", s, 'bold')
    canvas:drawText(x, y, datos)
    canvas:flush()
end

function cuadroResultados()
    canvas:attrColor('black')
    canvas:drawRect('fill',5,5,320,350)
    mostrarDatos(40,8,16,"Aplicacion TDT",'white')
    canvas:attrColor('white')
    canvas:drawRect('fill',25,25,300,330)
    canvas:attrColor('gray')
    canvas:drawRect('fill',25,25,300,30)
    mostrarDatos(30,30,15,"Resultados",'green')
    canvas:flush()
end

cuadroResultados()

```

Anexo 9. Codigo del archivo cuadroTablaPos.lua.

Dibuja en pantalla el cuadro en donde se muestran la tabla de posiciones.

```

function mostrarDatos(x,y,s,datos,color)
    canvas:attrColor(color)
    canvas:attrFont("Tiresias", s, 'bold')
    canvas:drawText(x, y, datos)
    canvas:flush()
end

function cuadroTablaPos()
    canvas:attrColor('black')
    canvas:drawRect('fill',5,5,320,350)
    mostrarDatos(40,8,16,"Aplicacion TDT",'white')
    canvas:attrColor('white')
    canvas:drawRect('fill',25,25,300,330)
    canvas:attrColor('gray')
    canvas:drawRect('fill',25,25,300,30)
    mostrarDatos(30,30,15,"Tabla de posiciones",'red')
    canvas:flush()
end

cuadroTablaPos()

```

Anexo 10. Diccionario de Datos de Tablas

A continuación se detalla el diccionario de datos de la base de datos de la plataforma web.

Tabla ci_sessions (manejo de sesiones en codeigniter)

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
session_id	Varchar(40)	Primary Key, not null	Id de la sesión actual
ip_address	Varchar(45)	Not null	Dirección ip del equipo donde inicia sesión
user_agent	Varchar(120)	Not null	Nombre y versión del SO y navegador web
last_activity	Int	Not null, unsigned	Estampa de tiempo que indica la duración de la sesión
user_data	Text	Not null	Datos del usuario en la sesión actual (cookies)

Tabla de pre- calendario

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
pcl_id	Int	Primary key auto-incrementable	Id del registro
pcl_temporada	Int	Primary key	Temporada del precalendario
pcl_generado	Tinyint	Not null	Flag que indica si el calendario de la actual temporada ha sido generado
pcl_user_insert	Varchar(10)	Nullable	Usuario que creo el registro
pcl_fecha_insert	Datetime	Nullable	Fecha y hora de creación del registro

Tabla de calendario

COLUMNA	TIP DE DATO	DETALLE	DESCRIPCION
cal_id	Int	Primary key auto-incrementable	Id del registro
cal_temporada	Int	Primary key	Año en el cual se desarrolla el campeonato
cal_etapa	Int	Primary key	Número correspondiente de etapa dentro del campeonato
cal_num_fecha	Int	Primary key	Número de fecha asociado al partido
cal_elocal	Int	Primary key	Id del equipo local
cal_evisit	Int	Primary key	Id del equipo visitante
cal_user_insert	Varchar(10)	Not null	Usuario que creo el registro
cal_fecha_insert	Datetime	Not null	Fecha y hora de creación del registro

Tabla de configuraciones (parámetros)

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
con_id	Int	Primary key autoincrementable	Id del registro
con_param	Varchar(15)	Primary key not null	Código asociado al parámetro
con_nombre	Varchar(50)	Not null	Nombre de la opción de configuración
con_valor_número	Int	Nullable	Valor de la opción si es un número
con_valor_cadena	Varchar(200)	Nullable	Valor de la opción si es un texto
con_user_insert	Varchar(10)	Not null	Usuario que creo la opción

con_user_update	Varchar(10)	Not null	Usuario que hizo la ultima modificación a la opcion
con_fecha_insert	Datetime	Nullable	Fecha y hora de creación del registro
con_fecha_update	Datetime	Nullable	Fecha y hora de la ultima modificación del registro

Tabla de equipos

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
equ_id	Int	Primary key auto-incrementable	Id del registro
equ_nombre	Varchar(50)	Not null	Nombre del equipo
equ_dt	Varchar(30)	Not null	Nombre del director técnico
equ_estadio	Varchar(50)	Not null	Estadio del equipo donde juega de local
equ_estado	Varchar(1)		Estado de actividad del equipo (A/I) Activo, Inactivo
equ_imagen	Varchar(100)		Imagen del equipo
equ_user_insert	Varchar(10)	Not null	Usuario que creo el registro
equ_user_update	Varchar(10)	Not null	Usuario que realizo la ultima modificación al registro
equ_fecha_insert	Datetime	Not null	Fecha y hora de creación del registro
equ_fecha_update	Datetime	Not null	Fecha y hora de la ultima modificación del registro

Tabla de estados

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
est_id	Int	Primary key auto-incrementable	Id del registro
est_estado	Varchar(15)	Not null	Estado

Tabla de goleadores

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
gol_id	Int	Primary key auto-incrementable	Id del registro
gol_temporada	Int	Primary key auto-incrementable	Temporada del campeonato
par_fecha	Int	Primary key auto-incrementable	Id de la fecha correspondiente al actual campeonato
par_id	Int	Foreign Key	Id del partido
jug_id	Int	Foreign Key	Id del jugador
jug_goles	Int		Goles que ha anotado el jugador

Tabla de goles y tarjetas

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
glt_id	Int	Primary key auto-incrementable	Id del registro
par_id	Int	Not null	Id del partido relacionado a la fecha
glt_temporada	Int	Primary key not null	Año en el cual se desarrolla el campeonato
glt_num_fecha	Int	Not null	Fecha actual del campeonato
glt_jug_local	Int	Nullable	Id del jugador del equipo local
glt_gol_local	Int	Nullable	Gol del equipo local
glt_amon_local	Varchar(20)	Nullable	Amonestación al jugador local

glt_jug_visit	Int	Nullable	Id del jugador del equipo visitante
glt_gol_visit	Int	Nullable	Gol del equipo visitante
glt_amon_visit	Varchar(20)	Nullable	Amonestación al jugador visitante
glt_num_ama_loc	Int	Valor por default 0	Número de tarjetas amarillas del jugador del equipo local
glt_num_ama_vis	Int		Número de tarjetas amarillas del jugador del equipo visitante
glt_minuto	Int		Minuto del partido en el cual transcurrió el evento
glt_autogol	Tinyint	Valor por default 0	Flag que indica si es autogol

Tabla de jugadores

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
jug_id	Int	Primary Key not null auto-incremental	Id del registro
jug_nombres	Varchar(30)	Not null	Nombres del jugador
jug_apellidos	Varchar(30)	Not null	Apellidos del jugador
jug_edad	Varchar(2)	Not null	Edad del jugador
jug_posicion	Varchar(20)	Not null	Demarcacion del jugador
jug_dorsal	Varchar(2)	Not null	Número del dorsal del jugador
equ_id	Int	Foreign key, not null	Código asociado al equipo del jugador
jug_user_insert	Varchar(10)	Not null	Usuario que creo el registro
jug_user_update	Varchar(10)	Not null	Usuario que realizo la ultima modificación del registro

jug_fecha_insert	Datetime	Not null	Fecha y hora del registro
jug_fecha_update	Datetime	Not null	Fecha y hora de la ultima modificación del registro

Tabla de partidos

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
par_id	Int	Primary key not null auto incremental	Id del registro
par_temporada	Int		Año actual en el cual se desarrolla el campeonato
par_etapa	Int		Número de etapa correspondiente dentro del campeonato
par_num_fecha	Int	Not null	Jornada del partido
par_equ_local	Int	Foreign Key not null	Equipo local
par_equ_visit	Int	Foreign Key not null	Equipo visitante
par_gol_local	Int	Not null	Goles del equipo local
par_gol_visit	Int	Not null	Goles del equipo visitante
par_fecha	Datetime	Not null	Fecha y hora del partido
par_estadio	Varchar(80)	Not null	Estadio en donde se realiza el partido
par_arbitro	Varchar(30)	Not null	Arbitro del partido
par_estado	Int	Foreign key not null	Estado del partido(1er tiempo, 2do tiempo, etc)
par_incidencias	Varchar(80)	Nullable	Eventos que ocurren durante el partido
par_usu_insert	Varchar(10)	Not null	Usuario que creo el registro
par_usu_update	Varchar(10)	Not null	Usuario que realizo la ultima modificación del registro

par_fecha_insert	Datetime	Nullable	Fecha y hora de creación del registro
par_fecha_update	Datetime	Nullable	Fecha y hora de última modificación del registro

Tabla de posiciones

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
pos_id	Int	Primary key not null auto incremental	Id del registro
pos_temporada	Int	Primary key not null	Año actual en el que se desarrolla el campeonato
pos_etapa	Int	Not null	Número de etapa correspondiente dentro del campeonato
pos_equ_id	Int	Primary key not null	Id del equipo
pos_pj	Int	Not null	Número de partidos jugados
pos_pg	Int	Not null	Número de partidos ganados
pos_pe	Int	Not null	Número de partidos empatados
pos_pp	Int	Not null	Número de partidos perdidos
pos_gf	Int	Not null	Número de goles a favor
pos_gc	Int	Not null	Número de goles en contra
pos_gd	Int	Not null	Número de gol diferencia
pos_pts	Int	Not null	Número de puntos del equipo
pos_user_insert	Varchar(10)	Not null	Usuario que inserto el registro
pos_user_update	Varchar(10)		Usuario que modifico el registro
pos_fecha_insert	Datetime	Nullable	Fecha y hora de la inserción del registro

pos_fecha_update	Datetime	Nullable	Fecha y hora de la modificacion del registro

Tabla de resultados

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
res_id	Int	Primary key not null auto incremental	Id del registro
res_temporada	Int	Not null	Año actual en el que se desarrolla el campeonato
res_etapa	Int	Not null	Número de etapa correspondiente dentro del campeonato
par_id	Int	Foreign key	Código del partido (referencia de tbl_partidos)
par_num_fecha	Int		Número de la fecha del partido en el campeonato
res_equ_local	Int	Not null	Código del equipo local
res_equ_visit	Int	Not null	Código del equipo visitante
res_gol_local	Int	Not null	Goles del equipo local
res_gol_visit	Int	Not null	Goles del equipo visitante
res_pts_local	Int	Not null	Puntos que obtuvo el equipo local como resultado
res_pts_visit	Int	Not null	Puntos que obtuvo el equipo local como resultado
res_fecha	Datetime	Not null	Fecha y hora de inicio del partido
res_estadio	Varchar(80)	Not null	Estadio donde se desarrollo el partido el partido

res_estado	Int	Not null	Estado del partido
res_incidencias	Varchar(80)	Nullable	Incidencias que ocurren durante el partido
res_fecha_insert	Datetime	Nullable	Fecha y hora de creación del registro
res_fecha_update	Datetime	Nullable	Fecha y hora de última modificación del registro

Tabla de roles de usuario

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
rol_id	Int	Primary key not null auto incremental	Id del registro
rol_nombre	Varchar(15)	Not null	Nombre del rol asignado al usuario

Tabla de sesiones

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
ses_id	Int	Primary key not null auto-incremental	Id del registro
ses_ip	Varchar(15)	Primary key not null	Ip de la maquina en la cual se accedió al sistema
ses_agent	Varchar(15)	Not null	Nombre del navegador en el cual accedió al sistema
ses_os	Varchar(15)	Not null	Nombre y versión del sistema operativo de la maquina o dispositivo desde el cual accedió al sistema

ses_mobile_agent	Varchar(15)	Nullable	Nombre del dispositivo móvil desde el cual se accedió al sistema
ses_user	Varchar(10)	Primary key not null	Usuario que inicio sesión
ses_fecha_login	Datetime	Not null	Fecha y hora de inicio de sesión
ses_fecha_logout	Datetime	Nullable	Fecha y hora de cierre de sesión

Tabla de usuarios

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCION
usu_id	Int	Primary key not null auto incremental	Id del registro
usu_cedula	Varchar(10)	Primary key not null	Cedula del usuario
usu_password	Varchar(80)	Not null	Password del usuario
usu_nombre	Varchar(30)	Not null	Nombres del usuario
usu_apellido	Varchar(30)	Not null	Apellidos del usuario
usu_email	Varchar(50)	Nullable	Email de usuario
usu_telefono	Varchar(10)	Nullable	Telefono convencional del usuario
usu_celular	Varchar(10)	Nullable	Celular del usuario
usu_direccion	Varchar(50)	Nullable	Dirección del domicilio del usuario
rol_codigo	Int	Foreign key Not null	Código asociado al rol del usuario en el sistema
usu_user_insert	Varchar(10)	Not null	Usuario que creo el registro
usu_user_update	Varchar(10)	Nullable	Usuario que hizo la ultima modificación del registro
usu_fecha_insert	Datetime	Nullable	Fecha y hora de creación del registro

usu_fecha_update	Datetime	Nullable	Fecha y hora de la ultima modificación del registro
-------------------------	----------	----------	---

Anexo 11. Manual de Usuario Plataforma Web

Objetivo de Manual

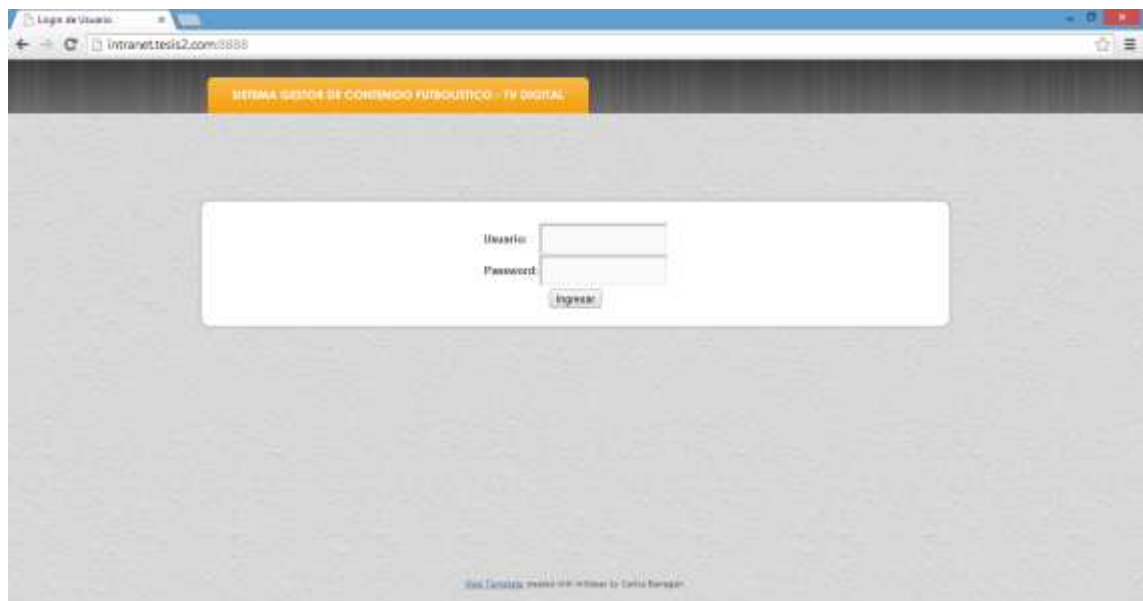
Este manual pretende guiar al usuario operador de la plataforma web para que este en capacidad de poder manejar las diversas opciones que presenta y así generar la base de información respecto al campeonato ecuatoriano de futbol.

Se entiende por plataforma web a la aplicación del lado del servidor que alimenta la base de datos del sistema y a su vez esta misma aplicación enviara los datos a la aplicación cliente que el usuario televidente podrá utilizar mediante su receptor de televisión digital.

El usuario operador será el único responsable del ingreso de información concisa y acorde a los resultados de cada partido que se dispute en el campeonato. Así mismo el sistema ofrece una interfaz de fácil manejo y aprendizaje, además de roles asociados a cada usuario (Administrador, Publicador) los cuales se detallara en las siguientes secciones.

Nota: Se recomienda utilizar el explorador google chrome para una mejor experiencia de usuario. Si presenta problemas proceda a utilizar Firefox.

Inicio de Sesión en el Sistema



Esta es la pantalla inicial en la cual el operador deberá ingresar sus credenciales. El campo usuario es el número de cedula del operador y el campo password es la clave secreta que se le asignara a dicho usuario.

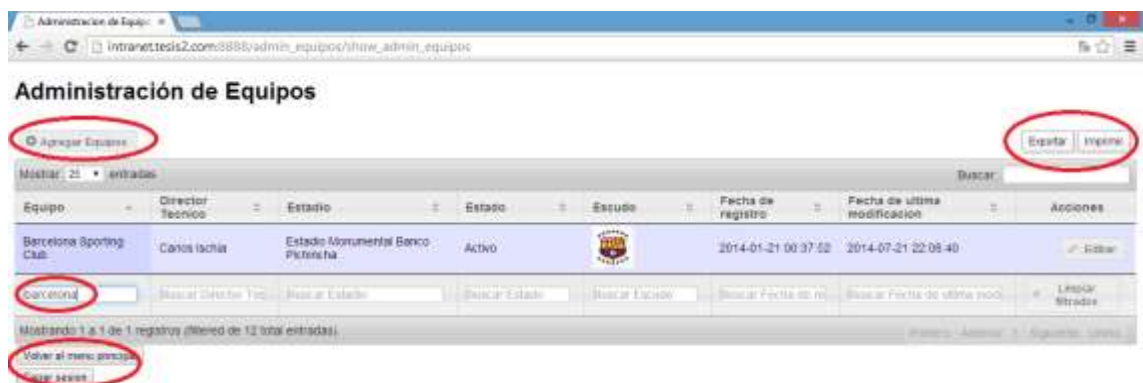
Una vez que ingrese sus datos, deberá dar click en Ingresar lo cual lo redirigirá al menú principal de la plataforma.



Vista del menú principal. A continuación se revisaran las diversas opciones del sistema.

Registro de Equipos

Como primera opción esta el módulo de equipos. En este el operador podrá ingresar datos de un nuevo equipo de la primera división del campeonato, así como actualizar información respecto a ese o equipo o eliminarlo del sistema en caso que dicho equipo no siga presente en la serie A.



El botón superior mostrara al operador la pantalla de registro del nuevo equipo.

En la parte media están presentes los filtros de búsqueda de información. Estos se adaptan a cada columna presente en cada interfaz, con ellos se pueden hacer búsquedas para agilizar la información, estos filtros están presentes en todas las opciones de la plataforma.

En la parte inferior hay dos botones: Volver al menú principal, el cual mostrara al usuario el menú principal y el botón Cerrar sesión en caso que el operador ya termine su trabajo y deba salir del sistema. Estos dos botones están presentes en todas las opciones de la plataforma para mayor accesibilidad al operador.

Finalmente a la derecha están dos botones de utilidad: Exportar, el cual permite guardar el contenido actual en formato csv y el botón imprimir para su respectivo uso.

La siguiente pantalla es la que se muestra al agregar un nuevo equipo

The screenshot displays a web browser window with the title 'Administración de Equipos'. The address bar shows 'intranet.tecisl2.com:8888/admin Equipos/show_admin Equipos/ajf'. The main content area is titled 'Administración de Equipos' and contains a form labeled 'Agregar Equipos'. The form has the following fields: 'Equipo' (text input with 'Club 1'), 'Director Técnico' (text input with 'Técnico 1'), 'Estadio' (text input with 'Estadio 1'), and 'Escudo' (dropdown menu with a shield icon). Below the form are three buttons: 'Guardar', 'Guardar y volver a la lista', and 'Cancelar'. At the bottom left of the form area, there are two more buttons: 'Volver al menú principal' and 'Cerrar sesión'. The 'Guardar' and 'Guardar y volver a la lista' buttons are circled in red in the image.

Aquí se ven los siguientes campos: Equipo, el cual es el nombre del equipo a registrar. Director técnico, Estadio y un control que permite elegir el escudo del equipo. Cabe recalcar que los 3 primeros campos son obligatorios y en caso de dejarlos el blanco el sistema mostrara una advertencia y no dejara continuar el proceso de registro. Así mismo estan presente dos botones en la parte inferior, volver al menú principal y cerrar sesión.

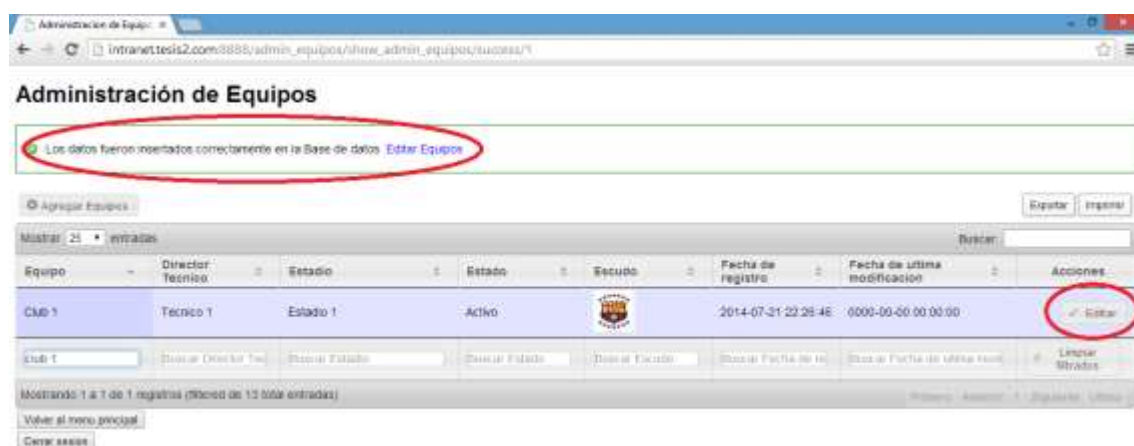
Una vez ingresado los datos, hay dos opciones de guardado:

Guardar, al elegir esta opción el registro se efectúa y se mantendrá en la misma pantalla en caso que se desee ingresar otro nuevo equipo.

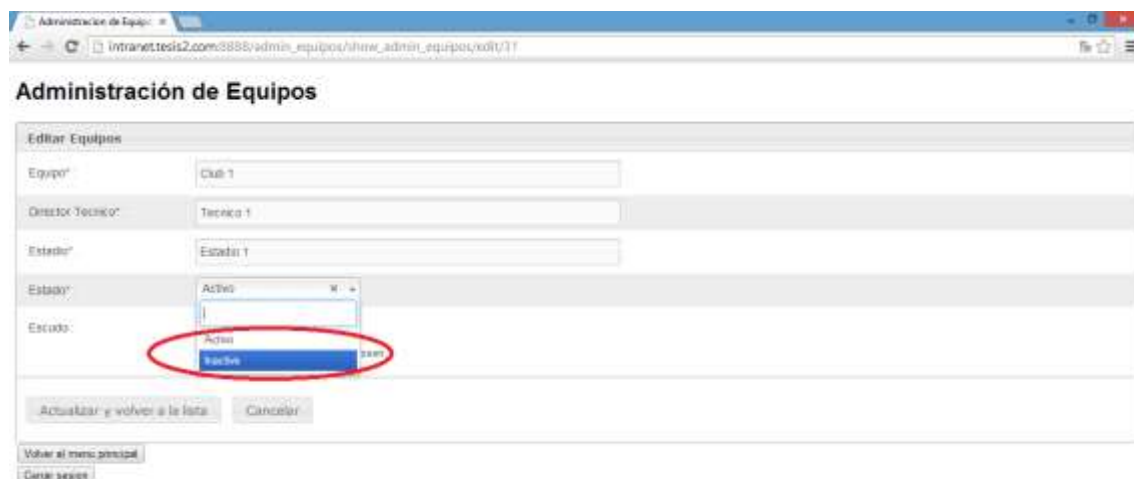
Guardar y volver a la lista, esta opción guardar el registro y redirige a la página donde se muestra la lista de los equipos registrados.

El botón cancelar anula la operación, muestra un mensaje de confirmación al usuario y si elige cancelar regresa a la lista principal, caso contrario permanece en el formulario de ingreso de datos.

Si todo marchó bien, saldrá esta pantalla mostrando un mensaje de éxito en la parte superior que indica que los datos fueron ingresados correctamente.



En la parte derecha resaltada esta el botón “Editar” el cual carga en pantalla los datos del equipo para poder actualizar su información en caso que se requiera.



Algo importante al editar la información de un equipo es el campo “estado”, si esta en “Activo” significa que el equipo será tomado en cuenta por el sistema para el actual campeonato, es decir asume que esta en la serie A, caso contrario si esta “Inactivo” significa que el equipo no esta en la serie A y no será tomado en cuenta por el sistema.

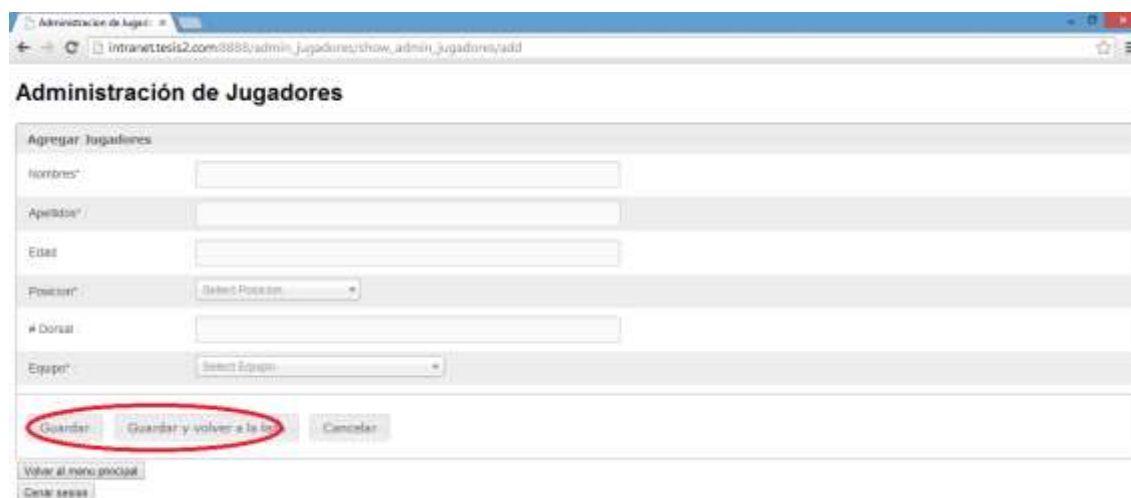
Ningún equipo ni jugador ni partidos ni resultados se podrá borrar, ya que la plataforma almacenara información estadística a través del tiempo.

Registro de Jugadores

Esta opción es similar a la anterior y aquí se ingresaran los datos de los jugadores de cada equipo del campeonato.

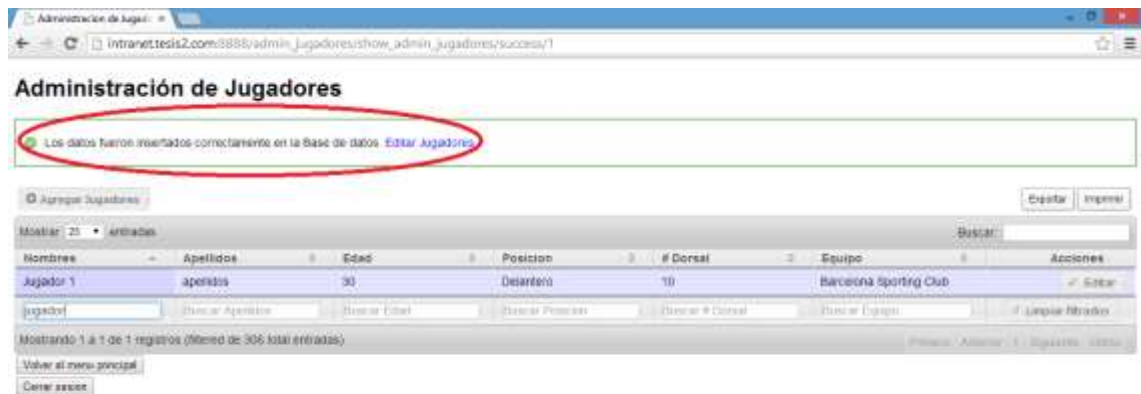


Al dar click en “Agregar Jugadores” se mostrara la pantalla para el ingreso de la información.



Los campos marcados con asteriscos son obligatorios. A cada jugador se le asigna el equipo en el cual juega y también su número de dorsal.

Si todo sale bien, sale el mismo mensaje que apareció al registrar el equipo. Así mismo están presentes las opciones para editar el registro del jugador o para borrar sus datos del sistema.



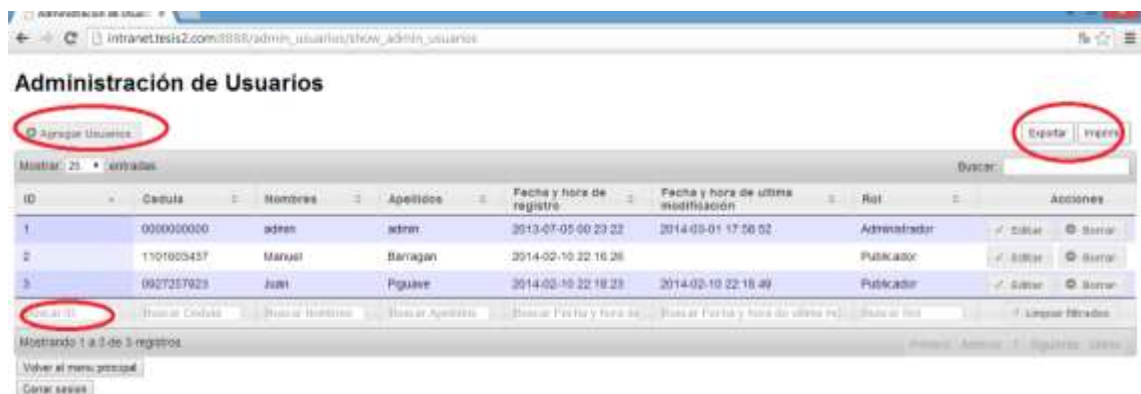
Creación de Usuarios y Privilegios

En la opción “Usuarios”, existen por defecto dos roles definidos:

Administrador: este rol permite el acceso a todas las opciones del sistema.

Publicador: este rol permite el acceso a las opciones del sistema a excepción del módulo de usuarios.

Solo los usuarios administradores pueden crear nuevos usuarios. Los usuarios publicadores solo se encargaran del manejo del contenido futbolístico.



Se presenta aquí una lista con 3 usuarios registrados, dos de ellos con rol Administrador. Para registrar un nuevo usuario debe dar click en “Agregar usuarios”.

Administración de Usuarios

Agregar Usuarios

Cedula*

Password*

Nombre*

Apellido*

Email

Telefono

Celular

Direccion

Rol:

Se presentan todos los campos en los cuales se ingresan los datos del nuevo usuario. El campo cedula servirá como “nombre de usuario” al momento de ingresar al sistema, los campos marcados con asterisco son requeridos y no se los debe dejar en blanco.

En rojo están resaltados los roles predefinidos. Si todo marcha bien, se podrá ver el mensaje de éxito, confirmando el ingreso de la información.

Administración de Usuarios

Los datos fueron insertados correctamente en la Base de datos. [Editar Usuarios](#)

Mostrar 25 entradas

ID	Cedula	Nombre	Apellido	Fecha y hora de registro	Fecha y hora de última modificación	Rol	Acciones
1	000000000	admin	admin	2013-07-05 00:23:23	2014-05-01 17:50:52	Administrador	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
2	110100437	Manuel	Barragan	2014-02-10 22:16:25		Publicador	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
3	097207923	Juan	Piguine	2014-02-10 22:16:23	2014-02-10 22:16:46	Publicador	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
4	0229054367	User 1	default	2014-07-21 22:59:14		Publicador	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>

Mostrando 1 a 4 de 4 registros

Módulo de Partidos

El módulo de partidos permite visualizar encuentros que se disputan en cada fecha del campeonato. Aquí no hay opción para agregar partidos, estos se generan automáticamente en el módulo “Calendario” en el menú principal.

Solamente se pueden editar el estadio donde se disputa el encuentro, árbitro en cada partido y el estado del partido (No jugado, 1er tiempo, 2do tiempo y Finalizado).

Modulo de Partidos

Buscar

Editar Imprimir

Temporada	Etapa	Jornada	Equipo local	Goles equipo local	Equipo visitante	Goles equipo visitante	Estado	Fecha y hora del partido	Estado del partido	Acciones
2014	1	Fecha # 20	Deportivo Guabo	2	Centro deportivo Omeo	1	Estadio Olimpico Atahuapla	16/07/2014 - 19:00	Finalizado	Editar
2014	1	Fecha # 20	Deportivo Cuenca	1	Club Sport Emelec	0	Estadio Alejandro Serrano Aguilar	16/07/2014 - 20:00	Finalizado	Editar
2014	1	Fecha # 20	Independiente Jose Teran	0	Universidad Catolica	0	Estadio General Rumiñahui	16/07/2014 - 16:00	Finalizado	Editar
2014	1	Fecha # 20	Mushat Rana Sporting Club	2	Marta Futbol Club	1	Estadio General Rumiñahui	16/07/2014 - 16:00	Finalizado	Editar
2014	1	Fecha # 20	Barcelona Sporting Club	4	Liga de Liga	0	Estadio Monumental Banco Pichincha	19/07/2014 - 16:45	Finalizado	Editar
2014	1	Fecha # 20	Club Deportivo El Nacional	0	Liga Deportiva Universitaria	1	Estadio Olimpico Atahuapla	20/07/2014 - 12:00	Finalizado	Editar

Mostrando 1 a 6 de 6 registros (filas de 130 total entradas)

Volver al menu principal

Cerrar sesion

Se muestran las columnas “Temporada” y “Etapa”, las cuales pueden ser usadas como filtros para búsquedas de partidos pasados. En este caso se muestra información de ejemplo, filtrados por la jornada # 20. A la derecha esta el estado del partido y el boton “Editar”.

Módulo de Resultados

En el módulo de resultados se pueden visualizar la información de cada encuentro cuando finaliza.

Modulo de Resultados

Buscar

Editar Imprimir

Temporada	Etapa	Jornada	Equipo local	Goles equipo local	Equipo visitante	Goles equipo visitante	Estado	Fecha y hora del partido	Estado del partido
2014	1	Fecha # 20	Deportivo Guabo	2	Centro deportivo Omeo	1	Estadio Olimpico Atahuapla	16/07/2014 - 19:00	Finalizado
2014	1	Fecha # 20	Deportivo Cuenca	1	Club Sport Emelec	0	Estadio Alejandro Serrano Aguilar	16/07/2014 - 20:00	Finalizado
2014	1	Fecha # 20	Independiente Jose Teran	0	Universidad Catolica	0	Estadio General Rumiñahui	16/07/2014 - 16:00	Finalizado
2014	1	Fecha # 20	Mushat Rana Sporting Club	2	Marta Futbol Club	1	Estadio General Rumiñahui	16/07/2014 - 16:00	Finalizado
2014	1	Fecha # 20	Barcelona Sporting Club	4	Liga de Liga	0	Estadio Monumental Banco Pichincha	19/07/2014 - 16:45	Finalizado
2014	1	Fecha # 20	Club Deportivo El Nacional	0	Liga Deportiva Universitaria	1	Estadio Olimpico Atahuapla	20/07/2014 - 12:00	Finalizado

Mostrando 1 a 6 de 6 registros (filas de 130 total entradas)

Volver al menu principal

Cerrar sesion

Aquí no se pueden editar estos registros, son de solo lectura y se transfieren aquí cuando se finaliza el partido correspondiente.

Módulo de Tabla de Posiciones

En este módulo se visualiza la tabla de posiciones la cual va cambiando de acuerdo a los resultados que se vayan generando con los partidos del campeonato.

The screenshot shows a web browser window with the URL `intranet.tesis2.com:8888/admin_posiciones/show_admin_posiciones`. The page title is "Tabla de Posiciones". The table has the following columns: Temporada, Etapa, Equipo, Partidos jugados, Goles a favor, Goles en contra, Gol diferencia, and Puntos. The data rows are for the year 2014, with 1 stage. The teams and their statistics are as follows:

Temporada	Etapa	Equipo	Partidos jugados	Goles a favor	Goles en contra	Gol diferencia	Puntos
2014	1	Club Sport Emelec	20	33	13	+20	41
2014	1	Independiente Jose Teran	20	32	17	+15	36
2014	1	Liga de Loja	20	24	29	-5	33
2014	1	Barcelona Sporting Club	20	23	17	+6	31
2014	1	Liga Deportiva Universitaria	20	20	19	+1	29
2014	1	Universidad Católica	20	25	25	0	28
2014	1	Deportivo Guano	20	19	19	0	25
2014	1	Centro deportivo Olmedo	20	20	23	-3	24
2014	1	Club Deportivo El Nacional	20	23	34	-11	23
2014	1	Mushuc Runa Sporting Club	20	17	24	-7	20
2014	1	Deportivo Cuenca	20	19	27	-8	20
2014	1	Manta Fútbol Club	20	18	26	-8	20

At the bottom of the table, there are several filter buttons: "Mostrar Temporada", "Mostrar Etapa", "Mostrar Equipo", "Mostrar Partidos jugados", "Mostrar Goles a favor", "Mostrar Goles en contra", "Mostrar Gol diferencia", and "Mostrar Puntos". A search bar is located at the top right of the table area.

En este caso se cuenta con informacion de ejemplo hasta la fecha # 20, asi mismo se muestra los goles a favor, en contra, gol diferencia y los puntos que acumula el equipo.

Aquí también se puede filtrar la informacion de acuerdo a la columna que se desea utilizar como filtro.

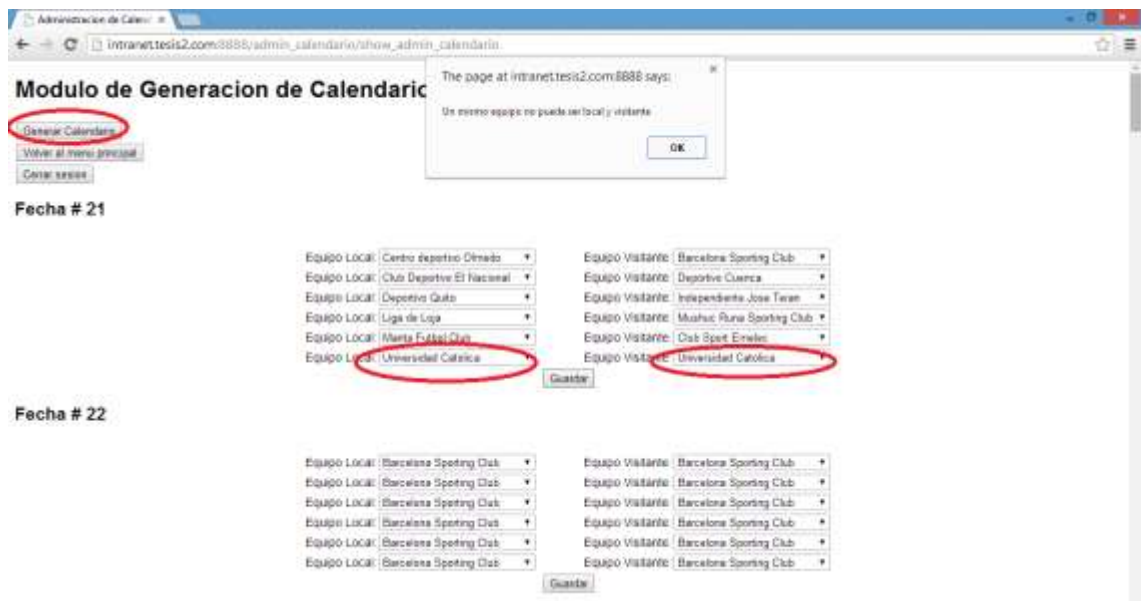
Módulo de Calendario

En este módulo se presenta el calendario completo del campeonato y los controles para poder generar los partidos. Cabe recalcar que el número de partidos del campeonato y el número de partidos por campeonato se pueden configurar en el módulo de "Parámetros" para que el sistema se ajuste al sistema de campeonato actual.

En cada fecha se pueden emparejar los encuentros. Del lado izquierdo están cargados los equipos que jugaran de locales y en la derecha están los equipos que jugaran de visitante.

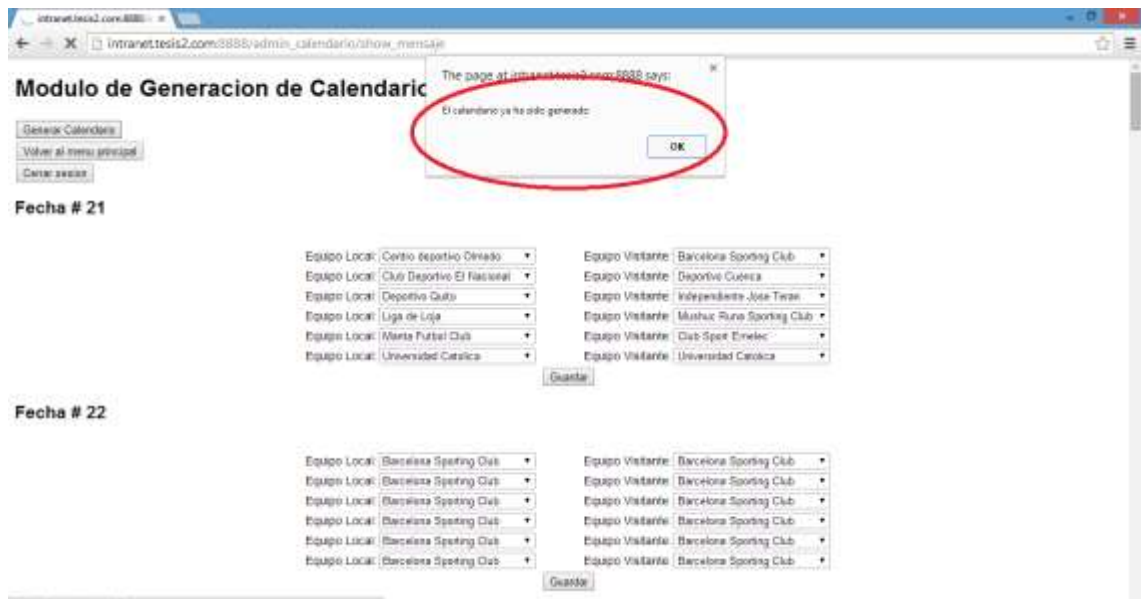


Si de manera errónea elige a un mismo equipo como local y visitante, el sistema le lanza la respectiva advertencia.



Esto evita obviamente generar informacion inconsistente.

El botón enmarcado en la izquierda superior, “Generar calendario”, se encarga de generar el calendario para el año actual con el número de etapas y fechas que están guardadas en módulo de parámetros. Si intenta generar de nuevo el calendario en la actual temporada, el sistema le mostrara un mensaje indicando que el calendario ya ha sido generado.



Módulo de Parámetros

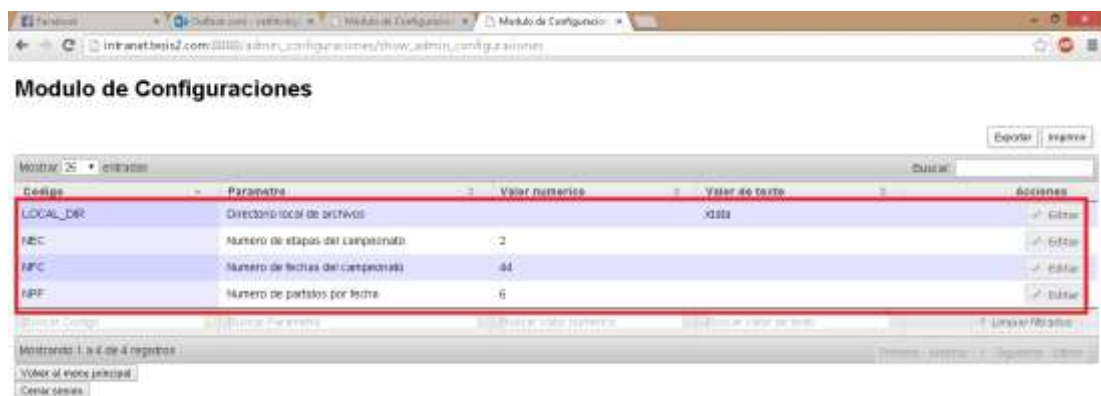
En el módulo de parámetros (o configuraciones) están los valores preestablecidos.

LOCAL_DIR: es el directorio local en el cual se almacenan los archivos que alimentaran la aplicación cliente.

NEC: número de etapas del campeonato.

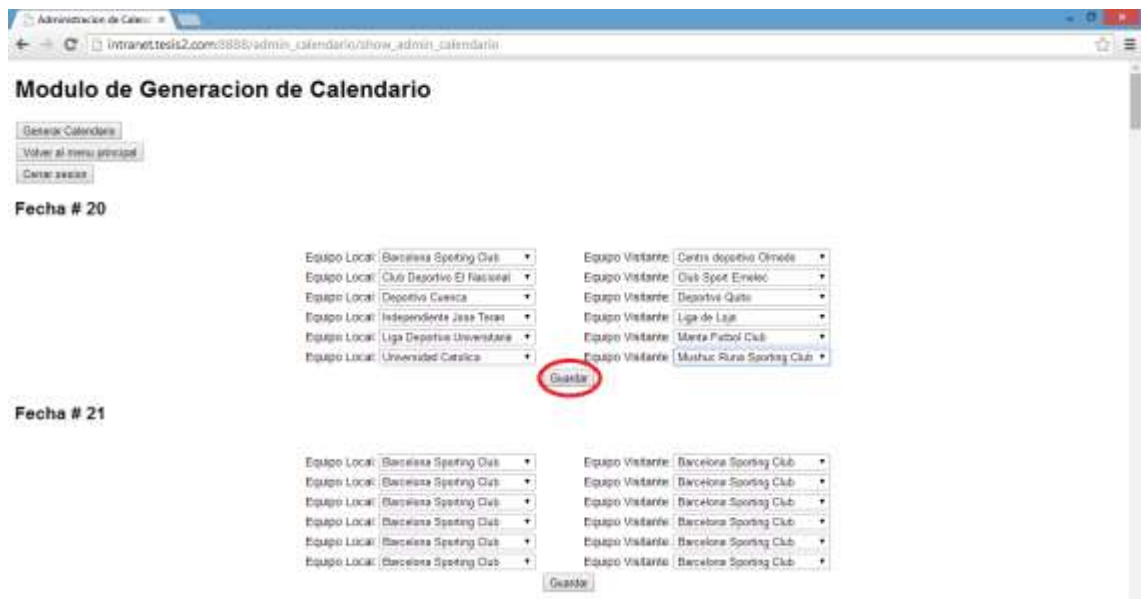
NFC: es el número de fechas del campeonato.

NPF: es el número de partidos por fecha.



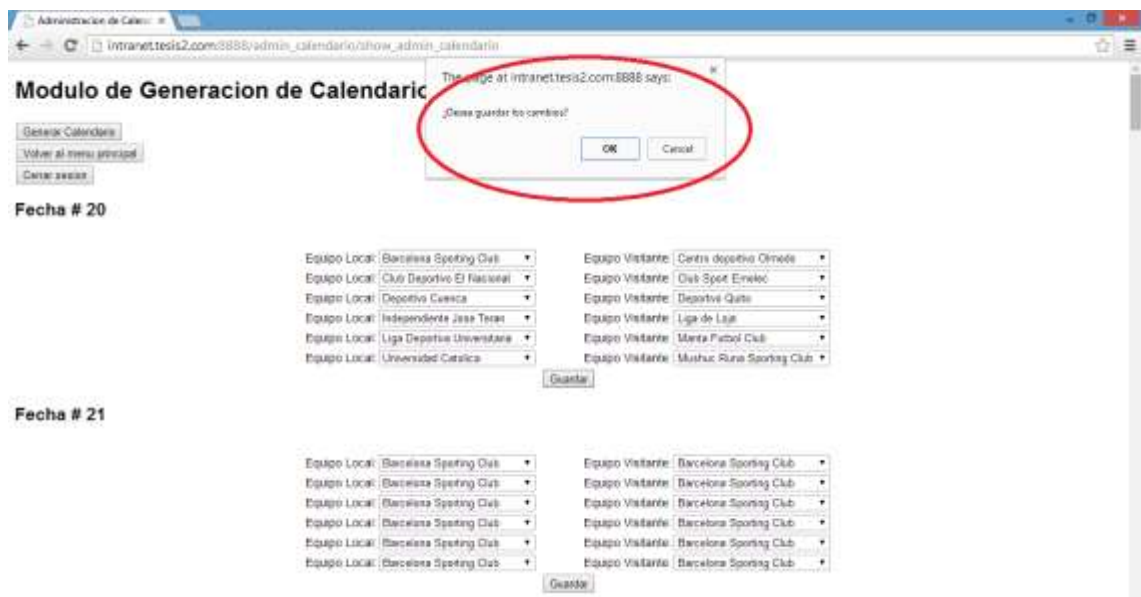
Módulo de Goles y Tarjetas

Para poder utilizar este módulo, primero es necesario generar los partidos en el módulo de calendario. Una vez que se genera los partidos correspondientes a la fecha actual del campeonato, en el módulo de goles y tarjetas aparecerán las opciones para poder ingresar los goles y las tarjetas en cada partido y también el minuto en el cual sucedió.



En este ejemplo, se van a generar los partidos correspondientes a la jornada # 20, elegimos los equipos locales y visitantes. Para terminar el proceso, dar click en el botón “Guardar”.

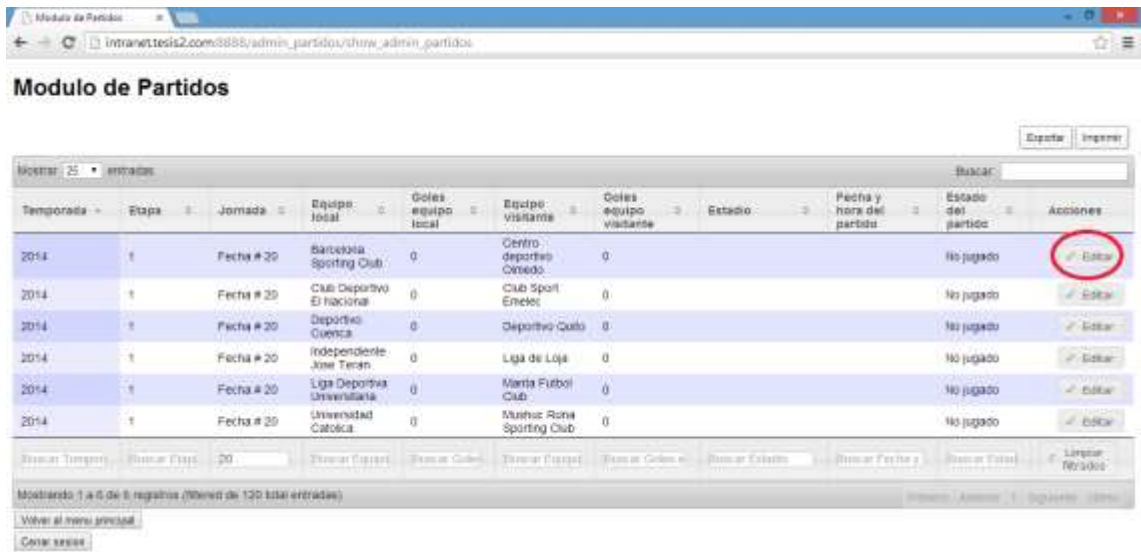
Primero le preguntara si desea guardar los cambios, dándole opción a cancelar y revisar en caso que desee corregir y cambiar los equipos que se encuentran emparejados.



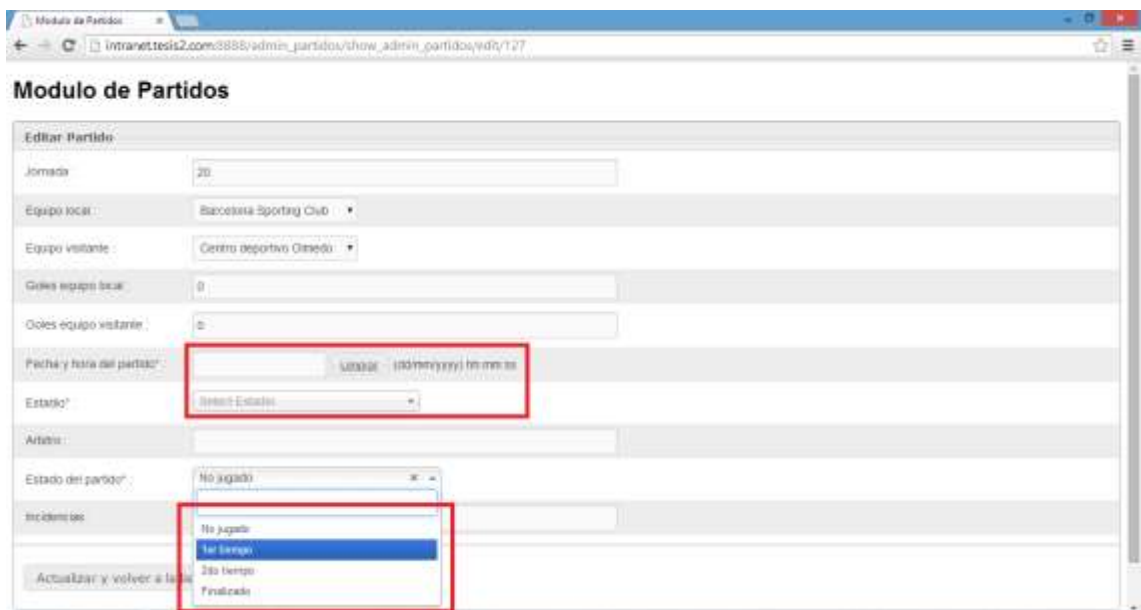
Si desea guardar click en “OK”, caso contrario en “Cancel”.

Una vez guardado lo anterior y antes de generar los goles y tarjetas, es necesario cambiar el estado del partido. Por defecto cuando son generados tienen estado “No

jugado”, para poder cambiar este estado se debe dirigir al módulo de partidos. Ahí se mostraran los partidos de la fecha actual.



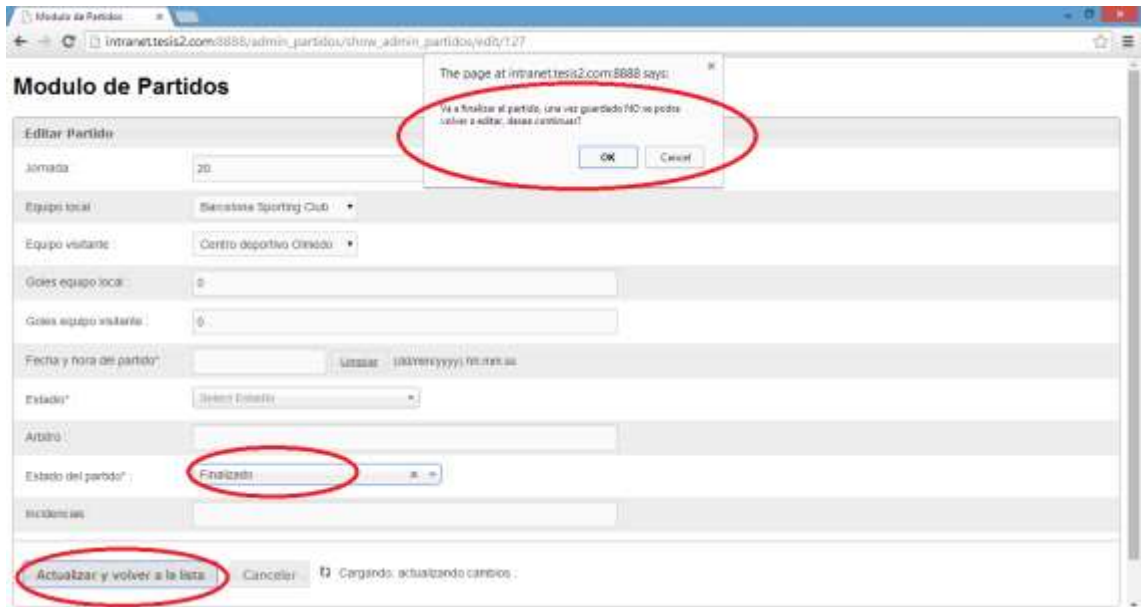
Debe dar click en donde dice “Editar”, le mostrara esta siguiente pantalla:



Le muestra los nombres de los equipos que se enfrentan y resaltado en rojo están los campos que son editables. Fecha y hora del partido, al dar click en la caja de texto se desplegara un calendario en donde podrá elegir la hora y fecha del encuentro, podrá también digitar el nombre del árbitro del encuentro y abajo podrá cambiar el estado del partido. Al cambiarlo a 1er tiempo o segundo tiempo, el partido podrá ser modificado en el módulo de goles y tarjetas, caso contrario si usted cambia el estado a “Finalizado” le mostrara un mensaje de advertencia indicándole que el partido ya no será modificable y se mantendrá con los valores que haya puesto el operador. Mas

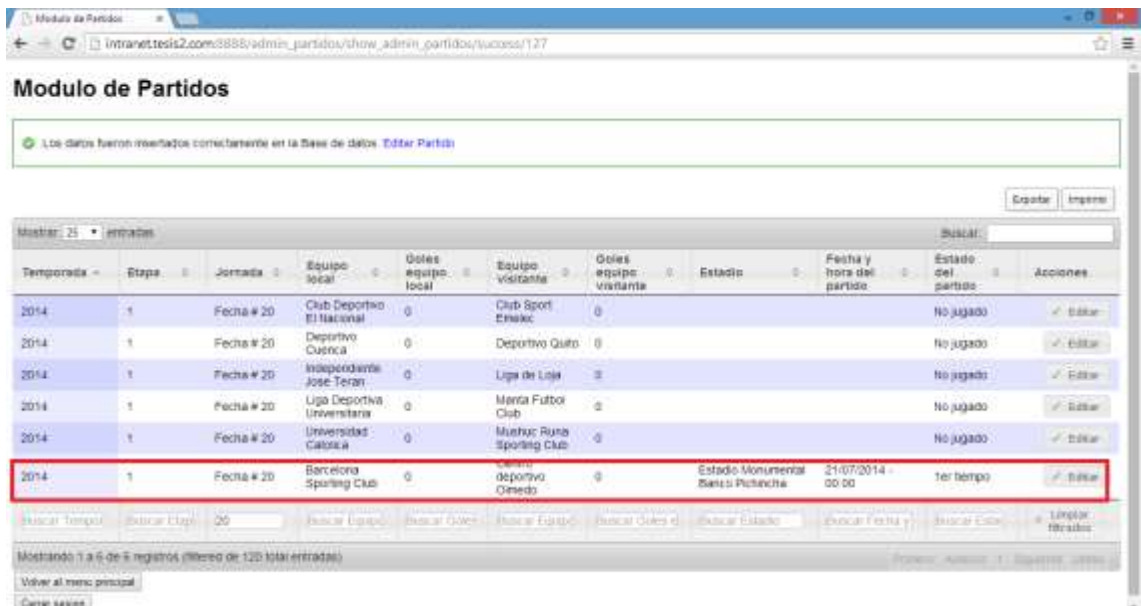
abajo hay una caja de texto que dice “Incidencias”, este campo es opcional y se lo puede dejar en blanco.

Este es el mensaje de confirmación para finalizar el partido. Para confirmar click en “Ok”, caso contrario click en “Cancel”.



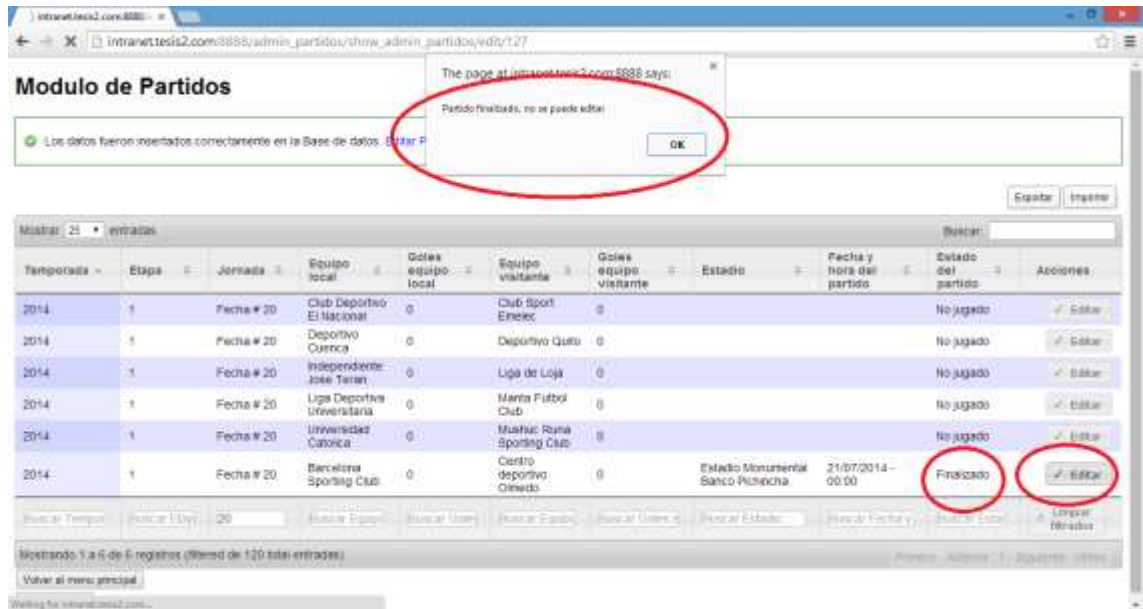
El botón “Actualizar y volver a la lista” sirve solo cuando no se desea finalizar el partido pero si se desea cambiar la fecha, el estadio donde se juega o el árbitro que dirige el cotejo (opcional).

Si todo sale bien, le debería salir una pantalla como esta.



El estado del partido ya ha pasado a “1er tiempo” y el mensaje de confirmación sobre los cambios guardados.

Si intenta editar un partido finalizado, el sistema NO se lo permitirá y le mostrara el siguiente mensaje.




Regresando al módulo de goles y tarjetas, debería salir una pantalla como esta.




En la parte superior izquierda se muestra la jornada actual que se esta jugando.

Los partidos que han sido finalizados no se muestran, ni tampoco los que tienen el estado “No jugado”. (Si no se generan partidos o si hay partidos guardados pero ninguno con estado igual a primer tiempo, este módulo saldrá vacío).

Los botones de color rojo y amarillo sirven para marcar las amonestaciones de los jugadores tanto del equipo local (izquierda) como visitante (derecha).

La imagen del balón  sirve para registrar un gol al jugador seleccionado de acuerdo al equipo.

La imagen del balo con el símbolo de prohibición  sirve para registrar un autogol hecho por el jugador seleccionado de acuerdo al equipo.

Al lado derecho esta la columna “Minuto” en la cual el operador debe elegir el minuto en el cual se desea registrar la acción del partido (gol, autogol, tarjeta roja, amarilla). Y también una columna de estado del partido.

Cuando el operador desee registrar cualquier acción (gol, autogol, tarjeta roja, amarilla), el sistema le preguntara si desea guardar los cambios, dándole la opción de cancelar la acción y revisar de nuevo cuidadosamente si eligió al jugador que marco el gol o si desea escoger la amonestación correcta.

Aquí se intenta conceder un gol al jugador Adrian Bone del equipo de El Nacional, si da click en OK se registra el gol y se incrementa el contador de goles del equipo local, de igual manera aplica lo mismo para el equipo visitante. Para marcar una amonestación, se elige al jugador del equipo que recibió la amonestación y dependiendo de la misma se da click en el botón amarillo o rojo (los del lado izquierdo son para los jugadores del equipo loca, los del lado derecho para los jugadores del equipo visitante).



The screenshot shows a web browser window with the URL `intranettesis2.com:8888/admin_goles_tarjetas/show_admin_goles_tarjetas`. The page title is "Modulo de Goles y Tarjetas". A dialog box is open, asking "¿Desea guardar los cambios?" with "OK" and "Cancel" buttons. Below the dialog is a table with the following columns: "Equipo Local", "Jugadores Equipo Local", "Amonestacion", "Equipo Visitante", "Equipo Visitante", "Minuto", and "Estado". The table contains data for several teams and players, including Adrian Bone, Aljondra Pizzotti, Andres Lamas, Alcaides Dominguez, Alejandro Espinoza, and others.

Equipo Local	Jugadores Equipo Local	Amonestacion	Equipo Visitante	Equipo Visitante	Minuto	Estado
Club Deportivo El Nacional	Adrian Bone	0	Club Sport Emelec	Club Sport Emelec	1	1er tiempo
Deportivo Cuenca	Aljondra Pizzotti	0	Deportivo Quito	Deportivo Quito	1	1er tiempo
Independiente Jose Terán	Andres Lamas	0	Liga de Loja	Liga de Loja	1	1er tiempo
Liga Deportiva Universitaria	Alcaides Dominguez	0	Maria Futbol Club	Maria Futbol Club	1	1er tiempo
Universidad Católica	Alejandro Espinoza	0	Musfik Rona Sporting Club	Musfik Rona Sporting Club	1	1er tiempo

Si se confirma la operación, se muestran los cambios en el mismo módulo.

Modulo de Goles y Tarjetas

Fecha # 20

Equipo Local	Jugadores Equipo Local	Amonestaciones:	Resultados	Amonestaciones:	Jugadores Equipo Visitante	Equipo Visitante	Minuto	Estado
Club Deportivo El Nacional	Adrián Bone	 	1 - 0	 	Ángel Mesa	Club Sport Emelec	1	1er tiempo
Deportivo Cuenca	Alejandro Pazos	 	0 - 0	 	Alejandro Pansa	Deportivo Quito	1	1er tiempo
Independiente José Terán	Andrés Lamas	 	1 - 0	 	Amaranto Gomez	Liga de Loja	1	1er tiempo
Liga Deportiva Universitaria	Alexander Dominguez	 	0 - 0	 	Andrés Valverde	Manta Fútbol Club	1	1er tiempo
Universidad Católica	Alejandro Espinosa	 	0 - 0	 	Bryan Rodríguez	Mushut Rana Sporting Club	1	1er tiempo

Ver al inicio anterior

Centrar pantalla

En este caso el gol fue concedido a Adrián Bone.

Sincronización

La opción de sincronización es muy importante. Mediante este proceso el sistema genera archivos de texto con la información actualizada (resultados, goleadores, próximos encuentros, tabla de posiciones) y estos archivos se almacenan en el directorio especificado dentro del servidor.

El servidor a su vez transmitirá estos archivos junto con una aplicación cliente (desarrollada aparte y que no forma parte de este manual) mediante el espacio aéreo hasta el receptor digital en cada hogar que sintonice el canal.

Una vez que el receptor logre recibir la aplicación cliente junto con los archivos de texto, el televidente (usuario final) podrá estar informado sobre las novedades en el campeonato de fútbol.