MIDDLEWARE GINGA EN EL DESARROLLO DE APLICACIONES INTERACTIVAS PARA LA TELEVISIÓN DIGITAL TERRESTRE. APLICATIVO: PROTOTIPO DE UN PORTAL INTERACTIVO PARA LA COMPRA Y VENTA DE PRODUCTOS POR TELEVISIÓN

Oscar Rodríguez.[1]

[1] Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte, Av. 17 de Julio 5-21, Ibarra, Imbabura

odrodriguez@utn.edu.ec

Resumen. *La señal digital terrestre en el Ecuador, es un tema poco conocido que ha tomado impulso desde abril del año 2009; en este año es cuando se empieza a analizar los diferentes estándares de transmisión y por consiguiente se toma la decisión de adaptar el estándar brasileño ISDB-Tb, que es una variación del estándar japonés. Este proceso de cambio permite que se abran nuevas puertas para los desarrolladores de software, como lo es la programación de aplicaciones para television digital.*

Una de las herramientas que está tomando auge es el GINGA, la cual es un grupo de tecnologías con mejoras brasileñas que enlazan el sistema operativo con la parte de hardware, permitiendo realizar aplicaciones que interactúan con la señal transmitida al televisor, independientemente de la plataforma de hardware que manejen los fabricantes. Además, es un middleware que maneja diversos tipos de lenguajes y librerías de funciones que permiten el desarrollo fácil y rápido de aplicaciones interactivas.

Palabras Claves

Ginga, Middleware, Interactividad, LUA, NCL, JAVA.

Abstract. *The digital terrestrial signal in Ecuador, is a little known theme that has taken impulse since April of 2009; In this year is when it begins to analyze the different standards of transmission and therefore the decision is made to adapt the Brazilian standard ISDB-Tb, which is a variation of the Japanese standard. This process of change allows new doors for software developers, such as the programming of applications for digital television.*

One of the tools that is gaining upgrade is the GINGA, which is a group of technologies with Brazilian improvements that link the operating system with the hardware part, allowing to realize applications that interact with the transmitted signal to the television, independently of the platform of Hardware that manufacturers handle. In addition, it is a middleware that handles different types of languages and function libraries that allow the easy and fast development of interactive applications.

Keywords

Ginga, Middleware, Interactivity, LUA, NCL, JAVA.

1. Introduction

In the province of Imbabura there is not enough information about interactive application development for digital terrestrial television, since the digital signal is not yet accessible in that province. This scientific article aims to provide a basic guide in the aspect of interactive applications for future generations of software developers, especially with the ginga tool since it allows many functionalities through its components.

Ginga was born in the country of Brazil as a project by the Pontificia Universidad Católica de Rio de Janeiro. This middleware is born with this name because ginga is a unique and indescribable quality that distinguishes the Brazilian people, in addition it is a movement of the traditional dance capoeira; A dance that was born as a form of entertainment of the slaves, its movements are similar to a fight but without physical contact. Ginga was born as free software because of the need to realize social / digital inclusion in all households, allowing everyone to have access to learning, social services, information, among others through their television (López & Oleas, 2012, págs. 40-41-42-43).

The Ginga middleware is a layer of intermediary software that allows the development of interactive applications for DTT regardless of the hardware platform of the manufacturers and access terminals. It supports the development of applications either using a declarative, imperative paradigm or both. The two execution environments are required in fixed and portable receivers, whereas only the declarative environment is required in portable receivers (Comunidad Peruana de Ginga, 2015).

In this paper a study of the middleware ginga in the development of interactive applications for digital terrestrial television is carried out by means of the development of a prototype client-server system that will serve to demonstrate the true scope of the tool and also to compare the programming languages JAVA And NCL used by the middleware.

2. Materials y Methods

Ginga middleware refers to software that includes a set of services that allow the interaction between various applications, operating systems, hardware, networks, among others. The essential part of a middleware is that it works independently of the platforms, this refers to that it can integrate distributed applications in heterogeneous environments facilitating the design and development of the systems.

This article describes the programming languages used by ginga, return channel, and application interactivity.

2.1 NCL Language

NCL is a declarative language, which focuses on developing the problem by presenting a series of characteristics that indicate what must be done to reach the solution. One of the advantages is that the learning curve of this language is very small since any programmer finds it easy to be coupled to the development model of NCL.

The functionality of this language is divided into media objects, regions, descriptors, ports, links and connectors; which are observed in the syntax shown in figure 1.

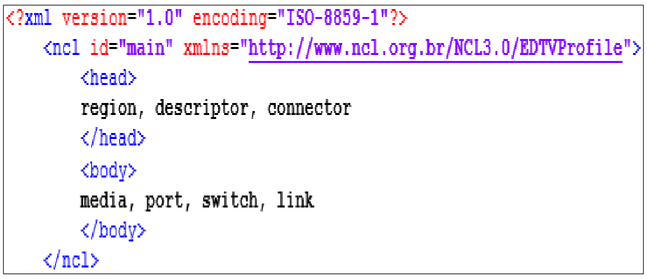
The media objects are defined as the part where the content is displayed, this content can be audio, video, text, images, among others.

The next step is to define the areas where the media objects will be displayed, these areas are defined by the region label.

The descriptors are responsible for displaying the assigned objects. This tag is responsible for giving parameters to the stockings.

The ports allow you to start the presentation of the media objects, in addition it gives a presentation order.

The Links and Connectors allow to control what is going to be executed by means of conditions and actions.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
    <ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
        <head>
        region, descriptor, connector
        </head>
        <body>
        media, port, switch, link
        </body>
    </ncl>
```

Figure 1. NCL Language Syntax
Source. CreaTV Digital, s.f.

2.2 LUA Language

LUA is a language developed in the laboratories of the Pontifical Catholic University of Rio de Janeiro with the main characteristic of helping other languages that emerged in the same site as for example Ginga-NCL. It is for such reason that LUA is not such a robust language and should be embedded in another to be compiled.

LUA has been used in some areas of programming such as the creation of video games or robotics because it is a language that can work in various systems and is also free code.

In order to make connections between LUA and NCL easier, new features were developed by LUA, such as the NCLua object, which also has several events that can be triggered by the remote control of a television. In this way it is possible to reinforce the part of the view that is handled by NCL.

There are three main modules that help to enhance interactivity, the first is the Event module allows NCLua to generate remote control or interactivity events in the LUA document.

The second is the settings module that allows interacting with the NCL document by means of the "application / x-ncl-settings" property that is declared in the interactive application view.

The third is the canvas module that provides functionalities for interactive digital television systems allowing you to graph and create new objects to be displayed in some assigned region in NCLua.

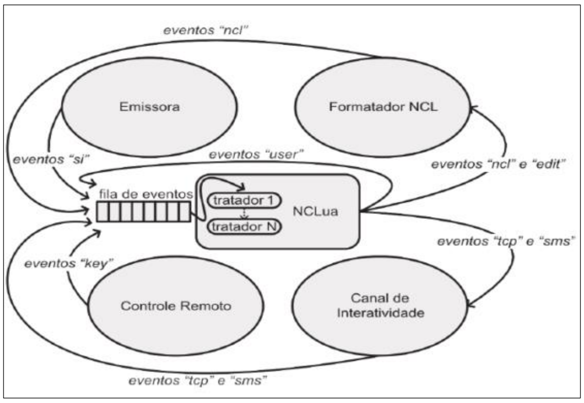


Figure 2. Event-Oriented Programming
Source. (Riberi F. , 2012)

2.3 JAVA Language

This is one of the languages with its well-founded bases, since the language java is not new, its beginnings begin at the beginning of the 90 in charge of the company Sun Microsystems; after a time this company would be bought by Oracle Corporation. To understand Java as a programming language, you must first understand the basic concepts of such technology.

This technology has gained a large place in the systems development market as it has been the basis for very large projects. Currently, it is a priority to install Java Virtual Machine (JVM), in order to run systems developed with java technology.

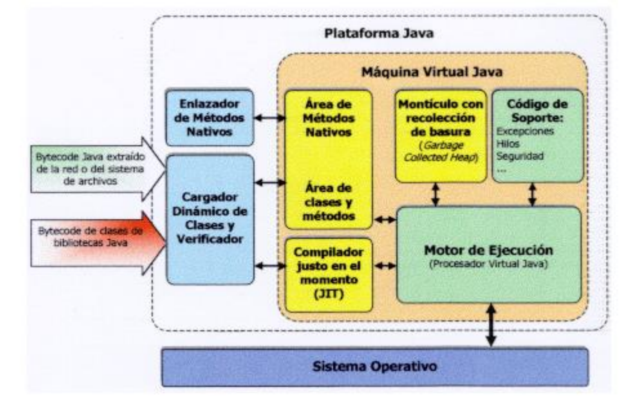

Plataforma Java
Máquina Virtual Java
Enlazador de Métodos Nativos
Área de Métodos Nativos
Montículo con recolección de basura
(Garbage Collected Heap)
Código de Soporte:
Excepciones
Hilos
Seguridad
...
Bytecode Java extraído de la red o del sistema de archivos
Bytecode de clases de bibliotecas Java
Cargador Dinámico de Clases y Verificador
Área de clases y métodos
Compilador justo en el momento (JIT)
Motor de Ejecución
(Procesador Virtual Java)
Sistema Operativo

Figure 3. Java Runtime System Architecture
Source. (Menchaca & García, s.f.)

2.4 Creating the System on the Server

As mentioned above, the prototype has two subsystems, both on the server and on the client respectively.

The System on the server will be developed with JAVA language, in turn you must create the web services that will be consumed by the client, these will be developed through the SOAP protocol.

For the development of the System first the entities that come from a Postgresql database are generated.

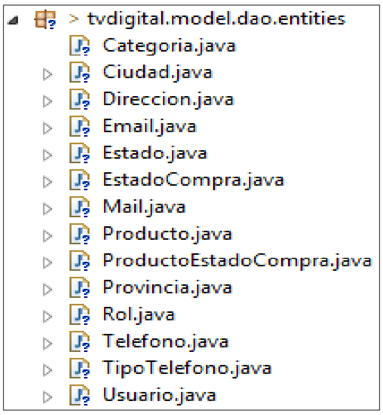

```
> tvdigital.model.dao.entities
    Categoria.java
  ▷ Ciudad.java
  ▷ Direccion.java
  ▷ Email.java
  ▷ Estado.java
  ▷ EstadoCompra.java
  ▷ Mail.java
  ▷ Producto.java
  ▷ ProductoEstadoCompra.java
  ▷ Provincia.java
  ▷ Rol.java
  ▷ Telefono.java
  ▷ TipoTelefono.java
  ▷ Usuario.java
```

Figure 4. Entity Package

The next step is to develop the handler, which will be in charge of performing direct queries to the database through the JPQL language.

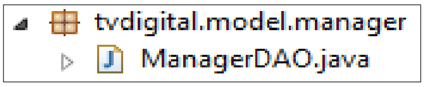

```
tvdigital.model.manager
  ▷ ManagerDAO.java
```

Figure 5. Manager Package

The following class pack consists of the Managed Bean, which are objects that will later be called from the xhtml pages.

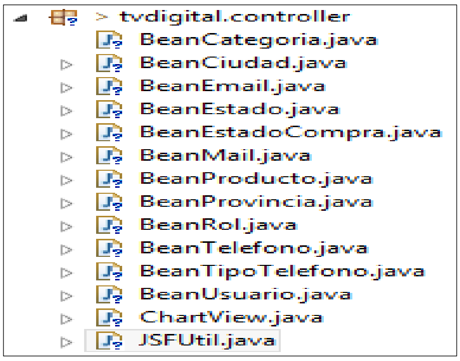
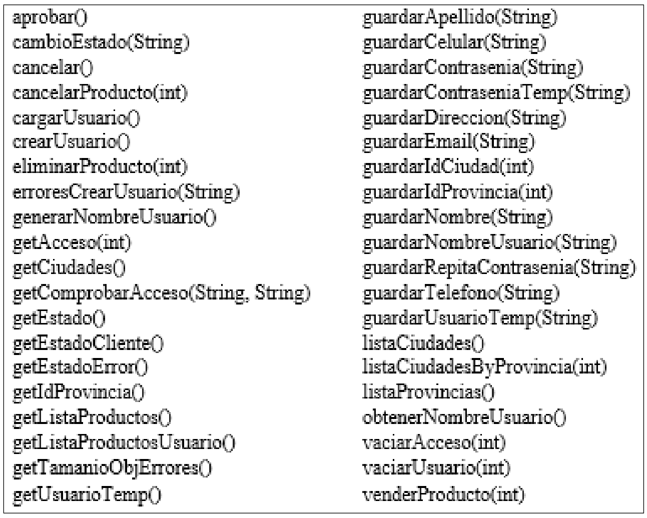

```
> tvdigital.controller
    BeanCategoria.java
  ▷ BeanCiudad.java
  ▷ BeanEmail.java
  ▷ BeanEstado.java
  ▷ BeanEstadoCompra.java
  ▷ BeanMail.java
  ▷ BeanProducto.java
  ▷ BeanProvincia.java
  ▷ BeanRol.java
  ▷ BeanTelefono.java
  ▷ BeanTipoTelefono.java
  ▷ BeanUsuario.java
  ▷ ChartView.java
  ▷ JSFUtil.java
```

Figure 6. Bean Manejados y Clases Util

Because the prototype will be developer with the four-layer architecture, web services are the component that layer services that will subsequently publish the methods so that they can be consumed by the client on the digital TV or decoder.



aprobar()
cambioEstado(String)
cancelar()
cancelarProducto(int)
cargarUsuario()
crearUsuario()
eliminarProducto(int)
erroresCrearUsuario(String)
generarNombreUsuario()
getAcceso(int)
getCiudades()
getComprobarAcceso(String, String)
getEstado()
getEstadoCliente()
getEstadoError()
getIdProvincia()
getListaProductos()
getListaProductosUsuario()
getTamanioObjErrores()
getUsuarioTemp()
guardarApellido(String)
guardarCelular(String)
guardarContrasenia(String)
guardarContraseniaTemp(String)
guardarDireccion(String)
guardarEmail(String)
guardarIdCiudad(int)
guardarIdProvincia(int)
guardarNombre(String)
guardarNombreUsuario(String)
guardarRepitaContrasenia(String)
guardarTelefono(String)
guardarUsuarioTemp(String)
listaCiudades()
listaCiudadesByProvincia(int)
listaProvincias()
obtenerNombreUsuario()
vaciarAcceso(int)
vaciarUsuario(int)
venderProducto(int)

Figure 7. Web Services Used in the Prototype of Purchase and Sale of Products by Digital Television.

2.5 EC2 de Amazon Server

For the publication of a web service on the Amazon EC2 platform, an instance was configured with the Linux operating system Ubuntu 14.04 LS.

Once installed the operating system proceeds to create a security group for the instance in which the following ports of exit and entrance for the Ubuntu virtual machine are opened:

- 80 (Port TCP)
- 8080 (Port TCP to tomcat)
- 5432 (Port to Postgresql)
- 22 (SSH to access to PUTTY)
- 443 (HTTPS)

Since the operating system installed in the instance only allows its access by console, we must establish a remote access by means of the putty tool in which we enter the public access of the created instance, the following details the access characteristics:

- Ubuntu (user)
- 54.91.186.81 (public ip to access)

In addition, the instance must have a public Key generated for the remote access, to acquire the key, we enter the created instance and choose the "Connect" option so that the key is then generated by means of a command line in Linux, the following are the commands that allow you to generate a key:

- ssh -i "ginga.pem" ubuntu@ec2-54-91-186-81.us-west-2.compute.amazonaws.com

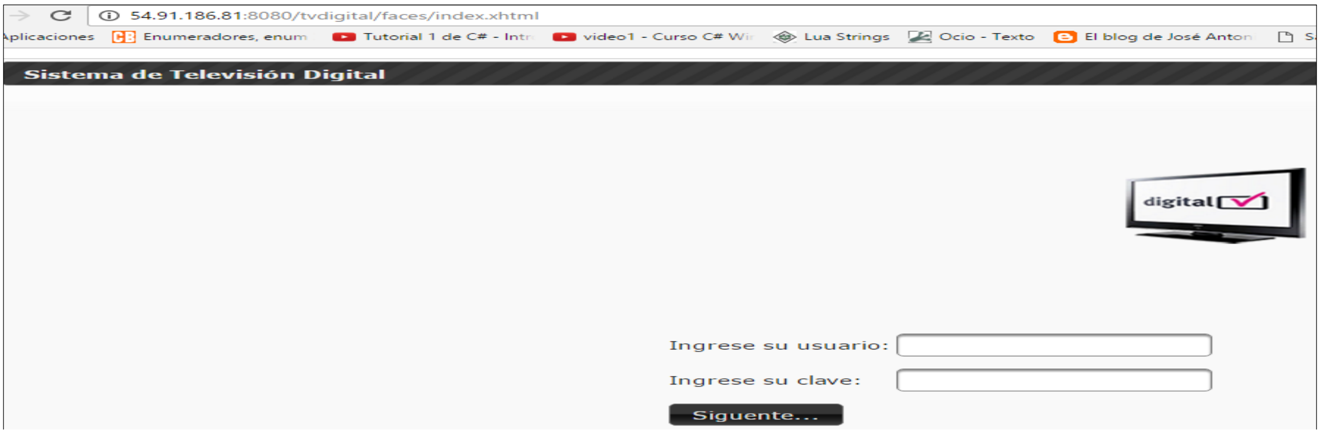
Now you must generate a file with the extension ".ppk" so that the putty program can read it as a key. To generate the file the program "puttygen" is used in which the file with extension ".pem" is entered to generate a private key with the extension ".ppk”

Then a java application is created with the help of the IDE eclipse in which a web service will be realized locally. Thanks to the facilities offered by AWS Amazon with its IaaS service, you only have to compress the locally developed application and copy it to the tomcat directory, usually it is /var/lib/tomcat7/webapps.

To access the application we must do it through port 8080, as shown at the following address:

- http://54.91.186.81:8080/tvdigital/faces/index.xhtml

Because the web service was generated in the java language it is imperative that you configure the Tomcat servlet container for public IP publishing of the instance created in EC2 as discussed above.



54.91.186.81:8080/tvdigital/faces/index.xhtml

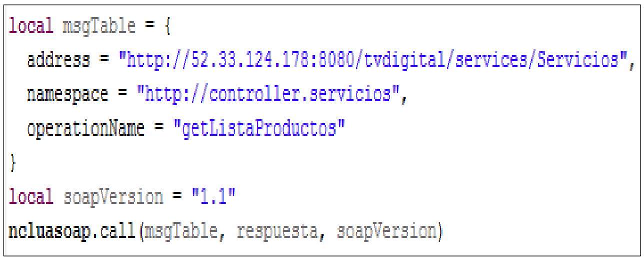
Sistema de Televisión Digital

Ingrese su usuario:
Ingrese su clave:

Figure 8. Main System Screen

2.6 Client System

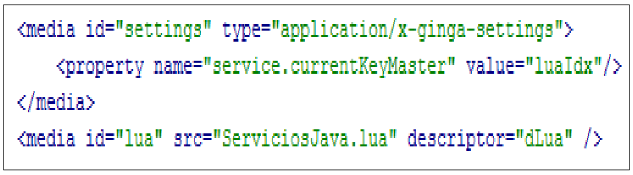
To consume the web services, it will be necessary to implement a class with the LUA language. This class will allow the consumption of published methods. This is done through functions that run on the client side. This class uses the ncluasoap library to call functions that allow the connection to the web services.



```
local msgTable = {
  address = "http://52.33.124.178:8080/tvdigital/services/Servicios",
  namespace = "http://controller.servicios",
  operationName = "getListaProductos"
}
local soapVersion = "1.1"
ncluasoap.call(msgTable, respuesta, soapVersion)
```

Figure 9. Consumption of a web service in LUA

Once the connection to the web services is made, the LUA class integrates with the NCL class to display the results by means of the <media> tag.

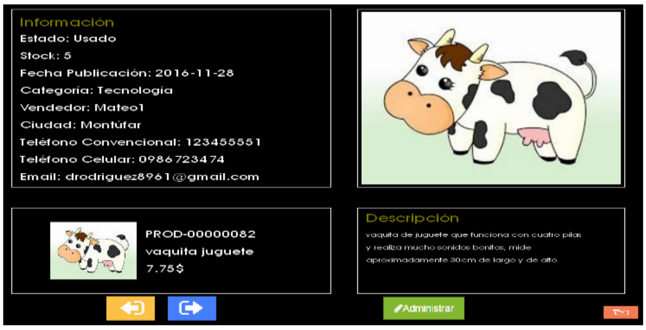


```
<media id="settings" type="application/x-ginga-settings">
    <property name="service.currentKeyMaster" value="luaIdx"/>
</media>
<media id="lua" src="ServiciosJava.lua" descriptor="dLua" />
```

Figure 10. Connection NCL and LUA

3. Results

The result of this research was successful since the server was connected to the client through the service layer. There were inconveniences in the tests with the physical decoder, but in the virtual decoder an interactive system was achieved with almost all the functionalities.



Información
Estado: Usado
Stock: 5
Fecha Publicación: 2016-11-28
Categoría: Tecnología
Vendedor: Mateo1
Ciudad: Montúfar
Teléfono Convencional: 123455551
Teléfono Celular: 0986723474
Email: drodriguez8961@gmail.com

PROD-00000082
vaquita juguete
7.75$

Descripción
vaquita de juguete que funciona con cuatro pilas y realiza mucho sonidos bonitos, mide aproximadamente 30cm de largo y de alto

Figure 11. Client System Main Screen

From the development of this prototype we obtained information about the tools that compose the ginga middleware.

Priority	Tool	Ginga	
		NCL	JAVA
1	Documentation		
	Amount of Documentation in Portuguese	4,5	2
	Amount of Documentation in Spanish	3	2,5
	Amount of Documentation in English	2	2
2	Development Environments	5	5
3	Support for S. O. (Linux, Microsoft)	5	5
4	Emulators	4,5	3
5	Facility of Installation	4	2
TOTAL		**28**	**21,5**
PERCENTAGE OF PERFORMANCE		**80%**	**61,43%**

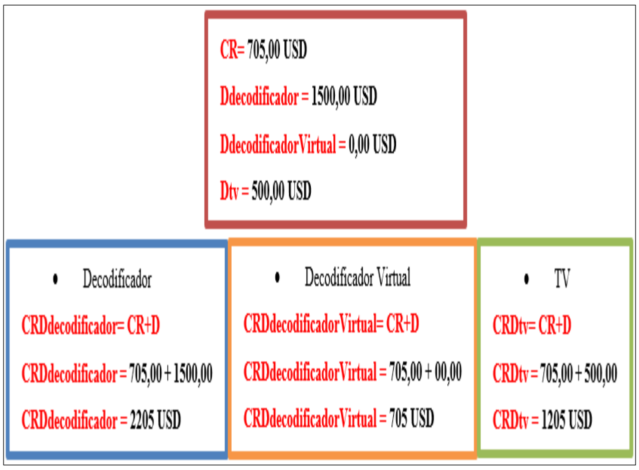
Table 1. Comparison table

A cost-benefit analysis was also performed based on the devices used. The following formula extracts the actual cost of the research depending on the device used.

$$CRD = CR + D \tag{1}$$

Where:

- CRD: Actual cost excluding the prices of digital TV, decoder and virtual decoder.
- Device (Decoder, Virtual Decoder, TV)
- CRD: Actual Cost Per Device (Decoder, Virtual Decoder, TV)



CR= 705,00 USD
Ddecodificador = 1500,00 USD
DdecodificadorVirtual = 0,00 USD
Dtv = 500,00 USD

- Decodificador

CRDdecodificador= CR+D
CRDdecodificador = 705,00 + 1500,00
CRDdecodificador = 2205 USD

- Decodificador Virtual

CRDdecodificadorVirtual= CR+D
CRDdecodificadorVirtual = 705,00 + 00,00
CRDdecodificadorVirtual = 705 USD

- TV

CRDtv= CR+D
CRDtv = 705,00 + 500,00
CRDtv = 1205 USD

Figure 12. Actual cost value depending on the device used

Device	Actual Cost ($)	Server Cost ($/MES)	Web Services Maintenance ($/MES)	Device Internal Memory (gb)
TV	1205	11,65	5	0
Physical Decoder	2205	11,65	5	2
Virtual Decoder	705	11,65	5	0,5

Table 2. Initial Values Table

VALUE OF CHARACTERISTICS	VMAX	PMAX	PM
Actual Cost	2205	10	3,1
Server Cost ($/MES)	11,65	10	10
Web Services Maintenance	5	10	10
Device Internal Memory (gb)	2	10	0

Table 3. Value of Features

Devices	Real Cost	Server Cost ($/MES)	Web Services Maintenance ($/MES)	Device Internal Memory (gb)	Total benefit	Device Price ($)	Cost/ Benefit
TV	5,4	10	10	0	25,4	500	0,2508
Physical Decoder	10	10	10	10	40	1500	0,093333
Virtual Decoder	3,1	10	10	2,5	25,6	1	125,6

Table 4. Cost-Benefit Analysis

The cost-benefit analysis was performed based on some factors that were the most relevant in the case study.

Once the initial values of each device were found, we calculated the maximum and minimum values of each characteristic to form Table 2 with the following formula:

$$PM = (VMIN * PMAX) / VMAX \qquad (2)$$

Where:

- PM: The minimum score is rounded to the smallest.
- VMIN: The minimum value of each column in table 1 of initial values
- PMAX: Maximum score that is granted, in this case will be a constant with a value of 10
- VMAX: The maximum value of each column in table 1 of initial values

Finally, the cost-benefit analysis table 3 is formed by means of the following formulas:

$$CR = (V * PMAX) / VMAX \qquad (3)$$

Where:

- CR: It is the actual cost score.
- V: value of each device
- PMAX: Maximum score that is granted, in this case will be a constant with a value of 10
- VMAX: The maximum value of each column in table 1 of initial values.

$$CB = (BT * 100) / V \qquad (4)$$

Where:

- CB: It is the cost-benefit value.
- BT: It is the total benefit that results from the sum of the characteristics
- 100: Constant
- V: The value of each Device.

As shown in Table 3, the highest benefit cost for prototype development is with the virtual decoder.

4. Conclusions

Ginga middleware is a tool with great capabilities for the development of interactive systems but is not yet well known by software developers since the digital signal in Ecuador has not yet fully covered the region and exists only in the main Cities.

The compatibility of ginga with imported televisions is very good, because most of these comply with the minimum characteristics that the middleware needs.

The NCL and JAVA languages used by the ginga middleware allow for interactive applications, but documentation is not enough because in Ecuador it is a new technology but in other countries applications are being implemented with other tools and ginga has stopped evolving.

There are several drawbacks in the realization of a comparison between NCL and JAVA since these two languages offer extraordinary potentialities working together, but separately they have similar drawbacks. The study can generate better results when the digital signal exists in the province of Imbabura and in turn throughout Ecuador.

The web application made with JSF generated some inconveniences that were solved at the same time, since the servlet container tomcat needs extra configurations to publish a web service from a server on the web.

The Postgresql database server is perfect for this type of applications because it has very good features and does not cause problems with the size of the data, even if they are very large. In addition, the Amazon AWS platform, used for hosting the web application, is compatible with Postgresql.

The web services generated by SOAP were very easy to implement but their operation is very basic since the published methods return an arrangement but not an object as such.

The NCL language in conjunction with LUA and JAVA enabled a prototype of buying and selling products on television that interacts with the viewer user, but with the evolution of technology, new languages appeared that allow for a more efficient and better development.

The times that are needed to make an interactive system for digital television are very broad. An inconvenience that causes setbacks and expands the delivery time of the system is the training, since the best information is in Portuguese, besides there is not enough documentation from reliable sources.

Although the NCL language is easily understandable and allows interactivity to be performed on the digital television, it can´t be used on its own, it must learn another language as it is LUA for the connections to the application developed in JSF.

In conclusion, interactive systems are the new path to be followed by software developers because digital television is still one of the main sources of entertainment for people worldwide, it is not the only one but one of the most used.

Gratitude

A special thank you to all the members of my family because they provide me with your company and support all the time.

To the Technical University of the North because its classrooms instilled values that I will apply in the personal and familiar scope.

To all the engineers who were my teachers in the different subjects because of them I learned all the things that today I will apply in the professional scope.

To Ing. Marco Pusdá for his guidance and support for the development and completion of this research.

To Ing. José Luis Rodríguez for his support at the beginning of this investigation

Finally to my friends with whom I shared and I will continue to share many experiences that I will never forget.

Bibliographic references

[1] Albornoz Luis A., G. L. (2012). La Televisión Digital Terrestre. Experiencias nacionales y diversidad en Europa, América y Asia. Buenos Aires: LA CRUJIA.

[2] Albornoz, L. A., & García Leiva, M. T. (2 de 6 de 2015). *Universidad Carlos III de Madrid*. Obtenido de http://e-archivo.uc3m.es/handle/10016/14621

[3] Alcalde, P. (2014). *Electrónica*. Ediciones Paraninfo S.A.

[4] Amazon Company. (15 de 12 de 2015). *amazon web services*. Obtenido de https://aws.amazon.com/es/ec2/

[5] Ayala, A. (25 de Agosto de 2014). *Televisión Digital Terrestre, Middleware Ginga*. Obtenido de null pointer exception apuntes, opinion y mas..: http://alejandroayala.solmedia.ec/?p=1774

[6] Baum, G., & Soares, L. (2012). *Ginga Middleware and Digital TV in Latin America*. doi:10.1109/MITP.2012.78

[7] Borranis Bureau Consultores. (6 de Septiembre de 2012). *#Ginga, para potenciar la #TV digital en #Sudamérica*. Obtenido de Borranis Bureau Consultores. Las últimas novedades del mercado, para emprendedores, profesionales y empresarios: http://news.borranisbureau.com/2012/09/06/ginga-para-potenciar-la-tv-digital-en-sudamerica/

[8] Cabezas, G., & Quezada, M. (Noviembre de 2012). Diseño e Implementación de un prototipo para un sistema de generación de aplicaciones interactivas con ginga-ncl para la evaluación de servicios masivos. Quito. Recuperado el 3 de Febrero de 2016, de http://bibdigital.epn.edu.ec/handle/15000/5167

[9] Comunidad Peruana de Ginga. (03 de 12 de 2015). Obtenido de Comunidad Ginga Perú: http://www.gingaperu.org/

[10] Coromina, Á. P.-U. (2005). *TV Digital e integración, ¿televisión para todos?* Librería-Editorial Dykinson, 2005.

[11] El Diario Manabita. (29 de Abril de 2015). Obtenido de El Diario Manabita de libre pensamiento: http://www.eldiario.ec/noticias-manabi-ecuador/264426-apagon-en-5-anos-mas/

[12] Espinoza, T., & Andrés, G. (2014). *Actualización de datos desde carrusel en aplicaciones GINGA Informe de Proyecto*. Recuperado el 12 de Febrero de 2016, de http://www.electronica.utfsm.cl/

[13] ginga.org. (18 de Enero de 2015). *Ginga*. Obtenido de Ginga: http://www.ginga.org.br/es/inicio

[14] Goette, E. (15 de 12 de 2015). *Emanuel Goette, alias Crespo*. Obtenido de http://emanuelpeg.blogspot.com/2012/05/eucalyptus-systems-y-nebula-nubes.html

[15] Google Sites. (17 de Enero de 2015). *Roles: Metodología XP*. Obtenido de Metodología XP: https://sites.google.com/site/xpmetodologia

[16] Google Sites. (s.f.). *Metodología XP*. Recuperado el 5 de 11 de 2015, de https://sites.google.com/site/xpmetodologia/marco-teorico/roles

[17] Instituto Nacional de Estadística y Censos [INEC]. (s.f.). *INEC*. Recuperado el 13 de Febrero de 2016, de http://www.ecuadorencifras.gob.ec/

[18] Jaramillo, A. (1 de Noviembre de 2014). *Tendencias: Diario El Comercio*. Obtenido de El Comercio.com: http://www.elcomercio.com.ec/tendencias/software-ginga-television-digital-ecuador.html

[19] Kazancigil, M. (2013). Xlet-based applications for seismic early warning and Emergency Services in the IDTV environment. doi:10.1109/CTS.2013.6567241

[20] Kezherashvili, B. (s.f.). *Computación en la Nube*. Almería.

[21] La Hora. (29 de Abril de 2015). Obtenido de La Hora: http://www.lahora.com.ec/index.php/noticias/show/1101516289#.VUGSjSGqqko

[22] La Televisión Digital: Fundamentos y teorías (Vol. I). (2009). Barcelona: Marcombo.

[23] Laboratotio de Investigación y Formación en Informática Avanzada. Facultad de Informática. UNLP. (2 de Enero de 2015). *Lifia*. Obtenido de Lifia: http://tvd.lifia.info.unlp.edu.ar/ginga.ar/

[24] López, M., & Oleas, K. (2012). Estudio metodológico para el diseño de interfaces entre el PC y el usuario utilizando ISDB.Tb y Middleware Ginga. Obtenido de http://dspace.espoch.edu.ec/handle/123456789/2912

[25] Menchaca, R., & García, F. (s.f.). *Revista Digital Universitaria*. Recuperado el 13 de Febrero de 2016, de http://www.revista.unam.mx/vol.1/num2/art4/

[26] Microsoft. (2016). *Microsoft Developer Network*. Obtenido de https://msdn.microsoft.com/es-es/library/bb669144.aspx

[27] Ministerio de Telecomunicaciones y Sociedad de la Información. (2014). *Ministerio de Telecomunicaciones y Sociedad de la Información*. Obtenido de Ministerio de Telecomunicaciones y Sociedad de la Información: http://www.telecomunicaciones.gob.ec/television-digital-terrestre-en-el-ecuador/

[28] Ministerio de Telecomunicaciones y Sociedad de la información. (2015). *Ministerio de Telecomunicaciones y Sociedad de la información.* Obtenido de http://www.telecomunicaciones.gob.ec/television-digital-terrestre-en-el-ecuador/

[29] Ministerio de Telecomunicaciones y Sociedad de la Información. (25 de 05 de 2015). *Ministerio de Telecomunicaciones y Sociedad de la Información*. Obtenido de http://www.telecomunicaciones.gob.ec/

[30] Oracle Corporation. (25 de 11 de 2015). *Java*. Obtenido de https://www.java.com/es/about/whatis_java.jsp

[31] Oracle Corporation. (s.f.). *Oracle*. Recuperado el 13 de Febrero de 2016, de http://www.oracle.com/technetwork/java/javaee/overview/index.html

[32] Ordax Cassá, J. M. (2012). *Programación web en java.* España: Ministerio de Educación de España. Obtenido de http://www.ebrary.com

[33] Ordoñez, J. L. (2011). *TDT: Televisión Digital Terrestre (Ciencia Divulgativa).* Barcelona: Creaciones Copyright.

[34] Paredes, A., Tonguino, N., Olmedo, G., & Acosta, F. (2012). *Performance analysis on return channel for interactive digital TV ISDB-Tb system.* doi:10.1109/LATINCOM.2012.6505993

[35] Quingaluisa, Á., Torres, J., Martínez, D., & Salvador, S. (2010). Estudio e Investigación del Middelware Ginga-J del estándar Brasileño de televisión digital . Caso Práctico : Desarrollo de una aplicación interactiva aplicando la metodología OpenUP / Basic como parte del Proyecto. Recuperado el 15 de Enero de 2016, de http://repositorio.espe.edu.ec/handle/21000/4748

[36] Riberi, F. (2012). *Un Prototipo de Desarrollo NCL para la Plataforma de Televisión Digital.* Recuperado el 5 de Febrero de 2016, de http://41jaiio.sadio.org.ar/sites/default/files/19_SSI_2012.pdf/

[37] Riberi, F. G. (2012). Un Prototipo de Desarrollo NCL para la Plataforma de Televisión Digital. Recuperado el 6 de Febrero de 2016, de http://41jaiio.sadio.org.ar/sites/default/files/19_SSI_2012.pdf

[38] Toledo Freitas, B., Susin, A., & Bonatto, A. (2014). *Ginga MiddleWare on a SoC for Digital Television Set-Top Box.* doi:10.1109/LASCAS.2014.6820290

[39] Torres, J. (28 de Junio de 2011). *Comunidad Ginga Ecuador*. Obtenido de http://comunidadgingaec.blogspot.com/2011/06/middleware-ginga.html

[40] Unión Internacional de Telecomunicaciones. (2013). *Medición de la Sociedad de la Información.* Recuperado el 13 de Febrero de 2016, de http://www.itu.int/en/ITU-D/Statistics/Pages/publications/mis2013.aspx

[41] UTFSM. (15 de Enero de 2015). *Ginga y TVD: Tarea Redes de Computadores II*. Obtenido de Ginga y TVD: http://www2.elo.utfsm.cl/~elo323/prep.html

[42] VV.AA. (2013). Estructura y Tecnología de Computadores I (Gestión y Sistemas). UNED.

[43] Wikipedia. (15 de 12 de 2015). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Eucalyptus_(inform%C3%A1tica)

About The Authors

Oscar Daniel Rodríguez Ortega. He graduated from the Computer Systems Engineering Degree at Universidad Técnica del Norte. He started his studies at the Technical University of North from the year 2008. He finished high school in 2007 obtaining the title in Mathematical Physicist.