

Sistema de Búsqueda, Almacenamiento y Procesamiento de Información para Generar Contenido Interactivo de Televisión Digital

Cevallos D. *; Cevallos F. *; Bernal I. *; Mejía D. *

*Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica
Quito, Ecuador (e-mail: davidcepn@yahoo.com;
epnfernando@gmail.com; {ivan.bernal; david.mejia}@epn.edu.ec)

Resumen: Este artículo presenta el desarrollo de un sistema de búsqueda, almacenamiento y procesamiento de información para generar contenido interactivo de televisión digital con Ginga-NCL. El sistema permite generar aplicaciones interactivas sin necesidad de que el usuario tenga conocimientos del lenguaje NCL haciendo uso de herramientas con una interfaz gráfica amigable. El sistema está conformado por un plug-in para el Entorno de Desarrollo Integrado (IDE) NCL Composer que genera automáticamente menús, y un Subsistema de Adquisición y Procesamiento de Datos (SAPDa) conformado por: a) un servicio web Windows Communication Foundation (WCF) que realiza búsquedas de información en sitios web y envía datos al receptor del televidente usando el canal de retorno, b) una base de datos por cada aplicación interactiva creada, c) una aplicación para consumir el servicio web permitiendo al usuario realizar las acciones de búsqueda, almacenamiento y procesamiento de información, y d) una aplicación para la generación de datos NCL (MIXER) encargada de combinar la información de la base de datos y los menús creados con el plug-in para obtener la aplicación interactiva final.

Palabras clave: televisión digital, Ginga-NCL, plug-in, servicios web, bot de búsqueda, canal de retorno.

Abstract: This article presents the development of a system for searching, storing and processing information to generate interactive content for digital television with Ginga-NCL. The system allows generating interactive applications without requiring the user to have knowledge of NCL language making use of tools with a friendly graphical interface. The system consists of a plug-in for the Integrated Development Environment (IDE) NCL Composer which automatically generates menus, and a Subsystem for Acquisition and Processing of Data (SAPDa) which consists of: a) a Windows Communication Foundation (WCF) web service that searches for information from different websites and send data to the viewer's receiver using the return channel, b) a database for each interactive application created, c) an application to consume the web service allowing the user to perform actions of searching, storing and processing of information, and d) an application for NCL data generation (MIXER) responsible for combining the information from the database and the menus created with the plug-in to get the final interactive application.

Keywords: digital television, Ginga-NCL, plug-in, web services, search bot, return channel.

1. INTRODUCCIÓN

Las aplicaciones interactivas son sin lugar a duda una de las principales ventajas de la televisión digital. El estándar ISDB-Tb (*International System for Digital Broadcast, Terrestrial, Brazilian version*) dispone del *middleware* denominado Ginga, el cual facilita la comunicación del flujo de datos de las aplicaciones con el hardware de los *Set Top Boxes* (STB), haciendo a las aplicaciones independientes del receptor. Para crear aplicaciones interactivas existen al momento pocas herramientas, algunas aún en desarrollo como el IDE NCL Composer. Sin embargo, el uso de estas herramientas demanda conocimientos de programación, por

lo que en este artículo se presenta un sistema de búsqueda, almacenamiento y procesamiento de información para generar contenido interactivo de televisión digital. Este sistema está formado por: un *plug-in* desarrollado para NCL Composer, que permite generar contenido interactivo para televisión digital de forma sencilla y sin tener conocimientos de programación, y un subsistema de adquisición y procesamiento de datos, el cual permite adquirir y procesar información que podrá ser radiada o enviada a la aplicación interactiva a través del canal de retorno.

El *plug-in* para el IDE NCL Composer ha sido desarrollado empleando la biblioteca Qt y el lenguaje C++, lo cual permite que el *plug-in* pueda emplearse en el IDE NCL Composer al igual que el resto de *plug-ins* oficiales. El servicio web WCF y las aplicaciones cliente y MIXER han sido desarrolladas

empleando la plataforma Visual Studio 2012, la cual facilita la creación de aplicaciones con interfaz gráfica de usuario para Windows. Para el almacenamiento de la información se ha empleado SQL Server 2008 por la facilidad que brinda para la creación de bases de datos relacionales y por la compatibilidad de conexión cliente-servidor que permite con la plataforma Visual Studio.

El artículo se ha organizado de la siguiente manera: se presenta un breve resumen sobre Ginga, se describe el sistema de búsqueda, almacenamiento y procesamiento de información, a continuación se presenta el *plugin* de generación automática de menús para NCL Composer, también se describe el servicio web basado en WCF, la aplicación cliente que permite consumir el servicio web y la aplicación para generación de datos NCL (MIXER), luego se presenta una aplicación interactiva para televisión digital enfocada en la temática de la Educación Superior del Ecuador, posteriormente se presentan las pruebas de funcionamiento y finalmente se indican las conclusiones.

2. GINGA

El *middleware* Ginga está compuesto por dos subsistemas entrelazados que permiten la creación de aplicaciones interactivas empleando un paradigma de programación declarativo o imperativo. Ginga emplea el lenguaje declarativo NCL [6], basado en XML (*eXtensible Markup Language*) que trabaja con etiquetas definidas; y emplea Java y Lua como lenguajes imperativos [6].

El emplear uno de estos paradigmas demanda conocimiento del desarrollador en estos lenguajes de programación, por lo que, hacer uso de herramientas con una interfaz gráfica amigable que generen automáticamente el código facilitará notablemente el desarrollo de aplicaciones interactivas. En

este caso, se ha desarrollado un sistema para generar aplicaciones interactivas con el ambiente declarativo Ginga-NCL y su extensión, a través de Lua, para la creación de aplicaciones híbridas.

3. SISTEMA DE BÚSQUEDA, ALMACENAMIENTO Y PROCESAMIENTO DE INFORMACIÓN

El sistema de búsqueda, almacenamiento y procesamiento de información para generar contenido interactivo de televisión digital se presenta en la Fig 1 y está conformado por:

- *Plug-in* para el IDE NCL Composer.
- Subsistema de Adquisición y Procesamiento Datos (SAPDa).

El *plug-in* para el IDE NCL Composer extiende la funcionalidad de dicho IDE para que sea capaz de crear automáticamente menús, generando de manera transparente para el usuario todo el código NCL correspondiente.

El SAPDa está conformado por:

- Un servicio web *Windows Communication Foundation* (WCF) que implementa un *bot* de búsqueda para consultar información de distintos sitios web y envía datos al STB del televidente a través del canal de retorno.
- Una aplicación de escritorio para consumir el servicio web permitiendo al usuario realizar las acciones de búsqueda, almacenamiento y procesamiento de información.
- Una aplicación de escritorio (MIXER), encargada de realizar la generación de datos NCL y la mezcla (*mix*) del contenido de la base de datos y los menús generados en base al *plug-in* para la generación del contenido interactivo a mostrarse al televidente.
- Una base de datos por cada aplicación interactiva creada, la cual será generada automáticamente por el servicio web.

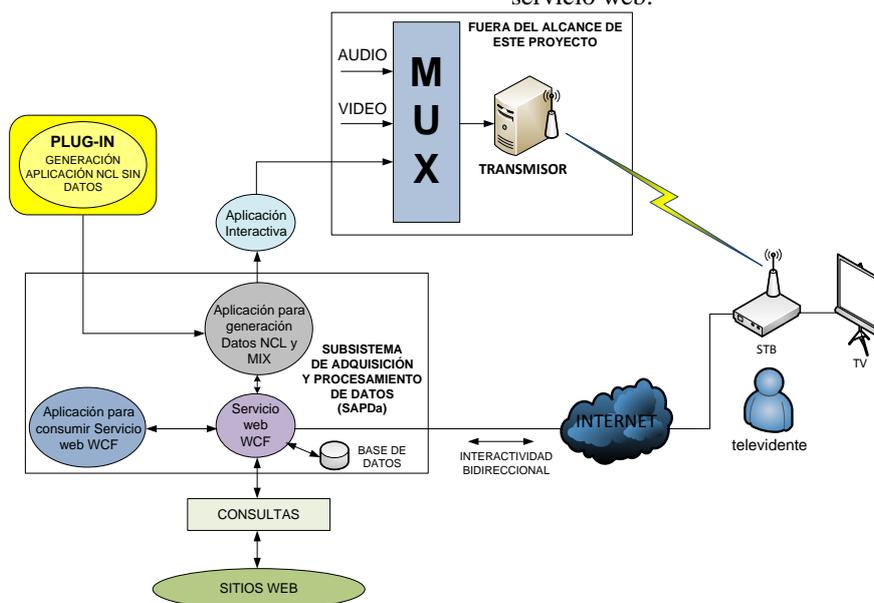


Figura 1. Diagrama del sistema

La aplicación interactiva final es el resultado de emplear el *plug-in* y el SAPDa.

El sistema brinda varias ventajas tales como:

- Posibilita la creación de aplicaciones interactivas en menor tiempo y sin necesidad de conocimientos avanzados del lenguaje NCL.
- Permite crear aplicaciones interactivas con menús de un alto nivel de interactividad. El *plug-in* genera automáticamente los menús con elementos media y varias características y funcionalidades que dan lugar a aplicaciones llamativas para el televidente, a diferencia de un menú típico creado con Ginga-NCL en el que el televidente, usualmente, hace uso únicamente de los botones ROJO, VERDE, AMARILLO y AZUL del control remoto para seleccionar las opciones que el menú ofrece.
- Permite obtener información de sitios web para que dicha información pueda ser almacenada en bases de datos y, posteriormente, radiada conjuntamente con la aplicación interactiva u obtenida a través del canal de retorno. Para esta última funcionalidad, el sistema genera *scripts* Lua de forma automática.
- Realiza de forma totalmente transparente para el usuario las acciones requeridas para el uso de la información en las bases de datos.

4. PLUG-IN DE GENERACIÓN AUTOMÁTICA DE MENÚS PARA EL IDE NCL COMPOSER

El *plug-in* lleva por nombre **Menu Creator** y permite al usuario crear uno o varios menús de manera sencilla generando automáticamente el código NCL respectivo.

Por cada uno de los menús creados por el usuario, el *plug-in* generará un nodo de contexto NCL [5]. Dicho nodo contiene todas las facilidades de interacción que un menú puede prestar al televidente tales como uso de estilos gráficos, menús con distintas propiedades de diseño (número de filas, número de columnas, dimensiones del menú, entre otras), división de menús en vistas (una vista es un fragmento de un menú), uso de viñetas de navegación (símbolos gráficos que indican al televidente cómo desplazarse entre las distintas vistas), creación de menús con filas y columnas de distinto tamaño, ingreso del texto a presentar, navegación entre los elementos de menús e inserción de enlaces NCL [5].

4.1 Entorno de Desarrollo Integrado NCL Composer

NCL Composer es una herramienta multiplataforma y flexible para crear aplicaciones en NCL para televisión digital interactiva (iDTV) [14]. El IDE NCL Composer está compuesto de tres partes:

- **Composer-core:** Implementa las reglas de negocio para la generación del documento NCL.
- **Composer-gui:** Implementa la interfaz gráfica de usuario en la que se cargan los *plug-ins*.
- **Composer-plugins:** Comprende el conjunto de *plug-ins* que conforman NCL Composer con su interfaz gráfica y hace uso de la lógica implementada en el composer-core.

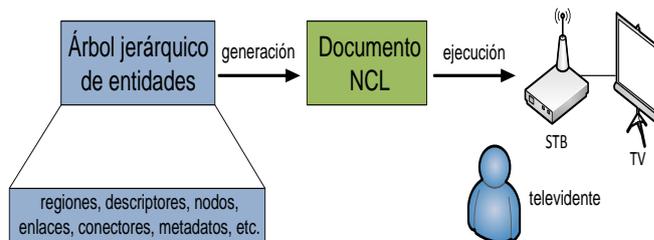


Figura 2. Lógica de trabajo de NCL Composer

El código fuente de NCL Composer desarrollado por el laboratorio TeleMídia está disponible en [11] y se lo puede versionar mediante el software de control de versiones git [4]. NCL Composer está basado en una estructura de árbol jerárquico de entidades. Una entidad es una clase que modela de manera lógica una determinada etiqueta del documento NCL pudiendo contener entidades hijas y al mismo tiempo ser hija de otra entidad (padre). La Fig 2 muestra la lógica de trabajo del modelo interno de NCL Composer. A partir del árbol jerárquico de entidades se genera el documento NCL.

Cada *plug-in* en NCL Composer puede gestionar el árbol jerárquico de entidades a través del composer-core de tal modo que permita al usuario generar el documento NCL evitándole la tarea de programar directamente el código y brindándole una interfaz gráfica amigable.

La Tabla 1 expone los *plug-ins* oficiales existentes a la fecha de la redacción de este artículo con los que cuenta NCL Composer, así como su respectiva funcionalidad.

4.2 Mecanismo Signals and Slots

Para que un *plug-in* pueda gestionar el árbol jerárquico de entidades debe invocar la lógica implementada en el composer-core mediante el mecanismo *signals and slots*. Este mecanismo permite al programador de la aplicación comunicar objetos sin la necesidad de que dichos objetos sepan el uno del otro [1].

La base del mecanismo consiste en que el objeto que desea comunicarse (en este caso una instancia del *plug-in*) debe emitir (*emit*) una señal (*signal*) y el objeto que recibe dicha señal (en este caso una instancia del composer-core) debe conectar (*connect*) dicha señal al *slot* correspondiente.

Tabla 1. *Plug-ins* oficiales del IDE NCL Composer [13]

<i>Plug-in</i>	Funcionalidad
NCL Textual View	Permite la interacción con el contenido textual NCL.
NCL Structural View	Permite la interacción visual con la estructura lógica del documento. En NCL, la estructura lógica es representada por objetos media, contextos y entidades de enlace.
NCL Layout View	Permite al usuario interactuar visualmente con la región donde un objeto media será presentado.
NCL Properties View	Permite al usuario el fácil acceso (y cambio) del contenido de las propiedades de un elemento seleccionado.
NCL Outline View	Presenta la estructura del documento como una vista de árbol. Esta vista permite navegar entre los elementos.
NCL Validator Plugin	Valida documentos NCL cada vez que se modificó el modelo de núcleo, dando información al usuario sobre los errores en el código fuente del documento que se está editando.

En este esquema, un *slot* se refiere a un método que realiza una funcionalidad y que será invocado cuando el objeto recibe la señal, de tal manera que los *plug-ins* gestionan el árbol jerárquico de entidades sin manipular directamente dichas entidades, sino que lo hacen de forma indirecta a través de la emisión de señales dirigidas al *composer-core*, quien realiza una funcionalidad (*slot*) determinada como se muestra en la Fig. 3.

Las señales más importantes a las que responde el *composer-core* se muestran en la Tabla 2.

4.3 Creación de un nuevo *plug-in* en NCL Composer

Un *plug-in* en NCL Composer está basado en dos interfaces de C++ cuyos métodos deben ser implementados en dos clases distintas [12]:

IPluginFactory: Contiene información sobre el *plug-in* que se presentará en el menú *Help About Plugins/Details* del *composer-gui*, como el ID (identificador) del *plug-in*, una cadena de texto que lo identifica unívocamente (ej. *epn.menuecreator*), el nombre registrado del *plug-in* (ej. *Menu Creator*), el propietario (ej. *EPN*), la licencia (ej. *LGPU*), una descripción (ej. *Plug-in* para creación de menús), un URL (*Uniform Resource Locator*) donde se puede obtener más información (ej. <http://ginga.epn.edu.ec>), entre otros aspectos. NCL Composer hace uso de esta interfaz para crear nuevas instancias del *plug-in* [12].

IPlugin: Contiene la funcionalidad del *plug-in*. Cuando se abre un nuevo proyecto por NCL Composer, se requiere una nueva instancia del *plug-in*. Esta interfaz se enlaza con una instancia del proyecto y será llamada cada vez que el proyecto sufra un cambio [12]. De igual manera, *IPlugin* es capaz de requerir cambios en el proyecto. La Tabla 3 describe los métodos más importantes de la interfaz *IPlugin*.

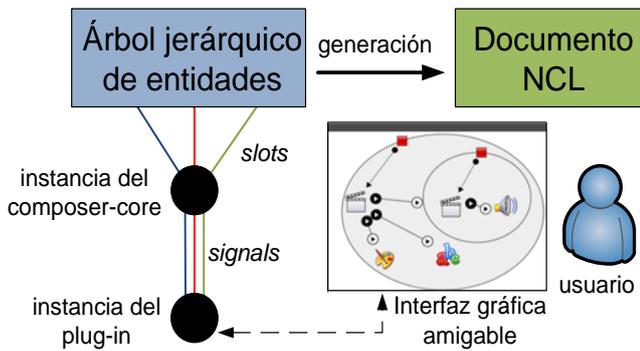


Figura 3. Comunicación entre un *plug-in* y *composer-core* a través del mecanismo *signals and slots*

Tabla 2. Algunas señales a las que responde *composer-core*

Signal	Slot
addEntity	Añade una nueva entidad NCL.
setAttributes	Establece los atributos de la entidad NCL (atributos de la etiqueta XML).
removeEntity	Elimina una entidad NCL.
setPluginData	Guarda los datos del <i>plug-in</i> en el archivo de proyecto de <i>Composer (.cpr)</i> .
getPluginData	Lee los datos almacenados del <i>plug-in</i> del archivo de proyecto <i>Composer (.cpr)</i> .
setCurrentProjectASDirty	Establece que se han realizado cambios en el proyecto y de ser necesario habilita el botón <i>Guardar</i> en el <i>composer-gui</i> .

Tabla 3. Algunos métodos de la interfaz *IPlugin*

Método	Funcionalidad
void init()	Se ejecuta tras instanciar el <i>plug-in</i> y permite cargar los datos guardados en el archivo <i>.cpr</i> .
QWidget* getwidget()	Obtiene el objeto que corresponde a la interfaz gráfica de usuario del <i>plug-in</i> a instanciar.
bool saveSubsesion()	Método que se ejecuta cuando se guarda el proyecto o sus cambios.

Al crear un nuevo proyecto de NCL *Composer* se crea una instancia de cada uno de los *plug-ins* y se llama al método *init()* para cargar los datos en caso de que sea un proyecto ya existente y al método *getwidget()* para obtener su interfaz gráfica. Para guardar los cambios de un proyecto, NCL *Composer* llama al método *saveSubsesion()* de cada uno de los *plug-ins* y guarda en el archivo *.cpr* los datos que cada *plug-in* requiera guardando diferenciándolos por el ID especificado en la interfaz *IPluginFactory*.

4.4 *Plug-in Menu Creator*

Menu Creator se ha diseñado en base a una estructura jerárquica de clases, como se presenta en la Fig 4.

El primer nivel contiene la clase *Estructura*. Una instancia de esta clase tiene referencias a una o más instancias de la clase *Menú* del segundo nivel. Cada una de estas instancias tiene como atributos las propiedades generales de un menú que ha diseñado el usuario tales como el número de filas, número de columnas, porcentaje de ocupación del menú en la pantalla (ancho y alto), número de filas y de columnas que se muestran por cada vista y la posición que ocupa el menú en la pantalla. Cada instancia de tipo *Menú* tiene referencias a una instancia de la clase *Texto*, correspondiente al texto del menú, y una o más instancias de la clase *Vista*; todos estos objetos se ubican en el nivel 3. En el nivel 4 se encuentran instancias de la clase *Botón*, que representan cada uno de los elementos que conforman una vista determinada de un menú. Cada una de estas instancias contiene tres instancias del nivel 5 de tipo *Cuadro* correspondientes al texto del elemento y dos imágenes de fondo, una para cuando el elemento no está seleccionado y la otra para cuando el televidente selecciona el elemento.

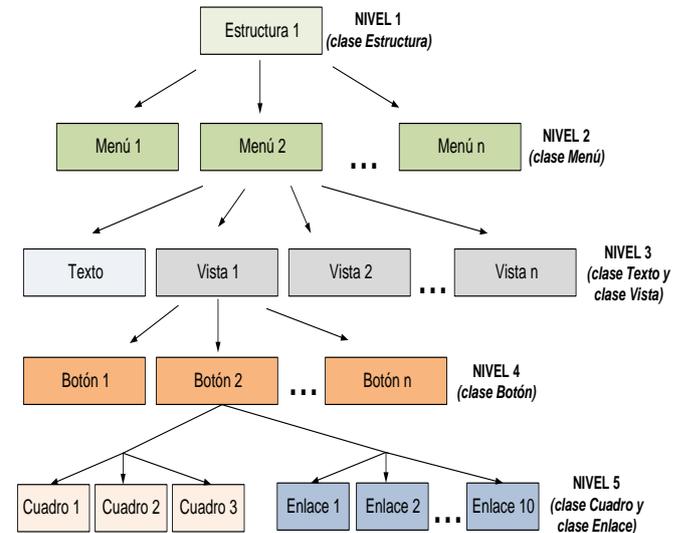


Figura 4. Estructura jerárquica de clases de *Menu Creator*

La estructura jerárquica de clases permite que todas las entidades NCL requeridas para los menús diseñados por el usuario sean añadidas al árbol jerárquico de entidades de NCL Composer, de manera que, posteriormente, el usuario pueda generar con el *plug-in* el código NCL de cada menú, como se muestra en la Fig 5. Para ello, la instancia del *plug-in* (creada gracias a la clase que implementa la interfaz IPluginFactory) emite las señales correspondientes a una instancia del composer-core mediante el mecanismo de *signals and slots* para generar el documento NCL.

5. SERVICIO WEB WINDOWS COMMUNICATION FOUNDATION (WCF)

WCF es una plataforma que permite la creación de servicios web empleando tecnologías de comunicación como SOAP (*Simple Object Access Protocol*) y REST (*Representational State Transfer*) [2].

SOAP es un protocolo que emplea XML como formato de datos para realizar invocación remota de métodos típicamente mediante HTTP (*HyperText Transfer Protocol*). Cada petición y respuesta se envuelve en un mensaje SOAP en formato XML. La principal ventaja de SOAP es que facilita la implementación de la aplicación cliente que hace uso del servicio web a través de un proxy que permite ejecutar métodos remotos como si se ejecutasen localmente [10].

REST es una tecnología utilizada para implementar servicios web. Los servicios creados en base a esta arquitectura son comúnmente denominados servicios RESTful.

A diferencia de los servicios web implementados con SOAP, los servicios RESTful no requieren de una envoltura especial para la comunicación y además no están limitados a retornar datos únicamente en formato XML sino que pueden hacer uso de otros formatos como JSON (*JavaScript Object Notation*), HTML (*HyperText Markup Language*), texto plano e inclusive archivos multimedia [2].

Estas características hacen que los servicios RESTful sean ideales para la comunicación con el STB del televidente a través del canal de retorno. Sin embargo, la implementación de la aplicación cliente puede llegar a ser más complicada pues requiere manejar espacios de nombres URL para la invocación de métodos [3].

El servicio web está implementado empleando el lenguaje de programación C# usando SOAP y REST. Será consumido por la aplicación cliente y por el MIXER a través de SOAP, y mediante la arquitectura REST, es capaz de enviar datos en texto plano a través del canal de retorno para ser consumido por el STB del televidente. El servicio web WCF realiza principalmente cuatro tareas:

- Busca información que el usuario requiere a través de consultas a distintos sitios web.
- Almacena los datos encontrados mediante consultas a sitios web o aquellos que el usuario ingrese manualmente, en una base de datos creada automáticamente a través del mismo servicio.
- Realiza consultas en la base de datos respectiva para proveer los datos a la aplicación MIXER.
- Responde a la petición del STB del televidente para proveer datos específicos a través del canal de retorno.

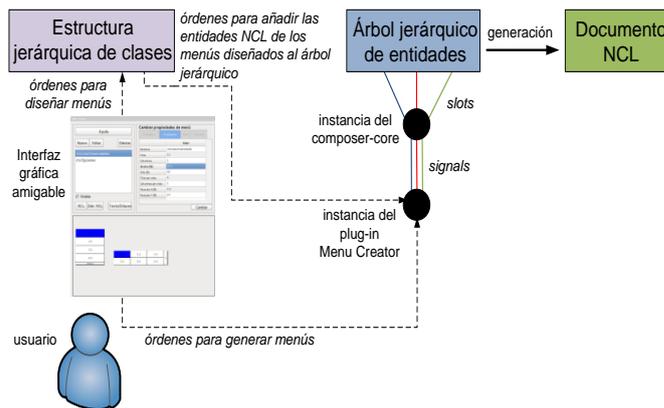


Figura 5. Lógica de trabajo de Menu Creator

5.1 Búsqueda de Información

Para la búsqueda de información el servicio web implementa un *bot* de búsqueda que permite al usuario ingresar una lista de sitios web de confianza (URL) y un grupo de palabras clave. Por cada URL ingresado, el *bot* realiza una búsqueda en Google empleando todas las palabras clave especificadas para obtener, a través de los resultados de la búsqueda, los URL correspondientes a distintas páginas web contenidas en el sitio web en cuestión. De esta manera, el proceso es mucho más rápido que explorar recursivamente cada sitio web en busca del listado de páginas web en las que se encuentre información útil para el usuario.

Por cada URL obtenido el *bot* de búsqueda realiza un análisis de la información de la página web mediante la librería *HTML Agility Pack* [7], obteniendo el respectivo documento DOM (*Document Object Model*) [8] y, a través de la navegación de cada nodo HTML, los párrafos que mejor se ajusten a la búsqueda del usuario en base a un algoritmo de puntuaciones.

5.2 Base de Datos

El servicio web crea por cada tema una base de datos SQL y por cada subtema una tabla para dicha base de datos. En este esquema, un tema corresponde a la temática de una aplicación interactiva que el usuario está creando.

Cada tabla que se crea tiene elementos de tipo cadena con una longitud máxima de 3000 caracteres. El usuario puede ingresar los datos que desea almacenar mediante el sistema de búsqueda explicado anteriormente o puede hacerlo de forma manual.

5.3 Consultas a la base de datos para proveer datos a la aplicación MIXER

La aplicación MIXER accede a los datos a través del servicio web, y puede trabajar con varias aplicaciones interactivas, para cada una de las cuales se debe cargar el documento NCL creado en base al IDE NCL Composer y su base de datos respectiva (identificada con el tema de la aplicación interactiva).

5.4 Respuesta al televidente a través del canal de retorno

La aplicación MIXER es capaz de crear *scripts* Lua dentro del carpeta del proyecto. Se puede utilizar Lua para convertir al STB del televidente en un cliente web capaz de solicitar determinados datos a un servidor web remoto usando el canal de retorno.

La Fig 6 muestra el mecanismo de comunicación entre un servidor y el STB. En este esquema, el canal de retorno es un canal bidireccional en el que el cliente web realiza una petición HTTP del contenido que desea y el servidor web responde con dicho contenido. Como Lua es un lenguaje imperativo, el contenido puede adaptarse antes de ser presentado al televidente, por ejemplo, si el texto sobrepasa las dimensiones del área disponible será dividido en varias líneas.

La librería *socket.http* de Lua permite la comunicación a través del protocolo HTTP con un servidor web remoto [9]. La librería crea un *socket TCP (Transmission Control Protocol)* de manera que la aplicación Lua pueda enviar peticiones HTTP (puerto destino 80) y recibir la información correspondiente desde el servidor web.

6. APLICACIÓN CLIENTE PARA CONSUMIR SERVICIO WEB WCF

La aplicación cliente desarrollada para consumir el servicio web es una aplicación de escritorio implementada empleando el lenguaje C# y lleva por nombre **Textual Data Creator**. Se usa esta aplicación para hacer uso de las funciones de búsqueda, almacenamiento y procesamiento de información ofrecidas por el servicio web. La aplicación cuenta con tres módulos: un módulo de ingreso del contenido a visualizarse en el televisor, un módulo de gestión de usuarios y un módulo de configuración del servicio web.

7. APLICACIÓN MIXER

La aplicación MIXER es una aplicación de escritorio desarrollada empleando C# y lleva por nombre **NCL-Textual Data Mixer**. La aplicación permite generar el contenido multimedia que será presentado en los elementos del menú construido en base al *plug-in*. Para ello hace uso del documento NCL previamente generado por el diseñador y la base de datos con el contenido (texto) que se presentará en el televisor.

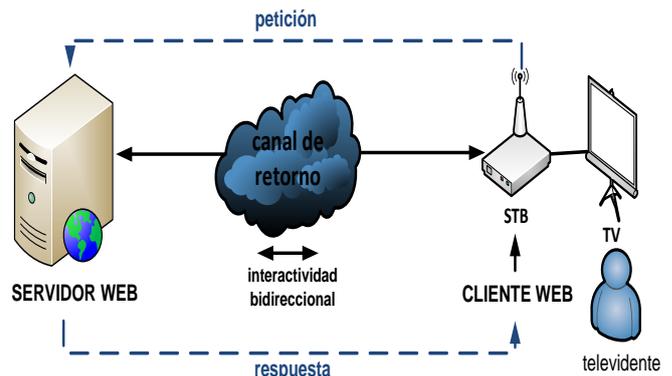


Figura 6. Comunicación entre servidor y STB

Su funcionamiento trabaja en base a tres tipos de metadatos. Estos metadatos deben ser escritos en el documento NCL mediante el IDE NCL Composer. La Fig 7 muestra la estructura de los tres tipos de metadatos. Todos poseen un identificador de un elemento NCL cuya fuente (*src*) es una imagen de extensión .png. El primer tipo de metadato identifica un ID de un nodo de contexto [5] que hace referencia a un menú creado mientras que el segundo y tercer tipo identifican un objeto media [5].

El primer tipo de metadato tiene como fin completar con datos un menú e identifica una tabla de la base de datos que contiene el texto a exhibirse en cada uno de los elementos del menú especificado. El texto que se muestra en cada elemento del menú podría contener varias líneas y en caso de ser muy extenso la aplicación adecúa el contenido al espacio disponible omitiendo de ser necesario alguna parte del texto y completándola con puntos sucesivos.

El segundo tipo de metadato se utiliza para obtener el dato a presentarse cuando el televidente selecciona un elemento específico del menú, e identifica dicho dato a través del número de fila y columna de la tabla en donde se encuentra almacenado. Si el texto del dato supera el espacio donde se presentará, entonces se redimensionará automáticamente su tamaño hasta que pueda caber en el área disponible.

El tercer tipo de metadato cumple con una función similar al del segundo tipo pero el televidente obtiene el dato a través del canal de retorno, lo que permite reducir el tamaño de la aplicación interactiva a ser radiada. La letra *r* al final de la estructura de este metadato lo diferencia del segundo. La aplicación generará automáticamente un *script* Lua, que al ser ejecutado por el STB traerá el dato mediante una petición HTTP al servicio web a través del canal de retorno.

8. APLICACIÓN INTERACTIVA SOBRE ASPECTOS DE EDUCACIÓN SUPERIOR DEL ECUADOR CON GINGA-NCL

El sistema de búsqueda, almacenamiento y procesamiento de información es capaz de generar con las herramientas presentadas anteriormente, contenido interactivo para televisión digital sobre cualquier temática. Para demostrar la funcionalidad del sistema, se ha realizado una aplicación interactiva para informar a la población acerca de las once universidades de categoría A del Ecuador.

La aplicación está conformada principalmente por dos menús. El primero consiste en un menú de once filas y una columna, que se presenta mediante vistas conformadas por cuatro elementos, con viñetas de navegación que permiten que el televidente pueda moverse entre las distintas vistas y seleccionar un elemento. En cada elemento se presenta una universidad y se puede navegar entre éstas a través de las teclas UP y DOWN del control remoto. El foco se mueve de acuerdo a las indicaciones del televidente para señalar la universidad que puede seleccionar mostrando su respectivo escudo. Para seleccionar una universidad se debe presionar la tecla OK del control remoto. Una vez que se ha elegido una universidad se activa el segundo menú. Este segundo menú está conformado por dos filas y seis columnas y muestra vistas de seis elementos.

```

METADATO 1: <meta name=id_contexto content=nombre_tabla>
METADATO 2: <meta name=id_media content=nombre_tabla/#fila/#columna>
METADATO 3: <meta name=id_media content=nombre_tabla/#fila/#columna/r>
    
```

Figura 7. Tipos de metadatos

Para navegar entre estas vistas el televidente puede hacer uso de las viñetas de navegación. Además, este menú ofrece al televidente los distintos aspectos de cada universidad, tales como: Fundación, Misión, Visión, Ubicación, Autoridades, Lema, Sitio web, Contacto, Pregrado y Postgrado. En cada vista se ha dejado un elemento del menú para brindar la opción de regresar al primer menú, permitiendo que el televidente pueda seleccionar otra universidad de su interés. La información que se presentará en el televisor al seleccionar los distintos elementos puede ser obtenida de la misma aplicación o a través del canal de retorno. Esta última opción es mucho más recomendable cuando se tiene información dinámica (en vivo) o cuando los datos son demasiado grandes, lo que involucraría radiar una aplicación de mayor tamaño. En este caso, los datos que abarcan gran cantidad de contenido como la misión, visión y oferta de pregrado y postgrado de las universidades serán obtenidos usando el canal de retorno mediante peticiones HTTP al servicio web. El resto de datos está almacenado dentro de la misma aplicación.

8.1 Creación de una base de datos e ingreso de información a través del servicio web y bot de búsqueda

Empleando la aplicación Textual Data Creator se puede hacer uso del servicio web para crear una base de datos. El servicio web por cada tema especificado crea una base de datos y por cada subtema una tabla.

Para la aplicación interactiva se creó una base de datos denominada **EducacionSuperior** con trece tablas. La primera tabla almacena los acrónimos de las once universidades, información que será utilizada para el menú 1. La segunda tabla permite almacenar los aspectos de las universidades y se la usará en el menú 2. Finalmente, por cada una de las once universidades se creará una tabla que almacene sus datos. La Fig 8 muestra una captura de pantalla de la aplicación.

Para ingresar los datos de las tablas se presentan dos alternativas, la primera es ingresar el texto directamente en cada elemento de la tabla y la segunda hacer uso del *bot* de búsqueda para consultar dicho elemento en los sitios web de confianza. Se ingresaron los datos de las primeras dos tablas de forma manual pues contienen los identificadores de las universidades y los aspectos que se ofrecen al televidente. La Fig 9 muestra la primera tabla con la información ingresada por el usuario de forma manual.

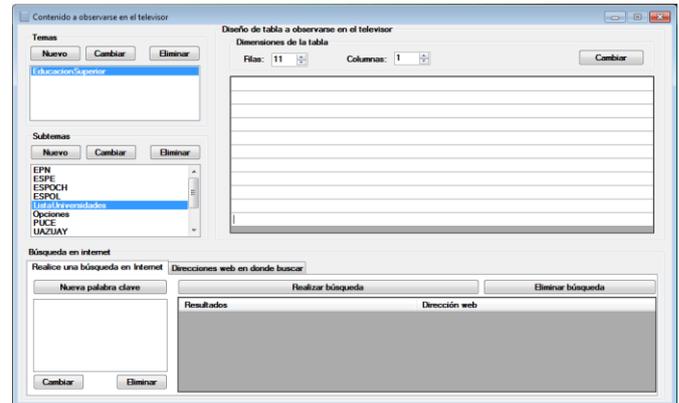


Figura 8. Base de datos y tablas creadas con Textual Data Creator

Para ingresar la información de las universidades es aconsejable hacer uso del *bot* de búsqueda. Para ello, se debe especificar uno o varios sitios web de confianza donde realizar la búsqueda y una o más palabras clave. Por ejemplo, si se desea buscar la misión de la EPN se podría especificar el URL <http://www.epn.edu.ec> como sitio web de confianza y como palabra clave misión. El *bot* de búsqueda dispone de un algoritmo de puntuaciones que ofrece los mejores párrafos encontrados. Además, la aplicación presenta el URL de las páginas web encontradas, para que el usuario pueda navegar y mirar en detalle la información. La Fig 10 muestra el resultado de la búsqueda de la misión de la EPN mediante el *bot*.

8.2 Creación de la aplicación interactiva mediante el IDE NCL Composer usando el plug-in Menu Creator

Para crear la aplicación interactiva que se ejecutará en el STB del televidente se hizo uso del IDE NCL Composer, el cual permite crear un proyecto que contendrá todos los archivos necesarios de la aplicación, y mediante el *plug-in* Menu Creator se crearon los dos menús mencionados anteriormente. Aunque el *plug-in* permite el ingreso manual de los textos que se presentarán en los menús, para este caso se crearon los menús vacíos para colocar el contenido posteriormente mediante la aplicación MIXER.

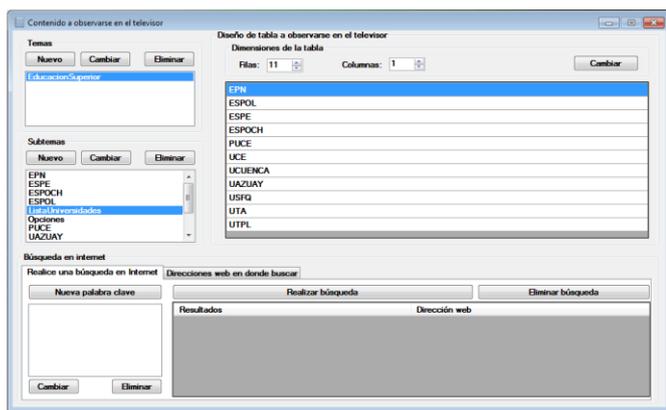


Figura 9. Ingreso de datos de forma manual en la primera tabla

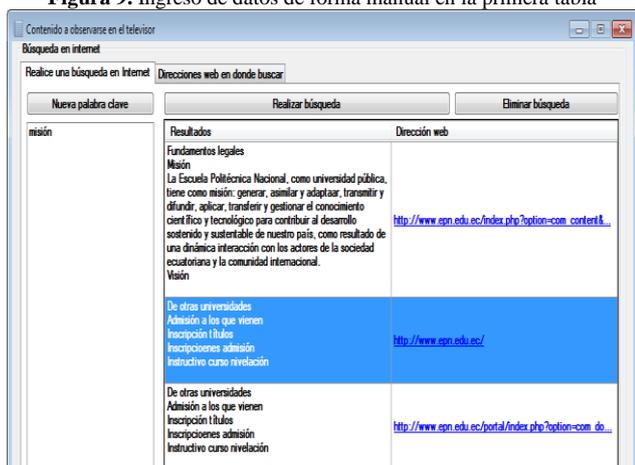


Figura 10. Recolección de información de sitios web a través del bot de búsqueda

Para el primer menú se especificaron el número de filas y de columnas, su ancho y alto, el número de filas y columnas de la vista y los porcentajes de ocupación en la pantalla. De igual manera se realizó el segundo menú. Cada uno de los menús diseñados se pueden mover para ser ubicados en el área de la pantalla. Además, dentro del proyecto creado se debe especificar una carpeta de plantillas en la que se almacenan las imágenes en formato PNG que serán empleadas por los elementos del menú, el foco, las viñetas de navegación y el texto en caso de ser ingresado. La Fig 11 presenta ambos menús diseñados en Menu Creator.

Luego de establecer las propiedades de los menús se procede a generar el código NCL de cada uno de ellos. Al generar el código NCL de un menú se obtendrá un nodo de contexto [5].

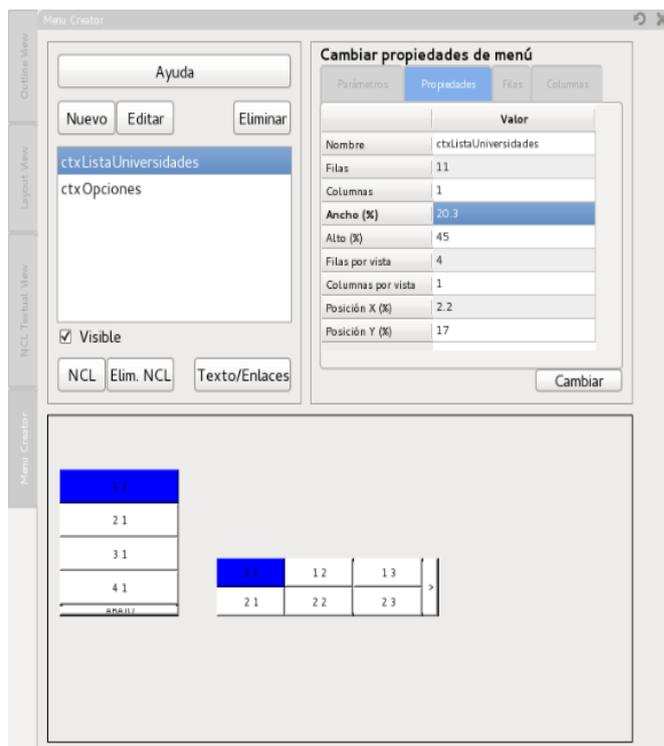


Figura 11. Menús diseñados en Menu Creator

La Fig 12 muestra la vista estructural de la aplicación con los nodos de contexto correspondientes al primer y segundo menú. Cada nodo puede ser manipulado como un todo brindando facilidad en el diseño de la aplicación.

A continuación se debe determinar los enlaces entre los nodos participantes y cómo deben ir presentándose al televidente. En esta aplicación, por ejemplo, en primer lugar se presenta un ícono de interactividad en la parte superior derecha del televisor, para indicarle al televidente que la programación dispone de una aplicación interactiva, y cuando el televidente pulse la tecla OK del control remoto se mostrará una imagen de fondo, el video de la programación redimensionado en un costado y el primer menú en el lado izquierdo para que el usuario mire el listado de universidades de categoría A. Cuando el televidente seleccione una universidad aparecerá entonces el segundo menú presentando la información de la universidad seleccionada.

Finalmente, para que la aplicación MIXER pueda asociar los elementos de los menús con la información de la base de datos es necesario ingresar metadatos en el documento NCL. Esto puede hacerse mediante el *plug-in* NCL Outline View. Para esto, se crearon dos metadatos del primer tipo para asociar el listado de universidades y los aspectos que se pueden ver de cada universidad con el primer y segundo menú respectivamente.

Por cada universidad será necesario crear diez metadatos. Seis de ellos corresponden a metadatos del segundo tipo para asociar los datos de Fundación, Ubicación, Autoridades, Lema, Sitio web y Contacto con imágenes que permitan presentar dichos datos, y cuatro metadatos del tercer tipo para asociar los datos de Misión, Visión, Pregrado y Postgrado con *scripts* Lua que permitan traer el contenido a través del canal de retorno.

8.3 Ingreso automático de datos en los menús y generación de scripts Lua a través de la aplicación MIXER

Con el fin de ingresar el contenido en los menús generados con el *plug-in* Menu Creator de forma automática, se puede hacer uso de la aplicación MIXER.

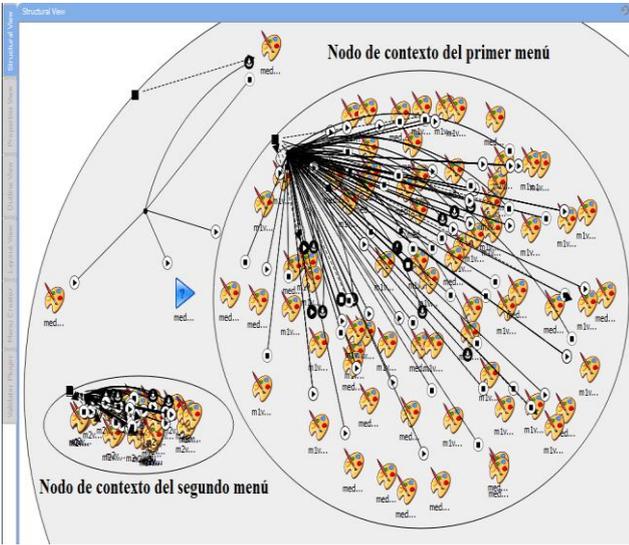


Figura 12. Nodos de contexto generados con el *plug-in* Menu Creator

Esta aplicación asociará cada elemento de los menús con el contenido de la base de datos en función de los metadatos establecidos en el documento NCL. Si los datos deben ser transmitidos por el canal de retorno, se generará automáticamente un *script* Lua dentro de la carpeta del proyecto para esta funcionalidad.

La aplicación MIXER es una aplicación de múltiples instancias, es decir, es capaz de tratar la asociación de datos con documentos NCL correspondientes a varias aplicaciones interactivas. La Fig 13 muestra la aplicación MIXER en la que se ha especificado la ubicación del documento NCL y el nombre de la base de datos anteriormente creados. La opción Parámetros de la aplicación MIXER realiza el trabajo de asociación. En caso de que suceda algún error debido a una discordancia entre las dimensiones del menú y la tabla de la base de datos que se asocia, la aplicación enviará un mensaje de advertencia y asociará los elementos de manera normal dejando casillas vacías del menú en caso de que las dimensiones de la tabla sean menores a las del menú, o ignorando datos en caso de que las dimensiones del menú sean menores a las de la tabla de la base de datos. Si no hay ningún inconveniente en el proceso de asociación se indicará el éxito del mismo. Además, la aplicación asocia datos específicos para los metadatos de tipo dos y genera *scripts* Lua por cada metadato de tipo tres. La Fig 14 muestra a la aplicación MIXER asociando datos con los menús creados.

De esta manera se logra completar el documento NCL y con ello se obtiene la aplicación interactiva final que podrá ser ejecutada en el STB del televidente.

9. PRUEBAS DE FUNCIONAMIENTO

Las pruebas de funcionamiento de la aplicación interactiva desarrollada con el sistema de búsqueda, almacenamiento y procesamiento de información se llevaron a cabo en un STB de marca EITV.

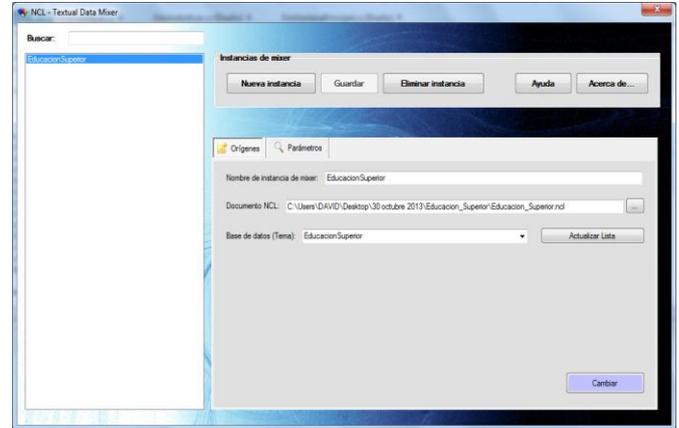


Figura 13. Aplicación MIXER

Elemento	Tabla	Estado	Propiedad	Valor
ctxListaUniversid...	ListaUniversidades	CORRECTO	Ancho máx. Lienzo (p.px)	600
ctxOpciones	Opciones	CORRECTO	Alto máx. Lienzo (p.px)	200
medFundacionEPN	EPN/1/1	CORRECTO	Tipo de letra	Arial
medFundacionE...	ESPOL/1/1	CORRECTO	Tamaño de letra	50
medFundacionE...	ESPOCH/1/1	CORRECTO	Color	Color (A=255, R=0, G=1)
medFundacionE...	PLICE/1/1	CORRECTO		

Figura 14. Asociación de datos con la aplicación MIXER

Una vez que la aplicación se carga en el STB se visualiza el ícono de interactividad sugiriendo al televidente pulsar la tecla OK del control remoto para poder ingresar a la aplicación, como se muestra en la Fig 15.

En la Fig 16 se muestra la pantalla inicial de la aplicación interactiva que se presenta al televidente al pulsar la tecla OK, en la que se puede apreciar una imagen de fondo, el video de la programación redimensionado y el listado de universidades de categoría A con el primer menú desarrollado. El televidente puede desplazarse entre los elementos del menú con las teclas UP y DOWN del control remoto. El foco se mueve de acuerdo a las indicaciones del televidente señalando la universidad que puede ser elegida y se muestra su respectivo escudo en la parte inferior. Al desplazarse el foco, la imagen del escudo cambia automáticamente para mostrar el escudo de la universidad correspondiente al elemento del menú donde el foco se ha desplazado.

Las universidades presentadas en los elementos del menú se muestran en vistas de cuatro elementos. Al pulsar la tecla OK sobre las viñetas de navegación del menú, se mostrará la vista de cuatro elementos anterior o la vista de cuatro elementos siguiente según corresponda. Cuando se selecciona una universidad pulsando la tecla OK, se presenta el segundo menú desarrollado en el centro del televisor, como se muestra en la Fig 17, ofreciendo al televidente los aspectos que puede visualizar de la universidad elegida.

Los datos correspondientes a la fundación, ubicación, autoridades, lema, sitio web e información de contacto de la

universidad se presentan al televidente tan pronto son seleccionados con el foco del menú. En el caso de los datos de misión, visión, oferta de pregrado y de postgrado, la aplicación pide al televidente pulsar la tecla OK para generar una petición HTTP al servidor remoto a través del canal de retorno y presentar la información en el televisor.



Figura 15. Ícono de interactividad



Figura 16. Aplicación interactiva con el primer menú desarrollado



Figura 17. Aplicación interactiva con ambos menús desarrollados

En cada una de las dos vistas del menú se presenta un elemento de regresar. Al seleccionar uno de estos elementos con la tecla OK del control remoto, la aplicación regresa nuevamente al primer menú para poder elegir una universidad.

10. CONCLUSIONES

El sistema de búsqueda, almacenamiento y procesamiento de información desarrollado genera contenido interactivo haciendo uso de herramientas con una interfaz gráfica amigable. Esto permite a cualquier tipo de usuario, sin necesidad de tener conocimientos en programación, crear

aplicaciones interactivas de manera mucho más sencilla y ahorrar tiempo en la producción de las mismas.

El *plug-in* de generación automática de menús maneja el árbol jerárquico de entidades del IDE NCL Composer a través del *composer-core*. Para que el *plug-in* sea capaz de comunicarse con el *composer-core* y generar el código NCL de menús de manera automática se hizo uso del mecanismo *signals and slots*. Cada vez que se genera un menú con el *plug-in* se crea un nodo de contexto con varias funcionalidades, lo cual facilita notablemente la generación de aplicaciones interactivas de televisión digital.

El *plug-in* para el IDE NCL Composer se realizó implementando los métodos de las interfaces *IPluginFactory* y *IPlugin*, permitiendo que el *plug-in* pueda ser utilizado dentro del IDE al igual que el resto de *plug-ins* oficiales.

El sistema de generación de contenido interactivo posibilita la creación de aplicaciones híbridas a través de la aplicación MIXER, la cual además de asociar en base a metadatos el contenido de las bases de datos con los elementos de menús generados con el *plug-in*, genera de forma automática *scripts* Lua dentro de la carpeta del proyecto.

El servicio web WCF desarrollado es un servicio flexible capaz de emplear varias tecnologías de comunicación distribuidas como SOAP y REST en la misma plataforma. SOAP facilitó la implementación de las aplicaciones cliente y MIXER a través de un proxy mientras que REST permitió enviar datos en texto plano a través del canal de retorno para mostrar información al televidente.

El *bot* de búsqueda implementado facilita la obtención de información en sitios web al presentar los mejores párrafos obtenidos mediante un algoritmo de puntuaciones. El desarrollo de *bots* de búsqueda capaces de obtener información que responda con exactitud a las preguntas del navegante en Internet es aún un campo en desarrollo. En la actualidad se impulsa la investigación de las tecnologías involucradas en la semántica web, la cual nació años atrás para enriquecer la búsqueda y la explotación de los resultados en la web.

REFERENCIAS

- [1] BLANCHETTE, Jasmin; SUMMERFIELD, Mark, "C++ GUI Programming with Qt 4". USA: Prentice Hall, 2006.
- [2] DEITEL, Paul; DEITEL, Harvey, "C# 2010 for Programmers". Cuarta edición. Crawfordsville, USA: Pearson Education, 2011.
- [3] GARCÍA, Thedy. "SISTEMAS DISTRIBUIDOS RESTful". [Online]. <http://tedhygarcia.blogspot.com/2012/06/restful.html#1/2012/06/restful.html>
- [4] Git Foundation. "Git distributed even if your workflow isn't". [Online]. <http://git-scm.com/documentation>
- [5] GOMES, Luiz; FERREIRA Rogério, "Nested Context Model 3.0 Part 1 - NCM Core" PUC-Río, Río de Janeiro.
- [6] GOMES, Luiz; JUNQUEIRA, Simone, "Programando em NCL 3.0", Segunda edición. Río de Janeiro, Brasil, 2012.
- [7] HTML Agility Pack. "HTML Agility Pack. Project Hosting for Open Source Software". [Online]. <http://htmlagilitypack.codeplex.com/>

- [8] KYRNIN, Jennifer, “*What is the DOM*”. [Online]. <http://webdesign.about.com/od/dom/a/aa120202a.htm>
- [9] “*Network Support for the Lua Language. LuaSocket: HTTP support*”. [Online]. <http://w3.impa.br/~diego/software/luasocket/http.html>
- [10] SÁNCHEZ, Cristian, “*Desarrollando Aplicaciones con Visual Basic.NET 2008*”, Primera edición. Lima, Perú. Empresa Editora Macro E.I.R.L, 2008.
- [11] TeleMídia (PUC-Rio). “*How to: build NCL Composer from source code*”. [Online]. http://composer.telemidia.puc-rio.br/en/doc/tutorial/how_to_build_ncl_composer_from_source_code
- [12] TeleMídia (PUC-Rio). “*How to: Create an NCL Composer Plug-in*”. [Online]. http://composer.telemidia.puc-rio.br/en/doc/tutorial/how_to_create_a_plug-in_to_ncl_composer
- [13] TeleMídia (PUC-Rio). “*NCL Composer Plug-ins*”. [Online]. <http://composer.telemidia.puc-rio.br/en/plug-ins/start>
- [14] TeleMídia (PUC-Rio). “*Welcome to NCL Composer*”. [Online]. http://composer.telemidia.puc-rio.br/en/start?redirect=1#welcome_to_ncl_composer