

# Menu Creator: Plug-in de Generación Automática de Menús para el IDE de Desarrollo de Aplicaciones Interactivas NCL Composer

**David Fabián Cevallos Salas**

Escuela Politécnica Nacional, Quito, Pichincha, Ecuador, davidcepn@gmail.com

**Fernando Andrés Cevallos Salas**

Escuela Politécnica Nacional, Quito, Pichincha, Ecuador, epnfernando@gmail.com

**Raúl David Mejía Navarrete**

Escuela Politécnica Nacional, Quito, Pichincha, Ecuador, david.mejia@epn.edu.ec

**Iván Marcelo Bernal Carrillo**

Escuela Politécnica Nacional, Quito, Pichincha, Ecuador, ivan.bernal@epn.edu.ec

## RESUMEN

Este artículo presenta un *plug-in* para el Entorno de Desarrollo Integrado (IDE) NCL Composer, capaz de generar menús con varias características y funcionalidades personalizables. El *plug-in* brinda una interfaz gráfica amigable al usuario para diseñar menús y genera de forma automática el código en lenguaje NCL (*Nested Context Language*) de los menús diseñados. De esta manera, el usuario puede crear fácilmente aplicaciones interactivas basadas en Ginga-NCL para televisión digital, las cuales presenten al televidente menús con un alto nivel de interactividad, en menor tiempo de desarrollo y sin necesidad de tener conocimientos avanzados del lenguaje NCL. Como ejemplo, se presentan tres aplicaciones interactivas que fueron desarrolladas empleando el *plug-in*.

**Palabras claves:** plug-in, Ginga, televisión digital, NCL, Composer.

## ABSTRACT

This paper presents a *plug-in* for the NCL Composer IDE (*Integrated Development Environment*), which is able to generate menus with several customizable features and functionalities. The *plug-in* provides a user-friendly graphical interface to design menus and automatically generates the NCL (*Nested Context Language*) code of the menus designed. In this way, the user is able to easily create interactive applications based on Ginga-NCL for digital television, which show to the viewer menus with a high level of interactivity, in a shorter development time and without requiring advanced knowledge about the NCL language. As example, three interactive applications that were developed using the *plug-in* are presented.

**Keywords:** plug-in, Ginga, digital television, NCL, Composer.

## 1. INTRODUCCIÓN

El estándar de televisión digital ISDB-Tb (*International System for Digital Broadcast, Terrestrial, Brazilian version*) fue adoptado por varios países, principalmente de América del Sur. Una de las ventajas de la televisión digital es la posibilidad de crear aplicaciones interactivas. Para ello, el estándar ISDB-Tb dispone de su propio *middleware* llamado Ginga, el cual está compuesto por dos subsistemas entrelazados que permiten la creación de

aplicaciones interactivas empleando un paradigma de programación declarativo (Ginga-NCL), imperativo (Ginga-J) o ambos (Gomes y Junqueira, 2012).

Para crear aplicaciones interactivas existen al momento pocas herramientas, algunas aún en desarrollo como el IDE NCL Composer. Este IDE permite crear aplicaciones interactivas basadas en Ginga-NCL y está conformado por un conjunto de *plug-ins* que extienden su funcionalidad para facilitar el trabajo al usuario. Sin embargo, el uso de esta herramienta demanda conocimientos del lenguaje NCL, aún con los *plug-ins* existentes, por lo que en este artículo se presenta un *plug-in* que permite generar el código NCL de menús de manera automática, facilitando el desarrollo de aplicaciones interactivas para televisión digital con el paradigma declarativo.

## 2. ENTORNO DE DESARROLLO INTEGRADO NCL COMPOSER

NCL Composer está basado en una estructura de árbol jerárquico de entidades (usualmente llamadas entidades NCL). Este árbol está estructurado según el modelo conceptual NCM (*Nested Context Model*) y hace uso del formato XML (*eXtensible Markup Language*), por lo que cada entidad está definida por una etiqueta y puede contener entidades hijas y ser a la vez hija de otra entidad padre. Estas entidades son las encargadas de generar el documento NCL con el contenido interactivo a presentarse al televidente, como se ilustra en la Figura 1.

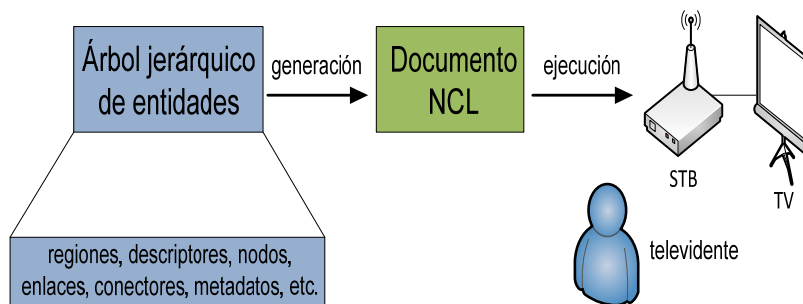


Figura 1: Lógica de trabajo de NCL Composer

El IDE NCL Composer está compuesto de tres partes (TeleMídia PUC-Rio NCL Composer, 2014):

- **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.

Cada *plug-in* en NCL Composer puede gestionar el árbol jerárquico de entidades a través del composer-core, de tal modo que permite al usuario generar el documento NCL evitándole la tarea de programar directamente el código NCL y brindándole una interfaz gráfica amigable (TeleMídia PUC-Rio Plug-ins, 2014).

## 3. 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 establecer la comunicación entre distintos objetos (instancias de clases), sin la necesidad de que dichos objetos sepan el uno del otro (Blanchette y Summerfield, 2006).

La base del mecanismo consiste en que el objeto que desea comunicarse (en este caso una instancia del *plug-in*) debe emitir una señal y el objeto que la recibe (en este caso una instancia del composer-core) debe conectar dicha señal a un determinado *slot*, como se muestra en la Figura 2. Algunas señales a las que responde el composer-core se muestran en la Tabla 1. Un *slot* es un método que realiza cierta funcionalidad. Por lo tanto, 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.

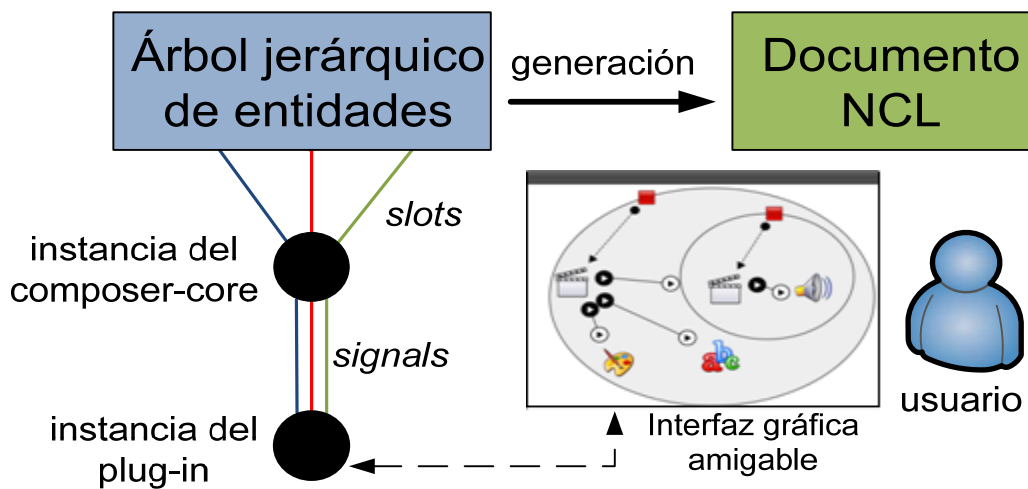


Figura 2: Mecanismo signals and slots

Tabla 1: Algunas señales a las que responde el composer-core

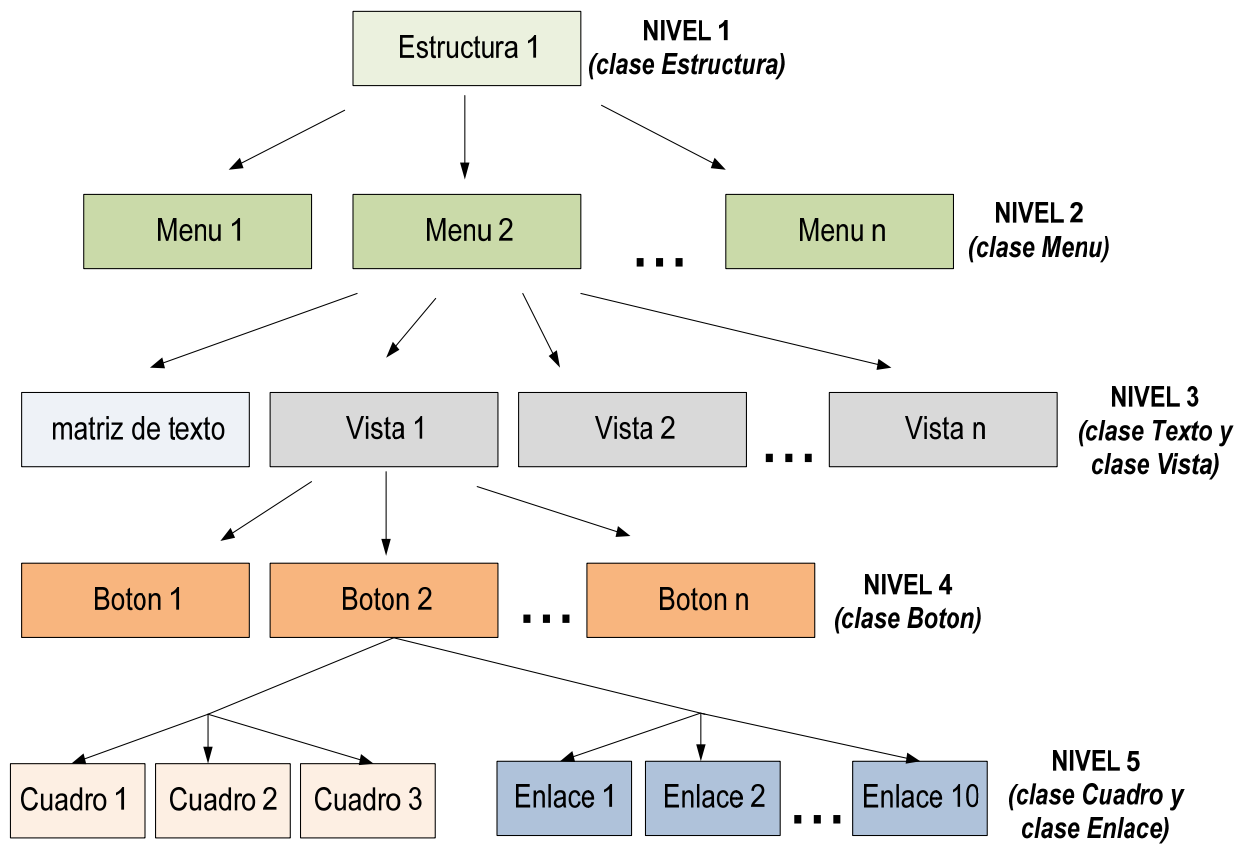
Signal	Slot
addEntity	Añade una nueva entidad NCL.
setAttributes	Establece los atributos de la entidad NCL (atributos de la etiqueta XML).
removeEntity	Elimina una entidad NCL.
setPluginData	Guarda los datos del <i>plug-in</i> en el archivo del proyecto Composer (.cpr).
setCurrentProjectAsDirty	Establece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

#### 4. DISEÑO E IMPLEMENTACIÓN DEL PLUG-IN

El *plug-in* se ha desarrollado haciendo uso de Qt Creator con su lenguaje nativo C++ y lleva por nombre Menu Creator. Ha sido diseñado definiendo una estructura jerárquica de clases y dos clases adicionales que implementan los métodos de las interfaces *IPluginFactory* e *IPlugin*.

La estructura jerárquica de clases está conformada por cinco niveles y se muestra en la Figura 3. El primer nivel contiene la clase *Estructura*. Una instancia de esta clase tiene referencias a una o más instancias de la clase *Menu* 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 (una vista es un fragmento de un menú) y la posición que ocupa el menú en la pantalla. Cada instancia de tipo *Menu* tiene referencias a instancias de la clase *Texto* (dispuestas en forma de matriz), correspondientes al texto de cada uno de los elementos que conforman el 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 *Boton*, 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 (foco).

La clase *Cuadro* es la abstracción de tres entidades NCL: una región que indica en donde se posicionará el elemento del menú y sus dimensiones, un elemento media que indica la imagen a abrir y un descriptor para enlazar la región con el elemento media. Además, cada instancia de la clase *Boton* tiene referencias a diez instancias de la clase *Enlace* del nivel 5, correspondientes a las posibles acciones que realiza el televidente a través del control remoto: ir arriba, ir abajo, ir derecha, ir izquierda, presionar botón ROJO, VERDE, AMARILLO, AZUL, ESC y OK. Cada instancia de tipo *Enlace* representa una entidad NCL de tipo *link* que permite la navegación a través de los elementos del menú.



**Figura 3: Estructura jerárquica de clases del plug-in Menu Creator**

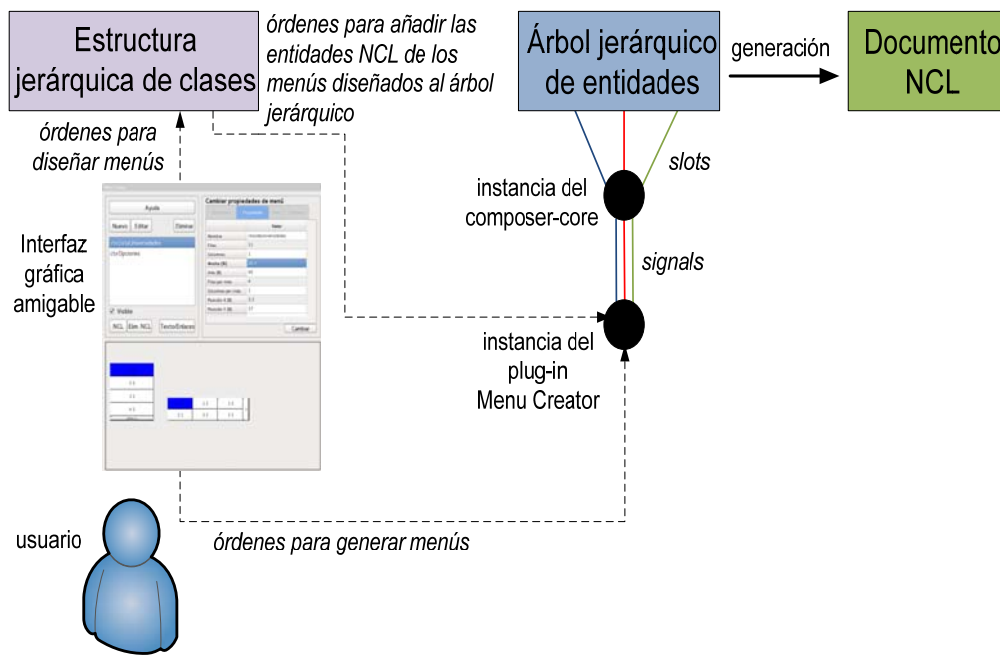
La clase que implementa los métodos de la interfaz *IPluginFactory* permite crear una nueva instancia del *plug-in* cuando se crea un proyecto de NCL Composer. Además, contiene información sobre el *plug-in*, la que se presentará en el menú *Help About Plugins/Details* del composer-gui, como el nombre registrado del *plug-in* (ej. Menu Creator), la versión (ej. 0.1), el propietario (ej. Escuela Politécnica Nacional), entre otros datos.

La clase que implementa los métodos de la interfaz *IPlugin* contiene la funcionalidad del *plug-in*. Algunos métodos de esta interfaz se muestran en la Tabla 2. Cuando se abre un proyecto en NCL Composer, una instancia de esta clase se enlaza con una instancia del proyecto y será utilizada cada vez que el proyecto sufra un cambio para poder guardar la información del trabajo del usuario.

**Tabla 2: 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 .cpr.
QWidget* getWidget()	Obtiene el objeto que corresponde a la interfaz gráfica de usuario del <i>plug-in</i> a instanciar.
bool saveSubsession()	Método que se ejecuta cuando se guarda el proyecto o sus cambios.

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 Figura 4. Para ello, la instancia del *plug-in* (creada gracias a la clase que implementa los métodos de la interfaz *IPluginFactory*) emite las señales correspondientes a una instancia del composer-core mediante el mecanismo *signals and slots* para generar el documento NCL.



**Figura 4: Lógica de trabajo del plug-in Menu Creator**

La Figura 5 presenta la ventana principal de trabajo del *plug-in* Menu Creator. La parte superior izquierda de la ventana está destinada a listar los menús diseñados con el *plug-in* y a mostrar las diferentes opciones que el *plug-in* ofrece a manera de botones. La parte superior derecha se utiliza para el ingreso de los datos dependiendo del botón que haya seleccionado el usuario. La parte inferior muestra el menú o los menús que se han diseñado y su posición en el televisor mediante un navegador interactivo, el cual permite al usuario arrastrar cada menú para determinar su posición y navegar entre los elementos del mismo.



**Figura 5: Ventana principal de trabajo del plug-in Menu Creator**

Por medio de esta interfaz gráfica, el *plug-in* permite generar menús ofreciendo varias características y funcionalidades tales como el uso de estilos gráficos (plantillas gráficas), menús con distintas propiedades de diseño (número de filas, número de columnas, dimensiones del menú, entre otras), división de un menú en vistas (fragmentos), uso de viñetas de navegación (símbolos gráficos que indican al televidente cómo desplazarse entre las distintas vistas de un menú), 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.

El *plug-in* generará un nodo de contexto NCL por cada menú diseñado. Dicho nodo contiene todo el diseño del menú que realizó el usuario, así como también los elementos media que estructuran dicho menú. Esto es de gran utilidad, ya que el usuario podrá realizar varias acciones sobre el menú diseñado tales como iniciar su presentación, detenerlo, pausarlo o reiniciarlo, al ordenar la realización de dichas acciones sobre el nodo de contexto que representa al menú. Además, la manipulación del menú dentro del diseño de la aplicación interactiva se simplifica, pues un nodo de contexto puede ser manipulado como una única entidad (Gomes y Ferreira, 2012), evitando de esta manera la necesidad de manipular directamente cada elemento media que estructura el menú.

## 5. EJEMPLOS DE APLICACIONES INTERACTIVAS DESARROLLADAS

Empleando el *plug-in* Menu Creator se han desarrollado, como ejemplo, tres aplicaciones interactivas que emplean menús para permitir al televidente seleccionar la información que desea visualizar mediante el control remoto.

### 5.1 PRIMERA APLICACIÓN INTERACTIVA: FAUNA DE LAS ISLAS GALÁPAGOS

La primera aplicación interactiva presenta información acerca de la fauna de algunas islas del archipiélago de Galápagos, las cuales se listan en un único menú. Cuando la aplicación interactiva es cargada, se presenta al televidente un ícono de interactividad como el de la Figura 6, sugiriéndole pulsar el botón OK del control remoto.



**Figura 6: Ícono de interactividad de la aplicación interactiva de las Islas Galápagos**

Al pulsar el botón OK, se presenta la aplicación al televidente con una imagen de fondo, el video de la programación redimensionado en un costado y el menú desarrollado con el listado de islas. El televidente puede seleccionar una isla de su interés mediante el botón OK del control remoto. La información con la fauna de la isla seleccionada se presenta en el lado izquierdo, como se puede apreciar en la Figura 7.



**Figura 7: Aplicación interactiva que presenta información de las Islas Galápagos**

## **5.2 SEGUNDA APLICACIÓN INTERACTIVA: UNIVERSIDADES CATEGORÍA A (2009-2013)**

La segunda aplicación interactiva desarrollada presenta información general acerca de las once universidades del Ecuador que conformaron la categoría A durante el período 2009-2013, como resultado de la evaluación realizada por el CONEA (Consejo Nacional de Evaluación y Acreditación de la Educación Superior del Ecuador). Esta aplicación posee dos menús y además dispone de *scripts* Lua que permiten obtener información de datos como la misión, visión y oferta de pregrado y postgrado de las universidades a través del canal de retorno.

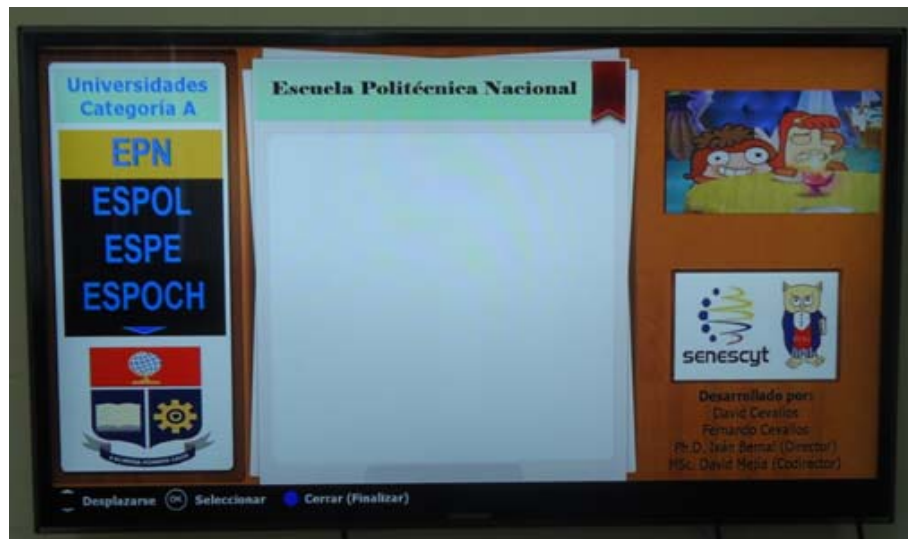
Una vez que la aplicación interactiva se carga, se visualiza el ícono de interactividad sugiriendo al televidente pulsar el botón OK del control remoto para poder ingresar a la aplicación, como se muestra en la Figura 8.



**Figura 8: Ícono de interactividad de la segunda aplicación interactiva**

En la Figura 9 se muestra la pantalla inicial de la aplicación interactiva que se presenta al televidente al pulsar el botón OK, en la que se puede apreciar una imagen de fondo, el video de la programación redimensionado y las

cuatro primeras universidades de categoría A en el primer menú desarrollado. El televidente puede desplazarse entre los elementos del menú con las teclas ARRIBA y ABAJO 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.



**Figura 9: Segunda aplicación interactiva con el primer menú desarrollado**

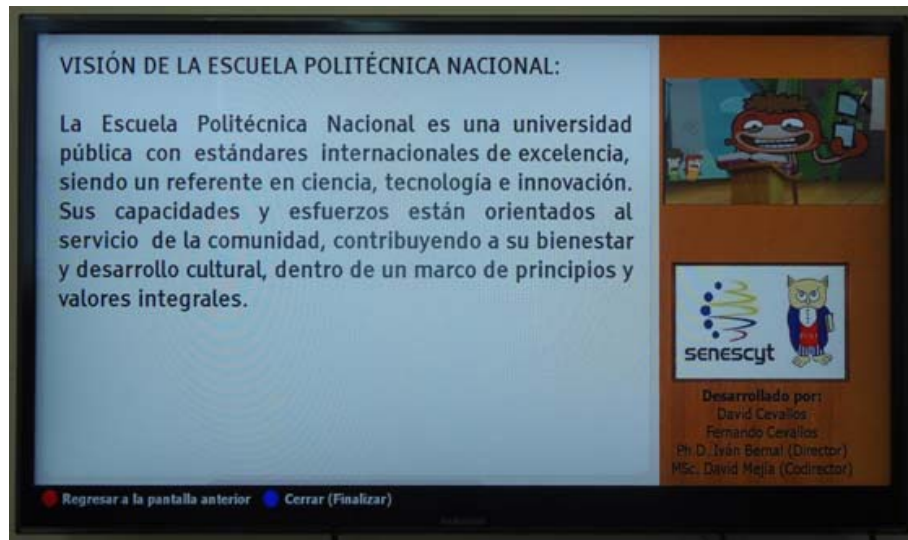
Las universidades que conforman el menú se muestran en vistas de cuatro elementos. Al pulsar el botón OK sobre las viñetas de navegación del menú, se mostrará la vista de cuatro elementos anterior o la vista de cuatro elementos siguiente según corresponda. Cuando se selecciona una universidad pulsando el botón OK, se presenta el segundo menú desarrollado en el centro del televisor, como se muestra en la Figura 10, 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 botón OK del control remoto.



**Figura 10: Segunda aplicación interactiva con ambos menús desarrollados**



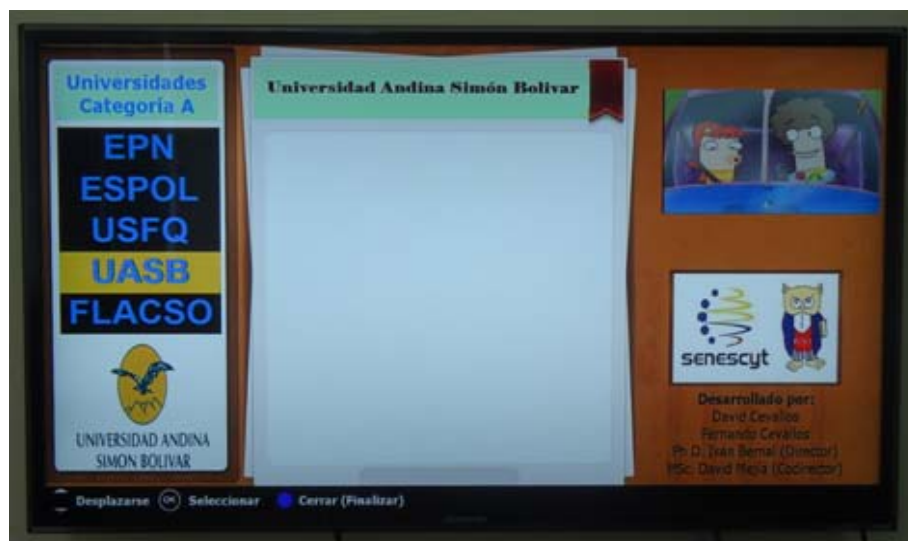
Para el caso de los datos de la misión, visión y oferta de pregrado y postgrado, la aplicación genera una petición a un servidor remoto a través del canal de retorno y presenta la información obtenida en el televisor, en una nueva pantalla con una segunda imagen de fondo, como se puede apreciar en la Figura 11.



**Figura 11: Presentación de información obtenida a través del canal de retorno**

### 5.3 TERCERA APLICACIÓN INTERACTIVA: UNIVERSIDADES CATEGORÍA A (2013-ACTUAL)

A partir de la segunda aplicación se ha desarrollado, como ejemplo de la flexibilidad que el *plug-in* brinda en la creación de aplicaciones interactivas, una tercera aplicación que presenta información general acerca de las cinco universidades que en la actualidad conforman la categoría A (año 2014), como resultado de la evaluación realizada por el CEAACES (Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior) en noviembre de 2013. El funcionamiento de esta tercera aplicación interactiva es exactamente igual que el de la segunda, y para obtenerla únicamente fue necesario modificar las propiedades de diseño del primer menú, de manera que ahora presente un listado con las cinco universidades. La Figura 12 presenta la aplicación interactiva con el primer menú modificado, y la Figura 13 presenta la aplicación con ambos menús desarrollados.



**Figura 12: Tercera aplicación interactiva con el primer menú modificado**



Figura 13: Tercera aplicación interactiva con ambos menús desarrollados

## 6. CONCLUSIONES

El *plug-in* de generación automática de menús para el IDE NCL Composer facilita la creación de aplicaciones interactivas basadas en Ginga-NCL, permitiendo que un usuario genere contenido interactivo sin tener conocimientos avanzados del lenguaje NCL. Los menús que el *plug-in* es capaz de crear son de un alto nivel de interactividad, porque genera 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.

El *plug-in* fue creado en Qt Creator con su lenguaje nativo C++ y se implementaron los métodos de las interfaces IPluginFactory e IPlugin, permitiendo que el *plug-in* pueda ser utilizado dentro del IDE al igual que el resto de *plug-ins* originales.

NCL Composer se basa en un árbol jerárquico de entidades y hace uso del formato XML para la generación de aplicaciones interactivas. Una instancia del *plug-in* desarrollado maneja dicho árbol a través de una instancia del composer-core. Para que la instancia del *plug-in* sea capaz de comunicarse con la instancia del composer-core y generar el código NCL de menús de manera automática se hizo uso del mecanismo *signals and slots*.

## REFERENCIAS

- Blanchette, J., y Summerfield, M. (2006). *C++ GUI Programming with Qt 4*, 1<sup>ra</sup> edición, Prentice Hall, USA.
- Gomes, L.F., y Ferreira, R. (2012). "Nested Context Model 3.0 Part 1 - NCM Core", Monografía en Ciencias de Computación, Pontificia Universidad Católica de Río de Janeiro, Río de Janeiro, Brasil.
- Gomes, L.F., y Junqueira, S.D. (2012). *Programando em NCL 3.0*, 2<sup>da</sup> edición, Ed. Río de Janeiro, Brasil.
- TeleMídia PUC-Rio NCL Composer. (2014). Welcome to NCL Composer, [http://composer.telemidia.puc-rio.br/en/start?redirect=1#welcome\\_to\\_ncl\\_composer](http://composer.telemidia.puc-rio.br/en/start?redirect=1#welcome_to_ncl_composer). (2014/02/22).
- TeleMídia PUC-Rio Plug-ins. (2014). How to: Create an NCL Composer Plug-in, [http://composer.telemidia.puc-rio.br/pt-br/doc/tutorial/how\\_to\\_create\\_a\\_plugin\\_to\\_ncl\\_composer](http://composer.telemidia.puc-rio.br/pt-br/doc/tutorial/how_to_create_a_plugin_to_ncl_composer). (2014/02/24).

## *Autorización y Renuncia*

*Los autores autorizan a LACCEI para publicar el escrito en las memorias de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que está expresado en el escrito.*