

EXPERIENCIAS Y PROTOTIPOS DE APLICACIONES DE TV DIGITAL INTERACTIVAS.

Fernanda Oyarzo, Franco Herrera, Mirtha Miranda y Sandra Casas
Laboratorio de TV Digital – Instituto de Tecnología Aplicada
Universidad Nacional de la Patagonia Austral
Rio Gallegos – Argentina
laboratorio.tvdt.unpa.uarg@gmail.com

Palabras clave: TV Digital, Ginga, Aplicaciones interactivas, NCL, LUA, Prototipos.

Resumen: La TV Digital es una tecnología, que va a transformar la experiencia de ver televisión. Imagen y sonido en alta definición junto con la interactividad son posibles gracias al pasaje de la transmisión analógica a la digital, la cual permite enviar datos, video, audio, aplicaciones, etc. a través de los canales de transmisión. Estas señales digitales son más eficientes que las analógicas y tienen como principal ventaja la posibilidad de enviar servicios a través del mismo canal, permitiendo un uso eficiente del espectro de transmisiones.

Emergen nuevas posibilidades para el desarrollo de software, puesto que una aplicación interactiva de TV Digital es un software, específicamente una aplicación multimedia a través de la cual el televidente puede interactuar vía control remoto. Esto significa que el televidente también puede recibir video/audio, software que posibilitan que interactúe con el contenido vinculado.

En estos nuevos escenarios resulta necesario llevar adelante experiencias de desarrollo de aplicaciones interactivas para TV Digital. Poner en práctica procesos, estrategias y herramientas, y su consecuente análisis promete arribar a propuestas razonables. En este sentido, este trabajo condensa las experiencias en el desarrollo de prototipos realizados en el Laboratorio de TV Digital de la UNPA que se iniciaron en el año 2011.

1. INTRODUCCIÓN

La TV Digital [1] [2] [3] es el conjunto de tecnologías de generación, transmisión y recepción de imagen y sonido a través de información digital. Esto permite que los errores en la transmisión y recepción de la TV analógica (“fantasmas” y “lluvia”) se corrijan y de esta manera no existan interferencias ni distorsiones en pantalla, generando una imagen y sonido superior a la TV actual (analógica). Para tener acceso a la TV Digital (terrestre), cada televidente deberá contar con un equipo receptor o puede estar integrado dentro de algunos nuevos televisores que explícitamente lo informen.

El Sistema Argentino de Televisión Digital Terrestre (SATVD-T) [3] como todos los otros sistemas de televisión respeta el modelo broadcasting. Este modelo se basa en la existencia de: un conjunto reducido de emisores de contenido (canales de televisión) y un conjunto suficientemente grande de receptores (televidentes). En este modelo, los receptores no tienen

control directo sobre el contenido emitido sino que lo tienen sobre la sintonización de un programa dado. Esto implica que el modelo puede implementarse sin un canal de retorno que conecte a los televidentes a los emisores u otros servicios.

El SATVD-T está basado en la norma Integrated Service Data Broadcasting (ISDB-Tb) o Transmisión Digital de Servicios Integrados. El mismo consiste en un conjunto de normas japonesas (la b indica que es la Versión Brasileña), donde se definen formas para transmitir contenidos digitales por aire. Los datos pueden ser actualizaciones de aplicaciones o sistemas de archivos. Esta norma utiliza el middleware denominado Ginga [4] que permite ejecutar aplicaciones NCL [5] / LUA [6].

El decodificador (o set-top-box) es un dispositivo que permite la recepción tanto de la señal digital como la analógica, y genera la señal de video y audio que alimenta el televisor o display. Su objetivo es realizar todas las operaciones necesarias para que las señales recibidas a modo de radio frecuencia se transformen en video y audio: captación de las señales, decodificación, descompresión, etc. El set-top-box del SATVD-T implementa el middleware Ginga ar [7].

Las aplicaciones que son parte de la transmisión digital respetan el mismo modelo, pero son los televidentes los que tienen el control sobre el uso de las mismas, dado que generalmente la aplicación puede ser desactivada, o el televidente puede cancelar la ejecución de la aplicación cambiando el canal que sintoniza. Las mismas están asociadas al video principal del canal en el cual se transmiten y por ende aumenta la experiencia del televidente respecto de la televisión tradicional, con nuevos tipos de interacciones entre el cliente y los contenidos. Es por esto que suele hablarse de TV Digital interactiva.

El presente trabajo plantea un acercamiento al desarrollo de aplicaciones para TV Digital interactivas. Con el objeto de obtener experiencias concretas de desarrollo de software, se representa un escenario viable que consta de un programa de TV y usuarios reales, un equipo de desarrollo que requiere organizar las tareas mediante herramientas básicas: un proceso, artefactos simples que vehiculen la especificación, diseño, implementación y prueba de los prototipos.

Este documento se estructura de la siguiente manera: en la Sección 2 se presentan brevemente las tecnologías que intervienen en el desarrollo de TV Digital, en la Sección 3 se exponen de manera sintética el proceso, las herramientas y artefactos empleados en el desarrollo de prototipos de TV Digital interactiva; en la Sección 4 se presentan tres de las experiencias realizadas; en la Sección 5 analizan los resultados obtenidos; en la Sección 6 se presentan algunos trabajos relacionados; y en la Sección 7 se presentan se exponen las conclusiones.

2. INTERACTIVIDAD, MIDDLEWARE Y LENGUAJES DE PROGRAMACIÓN

2.1 Interactividad

La interacción es un elemento básico en la comunicación humana, pero hasta el momento los ciudadanos sólo han sido espectadores pasivos en la TV actual. La TV Digital le dará al ciudadano la oportunidad de interactuar con los contenidos, tomar decisiones, es decir, tener más control sobre la Televisión facilitando el ejercicio de la ciudadanía.

Los beneficios que dará la interactividad al ciudadano se irán incrementando con el tiempo. Este cambio radical, tecnológico y de objetivos recién se está explorando a nivel mundial y el límite que existe es la imaginación. Científicos, pedagogos, semiólogos y programadores se encuentran trabajando en este proyecto.

- En un primer nivel de interactividad el ciudadano podrá consultar información de la programación, una agenda de programas, el estado del tránsito y del tiempo, restringir contenidos no aptos para niños. Estas serán las primeras muestras de interactividad.

- En un segundo nivel de interactividad el espectador podrá participar, expresarse, opinar e influir en la programación on-line. Así también podrá realizar trámites a través de la TV, anotarse en cursos de educación a distancia etc.

Los productores televisivos (público, privado y de la sociedad civil) tendrán que estar mejor preparados y tener mayor creatividad al manejar estas nuevas variables, dado que el usuario, el ciudadano argentino, será más selectivo al poder optar en lo que quiere o no ver.

En términos computacionales, un programa de TV Digital interactivo es un software, específicamente una aplicación multimedia a través de la cual el televidente puede interactuar vía control remoto [8]. Esto significa que el televidente también puede recibir video/audio, softwares que posibilitan que interactúe con el contenido “vinculado”.

2.2 Ginga

Ginga [9] es el nombre que recibe el middleware que permite ejecutar aplicaciones interactivas dentro de un STB (Set-Top Box). Cómo en el mercado existen STBs de distintos fabricantes y puede variar la plataforma de hardware/software de los mismos, surge la necesidad de contar de un middleware que permita correr aplicaciones sin importar que STB se disponga.

El middleware abierto Ginga se subdivide en dos subsistemas principales interrelacionados, que permiten el desarrollo de aplicaciones siguiendo dos paradigmas de programación diferentes. Estos dos subsistemas se llaman Ginga-J (para aplicaciones procedurales Java) y Ginga- NCL (para aplicaciones NCL).

En Argentina, se trabaja sobre la versión Ginga-NCL [10] y esto se debe a que Ginga-NCL es en la actualidad el único middleware estándar internacional tanto para IPTV como para TV Digital (en todos los tipos de dispositivo). Y también es el único estándar abierto. El mismo fue creado en la Pontífica Universidade Católica de Río de Janeiro (PUC-Rio) y ofrece una infraestructura de presentación de aplicaciones de multimedia/hipermedia desarrolladas sobre el paradigma declarativo, escritas en el lenguaje NCL y el lenguaje de scripting LUA.

Prácticamente todos los middleware para TV Digital terrestre ofrecen soporte para el desarrollo de aplicaciones usando dos paradigmas de programación. En la Tabla 1 se especifican los ambientes de los middleware de los sistemas americanos, europeo, japonés y brasilero, para receptores fijos y móviles.

Ginga.ar [11] es una implementación estándar Ginga-NCL, desarrollada por el equipo de TV Digital del LIFIA-UNLP, a partir de la implementación de referencia Ginga-NCL. El middleware Ginga.ar fue portado a la arquitectura ST e instalado en Set Top Boxes comerciales diseñados y producidos en Argentina. Estos Set Top Boxes son distribuidos por

el proyecto de TV Digital del Gobierno Argentino. Ginga.ar es Software Libre, las licencias utilizadas son GPLv2 y LGPLv2, e incluye librerías con otras licencias libres.

Tabla 1. Ambientes de aplicaciones para receptores fijos y móviles.

Middleware	Sistema de TVD	Ambiente Declarativo	Ambiente Procedural
ACAP	Americano/ATSC	ACAP-X[ATSC A-101 2005] (lenguaje declarativo=como XHTML; lenguaje no declarativo=ECMAScript)	ACAP-J [ATSC A 101 2005] (lenguaje no declarativo = Java)
MHP	Europeo/DVB-T	DVB-HTML [ETSI TS 102 812 V1.2.2 2006] (lenguaje declarativo = como XHTML; lenguaje no declarativo = ECMAScript)	MHP [ETSI TS 102 812 V1.2.2 2006] (lenguaje no declarativo = Java)
ARIB-BML	Japonés/ISDB-T	ARIB – BML [ARIB B-24 2004] (lenguaje declarativo = BML como XHTML; lenguaje no declarativo = ECMAScript)	Opcional (GEM como [ETSI TS 102 819 V1.2.1 2005]) Sin implementación
Ginga	Brasileño/SBTVD	(lenguaje declarativo = NCL; lenguaje no declarativo = LUA)	Ginga-J (lenguaje no declarativo = Java)

2.3 Nexted Context Lenguaje (NCL)

NCL [12] [13] es un lenguaje declarativo basado en el modelo conceptual NCM (Fig. 1), básicamente es un documento XML. El lenguaje define claramente cómo los objetos media (elementos de contenido multimedial, es decir, los elemento a mostrar como por ejemplo videos, imágenes, sonidos, etc.) son estructurados y relacionados, en el tiempo y en espacio. Como es un lenguaje de marcado, no especifica los tipos del contenido de los objetos media de una aplicación.

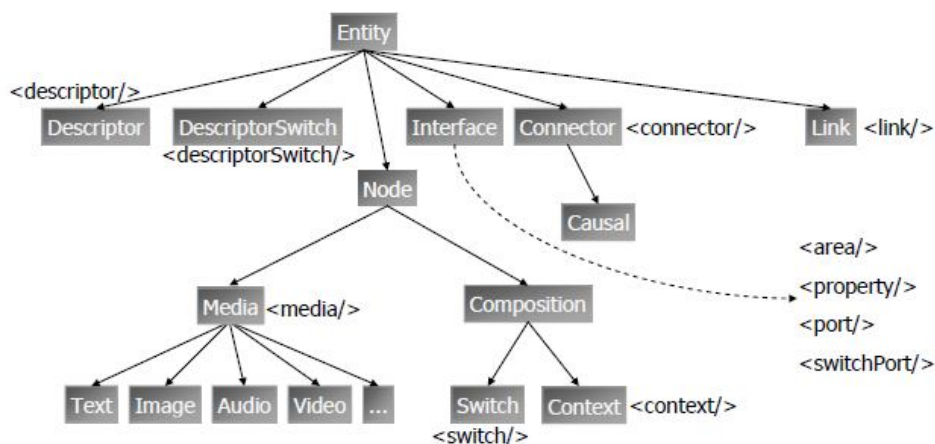


Figura 1: Entidades NCM y elementos del lenguaje NCL

Entre las entidades más significativas de NCL se pueden nombrar a Region, Descriptor, Media, Connector, Link, Context, entre otras. Además, se pueden especificar atributos para describir límites, posiciones, comportamientos de los objetos. Por ser NCL un lenguaje declarativo cuenta con un alto nivel de abstracción, lo que facilita su utilización para

desarrollar aplicaciones interactivas. A su vez, NCL permite ejecutar scripts escritos en lenguaje LUA. La mayoría de las aplicaciones NCL/LUA se componen de dos partes bien identificadas, la lógica de la aplicación se realiza utilizando LUA, mientras que todo lo que implica la presentación y sincronismo espacio/temporal de objetos multimedia se realiza con NCL.

Un programa NCL es un documento XML, esto implica que debe tener un encabezado. Luego se comienza con la estructura de la aplicación con el tag `<ncl>` con su respectivo atributo `id` y `xmlns`. El elemento `<head>` contiene bases de elementos referenciados por el núcleo de la aplicación NCL (definido en el elemento `<body>`), como ser las regiones, los descriptores, las transiciones, los conectores y las reglas. El elemento `<body>` contiene los elementos que definen el contenido de la aplicación propiamente dicha, tal como los objetos media, links, contextos y objetos switches. Se representa la estructura NCL de la siguiente forma:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
<head>
  Región ----- ¿Dónde mostrar los medias?
  Descriptor -----¿Qué propiedad tiene el media?
  Conector -----¿Cuándo mostrar el media?
</head>
<body>
  Puerta -----¿Quién comienza?
  Media ----- ¿Qué mostrar?
  Link -----¿Cuándo mostrar el media?
</body>
```

En el Anexo A se proporcionan brevemente características y ejemplos de NCL.

3. MATERIALES Y MÉTODOS

En la presente Sección, se exponen y describen el proceso empleado y las herramientas y artefactos usados en las experiencias descritas en la Sección 4.

Proceso. Para el desarrollo de los prototipos se ha empleado un proceso incremental, cuyo objetivo principal fue permitir el acercamiento y abordaje a la especificación, diseño, implementación y prueba de aplicaciones de TV Digital interactiva. Las herramientas usadas han sido: plantilla de especificación y prueba (PEyP)¹; cuadros de escena², diagramas NCL³, código NCL. El proceso se inicia desarrollando manualmente un bosquejo de la aplicación, mediante el gráfico de los cuadros de escenas y sus secuencias. Luego y siguiendo la

¹ Artefacto desarrollado en Laboratorio de TV Digital – UNPA (2012)

² Rosica A., “Construcción de un Storyboard”. UNLP FBA.DCV. Cátedra de tecnología - 2006.

www.aroefrpsoca.homelinux.com

³ Cardozo, S. y Schwartz, S., “Crea TV Digital” - 40JAIIO - EST 2011 - ISSN: 1850-2946 – pp 440-460

secuencia de las escenas, por cada una de estas se desarrolla la especificación, el diagrama NCL, se codifica el documento NCL y se llevan adelante las pruebas.

Cuadro de Escena (Fig 2). Es un gráfico realizado manualmente sobre papel, en el que se dibujan pantallas de TV junto a los elementos de interactividad que se incorporarán. Por ejemplo: botones, menús, imágenes, texto, etc. Se desarrollan en forma conjunta con los usuarios.



Figura 2: Cuadro de Escena

Plantilla de Especificación y Prueba (PEyP). Consta de dos partes principales (Fig. 3): especificación y pruebas. La especificación, correspondiente a un determinado prototipo-escena, junto con descripciones textuales de la misma, tales como los eventos que la dominan y la enumeración de elementos medias que la componen. Esta sección de la plantilla pretende documentar los requisitos de la escena.

En la sección destinada a las pruebas, aunque se documenta en la misma plantilla, se lleva adelante al final del proceso. Se indica al menos una prueba por cada evento citado anteriormente, los objetivos de la prueba, los resultados esperados/obtenidos y la solución aplicada cuando los resultados obtenidos no son los esperados.

Prototipo:					Especificación
Escena:					
Denominación escena:					
Descripción:					
Eventos:					
Medias					
Textos	Videos	Gráficos	Audios	Otras	
Observaciones:					Prueba
Prueba #	Objetivo	Resultado Esperado	Resultado Obtenido	Solución	

Figura 3: Modelo de PEyP

Diagrama NCL (Fig. 4): Estos diagramas representan gráficamente la visión estructural de un documento NCL. En esencia es un grafo, en el cual los nodos representan medias y las aristas indican los eventos y acciones que los relacionan. En estos diagramas se representan

con íconos específicos los distintos elementos NCL, como las puertas, las distintas tipos de medias y los eventos / acciones etiquetan las aristas.

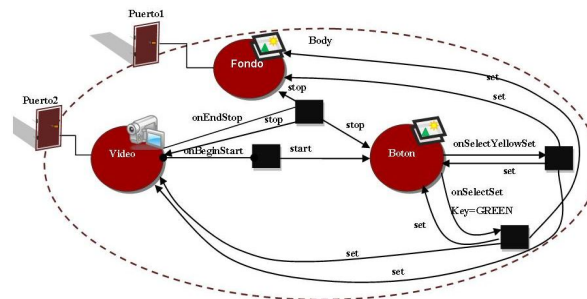


Figura 4: Diagrama NCL

4. EXPERIENCIAS

Las experiencias que a continuación se exponen forman parte del desarrollo de varios y diversos prototipos que se realizaron en la Laboratorio de TV Digital de la UNPA. Este subconjunto de prototipos fue escogido ya que condensan de manera simple pero completa la totalidad de las actividades realizadas.

La iniciativa propuesta consistió en desarrollar una aplicación interactiva para el programa de televisión denominado CAMPUS. CAMPUS es el magazine universitario de la Unidad Académica Río Gallegos, de la Universidad Nacional de la Patagonia Austral. CAMPUS es producido, editado, y filmado por la universidad y se emite semanalmente por canales de aire y cable, de la provincia de Santa Cruz y en un canal de YouTube⁴. Para poder incorporar interactividad a CAMPUS, se desarrollaron prototipos de manera incremental.

El objetivo fue realizar un producto funcional en el cual se mostrara un video del programa, ofreciendo interactividad al televidente a través de botones de colores y las teclas de navegación del control remoto. Estos elementos permitían acceder a información adicional del staff de CAMPUS.

El cliente para esta experiencia estuvo constituido por el productor general, un diseñador, y un conductor de CAMPUS. Mientras que el equipo de desarrollo está formado por dos analistas de sistemas, un licenciado en sistemas y un doctor en sistemas que dirigió el proyecto.

4.1 PROTOTIPO I

El prototipo I tuvo como finalidad aplicar los conceptos de media, región, descriptor, transición, puerta de entrada, conector y link y pasar de una escena a otra, por medio de elementos aportados por la interactividad.

La funcionalidad básica comprendía dos escenas. La primera, denominada “Presentación” en la cual se debía visualizar el video sobre un fondo y una imagen que permite interactuar mediante el control remoto. La segunda escena, llamada “VideoReducido”, debía provocar que al presionar el botón amarillo del control remoto, se redujera el tamaño del video

⁴ <http://www.youtube.com/user/extensionuarg>.

continuando su reproducción. En esta nueva vista, se permitirá retornar a la etapa de “Presentación”, seleccionando el botón verde. La aplicación termina una vez finalizada la reproducción del video.

A continuación se presentan las PEyE, el diagrama NCL, el código NCL (Anexo C), la documentación de las pruebas e imágenes de la aplicación prototipada.

Prototipo: # 01				
Escena: 1-2				
Denominación escena: Presentación				
Descripción: En esta escena se visualizará un video sobre un fondo y un botón mediante el cual se podrá interactuar con las teclas del control remoto.				
Eventos: al presionar el botón amarillo el video se reduce a 45.9%,11.5%,50.2%,39.6% y se muestra la nueva escena#2.				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonAmarillo.jpg		
Observaciones: las imágenes no son definitivas				

Plantilla 1.1. Especificación de la escena 1 - prototipo I

Prototipo: # 01				
Escena: 2-2				
Denominación escena: VideoReducido				
Descripción: En esta escena se visualizará el video con las nuevas dimensiones y un botón verde que permitirá regresar a la escena#1.				
Eventos: al presionar el botón verde la aplicación muestra la escena#1				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonVerde.jpg		
Observaciones: las imágenes no son definitivas				

Plantilla 1.2. Especificación de la escena 2 - prototipo 1.

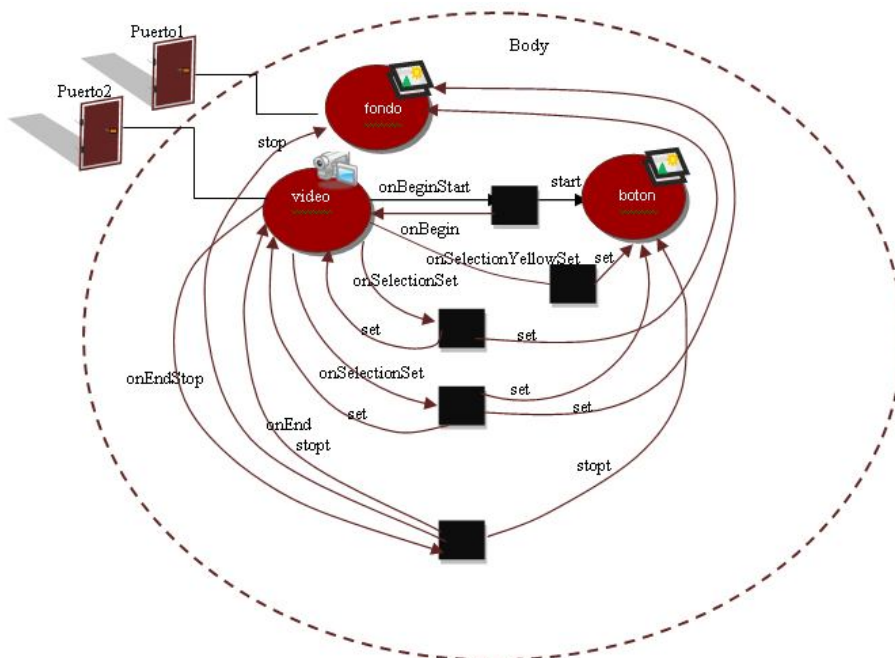


Figura 5. Diagrama NCL - prototipo I

Prototipo: # 01				
Escena: 1-2				
Denominación escena: Presentación				
Descripción: En esta escena se visualizará un video sobre un fondo y un botón mediante el cual se podrá interactuar con las teclas del control remoto.				
Eventos: al presionar el botón amarillo el video se reduce a 45.9%,11.5%,50.2%,39.6% y se muestra la nueva escena#2.				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonAmarillo.jpg		
Observaciones: las imágenes no son definitivas				
Prueba #	Objetivo	Resultado Esperado	Resultado Obtenido	Solución
1	Probar funcionalidad al presionar el botón amarillo	Reducción del video	Video reducido en otra posición incorrecta.	Corrección de las coordenadas.
2	Probar funcionalidad al presionar el botón amarillo	Continuidad de la reproducción del video	El esperado	
3	Probar funcionalidad al presionar el botón amarillo	Se visualiza botón verde	El esperado	
4	Probar funcionalidad al presionar el botón amarillo	Cambio imagen de fondo	El esperado	
5	Finalización aplicación	La aplicación termina cuando el video finaliza	La aplicación no finaliza nunca	Detener todas las medias activas. Se agrega a la media (video) un área con una duración en segundos, igual a la del video.

Plantilla 1.1. Pruebas de la escena 1 - prototipo I

Prototipo: # 01				
Escena: 2-2				
Denominación escena: VideoReducido				
Descripción: En esta escena se visualizará el video con las nuevas dimensiones y un botón verde que permitirá regresar a la escena#1.				
Eventos: al presionar el botón verde la aplicación muestra la escena#1				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonVerde.jpg		
Observaciones: las imágenes no son definitivas				
Prueba #	Objetivo	Resultado Esperado	Resultado Obtenido	Solución
1	Funcionalidad botón verde	Ampliación del video	Video ampliado	
2	Funcionalidad botón verde	Continuidad de la reproducción del video	El esperado	
3	Funcionalidad botón verde	Se visualiza botón amarillo y se detiene el botón verde	El esperado	
4	Funcionalidad botón verde	Cambio imagen de fondo	El esperado	

Plantilla 1.2. Pruebas de la escena 2 - prototipo I

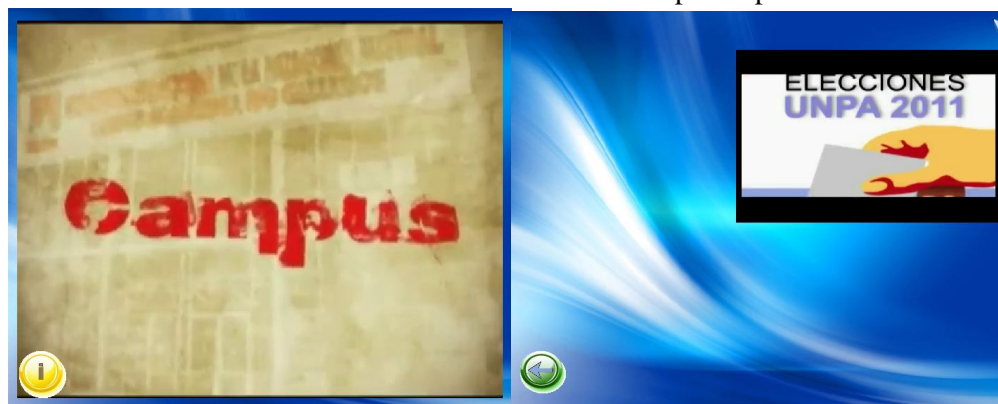


Figura 6. Ejecución del Prototipo I

4.2 PROTOTIPO II

El Prototipo II tuvo por finalidad incorporar nuevas medias y un incremento en la complejidad del Prototipo I. Se agregaron más componentes (media, región, descriptor, transición, puerta de entrada, conector y link) pero a la vez, se otorgó mayor funcionalidad y uso a los elementos conectores y links. El uso de más elementos gráficos, implicó administrar la distribución de la pantalla. En este sentido, el elemento “región” también adquirió más relevancia.

La funcionalidad constó de dos escenas. La primera escena también denominada “Presentación” ya que no varía en cuanto al Prototipo I.

La segunda escena llamada “VideoReducidoMenuDeColores”, que es muy similar al Prototipo I pero se suman cuatro medias, añadiendo para las mismas cuatro regiones y cuatro descriptores para cada media respectivamente. En este caso, estas medias no provocarían eventos y/o acciones.

A continuación se presentan las PEyE, el diagrama NCL, el código NCL (Anexo C), la documentación de las pruebas e imágenes de la aplicación prototipada.

Prototipo: # 02				
Escena: 1-2				
Denominación escena: Presentación				
Descripción: En esta escena se visualizará un video sobre un fondo y un botón mediante el cual se podrá interactuar con las teclas del control remoto.				
Eventos: al presionar el botón amarillo el video se reduce a 45.9%,11.5%,50.2%,39.6% y se muestra la nueva escena#2.				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonAmarillo.jpg		
Observaciones: las imágenes no son definitivas				

Plantilla 2.1. Especificación de la escena 1 - prototipo II

Prototipo: # 02				
Escena: 2-2				
Denominación escena: VideoReducidoMenuDeColores				

Descripción: En esta escena se visualizará el video con las nuevas dimensiones, un menú de colores donde cada uno representará a un participante del programa y un botón verde que permitirá regresar a la escena#1.				
Eventos: al presionar el botón verde la aplicación muestra la escena#1, el menú de colores sobre los participantes no funcionará				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonVerde.jpg frame_blue.jpg frame_red.jpg frame_green.jpg frame_yellow.jpg		

Plantilla 2.2. Especificación de la escena 2 - prototipo II

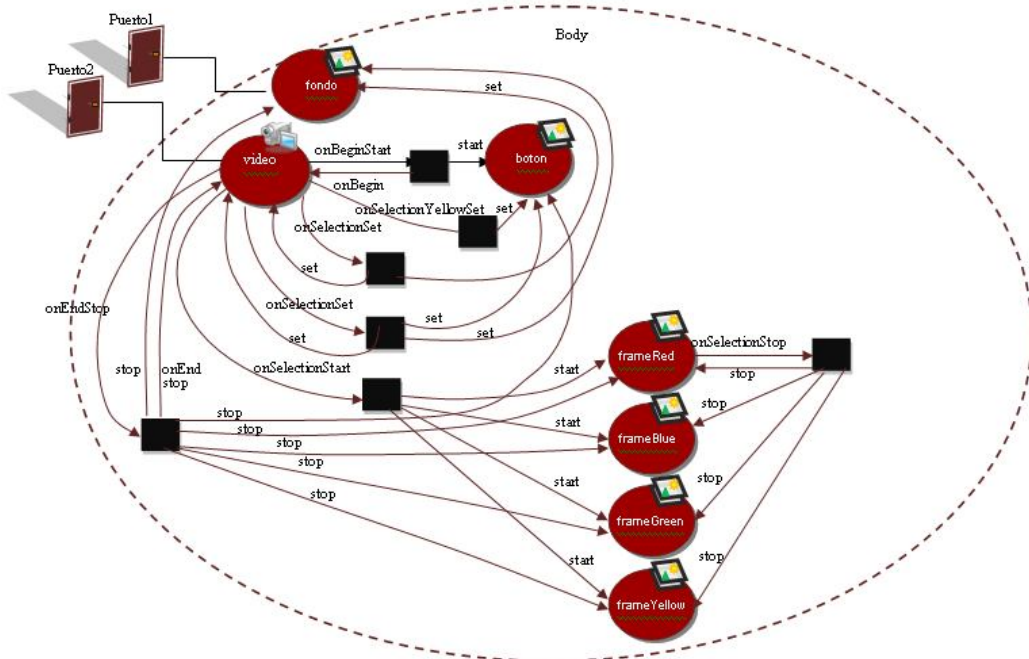


Figura 7. Diagrama NCL - prototipo I

Prototipo: # 02				
Escena: 1-2				
Denominación escena: Presentación				
Descripción: En esta escena se visualizará un video sobre un fondo y un botón mediante el cual se podrá interactuar con las teclas del control remoto.				
Eventos: al presionar el botón amarillo el video se reduce a 45.9%,11.5%,50.2%,39.6% y se muestra la nueva escena#2.				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonAmarillo.jpg		
Observaciones: las imágenes no son definitivas				
Prueba #	Objetivo	Resultado Esperado	Resultado Obtenido	Solución
1	Probar funcionalidad al presionar el botón amarillo	Reducción del video	El esperado	
2	Probar funcionalidad al	Continuidad de la	El esperado	

	presionar el botón amarillo	reproducción del video		
3	Probar funcionalidad al presionar el botón amarillo	Se visualiza botón verde	El esperado	
4	Probar funcionalidad al presionar el botón amarillo	Cambio imagen de fondo	El esperado	
5	Probar funcionalidad al presionar el botón amarillo	Se inician 4 medias que representan el menú de colores	Las medias aparecen superpuestas, no en las coordenadas deseadas	Corrección de las coordenadas.

Plantilla 2.1. Pruebas de la escena 1 - prototipo II

Prototipo: # 02				
Escena: 2-2				
Denominación escena: VideoReducidoMenuDeColores				
Descripción: En esta escena se visualizará el video con las nuevas dimensiones, un menú de colores donde cada uno representará a un participante del programa y un botón verde que permitirá regresar a la escena#1.				
Eventos: al presionar el botón verde la aplicación muestra la escena#1, el menú de colores sobre los participantes no funcionará				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonVerde.jpg frame_blue.jpg frame_red.jpg frame_green.jpg frame_yellow.jpg		
Observaciones: las imágenes no son definitivas				
Prueba #	Objetivo	Resultado Esperado	Resultado Obtenido	Solución
1	Funcionalidad del botón verde	Se detienen las 4 medias que forman el menú de colores	El esperado	
2	Funcionalidad del botón verde	Se detiene el botón verde y se inicia el amarillo	El esperado	
3	Funcionalidad del botón verde	Se modifica la propiedad de la media del fondo	El esperado	
4	Funcionalidad del botón verde	Continuidad de la reproducción del video	El esperado	

Plantilla 2.2. Pruebas de la escena 2 - prototipo II

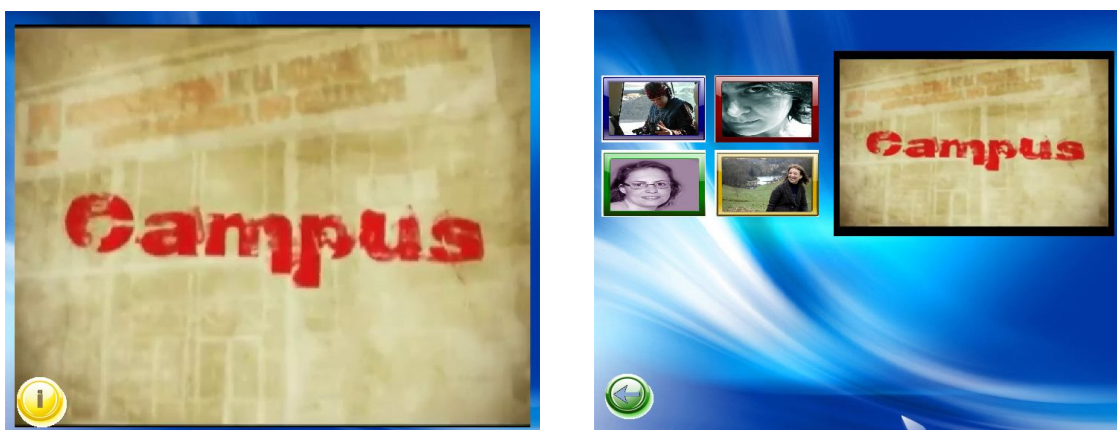


Figura 8. Ejecución del Prototipo II

4.3 PROTOTIPO III

El Prototipo III tuvo por objeto continuar incrementando la complejidad de los prototipos I y II. Por lo cual, se utilizan más conectores y links para trabajar con distintas medias, en algunos casos, de manera concurrente. Estos eventos incorporados permiten no sólo la interacción a través de los botones de color del control remoto, sino que además, se pueden seleccionar medias con los botones de desplazamiento. Se modifican los iconos de inicio de interacción y retorno al menú de la aplicación interactiva, y la forma de seleccionarlos. Se agregan otros elementos multimedia como archivos de texto e imágenes, para mostrar información de las medias incorporadas en el Prototipo II, que representan los 4 botones de color del control remoto. Además de forma interna y para mejorar la legibilidad, se utiliza un archivo (NCL) separado del inicial, para los conectores, debido al incremento de código.

Para este prototipo, la funcionalidad consta de 6 escenas: Presentación, VideoReducidoMenuDeColores y las 4 posibles escenas a seleccionar según la escena de interactividad (rojo, amarillo, azul, verde).

Escena de Presentación, sobre un fondo se reproduce el video y se agrega un ícono que representa la forma de interactuar, a través del botón menú del control remoto.

Escena VideoReducidoMenuDeColores. El fondo es modificado, el video reducido sigue su reproducción, se modifica el ícono que permite volver a la pantalla anterior y se agregan 4 imágenes que representan los colores del control remoto, cada una de las cuales permite acceder a una interacción con la aplicación mostrando información de cada integrante del staff de CAMPUS.

Escena InformacionExtraPersonajeSeleccionado: a) Al seleccionar botón: el fondo es el mismo que en la escena VideoReducidoMenuDeColores y el video sigue reproduciéndose con la misma distribución. Se detienen las medias correspondientes a los colores de control remoto rojo, verde y amarillo. Se amplía el tamaño de la media azul seleccionada, se agregan un texto descriptivo y una imagen de fondo a este texto. Además se modifica el icono atrás por el menú, que permite volver a la escena anterior. b) Al seleccionar botón rojo: similar a la escena anterior. Se detienen las medias correspondientes a los colores de control remoto azul, verde y amarillo. Se amplía el tamaño de la media roja seleccionada y se agrega texto descriptivo e imagen correspondiente a esta media. c) Al seleccionar botón verde: similar a las escenas anteriores. Se detienen las medias correspondientes a los colores de control remoto azul, rojo y amarillo. Se amplía el tamaño de la media verde seleccionada y se agrega texto descriptivo e imagen correspondiente a esta media. d) Al seleccionar el botón amarillo: similar a las escenas anteriores. Se detienen las medias correspondientes a los colores de control remoto azul, rojo y verde. Se amplía el tamaño de la media amarilla seleccionada y se agrega texto descriptivo e imagen correspondiente a esta media.

A continuación se presentan los cuadros de escena (Anexo B), las PEyE, el diagrama NCL, el código NCL (Anexo C), la documentación de las pruebas e imágenes de la aplicación prototipada. Por cuestiones de espacio la PEyP se presenta completa al principio.

Prototipo: # 03
Escena: 1-3
Denominación escena: Presentación
Descripción: En esta escena se visualizará un video sobre un fondo y un botón mediante el cual se podrá

interactuar con las teclas del control remoto.				
Eventos: al presionar el botón menú el video se reduce a 45.9%,11.5%,50.2%,39.6% y se muestra la nueva escena#2.				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonMenu.gif fondoMaximizado.gif		
Observaciones: las imágenes no son definitivas				
Prueba #	Objetivo	Resultado Esperado	Resultado Obtenido	Solución
1	Visualización de las medias	Las medias se visualizan correctamente	el esperado	
2	Funcionalidad del Botón Menú	Al presionar el botón (Menú del control remoto se visualiza escena#2	El esperado	

Plantilla 3.1. Especificación y Pruebas de la escena 1 – prototipo III

Prototipo: # 03				
Escena: 2-3				
Denominación escena: VideoReducidoMenuDeColores				
Descripción: En esta escena se visualizará el video con las nuevas dimensiones, un menú de colores donde cada uno representará a un participante del programa y un botón atrás que permitirá regresar a la escena#1.				
Eventos:				
<ul style="list-style-type: none"> • Al presionar el botón atrás la aplicación muestra la escena#1 • El menú de colores sobre los participantes queda habilitado para su funcionamiento. • Al seleccionar el video mediante las teclas arriba o abajo y hacer clic en OK o enter la aplicación volverá a la escena#1 				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonAtras.gif frame_blue.gif frame_red.gif frame_green.gif frame_yellow.gif fondoMinimizado.gif		
Observaciones: las imágenes no son definitivas				
Prueba #	Objetivo	Resultado Esperado	Resultado Obtenido	Solución
1	Funcionalidad botón Atrás	Volver a la escena #1	El esperado	-
2	Funcionalidad menú colores	Al presionar los botones rojo, verde, amarillo o azul. Se visualiza escena#3 con las medias correspondientes a cada color	Algunas medias se visualizan en coordenadas incorrectas y algunas quedan superpuestas.	Se corrigen regiones, Se agregan y modifican conectores y links.
3	Navegación por tecla	Desplazamiento en el menú de selección (video - botón)	No se desplaza	Se corrigen focusIndex en descriptores
4	Selección por menú video	Al seleccionar el video y hacer enter se regresa a la escena #1	El esperado	-
5	Selección por menú botón atrás	Al seleccionar el botón y hacer enter se regresa a la escena#1	El esperado	-

Plantilla 3.2. Especificación y Pruebas de la escena 2 - prototipo III

Prototipo: # 03
Escena: 3-3

Denominación escena: informacionExtraPersonajeSeleccionado				
Descripción: En esta escena se visualizará el video con las mismas dimensiones de la escena#2, el frame del personaje seleccionado con la dimensión (1%,12%,42.8%,39.6%), un texto sobre ese un rectángulo y un botón Menú que permitirá regresar a la escena#2.				
Eventos:				
<ul style="list-style-type: none"> Al presionar alguno de los colores del menú (rojo, verde, amarillo, azul) el frame del color seleccionado tomara las nuevas dimensiones, se visualizara un rectángulo y sobre esta última media un texto. Al presionar tecla de menú del control remoto, la aplicación muestra la escena#2. Al seleccionar el video mediante las teclas arriba o abajo y hacer clic en OK o enter la aplicación volverá a la escena#1 				
Medias				
Textos	Videos	Gráficos	Audios	Otras
	Campus.avi	botonMenu.gif botonMenu2.gif frame_seleccionado.gif descripcion.gif descripcion.txt		
Observaciones: las imágenes no son definitivas				
Prueba #	Objetivo	Resultado Esperado	Resultado Obtenido	Solución
1	Funcionalidad del frame azul del menú de colores	Al presionar el botón azul del control remoto, iniciar media descripción y media texto (información personaje del botón azul). Detener los frame verde, rojo y amarillo. Ampliar la resolución del frame azul (media).	La visualización de las medias es correcta. Las coordenadas para la presentación de la media texto no son las correctas. No se detienen las medias verde, rojo y amarillo. Se amplia correctamente el frame azul.	Se corrige la región asociada a la media texto. Se agrega conector y link para la detención de los frame (verde, rojo y amarillo).
2	Funcionalidad del frame verde del menú de colores	Al presionar el botón verde del control remoto, iniciar media descripción y media texto (información personaje del botón verde). Detener los frame azul, rojo y amarillo. Ampliar la resolución del frame verde (media).	La visualización de las medias es correcta. No se detienen las medias azul, rojo y amarillo. Se amplia correctamente el frame verde.	Se utiliza conector y se agrega link para la detención de los frame (azul, rojo y amarillo). Se genera código ncl de conectores para importar desde el código principal.
3	Funcionalidad del frame rojo del menú de colores	Al presionar el botón rojo del control remoto, iniciar media descripción y media texto (información personaje del botón rojo). Detener los frame azul, verde y amarillo. Ampliar la resolución del frame rojo (media).	El esperado	-
4	Funcionalidad del frame amarillo del menú de colores	Al presionar el botón amarillo del control remoto, iniciar media descripción y media texto (información personaje del botón amarillo). Detener los frame azul, verde y rojo.	El esperado	-

		Ampliar la resolución del frame amarillo (media).		
5	Funcionalidad del botón Menú	Al presionar botón menú del control remoto, regresa a la escena #2.	Al presionar el botón menú, retornaba a la escena #1.	Se agregó otra media botonMenu2, con el mismo recurso que la media botonMenu. Agregar conector y link.
6	Selección por menú video	Al seleccionar el video y hacer enter se regresa a la escena #1	El esperado	

Plantilla 3.3. Especificación y Pruebas de la escena 3 - prototipo III

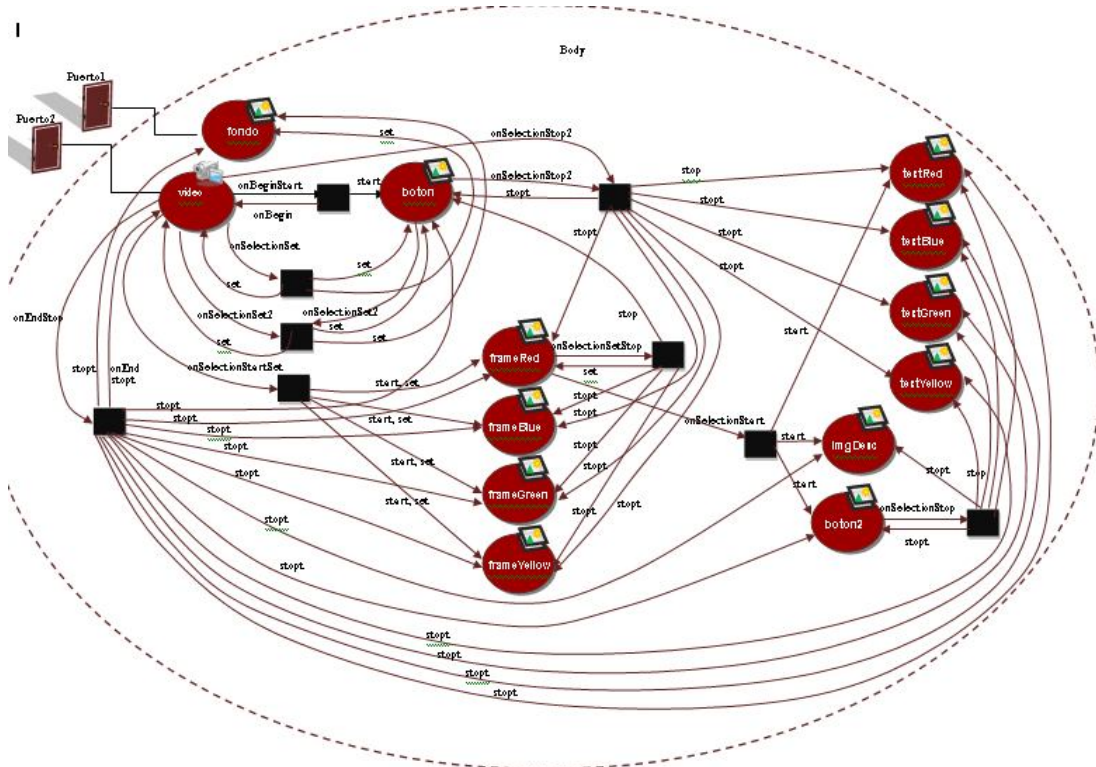


Figura 9. Diagrama NCL - prototipo III

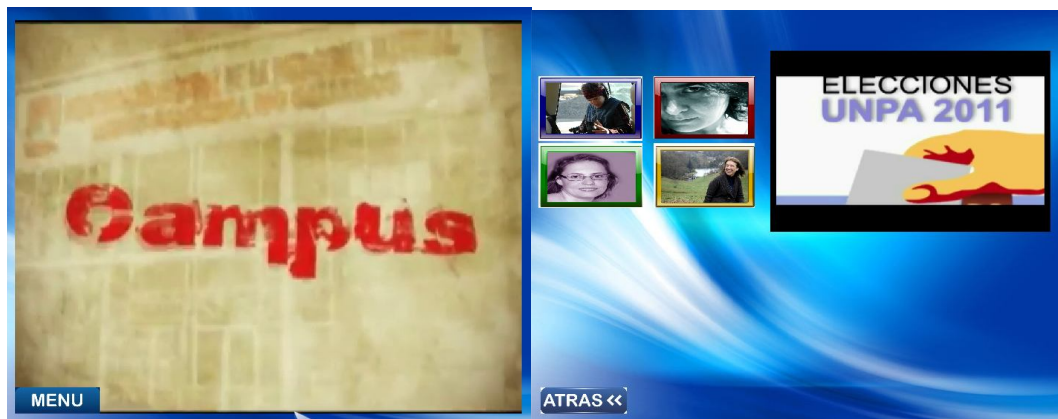




Figura 10. Ejecución del Prototipo III

5. ANÁLISIS DE LA EXPERIENCIA

En la Tabla 2 se presenta el análisis de las experiencias realizadas, identificando los aspectos positivos y negativos de las herramientas/artefactos empleados en el desarrollo de los prototipos.

Tabla 2. Análisis de Experiencias.

Herramienta / Artefacto	Aspecto Positivo	Aspecto Negativo
Cuadro de Escena	<ul style="list-style-type: none"> - Muy adecuada para la comunicación con los usuarios. - Práctica y útil para relevar y capturar requerimientos de los usuarios. - Fácil para mapear a PEyP. 	<ul style="list-style-type: none"> - La flexibilidad, libertad y falta de restricciones para su elaboración dejan demasiados detalles sin definir en análisis inicial, a los que hay que retornar durante el diseño e incluso la implementación.
PEyP	<p><u>Especificación:</u></p> <ul style="list-style-type: none"> - Apropiaada para la comunicación entre los desarrolladores. - Sencilla de elaborar y actualizar. 	<p><u>Especificación:</u></p> <ul style="list-style-type: none"> - Ambigüedad e incompletitud para especificar la ubicación de las medias. (espacio-tiempo)

	<ul style="list-style-type: none"> - Relativamente simple para mapear de diagrama NCL. <p><u>Pruebas:</u></p> <ul style="list-style-type: none"> - Clara y eficiente para el desarrollo y la documentación de pruebas funcionales. 	<p><u>Pruebas:</u></p> <ul style="list-style-type: none"> - No contempla el desarrollo y registración de pruebas no funcionales.
Diagrama NCL	<ul style="list-style-type: none"> - Ofrece visión estructural del programa. - Es mapeable fácilmente a código. 	<ul style="list-style-type: none"> - Carece de elementos para representar la visión espacio-temporal. - En aplicaciones mediana-grandes el gráfico es complejo y pierde claridad.
NCL	<ul style="list-style-type: none"> - Los prototipos desarrollados requirieron diversos elementos del lenguaje: media-region-descriptor-puerta-conector/link- transición. - Se han empleado medias de tipo imagen, texto y video. - Se implementó esquema de modularización de archivos mediante el mecanismo de "Importar base de conectores". 	<ul style="list-style-type: none"> - Los prototipos desarrollados no emplearon medias de tipo audio y scripts LUA y elementos conectores "onPause" y "onResume" - No fue posible usar elementos "transición" diferentes a "fade" y "barWipe". - No se implementó esquema de modularización de archivos mediante el mecanismo de contextos
Ginga.ar		<ul style="list-style-type: none"> - Al ser una adaptación de Ginga.br aun no finalizada, el estándar aún no está completo. Lo que deriva en que ciertos códigos tengan distintos comportamientos en ambos middleware, como es el caso del manejo de transiciones y o los diferentes tamaños de display.

Lecciones Aprendidas

- La ambigüedad e incompletitud de la PEyP se reflejó con claridad en las pruebas que revelaron coordenadas incorrectas y/o superposición de medias.
- La ambigüedad de la PEyP sumada a la falta de elementos descriptores de espacio-tiempo del diagrama NCL, pueden ocasionar que una misma especificación se codifique como diferentes documentos NCL.
- Las variables que hacen a las características espacio-temporales para este tipo de aplicaciones deben estar documentadas en la especificación y diseño.
- En cuanto al Diagrama NCL cuando la aplicación requiere múltiples medias con sus respectivos eventos/acciones se torna ilegible como se aprecia en la Fig. 9. y pierde significativamente utilidad para los desarrolladores.
- Durante la ejecución de las pruebas funcionales ocurrieron ciertos eventos que no fueron registrados en la PEyP, como ser: falta de sincronismo entre la presentación de las distintas medias, retardo en la reproducción de medias y eventos, distorsiones en la resolución de pantalla (o video).
- Algunos elementos NCL, por ejemplo de tipo transición, aún no funcionan en Ginga.ar por lo que no han sido probados.

- Ginga.ar aún no es totalmente compatible en cuanto a sus funciones a Ginga.br por lo que ciertos comportamientos que aparecen como “errores de código” desaparecen cuando se prueba en Ginga.br. Esta duplicación de pruebas del mismo código en los diferentes middleware conlleva más tiempo y esfuerzos.

6. TRABAJOS RELACIONADOS

El trabajo propuesto en [14] presenta un modelo para el desarrollo de programas de TVDI, basado en metodologías ágiles, que integra actividades inherentes al proceso de producción de TV y actividades de desarrollo de software. El modelo es pautado en cuatro aspectos: 1) Filosofía de trabajo, 2) Proceso, 3) Roles y Responsabilidades y 4) Artefactos. La definición de la filosofía de trabajo fundamenta el proceso de modelado e implementación. Este proceso describe fases y ciclos bien definidos con detalle de las actividades a realizar. La descripción del perfil de los recursos humanos involucrados (programador, ingeniero de prueba, consultor, guionista, director y etc.) así como la responsabilidad de cada uno de ellos dentro de las actividades definidas, evita distorsiones a lo largo de la ejecución del modelo y permite que todo el equipo construya de forma colaborativa un programa TVDI. El final de cada actividad da como resultado la producción de artefactos (documentos, códigos, medias y submodelos producidos durante el proceso) de los cuales se puede destacar: Storyboard [15], timeline, flujo de interactividad y esbozos de la interface a ser contemplados. El modelo divide el ciclo de vida del desarrollo de programas TVDI en 5 fases cortas, con Iteraciones constantes y con fuerte integración del usuario (telespectador) y el equipo de desarrollo. Siendo ellas: Concepción, Elaboración, Construcción, Prototipado/Prueba y Liberación.

Buscando contribuir a la solución del problema de modelado de aplicaciones TVDI en [16] se presenta un modelo denominado StoryToCode. Este modelo permite especificar programas TVDI basados en la utilización de componentes software. StoryToCode permite la transformación de un modelo conceptual de alto nivel en una descripción del conjunto abstracto de elementos componentes y posteriormente la transformación de esos elementos en código. En StoryToCode un componente software es tratado como un elemento especial que puede ser reutilizado en otros contextos (otras plataformas de ejecución como Web, móvil, etc.). StoryToCode está inspirado en MDA (Model Driven Architecture) y permite proyectar e implementar una aplicación, independiente del contexto, llevando en consideración las particularidades de un programa TVDI. El objetivo de StoryToCode es disminuir la responsabilidad del generador de contenido a través de la descentralización de las etapas de producción, que está fuera de su universo de trabajo original: la especificación y la implementación de un artefacto software. Se espera, como consecuencia de su utilización, permitir la participación de otros actores en el proceso productivo de un programa TVDI y disminuir el esfuerzo requerido durante ese proceso y el reaprovechamiento de los componentes de software generados en dominios diferentes.

7. CONCLUSIONES

El acercamiento al desarrollo de aplicaciones interactivas para TV Digital planteado ha implicado no sólo la codificación de programas. Se ha seguido el clásico esquema de desarrollo de software convencional e incremental, ya que ha sido posible establecer un proceso lógico, coherente y repetible. Se han empleado artefactos específicos para las distintas actividades: especificación, diseño, implementación y prueba. El proceso, las actividades y sus respectivas herramientas y artefactos han sido puestos a prueba, en el desarrollo de varios y diversos prototipos. Luego han sido analizados críticamente, desde su utilidad y practicidad. El análisis indica: el uso de cuadros de escenas para identificar requisitos de usuarios resulta ser una herramienta apropiada; la PEyP debe ser mejorada de manera tal que incluya información referente a los atributos espacio-temporales de los elementos medias/eventos/acciones; el diagrama NCL debe ser redefinido para que aporte una visión espacio-temporal y resulta útil en aplicaciones pequeñas y sencillas, pero debe ser reemplazado por otras herramientas cuando la aplicación es mediana y/o compleja. Respecto del código deben desarrollarse más prototipos que requieran funcionamiento aportado por los elementos aun no usados. La PEyP facilita el desarrollo de pruebas funcionales, que buscan la validación de la interfaz y el comportamiento de los elementos interactivos, sin embargo deben incorporarse mecanismos para realizar y documentar otros tipos de pruebas. El trabajo futuro se orienta a mejorar las deficiencias señaladas. Con este objetivo se analizarán enfoques de desarrollo de software ágiles y herramientas de diseño.

REFERENCIAS

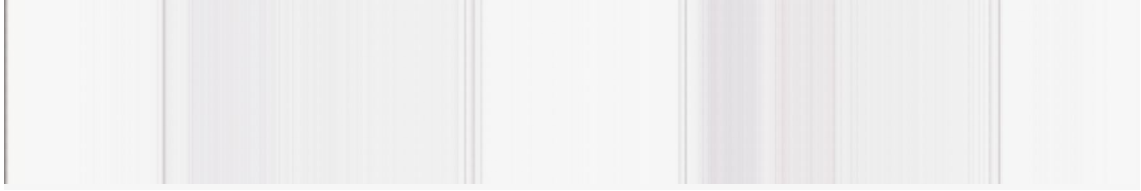
- [1] TV Digital Argentina http://www.tda.gob.ar/contenidos/tv_digital.html (2013)
- [2] Página Televisión Digital Terrestre en Latinoamérica <http://www.tdt-latinoamerica.tv> (2012)
- [3] Página de la Red AUTI (Red Temática en Aplicaciones y Usabilidad en Televisión Digital Interactiva) <http://readuti.net/info> (2013)
- [4] Página Oficial Ginga Brazil. <http://www.ginga.org.br>. (2013)
- [5] Página Oficial de NCL <http://www.ncl.org.br/> (2013)
- [6] Página Oficial Lenguaje LUA. <http://www.lua.org> (2013)
- [7] Página Oficial Ginga Argentina. <http://ginga.org.ar/> (2013)
- [8] Rodrigues R.F. y Soares R. F. “Producción de Contenido Declarativo para TV Digital”. En XXXIII *SemiSH*, Brasil, 2006.
- [9] Barbosa S.D.J. y Soares L.F.G., “TV digital interativa no Brasil se faz com Ginga: Fundamentos, Padroes, Autoria Declarativa e Usabilidade”. Em T. Kowaltowsky & K Breitman (orgs). *Atualizacoes em Informática 2008*. Rio de Janeiro, RJ. Editora PUC-Rio, 2008. pp 105-174.

- [10] Soares L.F.G, Rodrigues R.F. y Moreno M.F. “Ginga-ncl: the declarative environment of the brazilian digital TV system”. In Journal of the Brazilian Computer Society, v. 12, pp 37-46. SBC, 2007. <http://www.telemidia.puc-rio.br/en/node/323>
- [11] Comunidad Ginga Argentina. <http://comunidad.ginga.org.ar/>. (2013)
- [12] Soares L.F.G. y Rodrigues R.F. “Nested Context Model 3.0: Part 1 – NCM Core”, Technical Report, Departamento de Informática. PUC-Rio – Mayo 2005. ISSN:0103-9741
- [13] Página club NCL <http://club.ncl.org.br/> (2013)
- [14] Veiga E., y Tavares T., “Um modelo de Processo para o Desenvolvimento de Programas para TV Digital e Interativa”. WebMedia 2006 - USA.
- [15] Medrano C., “El Storyboard. Guiones para medios audiovisuales” - 2009. <http://elblogdecarlosmedrano.blogspot.com.ar/2012/01/storyboard.html>
- [16] Marques N. y Santos C., “StoryToCode: Um Modelo Baseado em Componentes para Especificação de Aplicações de TV Digital e Interativa Convergentes.” WebMedia 2009 - Brazil.

ANEXO A

	Elemento	Descripción
¿Dónde mostrar los medias?	Región	Representan el área de la pantalla donde se mostrará el media. Éstas pueden ser indicadas en pixeles o porcentaje. Son representadas mediante el elemento <code><region></code> . Ejemplo: // una región con un ancho y alto del 50% de la pantalla. <code><region id="reg1" height="50%" width="50%" zIndex="1" /></code>
¿Qué propiedad tiene el media?	Descriptor	Los descriptores especifican cómo los medias asociados a ellos serán exhibidos. Son representados mediante el elemento <code><descriptor></code> . Ejemplo: // un descriptor que proporciona una transparencia del 50% a un media. <code><descriptor id="idDesc" region="rgTVmenu"></code> <code><descriptorParam name="transparency" value="50%" /></code> <code></descriptor></code>
¿Quién comienza?	Puerta	Nos indica qué media iniciará al comenzar la aplicación. Son representados mediante el elemento <code><port></code> . Ejemplo: // el media "idImg" comenzara al iniciar la aplicación. <code><port id="pEntrada" component="idImg" /></code>
¿Cuándo mostrar el media?	Conector/Link	Permite especificar acciones sobre objetos media a partir de la ocurrencia de eventos. <u>Eventos:</u> - <i>onBegin</i> : Cuando la presentación de un media es iniciado. - <i>onEnd</i> : Cuando la presentación de un media es terminado. - <i>onAbort</i> : Cuando la presentación de un media es abortado. - <i>onPause</i> : Cuando la presentación de un media es pausado. - <i>onResume</i> : Cuando la presentación de un media es reanudado. - <i>onSelection</i> : Cuando una tecla relacionada al media es presionada. <u>Acciones:</u> - <i>start</i> : Inicia la presentación del media asociado. - <i>stop</i> : Termina la presentación del media asociado. - <i>abort</i> : Aborta la presentación del media asociado. - <i>pause</i> : Pausa la presentación del media asociado. - <i>resume</i> : Reanuda la presentación del media asociado. - <i>set</i> : Establece un valor de una propiedad asociada al media.

ANEXO B



ANEXO C

CODIGO PROTOTIPO I

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>

<!--..... TRANSITION .....-->
<!-- Especifica efectos de transicion/presentacion sobre un elemento/componente media -->

    <transitionBase>
      <transition id="transBoton" type="fade" dur="5s" />
<!-- Efecto de esfumado con duracion de 5 segundos -->
    </transitionBase>
<!-- que sera aplicado sobre un elemento media -->

<!--..... REGION .....-->
<!-- Determina un area de presentacion, en donde se mostraran los objetos media -->

    <regionBase>
      <region id="regFondo" width="100%" height="100%" />
      <region id="regVideo" width="693px" height="538px" left="13px"
        top="18px"/>
      <region id="regBoton" width="70px" height="70px" left="13px"
        bottom="20px"/>
    </regionBase>

<!--..... DESCRIPTOR .....-->
<!-- Especifica la informacion temporal y espacial necesaria para presentar c/ componente -->
<!-- Indica en que region se mostrara un objeto media y con que propiedades -->

    <descriptorBase>
      <descriptor id="descFondo" region="regFondo" />
      <descriptor id="descVideo" region="regVideo"/>
      <descriptor id="descBoton" region="regBoton" transIn="transBoton" />
    </descriptorBase>

<!--..... CONNECTOR .....-->
<!-- Determina sincronismo, por mecanismo de causalidad -->
<!-- Permite especificar acciones a traves de la ocurrencia de eventos (condiciones) -->

    <connectorBase>
      <causalConnector id="onSelectionSet">
        <connectorParam name="var"/>
        <connectorParam name="key"/>
        <simpleCondition key="$key" role="onSelection" />
        <simpleAction max="unbounded" role="set" value="$var" qualifier="par" />
      </causalConnector>

      <causalConnector id="onSelectionYellowSet">
        <connectorParam name="var"/>
        <simpleCondition key="YELLOW" role="onSelection" />
        <simpleAction max="unbounded" role="set" value="$var" qualifier="seq" />
      </causalConnector>

      <causalConnector id="onBeginStart">
        <simpleCondition role="onBegin" />
        <simpleAction max="unbounded" role="start" qualifier="seq"/>
      </causalConnector>
      <causalConnector id="onEndStop">
        <simpleCondition role="onEnd" />
        <simpleAction max="unbounded" role="stop" qualifier="par" delay="2s"/>
      </causalConnector>
    </connectorBase>
  </head>

  <body>

<!--..... MEDIA .....-->
<!-- Define un objeto de tipo multimedial, su tipo y localizacion -->

    <media id="video" descriptor="descVideo" src="media/video/video.avi">
      <area id="areaVideo" begin="0s" end="35s"/>
      <property name="bounds" />
    </media>
  </body>
</ncl>
```



```

</media>
<media id="fondo"
descriptor="descFondo"src="media/image/fondo_campus_pantalla_completa.jpg">
  <property name="src"/>
</media>
<media id="boton" descriptor="descBoton" src="media/image/botonAmarillo.gif">
  <property name="src"/>
</media>

<!--..... PORT .....-->
<!-- Son puertas de entrada a los contextos. Mapean la puerta a un elemento del contexto. -->
<!-- Un port que apunta a un objeto media indica que comenzara la presentacion del mismo, al
iniciar la aplicacion -->

  <port id="puerto1" component="fondo"/>
  <port id="puerto2" component="video"/>

<!--..... LINK .....-->
  <!-- Los conectores son templates que se instancian como links -->
  <!-- Un link mediante una referencia a un conector, permite asociar componentes a las
condiciones y a las acciones. Definiendo una relacion espacio-temporal. -->

  <link xconnector="onSelectionSet">
    <bind component="video" role="onSelection">
      <bindParam name="key" value="YELLOW" />
    </bind>
    <bind component="video" interface="bounds" role="set">
      <bindParam name="var" value="45.9%,11.5%,50.2%,39.6%"/>
    </bind>
    <bind component="fondo" interface="src" role="set">
      <bindParam name="var" value="media/image/videoMinimizado.jpg"/>
    </bind>
  </link>

  <link xconnector="onSelectionYellowSet">
    <bind component="video" role="onSelection"/>
    <bind component="boton" interface="src" role="set">
      <bindParam name="var" value="media/image/BotonVerde.gif"/>
    </bind>
  </link>

  <link xconnector="onSelectionSet">
    <bind component="video" role="onSelection">
      <bindParam name="key" value="GREEN" />
    </bind>
    <bind component="fondo" role="set" interface="src">
      <bindParam name="var" value="media/image/fondo_campus_pantalla_completa.jpg"/>
    </bind>
    <bind component="video" role="set" interface="bounds">
      <bindParam name="var" value="1.8%,3.2%,96.4%,94%"/>
    </bind>
    <bind component="boton" role="set" interface="src">
      <bindParam name="var" value="media/image/botonAmarillo.gif"/>
    </bind>
  </link>

  <link xconnector="onBeginStart">
    <bind component="video" role="onBegin"/>
    <bind component="boton" role="start"/>
  </link>

  <link xconnector="onEndStop">
    <bind component="video" interface="areaVideo" role="onEnd"/>
    <bind component="boton" role="stop"/>
    <bind component="video" role="stop"/>
    <bind component="fondo" role="stop"/>
  </link>

</body>
</ncl>

```

CODIGO PROTOTIPO II

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

```

<!-- Generated by NCL Eclipse -->
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
<head>
  <transitionBase>
    <transition id="transBoton" type="fade" dur="5s" />
  </transitionBase>
  <regionBase>
    <region id="regFondo" width="100%" height="100%" />
    <region id="regVideo" width="693px" height="538px" left="13px" top="18px"/>
    <region id="regBoton" width="70px" height="70px" left="13px" bottom="20px"/>
    <region id="regFrameBlue" width="140px" height="90px" left="10px" top="90px"/>
    <region id="regFrameRed" width="140px" height="90px" left="162px" top="90px"/>
    <region id="regFrameGreen" width="140px" height="90px" left="10px" top="190px"/>
    <region id="regFrameYellow" width="140px" height="90px" left="162px" top="190px"/>
  </regionBase>

  <descriptorBase>
    <descriptor id="descFondo" region="regFondo" />
    <descriptor id="descVideo" region="regVideo" focusIndex="0" moveDown="1"/>
    <descriptor id="descBoton" region="regBoton" transIn="transBoton" focusIndex="1" moveUp="0"
  />
    <descriptor id="descFrameBlue" region="regFrameBlue"/>
    <descriptor id="descFrameRed" region="regFrameRed"/>
    <descriptor id="descFrameGreen" region="regFrameGreen"/>
    <descriptor id="descFrameYellow" region="regFrameYellow"/>
  </descriptorBase>
  <connectorBase>
    <causalConnector id="onSelectionSet">
      <connectorParam name="var"/>
      <connectorParam name="key"/>
      <simpleCondition key="$key" role="onSelection" />
      <simpleAction max="unbounded" role="set" value="$var" qualifier="par" />
    </causalConnector>

    <causalConnector id="onSelectionYellowSet">
      <connectorParam name="var"/>
      <simpleCondition key="YELLOW" role="onSelection" />
      <simpleAction max="unbounded" role="set" value="$var" qualifier="seq" />
    </causalConnector>

    <causalConnector id="onBeginStart">
      <simpleCondition role="onBegin" />
      <simpleAction max="unbounded" role="start" qualifier="seq"/>
    </causalConnector>
    <causalConnector id="onEndStop">
      <simpleCondition role="onEnd" />
      <simpleAction max="unbounded" role="stop" qualifier="par" delay="2s"/>
    </causalConnector>
    <causalConnector id="onSelectionStart">
      <connectorParam name="var"/>
      <connectorParam name="key"/>
      <simpleCondition key="$key" role="onSelection" />
      <simpleAction max="unbounded" role="start" qualifier="par" />
    </causalConnector>

    <causalConnector id="onSelectionStop">
      <connectorParam name="var"/>
      <connectorParam name="key"/>
      <simpleCondition key="$key" role="onSelection" />
      <simpleAction max="unbounded" role="stop" qualifier="par" />
    </causalConnector>
  </connectorBase>
</head>
<body>
  <media id="video" descriptor="descVideo" src="media/video/video.avi">
    <area id="areaVideo" begin="0s" end="35s"/>
    <property name="bounds" />
  </media>

  <media id="fondo" descriptor="descFondo" src="media/image/fondo_campus_pantalla_completa.jpg">
    <property name="src" />
  </media>

  <media id="boton" descriptor="descBoton" src="media/image/botonAmarillo.gif">
    <property name="src" />
  </media>

```

```

    <media id="frameRed" descriptor="descFrameRed" src="media/image/Red_Frame.jpg">
  </media>
    <media id="frameBlue" descriptor="descFrameBlue" src="media/image/Blue_Frame.jpg"></media>
    <media id="frameGreen" descriptor="descFrameGreen"
src="media/image/Green_Frame.jpg"></media>
    <media id="frameYellow" descriptor="descFrameYellow"
src="media/image/Gold_Frame.jpg"></media>

    <port id="puerto1" component="fondo"/>
    <port id="puerto2" component="video"/>

    <link xconnector="onSelectionSet">
      <bind component="video" role="onSelection">
        <bindParam name="key" value="YELLOW" />
      </bind>
      <bind component="video" interface="bounds" role="set">
        <bindParam name="var" value="45.9%,11.5%,50.2%,39.6%"/>
      </bind>
      <bind component="fondo" interface="src" role="set">
        <bindParam name="var" value="media/image/videoMinimizado.jpg"/>
      </bind>
    </link>

    <link xconnector="onSelectionYellowSet">
      <bind component="video" role="onSelection"/>
      <bind component="boton" interface="src" role="set">
        <bindParam name="var" value="media/image/BotonVerde.gif"/>
      </bind>
    </link>

    <link xconnector="onSelectionSet">
      <bind component="video" role="onSelection">
        <bindParam name="key" value="GREEN" />
      </bind>
      <bind component="fondo" role="set" interface="src">
        <bindParam name="var" value="media/image/fondo_campus_pantalla_completa.jpg"/>
      </bind>
      <bind component="video" role="set" interface="bounds">
        <bindParam name="var" value="1.8%,3.2%,96.4%,94%"/>
      </bind>
      <bind component="boton" role="set" interface="src">
        <bindParam name="var" value="media/image/botonAmarillo.gif"/>
      </bind>
    </link>

    <link xconnector="onBeginStart">
      <bind component="video" role="onBegin"/>
      <bind component="boton" role="start"/>
    </link>

    <link xconnector="onEndStop">
      <bind component="video" interface="areaVideo" role="onEnd"/>
      <bind component="boton" role="stop"/>
      <bind component="video" role="stop"/>
      <bind component="fondo" role="stop"/>
      <bind component="frameRed" role="stop"/>
      <bind component="frameBlue" role="stop"/>
      <bind component="frameGreen" role="stop"/>
      <bind component="frameYellow" role="stop"/>
    </link>

    <link xconnector="onSelectionStart">
      <bind component="video" role="onSelection">
        <bindParam name="key" value="YELLOW" />
      </bind>
      <bind component="frameRed" role="start"/>
      <bind component="frameBlue" role="start"/>
      <bind component="frameGreen" role="start"/>
      <bind component="frameYellow" role="start"/>
    </link>

    <link xconnector="onSelectionStop">
      <bind component="video" role="onSelection">
        <bindParam name="key" value="GREEN" />
      </bind>

```

```

<bind component="frameRed" role="stop" />
<bind component="frameBlue" role="stop" />
<bind component="frameGreen" role="stop" />
<bind component="frameYellow" role="stop" />
</link>

</body>
</ncl>

```

CODIGO PROTOTIPO III

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <transitionBase>
      <transition id="transBoton" type="fade" dur="0s" />
    </transitionBase>

    <regionBase>
      <region id="regFondo" width="100%" height="100%" />
      <region id="regVideo" width="693px" height="538px" left="13px" top="18px" />
      <region id="regBoton" width="116px" height="36px" left="13px" bottom="20px" />
      <region id="regFrameBlue" width="140px" height="90px" left="10px" top="90px" />
      <region id="regFrameRed" width="140px" height="90px" left="162px" top="90px" />
      <region id="regFrameGreen" width="140px" height="90px" left="10px" top="190px" />
      <region id="regFrameYellow" width="140px" height="90px" left="162px" top="190px" />
      <region id="regImgDesc" width="690px" height="150px" left="8px" bottom="100px" />
      <region id="regTextDesc" width="780px" height="150px" left="16px" bottom="86px" />
    </regionBase>

    <descriptorBase>
      <descriptor id="descFondo" region="regFondo" />
      <descriptor id="descVideo" region="regVideo" focusIndex="0" moveDown="1" />
      <descriptor id="descBoton" region="regBoton" transIn="transBoton" focusIndex="1"
moveUp="0" />
      <descriptor id="descFrameBlue" region="regFrameBlue" />
      <descriptor id="descFrameRed" region="regFrameRed" />
      <descriptor id="descFrameGreen" region="regFrameGreen" />
      <descriptor id="descFrameYellow" region="regFrameYellow" />
      <descriptor id="descImgDesc" region="regImgDesc" />
      <descriptor id="descTextDesc" region="regTextDesc" />
    </descriptorBase>

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

  <body>
    <media id="video" descriptor="descVideo" src="media/video/video3.mp4">
      <area id="areaVideo" begin="0s" end="764s" />
      <property name="bounds" />
    </media>
    <media
      id="fondo"
      descriptor="descFondo"
src="media/image/fondo_campus_pantalla_completa.jpg">
      <property name="src" />
    </media>
    <media id="boton" descriptor="descBoton" src="media/image/menu.gif">
      <property name="src" />
    </media>
    <media id="frameRed" descriptor="descFrameRed" src="media/image/Red_Frame.jpg">
      <property name="bounds" />
    </media>
    <media id="frameBlue" descriptor="descFrameBlue" src="media/image/Blue_Frame.jpg">
      <property name="bounds" />
    </media>

```

```

<media id="frameGreen" descriptor="descFrameGreen" src="media/image/Green_Frame.jpg">
  <property name="bounds" />
</media>
<media id="frameYellow" descriptor="descFrameYellow" src="media/image/Gold_Frame.jpg">
  <property name="bounds" />
</media>
<media id="imgDesc" descriptor="descImgDesc" src="media/image/Fondo_escribir.gif"/>
<media id="textRed" descriptor="descTextDesc" src="media/files/textRed.txt">
  <property name="fontSize" value="20"/>
  <property name="fontColor" value="white"/>
</media>
<media id="textBlue" descriptor="descTextDesc" src="media/files/textBlue.txt">
  <property name="fontSize" value="20"/>
  <property name="fontColor" value="white"/>
</media>
<media id="textYellow" descriptor="descTextDesc" src="media/files/textYellow.txt">
  <property name="fontSize" value="20"/>
  <property name="fontColor" value="white"/>
</media>
<media id="textGreen" src="media/files/textGreen.txt" descriptor="descTextDesc">
  <property name="fontSize" value="20"/>
  <property name="fontColor" value="white"/>
</media>
<media id="boton2" descriptor="descBoton" src="media/image/menu.gif">
  <property name="src"/>
</media>

<port id="puerto1" component="fondo"/>
<port id="puerto2" component="video"/>

<link xconnector="conector#onBeginStart">
  <bind component="video" role="onBegin"/>
  <bind component="boton" role="start"/>
</link>

<link xconnector="conector#onEndStop">
  <bind component="video" interface="areaVideo" role="onEnd"/>
  <bind component="boton" role="stop"/>
  <bind component="video" role="stop"/>
  <bind component="fondo" role="stop"/>
  <bind component="frameRed" role="stop"/>
  <bind component="frameBlue" role="stop"/>
  <bind component="frameGreen" role="stop"/>
  <bind component="frameYellow" role="stop"/>
  <bind component="imgDesc" role="stop"/>
  <bind component="textGreen" role="stop"/>
  <bind component="textBlue" role="stop"/>
  <bind component="textRed" role="stop"/>
  <bind component="textYellow" role="stop"/>
  <bind component="boton2" role="stop"/>
</link>

<link xconnector="conector#onSelectionSet">
  <bind component="video" role="onSelection">
    <bindParam name="key" value="MENU" />
  </bind>
  <bind component="video" interface="bounds" role="set">
    <bindParam name="var" value="45.9%,11.5%,50.2%,39.6%"/>
  </bind>
  <bind component="fondo" interface="src" role="set">
    <bindParam name="var" value="media/image/videoMinimizado.jpg"/>
  </bind>
  <bind component="boton" interface="src" role="set">
    <bindParam name="var" value="media/image/atras.gif"/>
  </bind>
</link>

```

```

<link xconnector="conector#onSelectionSet2">
  <bind component="boton" role="onSelection"/>
  <bind component="fondo" role="set" interface="src">
    <bindParam name="var" value="media/image/fondo_campus_pantalla_completa.jpg"/>
  </bind>
  <bind component="video" role="set" interface="bounds">
    <bindParam name="var" value="1.8%,3.2%,96.4%,94%"/>
  </bind>
  <bind component="boton" role="set" interface="src">
    <bindParam name="var" value="media/image/menu.gif"/>
  </bind>
</link>

<link xconnector="conector#onSelectionSet2">
  <bind component="video" role="onSelection"/>
  <bind component="fondo" role="set" interface="src">
    <bindParam name="var" value="media/image/fondo_campus_pantalla_completa.jpg"/>
  </bind>
  <bind component="video" role="set" interface="bounds">
    <bindParam name="var" value="1.8%,3.2%,96.4%,94%"/>
  </bind>
  <bind component="boton" role="set" interface="src">
    <bindParam name="var" value="media/image/menu.gif"/>
  </bind>
</link>

<link xconnector="conector#onSelectionStartSet">
  <bind component="video" role="onSelection">
    <bindParam name="key" value="MENU" />
  </bind>
  <bind component="frameRed" role="start"/>
  <bind component="frameBlue" role="start"/>
  <bind component="frameGreen" role="start"/>
  <bind component="frameYellow" role="start"/>
  <bind component="frameRed" role="set" interface="bounds">
    <bindParam name="var" value="23%,16%,19%,15%"/>
  </bind>
  <bind component="frameBlue" role="set" interface="bounds">
    <bindParam name="var" value="1.5%,16%,19.5%,15.6%"/>
  </bind>
  <bind component="frameGreen" role="set" interface="bounds">
    <bindParam name="var" value="1.5%,32%,19.5%,15.6%"/>
  </bind>
  <bind component="frameYellow" role="set" interface="bounds">
    <bindParam name="var" value="23%,32%,19%,15.6%"/>
  </bind>
  <bind component="boton" role="start"/>
</link>

<link xconnector="conector#onSelectionStop2">
  <bind component="boton" role="onSelection"/>
  <bind component="frameRed" role="stop"/>
  <bind component="frameBlue" role="stop"/>
  <bind component="frameGreen" role="stop"/>
  <bind component="frameYellow" role="stop"/>
  <bind component="imgDesc" role="stop"/>
  <bind component="textGreen" role="stop"/>
  <bind component="textBlue" role="stop"/>
  <bind component="textRed" role="stop"/>
  <bind component="textYellow" role="stop"/>
</link>

<link xconnector="conector#onSelectionStop2">
  <bind component="video" role="onSelection"/>

```

```

<bind component="frameRed" role="stop"/>
<bind component="frameBlue" role="stop"/>
<bind component="frameGreen" role="stop"/>
<bind component="frameYellow" role="stop"/>
<bind component="imgDesc" role="stop"/>
<bind component="textGreen" role="stop"/>
<bind component="textBlue" role="stop"/>
<bind component="textRed" role="stop"/>
<bind component="textYellow" role="stop"/>
</link>

<link xconnector="conector#onSelectionSetStop">
  <bind component="frameGreen" role="onSelection">
    <bindParam name="key" value="GREEN" />
  </bind>
  <bind component="frameGreen" interface="bounds" role="set">
    <bindParam name="var" value="1%,12%,42.8%,39.6%"/>
  </bind>
  <bind component="boton" interface="src" role="set">
    <bindParam name="var" value="media/image/menu.gif"/>
  </bind>
  <bind component="frameRed" role="stop"/>
  <bind component="frameBlue" role="stop"/>
  <bind component="frameYellow" role="stop"/>
  <bind component="boton" role="stop"/>
</link>

<link xconnector="conector#onSelectionSetStop">
  <bind component="frameRed" role="onSelection">
    <bindParam name="key" value="RED" />
  </bind>
  <bind component="frameRed" interface="bounds" role="set">
    <bindParam name="var" value="1%,12%,42.8%,39.6%"/>
  </bind>
  <bind component="boton" interface="src" role="set">
    <bindParam name="var" value="media/image/menu.gif"/>
  </bind>
  <bind component="frameGreen" role="stop"/>
  <bind component="frameBlue" role="stop"/>
  <bind component="frameYellow" role="stop"/>
  <bind component="boton" role="stop"/>
</link>

<link xconnector="conector#onSelectionSetStop">
  <bind component="frameBlue" role="onSelection">
    <bindParam name="key" value="BLUE" />
  </bind>
  <bind component="frameBlue" interface="bounds" role="set">
    <bindParam name="var" value="1%,12%,42.8%,39.6%"/>
  </bind>
  <bind component="boton" interface="src" role="set">
    <bindParam name="var" value="media/image/menu.gif"/>
  </bind>
  <bind component="frameGreen" role="stop"/>
  <bind component="frameRed" role="stop"/>
  <bind component="frameYellow" role="stop"/>
  <bind component="boton" role="stop"/>
</link>

<link xconnector="conector#onSelectionSetStop">
  <bind component="frameYellow" role="onSelection">
    <bindParam name="key" value="YELLOW" />
  </bind>
  <bind component="frameYellow" interface="bounds" role="set">
    <bindParam name="var" value="1%,12%,42.8%,39.6%"/>
  </bind>

```

```

    <bind component="boton" interface="src" role="set">
      <bindParam name="var" value="media/image/menu.gif"/>
    </bind>
    <bind component="frameGreen" role="stop"/>
    <bind component="frameRed" role="stop"/>
    <bind component="frameBlue" role="stop"/>
    <bind component="boton" role="stop"/>
  </link>

  <link xconnector="conector#onSelectionStart">
    <bind component="frameYellow" role="onSelection">
      <bindParam name="key" value="YELLOW" />
    </bind>
    <bind component="imgDesc" role="start"/>
    <bind component="textYellow" role="start"/>
    <bind component="boton2" role="start"/>
  </link>

  <link xconnector="conector#onSelectionStart">
    <bind component="frameRed" role="onSelection">
      <bindParam name="key" value="RED" />
    </bind>
    <bind component="imgDesc" role="start"/>
    <bind component="textRed" role="start"/>
    <bind component="boton2" role="start"/>
  </link>

  <link xconnector="conector#onSelectionStart">
    <bind component="frameBlue" role="onSelection">
      <bindParam name="key" value="BLUE" />
    </bind>
    <bind component="imgDesc" role="start"/>
    <bind component="textBlue" role="start"/>
    <bind component="boton2" role="start"/>
  </link>

  <link xconnector="conector#onSelectionStart">
    <bind component="frameGreen" role="onSelection">
      <bindParam name="key" value="GREEN" />
    </bind>
    <bind component="imgDesc" role="start"/>
    <bind component="textGreen" role="start"/>
    <bind component="boton2" role="start"/>
  </link>

  <link xconnector="conector#onSelectionStop">
    <bind component="boton2" role="onSelection">
      <bindParam name="key" value="MENU" />
    </bind>
    <bind component="imgDesc" role="stop"/>
    <bind component="textGreen" role="stop"/>
    <bind component="textRed" role="stop"/>
    <bind component="textBlue" role="stop"/>
    <bind component="textYellow" role="stop"/>
    <bind component="boton2" role="stop"/>
  </link>
</body>
</ncl>

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="Conectores" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <connectorBase>
      <causalConnector id="onSelectionSet">
        <connectorParam name="var"/>
        <connectorParam name="key"/>

```



```

    <simpleCondition key="$key" role="onSelection" />
    <simpleAction max="unbounded" role="set" value="$var" qualifier="par" />
  </causalConnector>

<!-- Conector que especifica que según la condición de seleccionar un componente multimedia-->
  <!-- se ejecutara la acción de settear el valor definido por parámetro-->
  <causalConnector id="onSelectionSet2">
    <connectorParam name="var"/>
    <simpleCondition role="onSelection"/>
    <simpleAction max="unbounded" role="set" value="$var" qualifier="par" />
  </causalConnector>

  <causalConnector id="onSelectionSetStop">
    <connectorParam name="var"/>
    <connectorParam name="key"/>
    <simpleCondition key="$key" role="onSelection" />
    <compoundAction operator="seq">
      <simpleAction max="unbounded" role="set" value="$var" qualifier="par" />
      <simpleAction max="unbounded" role="stop" qualifier="par" />
    </compoundAction>
  </causalConnector>

  <causalConnector id="onBeginStart">
    <simpleCondition role="onBegin" />
    <simpleAction max="unbounded" role="start" qualifier="seq"/>
  </causalConnector>

  <causalConnector id="onEndStop">
    <simpleCondition role="onEnd" />
    <simpleAction max="unbounded" role="stop" qualifier="par" delay="2s"/>
  </causalConnector>

  <causalConnector id="onSelectionStart">
    <connectorParam name="var"/>
    <connectorParam name="key"/>
    <simpleCondition key="$key" role="onSelection" />
    <simpleAction max="unbounded" role="start" qualifier="par" />
  </causalConnector>

  <causalConnector id="onSelectionStartSet">
    <connectorParam name="var"/>
    <connectorParam name="key"/>
    <simpleCondition key="$key" role="onSelection" />
    <compoundAction operator="seq">
      <simpleAction max="unbounded" role="start" qualifier="par" />
      <simpleAction max="unbounded" role="set" value="$var" qualifier="par" />
    </compoundAction>
  </causalConnector>

  <causalConnector id="onSelectionStop">
    <connectorParam name="var"/>
    <connectorParam name="key"/>
    <simpleCondition key="$key" role="onSelection" />
    <simpleAction max="unbounded" role="stop" qualifier="par" />
  </causalConnector>

  <causalConnector id="onSelectionStop2">
    <connectorParam name="var"/>
    <simpleCondition role="onSelection" />
    <simpleAction max="unbounded" role="stop" qualifier="par" />
  </causalConnector>

</connectorBase>
</head>
</ncl>

```