

Un Prototipo de Desarrollo NCL para la Plataforma de Televisión Digital

RIBERI, FRANCO G.*

Resumen

En este trabajo se presenta una aplicación para la plataforma de televisión digital. Esta aplicación está integrada a un video educativo sobre Medicina Veterinaria, y cumple la función de proveer información adicional sobre el tema abordado en el video. Esta información se presenta opcionalmente (el usuario decide si visualizarla o no) de forma similar a una aplicación web, brindando al usuario la posibilidad de elegir a través de menús interactivos la información deseada.

La aplicación está desarrollada sobre la plataforma middleware GINGA, y combina características de los lenguajes Nested Context Language NCL (declarativo) y LUA (imperativo), para conseguir interactividad con el usuario. Esta interactividad incluye funcionalidades relativamente simples, como la provisión de diferentes tipos de información en los distintos momentos del video, y algunas más elaboradas, como la posibilidad de realizar actividades de autoevaluación (tests, preguntas y respuestas, etcétera). La aplicación puede ejecutarse/visualizarse a través de su transmisión por TV digital, o alternativamente como una aplicación *stand-alone*, provista en un live CD.

Palabras clave: TV Digital, NCL, LUA, GINGA.

1. Introducción

La televisión es un medio de comunicación que actualmente se encuentra en constante crecimiento, existe un cambio masivo que los diferentes países del mundo están atravesando, este cambio consiste en el salto de televisión analógica a la televisión digital.

El impacto de la TV digital es mucho más significativo que una simple mejora en la calidad de imagen y sonido. La TV Digital es un sistema permite mayor flexibilidad, facilitando expandir las funcionalidades del sistema mediante aplicaciones, en otras palabras, desarrollo de software para televisión. Tales aplicaciones son programas computacionales residentes en dispositivos receptores (conocidos como STB¹) o provenientes de datos enviados conjuntamente con el audio y video principal de un programa televisivo. La integración de una capacidad computacional a los STB permite el surgimiento de una amplia gama de servicios y programas de TV compuestos no sólo por el audio y video principal, sino también

*Departamento de Computación. Facultad de Ciencias Exactas, Físico-Químicas y Naturales. Universidad Nacional de Río Cuarto

¹Set-Top-Box

por otros videos, imágenes, textos, etcétera, sincronizados en una aplicación muchas veces guiada por la interacción del usuario telespectador. El universo de las aplicaciones puede ser particionado en un conjunto de aplicaciones declarativas, desarrolladas en el lenguaje de programación NCL, y un conjunto de aplicaciones imperativas desarrolladas en el lenguaje LUA, que pueden estar relacionadas entre sí.

Es necesario que las aplicaciones sean independientes de la plataforma de hardware y software, por lo cual se añade una nueva capa al modelo de referencia del sistema TV Digital, denominada GINGA, este es un middleware que permite ejecutar aplicaciones interactivas dentro de un STB. Dada la innovación que esta tecnología presenta, se optó por el desarrollo de una aplicación para esta plataforma. Tal aplicación tiene como objetivo la digitalización de un video particular, entendiéndose por digitalización, la agregación de interactividad mediante las herramientas provistas por TV Digital.

El video sobre el cual se realizó la aplicación lleva el nombre “*Punto por Punto: Suturas*”. Esta producción audiovisual fue provista por la Facultad Agronomía y Veterinaria, más precisamente por el área de Medicina Veterinaria de la Universidad Nacional de Río Cuarto (U.N.R.C.).

“*Punto por Punto: Suturas*” fue elaborado, diseñado e implementado en el marco de un Proyecto de Innovación e Investigación para el mejoramiento de la Enseñanza de Grado (PIMEG) Res Rec 1331/08. Secretaria Académica, Secretaria de Ciencia y Técnica y Secretaria de Planificación y Relaciones Institucionales de la UNRC.

En la sección 2 se describe el sistema de TV Digital en Argentina y algunos detalles técnicos. La sección 3 describe brevemente el middleware utilizado. En las secciones 4 y 5 se detallan algunas características de los lenguajes de programación ya mencionados. Por último, las secciones 6, 7 y 8 describen el producto desarrollado. Finalmente, se plantean algunos interrogantes, conclusiones de esta experiencia y trabajo futuro.

2. Sistema de Televisión Digital

Televisión Digital refiere al conjunto de tecnologías de transmisión de imagen y sonido a través de señales digitales. La representación digital de la señal (en bits) trae aparejada enumerables ventajas, tales como la reconstrucción de la señal cuando ésta es atenuada o experimenta leves perturbaciones, permite detectar y corregir errores que se utilizan cuando la señal llega al receptor, facilita para el procesamiento de la señal, mejora la calidad de imagen y sonido, entre otras.

En un sistema de televisión analógica el canal de transmisión sufre diversas formas de interferencias y ruidos en la señal original que limitan la capacidad del sistema. Estos ruidos no pueden ser evitados, provocando una reducción de la calidad de la señal. La disminución de la calidad depende principalmente de una relación señal/ruido (S/N). Cuando la relación disminuye, se reduce la calidad de señal.

En los sistemas digitales, a baja relación de S/N aumenta la probabilidad de error de bit, lo que significa una modificación en la señal. Para evitar este problema, todos los patrones de transmisión utilizan códigos de corrección de errores. Esto significa, si la tasa de error esta por debajo de un cierto umbral, el sistema de corrección de errores es capaz de corregir todas las fallas introducidas por el ruido en la señal lo cual conlleva que no haya pérdida de calidad en la imagen. Sin embargo, si la tasa de error

es superior al umbral mencionado, el sistema no será capaz de corregir el código transmitido. Éste es uno de los motivos por los cuales se acostumbra a decir que la televisión digital tiene una señal perfecta (“todo o nada”), sin “ruido” o ninguna señal.

La TV Digital permite mejor la calidad de imagen y sonido gracias a las técnicas de compresión de datos, lo que permite la producción de señales de alta resolución (HDTV).

El Sistema Argentino de TV terrestre define que cada canal tiene un ancho de banda de 6 Mhz. La televisión digital permite que en el mismo canal de 6 MHz, se transmita imagen de alta definición, con los siguientes parámetros: 30 cuadros por segundo a 1920 píxeles por línea, 1080 líneas activas entrelazadas en un televisor 16x9.

Las técnicas de compresión digital también permiten múltiples programas de más baja calidad en lugar de tener un sólo programa de alta calidad que ocupa toda la banda de 6 Mhz. Ésto es lo que se conoce como *multiprogramación*.

TV Digital proporciona un nivel de flexibilidad inalcanzable con la radiodifusión analógica. Un componente importante de esta flexibilidad es la posibilidad de ampliar las funciones del sistema para aplicaciones construidas sobre la base de un sistema de referencia estándar. Estas aplicaciones son programas informáticos que residen en un dispositivo de recepción o datos enviados conjuntamente con el audio y video principal de un programa televisivo. Así, una de las características más importantes de la TV Digital es la integración de cálculo en el dispositivo receptor, permitiendo el surgimiento de nuevos servicios.

En términos generales, el receptor procesa la señal digital recibida mediante la extracción de una colección de elementos del programa (video principal, audio principal, otros videos y audios, etcétera), llamados objetos multimedia que conforman el servicio seleccionado por el consumidor y que provienen de lo que se denomina **Carrusel de Datos** [1]. Esta selección se realiza mediante el servicio del sistema y la información, que también se transmiten conjuntamente con una solicitud que se encarga de la presentación final del programa.

El carrusel de datos, Figura 1, da soporte al envío cíclico de datos, dado que la sincronización de un canal televisivo puede ser realizado en cualquier momento, por lo cual los datos que no tengan relación temporal especificada por medio de etiquetas de tiempo deben ser enviados cíclicamente. De esta forma, la recepción de datos será independiente del instante de tiempo.

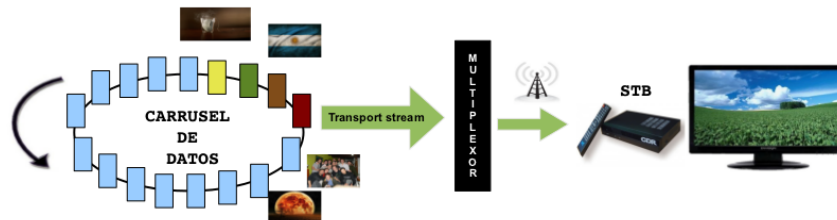


Figura 1: Carrusel de datos.

La capacidad de cálculo necesaria para el nuevo sistema puede ser integrada en diversos dispositivos de visualización modernos. Debido a

que hay un gran legado de aparatos de televisión analógica, una solución simple para ofrecer estos nuevos servicios receptores es utilizar junto a la televisión, un “sistema de procesamiento” capaz de manejar correctamente la señal de radiodifusión, que decodificará y mostrará la programación, aplicaciones y servicios avanzados. Este sistema de tratamiento consiste en una caja convertidora de la señal digital (o STB).

Un sistema de televisión digital es un típico sistema *Cliente/Servidor*. El entorno de servidor consiste en un organismo de radiodifusión (parte izquierda de la Figura 2) o un proveedor de contenido, y el cliente, que es el visor de entorno de usuario (parte derecha de la Figura 2).

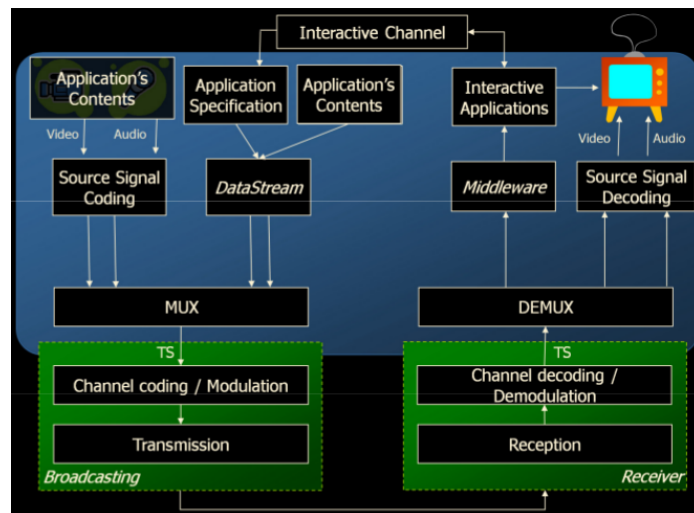


Figura 2: Sistema de TV Digital.

Desde otro punto de vista, un sistema de televisión digital consiste en un conjunto de normas que rigen cada uno de los procedimientos descritos en la Figura 2. La norma argentina de televisión digital es ISDB-Tb²[2], la cual se trata de la versión adaptada por Brasil que obliga que todo receptor sea capaz de recibir y ejecutar aplicaciones GINGA.

Otra perspectiva interesante de televisión digital es que puede ser utilizada con fines didácticos ya que permite videos interactivos en los cuales los usuarios telespectadores pueden desplazarse hacia diferentes perspectivas a través de sus decisiones e interacciones (primer nivel de interacción).

2.1. Aspectos Técnicos

2.1.1. Codificación de audio

El sistema americano de TV Digital utiliza el estándar de codificación AC3/ATSC [2], actualmente conocido como Dolby Digital (DD).

El sistema europeo de TV Digital utiliza el estándar MP2 (MPEG-1 Layer 2) [3], también adoptado por el sistema japonés. A diferencia de éstos, el sistema brasileño ha adoptado el estándar MPEG-4 [4] para la codificación de audio.

²International System for Digital Broadcast, Terrestrial, versión brasilera.

2.1.2. Codificación de video

El sistema americano, europeo y japonés de TV Digital terrestre adoptan MPEG-2 como su estándar para la codificación de vídeo.

El sistema brasileño utiliza una técnica de codificación más reciente: H.264 [5], también conocido como MPEG-4 Parte 10 o MPEG-4 AVC (Advanced Video Coding).

2.1.3. Sistema de transporte

Tanto el sistema americano, europeo, japonés y brasileño de TV digital terrestre cuando estandarizan la información audiovisual junto con los datos, deben ser multiplexados en una sola corriente. Todos los sistemas mencionados adoptan el mismo estándar de la multiplexación MPEG-2 [6], con pocas variaciones. MPEG-2 adiciona flujos elementales de información de audio y video principal para su exhibición sincronizada. La sincronización se realiza siguiendo el paradigma del eje de tiempo (time-line), añadiendo marcas de tiempo (timestamps). La generación y la multiplexación de los flujos de datos también están determinadas por dicha norma.

2.1.4. Modulación

El proceso de modulación es el responsable de recibir un flujo de transporte y colocarlo en un canal de frecuencia. Las técnicas de modulación digital se pueden dividir en dos grupos: uno con portadora única (sistema Chino) y otro con múltiples operadores, (sistema europeo, sistema japonés). El sistema brasileño utiliza la misma modulación del sistema japonés.

2.1.5. Canal de retorno o canal interactivo

Un sistema de TV digital puede funcionar sin un canal de retorno. En este caso, las aplicaciones pueden utilizar sólo los datos que se emitirán. Cuando las aplicaciones permitan la interacción real del usuario, el servicio ofrecido se denomina sitio interactivo.

La televisión digital pueden incluir el uso de un canal de retorno, el cual puede ser unidireccional, sólo permitiendo que el receptor envíe datos. Un segundo nivel de interactividad se define entonces, permitiendo al usuario telespectador el envío de datos (e.g., votación).

El canal de retorno también puede ser asimétrico bidireccional, permitiendo que el receptor pueda descargar los datos utilizados por las aplicaciones. En este caso, una aplicación puede recibir los datos transmitidos por la red o de vuelta. Una tercera parte del nivel de interactividad también está garantizada, permitiéndole a los usuarios telespectadores el acceso a los datos no provenientes de la emisora, por ejemplo, lo que permite navegar por la web.

Un canal de retorno bidireccional también puede permitir el envío de datos a través de banda ancha. Este nivel de interactividad, llamado interactividad plena, establece, entre otras cosas, lo que ha sido llamada “*Televisión Social*” o “*Televisión Comunitaria*”.

3. Middleware Ginga

El sistema de TV Digital utiliza un middleware que permite ejecutar aplicaciones interactivas dentro de un STB. (figura 3). Una aplicación puede realizarse directamente en el sistema operativo de un receptor. Sin embargo, los sistemas operativos en general no están preparados para dar un buen soporte a las aplicaciones de la televisión digital. Además, una aplicación de televisión debe ser capaz de funcionar en cualquier plataforma hardware y software. Para que que las aplicaciones sean independientes de plataforma de hardware y software, se añade una nueva capa a los puntos de referencia de un sistema TV digital denominada GINGA [10].

El middleware es uno de los componentes más importantes de un sistema de televisión digital porque en la práctica, es lo que rige las relaciones entre dos grandes industrias: las productoras de contenidos y las fabricantes de receptores. El middleware abierto GINGA se divide en dos subsistemas, que permiten el desarrollo de aplicaciones siguiendo dos paradigmas de programación diferentes. Estos dos subsistemas se llaman *Ginga-J* [8] (para aplicaciones procedurales Java) y *Ginga-NCL* [9] (para aplicaciones declarativas NCL).

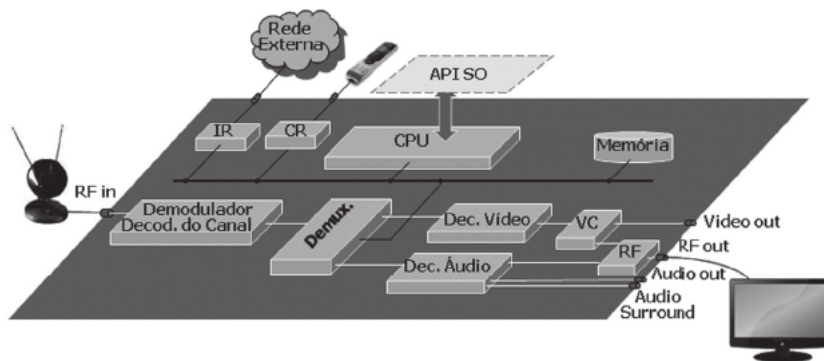


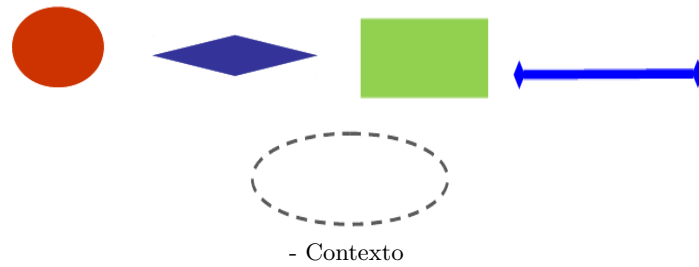
Figura 3: Estructura del Middleware.

4. Nested Context Lenguaje NCL

NCL [1] es un lenguaje declarativo, similar a un XML que separa los contenidos multimediales de la estructura de la aplicación. Define cómo los objetos media son estructurados y relacionados, tanto en tiempo como en espacio. El lenguaje está basado en un modelo conceptual de datos llamado Nested Context Model (NCM) [1]. Este modelo conceptual permite representar los conceptos estructurado de datos, así como los eventos y relaciones entre ellos. Además define las reglas y operaciones para la su manipulación.

Las entidades básicas que permite representar el lenguaje NCL son:

- Objetos Media
- Descriptor
- Región
- Link



Un *objeto media* puede ser un video, audio, texto, html, LUA, o una imagen. Un *descriptor* indica en que región se mostrará el objeto media y con que propiedades. Una *región* representa un área física de un dispositivo en la cual ciertos objetos media serán visualizados. Los *link* permiten especificar acciones sobre objetos media a partir de la ocurrencia de eventos (e.g., al presionar el botón rojo del control remoto, mostrar una imagen). Por último, un *contexto* permite agrupar elementos NCL y reutilizarlos.

Gráficamente puede observarse esta representación en la Figura 4.



Figura 4: Asociación entre media, descriptor y región.

4.1. Sintaxis del Lenguaje

La estructura general de un documento NCL es la siguiente:

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

5. Lua

LUA [1][7] es un lenguaje de programación imperativo, estructurado, bastante ligero y libre. Una explicación más detallada a cerca de la sintaxis y semántica de las sentencias LUA escapan a este trabajo, por lo cual se explicarán los conceptos del lenguaje necesarios para aplicaciones de TV Digital.

Para adecuar el ambiente de TV Digital e integrarse con NCL, el lenguaje LUA fue extendido con nuevas funcionalidades. Por ejemplo, un objeto NCLua precisa comunicarse con el documento NCL para saber cuando el objeto <media> correspondiente es iniciado por un link. Un NCLua también puede responder a teclas del control remoto. La diferencia entre un NCLua y un programa Lua puro es el hecho de ser controlado por un documento NCL.

Algunas bibliotecas proporcionadas por LUA para scripts NCLua son:

Módulo event: permite que objetos NCLua se comuniquen con el documento NCL y otras entidades externas (tales como control remoto y canal de interactividad).

Módulo canvas: ofrece funcionalidades para diseñar objetos gráficos en una región de NCLua.

Módulo settings: ofrece acceso a las variables definidas en el objeto settings de un documento NCL (objeto de tipo "application/x-ncl-settings").

Módulo persistent: exporta una tabla con variables persistentes entre ejecuciones de objetos imperativos.

Los mecanismos de integración de objetos NCLua con documentos NCL tienen sustento en el paradigma de programación orientada a eventos. (figura 5)

No sólo la comunicación con el NCL, sino también cualquier interacción con aplicaciones externas tales como canal interactivo, control remoto y el temporizador, tienen sustento en la difusión y recepción de eventos. El módulo *event* de NCLua utilizado para este propósito es la extensión más importante de Lua, y su comprensión es esencial para el desarrollo de cualquier aplicación que utiliza objetos NCLua.

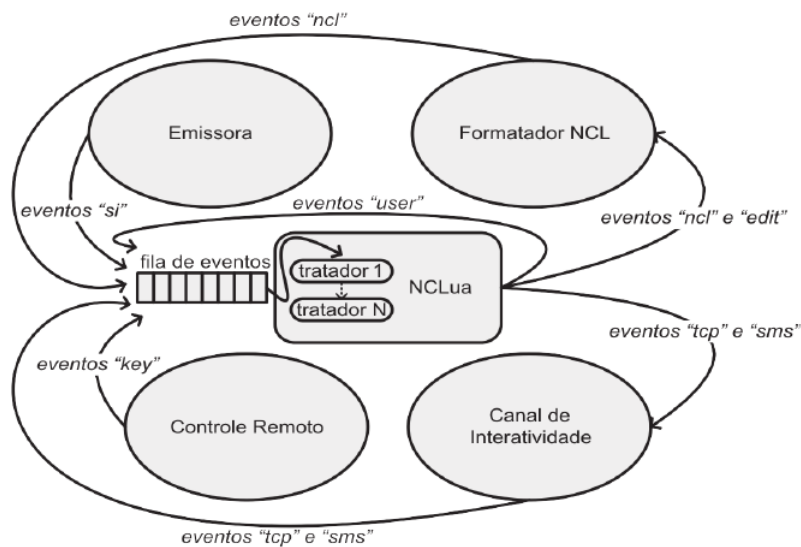


Figura 5: Programación orientada a eventos.

6. Producto desarrollado

6.1. Objetivo

Esta aplicación tiene por objetivo, proveer un prototipo de aplicación NCL-LUA que permita la digitalización de un video particular, entendiéndose por tal, la posibilidad de agregar interactividad mediante el uso de las herramientas que brinda la TV Digital.

Se plantea definir un modelo que respete las buenas prácticas de programación, y que permita tanto a los desarrolladores como a los telespectadores obtener parámetros sobre los alcances de la TV Digital.

6.2. Descripción

La aplicación permite al telespectador visualizar, con los 4 botones de interactividad GINGA, información adicional relevante al video-documental ordenada mediante un menú. Estos botones son un estándar en el control remoto del STB, los cuales poseen distintos colores: rojo, verde, amarillo y azul. El middleware GINGA en la PC, permite emular estos botones mediante las teclas F1, F2, F3 y F4 respectivamente. Retomando al menú, cada opción del mismo está relacionada a un color de estos botones.

El telespectador puede navegar a través de los menús que propone la aplicación utilizando estos botones de color y en algunos casos, cuando se lo solicite, el mismo deberá utilizar las flechas de navegación del control remoto. La aplicación provee además una funcionalidad de autoevaluación, la cual llamamos “*test points*”.

Es importante destacar que en ningún momento de la aplicación se pierde de vista el video-documental. Se trató de no quitar demasiado la atención del telespectador sobre el video por cuestiones pedagógicas.

Por último, la aplicación fue embebida en un *live-cd*, el cual lanza la aplicación GINGA de forma automática, dado que no se cuenta aún con los medios necesarios a nivel regional.

6.3. Diseño

El diseño de la aplicación fue desarrollado siguiendo la metodología proporcionada por el modelo conceptual de datos NCM.

Cada objeto media se asocia a una región a través de un descriptor. Por lo cual cada objeto a visualizar en la pantalla tiene asociado su correspondiente región y descriptor.

Como consecuencia de las dimensiones y complejidad en el modelado de toda la aplicación, se han adoptado algunas convenciones en el diseño. En la figura 6 se observa como se diseñó un conector, el cual es utilizado por los links para asociar una acción a una condición determinada.

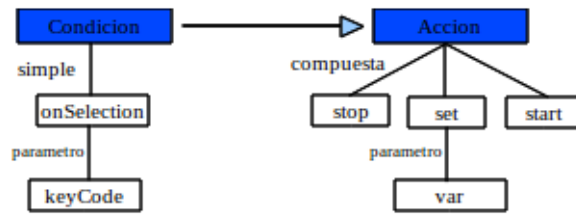


Figura 6: Conector onKeySeletionStopSetStart.

En la figura 7 se observa el diseño de un link.

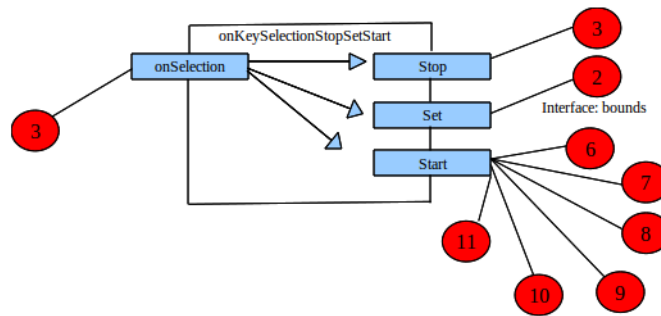


Figura 7: Ejemplo diseño de un link

Además de lo ya mencionado, se tuvieron en cuenta otros aspecto pedagógicos y audiovisuales, tales como por ejemplo redimensionar el video en el extremo superior derecho, no usar colores muy llamativos, resumir la información en cuadros o tablas en lugar de mostrar largos fragmentos de texto, no solapar más de 2 niveles de menú, entre otras.

6.4. Detalles de implementación

Para cumplir con el requisito funcional de añadir al video una opción de menú, al comenzar la aplicación no sólo se lanza el video principal con su correspondiente audio, sino también un media que representa el menú mencionado. Además se lanza un media “background” que cumple la función de fondo de aplicación, para evitar baches de imagen entre interacciones durante la aplicación, logrando de esta manera cierta consistencia estética. Asimismo se lanza un objeto “testLua”, el cual es necesario para realizar el envío de eventos necesarios hacia LUA, permitiendo así la exhibición de los diferentes tests dependiendo del capítulo que haya concluído.

Este lanzamiento inicial de objetos media se realiza mediante la definición de puertos, como se observa en el Código 1.

```

<port id="p0" component="Background"/>
<port id="p1" component="BotonInfo"/>
  
```

```
<port id="p2" component="videoPrincipal"/>
<port id="p3" component="audio"/>
<port id="p4" component="TestLua"/>
```

Código 1: Definición de puertos.

A continuación, se definieron todos los medias que forman parte de la aplicación (con su correspondiente descriptor y región, declarados previamente), es decir aquellos que en algún momento dado (ya sea de tiempo o por decisión del usuario telespectador) son exhibidos, inclusive aquellos mencionados en los ports. En la figura 8 se observa un ejemplo de como definir un media.

```
<media ... > se define a continuación
<descriptor id="dVideo" region="rgVideo"/>
<region id="rgVideo" width="100%" height="100%" zIndex="2"/>
```



```
<media id="BotonInfo" src=" ../RECURSOS/IMAGENES/info.png"
descriptor="dBotonInfo"/>
<descriptor id="dBotonInfo" region="rgBotonInfo"/>
<region id="rgBotonInfo" height="6%" width="6%" left="90%"
bottom="2%" zIndex="3"/>
```

Figura 8: Implementación de media.

Dentro de la declaración de los medias cabe destacar la definición del video principal, observado en el Código 2, dado que sobre éste recae la mayor parte de la aplicación. Esto hace referencia a la necesidad de su redimensión y la exhibición de los “test points”, cumpliendo de esta forma con los demás requerimientos funcionales mencionados.

Para la redimensión del video se define una propiedad “bounds” la cual es seteada con los valores dimensionales necesarios. Para la exhibición de los “test points” se especifican áreas dentro del video, las cuales poseen un tiempo de inicio y un tiempo de fin. Esto permite que cierto media se exhiba en un determinado tiempo y desaparezca en otro.

Se hace uso de las áreas para la exhibición de los test debido a que los diferentes capítulos se encuentra dentro de una unidad de video estáticamente y no se puede cambiar ese tiempo (*cabe destacar que no se pausa el video dado a que trae aparejado problemas con los instantes de tiempo de exhibición de los diferentes medias, además de problemas con el envío cíclico de los datos*).

```
<media id="videoPrincipal" src=" ../RECURSOS/VIDEO/video.avi"
descriptor="dVideoPrincipal">
  <property name="bounds"/>
  <area id="test1" begin="456s" end="486s"/>
  <area id="test1.2" begin="506s" end="536s"/>
  <area id="test2" begin="572s" end="602s"/>
  <area id="test3" begin="673s" end="703s"/>
  <area id="test4" begin="759s" end="789s"/>
  <area id="test5" begin="832s" end="850s"/>
  <area id="test6" begin="861s" end="891s"/>
```

```

<area id="test6.1" begin="931s" end="961s"/>
<area id="test7" begin="948s" end="978s"/>
<area id="test8" begin="1015s" end="1045s"/>
</media>

```

Código 2: Definición del media videoPrincipal.

Para el acceso a cada opción, se definieron *links*, teniendo soporte en los conectores, donde al seleccionar una opción deseada, el video principal se redimensiona y se exhibe la información correspondiente a dicha opción seleccionada.

Para los “*test points*”, fue necesario crear una imagen por cada test. El nombre de las imágenes se forma por la palabra “test” seguido de un índice “i”, el cual inicialmente es 1, y luego se incrementa según corresponda. Es decir, la imagen correspondiente al primer test llevará el nombre “test1”, y la imagen correspondiente al último test se llamará “testN”, donde N corresponde a la cantidad máxima de “*test points*”.

Así, en el archivo LUA se definen dos variables enteras: *testIndex* y *cantidadCorrectas*. La primera de ellas será utilizada para identificar que imagen se debe exhibir por pantalla dependiendo del número de capítulo. Esto es, la variable *testIndex* se inicializa en 0, cuando comienza un capítulo esta variable será incrementada en 1, por lo cual el test a exhibir será test1, y así sucesivamente hasta llegar al final de los capítulos. La segunda variable, *cantidadCorrectas*, será utilizada para llevar una cuenta de la cantidad de respuestas correctas. Al instante de responder un test, el usuario podrá ver si la respuesta es correcta o no, por lo que el archivo LUA también es el encargado de tomar la respuesta del usuario y mostrar por pantalla un tilde en caso de que la respuesta sea correcta, o una cruz en caso contrario.

Para realizar el envío de eventos desde el archivo NCL hacia LUA, fue necesario definir en el primero de ellos, cuatro propiedades sobre el media LUA, las cuales son: *ev*, *ev2*, *ev3*, *btn*.

La propiedad *ev* es seteada con el valor “nuevo” cuando comienza cada área del video principal (por ejemplo cuando comienza el área test1 a los 456 segundos, observar Código 2). Luego ese valor es utilizado por LUA en diferentes condiciones para determinar que evento ocurrió y realizar las acciones correspondientes.

La propiedad *ev2* es seteada por LUA para enviar eventos.

La propiedad *ev3* es seteada con el valor “showTest” en el momento de exhibir el resultado de los test, esto es, cuando el video principal termine y así lo disponga el usuario telespectador.

La propiedad *btn* es seteada con el valor “stop” cuando comienza cada área del video principal partiendo del área test2. Esto es usado para finalizar el test anterior. Por ello, se observó la necesidad de agregar un área fantasma al video, en la cual no se exhibe ningún test, pero es útil para frenar el último test real.

Además, la propiedad *btn* es usada para setear la respuesta del test. En este caso, si la respuesta corresponde a la opción 1, *btn* se setea con el valor “RED”, de lo contrario se setea con el valor “GREEN”.

Para la implementación del menú pinzas, se utilizó el elemento *switch*, ya que permite elegir que camino seguir según haya o no interacción del usuario.

6.5. Funcionalidad

Al comenzar la aplicación, se lanza junto al video principal un botón indicador de interactividad situado por estándar en el extremo inferior derecho de la pantalla, como se observa en la figura 9. Este botón es opcional, y permanece activo durante todo el video, a excepción de aquellos instantes de tiempos en que se brinda la posibilidad de realizar un test.

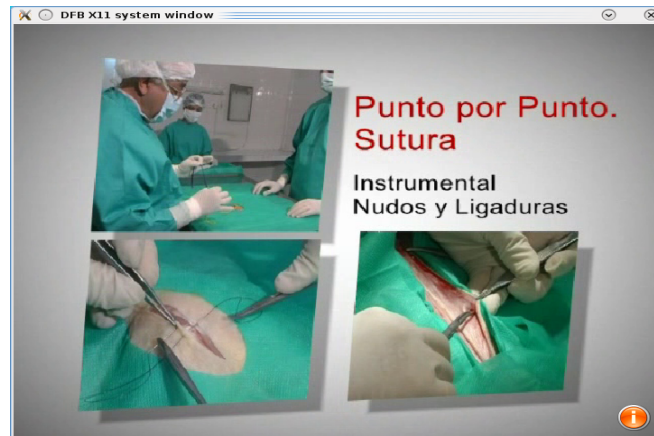


Figura 9: Inicio de la aplicación.

En todo momento, el telespectador tiene total control sobre lo que se exhibe en la pantalla, es decir, recae en él la decisión de visualizar la información extra que la aplicación brinda. En caso de que el usuario decida acceder a dicha información debe presionar el botón de menú del control remoto o en su defecto la tecla F5 del teclado dependiendo del dispositivo utilizado (televisor o computadora respectivamente). Efectuada esta acción, se desplegará un menú principal el cual administra las distintas opciones de información disponibles a través de los botones de color del control remoto y el video será redimensionado en el extremo superior derecho como se observa en la figura 10.

Se presentan 4 alternativas para el usuario: *sinopsis*, *instrumental*, *info anexa* y *volver*.

La opción *volver*, permite visualizar nuevamente el video en toda la pantalla como se observó en la figura 9. Las restantes opciones, añaden información en la pantalla sin pasar a otro estado, es decir, el video no se redimensiona, sino que la información se exhibe directamente en la parte inferior.

La opción *sinopsis*, presenta datos interesantes a cerca del video, datos de sus autores, contenidos, etcétera. Estos datos se visualizan en forma de texto. Para una perfecta apreciación del contenido, la información se exhibe por partes con un tamaño adecuado, donde el telespectador puede acceder a más información a través de las flechas de navegación como se observa en la figura 11.

La opción *instrumental*, permite el acceso a un menú secundario (figura 12), por lo cual, el menú principal es deshabilitado. La opción más interesante del menú secundario es *instrumental gral*, la cual despliega un nuevo menú desplegable, el cual consta de diversos instrumentales utilizados durante el video. Por cada instrumental, se exhibe una imagen y

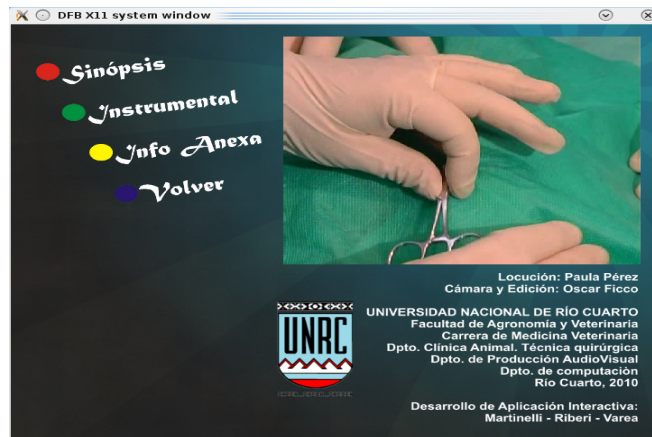


Figura 10: Acceso Menú Principal.

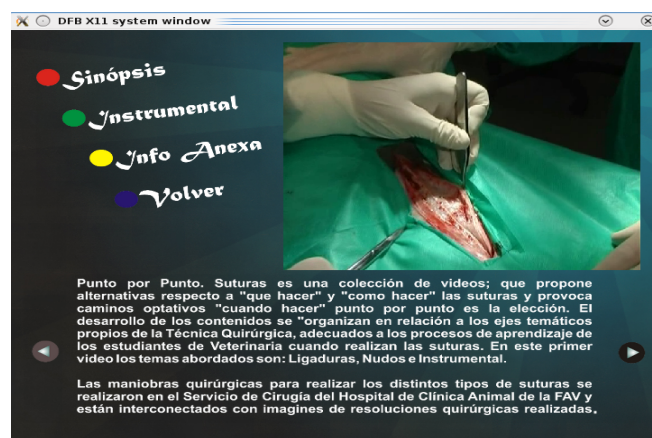


Figura 11: Opción sinopsis.

una pequeña descripción de la misma como se observa en la figura 13. El desplazamiento a través de este menú de instrumental se realiza mediante las flechas de navegación.

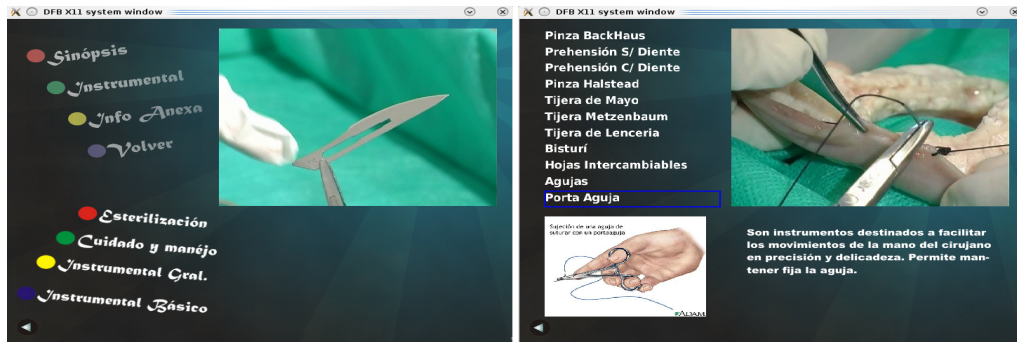


Figura 12: Menú Secundario.

Figura 13: Menú Instrumental

Las demás opciones no mencionadas hasta ahora, añaden información al video, ya sea en forma de cuadros como se muestra en la figura 14 o en forma de texto.

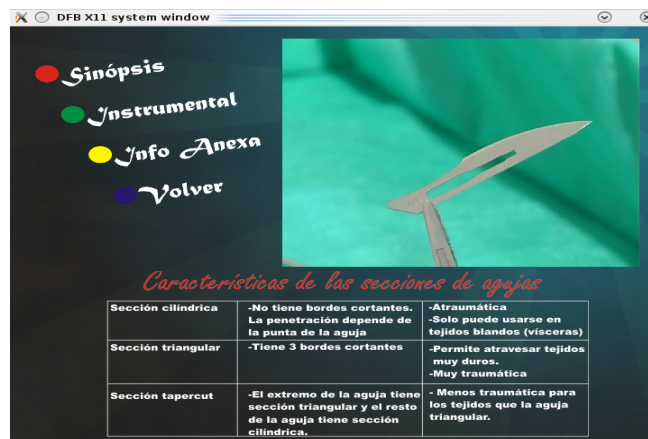


Figura 14: Opción Info Anexa.

Como se mencionó anteriormente, es posible la autoevaluación al finalizar cada capítulo del video. Para indicar que se encuentra disponible un "test point", se exhibe un botón celeste en el extremo inferior izquierdo de la pantalla (figura 15). Si el telespectador decide autoevaluarse deberá acceder a dicho test, en cuyo caso se exhiben diferentes preguntas multiple opción o un Verdadero/Falso referentes al capítulo transcurrido previamente (figura 16). Dicha pregunta puede ser contestada o ignorada por parte del usuario. En caso de que el usuario decida contestar el test, se le informará inmediatamente la veracidad de su respuesta (figuras 17 y 18) y al finalizar la aplicación se exhibirán los resultados de sus respuestas. Cada test tiene asociado un tiempo determinado, por lo cual, transcurrido el mismo, desaparece. Esta fue una decisión de implementación.



Figura 15: Opción test.

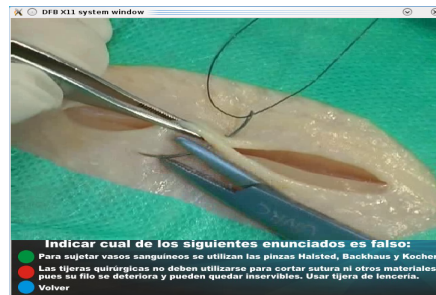


Figura 16: Acceso test.

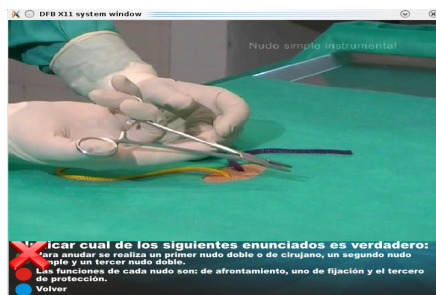


Figura 17: Respuesta incorrecta.

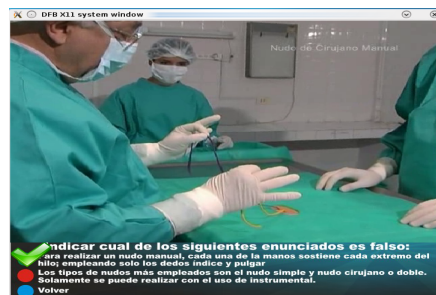


Figura 18: Respuesta Correcta.

7. Consideraciones durante la implementación

Durante el desarrollo de la aplicación surgieron diferentes conflictos, entre los cuales se pueden mencionar:

- El uso de muchos media de textos parece ser lo más lógico, dado a que son formatos más livianos en comparación con imágenes, pero resulta ser que la randarización de imágenes es mucho más eficiente que la de texto.
- En una primera instancia el menú principal de la aplicación se pensó usando focus (atributo del descriptor), es decir, declarando para cada opción un media texto. Al desplazarnos por las opciones se observó que el foco quedaba en los objetos que se atravesaban anteriormente, lo que se debe a un error en la implementación de la versión de GINGA utilizada. Dicho suceso se observa en la figura 19.
- En la implementación de los tests en primera instancia se realizó completamente en NCL, pero se observó que se necesitaban muchas líneas de código, e incluso era imposible no repetir código dado a que no es posible modularizar.
- Usar los colores del control remoto para realizar acciones (rojo, verde, amarillo y azul).
- Mantener el orden en el que aparecen los colores en el control remoto.
- Limitaciones técnicas de la plataforma:
 - Tamaño de píxel: monitores (píxeles cuadrados); pantallas de TV (píxeles rectangulares).



Figura 19: Menú con focus.

- Problemas en la pantalla de la TV: imágenes y gráficos estáticos y dinámicos.
- Colores: la pantalla de TV tiene una gama más limitada que los monitores.
- Tipos de archivos: se recomienda usar fuentes TTF, imágenes JPG o PNG, videos MP4 o AVI, sonidos MP3 principalmente.

8. Producto obtenido

Dado a que en la actualidad aún no se encuentra en funcionamiento la transmisión digital en nuestra región, se decidió embeber la aplicación final en un CD que permita su ejecución directa en una PC. Esto es, se generó un CD que contiene el middleware GINGA y la aplicación final, el cual permite bootear automáticamente, otorgándole al usuario la ventaja de no tener que mediar con instalaciones de por medio.

El live-cd arranca sobre un sistema de archivos (dentro de una carpeta root). Los scripts de inicio ejecutan un script /etc/rc.local el cual realiza un chroot a un sistema de archivos que contiene ginga instalado (y sus aplicaciones).

Desde este script (/etc/rc.local) se hace un chroot y se ejecuta la aplicación GINGA en ese nuevo ambiente. Al finalizar la aplicación y retornar del chroot, se realiza un reboot del sistema.

9. Conclusiones y Trabajo Futuro

Se ha presentado el desarrollo de una aplicación para la plataforma de TV Digital que combina el uso de dos lenguajes de programación basados en paradigmas diferente, por un lado, NCL para la descripción de la integración entre datos, video y sonido (y su respectiva sincronización), y por el otro, LUA para la representación de aspectos más algorítmicos.

La TV Digital nos lleva a proyectarnos no sólo desde lo que el control remoto nos permitirá sino de comenzar a tener en cuenta que es una herramienta que puede favorecer a la inclusión social en el campo de la educación presencial, semi-presencial y a distancia.

Dado el significativo impacto que la TV digital está teniendo en nuestro país, particularmente debido al alcance que la misma logrará en todo el territorio y las posibilidades que brindará para la difusión de contenidos, esta experiencia constituye un punto de partida para la posibilidad de futuros desarrollos basados en esta plataforma desde la UNRC. La rapidez con la que se espera tener el sistema de TV digital terrestre en completo funcionamiento nos impondrá considerables demandas en la generación de contenidos, y la adaptación o extensión de contenidos multimediales existentes con capacidades de interacción brindados por la plataforma. Este desarrollo ha dado lugar a la interacción con un equipo de trabajo en la Institución ya mencionada, vinculando diferentes ciencias, como la informática, la pedagogía y la medicina veterinaria.

Entre los trabajos futuros, se espera adaptar otros productos desarrollados por el departamento de Audiovisuales de la UNRC, con características similares a las mostradas en la aplicación presentada.

Referencias

- [1] Luis Fernando Gomes Soares, Simone Diniz Junqueira Barbosa. *“Programando Em NCL”*. Elsevier Editora Ltda. Rio de Janeiro. Brasil. ISBN 978-85-352-3457-2. 2009.
- [2] Advanced Television Systems Committee. *“Digital Audio Compression Standard (AC-3, E-AC-3)”*. Washington, D.C., junho.
- [3] International Organization for Standardization/International Electrotechnical Commission. *“Information Technology — Coding of Moving Pictures and Associated Digital Storage Media at up to About 1.5 Mbits/s, Part 3: Audio”*. ISO/IEC 11172-3.
- [4] Associação Brasileira de Normas Técnicas. *“Televisão digital terrestre — codificação de vídeo, áudio e multiplexação, Parte 2: Codificação de áudio”*. Sistema Brasileiro de TV Digital Terrestre, NBR 15602-2.
- [5] International Organization for Standardization/International Electrotechnical Commission. *“Information Technology — Coding of Audio-Visual Objects, Part 10: Advanced Video”*, ISO/IEC 14496-10.
- [6] International Organization for Standardization/International Electrotechnical Commission. *“Information Technology — Generic coding of moving pictures and associated audio information, Part 1: Systems”*. ISO/IEC 13818-1.
- [7] Ierusalimschy R. *“Programming in Lua”*. ASCII Media Works, Aug 2009, ISBN 978-4048677974.
- [8] Filho G.S., Leite L. E., Batista C.E. *“Ginga-J: The Procedural Middleware for the Brazilian Digital TV System”*.
- [9] Soares L. F. G., Rodrigues R.F., Moreno M.F. *“Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System”*.
- [10] Arturo Zambrano, Juan Antonio Zubimendi, Javier Búcar. *“An Alternative Architecture for Ginga”*. LIFIA, Facultad de Informática Universidad Nacional de La Plata.