

DISEÑO DE PUBLICACIONES CON INFORMACIÓN CAMBIANTE

Carlos Alberto Delgado Rivera



UNIVERSIDAD NACIONAL DE COLOMBIA

Facultad de Artes

Bogotá, abril de 2004

CONTENIDO

	página
1. La información es dinámica	3
2. Los nuevos medios	6
3. El escenario de la representación	9
4. Sistemas independientes	11
5. Tejiendo la red	13
6. Consulta a una base de datos	16
7. El soporte líquido	18
8. Orientando al navegante	20
9. El modelo en la práctica	22
Ejemplos 1 y 2: Actualización automática	23
Ejemplos 3 y 4: Listas y conjuntos	27
Ejemplo 5: Cambio de categorías	30
10. Bibliografía	32
11. Referencias en internet	33

1. LA INFORMACIÓN ES DINÁMICA

Me refiero al presente como "La era de la información", no solo por los avances en las tecnologías de comunicación, sino en un sentido más amplio en donde la información se ha constituido en un bien de consumo y en un instrumento de poder. El desarrollo de las tecnologías de comunicación no debe verse como la causa de este fenómeno contemporáneo ya que a su vez, es también su consecuencia, es la realización de una necesidad social. Los beneficios que se pueden derivar del control de la información son una presión muy fuerte que acelera los desarrollos tanto en aspectos tecnológicos como en el manejo y administración de la información. Tener acceso a información relevante puede significar una ventaja muy grande cuando se trata de tomar una decisión de inversión o definir una estrategia para un encuentro bélico o deportivo. Son las circunstancias mismas y los actores que participan en ellas los que definen si una cierta información es valiosa o no; el valor de la información depende de lo que podamos hacer con ella, de lo que signifique para nosotros, depende del momento y de las circunstancias. Hay momentos cuando lo más valioso es saber lo que nadie más sabe y momentos cuando lo importante es saber lo que todos saben para no estar aislado. La información se convierte en una necesidad cuando nos referimos a conocer donde encontrar alimento o como evitar un peligro; pero lo que llamamos *estar informado* o *saber lo que está pasando* es solo un deseo que nunca termina de ser satisfecho.

(¹). BARLOW, J. Perry. «Vender vino sin botellas» en *El Paseante*, No.27-28 'La revolución digital y sus dilemas', pp. 10-22.

La información no es un objeto material y por lo tanto no ocupa espacio, no está en ningún lugar. Es una experiencia, es una acción, está hecha de lo que están hechos los recuerdos: de palabras y de imágenes mentales. La información puede dejar huella, puedo escribir palabras, hacer dibujos, grabar sonidos o tomar fotografías para construir objetos que permitan trasladar la información a otro momento o lugar. Pero la información no requiere ser materializada para poderse transmitir, igualmente puedo convertir sonidos, imágenes y palabras en ondas electromagnéticas que viajan por el aire a la velocidad de la luz o que atraviesan las paredes pues no tienen cuerpo físico. La información se difunde y propaga, no se distribuye como los objetos materiales; aunque se transfiera, el emisor no deja de poseerla. Cuando yo le indico a otra persona donde queda la entrada, no por eso dejo de saberlo; no es como cuando entrego la llave de la puerta, dejo entonces de tenerla y no puedo volver a entrar. La información repercute en quien la recibe, a medida que la experimenta cada cual interpreta la misma información de manera diferente y a su vez, la transforma en el proceso.

La información sobre asuntos cambiantes cambia con ellos, la noticia de actualidad que consulto por la mañana no es la misma que aparece por la tarde, la actualidad es un concepto cambiante que se refleja en la información que la representa. Si coloco una cámara de video frente a alguien que está hablando y transmito la imagen, aparece en la pantalla del receptor una imagen vinculada con la escena que capta el encuadre de la cámara, cuando cambia la escena cambia consecuentemente la imagen transmitida. Ocurre lo mismo si la información que se presenta son cotizaciones de bolsa, estados atmosféricos o resultados deportivos; la información es un concepto dinámico, cambia al ritmo del tema tratado, a veces cambia mucho y muy frecuentemente y a veces permanece mucho tiempo invariable, pero para mantenerse significativa y actual necesariamente cambia. El cambio es una característica propia de la información, es consecuencia de existir en una dimensión temporal, el cambio ocurre como parte del proceso de comunicación.

La misma información puede existir de distintas maneras simultáneamente, cuando tengo una misma fotografía en varias resoluciones diferentes no tengo información diferente, tengo más o menos densidad de la misma información. Una noticia puede aparecer como un párrafo corto de texto o como un hipertexto con fotografías y enlaces con otra cantidad de textos sobre el mismo tema; ambas opciones pueden estar presentes simultáneamente sin que una descarte a la otra, son diferentes niveles de profundidad de una misma información. Solo el usuario define si un enlace está saltando a otro tema o si está explorando

en profundidad. El sentido del vínculo lo define el usuario, se le presentan alternativas pero es él quien finalmente hace la asociación de ideas entre una información y otra. Buen ejemplo son los almacenes que ofrecen una base de datos de sus productos organizada por sesiones como *electrodomésticos* o *ropa para dama*, mientras el usuario atraviesa con un buscador el ítem *cámaras digitales* comparando la oferta del mismo producto en las bases de datos de muchos almacenes diferentes, o el usuario que consulta información sobre un mismo acontecimiento en diferentes portales de noticias para obtener una visión más completa del mismo.

Durante muchos años hemos tratado de hacer perdurar la información más allá de la fragilidad de nuestra memoria escribiendo libros, narrando nuestra historia, almacenando imágenes y palabras. La red informática nos propone ahora trazar las relaciones que establecemos entre los libros, hacer evidentes los múltiples vínculos entrecruzados que establece nuestra mente entre las palabras y de las palabras con imágenes y sonidos. Se parece mucho más a la manera como guardamos los recuerdos en la memoria, como pequeños fragmentos de información interrelacionada.

La información en la red no es texto escrito, no es la huella que deja la información al pasar sino la información misma que fluye en presente, el texto escrito es reemplazado por el hipertexto, cada texto queda convertido en una sola de las opciones entre una multiplicidad de alternativas. Podemos pensar que si cada texto nos puede remitir a otro, existe entonces un solo gran hipertexto que los contiene a todos. Parecería que hemos alcanzado el mito de la enciclopedia que contiene todo el saber humano, es sólo que esta gran enciclopedia es un laberinto infinito en continua transformación.

Las imágenes representadas en la pantalla son producidas por la información que fluye por la red. Estas imágenes responden a la acción de leer secuencias numéricas. Son números que representan información de acuerdo a unas condiciones variables en cuanto a la configuración del equipo, la aplicación con que se haga la lectura y el tipo de conexión con que se cuente. Un sitio en la red, o como se le conoce más comúnmente una *página de internet*, existe dentro de este entorno digital. Un entorno definido por parámetros de usuario y por la dinámica de la información que se presenta. Estamos frente a un objeto virtual que se construye al momento de ser consultado, un objeto definido por secuencias numéricas que representan información y por alternativas de posibles relaciones de continuidad con otros objetos virtuales de características similares.

2. LOS NUEVOS MEDIOS

Si reconocemos a internet como un nuevo medio de comunicación, debemos entonces analizar cuales son las características propias que lo definen como tal y que implicaciones conlleva. Debemos identificar y analizar las características propias de los medios interactivos para poder desarrollar nuevos esquemas en los procesos de comunicación y diseño. Un buen punto de partida para comprender las particularidades de la red informática lo brinda Lev Manovich en su libro "The Language of New Media"¹ cuando identifica cinco principios básicos que marcan la diferencia entre los medios digitales y los medios tradicionales o analógicos:

En primer lugar tenemos el principio que dice que los objetos que componen la red informática pueden ser definidos utilizando lenguaje numérico. **La representación numérica** de por si es sólo un formato de codificación pero significa que todo objeto virtual posee la capacidad de ser transformado algorítmicamente. Trabajar con objetos definidos matemáticamente nos permite procesarlos, los números pueden ser comparados y ordenados, pueden ser objeto de transformaciones programadas. Puedo crear rutinas de procesamiento para aclarar una foto, subir el tono de un sonido o traducir un texto a otro idioma.

En segundo término tenemos **la modularidad**. Los objetos digitales tienen identidad propia y a su vez están compuestos de otros objetos que también mantienen una identidad propia. Una página WEB está compuesta de textos, imágenes, sonidos y animaciones que pueden ser editados de manera independiente ya que la página únicamente los vincula y no los incorpora. El modelo se repite a otras escalas en ambos sentidos, un portal está compuesto de páginas que pueden ser editadas de manera independiente y un texto a su vez también está compuesto de caracteres que pueden ser editados de manera independiente, esta estructura fractal es inherente a la red, es lo que la define y determina.

El tercer principio que nombra Manovich es **la automatización**. Es la consecuencia directa de la capacidad de programar el procesamiento de la información. Decir que algo se hace automáticamente es otra manera de decir que el computador "sabe lo que tiene que hacer" o que está programado para hacer algo cuando se presenten unas condiciones definidas con anterioridad.

(?). MANOVICH, Lev . " The Language of New Media", Cambridge, ms. MIT Press, 2000

El cuarto principio es **la variabilidad**. "Un objeto en los nuevos medios no es algo fijo para siempre sino algo que potencialmente puede existir en un número infinito de versiones diferentes". La misma información se puede presentar de maneras muy diferentes, la misma fotografía puede existir en diferentes resoluciones, la misma base de datos alimenta de contenido diferente a una misma página según se definan unas condiciones diferentes por parte de quien la consulta. La información se transforma al ser consultada según parámetros de usuario..

Por último al quinto principio lo llama **transcodificación**. Se refiere a que el computador trabaja con una estructura de datos y códigos que no se corresponde con categorías como son los géneros literarios sino que le es propia a la lógica del computador. Términos como empaquetar, apilar, encadenar o ramificar la información no tienen sentido en nuestra comunicación, pero son procesos usados por el computador al momento de procesar la información. Esta manera de estructurar la información se refleja necesariamente en la comunicación entre computadores y entre nosotros y el computador.

La posibilidad que tenemos de procesar la información y de programar los procesos se constituye en la principal característica que diferencia una red informática de los demás medios de comunicación. Cuando utilizamos el computador como medio de comunicación es muy importante tener en cuenta las posibilidades que nos ofrece como procesador de información y no sólo como transmisor de esta. Esta característica nos remite el principio de variabilidad, la información se transforma al ser procesada y transmitida además de la transformación propia que ocurre con el paso del tiempo; la transformación es algo inherente a la información, es parte de ella.

El principio de la modularidad es el modelo estructural que nos va a determinar la manera como se organiza la información en la red. La posibilidad de vincular o relacionar un objeto con otro (texto, imagen, sonido, página, película, etc.) nos ayuda a definir los tres elementos básicos para construir una red: **usuarios, objetos y vínculos**. Los usuarios son todos los posibles sujetos que pueden actuar en el sistema. Los objetos contienen la información y se relacionan unos con otros a través de vínculos. Los vínculos que relacionan los objetos no son puentes entre objetos materiales sino relaciones de continuidad entre cadenas dinámicas de información. Los vínculos son como el intercambiador de autopista que redirecciona continuamente sin detener el flujo.

El diseñador de una publicación en la red solo propone posibilidades, es el usuario quien va anudando estas cadenas, unas a continuación de otras. El usuario es quien define el flujo, es quien construye el hipertexto. A medida que se transforma la imagen que se despliega sobre la pantalla se construye una narración, un relato, una secuencia cargada de significado. El usuario completa la narración cuando navega el documento, cuando selecciona entre posibles alternativas y le da sentido a la secuencia. En un entorno de red la narración lineal da paso a una narración ramificada, a una narración personalizada donde cada usuario construye su propio relato. No solo con las opciones que le presenta el documento, sino que salta de un documento a otro según sus propios intereses. Navegar es interactuar con la información, es un concepto dinámico, es una acción en el tiempo. Navegar es redireccionar continuamente una línea de texto codificado, es una acción sobre un flujo dinámico.

Si además de navegar podemos intervenir sobre la información, la lectura se convierte en una conversación. La bidireccionalidad de la comunicación convierte al texto escrito en una conversación en presente. Cuando conversamos, oímos y hablamos; las palabras discurren en un continuo ir y venir con nuestro interlocutor. El hipertexto es una construcción colectiva; el computador conectado en red permite la construcción de una narración común, un cuento que contamos entre todos para poder actuar dentro de él. Una vez acordado un lenguaje común cada cual construye los objetos y los vínculos que desee; el laberinto lo construimos entre todos.

En el diseño de publicaciones tradicionales existen géneros como enciclopedia, revista, catálogo o ficha técnica y en los medios audiovisuales otros como noticiero, entrevista, videoclip o telenovela. Son formatos de narración o representación de información, son modos de relatar, modos que se construyen con la apropiación que hacen las personas del medio. Un género es un imaginario colectivo que se construye con el uso, es un formato editorial que debe ser aceptado y asimilado por los usuarios para ser reconocido como tal. De igual manera tenemos géneros como el *portal de noticias*, el *chat* o el *administrador de correo*. Son los nuevos géneros que están aportando los nuevos medios. Los géneros propios de la red informática son consecuencia de la capacidad de procesar e interactuar con la información, permiten comunicación en doble vía y mantienen una actualización constante de la información. El desarrollo de los géneros editoriales para la red está apenas en sus inicios, no constituyen todavía una verdadera garantía de aceptación sobre un modo de decir las cosas, es muy difícil predecir cuales serán los que se consoliden después de 10 o 20 años.

3. EL ESCENARIO DE LA REPRESENTACIÓN

El diccionario define **Interfaz** como: "*Superficie de contacto. Conexión funcional entre dos sistemas independientes*". Es particularmente interesante la definición de interfaz como una superficie de contacto entre un sujeto y una cantidad de información estructurada como sistema. La imagen proyectada en la pantalla es a la vez superficie de contacto y conector funcional. La terminación "faz" se refiere a una cara o fachada que se le pone por delante a la información para presentarla ante los demás, es una representación o puesta en escena de la información. Un sistema de representación y una base de datos son sistemas complejos independientes que establecen una relación funcional a través de esta imagen en la pantalla. Un sistema de representación lo constituyen las reglas del juego que definimos para presentar la base de datos.

El escenario es la imagen o conjunto de imágenes que identifican al sistema o subsistema de información. La representación de las acciones ocurre en un escenario; la interacción es algo muy teatral, es la representación de un libreto en un escenario. El vano o vacío que marca la boca del escenario es el mejor referente que tenemos para entender lo que ocurre en el plano de la pantalla. Se nos presentan actores interpretando roles, frente a fondos y piezas de utilería colocados como referentes de otros espacios. Nada de esto está en la pantalla, la imagen de la pantalla siempre representa, los elementos y los fondos son solo recursos narrativos que sirven de escenario a la acción que se narra. El escenario no se narra, solo se describe, la acción es la que lleva el hilo de la narración. En un escenario virtual sin embargo, cualquier elemento del escenario puede asumir un rol de actor e iniciar una acción. No es como en el teatro clásico, en la pantalla del computador ser actor o ser un objeto de utilería son solo estados de una misma entidad, son roles que se asumen de acuerdo a como se den las acciones.

Diseñar una interfaz es establecer un marco de referencia para unas acciones, es proponer el escenario donde se puedan dar las acciones. Las reglas básicas de esta representación las propuso Steve Jobs en los ochentas con el sistema operativo del computador Apple. Conocida como GUI (Graphic User Interface), sentó las bases de la comunicación con la imagen de la pantalla. Una vez se definieron conceptos como cursor, ventana, menú desplegable, seleccionar, arrastrar, click y dobleclick, se generalizó de manera sorprendente este modo de actuar en el escenario. A partir de entonces casi todas las interfaces que se diseñan, siguen con sus variantes naturales, estos mismos modos.

De alguna manera el Apple logró agrupar unos modos de fragmentar y clasificar la información que ya estaban en el diario, la enciclopedia y otros medios impresos y los incorporó a su sistema operativo. El fragmentar la información y presentarla dentro de marcos que permiten simultaneidad con otro tipo de información es ya una tradición del siglo XIX. También era ya utilizado el recurso de terminar un artículo con una sugerencia de que continua o se puede ampliar el contenido en otra página. La metáfora del escritorio, el cajón y la carpeta, obedece a una manera lógica de organizar y trabajar con información. Podrían existir otras maneras igualmente adecuadas pero difícilmente se puede competir contra el imaginario del modo de actuar característico del hombre frente a texto escrito sobre una hoja de papel.

El mundo virtual es el mundo de la representación, lo imaginamos como una ventana abierta al infinito de la fantasía. Pero no es tan profundo como parece, en verdad es solo un escenario y no llega sino hasta el telón de fondo. Se construye por medio de la adición de capas donde se colocan objetos virtuales. La última de atrás es el fondo, a todas las demás las percibimos como telones por delante del fondo. A medida que se superponen, las capas más cercanas tapan a las de más atrás. Podemos contar con efectos de transparencia pero el fondo nunca puede ser transparente. La representación no admite tener una imagen detrás del fondo del escenario, cuando trata de ocurrir, automáticamente se convierte en el fondo de la representación ya que siempre situamos el fondo como la imagen de más atrás. El movimiento como tal no existe, es solo la ilusión de movimiento que se produce por la sucesión de imágenes estáticas que se reemplazan a un ritmo de 30 imágenes por segundo como en el cine.

Desde la antigüedad se han utilizado rollos, paredes, tablas y todo tipo de soportes para el texto y la imagen. Pero desde que utilizamos la hoja de papel, desarrollamos una manera particular de fragmentar la información en unidades de representación. Una página es una unidad de información, unidad de cantidad en muchos casos, pero principalmente unidad de significado. Una agenda nos presenta un mes, una semana o un día por página sin confundirnos, mientras el modo de fragmentar sea lógico, no importa el tamaño de los fragmentos. Una portada, un índice, una introducción o un inicio de capítulo pueden ocupar más de una página en un libro; son unidades de información con características propias bien definidas que ocupan una o varias hojas pero con un mismo diseño de página. Numerar las páginas les asigna una identidad que las diferencia de todas las demás en el caso del libro impreso, pero en la web no hace sentido identificar los fragmentos por números; de hecho ya son números.

La unidad que confiere una *toma* en el lenguaje cinematográfico se combina con el concepto *página* y se produce lo que conocemos como un *sitio*. A un sitio lo define un escenario, solo aceptamos unos pocos cambios en el escenario hasta cuando lo consideramos otro escenario. Cambiar de escenario es pasar de un sitio a otro, es la acción de navegar una red informática. Un conjunto de escenarios que comparten una identidad gráfica son también un sitio. Un nodo en la red es también una red. Todo entra dentro del principio de modularidad; un objeto es un conjunto formado por objetos, que a su vez son conjuntos, y etc. Un sitio es un conjunto de objetos que al desplegarse definen un escenario. Pero un sitio también es un conjunto de escenarios construido a partir de variaciones de los mismos objetos.

4. SISTEMAS INDEPENDIENTES

La manera tradicional de diseñar publicaciones empieza por definir un formato de papel, una retícula y una fuente o conjunto de fuentes. Cada página aunque se diseña como parte de un sistema de representación, se construye como un objeto único. El contenido que se desea presentar es determinante al momento de tomar decisiones de forma. Si deseamos tener más de una versión de la publicación (la misma información en varios idiomas, versión para el profesor diferente a la versión para el estudiante o un catálogo de productos con contenidos diferentes dirigidos a sectores específicos de mercado) debemos repetir casi todo el trabajo de diseño para cada caso. Actualizar el contenido de una enciclopedia puede convertirse en un trabajo tan complejo como hacerla de nuevo.

Los avances en el desarrollo de sistemas de administración de la información han sido significativos y nos han traído conceptos nuevos que se deben incorporar a los conocimientos que se manejan en el diseño y publicación de información. Conceptos como el de utilizar metalenguajes. Un metalenguaje nos permite acompañar la información con información acerca de la información. Nos permite agregarle unas etiquetas al texto que lo acompañan y le dan significado a lo que antes eran solo párrafos de texto. Si nos apoyamos en la utilización de un metalenguaje, podemos procesar la información. Un párrafo de texto ya no es solo eso sino que se puede catalogar como <conclusiones>, como <título>, o como <autor>. Es una manera diferente de enfrentar la administración de la información, aporta elementos adicionales que permiten sistematizar y programar los procesos. Si le adicionamos etiquetas a la información,

podemos incorporarla como parte de un sistema lógico de organización. Una vez la información es parte de un sistema, son múltiples las ventajas que se obtienen.

El método recomendado para diseñar publicaciones con contenido variable consiste en independizar la estructura de organización de la información, de la manera como esta se visualiza o representa. La información se clasifica y codifica en estructuras lógicas de organización por una parte, y por la otra, se diseñan plantillas que invocan la información y la despliegan según propiedades definidas sobre unas variables. La imagen que vemos en la pantalla cambia fácilmente en la medida que cambia la información que la genera. La información deja de estar amarrada a una única manera de representarse y permite tener múltiples versiones de una misma publicación, también permite definir varios conjuntos de estilos y dejar al usuario tomar parte en la decisión sobre cómo se muestra la información. Es la única manera en que podemos trabajar con información que genera el usuario en el momento mismo de consultar el documento, como el caso de un *chat*. Es el modelo que emplean la mayoría de portales en internet que manejan información cambiante, es el modelo que ayudaron a desarrollar publicaciones como Hotwired y Yahoo. El modelo lo componen las dos partes: el sistema de organizar la información y el sistema de representación.

Los contenidos que se desean publicar, por lo general han sido desarrollados para otros medios y no se adaptan al modelo, es necesario entonces recomponer la información para que pueda entrar a formar parte de un sistema dentro de una red informática. Para poder construir un sistema de representación debemos contar primero con una estructura de organización de la información. Una página o un tamaño de fuente forman parte de un sistema de representación de la información y no de una manera de organizarla. La información se debe clasificar y codificar en estructuras lógicas, estructuras que permitan procesar la información como son los directorios y las bases de datos. Si se cuenta con este tipo de estructuras, el computador puede entonces construir listas, cadenas y estructuras ramificadas, puede establecer relaciones cruzadas entre diferentes elementos y puede procesar algorítmicamente la información. La información se tiene que fragmentar y clasificar; los fragmentos son los que pueden llevar etiquetas, etiquetas con información acerca de la información, información sobre una manera de clasificar la información. La clasificación nos permite establecer jerarquías y construir conjuntos y subconjuntos con los elementos, es la base para cualquier organización.

5. TEJIENDO LA RED

El primer paso para construir una estructura lógica de organización es definir los diferentes usuarios y los diferentes roles que puede asumir cada uno frente al sistema. A partir de esta información podemos construir la lista de los casos de uso que nos interese desarrollar. Una definición clara de necesidades desde el principio es importante para entender la magnitud del proyecto. El siguiente paso es definir las relaciones posibles entre los objetos y la información que estos contienen. Con esto llegamos a construir un mapa de relaciones entre los diferentes elementos que componen el sistema. Este gran mapa de todo el sistema debe definir también las relaciones con otros sistemas y las posibles salidas del sistema. El mapa de relaciones se traduce fácilmente en un mapa de navegación si cambiamos de punto de vista y nos situamos como usuarios frente al sistema. Cada relación en cada caso de uso se corresponde con un posible vínculo de navegación entre cada uno de los fragmentos de información que componen el sistema. Aunque maneje información muy dinámica, el sistema en sí mismo no lo es tanto. Debemos desde un principio dejar establecidas las posibles relaciones ya que un cambio más adelante significaría rediseñar la base de datos.

Una vez tenemos definido el mapa de relaciones que queremos desarrollar, el siguiente paso es definir la manera en que queremos que se den estas relaciones. En la medida en que se va tejiendo esa red de relaciones surge la pregunta: ¿Qué quiero que ocurra en un momento dado? Según el concepto de programación por objetos, la secuencia que ocurre es:

Un **objeto** - ante un **evento** - desencadena una **acción**

O sea que si determinamos la acción que deseamos generar, debemos proponer el objeto y el evento. Colocamos objetos virtuales que al ser seleccionados desencadenan acciones preestablecidas. Cada objeto contiene sus propias posibilidades de acción. La multiplicidad de objetos en la pantalla implica una multiplicidad de posibilidades. Cada vez que el usuario se enfrenta con la imagen de la pantalla y acepta la invitación a actuar, queda enfrentado a una sucesión infinita de momentos de inestabilidad que le piden tomar decisiones. Es importante anotar que los objetos existen y se transforman a lo largo de una línea de tiempo y que los eventos que pueden ocurrir son categorías de temporalidad. Diseñar interactividad consiste en proponer un sistema de posibles relaciones entre objetos virtuales y diferentes situaciones o momentos relativos a una red de relaciones de temporalidad. La red existe en presente pero se diseña pensando al futuro. Se diseñan los objetos virtuales que van a estar presentes cuando un

posible usuario se encuentre ante una situación previsible. Tratamos de imaginar relaciones que puedan ocurrir en el futuro y construimos los escenarios donde estas se representarán.

Un acontecimiento es cuando algo ocurre, es consecuencia de otro acontecimiento y no requiere de un interlocutor. En algunos casos podemos predecir los acontecimientos pero no podemos hacer que ocurran. En el caso en que tenemos la capacidad de intervenir sobre los acontecimientos hablamos de una acción. **Una acción** ocurre siempre a partir de la relación entre un usuario y un objeto virtual. Para que se de esta relación entre un objeto virtual y un usuario se requieren dos elementos: un cursor que permita señalar un punto específico de la pantalla y una tecla que permita hacer click (puede ser en el ratón o con la tecla "enter"). Cualquier elemento en la pantalla que permita ser señalado y nombrado o cualquiera de las teclas del teclado puede utilizarse entonces como generador de una acción. El cursor se convierte en el punto de encuentro entre el usuario y el escenario que contiene los objetos virtuales. Más allá de unas coordenadas en la pantalla el cursor nos está señalando un momento en el tiempo, es el punto que marca el discurrir de las acciones; el cursor existe solo en presente y lo señala y define.

Un objeto es cualquiera de los elementos que componen el sistema y que se pueden llamar en diferentes momentos dentro del principio de modularidad que aplica a todos los objetos digitales. Los objetos virtuales están contruidos a partir de otros objetos y heredan sus características y atributos. Un objeto hereda las posibilidades de relación que trae el objeto que contiene. Un objeto debe invitarnos a la acción, debe insinuarse como opción para que la pueda identificar. Una vez que reconozco que un objeto es un posible interlocutor debo definir los términos de esa posible interlocución. Cualquiera que sea el evento generador debe tener un momento de decisión previo que implica que se de una acción de selección. Son varios los modos en que puede ocurrir la selección como se puede ver en el archivo [ejemplos/seleccion.html](#)

La interacción es entre nosotros y la información, cualquier interacción con otros usuarios es a través de información. Interactuamos cuando seleccionamos, interpretamos, traducimos o compartimos información. El escenario donde esto puede ocurrir es una situación temporal, le asignamos una identidad propia para diferenciarlo de otros escenarios donde ocurren situaciones similares. A un momento particular que contiene una serie de alternativas de continuidad es lo que llamamos un sitio en la red, es una combinación particular de

elementos sobre la pantalla con identidad propia. Identidad visual por un lado, pero antes que ella está el hecho de poder diferenciar una situación particular de otras situaciones distintas. Aunque cambie la imagen que vemos en la pantalla, podemos sentir que estamos en el mismo sitio. No es algo que dependa de la ubicación del servidor donde se encuentra almacenada una información (podemos construir un sitio con información proveniente de varios servidores diferentes simultáneamente). Un sitio son todas las variaciones que puede tener una combinación de elementos en un mismo escenario.

Un sitio es a un mismo tiempo un nudo de convergencia y de bifurcación, es el punto de partida para llegar a otras situaciones y es también el punto de llegada. Un sitio se transforma en otro como consecuencia de que algo ha ocurrido, ya sea causado o no por mis decisiones. Todo acontecimiento es consecuencia de otro anterior: los acontecimientos y las acciones se suceden unas a otras, se encadenan y fluyen siguiendo cursos, transcurren en el tiempo. La imagen en la pantalla es el escenario de los acontecimientos, representa el transcurrir de las acciones; pero para que estas sigan su curso, debemos siempre mantenernos en constante interacción.

La interacción se puede dar de muchas maneras: puede ser una experiencia individual o una colectiva; puedo simplemente seleccionar entre alternativas preestablecidas o puedo intervenir sobre lo que veo. No es lo mismo navegar por la red consultando información que enviar un correo con un archivo adjunto, son niveles diferentes de interacción. Podemos situar el nivel básico de interacción en la selección entre alternativas preestablecidas, aceptamos lo que vemos así como lo vemos. Un nivel superior de la interacción es cuando el usuario puede actuar sobre el contenido de la información y no solo sobre su representación. Esta condición es indispensable para que dos individuos se puedan comunicar entre sí o para que varias personas puedan acceder a un mismo sitio y afectar la información presentada de manera simultánea. El término comunidad virtual se ha generalizado para referirse a estos grupos de personas que estando conectadas en la red establecen algún nivel de comunicación entre ellas. Dentro de este nivel se encuentran modos de participación como el chat y el foro, pero también puede referirse a un grupo de usuarios que estén descargando la misma canción simultáneamente.

En la acción de consultar el buzón de correo electrónico, el momento en que un usuario digita su clave de seguridad y su identidad es validada está ocurriendo un nivel de interacción complejo. Una transacción de dinero por internet

implica procesos aun más complejos de validación. Cuando la interacción es colectiva como en el chat se multiplican las variables. El elemento común que tienen estas acciones y que comparten con muchas otras formas de comunicación en red, es que todas constituyen una acción de consulta en una base de datos.

6. CONSULTA A UNA BASE DE DATOS

La consulta en una base de datos se define siguiendo una secuencia de acciones que permiten al usuario acceder lo más pronto posible a la información que solicita. La secuencia de acciones comienza cuando el usuario se encuentra ante un escenario que le ofrece unas alternativas de búsqueda según algún criterio de clasificación.

- A.** La primera acción debe ser la de escoger entre una serie de opciones de búsqueda en la base de datos. Se selecciona entre las categorías y subcategorías de clasificación que se han empleado para organizar la información. Una vez que el usuario decide la opción que desea escoger debe hacer click sobre la opción seleccionada y esto genera una orden de búsqueda.
- B.** El segundo evento entonces es la respuesta que recibe el usuario a la consulta que efectuó de acuerdo a la opción seleccionada. Esta respuesta se le presenta en términos de una lista con los títulos o una retícula de imágenes que representan los diferentes registros en la base de datos que cumplen con los criterios de clasificación que se seleccionaron.
- C.** La respuesta se constituye en una nueva serie de opciones, en este caso cada uno de los títulos de la lista o las imágenes de la retícula traen consigo la cualidad de poder actuar como botón, de tal manera que la tercera acción es la de seleccionar un título o una imagen.
- D.** Al seleccionar un título o una imagen se recibe una respuesta, la respuesta consiste en una navegación en profundidad dentro de la información. Obtenemos información adicional como una corta descripción, una imagen representativa o criterios más complejos de clasificación. Esta información debe ser necesaria para que el usuario sepa si era lo que buscaba o no. Es un nivel intermedio, no ofrece la información completa pero si lo suficiente para tomar una nueva decisión. Dependiendo de la complejidad de la base

de datos, este nivel puede subdividirse o puede dejarnos directamente en el quinto evento.

- E. El quinto evento tiene dos posibilidades: Una es llamar al registro seleccionado y consultarlo, la otra es devolverse en bucle a un evento anterior para seleccionar otro diferente. Si nos decidimos por un registro en particular, significa que ya encontramos lo que buscábamos y por lo tanto termina la consulta. En caso de optar por la segunda opción, no se generan nuevos eventos sino que repetimos los mismos pasos pero tomando decisiones diferentes.

Todo el esquema es circular, son acciones que se agrupan en bucles, segmentos de tiempo circular que conforman ciclos. No estamos frente a una narración lineal, la representación de la base de datos es una estructura modular compuesta de segmentos que son bucles de temporalidad. La acción básica es compartir información; seleccionar una alternativa se convierte en indicarle al computador la decisión que tomo; el computador a su vez, me comparte la información que almacena y genera. La secuencia analizada es solo para la consulta de información. En el caso de darse la oportunidad para afectar el contenido de la información, como en el chat, se generan otros bucles de acción para seleccionar donde insertar la información. La acción misma de insertar o consultar la información, es un bucle de tiempo.

Compartir información con un interlocutor es lo que llamamos conversación. La conversación es algo muy dinámico, una pregunta genera una respuesta que a su vez genera otra pregunta; un comentario genera otro que aprueba o disiente, se genera flujo de información en términos de una continua retroalimentación. Un *chat* o un foro son modos de conversación contruidos sobre una base de datos. La base de datos se compone de preguntas, respuestas y comentarios; se clasifica en temas y participantes y se genera en tiempo real. Una base de datos de este tipo se desarrolla en un presente expandido, parece tiempo real pero ocurre sobre un espacio y un tiempo virtuales: todo lo ocurrido hasta ahora, todos los comentarios y respuestas se nos presentan a un mismo tiempo, el tiempo pasado se comprime en presente y el futuro se alarga, la respuesta a mi pregunta puede tardar dos minutos o dos horas. Son flujos de bucles de retroalimentación de preguntas, respuestas y comentarios. Es decir que estamos construyendo una conversación dentro de un entorno virtual, nos convertimos en los personajes del laberinto, participamos en la construcción del laberinto.

Los registros en la base de datos nos proveen de la información necesaria para construir objetos virtuales. Un objeto material (como una camisa) es representado por un objeto virtual que se define a partir de información acerca de un número de referencia, una talla, un precio, un color y una fotografía. Además de representar objetos materiales, los objetos virtuales pueden representar símbolos como la boleta que da derecho a asistir a un concierto, o representar un evento tan virtual como es una transferencia de dinero. Pero la información del objeto virtual no necesariamente debe representar otra realidad, se puede estar representando a si misma. Un mensaje de correo electrónico no representa nada diferente que a si mismo y es clasificado en el momento en que se genera. Las bases de datos donde se colocan los mensajes de correo o los comentarios de un chat, son totalmente dinámicas, están siempre en continua transformación, están programadas para clasificar información que no se ha creado todavía.

7. EL SOPORTE LIQUIDO

La unidad de página que hemos desarrollado para la información escrita en una hoja de papel se ve transformada en la medida que la ventana del navegador no tiene un tamaño fijo. El soporte de la imagen se comporta como líquido, el contenido flota y se reacomoda cada vez que alteramos las proporciones de la ventana. Un sitio no es un objeto material fijo sino una secuencia de instrucciones sobre como desplegar una información. El resultado de seguir estas instrucciones es diferente según sean las características tecnológicas del equipo que se use, dependen de variables como la resolución del monitor, el sistema operativo y el navegador instalados, y la configuración personal.

Poner en escena una representación puede responder a una metáfora a la manera de la representación realista o ser simplemente un tema gráfico. Una interfaz puede ser un dibujo o una fotografía de un entorno conocido para facilitar la representación alegórica y la construcción de imágenes simbólicas; pero también puede resolverse sobre simbologías editoriales y apoyarse únicamente en el texto. Diseñar una interfaz no es un problema con una única solución correcta, es el reflejo de una forma particular de representar. La identidad que define a una serie de escenarios como conjunto está dada por una manera de hacer, por un estilo. Una manera de representar se define con un conjunto de reglas de comportamiento, no es una solución única sino una propuesta que aplica a una multiplicidad de escenarios.

El problema que se plantea entonces es: ¿Cómo reflejar una manera particular de representar sobre un soporte tan variable?. Se han propuesto varias soluciones que difieren en el grado de rigidez con que se asigna el diseño a la información representada. Estas van desde la total liquidez hasta la opción de una tabla rígida que flota sobre un medio líquido. Se pueden ver 5 ejemplos diferentes en el archivo ejemplos/liquido/index.html.

En el ejemplo vemos como al redimensionar la ventana del navegador, el contenido se reacomoda a la nueva ventana de diferentes maneras. Es claro que cuando el texto adquiere un tamaño tan pequeño que se vuelve ilegible, o una imagen se distorsiona tanto que pierde su sentido, alcanzamos el límite de redimensionar una ventana. Si le cambiamos la configuración de resolución al monitor, vamos a ver como ocurre también un reacomodo del contenido de la ventana; algo similar vamos a ver si consultamos los mismos archivos desde diferentes navegadores y sistemas operativos. La que se denomina como tabla mixta es un buen ejemplo de cómo un diseño se puede adaptar a muchas condiciones diferentes de visualización sin perder la intención de estilo; se bloquea el tamaño absoluto de algunas celdas, mientras que a otras se les asigna un tamaño en términos de un porcentaje del tamaño de la ventana.

Otro factor de variabilidad muy grande lo constituye la fuente y el tamaño al que se muestran los textos. El html permite definir tamaños como grande, mediano y pequeño o 1, 2, 3, 4, 5. En cualquier caso son medidas absolutamente relativas a unas condiciones de visualización, no se refieren a tamaños en milímetros sino a tamaños relativos a la resolución de la pantalla. La fuente que se especifica es solo una sugerencia pues depende de si el usuario tiene instalada esa fuente en su sistema o no. Convertir los textos a imagen garantiza que se desplegarán tal como queramos, pero pierden la capacidad de ser procesados o indexados, además de aumentar considerablemente el tamaño del archivo que se debe transferir. En lugar de pretender darle rigidez al diseño para tratar de obligar una única manera de mostrar, es mejor buscar una adecuada adaptación al soporte variable.

Cuando se diseña un sitio que comprende muchas páginas, todas con contenido variable, o cualquier sitio complejo por el volumen de información, se debe proponer un sistema de plantillas y estilos y no soluciones particulares. Un estilo hace parte de un sistema de representación que continua viejas tradiciones de jerarquías tipográficas para señalar un tipo particular de contenido (título, pie de página, cuerpo de texto, etc.). Un conjunto de estilos agrupa unas

características similares y les define unas proporciones y una paleta de colores particular. Este conjunto de estilos armónicos es la base que define la identidad de un sitio. En la carpeta [ejemplos/liquido/estilos.css](#) encontramos un ejemplo de un conjunto de estilos compuesto de: texto, título y botones. Si se construyen las páginas vinculadas a estos estilos, se garantiza un conjunto armónico que puede ser fácilmente cambiado. Las relaciones entre los estilos son de proporción, por eso se adaptan tan bien a las variaciones de visualización.

Una plantilla tiene unos pocos elementos fijos pero se construye principalmente con campos vacíos que se alimentan de contenido como resultado de una búsqueda en una base de datos. Puede servir para que el usuario ingrese la información al sistema o ser el escenario para presentar el contenido de la base de datos. Una plantilla es la puesta en escena de información variable, son estilos asociados con zonas de la pantalla, son solo reglas de juego de como escenificar procesos de comunicación, son el libreto de la narración ramificada. Diseñar una plantilla nos enfrenta a nuevos conceptos de realidad, diseñamos campos vacíos de consistencia líquida asociados a información dinámica. Con toda la virtualidad que esto implica, el resultado es una imagen que se proyecta en la pantalla y nos permite consultar la información.

8. ORIENTANDO AL NAVEGANTE

Para ser claramente comprendido, un modo de representar debe ser consistente y bien definido. Las dudas del usuario no deben ser sobre la interfaz sino sobre sus preferencias. Un buen sistema de representación le ofrece al usuario elementos de ayuda para facilitar la consulta. Los elementos más usados para facilitar la orientación y la consulta son: los mapas, los títulos y los íconos.

Mapas

Un mapa es el resultado de ver un conjunto desde más lejos, al alejarnos de algo dejamos de ver las minucias, se pierden los detalles y las pequeñas diferencias para ganar en claridad y en visión de conjunto. Es solo un punto de vista, es un imaginario que construimos en nuestra mente para alejarnos de lo que tenemos delante y poder abstraer solo los elementos más significativos y trazar relaciones entre ellos. Los mapas representan visualmente este esquema de relaciones para que sea más fácilmente comprendido, nos permiten "ver" relaciones que de otra forma permanecerían ocultas.

Un índice es un mapa que trazamos sobre las palabras, nos alejamos lo suficiente del texto para ver solo los títulos de los capítulos y poder mostrar las relaciones entre los textos. Es una mirada de conjunto sobre la información, me orienta sobre todo el contenido de un documento. Un índice se traduce fácilmente a un menú de navegación cuando trabajamos en formatos digitales, en muchas ocasiones aunque cambie su presentación, mantiene su esencia y nos indica la manera en que está fragmentado un contenido de información.

Títulos

Es poner en palabras la identidad de un sitio, es una frase corta que define el conjunto que se me presenta en un momento dado. Las posibles variaciones de un mismo título se definen con subtítulos. No es igual que ponerle nombre a los objetos para ayudar a identificarlos, eso también ayuda a la orientación pero en este caso se trata de identificar un sitio con un texto y no un objeto. Un título es la manera de ponerle nombre al conjunto, es una relación directa con el índice y por medio de éste podemos descubrir las relaciones que se establecen entre las diferentes secciones.

Iconos

Un ícono se refiere a una imagen simbólica que representa o refiere otra realidad diferente a sí misma. Un ícono es parte de un lenguaje, es un símbolo, actúa como la palabra, no está ahí para ser recorrido con la mirada sino para servir de enlace con la información que representa. Un ícono debe tener un límite bien definido para servir como tal, un ícono no puede ser ambiguo o indeterminado. En muchos casos se crea un ícono para representar una imagen o un escenario simplemente reduciendo considerablemente su tamaño, otras veces se emplea un fragmento o sección de una imagen para representar el conjunto. De cualquier manera, para que una imagen represente a otra debe guardar un parecido o estar contenida dentro de ella.

Una imagen pequeña de una manzana puede representar a la sección de manzanas, puede indicar la sección de frutas, puede referir a la agricultura o los alimentos, o inclusive se puede usar como símbolo para representar ropa para bebés o la ciudad de Nueva York; lo que nunca puede hacer es representar a la sección peras o bananos, debe haber algún tipo de relación de significado cultural en la asociación necesariamente.

9. EL MODELO EN LA PRACTICA

La relación entre un sistema de representación y uno de organización se puede lograr de diferentes maneras y con diferentes lenguajes de programación. Para facilitar la explicación a personas con pocos conocimientos de programación, utilizamos un método que consiste en diseñar la representación con un archivo en **Flash MX** que invoca la información contenida en el documento **xml**. Decidimos utilizar Flash por la cantidad y diversidad de posibilidades que ofrece desde el punto de vista gráfico y la facilidad de utilización por diseñadores con muy pocos conocimientos en lenguajes de programación. Flash al ser un software de autoría contiene su propio lenguaje de programación llamado **Action Script** que se asemeja muchísimo al lenguaje **Java** y puede integrarse con este. El archivo **swf** generado por Flash se inserta dentro de un sencillo documento html que lo contiene, permitiendo así integrarnos a una estructura de navegación a través de archivos en formato html. Los archivos html pueden ser leídos por cualquier navegador y se vinculan unos a otros fácilmente.

Como metalenguaje se propone utilizar el lenguaje **XML** (Extensible Markup Language). Es el estándar propuesto por la W3C para el intercambio universal de la información, provee un medio uniforme para la comunicación entre diversos tipos de plataformas, además de medios para verificar la estructura de la información que se intercambia y medios automáticos de transformación y presentación de la información. La información contenida en documentos **xml** puede ser muy dinámica, actualizarse continuamente e inclusive generarse en tiempo real. Podemos independizar un documento que contiene una variable con propiedades definidas por un lado, y una base de datos con información por el otro. La base de datos no tiene que ser una gran acumulación de información sino que puede ser una estructura muy dinámica que se construye en presente. Una base de datos es un modo de organizar información; un documento XML es un archivo de intercambio, es una estructura ramificada construida a partir de una búsqueda en la base de datos. Un documento html o una película Flash pueden llamar la información contenida en el documento XML y desplegarla en la pantalla. Esta estructura de relaciones entre la interfaz que se le presenta al usuario y la base de datos es el principio del diseño con información dinámica; al independizar la información de su presentación, podemos diseñar formas asociadas a contenidos dinámicos y transformables. Separamos un segmento de narración en componentes independientes que se recomponen según parámetros de programación. La transformación de la información depende de eventos y acciones que ocurren en el momento en que se la consulta.

Los documentos **xml** pueden ser contruidos digitándolos dato por dato o pueden ser generados por una base de datos a la que se le pide que los construya según los parámetros que se definen en un applet de java. El código **xml** se parece bastante al html en su gramática, sólo que no tiene predefinidos los posibles elementos, por tanto éstos se construyen a voluntad del programador. Para comprender más fácilmente veamos en el ejemplo siguiente como se puede vincular información dinámica en forma de texto, imagen, animación, sonido y vínculo a un URL con un archivo en donde representamos toda esta información de manera que se actualice automáticamente cuando cambie la información.

EJEMPLOS 1 Y 2: ACTUALIZACIÓN AUTOMÁTICA

En la carpeta denominada ejemplos se encuentran los archivos **llamarXML fla** y **lista.xml**, que son la base del ejercicio sobre como vincular un documento **xml** con una película en **Flash MX**. Para visualizarlos necesitaremos un sencillo editor de texto como *notepad* y una versión reciente de *Flash*. Empecemos entonces abriendo los dos archivos, cada uno en el programa correspondiente.

El archivo [ejemplos/llamarXML fla](#) nos define tres botones de navegación en términos de unas coordenadas, un estilo de letra y un área sensible sobre la que se remite a otra situación en caso de ser seleccionada. Estas propiedades se le asignan a campos de texto vacíos definidos como las variables *boton1*, *boton2* y *boton3*. Para darle valor a las variables se invoca el documento **lista.xml** donde se coloca la información respectiva al texto de los botones, al texto del título con la imagen correspondiente a cada uno, y también al sonido que se desea remitir en cada caso.

El documento [ejemplos/lista.xml](#) se construye por aparte de la siguiente manera: un tronco que llamamos "datos" y unas ramas que llamamos "objetos". Cada una de estas ramas está compuesta de elementos que llamamos **<boton>**, **<texto>**, **<imagen>** y **<sonido>**. La manera de nombrar un objeto es colocando etiquetas que se reconocen fácilmente por estar escritas dentro de los marcadores **<** y **>**. El valor correspondiente de cada uno de estos datos es el texto que encontramos entre el inicio y el final de cada etiqueta definida por los marcadores **<** **>** y **</ >**.

El documento **lista.xml** es sólo un texto que dice:

```
<?xml version="1.0" encoding="UTF-8"?>
<datos>
  <objeto>
    <boton>Tren</boton>
    <texto>velocidad máxima: 80 Km/h</texto>
    <imagen>imagenes/tren.jpg</imagen>
    <sonido>imagenes/tren.swf</sonido>
  </objeto>
  <objeto>
    <boton>Carro</boton>
    <texto>velocidad máxima: 200 Km/h</texto>
    <imagen>imagenes/carro.jpg</imagen>
    <sonido>imagenes/carro.swf</sonido>
  </objeto>
  <objeto>
    <boton>Avión</boton>
    <texto>velocidad máxima: 300 Km/h</texto>
    <imagen>imagenes/avion.jpg</imagen>
    <sonido>imagenes/avion.swf</sonido>
  </objeto>
</datos>
```

Otra manera de visualizar esta misma información es como una tabla con celdas donde cada fila corresponde a un objeto:

boton	texto	imagen	sonido
Tren	velocidad máxima: 80 Km/h	imagenes/tren.jpg	imagenes/tren.swf
Carro	velocidad máxima: 200 Km/h	imagenes/carro.jpg	imagenes/carro.swf
Avión	velocidad máxima: 300 Km/h	imagenes/avion.jpg	imagenes/avion.swf

Tenemos ahora otra forma de mostrar la información pero una misma información representada, este es el principio sobre el que todo se basa. Este esquema se puede complicar bastante cuando la estructura se ramifica más, pero entre tanto nos ayuda a comprender mejor las características de los documentos XML.

Cuando seleccionamos el primer cuadro de la película de flash, en la ventana de acciones encontramos el siguiente texto que se utiliza para llamar o invocar el documento **lista.xml**:

```
var datosxml = new XML();
datosxml.ignoreWhite = true;
datosxml.onLoad = loadllamar;
function llamar() {
    datosxml.load("lista.xml");
}
llamar();
function loadllamar(success) {
    if (success) {
        boton1 = (datosxml.firstChild.childNodes[0].childNodes[0].firstChild);
        boton2 = (datosxml.firstChild.childNodes[1].childNodes[0].firstChild);
        boton3 = (datosxml.firstChild.childNodes[2].childNodes[0].firstChild);
    }
}
```

Flash carga el archivo **lista.xml** y le remueve los espacios en blanco. Usamos las tabulaciones y renglones para leer y escribir más fácilmente el código, pero al computador le indicamos `datosxml.ignoreWhite = true;` para eliminar los espacios en blanco y así reconstruir la línea continua del hipertexto que fluye por la red.

Para nuestro primer caso el valor de `boton1 = (tren)`, el de `boton2 = (carro)` y el de `boton3 = (avión)`, que corresponden a la primera rama de cada objeto. Si escribimos en la película de Flash el código:

```
boton1=(datosxml.firstChild.childNodes[0].firstChild);
```

nos referimos al valor de la primera rama del tronco que corresponde a la palabra `<objeto>`. Si queremos referirnos a la primera rama de la primera rama, el código será entonces:

```
datosxml.firstChild.childNodes[0].childNodes[0].firstChild;
```

Si queremos referirnos a otros elementos del archivo `lista.xml` el código correspondiente que debemos agregar es `.childNodes` para cada rama de la rama. Los corchetes `[1]` se utilizan para referirnos a la segunda rama, la tercera, etc. O sea que para invocar el valor del título correspondiente a `boton1` el código correspondiente será:

```
(datosxml.firstChild.childNodes[0].childNodes[1].firstChild);
```

Es importante aclarar que la primera rama no es la uno sino la cero, para establecer el orden correcto. Igualmente se debe recordar que el nivel siguiente a una rama puede ser otra rama, o el valor de la rama si agregamos el texto `.firstChild`.

El valor de la variable "imagen" es un texto que indica una dirección en donde está almacenada una imagen en formato jpg. Esta información la utilizamos para dar la orden: `loadMovie(imagen, "foto");`

En donde "imagen" es el valor de la variable y "foto" es el nombre de una instancia de un movie clip que se encuentra vacío y se llena con la imagen que se encuentra en la dirección indicada en este valor. De esta manera invocamos no sólo un texto que llena un campo de texto vacío sino también una imagen que llena un *campo de imagen* vacío. De la misma manera que a un campo de texto vacío le podemos asignar atributos de fuente, color y tamaño, aunque el movie clip "foto" está vacío, posee características definidas de tamaño y ubicación donde debe desplegarse en la pantalla.

Este procedimiento lo usamos también para cargar archivos de sonido o animaciones y videos grabados en formato swf y hacer que sean parte de la información que se despliega al seleccionar un botón. En este ejemplo solo estamos llamando un sonido y una imagen, pero el espectro de posibilidades que se le abren a la narración audiovisual se amplía muchísimo pues implica trabajar con variables dinámicas y no con elementos fijos. Imágenes y sonidos pueden ser programados, un pequeño archivo en flash puede llamar imágenes, sonidos y videos desde una base de datos según se le pidan. Se pueden construir narraciones de manera algorítmica, se pueden encadenar segmentos de modo aleatorio; se puede aplicar programación de muchas maneras y muy fácilmente ya que los segmentos existen numéricamente como registros en una base de datos.

La aplicación es muy sencilla pero sirve para mostrar como la información puede estar en un documento independiente mientras que las propiedades con las que se va a mostrar esta información están en otro documento distinto que puede encontrarse en un lugar diferente, en cualquier otro equipo conectado a la red. Poder independizar estos dos aspectos es algo que se logró con la aparición de las tecnologías de red y que plantea el reto de trabajar en un escenario totalmente diferente para el diseñador. Los documentos que se diseñan de esta manera se pueden llenar de contenido con información dinámica que se transforma y actualiza continuamente sin necesidad de reelaborar el documento. Para ver más claramente las posibilidades que se abren, hagamos unos pequeños cambios en el documento xml y veamos como se recompone la información.

El código que utilizamos se refiere a tronco y ramas y a valores de estas, así como a la construcción de listados de elementos; el código responde a una posición relativa y no a un valor absoluto, puede ser aplicado igualmente a otros documentos xml y permite establecer vínculos con información dinámica que cambie frecuentemente sin cambiar el código. Es así como en el ejemplo 2 tenemos exactamente los mismos campos y el mismo código pero con la única diferencia que ahora llamamos el archivo [ejemplos/lista2.xml](#) en lugar de lista.xml.

Con este cambio se reemplaza toda la información, los objetos definidos pueden ser otros y las imágenes que los identifican no tienen que tener las mismas características de las anteriores. Se conservan características de forma y de comportamiento en los botones pero no en el contenido de los textos ni en las imágenes invocadas. Los rangos de variabilidad están dados por aspectos como el número máximo de caracteres que permite un campo de texto o las coordenadas del punto de inicio de una imagen.

Los botones ahora se llaman Manzana, Durazno y Mango en lugar de Tren, Carro y Avión. El comportamiento de los botones en cambio se mantiene igual, lo mismo que su presentación en cuanto a la fuente, el color y el efecto *rollover*. En donde cargábamos el sonido ahora es una fotografía, la acción es *loadmovie*, no importa si se refiere a un archivo en formato **.jpg** o a uno en formato **.swf**. El código: `(datosxml.firstChild.childNodes[2].childNodes[3].firstChild)`; se refiere a la cuarta columna sin importar cual sea la etiqueta de ésta.

EJEMPLOS 3 Y 4: LISTAS Y CONJUNTOS

Ocurre diferente en el tercer ejemplo, en el que llamamos el archivo [ejemplos/docentes.xml](#). En este caso se vuelve relevante el valor de la etiqueta de la columna, ya que es a partir de ésta que vamos a construir las listas. En este ejemplo, además de llamar la información, construimos una lista con los elementos contenidos en el documento XML. Una vez tenemos la lista le preguntamos por los valores relacionados con el elemento que seleccionemos de la lista. Este acto es el segundo evento en la consulta a una base de datos, es ya un primer nivel de retroalimentación.

En el ejemplo la lista corresponde al conjunto de profesores de la Escuela de diseño gráfico en el momento en que se conformó como escuela en el año 2002. Sobre cada profesor tenemos información acerca del nombre, el área de

docencia a la que pertenece y una fotografía del profesor. La lista la construimos con los nombres de los profesores, igualmente se podría hacer con cualquier otra de las columnas del documento, es decir con cualquiera de las etiquetas.

Se hace necesario entonces definir tres funciones básicas: `getChildName(obj, tag)`, `getChildValue(obj, value)` y `GetProfList(Propiedad, Valor)` que corresponden a buscar una etiqueta, buscar el valor correspondiente a una etiqueta y construir una lista. Las otras dos funciones que necesitaremos serán `changeHandler()` y `LoadProf(ProfName)` que corresponden a seleccionar un elemento de la lista y a llamar los demás datos asociados al elemento seleccionado. Son en total cinco funciones que definen toda la acción de construir la lista y luego seleccionar un elemento para obtener información adicional sobre él. Esta acción es un bucle de tiempo que se puede repetir muchas veces seleccionando un profesor diferente cada vez. El código completo entonces será:

```
listaprofes.setChangeHandler("changeHandler");
function changeHandler(){
    LoadProf(listaprofes.getValue());
}
function getChildName(obj, tag) {
    arrayhijos = obj.firstChild.childNodes;
    for (i=0; i<arrayhijos.length; i++) {
        if (arrayhijos[i].nodeName == tag) {
            return arrayhijos[i];
        }
    }
}
function getChildValue(obj, value) {
    arrhij = obj.childNodes;
    for (t=0; t<arrhij.length; t++) {
        if (arrhij[t].nodeName == value) {
            return arrhij[t].firstChild.nodeValue;
        }
    }
}
function GetProfList(Propiedad, Valor) {
    var doc = new XML();
    doc.onLoad = cargarlista;
    doc.ignoreWhite = true;
```

```

doc.load("docentes.xml", doc);
function cargarlista(success) {
    if (success) {
        listaprofes.removeAll();
        arrayhijos = doc.firstChild.childNodes;
        for (i=0; i<arrayhijos.length && !entro; i++) {
            items = arrayhijos[i].childNodes;
            if (getChildValue(arrayhijos[i], Propiedad) == Valor) {
                listaprofes.addItem(getChildValue(arrayhijos[i], "profesor"));
            } else {
                if (Propiedad == "") {
                    listaprofes.addItem(getChildValue(arrayhijos[i], "profesor"));
                }
            }
        }
    }
}

GetProfList("", "");
function LoadProf(ProfName) {
    var doc = new XML();
    doc.onLoad = cargarlista;
    doc.ignoreWhite = true;
    doc.load("docentes.xml", doc);
    function cargarlista(success) {
        if (success) {
            arrayhijos = doc.firstChild.childNodes;
            for (i=0; i<arrayhijos.length && !entro; i++) {
                items = arrayhijos[i].childNodes;
                if (getChildValue(arrayhijos[i], "profesor") == ProfName) {
                    nombre = ProfName;
                    area = getChildValue(arrayhijos[i], "tema");
                    limage = getChildValue(arrayhijos[i], "imagen");
                    loadMovie(limage, "foto");
                }
            }
        }
    }
}
}

```

En el siguiente caso vamos un paso adelante y construimos una nueva lista con los profesores que tengan un valor determinado para la etiqueta que señala el área de docencia. En realidad es la misma función `GetProfList(Propiedad, Valor)` con la que llamamos la lista completa, la que usamos para construir una lista con los profesores que a una propiedad (en este caso área de docencia) les corresponda un valor determinado. La orden se da en el botón con el código:

```
on (press) {  
    GetLista("tema", categoria4);  
}
```

Nótese que la propiedad "tema" esta definida por la etiqueta correspondiente en el documento XML, mientras en el inicio definimos la variable `categoria4 = "Informática Gráfica"`; con el fin de hacer genérico el botón de selección. La orden se refiere entonces a que construya una lista con todos los profesores cuya área de docencia sea Informática gráfica.

Es muy importante ser exacto y consistente en la manera de nombrar y escribir los elementos pues el sistema es sensible a cambios en las vocales tildadas, a letras en mayúscula e inclusive a espacios en blanco después de la palabra y fácilmente deja de reconocer un valor por estar escrito de manera diferente.

EJEMPLOS 5: CAMBIO DE CATEGORÍAS

En el último ejemplo tenemos varios factores adicionales que hacen más compleja la consulta de la información. Sobre la lista de proyectos encontramos dos botones que definen si la lista se hace por título del proyecto o por autor. Las demás subcategorías se mantienen activas y los botones generan las mismas clasificaciones, solo que según se escoga, la lista se despliega por autor o por título. La manera de lograrlo es con unos pequeños cambios en el código, ya que la función `GetTitulos(Propiedad, Valor)` es la misma. Para el caso de los títulos la función se asocia con la función `LoadTits(TitsName)` mientras que para los autores la función `GetTitulos` se asocia con `LoadTits(ArtName)`.

Además de llamar información de título, autor, director, línea de investigación y la imagen de cada proyecto, estamos llamando una dirección URL de la página correspondiente a cada proyecto. Para que el vínculo se active hace falta entonces tener un botón que tenga la acción asociada:

```
on (release) {  
    getURL(vinculo, "_blank");  
}
```

Teniendo en cuenta que en la función `LoadTits(propiedad, valor)` se definió el valor de la variable `vinculo = getChildValue(arrayhijos[i], "pagina");` O sea que estamos direccionando el botón a un URL donde está publicada la página del proyecto respectivo según se define en el documento [ejemplos/proyectos.xml](#) pero en una ventana nueva del navegador para no perder la interfaz de consulta.

Veamos también que la clasificación por línea de investigación es diferente a la de si son proyectos en curso de estudiantes o si son proyectos terminados de egresados, ambas están presentes simultáneamente así como la opción de construir la lista por título o por nombre para ambas maneras de clasificar. En un caso la acción es:

```
GetTitulos("tema", categoria3);
```

mientras que en el otro es:

```
GetTitulos("categoria", categoria1);
```

Esto es debido a que la función está dada en términos de (*propiedad, valor*) y por lo tanto podemos obtener simultáneamente clasificaciones de acuerdo a las diferentes posibilidades de **propiedad** que estén definidas en el documento XML.

En cuanto a la imagen que carga para ilustrar cada proyecto, tenemos unos pocos casos en que la imagen es una animación hecha en Flash en lugar del archivo en formato .jpg. En el documento `llamarXML fla` no importa si estamos refiriéndonos a un archivo **.swf** o a uno **.jpg** para que se ejecute la acción:

```
limage = getChildValue(arrayhijos[i], "imagen");  
loadMovie(limage, "foto");
```

Para efectos de este ejercicio se han colocado los cinco ejemplos en la película `llamarXML fla`, es importante aclarar que para poder hacer esto se han agregado unas acciones cuyo fin es evitar que un ejemplo se mezcle con otro. Principalmente se trata de la función `reset()`; que borra el valor de todas las variables. Para el que desee trabajar editando los ejemplos lo mejor es que seleccione el frame o cuadro del ejemplo seleccionado y lo copie completo en un nuevo documento, de esta manera evita los inconvenientes que se puedan presentar por esta causa.

10. BIBLIOGRAFIA

MANOVICH, Lev . " The Language of New Media", Cambridge, ms. MIT Press, 2000

BARLOW, J. Perry. "Vender vino sin botellas" en *El Paseante*, No.27-28 (s.f.): 'La revolución digital y sus dilemas', pp. 10-22.

Entrevista con Jaron Lanier en *El Paseante*, No.27-28 (s.f.): 'La revolución digital y sus dilemas' , pp. 79-80.

RINCÓN, Hernando. "La pantalla". Bogotá, U. Nacional de Colombia, 2001

CALABRESE, Omar. "La era neobarroca". Madrid, Cátedra, 1989

VEEN, Jeffrey. "Arte y Ciencia del diseño web", Madrid, Prentice Hall, 2001

NIELSEN, Jacob. "Usabilidad. Diseño de sitios web", Madrid, Prentice Hall, 2000

VAUGHAN, Tay. "Multimedia, Manual de referencia", McGraw Hill - Osborne Media, 2002

MACROMEDIA, "Utilización de Flash MX", Macromedia Inc., marzo 2002

GOLDFARB, Charles y PRESCOD, Paul. "Manual de XML", Madrid, Prentice Hall, 1999

Real Academia Española, "Diccionario de la lengua española" 22 edición, 2001

11. REFERENTES EN INTERNET

Los referentes en la red son ejemplos de diferentes soluciones al problema de grandes volúmenes de información resueltas siguiendo el modelo estudiado, aunque no necesariamente con la misma tecnología. El modelo se refiere a un sistema de organizar la información independiente del sistema de representación y no al empleo de Flash, aunque en varios sitios utilicen este programa.

<http://www.facartes.unal.edu.co>

En este sitio estan puestos a prueba los ejemplos. En varios casos los documentos XML son generados dinámicamente a partir de la consulta en una base de datos y en otros están escritos directamente. La metodología de trabajo empleada en este proyecto sirvió de base para el presente documento.

<http://www.sydneyoperahouse.com>

En este sitio es particularmente interesante el tour virtual.

<http://www.cocacola.com>

Presenta una gráfica muy impactante y acceso a una gran cantidad de páginas que deben ser actualizadas en diferentes paises.

<http://www.manovich.net>

Proyectos, textos y vínculos a gran cantidad de información relevante.

<http://www.surfstation.lu>

Un sistema muy claro de acceso a toda clase de opciones. Propone cambios importantes en la temporalidad del formato entrevista.

<http://www.entropy8.com>

Pone en duda muchos conceptos de diseño y se sostiene después de varios años.

<http://www.thecooker.com>

Revista web con links a proyectos y portafolios presentados con mucho humor.

<http://www.hotwired.com>

La versión web de la conocidísima revista WIRED

<http://www.onyro.com>

Una gráfica muy contemporánea sobre temas de diseño.

<http://www.discovery.com>

Ofrece acceso a la programación de 6 canales de televisión y temas relacionados con cada uno. Codifica por color para cada canal manteniendo los mismos modos de organizar y presentar información.

<http://www.zonezero.com>

Revista de fotografía liderada por Pedro Meyer.

<http://www.benetton.com>

Excelente gráfica sin necesidad de usar el esquema de cuatro columnas y la navegación a la izquierda.

<http://www.cartoonnetwork.com>

Como corresponde, presenta la gráfica característica del canal para niños.

<http://www.turntables.de>

interactividad con música

<http://www.moebio.com>

Una buena muestra de las posibilidades de Flash para crear interactividad. Ofrece acceso al código fuente de todos los ejemplos.

<http://www.disenovisual.com>

La página de los profesores de diseño visual de la Universidad de Caldas, temas de actualidad y vínculos a muchas otras.

<http://www.hotmail.com> y <http://www.yahoo.com>, como casi todos los portales de correo gratuito, sirven de ejemplo al problema. La solución tiende a ser bastante parecida en casi todos debido a la necesidad de captar grandes volúmenes de usuarios y al servicio de correo electrónico que sirve de gancho para atraer a los usuarios.