

COMUNICACIÓN EFICIENTE EN *WORLD WIDE WEB*:

# Guía para la *hipercomunicación*

## Raymond Colle

Profesor de la Escuela de Periodismo, analista de sistemas y editor de las páginas *Web* del Servicio de Computación de la Universidad Católica. [rcolle@puc.cl]

En este artículo se incluyen consideraciones acerca de las características que debería asumir un medio periodístico que pretenda operar a través de *World Wide Web* y se señalan algunos principios y consejos sobre estructuración de proyectos de *Web*, creación de documentos y facilitación de la lectura (*navegación*). Así, se espera contribuir a que se construyan sistemas eficientes en su objetivo comunicacional y que sean, además, fáciles de mantener.

**L**a explosiva expansión de *World Wide Web* y su uso creciente para todo tipo de propósito comunicacional no sólo conlleva altos beneficios para quienes pueden aprovechar lo que se ha transformado en un nuevo medio de comunicación masiva. Éste demanda, además, una actitud activa del receptor, quien elige más puntual y específicamente los contenidos que le interesan y puede desplazarse (*navegar*) a su gusto de una *página* a otra, e incluso pasar muy rápidamente de un emisor a otro, un poco a la manera del *zapping* en televisión (aquí conocido como *surfing*).

Paralelamente, es tarea del emisor asegurar una comunicación expedita y facilitar la *navegación*, es decir la creación por cada lector de una secuencia de lectura que le resulte atractiva y útil.

En este artículo nos preguntaremos acerca de las características que debería asumir un medio periodístico que pretenda operar a través de este nuevo canal de comunicación y señalaremos algunos principios y consejos destinados a los creadores de servicios *Web*. De este modo esperamos contribuir a que se construyan sistemas que aseguren el logro de su objetivo: eficiencia



en la comunicación, cosa especialmente relevante en medios periodísticos. Asimismo, se proponen soluciones para que estos sistemas sean fáciles de mantener.

## 1. NAVEGAR POR LA NOTICIA

El interés de las empresas periodísticas por la comunicación a través de *World Wide Web* se hizo patente muy rápidamente, aunque –en apariencia– sin tener una visión clara de qué información podrían entregar y cómo. En la mayoría de los casos, hasta ahora, los medios presentes en *WWW* nos ofrecen titulares o resúmenes de noticias escogidas.

¿Qué es lo que puede hacer un medio de prensa en este nuevo entorno? Lo primero es darse cuenta de que no se trata de un simple canal donde colocar información de la misma forma que en un diario o una revista, a pesar de que el componente verbal (escrito) parece ser –como en éstos– el más relevante. Estamos ante un *medio de*

*comunicación* totalmente nuevo, y la forma de hacer las cosas ha de replantearse completamente, del mismo modo en que debió hacerse cuando se pasó de la radio a la televisión.

### 1.1 CARACTERÍSTICAS NUEVAS

Esto implica considerar de modo diferente las *relaciones entre los recursos de expresión*: imagen, texto y sonido pueden ser combinados y de una manera totalmente nueva. Es lo propio y típico de los llamados *multimedios* computacionales. Y, en el caso de la imagen, no ha de olvidarse el aspecto dinámico de la misma: en efecto, las fotografías pueden ser reemplazadas por breves secuencias de video, aunque sea entremedio de un texto que se leerá en pantalla. El sonido puede ser acoplado a estas secuencias o ser transmitido en forma independiente.

También implica tener en cuenta que *el destinatario tiene una injerencia mucho*

*mayor*, mucho más selectiva, en la recepción de la información: quiere (y debe poder) elegir sólo las noticias que le interesen; decide mirar o no las ilustraciones, escuchar o no las secuencias sonoras. Y, sobre todo, ya acostumbrado a los principios de la navegación, quiere poder ampliar sus conocimientos de los hechos remontándose a los antecedentes o buscando otros hechos semejantes. La crónica típica del diario impreso –un texto estático, generalmente desvinculado de cualquier otro– no tiene lugar aquí. Hacen falta a la vez algo ágil (más cercano a la forma de los cables de agencias noticiosas) y algo modular, con un gran *cuidado en la adición y gestión de vínculos (links)* que han de unir un texto con uno o varios otros.

### 1.2 HACIA UNA ESTRUCTURA INFORMATIVA ESPECÍFICA

¿Cómo concebir una estructura que cumpla con la riqueza del nuevo recurso



## na página inicial (*index*) podría presentar las principales secciones, a modo de tabla de contenido, como lo haría una revista. También debería contener un *botón* que permita activar un motor de búsqueda.

tecnológico que se nos ofrece? Intentaremos formular aquí algunas sugerencias que surgen de nuestra experiencia como editores de Web y de nuestros estudios anteriores acerca de la estructura documental de la información periodística (ver bibliografía).

Una *página inicial (index)* podría presentar las principales secciones, a modo de tabla de contenido, como lo haría una revista. También debería contener un *botón* que permita activar un *motor de búsqueda (search)* que opere sobre todos los datos disponibles y permita al lector encontrar lo que haya sobre algún tema en particular, eventualmente con límites de tiempo (fechas), sin necesidad de revisar lo que haya en cada sección ni de navegar por múltiples páginas.

Luego, para cada *sección* se podría pasar a una página con los principales titulares de las noticias del día (o del período cubierto), eventualmente (pero no necesariamente) acompañadas de un resumen (*lead*). Los titulares serían a su vez *anclas* que, activadas, pasarían a la *página con el detalle* (verbo-icónico-sonoro) de la información.

Y aquí es donde deberían darse las principales diferencias con los medios tradicionales: si aparece el nombre de una persona (un político destacado, por ejemplo), éste debería operar como *ancla* y remitir al lector —si lo deseara— a una página con antecedentes acerca de esta persona. También debería haber un *ancla* que remita

al hecho noticioso anterior, de tal manera que se pueda remontar en el tiempo y entender la secuencia de los hechos. Si se trata de procesos políticos, se podría remitir a una breve historia o caracterización de la situación de un país (es decir enlazar el trabajo periodístico de actualidad con el interpretativo). Si se usan términos técnicos, se puede remitir a un diccionario, etc.

Recordemos que los archivos que componen un sistema documental periodístico son típicamente: crónica, biografías, instituciones, países (sistema político, estructuras ejecutiva y legislativa, etc.), fotografías, enciclopedia (definiciones, explicaciones de objetos, técnicas, etc., que no estén en enciclopedias publicadas), bibliografía, estadísticas. Todos éstos pueden mantenerse en bases de datos que pueden ser *interfaciadas* con WWW, como las de Oracle. Permitir la navegación desde un registro de una base de datos a otro de otra base puede ser hoy algo difícil (se ha de volver, típicamente, a una página normal de Web o a la interfaz de consulta), pero la velocidad con que se están desarrollando nuevas aplicaciones y soluciones orientadas a perfeccionar los servicios por WWW auguran pronta solución para ésta y otras dificultades (como la ausencia de caracteres tipográficos no ingleses en Oracle, algo más crítico para nosotros pero cuya solución está anunciada para 1998).

En resumen: un servicio noticioso *hipermedial* efectivo habría de construirse —

a nuestro juicio— sobre un conjunto de bases de datos, las páginas de títulos y las noticias del día, siendo las interfaces principales con los receptores y los medios para que éstos puedan navegar y obtener los conocimientos que buscan en el campo de la actualidad periodística.

Descritas de este modo las características y la estructura teórica del nuevo medio, conviene considerar ahora algunas reglas para optimizar el desarrollo práctico del proyecto y la posterior navegación de los clientes.

## 2. LA ESTRUCTURACIÓN INFORMÁTICA DEL PROYECTO DE WEB

Un proyecto de Web se presenta generalmente como un conjunto de páginas de *hipertexto* (Html) que conforman lo que llamaremos *módulo institucional*. Se tiende frecuentemente a llamarlo *Home Page*, pero, en sentido estricto, ésta es sólo la portada (con la identificación del organismo y el menú de los contenidos informativos o servicios ofrecidos). Este módulo puede ser el único instalado en el servidor de red de la institución o estar instalado junto a otros en el servidor de algún proveedor externo.

Es conveniente saber y tener en cuenta que los documentos básicos, en torno a los cuales se construye todo el sistema, son los que contienen los *textos* que se leerán en pantalla. Todos los otros recursos (fotos,

videos, secuencias de audio) van a integrarse y transmitirse a partir de instrucciones que están contenidas en estas páginas. El texto, por lo tanto, es la columna vertebral de un módulo de *Web*.

El módulo institucional contendrá toda la información que se desea hacer circular, dividida en *capítulos* (todas las entradas de primer nivel jerárquico a las cuales se llega desde la portada). Los capítulos se dividen eventualmente en *subcapítulos* y finalmente en *páginas*, cada una de las cuales constituye un documento de texto Html (con ilustraciones anexas). Las páginas se subdividen a su vez en *segmentos*, que son las distintas partes de las mismas a las cuales se puede acceder desde un menú al inicio de la misma página (señalados, en Html, por <A NAME=...>).

La estructura lógica es por lo tanto: módulo/capítulo/(subcapítulo)/página/segmento. Esta estructura debe quedar reflejada en la agrupación de los documentos en directorios y subdirectorios en el computador. Así, habrá un directorio general que incluirá todo el material correspondiente al módulo, y dicho directorio contendrá subdirectorios con recursos comunes (como los íconos de los botones de

navegación, logotipos, etc.) y subdirectorios que agruparán las páginas de un mismo capítulo o subcapítulo.

Llamaremos *dir\_1* el directorio principal (en la realidad podrá llamarse *www* o *data\_docs*, dependiendo del *software* del servidor). Deberá contener la portada (llamada *index.html*, de acuerdo con las normas) y los subdirectorios. Un primer subdirectorio, que nombraremos aquí *dir\_11*, contendrá los archivos de uso común, como las ilustraciones –íconos, botones, viñetas, etc.– que se repetirán en múltiples páginas. Luego se agregarán nuevos subdirectorios, conforme con las principales divisiones estructurales y de contenido.

Para visualizar mejor esta estructura, representaremos (GRÁFICO 1) la agrupación de las páginas (áreas grises) en directorios (rectángulos numerados) tal como el programador las agrupará (cosa habitualmente desconocida por el cliente-lector). Las líneas grises horizontales indican la secuencia básica de lectura. La letra *I* indica que el documento es un índice, es decir una página con un menú, que da acceso a páginas de los subdirectorios contenidos en ese mismo directorio.

Un directorio puede contener páginas

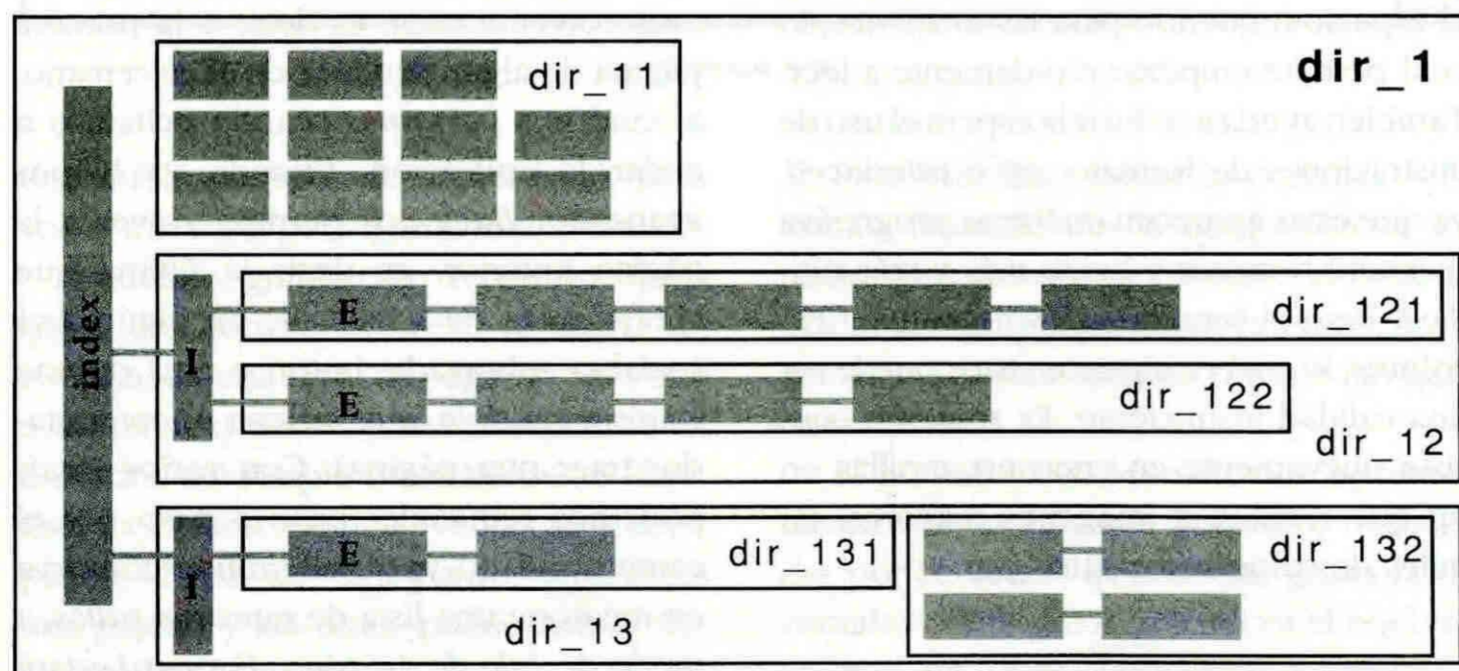
unidas en secuencia (para ser leídas una tras otra) como en el caso de los *dir\_121*, *122* y *131* (donde la letra *E* indica la página que encabeza la secuencia). Pero también puede contener páginas o archivos *paralelos*, como en el caso del *dir\_11*, o una mezcla secuencial/paralela como en el *dir\_132*. Y, como en estos casos, el acceso podría ser solamente a partir de *anclas* (*anchor*) a las cuales quedan vinculadas (*link*) desde otro archivo, y no desde un menú (un *ancla* es un objeto marcado que activa el *link* de navegación que lleva a otro documento).

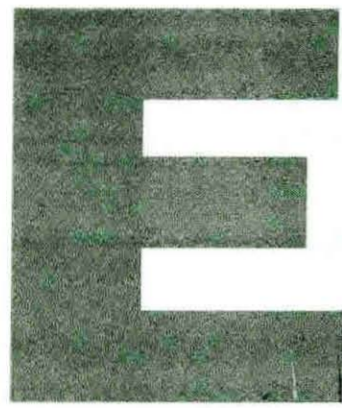
Así, en términos generales, conviene agrupar los archivos *de acuerdo con los contenidos* (capítulos y subcapítulos), pero también *de acuerdo con la naturaleza de los documentos*: los textos juntos en un subdirectorio, las fotos en otro, el audio en un tercero. Esto facilita la mantención, es decir la sustitución de los documentos, sin afectar las referencias de unos a otros (*hyperlinks*). Esto explica, por ejemplo, la estructura del *dir\_13*, donde un subdirectorio se reservó al texto y otro a las fotografías:

- en el *dir\_131* tenemos dos páginas de texto, en secuencia, mientras
- en el *dir\_132* tenemos dos pares de fotos, que son las requeridas por las páginas del *dir\_131*: la primera imagen de cada par es una *estampilla*, es decir una reproducción de tamaño reducido que aparecerá en una de las páginas de texto y que podrá ser activada como *link* hacia la segunda foto, la cual es una reproducción de mayor tamaño, que el lector recibirá como documento separado sólo si lo desea (justificaremos este proceder más adelante).

Dedicaremos así un directorio (del tipo *dir\_13*) a las noticias del momento (crónica), otro a las biografías, otro a los antecedentes geográficos, etc., según el tipo de servicio que podemos ofrecer y las bases de datos que podemos dejar vinculadas.

GRÁFICO N°1





**Es cierto que la velocidad de recepción depende de la capacidad de las redes, lo cual no puede ser controlado por el emisor. Pero sí es posible aumentar –o reducir– la velocidad controlando la estructura de cada página.**

### 3. LA CREACIÓN DE LOS DOCUMENTOS

Al crear los documentos, conviene asegurarse de que una página no demorará demasiado en cargarse en el computador del lector, lo cual dependerá tanto de la longitud del texto como –sobre todo– de la cantidad de ilustraciones (visuales o auditivas) que incluya. Es cierto que la velocidad de recepción depende de la capacidad de las redes (ancho de banda dividido por el número de usuarios simultáneos), lo cual no puede ser controlado por el emisor-programador. Pero sí es posible aumentar –o reducir– la velocidad controlando la estructura de cada página. Si el texto es breve y las ilustraciones pocas (y pequeñas), la transmisión será mucho más rápida. Pero la comunicación en su conjunto puede tornarse ineficiente (y enojosa) si se fragmenta demasiado el contenido, obligando al lector a activar *link* tras *link*, lo cual puede tener como resultado que ocupe más tiempo esperando la llegada de la información que leyéndola.

Conocemos ejemplos en ambos extremos: páginas que demoran cinco o más minutos en llegar debido a su extensión e ilustraciones, y páginas que sólo contienen un párrafo de texto y obligan a pedir otra en seguida. En este caso, sólo pueden recibir las páginas en forma rápida quienes estén conectados localmente al mismo servidor WWW (en el mismo *dominio*). Pero quienes estén alejados se aburrirán de esperar.

¿Cuál sería el adecuado término medio? Nuestra experiencia aconseja trabajar con documentos de una extensión del orden de 10 a 20Kb. Y en una página de texto de este tamaño, se han de evitar más de tres ilustraciones fijas –como las fotos– o una secuencia (audio o video), si bien podría haber además algunos pequeños *botones* (íconos destinados a facilitar la navegación). Así, si queremos agregar fotos u otras ilustraciones, conviene utilizar el sistema de *estampillas*, que consiste en insertar imágenes fijas pequeñas (de 200 píxeles por lado como máximo) que sean anclas de las ilustraciones de plena página o en secuencia que se quieran ofrecer.

Aumenta la velocidad de recepción de una página indicar en su programación el tamaño exacto de cada ilustración (<IMG WIDTH=value HEIGHT=value SRC...>): de este modo el texto será exhibido *saltando* el espacio requerido para las imágenes, lo cual permite empezar rápidamente a leer. También ayuda a reducir la espera el uso de ilustraciones de formato *.gif* e *interlaced*, ya que éstas aparecen en forma progresiva (primero borrosas y luego más y más nítidas). Pero el formato *.gif* sólo admite 256 colores, lo cual en algunos casos puede ser una calidad insuficiente. La solución consiste nuevamente en crear estampillas en *gif* que remitan a imágenes mayores, en miles de colores (formato *.jpg*).

### 4. CÓMO FACILITAR LA LECTURA (NAVEGACIÓN)

Quien haya navegado con cierta frecuencia por algún *hipermedio* habrá descubierto que no es siempre fácil descubrir *dónde está* ni menos aun volver con facilidad a alguna de las posiciones anteriores. Consideraré aquí los recursos de que dispone normalmente un *navegante* de *Web* y cómo puede un programador ayudarle a encontrar su camino... o a perderse más irremediabilmente.

#### 4.1 LAS COMPLEJIDADES DE LA NAVEGACIÓN

Todos los *browsers* (aplicaciones de usuarios) disponen al menos de dos comandos básicos: *Back* y *Home*, acompañados a veces de *Forward*, como en el *Netscape*.

*Home* nos asegura que siempre podremos volver *a casa*, es decir a la primera página de algún servidor de *Web* cercano, al cual nos *colgamos* cuando echamos a andar la aplicación. Cuando ya hemos avanzado, *Back* nos permite volver a la página anterior, es decir la última que vimos antes de hacer clic en algún *link* (palabra subrayada, botón o área de una *imagen sensible*, que ordena al computador traer otra página). Con varios *Back*, podemos retroceder paso a paso por el camino andado, ya que el *browser* conserva en memoria una lista de nuestros *saltos*, a modo de *hilo de Ariadna*. *Forward* estará

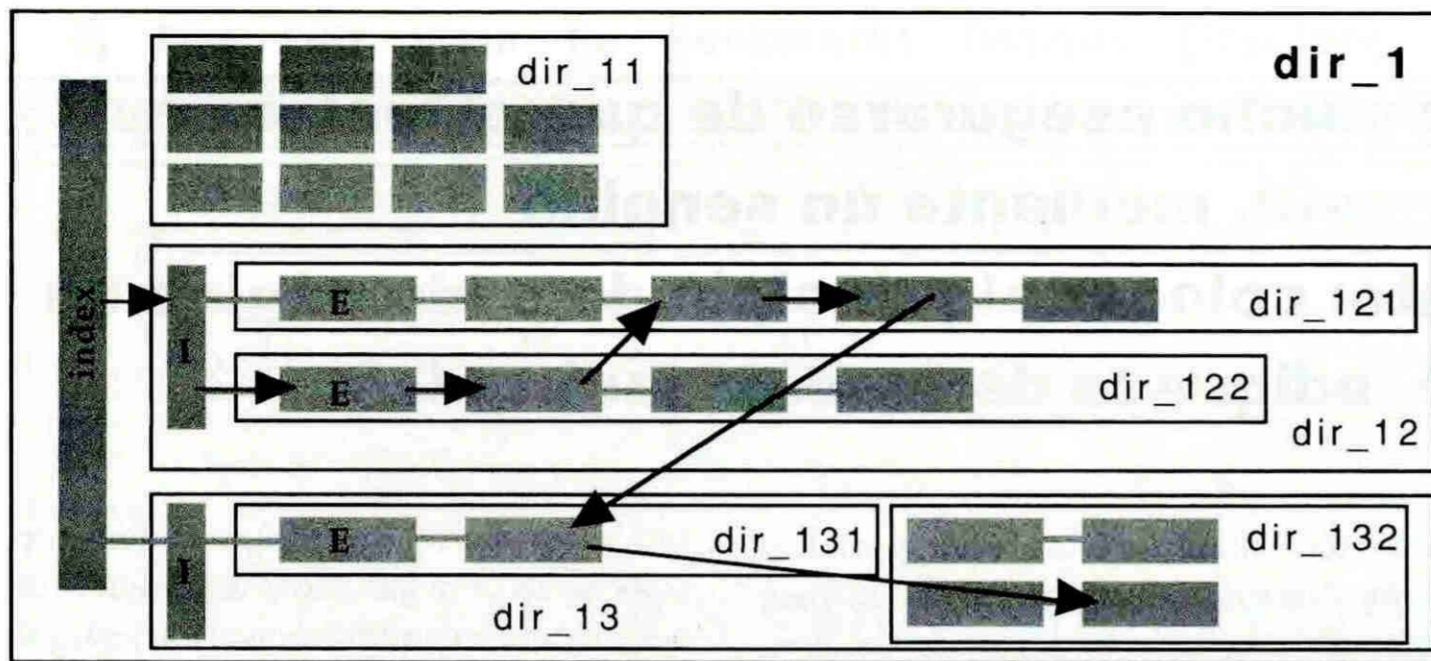


GRÁFICO N°2

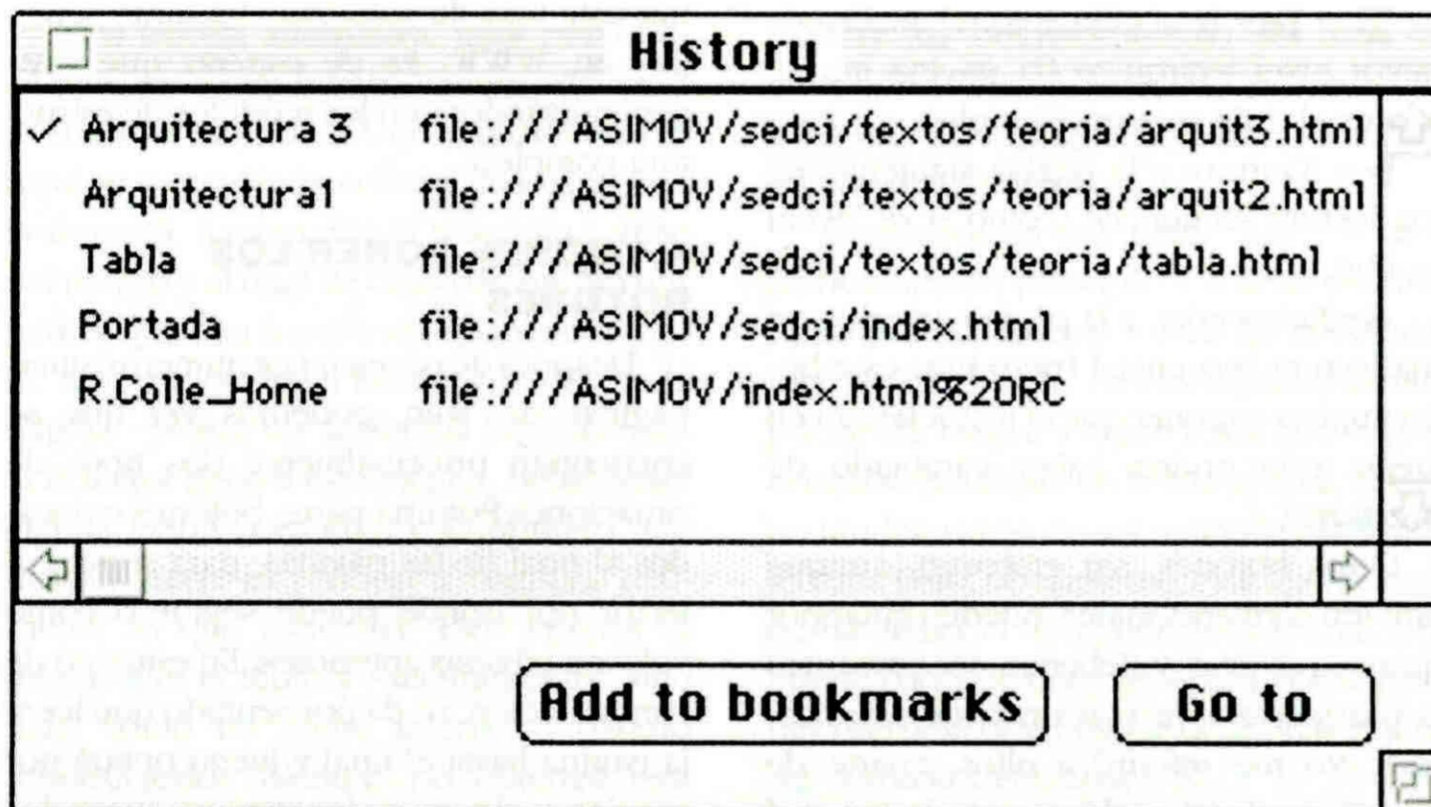


GRÁFICO N°3

disponible después de algún *Back*, para volver a avanzar por un camino ya recorrido.

Podemos graficar un recorrido típico como una lectura (o breve revisión) secuencial de una serie de páginas. Sólo que estas páginas no están necesariamente en un orden tan simple como las de un libro. Esto es lo característico –ventajoso y complejo a la vez– de los *hiperarchivos*. Entre una página y las otras, puede haber muchos caminos posibles.

Para visualizar mejor lo complejo del asunto, retomaremos la agrupación de las páginas en directorios tal como el programador las agrupó, pero agregaremos ahora líneas indicadoras de una posible navegación (GRÁFICO 2).

Si no existiesen los vínculos *hipermediales*, sólo podríamos avanzar y retroceder por las líneas básicas (las grises horizontales del gráfico), lo cual es el equivalente a un libro. Pero aquí tenemos –

generalmente– múltiples enlaces entre una serie y otra. Sobre el esquema básico colocamos ahora flechas negras para señalar el recorrido de un navegante casual (que no habrá usado el comando *Back*). Cada flecha corresponde aquí a un clic en un link que efectúa un cambio de página

#### 4.2 DÓNDE FALLA EL HILO DE ARIADNA

Si el navegante se pregunta dónde se encuentra cuando llega a la última página señalada por las flechas, por cierto no tendrá punto de referencia para saberlo. Habría que ser experto y tener la intención de «reconstruir» la estructura de directorios para saber por cuántos de ellos se puede haber pasado. Incluso sería muy posible estar conectado en ese momento con otro servidor de *Web*, aun ubicado en otro país.

Y si en este instante el lector desea volver a la segunda página del directorio 122, donde dejó la secuencia básica para dar *hipersaltos*, ¿deberá retroceder cuatro pasos (con *Back*)? ¿Y si dio un número mucho mayor de saltos y desea volver atrás? Hemos de saber que los *browsers* mantienen una «historia» de la secuencia de saltos. Esto se puede ver, por ejemplo, en el menú *Go...* y, más detalladamente, en la opción *View History* de *Netscape*. (GRÁFICO 3)

Pero la lista de *Go...* o *History* no permanece completa. Así, por ejemplo, si leímos algunas páginas del directorio 122 y luego volvemos al índice del cual depende (que encabeza el directorio 12) para entrar luego a la serie del directorio 131, nuestro paso por el 122 desaparecerá y la lista se volverá a construir a partir del índice 13 hacia el 131. Del mismo modo, si retrocedemos luego hasta el 1 y pasamos al 12 y al 121, ya no tendremos a la vista en el menú *Go...* las *bajadas* anteriores al 13. Esto es una característica de la estructura jerárquica en forma de árbol: cuando se

# Puede ayudar mucho asegurarse de que el lector sepa en qué lugar está, mediante un sencillo procedimiento: colocar al principio de cada página un epígrafe que indique la derivación jerárquica.

vuelve por una rama hacia el tronco y se elige otra rama, no queda rastro de la rama anterior. Aquí es donde falla el *bilo de Ariadna*. Pero al contrario, si hemos *saltado* de una rama a otra (sin volver al nodo en que se une a otras), podremos aún ver los puntos de *salto*.

Si bien esta dificultad sólo puede ser superada gracias al conocimiento de la misma y a cierta disciplina del navegante, tiene consecuencias importantes para el programador en relación con la inclusión de botones para navegar. Especial cuidado requieren los que ofrecen formas de retroceder en la lectura.

## 4.3 BOTONES DE NAVEGACIÓN

Tanto para *alegrar* gráficamente una página como para facilitar la navegación, es usual que se inserten botones especiales o íconos de navegación en las páginas. Éstos pueden estar en distintos lugares, frecuentemente debajo del título cuando son opciones de tipo *menú* o al final de las páginas cuando sugieren lecturas complementarias. Generalmente es ahí, al final de las páginas, que encontramos también botones para avanzar o retroceder longitudinalmente (es decir como en un libro).

Para que al lector le sea relativamente fácil entender a dónde irá al activar el botón-*link*, es indispensable que el autor haya previsto la dificultad. Esto significa usar un símbolo monosémico (esto es interpretable de una sola manera) o incluir un

texto que elimine toda confusión posible).

Mostramos aquí las convenciones más comunes que, salvo error garrafal del programador, no deberían prestarse a una mala interpretación:

▲ ó ◀ = Ir al principio: al *índice* de mayor nivel jerárquico (la página inicial, esto es el 1 de nuestro ejemplo).

▶ = Avanzar: a la página siguiente, en una lectura secuencial (como si se leyera un libro).

◀ = Retroceder: a la página anterior, en una lectura secuencial (pero usar este botón implica suponer que el lector tendrá en cuenta que podría haber cambiado de directorio).

Otros botones, sin embargo, aunque también convencionales, pueden guiar por pistas equívocas y deberían ser proscritos (o usarse siempre con un texto clarificador). No me referiré a ellos, aparte de advertir aquí del problema: un botón mal diseñado o de múltiple interpretación puede hacer que el lector derive –sin saber por qué ni cómo– hacia el encabezamiento de un directorio que, en sí, no tiene relación obvia con lo que estaba revisando.

Un medio periodístico debería recurrir mucho más a la navegación a partir de palabras marcadas (anclas) que a íconos que puedan sugerir cierta *secuencialidad*, por lo cual este tipo de problema no debería presentarse.

Sin embargo, se puede ayudar mucho asegurando que el lector sepa en qué lugar

está mediante un sencillo procedimiento: colocar al principio de cada página un epígrafe que indique la derivación jerárquica, como en el ejemplo adjunto (GRÁFICO 4). Pese a su utilidad, es poco frecuente encontrar este tipo de solución –bastante sencilla– en WWW. Es de esperar que vaya generalizándose en los módulos de estructura compleja.

## 4.4 DÓNDE PONER LOS BOTONES

Después de navegar por numerosísimas páginas de *Web*, podemos ver que se encuentran principalmente dos tipos de situaciones: Por una parte, botones colocados al final de las páginas, para sugerir al lector por dónde puede seguir o cómo volver a páginas anteriores. En este tipo de configuración, se da por sentado que leerá la página hasta el final y luego optará por seguir por alguno de los caminos sugeridos (o usar alguno de los comandos del *browser*). También se estima que si no deseaba leer la página, después de ver su título, hará un *Back* o un *Go...*

Otra situación tipo que tiene cierta frecuencia es la repetición de los botones del menú inicial (iguales o de tamaño menor) debajo del título. Aquí se ofrece al lector un conjunto de opciones para desviarse si no desea leer la página. Eventualmente la misma serie se repite al final para que, si leyó todo, no deba volver al principio para seguir navegando.

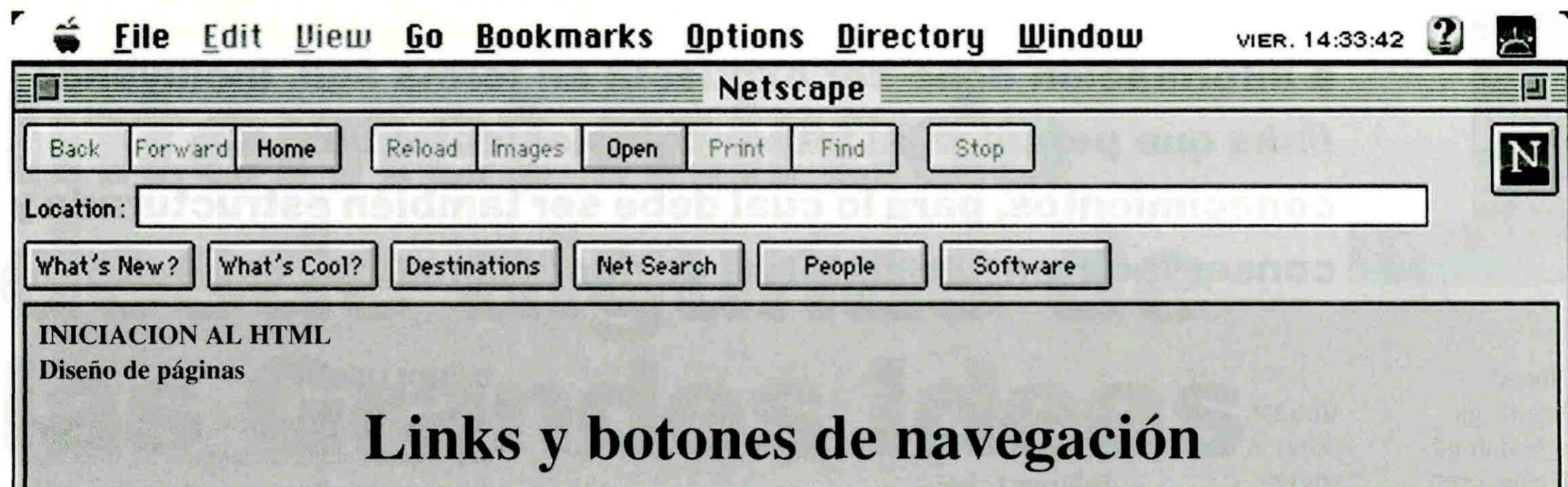


GRÁFICO N°4

Una tercera alternativa, muy poco frecuente, es la que hemos encontrado en el *Manual de Estilo* de la Universidad de Yale. Aquí se recomienda colocar una barra con botones de navegación al principio (¡antes del título!) y al final de cada página. No nos parece que esta sugerencia –poner botones de navegación en el borde superior de la página– pueda ser sostenida lógicamente. Si el lector llegó a esta página, debió tener alguna razón para ello, y lo primero que deseará hacer es verificar si contiene realmente lo que esperaba. Esto implica al menos leer el título y –seguramente– algo más, como un menú interno o párrafo introductorio. Mientras no obtenga esta información, no navegará, a menos que se dé cuenta de que se había equivocado, en cuyo caso usará el *Back* y no necesitará otros botones.

En consecuencia, lo más indicado sería:

- Si la página es corta: colocar los botones de navegación ya debajo del área de titulación, ya al final.
- Si la página tiene mayor extensión:

colocar los botones debajo del área de titulación (debajo del menú o índice del contenido de la página, si lo hay) y también al final de la página.

Si estamos en este último caso, la serie de botones del principio y la serie del final serán probablemente –pero no necesariamente– las mismas. En efecto, el contenido puede dar origen a algún tema nuevo para el cual se justifique un botón al final, el cual no tendría razón de ser antes de realizar la lectura. Pero, obviamente, este tipo de caso no ha de ser el más generalizado. Así, una buena barra de botones –para colocar debajo del título y al final de la página– podría ser como la del GRÁFICO 5.

### 5. DOCUMENTACIÓN

Como en todo proyecto de computación, es muy recomendable documentar adecuadamente los directorios y sus contenidos. Como mínimo, debería anotarse:

- La estructura jerárquica (directorios y nombres de los documentos contenidos en cada uno).

- Las anclas y los *links* estructurales (puntos de origen y de destino); con nombre HREF y significado; por ejemplo,

/dir\_121/: (Computación)

*Index*

<HREF = «ccien.html»> = Computación científica

<HREF = «sdistr.html»> = Sistema distribuido

*Páginas:*

<HREF = «ccien.html»> = Computación científica

Segmentos:

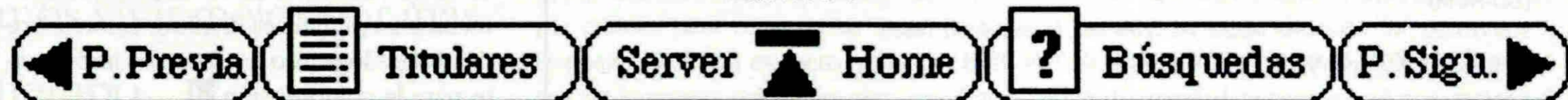
<NAME=«objetivos»> = Objetivos

<NAME=«recursos»> = Recursos Computacionales

etc.

- Para las ilustraciones: hacer una lista de los nombres codificados (f\_xxx.jpg) junto con la indicación de lo que representan (y es conveniente anotar el tamaño de cada una, ya que la definición de tamaño también acelera la recepción), por ejemplo:

GRÁFICO N°5





**L**a información debe ser expuesta en forma ágil, incluyendo *links* que permitan al lector complementar sus conocimientos, para lo cual debe ser también estructurada y conservada en un sistema de fácil consulta, como *WWW*.

/fotos/

board1.gif	402x236	«Motherboard»
e_mainfr.gif	200x136	Computador mainframe («e_» indica estampilla)
f_babbag.gif	90x128	Babbage (cara)
f_chip.gif	120x158	Chip
f_mainfr.jpg	400x272	Computador mainframe (foto)

## BIBLIOGRAFÍA



- COLLE, RAYMOND: «El archivo inteligente», en *Cuadernos de Información* N°9, 1994, pp. 37-41.  
 —: *Documentación periodística: Principios y aplicaciones*, Ed. Universidad Católica, Santiago, 1992.  
 HERELLIER, J.M.: *Le multimédia*, Sybex, Paris, 1994.  
 SHANK, R. & CLEARY, CH.: *Engines for Education*, Institute for the Learning Science, Evanston, 1994 (CD-ROM).  
 VON WODTKE, M.: *Mind over Media*, McGraw-Hill, Nueva York, 1993.  
*Yale C/AIM WWW Style Manual*, Universidad de Yale, 1994.

## CONCLUSIÓN

De todo lo anterior se pueden desprender algunas sugerencias básicas, importantes pero sencillas, especialmente para los programadores de módulos periodísticos:

- La información noticiosa debe ser expuesta en forma ágil, incluyendo *links* que permitan al lector complementar sus conocimientos, para lo cual debe ser también estructurada y conservada en un sistema especialmente diseñado de fácil consulta posterior, como las bases de datos con interfaz para *WWW*.

- Conviene controlar estrictamente la dimensión de los documentos, especialmente de las páginas de texto (que contienen los comandos Html).

- Conviene agrupar los documentos de acuerdo con su contenido (temas) y su naturaleza (texto, imágenes, sonidos).

- En estructuras simples, para la navegación son suficientes los comandos del *browser*. Es preferible no agregar botones, excepto —eventualmente— para hacer un menú inicial ilustrado.

- En estructuras más complejas, debe recordarse siempre que no se espera que el lector se dé cuenta del tipo y *amplitud* de los saltos que da al navegar; tampoco se espera que memorice su recorrido. El *browser* lo puede ayudar, pero esta ayuda es *muy* relativa. Por lo tanto, conviene recurrir a epígrafes si la información sobre divisiones lógicas o temáticas es importante para la comprensión. **CI**