



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA DE HIPERTEXTO

**TESIS PROFESIONAL
QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
JOSE RAUL VALDIVIA RIOS**

**DIRECTOR DE TESIS:
ING. ADOLFO MILLAN NAJERA**



MEXICO, D. F.

1993

**TESIS CON
MILLA DE CREDITO**



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE.

I.	Introducción	1
II.	Evolución y aplicaciones del Hipertexto	7
II.1	¿Por qué Hipertexto?	7
II.2	Evolución del Hipertexto	9
II.3	Aplicaciones	19
II.3.1	Aplicaciones en Computación	20
II.3.2	Aplicaciones generales	24
III.	Arquitectura de un Sistema de Hipertexto	30
III.1	Modelo HAM	30
III.2	Nodos, ligas, gráficas de información	32
III.2.1	Nodos	32
III.2.2	Ligas	35
III.2.3	Gráfica de información	37
III.3	Mecanismos de apoyo a la navegación	46
IV.	Diseño conceptual del Sistema de Hipertexto	58
IV.1	Consideraciones generales	58
IV.2	El Modelo Dexter de Hipertexto	59
IV.3	Descripción conceptual del Sistema	65
IV.4	Descripción funcional del Sistema	70
IV.4.1	Creación de documentos	72
IV.4.2	Perfil de un hiperdocumento	75
IV.4.3	Descripción detallada de los comandos	75
V.	Diseño detallado e implementación del Sistema de Hipertexto	85
V.1	Estructura interna del sistema	85
V.1.1	Representación interna de la red de Hipertexto	85
V.1.2	Criterios de creación, actualización y consulta	91
V.2	Estructura de módulos del sistema	94
V.3	Consideraciones de implementación	101
VI.	Caso de Estudio	102
VI.1	Metodología HDM para el diseño de hiperdocumentos	104
VI.2	Caso de Estudio	107
VI.2.1	Antecedentes	108
VI.2.2	Justificación de un sistema de hipertexto	111
VI.2.3	Diseño conceptual del hipertexto usando la metodología HDM	112
VI.2.4	Implementación del hipertexto	115
VI.3	Recomendaciones prácticas para el diseño de hiperdocumentos	120
VII.	Conclusiones	124
VII.1	Evaluación de la herramienta desarrollada	124
VII.2	Desarrollo futuro de la herramienta	126
VII.3	Evaluación del caso de estudio	128
VII.4	Conclusión	130
	Bibliografía	131

PROLOGO.

Sea en forma impresa o en archivos electrónicos almacenados en una computadora, el texto tradicional es secuencial, lo que significa que existe una secuencia lineal que determina el orden en el cual el texto puede ser leído. El hipertexto es la tecnología de escritura y lectura de texto en forma no secuencial a través de una computadora. Un documento creado bajo este concepto consiste en un conjunto de fragmentos de texto denominados nodos, relacionados entre sí a través de referencias llamadas ligas. Al proceso de viajar a través de esta red de información se le conoce como navegación, y al conjunto de nodos y ligas se le denomina hiperdocumento.

Así creado, un hiperdocumento no establece una secuencia fija para su consulta. Cada lector determina la trayectoria que ha de seguir para encontrar la información buscada, de acuerdo con su situación particular. Esto facilita enormemente el proceso de búsqueda y análisis de información de diversa índole: libros de consulta, manuales técnicos, procedimientos, cursos, etc.

El objetivo de esta tesis es desarrollar un sistema de hipertexto, programa de computadora que permita la creación y consulta de hiperdocumentos, proporcionando herramientas para la creación de nodos y ligas, y para la navegación a través de ellos.

En el primer capítulo, Introducción, se presentan los conceptos básicos relacionados con el hipertexto, con el objeto de entender el significado de esta nueva tecnología y los beneficios que aporta sobre el manejo tradicional de información textual.

En el capítulo II, Evolución y Aplicaciones del Hipertexto, hablaremos de los antecedentes históricos del hipertexto, así como la evolución de este concepto a través del tiempo, y como ha venido aplicándose paulatinamente en diversos ámbitos. Veremos en su oportunidad que el hipertexto puede ser aplicado en cualquier área del conocimiento humano, ya que en todas ellas se requiere manejar información.

En el capítulo III, Arquitectura de un sistema de hipertexto, se analiza un modelo que define la estructura general de un sistema de hipertexto. En términos generales, todo sistema de hipertexto puede ser definido como un conjunto de entidades de información asociadas a través de ciertas ligas, y manipuladas a través de diversas herramientas para hacer posible su consulta y actualización. Se detallan los componentes de un hiperdocumento: nodos, ligas, gráficas de información, y se plantean una serie de mecanismos de apoyo a la navegación basados en dichos componentes.

En el capítulo IV, Diseño conceptual del sistema de hipertexto, se establecen las consideraciones generales de diseño del sistema, en base a un modelo teórico conocido como el Modelo Dexter, y que consiste fundamentalmente de tres componentes o niveles: nivel de almacenamiento, nivel de componente y nivel de ejecución. Cada uno de ellos establecen una serie de criterios y recomendaciones para guiar la estructura y funcionamiento del sistema. Así, en base a ello se especifican las distintas funciones de que dispondrá el usuario y su interfase.

En el capítulo V, Diseño detallado del sistema de hipertexto, se analiza la estructura interna del sistema. Uno de los aspectos más importantes es la representación interna de la red de hipertexto a través de estructuras de datos

adecuadas que permitan lograr la flexibilidad necesaria durante la consulta y modificación de la información. Se discuten también los criterios para crear, modificar y consultar estas estructuras de acuerdo con las funciones que serán proporcionadas a los usuarios. Finalmente, se establece la estructura de módulos que componen al sistema.

En el capítulo VI, Ejemplo de aplicación, se presenta una metodología para la planeación y diseño general de hiperdocumentos, independiente de la herramienta que se utilice para desarrollarlos, y se documenta un caso de estudio del diseño e implementación de un hipertexto para la consulta del Plan de Beneficios para empleados de una empresa, utilizando esta metodología y la herramienta desarrollada en la presente tesis.

Finalmente, en el capítulo VII, Conclusiones, se evalúan los resultados del trabajo, y se plantean las bases para proyectos futuros de desarrollo de nuevas funciones y utilerías para el sistema de hipertexto.

I. INTRODUCCION.

En 1945, alguien concibió una idea: un sistema de almacenamiento y recuperación de información en línea que contuviera no sólo la literatura científica de la post-guerra, sino también notas personales, diagramas e incluso fotografías. La máquina, llamada 'memex' permitiría a un usuario establecer ligas que asociaran la información contenida en dos puntos cualesquiera de esta gran biblioteca. De esta forma, sería posible 'navegar' a través de ella.

Vannevar Bush, el asesor científico del presidente norteamericano Roosevelt y precursor de toda la investigación de la época de guerra, incluyendo el Proyecto Manhattan, fue quien imaginó este mecanismo, si bien nunca lo creó. Pues bien, este proyecto inconcluso se convirtió en el punto de partida para desarrollar el concepto de hipertexto como se le conoce en la actualidad.

El hipertexto, en su forma básica, es una especie de sistema manejador de bases de datos (DBMS por sus siglas en inglés) que permite conectar diferentes entidades de información a través de ligas asociativas. Como resultado de esta habilidad, un sistema de hipertexto representa, en una forma más compleja, un ambiente de procesamiento de información, para trabajo de grupos, comunicaciones y adquisición de conocimiento. Los sistemas de hipertexto pretenden imitar la habilidad del cerebro humano para almacenar y recuperar información a través de la asociación de ideas.

La unidad fundamental de información en un documento de hipertexto es llamada nodo. Idealmente, un nodo es una porción del documento que cubre un concepto. Un nodo puede caber en una sola pantalla de computadora, o puede ser tan pequeño como

una palabra o tan extenso como un libro. Un nodo puede contener una combinación de texto, gráficas y/u otras formas de información.

Los nodos se conectan entre sí a través de referencias cruzadas denominadas ligas. Para recuperar más información acerca de un tema en particular, el lector de hipertexto simplemente selecciona una liga, lo cual trae a la pantalla el contenido del nodo al cual hace referencia dicha liga. El número de ligas es variable dependiendo del contenido de cada nodo. Algunos nodos están relacionados con muchos otros, por lo cual tendrán muchas ligas, mientras que otros pueden servir sólo como destinos finales.

Bush predijo que sería necesario almacenar más que texto en cada uno de los nodos. Él estaba particularmente interesado en almacenar mensajes orales. En la actualidad, los sistemas de hipertexto que incorporan medios de información adicionales al texto, como son gráficas, sonido y video en movimiento, son llamados hipermedia.

La forma más sencilla de definir el hipertexto es comparándolo con el manejo tradicional de textos. Sea en forma impresa o en archivos electrónicos en una computadora, el texto tradicional es secuencial, lo que significa que existe una sola secuencia lineal que determina el orden en el cual el texto puede ser leído. En el caso de un libro, por ejemplo, hablamos de leer la página uno, la dos, y así sucesivamente.

El hipertexto no es secuencial, ya que no existe un solo orden que defina la secuencia de lectura de la información. La figura I.1 ilustra esta situación. Supongamos que el lector inicia con el fragmento A. En vez de tener un solo camino para continuar, esta estructura le ofrece tres opciones diferentes: ir a B, D o E.

Asumiendo que se decidiera por ir a B, el lector se enfrenta nuevamente a decidir su trayectoria, bien hacia C o E, y de E podría llegar a D. Como podemos observar, el lector habría podido viajar de A hacia D directamente, lo cual nos indica que puede haber diferentes caminos que unan dos elementos en una estructura de hipertexto.

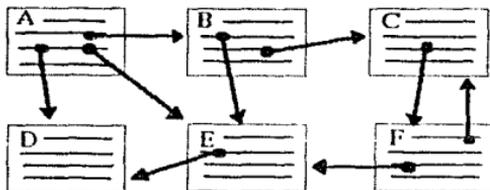


Figura 1.1. Estructura de Hipertexto.

Así pues, el lector decide al momento de consultar la información la trayectoria que seguirá. El autor del texto ha definido una serie de alternativas en vez de establecer un solo flujo de información. El proceso de moverse a través de esta red de nodos y ligas se conoce como navegación. Normalmente, un usuario

tiene la necesidad de regresar a un nodo que ha visitado anteriormente, por lo cual un sistema de hipertexto debe contemplar esta facilidad.

Un documento de hipertexto, al cual nos referiremos como hiperdocumento, es construido en parte por el autor -quien crea las ligas- y también por el lector, quien decide que trayectoria seguir a través de él. Debido a que un hiperdocumento contiene muchas posibilidades, el autor debe poner más cuidado a la estructura general del mismo, que cuando crea un documento de texto tradicional. Los lectores de hipertexto deben asumir un papel mucho más activo que el tradicional.

Uno de los resultados de la naturaleza no lineal del hipertexto es que es relativamente fácil para el lector 'perdersse' durante la navegación por nodos y ligas. Alan Littleford, autor de un sistema de hipermedia llamado HyperBase, denomina a este fenómeno 'Perdido en el Hiperespacio'. Cada liga seleccionada por el lector puede tener un claro significado, pero cuando el lector ha navegado por varias de ellas, la importancia del nodo actual en relación al que inició su consulta puede ser poco clara. Esto puede comprenderse si imaginamos seguir una cadena de sinónimos en un diccionario. El significado de la última palabra de la cadena puede ser muy diferente del de la primera, a pesar de la selección a través de sinónimos. Esta situación puede ser controlada disminuyendo la 'profundidad' del documento, es decir, la máxima longitud de una cadena de ligas.

Este fenómeno se manifiesta de varias formas; el lector no puede regresar a un nodo anterior, o pierde de vista a qué nodo intenta llegar. Precisamente uno de los problemas que enfrentan los autores de hiperdocumentos es que es mucho más difícil asegurarse de que el lector ve cada concepto del texto en el orden más 'coherente'.

Por ejemplo, el autor de un instructivo para volar un avión de control remoto, puede asumir que el lector ha leído información básica a través de las primeras secciones del documento (que tipo de baterías utilizar, como alimentar la gasolina al motor) antes de consultar cómo controlar el avión en pleno vuelo.

Es decir, el autor de un hiperdocumento no puede asumir que el lector ha leído nodos que lógicamente precedan al actual. Es por esto que el autor debe tratar de hacer cada nodo lo más 'autosuficiente' posible, es decir, claro, conciso y centrado en un concepto.

Existen diversas opiniones sobre que debe tener un sistema para poder ser considerado como hipertexto. Por ejemplo, Frank Halasz, uno de los autores de NoteCards, sugiere que un sistema de hipertexto debería incluir una representación explícita de la estructura de nodos y ligas, ya que no es suficiente con mostrar al usuario posicionado en un nodo las ligas que salen de ese nodo hacia otros. K. Eric Drexler, otro estudioso del hipertexto, ha hablado del uso de ligas bidireccionales que permitieran desplegar para un nodo determinado una lista de ligas que llevan hacia él, y no sólo de las que pueen hacia otros nodos de la red.

De igual modo, es fácil confundir el concepto de hipertexto con ciertas aplicaciones tradicionales. Por ejemplo, al principio de este capítulo mencionamos que el hipertexto es una especie de manejador de bases de datos. Ciertamente, existen similitudes entre ambos; un sistema de hipertexto requiere alguna forma de base de datos para almacenar y recuperar la información contenida en los nodos. Sin embargo, la diferencia fundamental radica en la perspectiva del usuario. Una base de datos tradicional tiene una estructura extremadamente rígida, generada por un lenguaje de definición de datos. Toda la información contenida en la base sigue la

misma estructura, de tal forma que puede haber miles de registros, todos ellos con los mismos campos para almacenar nombre, dirección, salario, etc.

Un sistema de hipertexto, por el contrario, no tiene una definición central ni una estructura regular. Siguiendo con el ejemplo, puede haber miles de nodos con información sobre empleados de una compañía. Sin embargo, algunos de ellos contendrán gran cantidad de información, mientras que otros serán mínimos. Puede ser que algunos empleados trabajen en proyectos relacionados con otros departamentos, por lo que sus nodos contendrán ligas hacia otros nodos. Cada liga existe porque tiene sentido en términos de la asociación entre los nodos que una, y no por cumplir una regla general para todos los nodos.

Y que decir de los procesadores de texto y los sistemas de composición de documentos (conocidos en inglés como 'desktop publishing'). Ambos han tenido un enorme impacto en la manipulación de textos, al igual que el hipertexto, y todos ellos son inconcebibles sin el uso de la computadora. Sin embargo, la diferencia fundamental radica en su finalidad: los dos primeros tienen como objetivo la producción de documentos impresos. Este no es el propósito del hipertexto. No tiene sentido pensar en obtener una impresión de un hiperdocumento en su totalidad. Recordemos que el texto convencional es el resultado de una serie de decisiones hechas por el autor, mientras que el hipertexto consiste de muchos 'textos virtuales' diferentes. El lector establece uno de estos textos al escoger que ligas seguir durante su navegación por el sistema. Es por ello que el hipertexto tiene sentido sólo si pensamos en una computadora, que nos permita manipular la información en línea. Siendo así, lo que el hipertexto requiere es suficiente espacio para mantener toda la información disponible en línea.

II. EVOLUCION Y APLICACIONES DEL HIPERTEXTO.

II.1 ¿ Por qué hipertexto ?

La revolución de las computadoras en los últimos años ha generado un crecimiento explosivo en la cantidad de información producida en forma de reportes, gráficas, cartas, datos, etc. Un problema inherente es que la información ha crecido aún más rápido que la tecnología usada para almacenarla. Por ejemplo, cada vez que un nuevo avión se produce en la compañía Boeing, cuatro camiones llenos de documentos viajan junto con él. Bajo esta situación, el problema ya no es encontrar información o construir un lugar suficientemente grande para almacenarla, sino poder tener mecanismos para distinguir la información relevante, de la que no lo es.

Otra razón para buscar tecnologías más eficientes para acceder información es el creciente uso de la llamada multimedia -mezcla de texto, imágenes, sonido y datos-. Es posible crear documentos mucho más valiosos a través de multimedia. Así, el hipertexto puede representar una herramienta para administrar y recuperar información de enormes 'bases de datos' de multimedia.

En general, el hipertexto es una solución potencial a problemas que involucran bases de datos voluminosas, fuertemente relacionadas, que pueden ser consultadas por diferentes usuarios para recuperar información de diversa índole. El tipo de información contenida en el sistema es casi irrelevante, es su estructura lo que determina cuando el hipertexto puede ser una buena alternativa. Por ejemplo, enciclopedias, diccionarios y otros textos similares usualmente son

consultados en forma aleatoria, no secuencial, por lo que son candidatos a integrar sistemas de hipertexto.

Lo mismo sucede con los manuales técnicos orientados a resolver problemas específicos. Por ejemplo, la documentación de un avión de guerra F-18 consta de trescientas mil páginas, las cuales se actualizan constantemente. Imaginemos el ahorro en costos de almacenamiento y mantenimiento de esta información si estuviera contenida en un sistema de hipertexto.

Las computadoras son capaces de almacenar, recuperar y presentar grandes cantidades de información. La inteligencia humana es mucho más eficiente al discernir entre la información útil y la que no lo es para resolver un problema. El hipertexto es una forma de interactuar con la información, basada en la inteligencia natural para resolver los problemas, y utilizando las fortalezas de la computadora. Un sistema de hipertexto trabaja en conjunto con el usuario que tiene la inteligencia para entender el contenido de los diversos nodos y determinar el camino a seguir: el sistema almacena la información y la presenta al usuario, el cual selecciona las ligas apropiadas hasta encontrar lo que busca. Un sistema de hipertexto bien diseñado permite al usuario explorar conceptos en forma no secuencial, obteniendo lo mismo panoramas globales de información que detalles específicos de un tema, aprendiendo por asociación y experimentando tantas veces como desee.

En 1989, Gerry Feper de la compañía IBM y algunos colegas trataron de implementar un sistema para soportar el mantenimiento a redes de computadoras en dos diferentes versiones: un sistema experto y un sistema de hipertexto. El experimento consistió en alimentar la misma información en ambos sistemas y someter un conjunto de problemas a resolver a veinte personas con experiencia en el área.

Utilizando el sistema de hipertexto, este grupo resolvió correctamente el 81% de los problemas, contra el 67% logrado a través del sistema experto. Sin embargo, utilizar el sistema experto fue más rápido, con 4 minutos por problema contra 5 utilizando el hipertexto. Cuando se preguntó a los usuarios que sistema preferían, el 50% eligió el hipertexto, contra 25% que eligió el sistema experto. La razón de esto fue la facilidad de actualizar la información en el primero, contra la dificultad de actualizar la base de conocimientos del sistema experto.

Además de los beneficios directos que un sistema de hipertexto tiene para el usuario, existen otras consideraciones en relación con el valor de la información y la importancia de su adecuado manejo. Es una frase muy trillada el que la información es el recurso más valioso de una organización en la actualidad. La disponibilidad de información adecuada en el momento adecuado es un factor crítico para tomar decisiones acertadas. Pues bien, un sistema de hipertexto permite concentrar la información y ponerla a disposición de los usuarios, asegurando que ellos tendrán acceso a las versiones más actualizadas, que todos verán los mismos documentos, y a pesar de ello mantendrán su autonomía en la consulta, ya que cada usuario puede tener requerimientos muy diferentes de los demás, y el hipertexto les da flexibilidad para navegar a través de la información, sin sujetarse a un esquema determinado.

II.2 Evolución del hipertexto.

El hipertexto tiene una historia considerable, a pesar de que la mayoría de nosotros hemos oído de él hasta hace poco tiempo.

Vannevar Bush (1890-1974) es considerado el padre del hipertexto. En 1945, propuso un sistema llamado Memex (siglas del inglés 'memory extender') que nunca fue construido, quedando sólo descrito en los apuntes de Bush. De hecho desarrolló algunas de sus ideas para el Memex en 1932 y 1933 y finalmente escribió un bosquejo en 1939, que no fue publicado hasta 1945 en el Atlantic Monthly bajo el título 'As We May Think'. Bush describió el Memex como una especie de archivo o biblioteca mecanizada, un dispositivo en el cual un individuo almacena sus libros, apuntes, artículos, y que puede ser consultado con gran rapidez y flexibilidad. El Memex almacenaría toda esta información en microfichas, las cuales serían guardadas en un escritorio muy especial. En él, habría varios equipos de proyección para permitir al usuario comparar diferentes microfichas, en forma muy similar a los sistemas de ventanas tan populares en las computadoras de hoy en día.

El Memex tendría incluso un digitalizador para la captura de nuevo material, y permitiría al usuario elaborar comentarios y notas. Una de las ideas de Bush era que habría en algún momento mucha información contenida en microfichas, lista para ser incorporada al sistema: libros, revistas, fotografías, periódicos.

La principal razón que impulsó a Bush a desarrollar esta idea fue su preocupación respecto a la explosión de información científica de su tiempo, que hacía muy difícil aún para los especialistas seguir los avances en alguna disciplina. Bush veía la necesidad de permitir a la gente encontrar información más fácilmente de lo que el texto impreso permitía. Y de acuerdo con su visión, la fortaleza más importante del Memex era la posibilidad de asociar conceptos, de tal forma que una pieza de información permitiera la selección automática de otra pieza relacionada. Esto es sin duda, el concepto de hipertexto. Adicionalmente al establecimiento de ligas individuales, Bush quería que el Memex soportara la creación de cadenas a través del material, en forma de una secuencia de ligas que

combinaran información relevante para un determinado tema. Incluso Bush preveía el surgimiento de una profesión dedicada a la creación de estas cadenas de información.

El artículo de Bush no pasó desapercibido. La revista Life incluso publicó una serie de ilustraciones imaginando cómo luciría el Memex, y un escenario de las distintas proyecciones a medida que el usuario completaba una liga. En la actualidad, existen computadoras con muchas de las funciones ideadas por Bush, pero basadas en una tecnología completamente diferente a la de la microfilmación.

Después del artículo de Bush, no hubo gran movimiento en el campo del hipertexto por unos veinte años. La industria estaba preocupada mejorando las computadoras hasta el punto que fuera costeable utilizarlas en forma interactiva, pero resultaba tan caro que era considerado un desperdicio dedicar los recursos de una computadora a tareas diferentes del procesamiento numérico, como podría ser el proceso de textos. Pese a esto, en 1962, Douglas Engelbart publica un documento: 'Augmenting Human Intellect: A Conceptual Framework', que se encargó de ocupar los siguientes treinta años de su agenda. Engelbart buscó la forma de definir e integrar la funcionalidad necesaria a las computadoras para permitirles aumentar las habilidades del hombre. Las funciones que él pensó necesarias incluían ligas entre textos, correo electrónico, bibliotecas de documentos, espacios de almacenamiento privados para que el usuario guardara sus archivos privados, pantallas de computadora con ventanas, y la facilidad de desarrollar trabajos en colaboración con otros usuarios. Este proyecto, conocido como Augment, fue conducido en el SRI (Stanford Research Institute) con un equipo que creció a 45 personas.

Una parte del proyecto Augment fue la llamada NLS (del inglés on-Line System), el cual tenía varias características de hipertexto. Durante el proyecto, los investigadores almacenaron todos sus documentos, reportes y memoranda en una especie de bitácora compartida, que les permitía incluir referencias a otros trabajos en sus propios escritos. Esta biblioteca creció hasta 100,000 artículos.

En 1968 Engelbart ofreció una demostración de NLS en una sesión de la 'Fall Joint Computer Conference'. Esta demostración impulsó enormemente el desarrollo de la computación interactiva. A pesar de este éxito, en 1975 el gobierno le retiró el apoyo financiero. Augment continuó como un servicio de automatización de oficinas. En la actualidad es comercializado por McDonnell Douglas. Engelbart continúa desarrollando sus ideas a la cabeza del proyecto 'Bootstrap' en la Universidad de Stanford.

El término 'hipertexto' fue inventado por Theodor Nelson en 1965. Nelson es un pionero del hipertexto con su proyecto Xanadu. Algunas partes de Xanadu funcionan como un producto de la compañía Xanadu Operating Co., desde 1990. Sin embargo, la visión fundamental de Xanadu no ha sido implementada nunca: la idea de un repositorio para cualquier cosa que cualquier persona hubiera escrito en cualquier tiempo. Nelson concibe el hipertexto como un medio literario, por lo que cree que todo está fuertemente relacionado y hay que mantenerlo en línea. Resulta evidente que esta cantidad de información no es posible mantenerla en la computadora de un solo usuario, por lo que cuando fuera requerida información no contenida en forma local, la computadora se conectaría a través de una red al repositorio, en forma transparente al usuario.

En Xanadu es posible direccionar cualquier palabra de cualquier documento desde otro documento, lo cual, en combinación con el almacenamiento distribuido de

información representa un enorme poderío de consulta y asociación de ideas. Además, en Xanadu nunca se borraría ningún texto, aún en el caso de ser actualizado, puesto que algunos usuarios podrían hacer referencia a versiones anteriores de un documento.

A pesar de que Xanadu fue parcialmente implementado hasta hace poco, Andries van Dam y un equipo de trabajo desarrollaron sistemas de hipertexto en los 60's, en la Universidad de Brown. El llamado 'Hiptertext Editing System' creado en 1967 fue el primer sistema funcional de hipertexto en el mundo. Fue diseñado para correr en una partición de memoria de 128 Kbytes en una computadora IBM/360 y fue financiado por IBM a través de un contrato de investigación. Este sistema tenía dos objetivos: producir documentos impresos fácil y rápidamente, y explorar el concepto de hipertexto. Después de terminado, la IBM lo vendió al 'Houston Manned Spacecraft Center' en donde se utilizó para producir documentación para las misiones del Apolo.

El segundo sistema de hipertexto fue PRESS (File Retrieval and Editing System) desarrollado en 1968 como continuación del primero, en la misma Universidad y en la misma plataforma IBM. Ambos sistemas tenían las características fundamentales del hipertexto, al relacionar y navegar hacia otros documentos, pero la mayoría de la interfase del usuario estaba basada en texto, y requería que éste especificara en forma indirecta los movimientos.

El último proyecto de hipertexto en la Universidad de Brown es llamado Intermedia, y se aplica en la enseñanza de la Biología y Literatura Inglesa. Intermedia es una herramienta utilizada por los profesores para preparar sus lecciones y material de cursos, y por los estudiantes para aprender y crear reportes. Una de las ideas fundamentales derivadas del trabajo de van Dam es la de 'web', es decir, un conjunto de ligas que permanecen juntas. Seleccionando un

'web', un lector de un hipertexto impone un conjunto particular de ligas en un documento, haciendo posible que diferentes usuarios establezcan sus propias ligas en el mismo documento.

En 1972, el último de los sistemas que Frank Halasz llama la primera generación de hipertexto, se desarrolló en la Universidad Carnegie-Mellon. Esta primera generación se caracteriza por funcionar en computadores grandes, utilizar exclusivamente texto como información, y proveer soporte para trabajo cooperativo en redes de hipertexto. El sistema, llamado ZOG, fue diseñado específicamente para dar un tiempo de respuesta rápido a un número considerable de usuarios.

La segunda generación de sistemas de hipertexto comienza en los 80's. con el advenimiento de las estaciones de trabajo, y la diferencia entre ambas generaciones radica en buena medida en las características inherentes de la estación, y está orientada hacia las interfases con el usuario y el manejo de información en multimedia. En 1983 surge KMS, la versión comercial de ZOG. KMS es una abreviatura de Knowledge Management System. Corre en estaciones de trabajo Sun y Apollo, y ha sido usado para un gran número de aplicaciones. KMS es capaz de almacenar texto y gráficas en sus nodos. Es muy apropiado para ambientes muy grandes, con muchas decenas de miles de nodos, en los cuales muchos diseñadores e ingenieros deben compartir los mismos documentos a través de una red.

KMS tiene una estructura de datos muy simple, basada en un sólo tipo de nodo llamado 'frame'. Un 'frame' puede ocupar la pantalla entera, aunque normalmente la pantalla es dividida en dos 'frames'. El usuario no puede mezclar nodos de tamaños diferentes, y no puede tener más de dos nodos en la pantalla al mismo tiempo. Esto puede parecer una limitación, aunque los defensores de KMS aseguran que es mucho mejor usar el mecanismo de navegación propio del hipertexto para cambiar el

contenido de la pantalla, que tener que moverse entre una infinidad de ventanas para encontrar la información deseada. Si un elemento de la pantalla no está ligado a otro nodo, al elegirlo se genera un nodo vacío que el usuario puede llenar; esto hace del proceso de creación de nodos y ligas una forma más de navegación. Es posible también, al activar una liga en la pantalla, correr un programa desarrollado en un lenguaje propio de KMS, lo que permite adecuar el producto a muchas aplicaciones.

En 1982, arranca un proyecto para diseñar el sistema 'Symbolics Document Examiner' dirigido a los usuarios de las estaciones de trabajo Symbolics. Este sistema tenía una interfase de hipertexto para la documentación en línea de las estaciones, y los usuarios lo aceptaron porque era la mejor forma de obtener esta información. El manual de Symbolics también existía en una versión impresa de 8000 páginas. Esta información fue representada en 10,000 nodos de hipertexto con 23,000 ligas, haciendo un total de 10 Mbytes de almacenamiento, y fue escrita a través de una interfase conocida como Concordia. La información fue modularizada en base a un análisis de las probables necesidades de los usuarios. El principio básico fue tener un nodo por cada pieza de información que un usuario pudiera querer consultar. Sin embargo, debido a que el concepto de hipertexto no era muy popular aún, la interfase se trató de hacer más parecida a un libro, en vez de tratar de llevar a los usuarios a utilizar la navegación. La información se dividió en capítulos y secciones, y tenía un índice. Para determinar la facilidad de uso del sistema, los diseñadores condujeron una encuesta a 24 usuarios. Dos de ellos prefirieron la versión escrita, pero el resto usó sólo la versión en hipertexto. Puede decirse que el 'Symbolic Document Examiner' representa uno de los primeros contactos del hipertexto con el mundo real, ya que es una aplicación práctica utilizada por clientes, sin embargo, la estación de trabajo Symbolics era tan cara

cuando se lanzó el producto, que pocos tuvieron la oportunidad de conocerlo y utilizarlo.

En el mismo año de 1981, Peter Brown comienza a desarrollar uno de los primeros sistemas comerciales de hipertexto para una computadora personal, el cual es llamado Guide; en 1985 la compañía Office Workstations Limited (OWL) comienza a comercializarlo para la computadora Apple Macintosh. El sistema Guide fue liberado para su uso en la IBM PC en 1987. Como la mayoría de los sistemas para computadoras personales que surgieron después, Guide ofrecía menos funcionalidad que los hipertextos basados en computadoras grandes, a cambio de una interfase gráfica para el usuario. Guide se basa en ventanas de texto con capacidad de 'scroll', aunque curiosamente las ligas al ser activadas causan que el texto avance o retroceda a una nueva posición.

Guide soporta tres tipos de ligas: de reemplazo, 'pop-ups' y saltos. La primera es utilizada para hacer una expansión en línea del texto asociado a la liga, a un texto nuevo y generalmente mayor que el anterior; este tipo de ligas forman una estructura jerárquica de texto y son útiles para representar texto en la forma de un libro tradicional con capítulos, secciones y párrafos; típicamente, la primera pantalla muestra los títulos de todos los capítulos, y el usuario puede expandir el nombre del capítulo que le interesa, reemplazándolo por la lista de secciones de ese capítulo. Luego puede posicionarse en una sección y expandirla para mostrar la lista de párrafos, y así sucesivamente. El segundo tipo de liga permite abrir pequeñas ventanas o 'pop-ups', útiles para anotaciones de pie de página. La tercera forma de liga es utilizada para saltar a un lugar diferente dentro del hiperdocumento, y el usuario tiene que activar un ícono especial para regresar al nodo anterior. Las tres formas diferentes de ligas se distinguen entre sí porque la forma del cursor varía al posicionarlo en ellas.

En el año de 1983, Ben Shneiderman arranca un proyecto de investigación en la Universidad de Maryland, conocido originalmente como TIES (proveniente de 'The Electronic Encyclopedia System'). Sin embargo, por problemas de derechos de autor fue cambiado a Hyperties, y desde 1987 es un producto comercial para computadoras personales compatibles con IBM, vendido por Cognitica Corporation.

Las técnicas de interacción de Hiperties son muy simples. Algunas porciones de texto en la pantalla están marcadas con video inverso y el usuario puede seleccionarlas. En vez de llevarlo directamente al nodo destino como la mayoría de los sistemas de hipertexto, Hiperties muestra al usuario una pequeña definición al final de la pantalla, lo cual le permite revisar rápidamente el contenido del nodo y decidir si completa la liga o no. Una liga en Hiperties lleva al usuario al principio del nodo destino, que puede tener varias páginas de longitud. Muchas de las características de diseño de este producto provienen de sus autores, que eran curadores de museos e historiadores en su mayoría.

En 1986, es liberado el producto NoteCards, por Xerox. Inicialmente desarrollado para la familia de máquinas D de Xerox, fue portado a estaciones de trabajo convencionales como Sun para hacerlo comercial. NoteCards soporta nodos con gráficas y animación al igual que con texto formateado. Además, provee gráficas con la estructura global de los hiperdocumentos, como una ayuda para la navegación dentro de ellos.

NoteCards es pionero en la aplicación de 'metáforas' al hipertexto; cada nodo es representado en la pantalla como una carta, y puede ser de la longitud que sea necesaria; cada nodo tiene un tipo dependiendo del contenido de la carta (texto, gráficas o imágenes) aunque el usuario puede definir otros tipos.

Quizá el paso definitivo para dar a conocer el concepto de hipertexto al 'mundo real' lo dio Apple cuando introdujo el producto HyperCard en 1987. Y una de las razones de su gran difusión es que a partir de ese año, el producto se regaló a cada persona que comprara una Macintosh. Como puede deducirse del nombre, Hypercard utiliza cartas para representar los nodos. Sin embargo, ningún nodo puede exceder la longitud de una carta. Hypercard cuenta con un lenguaje de programación llamado HyperTalk, muy fácil de aprender. Un hiperdocumento o colección de cartas es llamado un 'stack', y los programas en HyperTalk son llamados 'stackware'. Una de las facilidades del HyperCard es la posibilidad de crear botones en la pantalla, y asociar un programa a ellos. El botón (representación gráfica de la liga) es activado para ejecutar el programa asociado, que frecuentemente contiene una sola instrucción para hacer un salto de hipertexto.

Dado el desarrollo que ha tenido el concepto del hipertexto, en años recientes se han llevado a cabo dos conferencias. En 1987, en la Universidad de Carolina del Norte, se realizó la primera conferencia de la ACM sobre hipertexto, Hypertext'87. Dos años más tarde, se llevó a cabo la primera conferencia abierta, en Inglaterra: Hypertext'2.

Resumiendo, podemos decir que el hipertexto fue concebido en 1945, nació en los 60's, y fue desarrollándose durante los 70's para entrar al 'mundo real' en los 80's, con un crecimiento especialmente acelerado a partir de 1985. Actualmente, existen una serie de aplicaciones del hipertexto en proyectos reales, de las cuales hablaremos a continuación.

II.3 Aplicaciones.

Del material anterior podemos concluir que el número de aplicaciones factibles para el hipertexto es similar a la diversidad de textos impresos. Sin embargo, cada aplicación tiene requerimientos particulares, dependiendo de la naturaleza de la información a representar. Por ejemplo, un libro de historia, un manual de procedimientos o las instrucciones para operar una lavadora son tipos de texto muy diferentes entre sí, y cada uno de ellos requeriría diferentes tipos de soporte de hipertexto.

Así mismo, el hipertexto no es la solución a todos los problemas. Ben Shneiderman [01] (ver referencia en la Bibliografía) ha propuesto tres reglas para determinar cuando una aplicación es susceptible de desarrollarse a través de hipertexto; estas reglas definen cuales son las características que debe presentar la aplicación, y son llamadas por su autor las 'reglas de oro' del hipertexto:

- Un gran volumen de información, organizado en numerosos fragmentos.
- Los fragmentos están relacionados entre sí.
- El usuario necesita sólo una pequeña parte de la información en cualquier momento.

A continuación revisaremos algunas de las principales aplicaciones del hipertexto.

II.3.1 Aplicaciones en Computación.

Siendo el hipertexto un concepto fuertemente ligado a las computadoras, es natural su uso en aplicaciones orientadas a las ciencias de la computación. Algunas de ellas son:

Documentación en línea.

Esta es una de las aplicaciones más claras del hipertexto. Para el usuario de una computadora es muy importante contar con herramientas que le permitan un rápido acceso a las partes de la documentación que él requiere en un momento dado. El hipertexto es una solución muy adecuada para este requerimiento. En la actualidad existe una tendencia a entregar los programas comerciales de computadora con manuales en línea.

Sistemas de ayuda al usuario.

Los usuarios de computadoras frecuentemente requieren más información de la que pueden consultar en los manuales. De hecho, Robert L. Mack, del "IBM User Interface Institute", muestra que los usuarios consideran los manuales tradicionales como el segundo peor método de docencia para el aprendizaje inicial de un sistema de cómputo [02].

El hipertexto provee un ambiente para integrar diversas formas de ayuda al usuario, incluyendo el ya mencionado manual en línea, un programa tutor, un sistema de ayuda en línea y un sistema de mensajes de error. En este ambiente, es posible

para el usuario ligar un mensaje de error con la sección del sistema de ayuda que le da información adicional sobre el problema que lo causó. O bien si el usuario no comprende alguna palabra del mensaje, puede ligar su definición del manual de referencia en línea. Finalmente, si el usuario ha entendido el error, y desea repasar un determinado tema, podría ligar la porción adecuada del programa tutor, para cursar una lección. En términos generales, la gente puede recordar mucho mejor la información cuando pueda relacionarla con una situación concreta, y cuando han sido participantes activos, en vez de haber asumido un papel pasivo, como la sola lectura de un manual.

Ingeniería de Programación.

Durante el ciclo de vida de cualquier sistema, se genera una gran cantidad de documentos, que van desde las especificaciones funcionales y de diseño hasta la documentación de cambios y actualizaciones, y el hipertexto tiene un gran potencial para relacionar estos documentos entre sí. Por ejemplo, partiendo de un documento de requerimientos del usuario, es posible ligar las especificaciones de diseño que cubren un cierto requerimiento. Hecho esto, uno podría viajar del documento de diseño a la porción del código correspondiente y revisar la implementación. O bien, viajar en sentido inverso, de una porción de código, hacia el requerimiento que le dio origen.

Para obtener un beneficio completo, es necesario que el desarrollador se apega a alguna metodología de ingeniería de programación soportada por una herramienta CASE (Computer Aided Software Engineering). Tal es el caso del proyecto "Dynamic Design" en Tektronix [03], el cual basado en el sistema de hipertexto Neptune interconecta

especificaciones, documentos de diseño, documentación del usuario, programas fuentes y su documentación.

Es también factible utilizar el hipertexto en otro tipo de aplicaciones de programación, como la creación de editores de código de programas. Por ejemplo, es posible seleccionar una variable dentro de un programa y ligar su definición y comentarios asociados, o ligar la llamada a un procedimiento (CALL) con el texto del mismo.

Existe también interés en apoyar el proceso específico de diseño a través de hipertexto, como es el caso del proyecto gIBIS (graphical Issue Based Information System) de MCC (Microelectronics and Computer Technology Corporation) en Austin [04]. gIBIS, parte del proyecto "Design Journal", trata de capturar las razones detrás del proceso de desarrollo de un sistema. Es un sistema de hipertexto multiusuario, en el cual los participantes en el proceso aportan su experiencia a la resolución de todas las consideraciones de diseño, sugiriendo una solución al problema, y dando argumentos a favor y en contra de las diversas soluciones. Todo esto se representa en una estructura de hipertexto con tres tipos de nodos: problema, solución y argumentos. La figura II.1 muestra un ejemplo de una discusión sobre el tema de los manuales en línea. Existen tres soluciones propuestas: utilizar hipertexto, utilizar archivos electrónicos tradicionales, o continuar trabajando con manuales impresos. Se han ingresado varios argumentos, y el usuario ha seleccionado el argumento denominado Lectura, cuyo texto aparece en la ventana. Incluso las ligas muestran un pequeño icono para mostrar si el argumento es a favor o en contra.

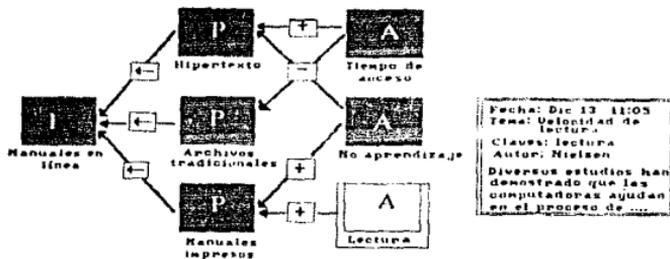


Figura H.1. Fundamentos de gIBIS.

Sistemas Operativos.

El hipertexto posee las características para influir fuertemente en el diseño de las interfaces de usuario de las computadoras, en particular de las personales. Las computadoras actuales están basadas fundamentalmente en el manejo de archivos como unidades básicas de manejo de información, que pueden ser localizados en un lugar determinado del espacio de almacenamiento, y que son generalmente usados por una sola aplicación. La mayoría de los sistemas de archivos los organizan en una estructura jerárquica, y llevan al usuario a viajar a través de varios niveles de subdirectorios hasta encontrar los archivos requeridos. El usuario frecuentemente olvida en donde guardó determinada información, y el sistema le permite en la mayoría de los casos hacer búsquedas sólo en base al nombre del archivo.

Fecha: Dic 13 11:03
 Tema: Usabilidad de lectura
 Claves: lectura
 Autor: Nizkor
 Diferentes estudios han demostrado que las computadoras ayudan en el proceso de ...

Sería posible crear un ambiente basado en hipertexto que permitiera a diferentes aplicaciones ligar en forma transparente la información generada por otras aplicaciones y almacenada en algún lugar de la computadora. Los usuarios podrían establecer la liga una sola vez, y posteriormente el sistema los llevaría de un nodo de información a otro, evitándoles la búsqueda manual.

II.3.2 Aplicaciones generales.

Exista un gran potencial para la aplicación del hipertexto a problemas comunes. He aquí algunos ejemplos.

Manuales técnicos.

Es posible elaborar un hiperdocumento que contenga descripciones de cómo reparar automóviles, ligado a videos que muestren a mecánicos especializados realizando diversas composuras, y mostrando típicos errores que pueden ser evitados.

El sistema IGD (Interactive Graphical Documents) [05] es un temprano ejemplo de un manual en hipertexto para reparación de equipo electrónico producido en la Universidad de Brown, con el apoyo de la Marina de los Estados Unidos. Está basado en mostrar gráficamente las reparaciones, en vez de describir el mismo material en forma puramente textual. Sería posible en un manual como este, que las ligas dependieran de la situación actual. Por ejemplo, puede ser que una pieza se pueda reparar por varios métodos, uno ordenado y cuidadoso, y otro rápido y provisional. En caso de que el técnico tuviera todo el tiempo para hacer la reparación, se

accesaría el primer procedimiento, y de lo contrario se mostraría el segundo. Actualmente existe una clara tendencia de muchas compañías automotrices a entregar sus manuales técnicos en sistemas de hipertexto, normalmente en discos ópticos; existe tal cantidad de modelos diferentes y tal complejidad en los motores que la elaboración y distribución de los manuales tradicionales está empezando a parecer incosteable.

Diccionarios y libros de referencia.

Varios diccionarios y libros de consulta han sido convertidos de su forma impresa tradicional a hipertexto. Destaca uno de estos proyectos, el "Oxford English Dictionary" (OED).

El OED es uno de los más grandes textos que han sido convertidos a hipertexto [06]. La versión original fue publicada en doce volúmenes de 1884 a 1928, con un suplemento de cuatro volúmenes de 1972 a 1986. Por su antigüedad, el OED no existía en medio electrónico por lo que tuvo que ser capturado manualmente, durante dieciocho meses. Una razón por la cual no se utilizó un proceso de digitalización y reconocimiento óptico del texto impreso es que era necesario distinguir en base a ciertas "banderas" la naturaleza de cada elemento del texto del diccionario, es decir, si se trataba de una palabra del diccionario, su definición, su etimología o las referencias a otra literatura. El texto está almacenado como una cadena continua de 570 Mbytes. No existen nodos explícitamente almacenados como unidades separadas, ni ligas almacenadas como apuntadores. El sistema usado para acceder el texto se basa en índices almacenados externamente al texto. Por lo tanto, se puede considerar que este sistema convierte a hipertexto el archivo de texto normal cada vez que es utilizado.

Una razón por la cual el hipertexto es un buen mecanismo de acceso al OED es que éste contiene 569,000 referencias cruzadas dentro del mismo diccionario. Muchas de estas ligas hacen referencia a variantes de una misma palabra, a sinónimos, o a otras palabras con relación a prefijos o sufijos. Por otro lado, la interfase al OED necesitaba permitir varios tipos de presentación de la información, pues cada usuario tiene sus propios intereses, y mientras unos estudian la evolución lingüística de la palabra y buscan citas de las más antiguas, otros se interesan sólo en una simple definición. Esto es soportado ampliamente por el hipertexto, pues en lugar de tener un sistema que automáticamente formatea el texto en una forma predefinida, la interfase de hipertexto del OED permite reestructurar el texto dinámicamente de acuerdo a la decisión de cada usuario.

Es claro que una de las mayores ventajas del hipertexto comparado con un trabajo impreso, es la posibilidad de ligar nodos con video en movimiento y sonido. Por ejemplo, Highlighted Data ha publicado una versión del Diccionario Webster en CD-ROM con grabaciones de como se pronuncian las 160,000 palabras que componen el diccionario.

Auditoría.

La auditoría es otra de las aplicaciones naturales del hipertexto, ya que es un proceso basado en relacionar información de diversas fuentes y verificar su consistencia. El trabajo del auditor incluye la recopilación y producción de un gran número de documentos y el relacionarlos entre sí para sustentar la veracidad de la información que contienen. Una gran cantidad de información, por ejemplo, puede reducirse a un renglón en un estado financiero, por lo que se requieren ligas entre los datos fuente y las conclusiones. En el caso de auditoría en grandes corporaciones internacionales, la ventaja de un medio electrónico como el

hipertexto es mayor, ya que involucra el intercambio de documentos generados en diversos lugares del mundo.

Estudios conducidos por las firmas Price Waterhouse Audit Research y Technology Group [07] indican que aproximadamente el 10% del tiempo invertido en una auditoría se dedica a producir, relacionar y revisar documentos de trabajo. Por lo tanto, un sistema de hipertexto para soportar al proceso de auditoría puede reducir el tiempo necesario para llevarlo a cabo.

Publicidad.

Muchas clases de publicidad pueden ser mejoradas a través del uso de hipertexto. Una de las características importantes del hipertexto es la de dar acceso a una gran cantidad de información, pero mostrando al usuario únicamente la que le interesa en un momento dado. Esto es muy útil en el desarrollo de catálogos de productos, ya que se puede reducir la complejidad en la búsqueda al mostrar a un cliente sólo las opciones que le interesan. Por ejemplo, la empresa automotriz Buick ha desarrollado su catálogo de autos en hipertexto incluyendo en él simulaciones de manejo y otros juegos para ganar atención. Además, hay una sección que permite al usuario comparar los autos Buick con autos similares de la competencia; con sólo elegir un ícono, se muestra una comparación detallada entre los vehículos, lo cual de haberse hecho en forma de texto impreso, seguramente se habría manejado a través de una tabla más complicada para entender por el cliente. Un funcionario de la Buick declaró en la revista Business Week en Octubre 9 de 1989 que el 12% de los que compraron autos después de recibir este catálogo decidieron a favor de la Buick. Esta tasa es el doble de su penetración normal de mercado.

Trabajos de grupo y organización de ideas.

El hipertexto, por su similitud con la forma natural del cerebro humano de relacionar conceptos, es un medio propicio para organizar ideas y proveer un ambiente de trabajo en equipo. Además de permitir la coordinación de fragmentos separados de información, el hipertexto permite a cada usuario la creación de nuevas ligas entre nodos, lo cual enriquece la información cada vez. El trabajo en grupos normalmente está soportado por la tecnología de correo electrónico y los foros o conferencias basadas en computadora, en los cuales varios participantes escriben mensajes con comentarios sobre mensajes escritos anteriormente por otros participantes. El hipertexto puede enriquecer este mecanismo de trabajo. Por ejemplo, el sistema HyperNews de la Universidad Técnica de Dinamarca es una interfase de hipertexto para un ambiente de conferencia electrónica [08]. El sistema conecta a través de ligas los diferentes mensajes relacionados con un tema.

Periodismo.

Mucho del trabajo de un periodista tiene que ver con la recopilación de información, investigación de artículos anteriores y su organización para crear nuevos artículos. Un periódico es un buen candidato para ser convertido a hipertexto. Para un lector, sería mucho más útil poder navegar a través de una estructura de fragmentos informativos relacionados entre sí, que tener que leer un periódico en forma secuencial. Wayne MacPhail, del periódico canadiense Hamilton Spectator, de Ontario, ha desarrollado algunos sistemas de hipertexto a través de una herramienta llamada PC-Hypertext [09].

Uno de ellos, por ejemplo, es una guía en hipertexto para investigadores que están buscando posibles causas de la enfermedad del SIDA.

Ahora que hemos analizado en términos generales el concepto de hipertexto y sus principales aplicaciones, podemos estudiar con más detalle los componentes fundamentales de un sistema de hipertexto, que nos servirán de base para el diseño de nuestra herramienta.

III. ARQUITECTURA DE UN SISTEMA DE HIPERTEXTO.

III.1 Modelo HAM.

Campbell y Goodman [10] han definido un modelo para un sistema de hipertexto de propósito general, conocido como HAM: Hypertext Abstract Machine. En este modelo, se pueden distinguir tres niveles:

1. Nivel de presentación: interfase con el usuario
2. Nivel de hipertexto (HAM): nodos y ligas
3. Nivel de base de datos: almacenamiento y acceso a datos

A continuación explicaremos cada uno de estos niveles.

Nivel de base de datos.

Este nivel está debajo de todos los demás dentro del modelo, y corresponde a las tareas comunes de almacenamiento y recuperación de información, que son en tanto independientes del hipertexto. Es necesario almacenar grandes cantidades de información en diversos medios de almacenamiento: discos magnéticos, ópticos, etc. y recuperar una porción en un momento dado, independientemente de como esté estructurada. Así mismo, este nivel debe manejar el acceso concurrente a la información, niveles de seguridad e incluso consideraciones de respaldo.

Desde el punto de vista de este nivel, los nodos de hipertexto son solo objetos de datos sin ningún significado particular. Cada uno de ellos forma una

unidad que sólo puede ser modificada por un usuario al mismo tiempo, y que ocupa una determinada cantidad de memoria.

Nivel de hipertexto (HAM).

Este nivel se ubica a la mitad del modelo, entre el nivel de base de datos y el de interfase con el usuario. En él es donde el sistema de hipertexto determina la naturaleza de sus nodos y ligas y donde se establece la relación entre ellos.

El nivel de hipertexto es el mejor candidato para la estandarización de formatos para hipertexto, ya que el nivel de base de datos es fuertemente dependiente en su formato de almacenamiento de la implementación física, y el nivel de presentación es totalmente diferente de acuerdo con la aplicación.

Nivel de Interfase de usuario .

Este nivel trata de la presentación de la información en el nivel de hipertexto, incluyendo aspectos como el de qué comandos deben estar disponibles para el usuario, como mostrar los nodos y las ligas, y cuando incluir diagramas o no. Sin embargo estas decisiones no pueden ser tomadas sólo a este nivel, sin considerar la naturaleza de la información en el nivel anterior. Supongamos que un cierto sistema de hipertexto define ligas de diferentes tipos. Corresponde al nivel de interfase determinar la forma de mostrar al usuario los diferentes tipos, que puede ser cambiando la forma del cursor, a través de iconos, o incluso con distintos colores, dependiendo de la cantidad de tipos diferentes de ligas.

III.2 Nodos, ligas, gráficas de información.

Hasta ahora hemos hablado de nodos y ligas. A partir de estos elementos se define la terminología básica de un sistema de hipertexto. Así, podemos decir que un hiperdocumento es un conjunto particular de nodos y ligas, independientemente del sistema que se utiliza para manipularlo. Una gráfica se define como el conjunto de ligas de un hiperdocumento. Hablemos más ampliamente de cada uno de estos elementos.

III.2.1 Nodos.

Los nodos son la unidad fundamental del hipertexto, aunque no hay consenso sobre lo que realmente constituye un nodo. Sin embargo podemos hacer una clasificación de los sistemas de hipertexto en dos grandes grupos: por un lado los sistemas basados en ventanas, que no imponen restricciones en cuanto al tamaño de los nodos, y sistemas basados en elementos de tamaño fijo llamados marcos o 'frames'. Un marco ocupa un determinado espacio en la pantalla de la computadora, no importa la cantidad de información que contenga. A menudo el tamaño del marco es definido como el tamaño de la pantalla, aunque esto no es una regla. Un ejemplo típico es el HyperCard, en donde los nodos consisten en cartas de tamaño fijo. Dado que en estos sistemas los nodos tienen un tamaño fijo, el usuario puede verse en la necesidad de fragmentar una cierta cantidad de información en varios marcos, si es que no cabe en uno sólo.

Por el contrario, los sistemas basados en ventanas requieren que el usuario utilice un mecanismo de avance y retroceso de página o 'scroll' en adición a los mecanismos propios del hipertexto para lograr el acceso a la parte deseada del nodo y mostrarla en la ventana. Así, a pesar de que el sistema puede mostrar sólo una

parte del nodo a través de la ventana en un momento dado, el nodo puede ser tan grande como se requiera, desapareciendo la necesidad de fragmentar textos que en una forma natural debieran integrar un solo nodo. Una desventaja de estos sistemas es que el diseñador del hipertexto no tiene control sobre lo que se muestra de un nodo en la ventana, ya que eso está en manos del usuario. Una ventaja, además de no tener que dividir el texto por restricción de espacio, es que las ventanas puedan ser de diferente tamaño, dependiendo de la importancia y naturaleza de la información que contienen. Los sistemas Guida e Intermedia son típicos ejemplos.

Una situación difícil de resolver es la cantidad de información que debiera contener un nodo. ¿ Es mejor dividir un texto determinado en muchos pequeños nodos, o en pocos nodos grandes ? Kreitzberg y Shneiderman [11] realizaron un pequeño experimento para investigar la respuesta. Ellos dividieron el mismo texto de dos formas: 46 artículos de entre 4 y 83 líneas, o 5 artículos de entre 104 y 150 líneas, en el sistema de hipertexto Hyperties. El resultado fue que los usuarios pudieron contestar preguntas acerca de la información bastante más rápido con la primera división, es decir, muchos nodos pequeños (125 segundos contra 178 segundos, por pregunta). De hecho, ellos recomiendan que ningún nodo sea mayor de 100 líneas.

Existe un fenómeno relacionado con esto, que se conoce como 'sobrecarga de información', y ocurre cuando el lector recibe más información de la que puede procesar en forma comprensible. Normalmente cuando se presenta información a una persona, el cerebro la evalúa, selecciona y retiene algunos elementos, descartando otros. Esta especie de filtrado separa la información relevante de la que no lo es. Cuando el flujo de información es muy grande, el proceso de filtrado se rompe, y el lector presta atención a la información irrelevante, ignorando la que puede ser importante.

El problema de la sobrecarga de información no es sólo cuestión de cantidad. Es también un problema de comprensión. Cuando la información es comprensible para el lector, éste la acepta y la organiza para su uso posterior. Cuando no lo es, se convierte en ruido y el lector la desecha. El hipertexto, que está diseñado para proporcionar acceso rápido a más información, puede favorecer este fenómeno.

Es por ello que Kreitzberg y Shneiderman han desarrollado lo que ellos llaman el paradigma de 'detalles sobre demanda', el cual sugiere que la sobrecarga de información de un hipertexto puede reducirse de acuerdo a las siguientes consideraciones:

- Presentar al inicio sólo la información necesaria para asegurar la comprensión.
- Presentar en cada nodo sólo la información necesaria. Necesaria aquí significa que la presentación debe ser completa pero no elaborada. Debe identificar conceptos relevantes en términos de lo que el usuario puede entender.
- Proporcionar detalles sobre demanda. La información subordinada debe ser eliminada de una presentación, el lector pueda accederla después a través de una liga. Cuando sea necesario, hay que establecer una liga para que el usuario pueda obtener detalles de un punto específico.

III.2.2 Ligas.

Las ligas son la otra unidad fundamental del hipertexto, además de los nodos. Las ligas casi siempre parten de un punto determinado dentro del nodo para proveer al usuario de un objeto explícito que debe activar para seguir esa liga y navegar hacia otro nodo. Frecuentemente esta situación crea una especie de menús implícitos, donde una parte del texto del nodo cumple una doble tarea, al ser información en sí misma y brindar el punto de partida a la liga. Es posible también tener los puntos de partida separados en forma de menús formales, aunque esto no favorece la impresión de los usuarios de estar trabajando con un hipertexto.

La mayoría de las ligas son explícitas en el sentido de que han sido definidas por alguien, para conectar un nodo origen con un nodo destino. Sin embargo, también existen las ligas implícitas, las cuales no son definidas como tales pero existen a partir de diversas propiedades de la información. Un ejemplo típico es el glosario automático del sistema Intermedia, a través de una utilidad llamada InterLex. Esta provee una liga desde cualquier palabra de cualquier documento de Intermedia a la definición de esa palabra en un diccionario. Sin embargo, es obvio que sería imposible tener definidas todas esas ligas explícitamente. En vez de eso, sólo cuando el usuario solicita la definición de una palabra, el sistema encuentra el nodo destino para la liga, o sea, dicha definición.

Existen otro tipo de ligas, llamadas ligas calculadas, las cuales son determinadas por el sistema mientras el usuario está leyendo el hipertexto, en vez de ser definidas estáticamente por el autor. Un ejemplo de estas ligas lo representa el sistema StrathTutor [12]. Este sistema no espera que el autor establezca ligas entre nodos, sino más bien que defina un conjunto de atributos

relevantes para cada nodo y áreas de interés en el nodo. Estos atributos son palabras tomadas de un vocabulario predefinido. Las áreas de interés son llamadas 'hotspots' y sirven como especie de puntos de partida a las ligas. Cuando el usuario activa un 'hotspot', el sistema 'conocerá' el interés del usuario, definido por la combinación de los atributos del nodo actual y el 'hotspot' elegido. Entonces, establece una liga a un nuevo nodo en el coincidan lo más posible sus propios atributos y este conjunto de atributos.

Adicionalmente a las ligas estándar, que unen a dos nodos, existen sistemas de hipertexto que poseen 'super-ligas', las cuales conectan más de un par de nodos. Las dos opciones más comunes para manejar esta situación son presentar un menú de los nodos a los cuales se puede navegar si se activa la liga, o bien navegar a todos los nodos destino a la vez. En el caso de presentar un menú, es necesario nombrar adecuadamente cada nodo para que el usuario entienda claramente sus opciones.

Existen fundamentalmente dos métodos empleados por los sistemas de hipertexto para definir las ligas en un hiperdocumento:

1. Comandos incorporados al texto: Este sistema consiste en definir las ligas a través de comandos especiales para el efecto, que se intercalan dentro del texto del documento. Una vez creado, el documento se somete a un procesamiento que interpreta los comandos y presenta al lector el hiperdocumento en su formato final, de tal forma que cuando éste activa una liga, el sistema sabe a que nodo dirigirse para presentarlo. Existen dos tipos de comandos: procedurales y declarativos. Los procedurales especifican qué debe de hacer una liga: La siguiente palabra es una liga al nodo X. Los declarativos especifican qué es una liga: la siguiente línea

es un título de capítulo, y definen una serie de reglas y convenciones para ligar un título de capítulo a diferentes párrafos.

2. Manipulación directa: Este método consiste en establecer las ligas directamente sobre el texto, en la forma que lo haría un editor WYSIWYG (What-You-See-Is-What-You-Get); el autor selecciona un texto con el cursor o con un ratón, luego elige de un menú la opción de 'Crear liga'. Luego elige el texto destino y completa la operación.

Ambos sistemas tienen sus ventajas y desventajas. El segundo es sumamente fácil, no requiere ningún conocimiento sobre comandos o sintaxis, ni está sujeto a errores por parte del usuario. Sin embargo, el primer método ofrece el mayor poderío y versatilidad para definir ligas de hipertexto, y ofrece una posibilidad muy importante: es factible procesar textos muy grandes para convertirlos a formato de hipertexto a través de programas de computadora que analicen el texto e incluyan los comandos necesarios para definir las ligas automáticamente. Este esquema es particularmente útil cuando hablamos de textos estructurados en forma regular, como podría ser el caso de libros, manuales, reglamentos, procedimientos, que generalmente están organizados por capítulos, secciones, tópicos, etc.

III.2.3 Gráfica de información.

El lector de un hiperdocumento constantemente debe responder tres preguntas:

- ¿ Dónde estoy ?
- ¿ A dónde quiero ir ?
- ¿ Cómo puedo llegar ahí ?

Esta situación se conoce como el 'problema de la navegación', por su similitud con la navegación. Pues bien, es fácil suponer que mientras más intrincadas sean las rutas de navegación, será más difícil para alguien llegar a su destino. Lo mismo sucede en un hiperdocumento. Y la complejidad de las rutas está directamente asociada con su gráfica de información.

En un hiperdocumento, los elementos para determinar la complejidad son:

1. El número total de nodos.
2. El número total de ligas.
3. El número promedio de ligas que parten de un nodo.

Existen diversas técnicas para ordenar la gráfica de información, de modo tal que se reduzca el impacto de estos valores en la complejidad del hiperdocumento. Por ejemplo, algunas de ellas permite tratar a un gran número de nodos como si fueran uno solo, mientras que otras proveen formas de que el usuario ignore la mayoría de las ligas del sistema sin perder su trayectoria hacia la información requerida.

Un sistema de hipertexto es capaz de producir documentos específicos eligiendo una trayectoria de navegación y agrupando la información encontrada a cada paso. Este proceso se conoce como linealización, ya que reduce la gráfica de información a un documento lineal. Esto es relativamente fácil de hacer si la gráfica tiene una estructura adecuada.

Las técnicas para ordenar una gráfica pueden clasificarse en dos grandes grupos. El primero de ellos se basa en forzar ciertas estructuras de conexión

entre los nodos. El segundo consiste en definir diferentes tipos de ligas. A continuación hablaremos de cada uno de ellos.

Antes de describir los diferentes patrones que existen para ligar un conjunto de nodos, hablaremos de las diferentes estrategias que pueden seguirse para navegar dentro de un hiperdocumento, y lo haremos porque la forma de moverse a través de la estructura, está estrechamente relacionada con el tipo de estructura de que se trata. Utilizar una analogía para explicar estas estrategias resulta muy útil. Pensemos en una persona que camina en una ciudad a través de calles y avenidas, buscando un cierto lugar.

La Estrategia de Identificación consiste en dar un identificador único al lugar que queremos visitar, de tal modo que podamos reconocerlo cuando lo veamos (por ejemplo, "el edificio de color morado"). Por sí sola, esta estrategia ocasiona una búsqueda exhaustiva. Hay que visitar todos los lugares, uno por uno, hasta encontrar el que corresponda a la descripción. Sin embargo, con la ayuda de ciertos elementos (por ejemplo, un automóvil) la búsqueda puede ser bastante rápida. Es por ello que esta estrategia no se usa sola, sino como componente de otras.

La Estrategia de Trayectoria consiste en utilizar una descripción de cómo llegar, paso a paso, al lugar deseado ("Siga derecho hasta el segundo semáforo, después doble a la izquierda y a la segunda cuadra, es un edificio morado (estrategia de identificación) en la acera izquierda"). Esta estrategia es útil cuando el número de caminos que parten de un lugar es mucho menor que el número total de lugares, de otra forma el problema de seleccionar el siguiente tramo de la trayectoria se vuelve tan complejo como encontrar el lugar buscado. Algunas veces hay más de una trayectoria que lleva al mismo lugar, pero aún funciona si la persona tiene instrucciones claras que seguir.

La Estrategia de Rumbo consiste en seguir determinado rumbo ("siga hacia el norte"). Como la anterior, esta estrategia evita una búsqueda exhaustiva, pero se diferencia en el marco de referencia para las instrucciones. En la de trayectoria, utilizamos referencias relativas a una localidad determinada, como las calles que parten de un punto dado; en la estrategia de dirección, la referencia es absoluta, no depende del lugar en que estemos, y aplica para todos los lugares del área total. Las referencias que se utilizan varían en su precisión, desde decir norte o sur, hasta hablar de grados, minutos y segundos como en la navegación moderna. Por supuesto que debe existir algún punto de referencia válido a todo el sistema, para poder establecer estas referencias absolutas (como los polos de la tierra).

La Estrategia de Distancia consiste en buscar dentro de un círculo alrededor del lugar donde se encuentra la persona, y a menudo se usa en combinación con la estrategia de dirección ("está a diez kilómetros de aquí, vaya dos kilómetros al sur, luego ..."). Esta estrategia es de poca utilidad si se puede viajar directamente de cualquier lugar a otro.

La Estrategia de Dirección es una mejora de la de rumbo, al establecer un conjunto de coordenadas, como pueden ser latitud y longitud, o los nombres de las calles. Imaginemos un área de diez calles por diez calles. En ella habrá cien intersecciones, de tal modo que si buscáramos una siguiendo la estrategia de identificación, habría que recorrer, en el peor de los casos, los cien lugares. Por otro lado, con la estrategia de dirección, lo más que tendríamos que buscar son veinte lugares (las diez calles en cada dirección) para dar con la intersección, lo que representa un ahorro del 80% de esfuerzo. A medida que el área crece, el ahorro es más significativo. Y dependiendo de la complejidad, es posible agregar nuevas dimensiones (como el número de piso del edificio).

Finalmente, la Estrategia de Región consiste en centrar la atención en determinadas regiones alrededor de puntos de interés, ignorando lo que pueda haber entre ellas (por ejemplo, tomar como referencia una estatua en la calle, y a varias cuadras de ahí, un museo; no interesa lo que haya en medio).

Ahora bien, un factor muy importante cuando hablamos de navegación es el sentido de las trayectorias (no es lo mismo viajar por una calle de un sentido, que por una de dos). Cuando hablamos de una gráfica de información, llamamos a un nodo el padre de otro si una liga parte del primero hacia el segundo (de padre a hijo). En algunos sistemas, el lector sólo puede moverse en esa dirección. En otros, es factible viajar en sentido opuesto, pero sólo si se ha llevado nota de la trayectoria para poder regresar. En otros más, las ligas no tienen sentidos, y el lector puede moverse a su gusto, empezando por donde quiera. Esta característica de las ligas hace una gran diferencia durante la navegación, porque las ligas de un sentido dan al lector una sensación de viajar de arriba hacia abajo, mientras que las de dos sentidos no.

A continuación describiremos los diferentes tipos de gráficas de información, suponiendo que las ligas son de un sentido. Esto no significa que el lector no pueda viajar de un nodo hijo a uno padre, sólo quiere decir que antes de hacerlo, primero tuvo que haber viajado del nodo padre al hijo. Además, los nodos están conectados de tal forma que hay al menos un nodo por el cual se puede empezar la navegación y alcanzar cualquier otro nodo de la gráfica, de tal forma que podemos considerar que todos los nodos están conectados.

Gráfica Lineal.

Esta es la gráfica más simple, en ella cada nodo tiene como máximo un nodo padre y un nodo hijo. Existen dos casos posibles, como se ve en la figura III.1.

Si cada nodo tiene exactamente un padre y un hijo, los nodos forman un anillo. De otra forma, exactamente un nodo no tiene hijo y otro nodo no tiene padre, y la estructura es una cadena con dos nodos en sus extremos.

La gráfica lineal soporta todas las estrategias de navegación (la estrategia de trayectoria sería muy similar a la de identificación en este caso), aunque dada la simplicidad de la estructura, no son muy necesarias. De acuerdo a nuestra suposición inicial, todas las ligas son en el mismo sentido, lo que hace posible viajar a través de los nodos en uno u otro sentido. Hay una única distancia entre dos nodos cualesquiera, definida como el número de ligas entre ellos, y una única dirección para cada nodo.

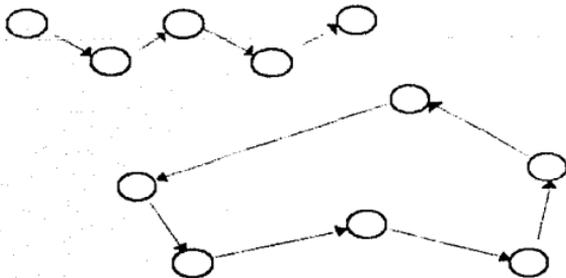


Figura III. 1. Gráficas lineales.

Gráfica Jerárquica.

En este tipo de gráfica, hay un nodo que no tiene padre y todos los demás tienen exactamente un nodo padre. Este nodo es la raíz de la estructura y usualmente es el punto de entrada al hiperdocumento. Todas las ligas apuntan en un sentido, por lo que es posible alcanzar cualquier otro nodo. Además, ya que se puede viajar en sentido inverso, es posible regresar desde cualquier nodo (hasta llegar a la raíz incluso) y luego bajar hacia cualquier otro. Como hay una única trayectoria entre dos nodos cualquiera, podemos definir distancias entre nodos, en términos de la profundidad en la estructura. Dado que los sentidos de navegación están claramente definidos (hacia arriba o hacia abajo) esta gráfica no soporta la estrategia de dirección. La figura III.2 muestra una gráfica jerárquica.

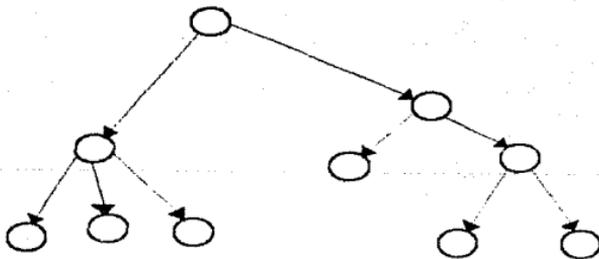


Figura III. 2. Gráfica Jerárquica.

Gráfica de Hiper cubo e Hipertoro.

En una gráfica de Hiper cubo, las ligas que intersectan a un nodo pueden provenir de dos, tres, cuatro o más conjuntos o dimensiones. En la figura III.3 se aprecia un hiper cubo de dos dimensiones.

Este tipo de gráfica soporta directamente la estrategia de dirección para navegación. Es útil para cuando es necesario comparar un cierto número de elementos a través de un cierto número de dimensiones. Podemos decir que un hipertexto basado en un hiper cubo de dos dimensiones es similar a una hoja de cálculo de texto, donde cada nodo es directamente adyacente a otros cuatro, uno en cada dirección. Existe una variante de esta gráfica. En un hiper cubo, si ligamos los nodos finales de cada cadena de tal forma que se convierta en anillo, obtenemos un hipertoro, que puede ser una forma más adecuada de representar un hiper documento que no hace diferencia entre primero o último elemento (figura III.3).

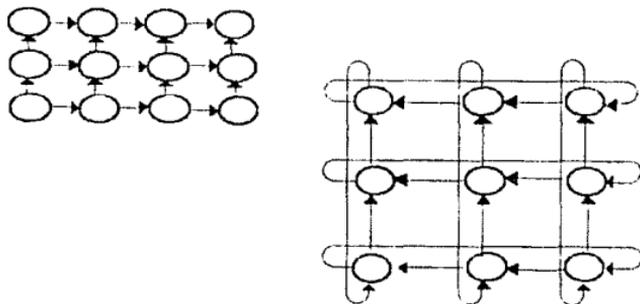


Figura III.3. Gráficas de Hiper cubo e Hipertoro.

En ambos casos, es clara la reducción en el número de nodos que hay que examinar para encontrar uno. Por ejemplo, si estructuramos cien nodos en una gráfica lineal, los lectores tendrán que navegar por un promedio de cincuenta nodos para encontrar uno en particular, pero si cambiamos la gráfica lineal por un hipercubo de dos dimensiones, la cantidad de nodos que hay que navegar se reduce a diez, y si aumentamos a tres dimensiones, habría que navegar por sólo cinco nodos. La gráfica de hipercubo soporta además las estrategias de navegación de trayectoria, la de rumbo y la de distancia.

Gráfica Dirigida Acíclica.

Este tipo de gráfica (DAG por sus siglas en inglés) es una estructura mucho más general que las que hemos estudiado hasta el momento. Sólo requiere que no haya ciclos en ella, es decir, que partiendo de un nodo y navegando en un sentido, es imposible regresar al mismo nodo. Las gráficas lineal, jerárquica y de hipercubo son DAG's; los anillos y los hipertoros tienen ciclos, por lo que no son DAG's. Muchos de los sistemas de hipertexto basados en esta gráfica tienen un sólo punto de entrada, formando un origen común para todos los nodos. Salvo que estén presentes otras regularidades, ni la estrategia de distancia ni la de dirección aplican en una DAG, pero la ausencia de ciclos implica que los sentidos de navegación están bien definidos, como trayectorias. La figura III.4 muestra una gráfica DAG.

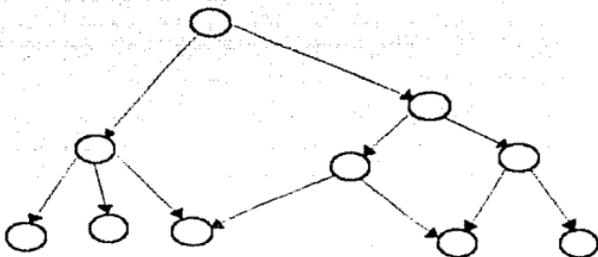


Figura III.4. Gráfica Dirigida Acíclica.

Gráfica Apilada.

Algunas veces una red de hipertexto consiste de un cierto número de regiones fuertemente relacionadas, entre las cuales existen muy pocas ligas. Esta estructura es la que conocemos como apilada ('clumped'), y soporta la estrategia de región. En este tipo de gráfica es común para el autor considerar cada región como un nodo, y determinar el tipo de gráfica que conecta estos nodos. Si se trata de una estructura regular, los lectores pueden llevar a cabo estrategias de navegación más poderosas para encontrar la región adecuada, y luego navegar libremente dentro de ella para encontrar el nodo buscado. La figura III.5 muestra una gráfica de este tipo.

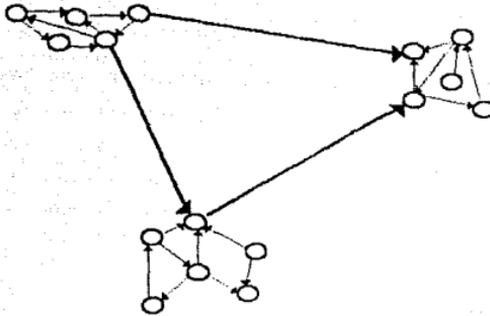


Figura III.5. Gráfica Apiñada.

Gráfica arbitraria.

El último tipo de gráfica consiste de cualquier número de nodos interconectados en cualquier forma, de ahí el nombre de arbitraria. Si ninguna de las restricciones mencionadas aplica, las estrategias de identificación y trayectoria son las únicas que aplican. Incluso si el número de ligas por nodo se incrementa, la estrategia de trayectoria pierde su utilidad. De esta forma, podemos distinguir entre una gráfica parcialmente arbitraria (con pocas ligas por nodo) y una gráfica totalmente arbitraria (con muchas ligas por nodo).

El siguiente diagrama resume las conclusiones acerca de qué topologías de gráficas de información soportan qué estrategias de navegación.

	IDENT.	TRAY.		RUM.		DIST.		DIR.	REG.
		Uniq	Mult.			Uniq	Mult.		
Lineal	x	x		x		x		x	
Anillo	x	x				x		x	
Jerárquica	x	x		x		x			
Hipercubo	x		x	x		x		x	
Hipertoro	x		x			x		x	
DAG	x		x	x			x		
Apilada									x
Arbitraria	x		x				x		

Fig. III.6. Gráficas y Estrategias de Navegación

III.3 Mecanismos de apoyo a la navegación.

A primera vista nos pueda parecer que la noción de estructura dentro de la forma en como se conectan los nodos en un hiperdocumento es completamente opuesta a una de las grandes fortalezas del hipertexto, o sea la habilidad de moverse libremente dentro de una red de información. Pues bien, existen diversos caminos para que los lectores de un hiperdocumento puedan tener las ventajas de navegar a través de una gráfica estructurada sin perder la flexibilidad que en primera instancia ofrece el hipertexto. Muchas de las ayudas para navegación comúnmente ofrecidas en los sistemas de hipertexto son mecanismos para inducir una topología de gráfica menos compleja en el hiperdocumento, y así ampliar el conjunto de estrategias de navegación que el lector puede aplicar. Estos mecanismos son de dos tipos: de "camino trillado" que ayudan a los lectores a encontrar lugares en la red que ya

han visitado, y de abstracción, que dan al lector una perspectiva diferente de un hiperdocumento complejo, con una estructura más simple.

Mecanismos de "camino trillado".

Estos mecanismos incluyen pilas de regreso, macros de trayectoria y separadores.

Las pilas de regreso llevan seguimiento de los nodos que el lector a visitado, de forma tal que se puede navegar hacia atrás hasta el punto de partida. Este mecanismo genera una gráfica menor con un patrón lineal, dentro del cual se tiene el menor rango posible de estrategias de navegación.

Una generalización de las pilas es lo que llamamos macro de trayectoria, la cual recuerda un camino seguido por el lector y le permite moverse hacia atrás o adelante tantas veces como lo desee. Esencialmente lo que hace es crear un subdocumento con una gráfica restringida, dentro de la cual la navegación es más simple. Generalmente la trayectoria almacenada es lineal o jerárquica. En caso de que se trate de una jerárquica, si el lector visita un nodo situado dentro de la macro siguiendo otra trayectoria, la liga anterior al nodo es eliminada de la macro y reemplazada por la nueva.

Finalmente los separadores permiten al lector marcar nodos dentro de la gráfica y viajar directamente a ellos desde cualquier punto del hiperdocumento. Este mecanismo induce una gráfica jerárquica de un nivel, cuya raíz es el nodo marcado.

Mecanismos de abstracción.

Estos mecanismos permiten al lector abstraerse de la complejidad del hiperdocumento total, y tener una nueva perspectiva reducida y menos compleja. Dos ejemplos son: guiones y ligas tipificadas.

Los guiones son trayectorias seleccionadas a través de un hiperdocumento que un usuario (quizá el autor) ha preparado para los lectores, generalmente lineales o jerárquicas. Como el lector tiene la facilidad de abandonar esta trayectoria en un punto dado para navegar por su cuenta, y regresar en forma inmediata al punto donde se quedaron, es muy fácil navegar a través de la estructura completa sin perderse.

A continuación veremos como las ligas pueden ser clasificadas en diferentes tipos. En primera instancia, esto puede aumentar el número de ligas en la estructura. Sin embargo, la gráfica formada por las ligas de un determinado tipo puede ser mucho más simple que la del hiperdocumento completo, y debido a que el lector puede identificar los diferentes tipos de ligas en todo momento, y pedir que solo esas se desplieguen, el hiperdocumento será más claro y fácil de leer.

Podemos distinguir entre tres clases de ligas que son usuales en hipertexto:

a) Ligas de asociación son las más comunes, y refleja diversas razones por las que un nodo "trae" otro nodo a la mente. Por ejemplo, un nodo puede dar la razón a las afirmaciones hechas en otro.

b) Ligas de agregación son aquellas que unen a un nodo que representa un todo con sus partes. Por ejemplo, un nodo representando un libro con los nodos representando sus capítulos, podrían ser unidos por ligas de agregación de tipo 'Capítulo'.

c) Ligas de revisión son aquellas que unen un nodo a versiones anteriores y posteriores del mismo nodo.

A continuación analizaremos más detalladamente cada uno de estos tipos de ligas, para posteriormente entender porqué pueden facilitar el uso de un hiperdocumento.

a) Ligas de Asociación.

Existen dos clases de ligas de asociación. Para entender la diferencia entre ambas, podemos utilizar su relación con las palabras. Si formamos grupos de palabras que tengan sentido, podemos hacer una clasificación entre aquellos grupos que contienen un verbo (a los cuales llamaremos cláusulas) y grupos que no tienen (que llamaremos palabras si son sólo una, o frases si son varias palabras). En realidad sólo las cláusulas dicen algo, proponen algo, por lo tanto constituyen una proposición.

Ahora bien, diversas cláusulas pueden combinarse para formar unidades mayores, llamadas párrafos, los cuales a su vez se combinan entre sí formando páginas, y capítulos, y así sucesivamente. Las mismas relaciones que unen a las cláusulas, unen a los párrafos, por lo que estas unidades más grandes siguen siendo proposiciones. De hecho, algo tan grande como un libro puede resumirse con una proposición. Lo mismo sucede en un hiperdocumento, en el cual una proposición puede variar en tamaño desde una cláusula hasta un nodo completo. Así, podemos establecer

dos clases de ligas de asociación dependiendo de las unidades que asocian: ligas palabra-proposición y ligas proposición-proposición.

Ligas palabra-proposición. Unen una palabra o frase con una proposición, y pueden ser de varios tipos:

- Liga de contenido, relaciona una palabra que nombra una proposición, con esa proposición. Por ejemplo, un nodo podría contener la frase 'Himno Nacional' y conectarse con una liga de contenido a otro que mostrara el texto del himno.

- Liga de identificación, que conecta una palabra con su definición, restringiendo su significado; el ejemplo típico es un glosario de términos.

- Liga de comentario, que da información adicional sobre una palabra o frase, pero no restringe su significado; por ejemplo, una liga de comentario podría relacionar el nombre de una persona con sus datos generales.

Ligas proposición-proposición. Estas ligas son de cuatro tipos, cada uno de ellos con varios tipos más.

- Ligas de orientación: definen el ambiente para el contenido de la proposición, y son de tres tipos:

- Liga de lugar, que une una proposición con el lugar donde aplica

- Liga de tiempo, que describe la relación cronológica

entre dos proposiciones

- Liga de circunstancia, que une una proposición a otra que describe su ambiente, excluyendo el lugar y tiempo

- Ligas de implicación: representan relaciones lógicas entre proposiciones y son de siete tipos:

- Liga de causa, que une una proposición con su causa

- Liga de propósito, que une una proposición con otra que describe su propósito

- Liga de condición, que une una proposición con otra cuya validez asegura la validez de la primera

- Liga de contrahecho, que une una proposición con otra cuya validez asegura la falsedad de la primera

- Liga de concesión, que relaciona una proposición a otra que podría pensarse que invalida a la primera, pero no lo hace

- Liga de advertencia, que relaciona una proposición que describe una obligación, con una consecuencia indeseable si no se cumple la primera

- Liga de evidencia que conecta una proposición con otras

que la soportan, como especie de síntomas

- Ligas de paráfrasis: conectan dos proposiciones que contienen información similar, y hay cinco tipos:

- Ligas de ampliación y resumen, que son ligas inversas entre sí, la primera conecta una proposición con sus detalles, y la segunda a la inversa (pueda pensarse en una especie de tabla de contenidos)

- Ligas de abstracción a instancia, que son inversas entre sí y conectan una proposición genérica con una forma específica de la misma

- Ligas de equivalencia, que unen dos proposiciones con el mismo contenido y diferente forma

- Ligas de ilustración: conectan proposiciones diferentes, pero donde una clarifica a la otra. Hay de tres tipos:

- Liga de método, que liga una proposición con otra que describe la forma en que se llevó a cabo

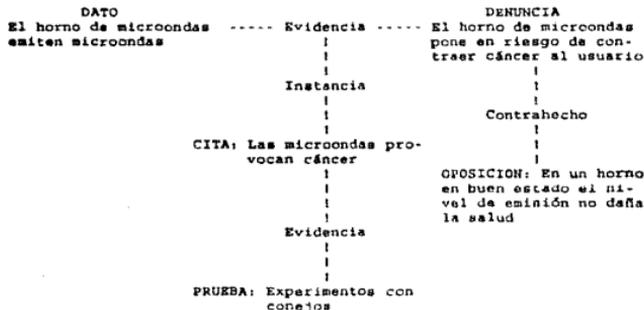
- Liga de comparación, que une dos proposiciones que son diferentes pero enfatiza puntos de interés común

- Liga de contraste, que une proposiciones diferentes y enfatiza las diferencias entre ambas

b) Ligas de Agregación.

Este tipo de ligas nos permiten relacionar un todo con sus partes. A menudo habrá otras ligas entre las partes, de asociación o de revisión, así como entre las partes y otros nodos que no forman parte del mismo todo.

Un ejemplo de este tipo de ligas puede ser el caso de un argumento lógico. De acuerdo con la estructura de Toulmin [13], cualquier argumento puede ser analizado en cinco componentes básicos que son: dato, denuncia, cita, prueba y oposición. En un hiperdocumento, un nodo puede apuntar al argumento, y este a su vez tener ligas de agregación hacia sus componentes. Analicemos el siguiente ejemplo:



En este caso, la representación en hipertexto podría tener un sólo nodo representando "Resuelto: el horno de microondas es un riesgo de cáncer". Este nodo estaría unido con una liga de agregación de tipo DENUNCIA a otro nodo que tuviera la denuncia; con otra liga de tipo DATO a un nodo conteniendo el dato, y así sucesivamente. Podemos notar que los componentes del argumento están unidos entre sí a través de ligas asociativas de diversos tipos, que permiten viajar a través de

nodos del mismo nivel, mientras que las ligas de agregación permiten viajar del todo a sus partes.

c) Ligas de revisión.

Es frecuente la necesidad en cualquier tipo de información, de almacenar versiones sucesivas. Podemos hablar de las revisiones de un programa de computadora para eliminar sus errores, o de los bosquejos de un contrato durante una negociación. Pues este tipo de ligas son la forma del hipertexto de relacionar nodos con esta característica. Las ligas podrían tener los tipos PREDECESOR y SUCESOR, por ejemplo, para ligar la versión anterior y la posterior de un nodo, respectivamente.

Las ligas definidas con cierto tipo, como lo hemos visto a lo largo de las clasificaciones, representan una herramienta para manipular un hiperdocumento, y una forma de reducir su complejidad. Por ejemplo, una de sus principales aplicaciones es lo que conocemos como filtros y acercamientos (zoom's). En el primer caso, el sistema de hipertexto muestra solamente algunas de las trayectorias que pueden seguirse a través de él. Cuando no es posible distinguir entre ligas de diferentes tipos, el sistema sólo tiene dos opciones: mostrar todas las ligas que salen de un nodo o no mostrar ninguna. Cuando las ligas están clasificadas, es factible reducir la complejidad de la gráfica de información, y esto permite al lector concentrarse en la información que le interesa. En el caso del acercamiento, a través de ligas de agregación como ya lo vimos en el ejemplo, es posible tratar un conjunto de nodos relacionados entre sí como uno solo, sin importar su

estructura interna de conexión, y luego navegar hacia dentro del conjunto, para observar las relaciones específicas entre sus componentes.

IV. DISEÑO CONCEPTUAL DEL SISTEMA DE HIPERTEXTO.

IV.1 Consideraciones generales.

La tecnología de hipertexto proporciona nuevos métodos para organizar y presentar información. En términos generales, hablar de hipertexto implica un conjunto de elementos de texto por un lado, y de un grupo de relaciones entre estos elementos, que pueden ser ordenadas en muchas formas y usadas para satisfacer un amplio rango de necesidades particulares de consulta a la información.

Más formalmente, el hipertexto tiene tres componentes fundamentales:

1. Una base de datos de información textual
2. Una red semántica que asocia los componentes textuales
3. Herramientas para crear y manipular esta combinación de componentes y relaciones

Así pues, por un lado podemos identificar al Sistema de hipertexto, refiriéndonos al conjunto de herramientas (programas de computadora) que nos permiten manipular una estructura de información: crearla, modificarla, navegarla, etc. Por otro lado, tenemos al hiperdocumento que es precisamente esta red de nodos de información unidos a través de ligas. Fuec bien, la utilidad de una implementación particular de hipertexto está determinada por la combinación de ambos elementos y sus características. Podemos tener una excelente herramienta de hipertexto, pero si no logramos construir un documento estructurado, comprensible, cuyos nodos contengan la información necesaria y estén relacionados en forma

coherente, no habremos creado una aplicación útil. En este capítulo nos orientaremos a definir las características de diseño de la herramienta de hipertexto para su posterior implementación. En el capítulo Ejemplo de aplicación, presentaremos una metodología para la planeación y desarrollo de un hiperdocumento, que es un tanto independiente de la herramienta que se emplee, si bien en este caso utilizaremos el sistema desarrollado.

Para determinar la estructura del sistema de hipertexto, nos basaremos en el Modelo Dexter. Este modelo ha sido producto del llamado Grupo Dexter, formado por muchos de los diseñadores de los primeros sistemas de hipertexto. Las dos juntas que dieron origen a este modelo fueron organizadas por Jan Walker y John Legget y auspiciadas principalmente por Digital Equipment Corporation y la Universidad A&M de Texas. El objetivo de este modelo es describir la arquitectura general de un sistema de hipertexto, y por ende proporcionar la base para implementar un formato estándar de intercambio de información entre sistemas de hipertexto [14].

IV.2 El Modelo Dexter de Hipertexto.

El modelo Dexter es un modelo representativo y general de un sistema de hipertexto. El modelo está compuesto por tres niveles separados:

1. Nivel de almacenamiento (storage layer): define como los componentes textuales y las ligas pueden ser relacionados para formar redes de hipertexto.

2. Nivel de componente (within-component layer): provee un mecanismo para definir la estructura interna y el contenido de cada componente.

3. Nivel de ejecución (runtime layer): provee un mecanismo para definir como los componentes del hipertexto y las ligas serán presentados y manipulados.

El modelo Dexter también establece mecanismos de relación entre niveles adyacentes: el nivel de almacenamiento y el de componente se comunican a través del mecanismo de "anchoring" o anclaje, mientras que el nivel de ejecución y el de almacenamiento lo hacen a través de las especificaciones de presentación.

La figura IV.1 muestra los tres niveles del modelo.

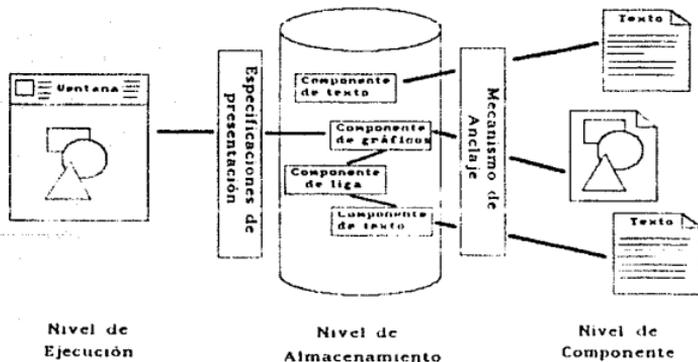


Figura IV.1. Modelo Dexter de Hipertexto.

El Nivel de Almacenamiento.

El modelo Dexter hace énfasis en este nivel y en los mecanismos de anclaje. La unidad fundamental del nivel de almacenamiento es el componente. Un componente es una entidad sintáctica y semánticamente discreta que representa un concepto independiente y que puede ser clasificada y vista en diferentes formas. Un nodo de un hiperdocumento, del que hemos hablado a lo largo de esta tesis, está definido dentro de este modelo como un componente.

El modelo Dexter no impone restricciones en cuanto a la composición o función de los componentes. Un componente puede consistir de texto, gráficas u otras formas de información. El único tipo de componente crítico para la definición del hipertexto es la liga. Una liga relaciona uno o más componentes entre sí. La conexión entre una liga y un componente se establece usualmente a través de una porción del componente llamada ancla (anchor). Así, un ancla queda definida en forma única por un identificador global del componente o nodo dentro del sistema y un área específica dentro del componente. Esto permite la modificación de un componente sin violar la integridad de otros componentes o ligas.

El Nivel de Componente.

El modelo Dexter da libertad a los desarrolladores para definir nuevos tipos de componentes, facilitando la transición de sistemas que contienen sólo texto, a sistemas que contienen texto y otros tipos de información, como imágenes o sonidos.

Para cada nuevo tipo de componente, es necesario definir dentro de este nivel un método para representar tanto la nueva entidad como los mecanismos de anclaje

para la entidad (esto significa la forma en que internamente se representaran las ligas hacia este nuevo componente, dentro de los demás).

El Nivel de Ejecución.

En este nivel se toman las decisiones relacionadas con el aspecto de un componente de hipertexto, a través de las especificaciones de presentación. Es en este nivel donde se determina la correspondencia entre la representación del texto y la presentación del mismo. Por ejemplo, si se establece una relación de uno a uno entre el nivel de almacenamiento y el de ejecución, esto significa que cada componente tendrá una y solo una "unidad" de presentación. Así pues, podemos representar un componente dentro de una ventana, y cada componente tendrá asociada una ventana de despliegue con capacidad de paginación para poder manejar componentes de tamaño mayor al de la ventana. Por otro lado, se pueda manejar el concepto de documento estructurado, en el cual se combinan múltiples componentes del nivel de almacenamiento. Esto requiere desarrollar especificaciones de presentación más complejas. Un ejemplo de esta filosofía es el documento compuesto, el cual presenta en una misma unidad diversos tipos de información (texto, imágenes, gráficas, etc.). Los marcadores de liga (es decir, las porciones dentro de un componente que identifican la presencia de ligas hacia otros componentes) están también definidas en este nivel, a través de especificaciones de presentación. Existen diversas opciones de diseño: utilizar tipos de texto resaltado en la porción que represente un marcador, o colocar los marcadores en áreas específicas de la ventana desplegada, o cambiar la forma del cursor cuando el usuario lo posiciona sobre un marcador, o utilizar íconos para esquematizar la presencia y tipo de liga.

Además de basar la estructura del sistema en los conceptos anteriores, es necesario considerar algunos lineamientos que nos ayudarán a guiar el diseño hacia la construcción de una herramienta útil.

Empecemos estableciendo los dos papeles que puede jugar el usuario de un sistema de hipertexto durante su interacción con el sistema. Por un lado, hablamos del autor del hiperdocumento, que se encarga de crear nodos y asociarlos entre sí con el propósito de poner esta información a disposición de otros usuarios, y por otra parte nos referiremos al lector, cuya actividad básica es la consulta de esta información. Para cada uno, el término utilidad adquiere un significado particular. Sin embargo, el beneficio que ambos obtengan de una herramienta dependerá en buena medida de los siguientes factores:

Facilidad de uso.

Implica que el lector rápidamente pueda utilizar la herramienta para realizar sus tareas. Un sistema de hipertexto debe ser fácil de aprender, de modo que los lectores puedan entender los comandos básicos y las opciones de navegación y usarlos para obtener la información que desean. Cuando alguien entra al sistema por primera vez, debe ser capaz de entender la primera pantalla, y obtener información de ella, además de poder comprender fácilmente la estructura del hiperdocumento. Además, la información contenida en cada nodo debe ser comprensible.

Eficiencia.

Una vez que el usuario ha aprendido a usar el sistema, suponiendo que desea obtener determinada información, debe ser capaz de llegar a ella rápidamente, o darse cuenta de que no existe dentro del hiperdocumento. Cuando el usuario llega a un nodo, debe ser capaz de orientarse fácilmente y entender su significado en relación con el nodo del que partió. En cuanto al autor, este debe ser capaz de

construir fácilmente una estructura de información, que pueda ser mantenida y modificada sin problemas.

Sencillos.

Un usuario casual es capaz de utilizar el sistema después de algún tiempo de no hacerlo, sin necesidad de aprender todo de nuevo. El usuario no tiene problemas para recordar los comandos básicos para navegar dentro del hipertexto, y además puede familiarizarse rápidamente con un hiperdocumento en particular. Así mismo, puede transferir este conocimiento acerca del uso y navegación dentro del sistema a otros usuarios. Refiriéndonos a los autores, para ellos es fácil regresar a una aplicación después de algún tiempo y revisarla o actualizarla sin problema. El autor recuerda la estructura básica de la información, o bien es ayudado por el sistema para hacerlo.

Precisión.

El sistema no permite a los usuarios cometer muchos errores, y en caso de que un usuario cometa alguno, es fácil para él corregirlo sin grandes pérdidas. En caso de que un usuario siga una liga erróneamente, puede regresar a nodos anteriores sin mayor problema. Para un autor, el sistema de hipertexto detecta los errores más comunes y le permite corregirlos, por ejemplo, el caso de ligas que no conducen a ningún lugar.

Aceptación.

Esto implica que los usuarios se sienten satisfechos con el sistema, les guste utilizarlo. El usuario prefiere usar el sistema de hipertexto, contra otras alternativas para la consulta de su información. Un usuario no se siente decepcionado de sus resultados al navegar dentro del sistema, por el contrario, siente que tiene control sobre la estructura de información, y que puede moverse

libremente dentro de ella para obtener su información. Para un autor, también es importante sentirse cómodo al utilizar el sistema, y contar las herramientas necesarias para crear fácilmente sus documentos.

A continuación estableceremos los criterios fundamentales que guían el diseño de nuestra herramienta de hipertexto, planteando el funcionamiento del sistema, los comandos para que el usuario (autor y lector) hagan uso de él, sus interfaces, etc., tratando de asegurarnos en todo momento a los lineamientos que acabamos de enumerar.

IV.3 Descripción conceptual del sistema.

Para efectos de facilitar la lectura de esta tesis, de aquí en adelante nos referiremos a nuestra herramienta con el nombre de Hipardoc.

Hipardoc es una herramienta que permite crear o desplegar información en línea utilizando la tecnología de hipertexto.

Hipardoc está diseñado en base al modelo Dexter de Hipertexto, por lo que podemos definir su estructura en base a los tres niveles anteriormente descritos:

Nivel de Almacenamiento.

Los componentes fundamentales de la red de hipertexto son:

- nodos de texto

- nodos no textuales
- ligas

Cada uno de los nodos de la red se almacena físicamente en un archivo de texto de tipo ASCII. Esto significa que se pueden generar nodos para el sistema a través de cualquier editor convencional de textos, lo cual ofrece beneficios importantes: la portabilidad entre diferentes ambientes, la estandarización y la facilidad de creación y actualización de hiperdocumentos. Un archivo puede almacenar uno o varios nodos. Esto es decisión del autor, y dependerá de sus requerimientos. Esto permite que varios autores trabajen simultáneamente en un mismo documento, y posteriormente integren sus componentes.

La diferencia entre los nodos textuales y los no textuales, es que los primeros contienen en sí mismos la información que será visualizada por el usuario durante la consulta de los documentos, mientras que los segundos tienen referencias a información externa al documento, como pueden ser gráficas, hojas de cálculo, video, audio, etc. Esto permite la integración de información de diferente naturaleza dentro del mismo documento, enfatizando así el concepto de hiperdocumento e hipermedia.

Las ligas son los componentes que nos permiten ligar entre sí los diferentes nodos, textuales y no textuales, para así crear un hiperdocumento. Una liga "parte" de un nodo, y llega a otro. El nodo origen de una liga sólo puede ser un nodo de texto, mientras que el nodo destino puede ser tanto un nodo textual como un no textual. Las ligas forman parte de los nodos que les dan origen, es decir que no están almacenadas por separado. En el nodo origen, la liga está identificada a

través de una porción de texto del nodo. A su vez, la liga apunta hacia el nodo destino en forma completa.

Nivel de Componente.

La estructura interna de un nodo, sea éste textual o no, es la misma. Fundamentalmente, un nodo está definido dentro de su almacenamiento, por dos delimitadores que marcan el principio y el final del nodo. Estos delimitadores en realidad son comandos que el usuario tiene a su disposición y que son interpretados por el sistema. Además de definir un nodo, el objetivo de estos comandos es establecer en forma dinámica las características de cada nodo, siempre dentro de los criterios establecidos en el Nivel de Ejecución.

Así pues, un nodo está compuesto fundamentalmente por dos clases de información:

- comandos
- texto

Los comandos tienen una sintaxis definida, y serán analizados a detalle en la sección de descripción funcional. Sin embargo, a modo de resumen, diremos que cada uno consta de una palabra clave que identifica al comando, y una serie de parámetros que establecen su ejecución. Cada comando es interpretado por el sistema durante la consulta del hiperdocumento. Esto implica que el documento permanece en su formato "fuente". El beneficio que este esquema proporciona está muy relacionado con la portabilidad y fácil integración de nuevos documentos a la red de hipertexto. Es decir, es posible tomar un documento en formato ASCII generado por

cualquier editor, y agregarle los comandos de Hiperdoc necesarios para "convertirlo" en un nuevo nodo. Incluso, dependiendo del nivel de estructuración del documento, es posible automatizar este proceso, de forma que a través de un programa escrito en cualquier lenguaje, se inserten los comandos necesarios dentro del texto.

Dentro de un nodo de texto, además de los comandos, están contenidas las ligas que permitirán al usuario la navegación a través de la red de hipertexto. Una liga está representada dentro del nodo por una porción de texto, que se delimita a través de caracteres especiales predefinidos por el sistema. Durante la consulta, estos caracteres se interpretan y dan lugar a lo que el usuario ve como "marcadores" de ligas. Una liga puede estar identificada por cualquier texto dentro de cualquier línea del nodo.

Para el caso de nodos no textuales, estos contienen fundamentalmente los comandos para tener acceso a la información externa asociada al nodo.

Nivel de Ejecución.

Hiperdoc es un sistema basado en ventanas, lo cual implica una relación de uno a uno con el nivel de almacenamiento. Por cada nodo de texto existirá una ventana para su representación al usuario. Esta ventana puede ser definida por el autor para cada uno de los nodos de texto que formarán el hiperdocumento. La forma de hacer esta definición es a través de los comandos que ya hemos mencionado. De esta forma, la información contenida en el nodo es desplegada en su correspondiente ventana, y existirán mecanismos de paginación y movimiento por línea, ya que en muchos casos el nodo no podrá ser desplegado en una sola página.

Dentro de un nodo, las ligas de hipertexto para navegar hacia otros nodos se asocian a un texto cualquiera dentro del nodo, de tal forma que cuando el lector selecciona este texto, viajará hacia el nodo destino definido para esa liga. Para que el lector identifique las ligas que han sido definidas dentro de un nodo, los textos correspondientes aparecen resaltados. Esta característica también puede ser definida por el autor a través de comandos.

Dentro de Hiperdoc es posible definir diferentes tipos de ligas. Como ya lo hemos discutido en el capítulo anterior, esto representa un mecanismo de ayuda para la navegación, de tal forma que el usuario puede solicitar en un momento dado, el desplegar solamente determinado tipo o tipos de liga, y reducir la complejidad del hiperdocumento. En Hiperdoc es posible definir hasta 3 tipos de ligas, y cada tipo se identifica con un color diferente que informa al usuario sobre el tipo de liga que está seleccionando.

Para el caso de nodos no textuales, las características de presentación de la información están determinadas por los sistemas propietarios de la información, y el autor dentro de Hiperdoc no tiene control sobre ella. Esto permite la integración de prácticamente cualquier herramienta.

IV.4 Descripción funcional del sistema.

Hemos planteado la estructura conceptual del sistema de hipertexto. A continuación, presentaremos el diseño del sistema desde el punto de vista funcional.

En la figura IV.2 se muestra una pantalla típica de Hiperdoc, que muestra una ventana correspondiente a un nodo cualquiera.

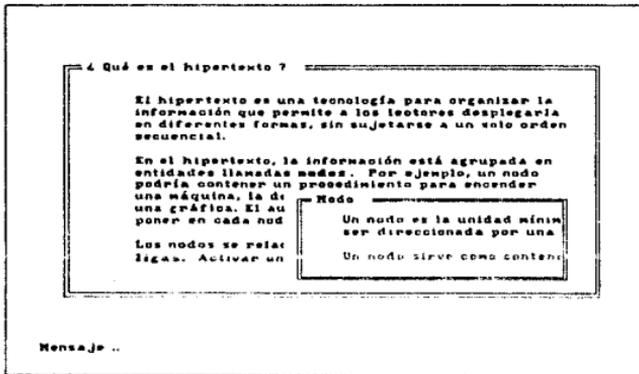


Figura IV.2. Pantalla de consulta.

En este caso, el usuario activó la liga nodos para navegar hacia otro nodo. Este nodo aparece en la ventana sobrepuesta, y contiene la definición de nodo. El tamaño y posición de la ventana dentro de la pantalla es decisión del usuario. El área que aparece como Mensaje .. sirve para notificar al usuario de errores.

Hiperdoc tiene una serie de comandos que pueden ser ejecutados por el usuario durante la consulta, y que pueden ser desplegados en cualquier momento solicitando Ayuda al sistema. La forma de ejecutar cualquier comando es a través de oprimir una tecla correspondiente a cada uno de ellos. Estos comandos permiten navegar dentro del hiperdocumento, solicitar ayuda al sistema o moverse dentro de un nodo. Los comandos permitidos se listan a continuación, clasificados en dos grupos: los comandos que son ejecutados oprimiendo la letra inicial (por ejemplo, A para invocar la Ayuda) y aquellos que son ejecutados con una tecla simbólica (por ejemplo, para avanzar línea por línea, se utiliza la flecha hacia abajo).

Comando	Acción
Ayuda	Despliega información acerca del uso de Hiperdoc, comandos, etc.
Previo	Permite al usuario navegar hacia el nodo previo al que está activo
Prim	Permite al usuario navegar hacia el primer nodo
Salir	Permite al usuario salir del sistema
Imprimir	Permite al usuario imprimir el contenido del nodo activo
Cambiar	Permite al usuario refrescar una ventana después de activar o desactivar tipos de ligas
Buscar	Permite al usuario buscar un texto dentro del nodo activo
Ligas	Permite al usuario activar y desactivar ciertos tipos de ligas durante la consulta

Comando	Acción
Flecha Arriba	Permite al usuario retroceder línea por línea dentro del nodo activo
Flecha Abajo	Permite al usuario avanzar línea por línea dentro del nodo activo
Flecha Izquierda	Permite al usuario moverse a la izquierda dentro del nodo activo
Flecha Derecha	Permite al usuario moverse a la derecha dentro del nodo activo
Av. Pag.	Permite al usuario avanzar página por página dentro del nodo activo
Re. Pag.	Permite al usuario retroceder página por página dentro del nodo activo
Tabulador Derecha	Permite al usuario avanzar a la siguiente liga dentro del nodo activo
Tabulador Izquierda	Permite al usuario retroceder a la liga anterior dentro del nodo activo
Return	Permite al usuario navegar a la liga activa dentro del nodo activo

IV.4.1 Creación de documentos en Hiperdoc.

Como ya vimos Hiperdoc emplea la filosofía de comandos incorporados al texto para la definición de ligas de hipertexto.

Un documento de Hiperdoc puede estar compuesto por uno o más archivos de texto. Un archivo de texto para Hiperdoc contiene dos tipos de información: el texto propio del documento, y los comandos que definen la estructura del hiperdocumento. En forma detallada, los comandos permiten:

1. Agrupar la información en nodos.
2. Identificar las ligas que asociarán estos nodos.
3. Controlar la presentación de la información al usuario (tamaño de las ventanas, posición en la pantalla, colores).
4. Ejecutar comandos para navegar hacia otros nodos.
5. Ejecutar aplicaciones fuera de Hiperdoc.

Estos comandos tienen algunas reglas básicas de sintaxis:

1. Todos empiezan con ':' en la primera columna de la línea.
2. Después del comando, pueden escribirse opciones y parámetros en cualquier orden.

Hemos dicho que una liga se asocia a un texto cualquiera dentro del nodo. Pues bien, para indicar que un texto tiene asociada una liga se utiliza un delimitador. Como es posible definir diferentes tipos de ligas, cada tipo tiene asociado un carácter diferente, dentro de un conjunto predefinido. Por ejemplo, para la liga de tipo 'definición' el delimitador puede ser '^', y para la de tipo 'comentarios' puede ser '~'. De esta forma, basta con delimitar la palabra o frase con este símbolo para definir una liga de hipertexto. Es obvio que podemos incluir estos delimitadores en cualquier lugar dentro del nodo, dentro de una misma línea. El texto que es utilizado para marcar una liga es el que aparece resaltado durante la consulta.

A continuación se listan los comandos permitidos en Hiperdoc para la creación de documentos, agrupados de acuerdo con el tipo de función que realizan.

1. Comandos para definición de nodos.

Comando	Acción
;inicio	Marca el inicio de un nodo
;fin	Marca el final de un nodo

2. Comandos para definición de ventanas.

Comando	Acción
;ventana	Define el tamaño, colocación y color de la ventana para un nodo
;titulo	Define el título para la ventana
;texto	Define los colores para el texto y las ligas dentro de la ventana
;cursor	Define el color de la liga donde está el cursor

3. Comandos para definir ayudas de navegación.

Comando	Acción
;accion	Permite la ejecución automática de uno o más comandos de Hiperdoc, como Previo o Primero
;regresa	Permite definir el regreso a un nodo particular

4. Comandos diversos.

Comando	Acción
;archivo	Indica en que archivo se encuentra un nodo
;registra	Registra la frecuencia con que se visita un nodo
;ejecuta	Permite correr una aplicación externa sin salir de Hiperdoc
;ligas	Permite determinar los nombres para cada uno de los tipos de liga. Se permite solo en el perfil del hiperdocumento

IV.4.2 Perfil de un Hiperdocumento.

En Hiperdoc es posible crear un perfil para cada hiperdocumento, en el cual se definen características generales que aplican a todos los nodos del mismo, y que son utilizadas por omisión, a menos que se definan otras diferentes dentro del nodo, en cuyo caso éstas tienen prioridad. El perfil de un hiperdocumento es un archivo con el mismo nombre que el archivo de hipertexto, pero con la extensión .PRO y en él es posible codificar algunos de los comandos presentados anteriormente:

```
,ventana
,texto
,cursor
,ligas
```

De estos comandos, el único que no puede ser codificado dentro de un nodo es el de ,ligas, ya que para todos los nodos de un hiperdocumento, los tipos de ligas que pueden ser definidas deben ser iguales.

La existencia de un perfil para el hiperdocumento es opcional. En caso de no existir, el producto contiene definiciones por omisión para todos los parámetros.

IV.4.3 Descripción detallada de los comandos.

A continuación presentamos la descripción detallada de los comandos.

1. Comandos para definición de nodos.

Comando ;INICIO

Este comando marca el inicio de un nodo, y es requerido para cualquier nodo.

Su sintaxis es:

```
;inicio(liga)
```

Parámetro

Significado

(liga)

Es el nombre de la liga que conduce a este nodo.
Es requerido, y debe escribirse tal como aparece,
con espacios y signos de puntuación.

Comando ;FIN

Este comando marca el final de un nodo, y es requerido para cualquier nodo.

Su sintaxis es:

```
;fin(liga)
```

Parámetro

Significado

(liga)

Es el nombre de la liga que conduce a este nodo.
Es requerido, y debe escribirse tal como aparece,
con espacios y signos de puntuación.

2. Comandos para definición de ventanas.

Comando `VENTANA`

Este comando controla el tamaño, posición y color de la ventana para un nodo.

Su sintaxis es:

```
,ventana(liga) fg=color bg=color reni=nn coli=nn nren=nn ncol=nn
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (ventana (.....)) a menos que se utilice el mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
fg=color	El color de la ventana ("foreground") : negro rojo café blanco cyan verde azul magenta gris Por omisión, es verde.
bg=color	El color de la ventana ("background") : negro rojo café gris cyan verde azul magenta Por omisión, es negro.
reni=nn	El número de renglón donde inicia la ventana, nn puede valer de 1 a 23. Por omisión, la ventana inicia en el renglón 1.
coli=nn	El número de columna donde inicia la ventana, nn puede valer de 1 a 78. Por omisión, la ventana inicia en la columna 1.
nren=nn	El número de renglones de la ventana, nn puede valer de 1 a 22. Por omisión es 20.
ncol=nn	El número de columnas de la ventana, nn puede valer de 1 a 78. Por omisión es 78.

Comando ,TITULO

Este comando controla el texto que aparece como título de un nodo, en el marco superior izquierdo de la ventana. Su sintaxis es:

```
,titulo(liga) texto
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (titulo {} ...) a menos que se utilice el mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
texto	Es el título deseado. Por omisión, se utiliza el nombre de la liga como título del nodo.

Comando ,TEXTO

Este comando controla los colores usados para las ligas y el texto. Su sintaxis es:

```
,texto(liga) fg=color bg=color lfgn=color lbgn=color
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (texto {} ...) a menos que se utilice el mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
fg=color	El color del texto ("foreground") : negro rojo café blanco cyan verde azul magenta gris Por omisión, es gris.
bg=color	El color del texto ("background") : negro rojo café gris

cyan verde azul magenta

Por omisión, es negro.

lfgn=color	El color del texto ("foreground") de liga tipo n: negro rojo café blanco cyan verde azul magenta gris n puede valer de 1 a 3. Por omisión, para 1 es rojo, para 2 es verde y para 3 es blanco.
lbgn=color	El color del texto ("background") de liga tipo n: negro rojo café gris cyan verde azul magenta n puede valer de 1 a 3. Por omisión es negro para todos los casos.

Comando ;CURSOR

Este comando controla los colores de la liga en donde se encuentra posicionado el cursor. Su sintaxis es:

```
;cursor(liga) fg=color bg=color
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (cursor {} ...) a menos que se utilice al mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
fg=color	El color del texto ("foreground") : negro rojo café blanco cyan verde azul magenta gris Por omisión, es rojo.
bg=color	El color del texto ("background") : negro rojo café gris cyan verde azul magenta Por omisión, es cyan.

3. Comandos para definir ayudas de navegación.

Comando ;ACCION

Este comando permite crear ligas que automáticamente ejecuten comandos de Hiperdoc cuando el usuario las seleccione :

- Comandos de navegación como Previo o Primero
- Comandos de paginación dentro del nodo

Su sintaxis es:

```
;accion(liga) ac
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (accion () ...) a menos que se utilice el mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
ac	Es la abreviatura para los comandos de Hiperdoc. Los valores permitidos son: FV Nodo Previo AI Avanza página SA Salir RP Regresa página PR Nodo Primero AL Avanza línea AI Avanza izq. RL Regresa línea AD Avanza der.

Comando ;REGRESA

Este comando permite crear una liga para regresar a un nodo en particular cuando el usuario la selecciona. Su sintaxis es:

```
;regresa(liga) nodo
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (regresa () ...) a menos que se utilice el mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
nodo	Es el nombre del nodo al cual se quiere regresar. Este nodo debe estar desplegado en ese momento en la pantalla. Si no es así, se regresa al primer nodo.

4. Comandos diversos.

Comando ;ARCHIVO

Este comando controla el nombre del archivo que contiene un nodo, y se utiliza sólo cuando el nodo no está en el mismo archivo que la liga que conduce a él.

Su sintaxis es:

```
;archivo(liga) archivo.ext nodo
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (archivo () ...) a menos que se utilice el mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
archivo.ext	Es el nombre del archivo que contiene el nodo. Puede indicarse un nombre completo de archivo, con los subdirectorios en que se encuentra.
nodo	Es el nombre del nodo al cual se va a navegar. Sólo es necesario cuando este nombre es distinto del nombre de la liga que conduce a él.

Comando ;REGISTRA

Este comando permite registrar la frecuencia con la que se selecciona un determinado nodo. Cuando alguien seleccione un nodo que tiene este comando,

se genera una anotación a un archivo. Su sintaxis es:

```
;registra(liga) archivo.ext
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (registra () ...) a menos que se utilice el mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
archivo.ext	Es el nombre del archivo en el que se desea registrar las ocurrencias. Puede indicarse un nombre completo de archivo, con los directorios en que se encuentra. En este archivo se escribe el nombre del nodo y su frecuencia de consulta.

Comando ;EJECUTA

Este comando permite ejecutar un programa externo cuando el usuario selecciona el nodo correspondiente; puede utilizarse para desplegar gráficos o imágenes. Cuando el programa termina de ejecutarse, el usuario regresa a Hiperdoc. Su sintaxis es:

```
;ejecuta(liga) programa
```

Parámetro	Significado
(liga)	Es el nombre de la liga que conduce a este nodo. Normalmente, puede omitirse (ejecuta () ...) a menos que se utilice el mismo nodo como destino de varias ligas. Debe escribirse tal como aparece
programa	Es el nombre del programa o comando a ejecutar. Puede indicarse el nombre completo del archivo ejecutable y los archivos de datos, con los directorios en que se encuentran.

Comando ,LIGAS

Este comando permite definir los nombres para los diferentes tipos de ligas que puedan ser definidos para el hiperdocumento. Sólo puede ser codificado en el perfil del hiperdocumento. Su sintaxis es:

```
,ligas( nombre1, nombre2, nombre3
```

Parámetro	Significado
nombre1	Es el nombre para el tipo de liga 1.
nombre2	Es el nombre para el tipo de liga 2.
nombre3	Es el nombre para el tipo de liga 3.

Como ya hemos mencionado, todos estos comandos se mezclan dentro del texto para definir la estructura del hiperdocumento. En la figura IV.3 se muestra un fragmento de un documento en su formato "fuente", antes de ser resuelto para ser mostrado al usuario. Esto nos da una idea básica de la organización de un documento en Hiperdoc, y del uso de los comandos que hemos presentado.

```

!intit( Qué es el hipertexto ??
!ventana() renic:1 colic:1 nren:25 nool:80
!texto() fg:azul bg:verde lfgi:blanco lhgi:verde
!cursor() fg:blanco bg:azul

```

El hipertexto es una tecnología para organizar la información que permite a los lectores desplegarla en diferentes formas, sin sujetarse a un sólo orden secuencial.

En el hipertexto, la información está agrupada en entidades llamadas "nodos". Por ejemplo, un nodo podría contener un procedimiento para encender una máquina, la definición de una palabra o una gráfica. El autor decide cual y cuanta información poner en cada nodo.

```

!int( Qué es el hipertexto ??
!-----
!inicio(nodos)
!ventana() renic:15 colic:40 nren:9 nool:35
!titulo() Nodo

```

Un nodo es la unidad mínima que puede ser direccionada por una liga de hipertexto.

```
!fin(nodos)
```

Figura IV.3. Documento fuente.

V. DISEÑO DETALLADO E IMPLEMENTACION DEL SISTEMA DE HIPERTEXTO.

V.1 Estructura interna del sistema.

En el capítulo anterior hablamos del diseño funcional del sistema de hipertexto, visto desde el punto de vista del usuario. Presentamos la estructura de la herramienta pegada a un modelo de hipertexto, así como las interfases de que dispone el usuario para su interacción con ella, con el objeto de crear o consultar documentos.

Ahora hablaremos de la estructura de la herramienta desde el punto de vista de sistemas. Esto incluye la representación interna de un hiperdocumento dentro del sistema, es decir, las estructuras de datos utilizadas para la creación de nodos y ligas de hipertexto, así como los mecanismos para manipularlas y poder implementar las funciones de navegación y consulta que estarán disponibles para el usuario. Así mismo, presentaremos la estructura de módulos del sistema, sus interfases y diseño, y finalmente los criterios de implementación de la herramienta.

V.1.1 Representación interna de la red de hipertexto.

Como dijimos en el capítulo anterior, un documento dentro de Hiperdoc está integrado por un conjunto de nodos textuales o no textuales, que pueden estar relacionados entre sí a través de un conjunto de ligas. Una de las características inherentes del hipertexto es la posibilidad de "crear" dinámicamente diferentes documentos virtuales o hiperdocumentos al ir eligiendo una trayectoria de navegación dentro de esta red de información. A medida que un usuario recorre nodos, va construyendo una estructura que le ofrece ciertas facilidades, como

regresar hacia nodos previos y elegir otras trayectorias, o bien viajar al inicio de una trayectoria.

Por estas características, la estructura de datos elegida para representar un hiperdocumento es una lista doblemente ligada. Cada nodo de la lista representa a su vez un nodo del documento. Cuando el usuario, posicionado en un nodo determinado, elige una liga y navega hacia un nodo destino, se crea un nodo de la lista y se añade a la misma. La razón de la doble ligadura es para permitir el recorrido de la estructura en ambas direcciones: hacia adelante al ir creando nodos, y hacia atrás cuando el usuario decide dar marcha atrás en la trayectoria elegida. Dado que la lista está apuntada por un nodo raíz, es posible también navegar al primer nodo, que representa el punto de partida de la trayectoria elegida por el usuario.

La figura V.1 muestra en términos muy generales la representación interna de un hiperdocumento. A partir de ahora a los nodos de esta lista los llamaremos nodos de hipertexto con el fin de evitar confusiones con nodos de otras estructuras que serán descritas más adelante.

Cada uno de los nodos de la lista está diseñado para almacenar toda la información relevante del nodo del hiperdocumento al que representa, mas no el contenido mismo del nodo, es decir, el texto del documento. Los datos almacenados en el nodo permiten extraer el texto del archivo correspondiente siempre que se requiera. Esto se hizo con el propósito de ahorrar en el consumo de memoria principal y permitir el manejo de documentos de gran tamaño, sin que la memoria sea un factor limitante, y quizá sacrificando un poco el tiempo de respuesta, que para este caso no es un factor crítico. Sin embargo, pensando en la factibilidad de

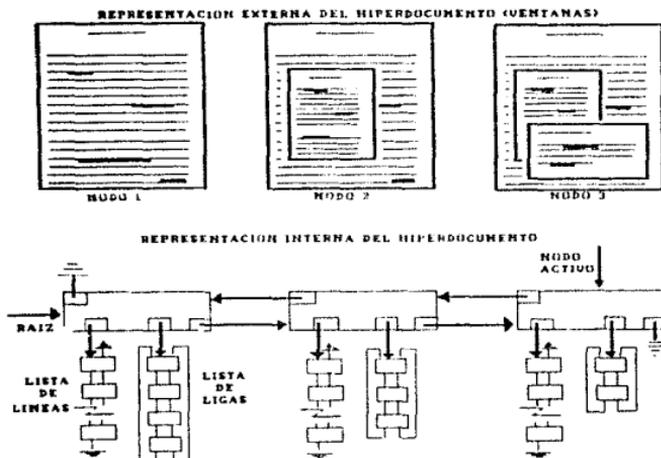


FIGURA VI. ESTRUCTURA GENERAL DE LA RED DE HIPERTEXTO.

convertir libros, manuales, y otros documentos de tamaño considerable a formato de hipertexto, la memoria sí puede ser un recurso limitado.

Así pues, en vez de almacenar cada una de las líneas de texto del documento, se crea una lista doblemente ligada que contiene un nodo por cada línea desplegada; cada nodo guarda el inicio de su correspondiente línea dentro del archivo físico. Cada que el usuario despliegue una nueva línea, se agregará un nodo a esta lista. De nuevo se deduce la razón de la doble ligadura: el usuario puede retroceder dentro del nodo, en cuyo caso será necesario leer líneas anteriores. Para ello se extrae del nodo correspondiente la posición de inicio de la línea en el archivo, se lee y se despliega. Así, para una página de 25 líneas por 80 columnas, si se almacenara todo el texto, se ocuparían 2000 bytes, mientras que de esta forma sólo se utilizan 300 bytes, logrando un ahorro de más del 500%. La asignación dinámica de memoria de acuerdo al requerimiento del usuario también permite el ahorro. Se

crearán tantos nodos como líneas despliegue el usuario, independientemente del tamaño del nodo de hipertexto.

De esta forma, cada nodo de hipertexto tiene un apuntador a su correspondiente lista de líneas. Cada nodo de la lista de líneas posee, además de la posición de inicio de cada línea en el archivo, dos apuntadores que lo ligan al resto de la lista, uno para cada sentido de recorrido.

Cuando el usuario selecciona una liga que lo conduzca a un nuevo nodo, si este nodo existe dentro del hiperdocumento, se creará un nodo de la lista que permanecerá almacenado en la memoria mientras que el usuario no elimine el nodo de la trayectoria navegando hacia atrás. Al nodo de hipertexto en que el usuario se encuentra en un momento dado, lo llamaremos nodo activo. Existe un apuntador al nodo activo, que se actualiza conforme el usuario navega por la red. Cuando el usuario nuevamente viaje hacia adelante, este nodo seguirá formando parte de la lista, pero dejará de ser el nodo activo, hasta que el usuario vuelva a posicionarse en él. Esto permite que pueda conservarse en la memoria el estado de cada nodo al navegar por el hipertexto. Es conveniente aclarar que cada vez que el usuario elimina un nodo de la lista al navegar hacia nodos previos, se libera la totalidad de la memoria que ocupaba dicho nodo, con el objeto de tener en todo momento el máximo de recursos disponible para nuevos nodos de hipertexto.

Recordando que Hiperdoc es un sistema basado en ventanas, podemos imaginar que en un momento dado, del nodo activo solamente se estará visualizando una porción cuyo tamaño dependerá de las características de la ventana definidas por el autor del documento para ese nodo.

Llamaremos a esta porción, página activa del nodo. Una página activa está delimitada por una línea inicial y una línea final dentro de la lista de líneas. Cada nodo de hipertexto tiene un apuntador a la línea inicial y otro a la línea final de su página activa. Esto permite controlar el avance y retroceso por línea o página dentro del nodo de hipertexto.

Recordemos que el sistema soporta también el avance y retroceso horizontal dentro del texto del nodo activo durante la consulta, ya que la ventana pudo haber sido definida más pequeña que el tamaño de las líneas, de modo tal que es necesario moverse horizontalmente para desplegar la totalidad del texto. Esta característica obliga a mantener almacenada en memoria las líneas completas de la página activa, con el objeto de que el movimiento horizontal del texto sea rápido. Por lo tanto cada nodo de hipertexto dentro de la lista contiene un arreglo en memoria para almacenar la página activa, así como un índice que apunta a la primera columna que se está desplegando y que se actualiza conforme el usuario avanza o retrocede.

Un nodo de hipertexto dentro del documento puede o no tener ligas definidas hacia otros nodos. En caso de que no las tenga, se considera un nodo terminal, y cuando el usuario navega a uno de estos nodos, no podrá seguir navegando hacia adelante; la única opción es retroceder sobre su trayectoria. Sin embargo, cuando el nodo sí tiene ligas, el usuario puede recorrer las diferentes ligas de la página activa, hasta elegir una que lo conducirá a un nuevo nodo. La liga sobre la que se encuentre posicionado el usuario en un momento dado se conoce como liga activa.

Para almacenar la información relativa a todas las ligas de la página activa se utiliza una lista circular doblemente ligada. La razón de que sea así es para permitir al usuario recorrer las ligas dentro de la ventana en forma circular y además en ambos sentidos. Cada nodo de la lista de ligas guarda información de la

liga que representa: el número de la línea dentro de la ventana donde se encuentra la liga, la columna inicial y final dentro de la línea, el texto que sirve de marcador a la liga, el tipo de liga de que se trata (recordemos que se soportan tres tipos diferentes) y un par de apuntadores para ligar el nodo al resto de la lista. Resumiendo, cada nodo de hipertexto contiene un par de listas: una para las líneas del nodo que vayan siendo desplegadas, y otra para las ligas presentes en la página activa del nodo. El nodo de hipertexto posee algunos otros datos, como son la posición y tamaño de su ventana asociada, un apuntador a la siguiente línea por leerse dentro del archivo físico, un apuntador a la liga activa dentro de la página y un arreglo en memoria para guardar el contenido de la pantalla antes de desplegar una nueva ventana, y poder restablecerla cuando el usuario decida retroceder en la lista de nodos de hipertexto. La figura V.2 muestra gráficamente la estructura interna de cada nodo de hipertexto.

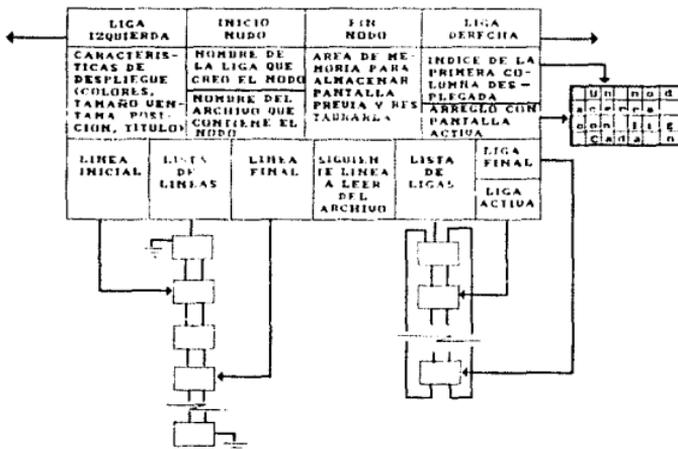


FIGURA V2. ESTRUCTURA DETALLADA DE UN NODO DE HIPERTEXTO.

V.1.2 Criterios de creación, actualización y consulta.

Esta es pues la representación interna de la red de hipertexto dentro de Hiperdoc. Hablemos ahora de los criterios de consulta y actualización de estas estructuras durante la consulta de los documentos. Es conveniente recordar que este ambiente se crea únicamente durante el tiempo de consulta. Cuando se editan los documentos, lo que en realidad se hace es construir el archivo fuente que será interpretado posteriormente por el sistema para construir documentos virtuales según los requerimientos del usuario.

La lista de nodos de hipertexto se crea cuando el usuario invoca al sistema de hipertexto con algún nombre válido de archivo fuente y opcionalmente con el nombre del nodo en el que desea iniciar la consulta (si no lo especifica, el nodo inicial por omisión es el que está delimitado por los comandos ;inicio() y ;fin(), es decir sin ningún nombre). Se identifica el nodo inicial del archivo, se crea su correspondiente nodo de hipertexto dentro de la lista y se añade a ella. Como se despliega la primera página de texto del nodo, se crea también a este tiempo la lista de líneas del nodo, y se establecen los apuntadores a las líneas inicial y final de la página. En caso de que esta página contenga ligas (al menos una), se creará la lista de ligas de la página activa, y se inicializa el apuntador de la liga activa a la primera liga de la página. Así también, la página activa se almacena en un arreglo en memoria, y el índice se inicializa apuntando a la primera columna del arreglo.

Cuando el usuario decide avanzar dentro del nodo, ya sea línea por línea o a través de páginas completas, la lista de líneas del nodo se irá actualizando, añadiendo todas las líneas que vayan siendo desplegadas. Esto implica que se actualizarán también los apuntadores a las líneas inicial y final de la página. Así

mismo, será necesario actualizar la página activa en memoria eliminando las líneas que dejen de pertenecer a ella y agregando las que se incorporan. Sin embargo, el cambio más significativo se da en la lista de ligas de la página. Cada que cambian las líneas desplegadas en la ventana, cambiarán también las ligas que el usuario puede seleccionar. Pueden distinguirse tres casos diferentes:

- Avance por línea.

Cuando el usuario avanza una línea dentro del nodo, la primera línea de la página desaparece de la ventana, y por lo tanto, si existían ligas en ella, también deben ser eliminadas de la lista. Recordemos que cada nodo de la lista de ligas guarda el número de línea de la liga dentro de la ventana. Por lo tanto, además de eliminar las ligas de la primera línea es necesario actualizar las posiciones de línea de las ligas restantes, restando uno a cada posición. La liga que anteriormente estaba en la segunda línea de la ventana, ahora estará en la primera, y así para las demás. Para la nueva línea, que aparece en la última posición de la página, es necesario detectar si contiene ligas. En caso afirmativo, esta(s) nueva(s) liga(s) deberá(n) agregarse a la lista de ligas de la página, al final. De esta forma el usuario podrá recorrerlas igual que a las demás.

- Retroceso por línea.

Este caso es similar al anterior, pero ahora la línea que desaparece es la última de la página, y la nueva línea toma la primera posición en la ventana. Ahora hay que eliminar de la lista todas las ligas que pertenecían a la última línea. También es necesario actualizar las posiciones de línea de las ligas restantes, sumando uno a cada posición. Así, la liga que antes estaba en la primera línea, ahora estará en la segunda, y así sucesivamente. Por último, debe analizarse la nueva línea para detectar las ligas que contenga, y agregarlas en la lista, al principio.

- Avance y retroceso por página.

Cuando el usuario decide avanzar o retroceder una página completa dentro del nodo, todas las ligas de la lista desaparecerán de la ventana, por lo que podemos eliminarlas a todas. Así, habrá que analizar todas las líneas de la nueva página y reconstruir la lista de ligas.

En forma similar, cuando el usuario avanza o retrocede en forma horizontal dentro de la ventana del nodo activo es necesario actualizar, por un lado, el índice apuntador a la primera columna desplegada, y por el otro, las ligas que deban aparecer en la ventana. Así, cuando el usuario avanza una posición a la derecha, el índice se incrementa en uno para apuntar a la siguiente columna, que ahora será la primera dentro de la ventana. En cuanto a las ligas, el criterio seguido es evaluar al momento de marcarlas en la ventana, cuales de ellas deben seguir apareciendo ya sea en forma completa o parcial, y cuales han desaparecido de la ventana como resultado del movimiento del texto. Sin embargo, no se elimina ninguna liga de la lista, solamente se despliegan o no dependiendo de su posición dentro de la ventana. Para ello se utilizan las posiciones inicial y final que para cada liga se almacenan en la lista de ligas del nodo. Por ejemplo, si hay una liga definida de la columna 1 a la columna 8, y el usuario avanza tres posiciones a la derecha, esta liga continuará siendo desplegada, aunque en forma parcial. Sin embargo, si el usuario avanza diez posiciones, es evidente que la liga "desaparecerá" de la ventana, con lo cual a pesar de seguir en la lista, será invisible para el usuario y no podrá ser seleccionada para navegar. Para el caso del retroceso horizontal, sucede algo similar. El índice se decrementa tantas unidades como posiciones se haya retrocedido, y se evalúa el inicio y fin de cada liga para decidir si debe ser desplegada o no. Este criterio de actualización

permite hacer más rápido el movimiento horizontal, ya que evita el estar actualizando la lista de ligas cada vez.

Algo similar ocurre cuando el usuario durante la consulta decide desactivar ciertos tipos de liga. Esto equivale a ocultar ligas de modo que sea menos complicada la navegación y el usuario pueda reducir el espectro de navegación de acuerdo a sus intereses. Cada que se marca una liga dentro de la ventana, se analiza si el tipo de la liga está activo o no. Si no lo está, la liga no será marcada independientemente de que su posición le permita ser desplegada. Sin embargo, ninguna liga será eliminada de la lista, ya que en cualquier momento el usuario puede cambiar la selección de tipos de liga activos.

V.2 Estructura de módulos del sistema.

A continuación presentamos el diseño detallado del sistema de hipertexto, de acuerdo con las funciones y estructuras de datos que hemos analizado anteriormente.

El diseño de la herramienta de hipertexto está basado en la metodología de Diseño Orientado al Flujo de Datos, que consiste en utilizar las características del flujo de información dentro del modelo del sistema, para derivar una estructura de programas. Así pues, partiendo del Diagrama de Flujo de Datos del sistema se puede obtener un Diagrama de Estructura utilizando dos posibles técnicas: análisis centrado en transformación y/o análisis centrado en transacciones.

El análisis centrado en transformación se aplica a un flujo de datos donde todas las transacciones siguen caminos muy similares, y el flujo puede separarse

claramente en tres partes: datos de entrada, procesamiento de los datos y datos de salida. La estructura de programas resultante refleja esta organización, al tener un módulo controlador de la recepción de los datos, uno que supervisa las operaciones sobre los datos y un tercero que coordina la producción de información de salida.

El análisis centrado en transacciones se aplica cuando el flujo de datos sigue diversas trayectorias dentro del sistema, presentándose así múltiples transacciones. En este caso, la estructura de programas contiene un módulo analizador que se encarga de determinar el tipo de transacción requerida, y un módulo despachador cuyo tarea es dirigir la transacción a un subsistema apropiado establecido para cada tipo de transacción.

Es muy probable que durante el diseño de un sistema haya que utilizar ambas técnicas, ya que en el Diagrama de Flujo de Datos pueden identificarse partes que requieran análisis por transacciones y otras que requieran análisis por transformación. Por ejemplo, para un sistema que presenta características de flujo por transacciones, es posible que para cada transacción haya que plantear a su vez un análisis por transformación.

En términos generales, los pasos a seguir para derivar una Estructura de Programas utilizando la metodología de Diseño Orientado al Flujo de Datos, son:

1. Revisar el modelo conceptual del sistema.

Esta etapa implica revisar los requerimientos de diseño y funcionalidad del sistema para determinar las características generales de flujo de información.

2. Determinar y refinar Diagramas de Flujo de Datos del sistema.

Esta etapa consiste en derivar un Diagrama de Flujo de Datos a partir del modelo conceptual del sistema, en donde representaremos las entidades externas al sistema que proveen datos de entrada y reciben información de salida, los almacenamientos de datos y los flujos de información.

3. Establecer si el Diagrama de Flujo de Datos está centrado en transformación o en transacciones.

En esta etapa, se determina la característica global de flujo de datos dentro del sistema, además de establecer áreas locales donde pueda aplicarse el análisis por transformación o transacciones.

4. Derivar una Estructura Preliminar de Programas.

De acuerdo con la característica general determinada en la etapa anterior, se establece un primer diagrama de estructura. Si se eligió el análisis por transformación, se determina el flujo de entrada, el flujo de salida y el centro de transformación, y se relacionan con un módulo que coordina la recepción de datos, otro que controle el procesamiento de los datos y un tercero que produzca la información de salida. Si se eligió el análisis por transacciones, se determina el centro de transacciones y cada una de las trayectorias, y se deriva una estructura con un módulo que analiza el tipo de transacción, y un módulo despachador que dirige la transacción al subsistema encargado de procesarla.

5. Detallar la Estructura de Programas.

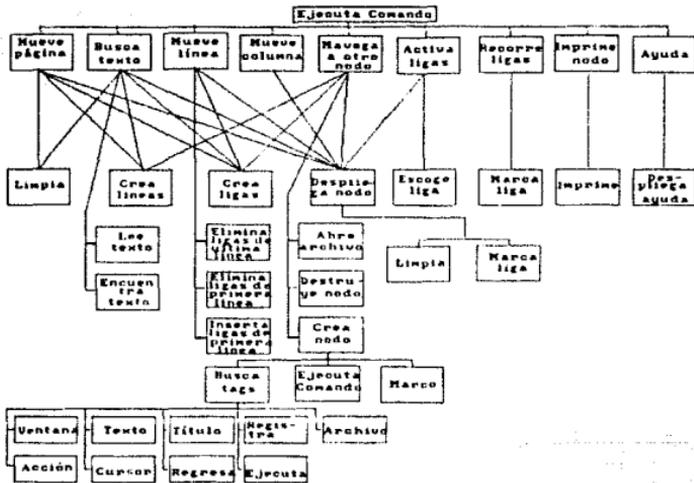
Esto consiste en detallar las subestructuras resultantes, de acuerdo con el Diagrama de Flujo de Datos, con el objeto de completar el Diagrama de Estructura. En el caso de análisis por transacciones, es necesario determinar la subestructura correspondiente a cada transacción.

Podemos distinguir de este diagrama que existe una tarea inicial que consiste en crear el primer nodo del hiperdocumento. Esto sucede cuando el usuario comienza la consulta. Si el nodo es creado sin problemas, entonces el sistema empieza a leer comandos, los cuales son ejecutados sobre el nodo de hipertexto activo. Por ejemplo, el avance de página, la búsqueda de texto o la impresión del nodo.

Cuando el usuario decida navegar hacia un nuevo nodo, se intenta entonces crear un siguiente nodo de hipertexto; si no hay errores, el nuevo nodo convierte en el nodo activo, y todos los comandos sucesivos se ejecutarán sobre él, hasta que el usuario solicite la navegación hacia otro nodo, ya sea hacia adelante o hacia atrás (es decir, regresar al nodo anterior, o al primero del hiperdocumento).

El siguiente diagrama muestra la Estructura detallada del módulo Ejecuta Comando que procesa los diferentes comandos del usuario. Estos módulos comparten varias rutinas comunes, como son las relativas a la creación de las listas de ligas y de líneas de cada nodo y las que tiene que ver con el despliegue del nodo en su correspondiente ventana. Hay algunas otras rutinas de uso más específico, como las de búsqueda de texto en el nodo, o de impresión, o bien la rutina de ayuda al usuario.

Detalle del Diagrama de Estructura del Sistema.



V.3 Consideraciones de implementación.

La herramienta de hipertexto será desarrollada para operar en una computadora personal. Se decidió desarrollarla utilizando Lenguaje C por las siguientes razones:

- El código resultante tiene gran portabilidad. Es posible transportar esta herramienta a otras plataformas sin modificaciones sustanciales. Las únicas rutinas dependientes del equipo son aquellas que tiene que ver con el nivel de presentación de la herramienta (interfase de usuario). Básicamente nos referimos a las rutinas de despliegue del nodo, de despliegue de la ventana y de marcado de ligas.

- Es posible utilizar con facilidad instrucciones de bajo nivel para las rutinas dependientes del equipo. Por ejemplo, para hacer más eficiente el programa y reducir el tiempo de respuesta, el manejo de la escritura del nodo a la pantalla se hará a través del manejo de la memoria de video de la computadora personal. Otro ejemplo son las rutinas de paginación dentro del nodo, para lo cual se utilizan interrupciones de software provistas en el ROM BIOS (Basic Input/Output System) para el manejo del video.

- La flexibilidad en el manejo de apuntadores y estructuras de memoria, ya que el manejo central de los nodos y ligas de hipertexto se realiza en base al concepto de memoria dinámica y apuntadores.

Los requerimientos básicos para poder ejecutar esta herramienta son: un equipo personal IBM compatible, 640 KB de memoria RAM y unidad de disco (puede ser flexible o duro, para mayor eficiencia en la ejecución se recomienda contar con un disco duro). El único programa requerido es el sistema operativo DOS.

VI. CASO DE ESTUDIO.

A lo largo de los capítulos precedentes, hemos desarrollado una herramienta para la construcción de hiperdocumentos. Esta herramienta, independientemente de la aplicación para cual se utilice, nos permitirá elaborar nodos de información, definir ligas para su interconexión, y construir un sistema de hipertexto para ser consultado.

Sin embargo, el uso de esta herramienta debe estar regido por un diseño previo. Al igual que en la construcción de sistemas de información, antes de poder construir programas, pantallas, archivos, etc., con las múltiples herramientas que tenemos a nuestro alcance, es necesario primero elaborar un diseño basándonos en alguna metodología que se adapte a las características del sistema que pretendemos lograr. Así pues, esta metodología nos permitirá analizar los requerimientos y construir un diseño que los satisfaga. Posteriormente, podremos partir del diseño para entonces construir los componentes de nuestro sistema.

Cuando hablamos de sistemas de hipertexto, podemos pensar en un gran conjunto de nodos de información, interconectados por diversas ligas. Sin embargo, parece razonable pensar que no podemos conectar absolutamente toda la información a nuestro alcance. En vez de esto, es preciso determinar donde será útil relacionar qué nodos de información. Para ello, es necesario identificar las características de la aplicación a la que nos estamos enfrentando, y diseñar una estructura de información que nos permita implementar un sistema de hipertexto coherente y útil para quienes posteriormente serán los usuarios.

Existen diferentes acercamientos al problema de diseñar hiperdocumentos. Básicamente podrían clasificarse como sigue:

Metodologías Prescriptivas: Atacan el problema de diseño predefiniendo topologías (redas, árboles, estructuras recursivas) como los elementos básicos a partir de los cuales se construirán los hiperdocumentos, e imponiendo una estructura regular en todos ellos.

Metodologías Constructivas: No prescriben ninguna estructura a priori. En vez de esto, pretenden auxiliar al autor a descubrir y clarificar regularidades en documentos específicos.

Metodologías Inductivas: Crean una estructura de hipertexto identificando las estructuras implícitas en documentos existentes e incorporándolas luego al hiperdocumento.

Metodologías basadas en Modelos: Estas metodologías están orientadas a proporcionar al autor con una especie de lenguaje para describir estructuras de hipertexto antes de que el hipertexto sea escrito, y poder comparar diferentes alternativas de diseño para el mismo documento.

El objetivo de este capítulo es presentar una metodología para el diseño conceptual de una estructura de hipertexto, y posteriormente presentar un caso de estudio en donde se aplicará esta metodología para diseñar un sistema de hipertexto que será finalmente implementado con nuestra herramienta. La metodología presentada se conoce como HDM (Hypermedia Design Model), y pertenece al tipo de metodologías basadas en modelos. El caso de estudio al que nos referimos es el desarrollo de un prototipo de hipertexto para el Programa de Prestaciones para empleados de la empresa IBM de México.

VI.1 Metodología HDM para el diseño de hiperdocumentos.

Podemos utilizar el término Hypermedia Design Model (HDM) [15] para denotar un lenguaje conceptual que puede ser utilizado para describir estructuras comunes que se presentan dentro de un grupo de hiperdocumentos relacionados entre sí, identificando sus propiedades estructurales y semánticas independientemente de su posterior implementación a través de una herramienta específica.

La metodología HDM permite a un autor de hipertexto describir la estructura de un hiperdocumento a un nivel general, sin especificar cada una de las ligas o nodos en forma individual. El contar con un modelo conceptual del sistema facilita enormemente su comprensión, descripción y actualización, así también sirve como punto de partida para el diseño e implementación de hiperdocumentos.

El objetivo principal de la metodología HDM es describir estructuras complejas como conjuntos de estructuras más pequeñas y simples. Mediante la construcción de un modelo, se identifican los componentes individuales que el autor utilizará para la construcción de una red de hipertexto, así como la forma de "ensamblar" estos componentes.

Es importante aclarar que la metodología HDM solamente describe la estructura de un hiperdocumento. El modelo resultante no describe la apariencia del documento: tipos de letras, interfaces gráficas, forma de indicar ligas, etc. Así mismo, el modelo no establece el contenido individual de cada nodo. En vez de esto, su objetivo es describir únicamente las relaciones estructurales y semánticas entre grupos de nodos.

Componentes.

En cualquier hiperdocumento podemos identificar diferentes nodos que están relacionados con un mismo tema. Por ejemplo, en una aplicación de hipertexto para manejo de recursos humanos, podríamos tener un nodo conteniendo los datos generales de una persona, otro nodo conteniendo la fotografía de la misma persona, y uno más con su curriculum. Los tres nodos poseen diferente información relacionada con un mismo sujeto. En la metodología HDM, decimos que cada uno de estos nodos representa una perspectiva diferente de la persona. Así, este conjunto de nodos que contienen información acerca de un individuo se conoce como componente. Cada nodo corresponde a una perspectiva del componente. La figura VI.1 muestra la relación entre componentes y perspectivas utilizando como ejemplo una obra musical.

ENTIDAD	LAS CUATRO ESTACIONES (VIVALDI)		
COMPONENTE	PRIMAVERA (etc.)		
PERSPECTIVAS	GRABACION DIGITAL	PARTITURA	COMENTARIOS TEXTUALES
	NODO	NODO	NODO

Fig. VI.1 Entidades, componentes y perspectivas

Entidades y Tipos de Entidades.

Una entidad es un conjunto de componentes que describen un objeto o concepto. La mayoría de los hiperdocumentos incluyen entidades de tipos fuertemente relacionados entre sí. Así por ejemplo, un manual de mantenimiento de partes mecánicas podría

incluir entidades de tipo Problema y Procedimiento de Reparación. Todas las entidades de un mismo tipo comparten el mismo conjunto de posibles perspectivas de los componentes y, potencialmente, ligas a otros tipos de entidades. Es decir, regresando a la figura VI.1. el tipo de entidad que se muestra es una obra musical compuesta por varios movimientos. Suponiendo que tengamos dos entidades de este tipo (una obra de Mozart y otra de Beethoven), ambas van a tener las mismas perspectivas sobre cada uno de sus componentes (movimientos): una grabación, la partitura, comentarios sobre la obra, etc. Como resultado de esto, todas las entidades del mismo tipo pueden compartir las mismas características básicas de presentación.

Ligas y Tipos de Ligas.

De la misma forma que la presentación de entidades del mismo tipo debe ser consistente, así también las ligas entre los componentes que integran una entidad deben serlo también. Estas ligas, que forman la infraestructura de una entidad, se conocen como ligas estructurales. Por ejemplo, si los componentes de un Procedimiento de Reparación son los diferentes pasos a seguir en un orden determinado, podría requerirse que cada paso estuviera ligado al paso anterior y al siguiente. O bien, si los componentes de un automóvil son sus partes, cada una de las partes podría estar ligada con las partes adyacentes.

De acuerdo con la metodología HDM, la estructura interna de una entidad es siempre jerárquica.

La segunda clase de ligas en HDM son las ligas aplicativas. Mientras que las ligas estructurales representan la estructura inherente de una entidad, las ligas aplicativas reflejan relaciones semánticas entre conceptos que no están relacionados estructuralmente dentro del hiperdocumento. Por la misma razón, la presencia de ligas aplicativas depende del contenido de entidades y componentes

específicos. Por ejemplo, todas las Obras de Shakespeare pudieran tener ligas estructurales a Actos, pero solo algunas pueden poseer ligas aplicativas a Reyes de Inglaterra. Normalmente las ligas aplicativas relacionan entidades de tipos diferentes.

De ligas conceptuales a ligas concretas.

Al conjunto de entidades y tipos de ligas se le llama Esquema. Un hiperdocumento en particular es descrito usando entidades y ligas específicas, las cuales constituyen una instancia o derivación del esquema.

Ahora bien, las entidades y componentes son objetos abstractos que HDM utiliza para representar conceptos. Cuando un autor crea un nuevo hiperdocumento a partir de un esquema, estas entidades abstractas se implementan tomando la forma de nodos o piezas de información que serán presentadas al lector. Es claro que esta implementación debe hacerse respetando las reglas establecidas en el modelo. Así, si el esquema no provee ligas entre peras y manzanas, el autor no debe ligar un nodo manzana y un nodo pera. Por otro lado, si el diseño establece una liga estructural entre una persona y un acta de nacimiento, el autor debería ligar al menos un nodo de cada persona a un acta de nacimiento.

Esta es la metodología que se aplicará en el diseño del hipertexto. A continuación se detalla el caso de estudio.

VI.2 Caso de Estudio.

La aplicación que desarrollaremos como caso de estudio tiene que ver con el Programa de Beneficios que actualmente ofrece la empresa IBM de México a todos sus empleados.

VI.2.1 Antecedentes.

El Programa de Beneficios consiste de una serie de Planes de Prestaciones relacionados con uno de tres aspectos:

- Protección
- Seguridad
- Oportunidad

Los Planes de Prestaciones se enumeran a continuación:

Protección.

- * Plan IBM Médico y de Hospitalización
- * Plan IBM de Protección por Enfermedades y Accidentes
- * Plan IBM de Exámen Médico Periódico

Seguridad.

- * Plan IBM de Seguro de Grupo y Beneficio a los Sobrevivientes
- * Plan IBM de Jubilación
- * Paquete IBM para Empleados Jubilados
- * Plan IBM de Ingresos por Incapacidad Total y Permanente
- * Plan IBM de Accidentes en Viajes

Oportunidad.

- * Plan IBM de Ahorros
- * Plan IBM de Reembolso de Colegiaturas
- * Plan IBM de Vacaciones
- * Plan IBM de Días Festivos

* Plan IBM de Gratificación de Fin de Año

* Plan IBM Múltiple

Cada uno de estos Planes deben ser consultados por el empleado para realizar prácticamente cualquier trámite relacionado con el Programa de Prestaciones: hacer efectivo el seguro médico en caso de una enfermedad o el seguro de vida en caso de accidentes, llevar a cabo la jubilación, solicitar vacaciones, reembolso de colegiaturas, etc.

A su vez estos Planes están relacionados con otras entidades. Por ejemplo, existen Procedimientos que indican cuales son los departamentos involucrados en un trámite, y que acciones tiene que realizar el empleado para llevarlo a cabo. También existen Documentos relacionados con cada uno de los Planes. Por ejemplo, existe una Solicitud de Reembolso de Colegiatura que es requisito llenar para tener acceso al Plan respectivo. Por otro lado algunos de estos Planes están relacionados con algunas de las legislaciones vigentes en nuestro país, por ejemplo, la Ley Federal del Trabajo o la Ley del Seguro Social, que proporcionan un marco de referencia para la ejecución del Plan.

Existe una Carpeta denominada "IBM: Usted y su Compañía" que usualmente se proporciona a cada empleado que ingresa a la compañía, en donde se explican detalladamente cada uno de los Planes de Prestaciones que ofrece la compañía. Este documento tiene algunos inconvenientes:

- Los Planes de Prestaciones constantemente son examinados y comparados con los de otras organizaciones, con el objeto de mejorarlos y ampliarlos a través del tiempo. Esto hace que la Carpeta pierda vigencia rápidamente.

- Mucha de la información referida en los Planes de Prestaciones no está incluida en la Carpeta por razones obvias: Documentos requeridos (formas, constancias), Leyes y Reglamentos, etc., lo cual dificulta la consulta y obliga a buscar en varias fuentes para resolver una pregunta en forma completa.

Por otro lado, existe una versión electrónica "on-line" del Programa de Prestaciones disponible para todos los empleados en la Red de Información de IBM. A través de un comando, es posible desplegar en una terminal de computadora los diferentes Planes, e imprimirlos. Esta implementación tiene claras ventajas sobre la Carpeta, como lo son la facilidad de actualizar su contenido, la facilidad de consulta y disponibilidad de la información. Sin embargo, esencialmente presenta problemas comunes a los de la documentación escrita:

- La información relacionada con los Planes no está presente en el sistema, de tal forma que sigue siendo necesario consultar fuentes externas para obtener información completa. Incluso, dentro de cada Plan hay frecuentes referencias a otros Planes, de modo que cada que el lector encuentra una referencia, necesita abandonar el Plan que se encuentra consultando para revisar otro (por ejemplo, en el Plan de Protección por Enfermedades y Accidentes se hace referencia al Plan de Ingresos por Incapacidad Total y Permanente).

- La información está presentada en forma secuencial, es decir, agrupada a través de un índice o temario, que es una lista de todos los planes disponibles, y el usuario tiene que seleccionar uno de ellos para entrar a consultarlo. Una vez dentro del Plan, la información es presentada secuencialmente, de tal forma que si el lector está interesado en consultar una sección específica del Plan (Exclusiones, por ejemplo) tiene que recorrer el documento hasta encontrarla.

VI.2.3 Justificación de un Sistema de Hipertexto.

Recordando el capítulo II, podemos decir que el hipertexto no es la solución a todos los problemas. Por lo cual existen tres "Reglas de Oro del Hipertexto" que nos ayudan a identificar si el sistema de información que estamos analizando es susceptible de implementarse a través de un hipertexto. Así, un sistema pueda considerarse apropiado para el hipertexto en la medida en que cumple con las siguientes Reglas:

- Existe un gran volumen de información, organizado en numerosos fragmentos.
- Los fragmentos están relacionados entre sí.
- El usuario necesita sólo una pequeña parte de la información en cualquier momento.

Analizando la aplicación del Programa de Prestaciones, podemos darnos cuenta que es una aplicación idónea para desarrollarse en un sistema de hipertexto, ya que cumple totalmente con las tres reglas:

- Los Planes de Prestaciones están organizados de manera sistemática, y todos ellos comparte la misma estructura, es decir, todos están divididos en cláusulas que varían según el Plan de que se trate: propósito del Plan, elegibilidad, administración, etc.
- Prácticamente en todos los Planes hay referencias cuando menos a otro Plan dentro del Programa de Prestaciones, al igual que a diversas entidades externas: Cartas de Instrucción, Leyes, Procedimientos, etc.
- Normalmente, la mayoría de los empleados están interesados en consultar una porción específica del Plan, como podría ser los gastos elegibles para el Plan Médico, o el procedimiento de reembolso de gastos múltiples. Rara vez un empleado

está interesado en leer toda la información sobre todos los Planes de Prestaciones. Normalmente esto lo hacen cuando reciben la Carpeta al ingresar a la compañía.

Partiendo de esta base, podemos emplear la metodología HDM para diseñar un modelo conceptual de hipertexto para esta aplicación.

VI.2.3 Diseño conceptual del hipertexto usando la metodología HDM.

De acuerdo con la descripción previa de la metodología HDM, podemos empezar identificando los diferentes tipos de entidades involucradas en el sistema. Básicamente, hablamos de los siguientes tipos:

1. Plan de Beneficio. Esta entidad se refiere a cada uno de los planes de prestaciones de la empresa, por ejemplo, el Plan IBM de Jubilación, o el Plan IBM Médico y de Hospitalización. La mayoría de estos Planes tiene un Procedimiento asociado. En algunos casos, un Plan no tiene un procedimiento asociado. Por ejemplo, el Plan de Gratificación de Fin de Año (Aguinaldo) no tiene un procedimiento, ya que el empleado no necesita tomar ninguna acción para recibir los beneficios.

2. Procedimiento. Esta entidad se refiere a cada uno de los procedimientos que un empleado debe seguir para obtener la prestación establecida en el Plan de Beneficio correspondiente. Así, para el Plan IBM de Reembolso de Colegiaturas, existe un Procedimiento de Reembolso de Colegiaturas que el empleado debe seguir para recuperar el costo de un estudio aprobado por la compañía, como podría ser un curso de inglés.

3. Documento. Cada uno de los Planes de Beneficio, y por consecuencia los Procedimientos, necesitan una serie de Documentos para llevarse a cabo. Hablamos de solicitudes, formas, comprobantes, etc. Del mismo modo, un Procedimiento puede generar documentos.

4. Leyes. Como mencionamos previamente, existe relación entre un Plan de Beneficio y alguna regulación legal vigente. Por ejemplo, el Plan IBM de Jubilación hace referencia a la Ley del Instituto Mexicano del Seguro Social. Por otro lado, en un momento dado una ley puede dar origen a un nuevo Plan, o modificar uno ya existente, como pueda ser el caso del "SAR" (Sistema de Ahorro para el Retiro).

5. Cartas de Instrucción. Esta entidad se refiere a ciertas regulaciones internas de la empresa, que sin ser políticas permanentes ni formar parte de los Planes de Beneficios ni Procedimientos, pueden modificarlos en un momento dado. Por ejemplo, los límites de gastos permitidos en viajes al extranjero son referidos en el Procedimiento de Reembolso de Gastos de Viajes, y pueden cambiar independientemente del procedimiento. Una carta de instrucción puede ser resultado de un cambio en la legislación. Por ejemplo, nuevos requisitos fiscales para los comprobantes de gastos médicos.

Utilizando estas entidades, es posible diseñar un modelo conceptual que refleje las relaciones entre cada una de ellas:

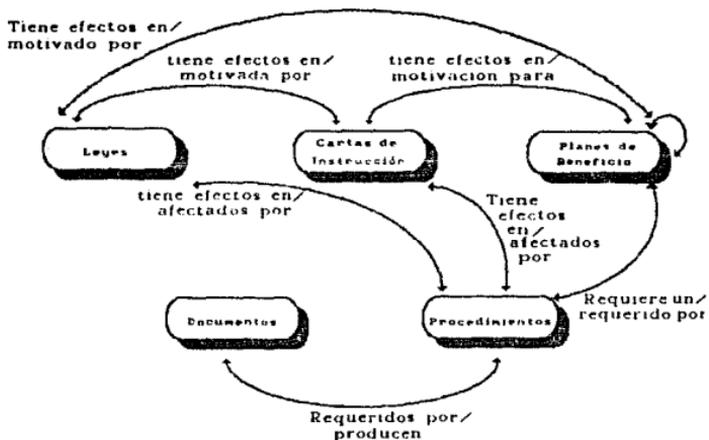


Fig. VI.2 Modelo Conceptual HDM

Este es un modelo a nivel general del hipertexto. Ahora bien, podemos partir de él para implementar hiperdocumentos específicos. Por ejemplo, a continuación se muestra la implementación del modelo para el caso del Plan IBM Médico y de Hospitalización.

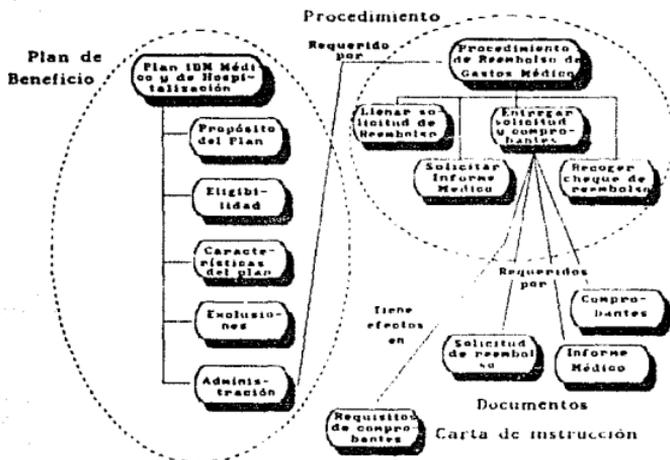


Fig. VI.3 Implementación del Modelo HDM

VI.2.4 Implementación del hipertexto.

El diagrama anterior muestra las ligas estructurales internas de cada entidad, así como las principales ligas aplicativas entre entidades. De aquí procedemos a la implementación física del hiperdocumento, tomando el contenido real de cada uno de los componentes y definiendo en base a él algunas ligas aplicativas adicionales. Por ejemplo, es conveniente mantener un glosario de términos, y definir ligas para aquellos términos dentro del Plan que requieran de otro nivel de detalle para permitir la comprensión de la información. A continuación se muestra una pantalla con la implementación real del hiperdocumento.

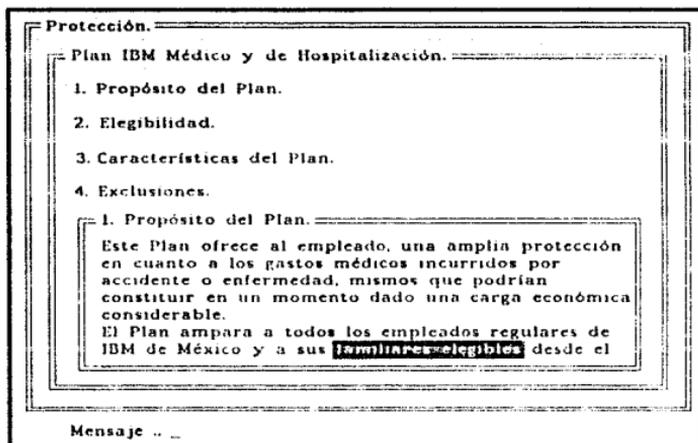


Fig. VI. 4 Aspecto final del hiperdocumento.

Para la captura de la información textual, se utilizó un editor de textos convencional para computadora personal, que permite la escritura de archivos en formato de texto, sin caracteres especiales de control de formato. En algunos casos, la información ya estaba capturada en archivos electrónicos de texto, los cuales se utilizan para la versión en línea del Programa de Prestaciones de la que hablamos anteriormente. En estos casos, fue suficiente con transmitir los archivos a la computadora personal, para posteriormente editarlos y agregar los comandos necesarios al texto para definir los nodos, ligas, características de despliegue, etc.

Cada una de las entidades se almacenó en un archivo separado. Es decir, el Plan de Jubilación está contenido en un archivo, el Procedimiento de Reembolso de Gastos Médicos en otro, cada carta de instrucción en otro, etc. Esto fue con el propósito de facilitar la edición de cada documento, y su posterior actualización. Así, puede haber un responsable de actualizar cada Plan, y el puede trabajar solamente en ese Plan sin afectar el resto de los documentos. A continuación se muestra uno de estos archivos, en el cual puede apreciarse el uso de los distintos comandos de Hiperdoc para definir nodos de hipertexto, ligas, características de despliegue de las ventanas, registro de estadísticas, etc.

```
.....  
!* Este nodo contiene la definición del Programa de Prestaciones  
.....  
;inicio(Prestaciones)  
PROGRAMA DE PRESTACIONES  
;titulo() PROGRAMA DE PRESTACIONES DE IBM DE MEXICO  
;ventana() bg=verde reni=2 coli=2 ncol=75 nren=18  
;cursor() fg=azul bg=gris  
;texto() fg=negro bg=gris  
;registra() estad.txt
```

El programa de Prestaciones de IBM de México, está diseñado de manera que la compañía cubra el costo total del mismo.

El propósito de este programa es proporcionar una base amplia sobre la cual el empleado obtenga beneficios económicos y el bienestar de su familia.

Constantemente se examina el programa y se compara con el de otras organizaciones locales y nacionales. A través de los años, ha sido sustancialmente mejorado y ampliado para hacer frente a las necesidades cambiantes de los empleados.

Los planes de prestaciones de IBM proporcionan una base para:

- Protección.- Contra pérdida temporal de ingresos y gastos médicos que hayan resultado de alguna enfermedad o accidente.
- Seguridad.- Proporcionando ingresos al jubilarse, al quedar inválido o en caso de muerte.
- Oportunidad.- A través de facilidades para ahorrar, asistencia educativa, vacaciones, días festivos, así como ayuda económica para disfrutar de sus días de descanso.

Usted puede decidir cómo afrontar sus contingencias individuales,

conociendo las formas en que IBM las cubre. Sólo de esta manera puede usted decidir si desea suplementar personalmente esa protección que IBM le ofrece.

Estos Planes de Prestaciones cubren a todos los empleados regulares de IBM de México.

El tiempo de servicio, constituye la base para recibir la protección de algunas prestaciones y se determina de acuerdo a los datos existentes en los archivos de la compañía.

IBM de México se reserva el derecho, de acuerdo a su juicio, de modificar los planes si eventos futuros indican que sea necesario hacerlo.

```
;fin(Prestaciones);
;.....
;* Este nodo contiene los Planes de Protección
;.....
;inicio(- Protección.)
PROTECCION.
```

-* Plan IBM Médico y de Hospitalización-

-* Plan IBM de Protección por Enfermedades y Accidentes-

-* Plan IBM de Examen Médico Periódico-

```
;fin(- Protección.)
;.....
;* Este nodo contienen los Planes de Seguridad
;.....
;inicio(- Seguridad.)
SEGURIDAD.
```

-* Plan IBM de Seguro de Grupo y Beneficio a los Sobrevivientes-

-* Plan IBM de Jubilación-

-* Paquete IBM para Empleados Jubilados-

-* Plan IBM de Ingresos por Incapacidad Total y Permanente-

-* Plan IBM de Accidentes en Viajes-

```
;fin(- Seguridad.)
;.....
;* Este nodo contiene los Planes de Oportunidad
;.....
;inicio(- Oportunidad.)
OPORTUNIDAD.
```

-* Plan IBM de Ahorros-

-* Plan IBM de Resembolso de Colegiaturas-

-* Plan IBM de Vacaciones-

-* Plan IBM de Días Festivos-

-* Plan IBM de Gratificación de Fin de Año-

-* Plan IBM Múltiple-

```
;fin(- Oportunidad.)
;.....
;* Los siguientes nodos están contenidos cada uno en un archivo
;* separado, con el fin de facilitar su actualización
```

```

;* Referirse al archivo correspondiente para los detalles de cada nodo
;.....
;inicio(* Plan IBM Médico y de Hospitalización)
;archivo() doc002.txt
;fin(* Plan IBM Médico y de Hospitalización)
;inicio(* Plan IBM de Protección por Enfermedades y Accidentes)
;archivo() doc003.txt
;fin(* Plan IBM de Protección por Enfermedades y Accidentes)
;inicio(* Plan IBM de Examen Médico Periódico)
;archivo() doc004.txt
;fin(* Plan IBM de Examen Médico Periódico)
;inicio(* Plan IBM de Seguro de Grupo y Beneficio a los Sobrevivientes)
;archivo() doc005.txt
;fin(* Plan IBM de Seguro de Grupo y Beneficio a los Sobrevivientes)
;inicio(* Plan IBM de Jubilación)
;archivo() doc006.txt
;fin(* Plan IBM de Jubilación)
;inicio(* Paquete IBM para Empleados Jubilados)
;archivo() doc007.txt
;fin(* Paquete IBM para Empleados Jubilados)
;inicio(* Plan IBM de Ingresos por Incapacidad Total y Permanente)
;archivo() doc008.txt
;fin(* Plan IBM de Ingresos por Incapacidad Total y Permanente)
;inicio(* Plan IBM de Accidentes en Viajes)
;archivo() doc009.txt
;fin(* Plan IBM de Accidentes en Viajes)
;inicio(* Plan IBM de Ahorros)
;archivo() doc010.txt
;fin(* Plan IBM de Ahorros)
;inicio(* Plan IBM de Reembolso de Colegiaturas)
;archivo() doc011.txt
;fin(* Plan IBM de Reembolso de Colegiaturas)
;inicio(* Plan IBM de Vacaciones)
;archivo() doc012.txt
;fin(* Plan IBM de Vacaciones)
;inicio(* Plan IBM de Días Festivos)
;archivo() doc013.txt
;fin(* Plan IBM de Días Festivos)
;inicio(* Plan IBM de Gratificación de Fin de Año)
;archivo() doc014.txt
;fin(* Plan IBM de Gratificación de Fin de Año)
;inicio(* Plan IBM Múltiple)
;archivo() doc015.txt
;fin(* Plan IBM Múltiple)

```

Con el objeto de explotar al potencial del hipertexto, es conveniente hacer uso, en la medida de lo posible y cuando la aplicación así lo requiera, de diversos medios de presentación de la información, aparte del textual. Para esta aplicación algunas de las entidades de tipo Documento son formas impresas que resulta muy útil poder integrar al hiperdocumento, de tal forma que para el lector sea posible visualizar la Solicitud de Reembolso de Gastos Médicos, por ejemplo, mientras está consultando el Procedimiento correspondiente. Dado que nuestra herramienta permite la

integración de herramientas externas, se decidió digitalizar algunos de los documentos más comunes, para poder integrar el despliegue de la imagen digitalizada durante la consulta del hiperdocumento. La digitalización se llevó a cabo utilizando un producto marca IBM llamado Personal Computer Image Document Utility (PC/IDU) y un scanner IBM modelo 3119 (blanco y negro, de cama plana). El despliegue de las imágenes se lleva a cabo utilizando una interfase gráfica del mismo producto, llamada IDUSHOW. Este es un programa que despliega la imagen digitalizada y permite hacer acercamientos (ZOOM) para poder apreciar detalles de la imagen.

VI.3 Recomendaciones prácticas para el diseño de hiperdocumentos.

Finalmente, presentaremos algunas recomendaciones que guiaron la implementación detallada del hiperdocumento. Debemos recordar que los objetivos principales al desarrollar un hipertexto son:

- * Hacer eficiente el acceso a la información
- * Incrementar la facilidad de uso de la información
- * Mejorar la percepción del usuario respecto a la información

En base a esto, se siguieron las siguientes consideraciones:

1. Acoplar las Estructuras Lógica y Física del Hipertexto.

Un hiperdocumento en realidad tiene dos estructuras implícitas:

a) La estructura lógica, que es establecida por el autor. Esta incluye la relación entre nodos a través de ligas (formando redes, árboles, gráficas), así como el contenido y diseño de nodos específicos.

b) La estructura física, que se refiere a las características del sistema en que se implementará el hipertexto, o sea, tipo de dispositivos de entrada/salida de información (resolución de monitores, teclado vs. ratón, etc.)

El autor del hiperdocumento debe en la medida de lo posible buscar un equilibrio entre ambas, de tal forma que el diseño del hiperdocumento sea factible de implementarse en la práctica. En nuestro caso de estudio, todos los nodos fueron diseñados para presentarse en un monitor de color tipo VGA de computadora personal. Del mismo modo, los documentos fueron digitalizados con una resolución apropiada para ser desplegados en un monitor del mismo tipo.

2. Tamaño y configuración de los nodos.

El tamaño de los nodos es una consideración importante de diseño. La experiencia práctica ha demostrado que es mejor mantener corto el tamaño de cada nodo. Una medida empírica puede ser limitar este tamaño a un número determinado de pantallas, por ejemplo, cinco por nodo como máximo. En nuestro caso, los nodos se mantuvieron en este límite.

Existe una razón para esta limitante, y tienen que ver con los efectos del exceso de información para un lector. Cuando éste recibe demasiados datos, puede perder el sentido de los que son relevantes, y entonces dejar de asimilar el valor de la información. Algunos criterios básicos para minimizar este efecto son:

- Proveer sólo la información necesaria inicial para asegurar que el lector comprenderá el propósito del nodo que está consultando.

- Identificar el conocimiento previo que el lector necesita para comprender la información que está consultando, y proveer ligas para que el lector pueda accederlo fácilmente y pueda interpretar correctamente el significado de la información. Recordemos que una premisa del hipertexto es que el usuario puede recorrer la información en forma arbitraria, de tal suerte que pueda llegar a una parte del documento sin haber revisado los antecedentes necesarios. Por ejemplo, en el Plan de Jubilación hay dos cláusulas: 6.Servicios continuos y 7.Retiro Anticipado. Están planteadas en ese orden sugiriendo que el lector consulte primero la de servicios, ya que en la de retiro anticipado se menciona este término. Sin embargo, si el lector decide consultar primero la segunda, es necesario asegurarse de que podrá acceder la definición de servicio continuo para que pueda entender el significado de retiro anticipado. Por lo tanto, se define una liga hacia la cláusula 6.

- Identificar los detalles que pueden derivarse de la información presentada al usuario, y asegurarse que haya ligas a un siguiente nivel de detalle de modo que el usuario que lo requiera pueda profundizar en algún aspecto. Por ejemplo, en nuestro hiperdocumento mostramos un nodo que describe el propósito del Plan Médico. En él se menciona que este Plan ampara al empleado y a sus familiares elegibles. Sin embargo no se abunda en la definición de familiar elegible. En vez de esto, se define la liga familiares elegibles para que en caso de necesitarlo, el empleado pueda obtener más información al respecto.

- Presentar la información en una secuencia lógica. A pesar de que el hipertexto proporciona libertad al usuario en la consulta de los nodos de información, es conveniente sugerir una secuencia de consulta cuando esto aplique. Por ejemplo, los pasos a seguir dentro del Procedimiento de Reembolso de Gastos Médicos son un ejemplo.

3. Limitar el número de ligas.

Recordamos que no toda la información disponible puede estar ligada entre sí. Frecuentemente, el exceso de ligas puede confundir al lector. Puede fijarse también un límite empírico de ligas por cada pantalla para evitar distraer demasiado al lector. Podríamos decir que de seis a ocho ligas por pantalla son un número razonable. Adicionalmente, cuando en un mismo nodo existe la posibilidad de establecer varias ligas que conduzcan al mismo lugar, es conveniente definir sólo una liga. Por ejemplo, si nos referimos en varios lugares a la "Solicitud de Vacaciones" dentro del mismo nodo, sólo deberíamos marcarla una vez como liga.

4. Utilizar sinónimos.

Es importante permitir al lector referirse a un mismo nodo a través de diferentes ligas cuyo concepto pueda ser el mismo pero expresado en diferente forma, aún si la diferencia es mínima, como por ejemplo en el caso de términos usados en singular y plural. Por ejemplo, si en un nodo nos referimos a los "familiares elegibles" y en otro hablamos de "familiar elegible", es claro que nos estamos refiriendo al mismo concepto y debemos conducir al usuario a un nodo común a través de ambas ligas.

VII. CONCLUSIONES.

VII.1 Evaluación de la herramienta desarrollada.

Considero que se alcanzó el objetivo fundamental planteado al inicio de este trabajo, y que fue el desarrollar una herramienta de propósito general para la construcción y consulta de sistemas de hipertexto. Utilizando dicha herramienta, una persona sin conocimientos previos puede en poco tiempo estar desarrollando sus propias aplicaciones de hipertexto en cualquier ámbito de información.

Una de las ideas principales que guiaron el diseño fue la posibilidad de aprovechar la información que normalmente cualquier organización va capturando en forma electrónica a través del tiempo. De ahí que se decidiera utilizar el mecanismo de comandos incorporados al texto como base para construir el hipertexto. Esto además facilita la portabilidad con otras herramientas, ya que el formato ASCII para almacenamiento de textos puede considerarse un estándar en el mercado de computadoras personales.

Otro criterio importante fue permitir la integración de otras herramientas. Actualmente existen productos altamente especializados en el manejo de ciertos tipos de información, como son los productos de multimedia. Esto hace que sea más eficiente integrar una herramienta para la manipulación de las imágenes o los sonidos dentro de un hipertexto, que pretender abarcar ese conocimiento dentro del desarrollo de la herramienta de hipertexto.

Dentro del ambiente de computadoras personales para el cual fue desarrollado este trabajo, existe un producto similar al nuestro, llamado Hyperties, el cual podemos

utilizar como marco de referencia para posicionar nuestra herramienta, estableciendo algunas ventajas y desventajas comparativas.

Hyperties es un producto de Cognetics Corporation, que se ha comercializado para la computadora IBM PC y compatibles, y que utiliza una estructura de presentación similar a la nuestra: en la pantalla de la computadora se muestran resultados los términos que sirven de ligas para navegar a otros nodos. Dentro de Hyperties, los nodos son llamados artículos y tienen todos la misma estructura: un título, una descripción breve (opcional) y el contenido. Una de las ventajas de Hyperties es que cuando el usuario selecciona una liga, se le muestra la descripción breve, de modo que si le interesa puede desplegar el nuevo artículo completo, y si no, continuar en el que se encontraba.

En Hyperties, una liga conduce siempre al inicio de un artículo completo. Es decir, no es posible direccionar fragmentos de información dentro del artículo. El usuario es conducido al principio, y él debe buscar la información. Una ventaja de nuestra herramienta es el concepto de nodos anidados, a través del cual es posible definir nodos dentro de otros nodos. Así, si nos interesa un fragmento dentro de un nodo, podemos definir un nodo anidado que abarque físicamente ese mismo fragmento, y que puede ser accedido desde cualquier otro nodo del hipertexto. Esto además evita la duplicación innecesaria de información, y asegura la consistencia de la misma.

Otra ventaja de nuestra herramienta sobre Hyperties es la posibilidad de activar y desactivar las ligas durante la consulta. En un momento dado, las ligas pueden distraer a un lector o hacer compleja la consulta de un nodo, por lo cual resulta muy útil poder "desaparecerlas" cuando se desee. Además, es posible hacerlo selectivamente, de acuerdo al tipo de liga, lo cual facilita la navegación dentro del hipertexto.

Una ventaja de Hyperties es que genera en forma automática un índice alfabético de todos los artículos contenidos en una base de datos. De esta forma, el lector puede acceder directamente cualquier artículo del hipertexto partiendo del índice. En el caso de nuestra herramienta, esta función fue considerada como una utilidad externa, y a continuación se describe como uno de los posibles desarrollos a futuro.

VII.2 Desarrollo futuro de la herramienta.

Existen algunas funciones que no han sido implementadas dentro de la herramienta de hipertexto, y que pueden proveerse en forma de utilidades adicionales a desarrollarse.

1. Análisis de un hiperdocumento.

Durante la creación de hiperdocumentos, especialmente de aquellos de gran tamaño, puede ser difícil para el autor llevar un control preciso de los nodos y ligas que ha creado. Es muy probable que pueda definir ligas que no conduzcan a ningún nodo, o bien construya nodos que no sean accedidos por ninguna liga. Así pues, el propósito de la utilidad propuesta es analizar un hiperdocumento y generar un reporte por cada uno de los documentos o archivos que lo componen, donde el autor pueda detectar estos errores y corregirlos. El parámetro de entrada a la utilidad sería el nombre del archivo de entrada a la red de hipertexto, y la salida consistiría en el conjunto de reportes, uno por cada documento del hipertexto, que proporcionarían para cada nodo información relativa a la ubicación del nodo dentro del documento, su tamaño, el número de posibles accesos a él, si es un nodo no referenciado, o si es un nodo que no ha sido definido, o bien si está duplicado.

2. Generación automática de índices.

Como se mencionó anteriormente, sería de gran utilidad para el lector disponer de un índice que lo pueda conducir a los distintos nodos del hiperdocumento. La segunda utilidad propuesta tiene esta función, y su trabajo es procesar los archivos que componen el hiperdocumento y generar un nuevo archivo que contenga un "nodo índice". Este índice no sería más que una lista ordenada alfabéticamente de ligas a todos los nodos de hipertexto de modo que el usuario pueda moverse en él al igual que lo hace en cualquier otro nodo del hiperdocumento, seleccionar la liga que desea y navegar a él. La única tarea que el autor tendría que hacer manualmente sería incluir en los nodos del hiperdocumento que lo desea una liga que conduzca a este nuevo "nodo índice", recordando que debe usar la tag `{archivo}` para indicar que el nodo se encuentra en un archivo externo.

Adicionalmente, sería conveniente dejar que el autor decida que nodos formarán parte del índice, por lo cual puede definirse una nueva tag, llamada `{indice}`, que indique a la utilidad que sólo aquellos nodos que contengan esta tag serán incluidos. Durante la consulta del hipertexto esta tag será ignorada al igual que un comentario, ya que sólo tiene significado para la generación automática del índice.

J. Búsqueda de textos a través de la red de hipertexto.

Actualmente la herramienta proporciona la facilidad de búsqueda de textos dentro del nodo activo. Sin embargo sería posible proporcionar una utilidad adicional que permitiera al usuario buscar un texto dentro de todos los nodos que componen un hiperdocumento determinado. Esto le facilitaría la localización de información. Esta utilidad podría ser invocada a través de un comando durante la consulta del hipertexto, y su parámetro de entrada sería el texto que se desea buscar. En forma similar a la generación automática de índices, la salida de esta utilidad podría ser el despliegue de un nodo temporal que tuviera una liga a cada uno de los nodos en los que se encontró el texto buscado para que el usuario pudiera navegar a

ellos. Este nodo, a diferencia del índice, sería temporal porque se destruiría cuando el usuario saliera de él.

4. Creación de un editor especializado.

Esta utilidad sería dirigida a los autores de hipertextos, con el propósito de facilitar la creación y/o adecuación de documentos, y su función básica sería proporcionar formas rápidas de definición de nodos y ligas, y codificación de tags. Por ejemplo, podrían definirse teclas de función para la inserción automática de los tags de inicio y fin de nodo en el archivo, para la definición automática de ligas, o bien para la codificación automática de tags proporcionando sólo los valores deseados para sus parámetros. Esto evitaría posibles errores de sintaxis al crear los documentos y agilizaría la labor de los autores de hipertexto, permitiéndoles concentrarse más en el diseño de los documentos y en su contenido.

VII.3 Evaluación del caso de estudio.

La implementación de un hipertexto para el Programa de Beneficios de IBM tiene ventajas significativas sobre los métodos tradicionales de manejo de información, como lo demuestra el prototipo que hemos desarrollado en la presente Tesis:

Toda la información requerida al alcance de la mano. Con un sistema de hipertexto, un empleado puede ahora tener acceso a toda la información relacionada con el tema que está consultando. Esta información no solamente es textual, sino que se complementa con imágenes.

Precisión en la consulta de información. La naturaleza lineal del texto tradicional, aún en forma electrónica, dificulta la obtención rápida de fragmentos específicos de información dentro de un conjunto de documentos. El hipertexto

facilita al usuario la selección de la información que le es útil, a través del uso adecuado de los nodos y ligas provistos.

Adaptabilidad del sistema a las necesidades de información del usuario. El hipertexto resulta útil para los usuarios independientemente de sus requerimientos de información. Para quien busca detalles, puede proporcionarlos en base a solicitud del lector. Para quien quiere información de nivel general, resulta valioso no tener que invertir más tiempo revisando información que no se requiere.

Disponibilidad de versiones actualizadas de los documentos. La naturaleza electrónica de los documentos los hace susceptibles de actualización periódica y disponibilidad inmediata para los usuarios. Adicionalmente, dado que la información no se duplica sino que se relaciona entre sí a través de ligas, se logra consistencia. Todos los empleados consultarán los mismos Planes y Procedimientos.

Facilidad de actualización. Para los autores del hipertexto, y encargados de actualizar y mantener los documentos electrónicos, es fácil agregar nuevos nodos a la red de hipertexto, o modificar los ya existentes, así como especificar las ligas a esas nuevas entidades de información.

Facilidad para registrar la consulta de la información. A través de la facilidad de registro provista por la herramienta, es posible dar seguimiento a los usuarios durante su interacción con el sistema. Se puede saber qué nodos han sido consultados y con qué frecuencia. Esta información puede resultar muy útil para adaptar el sistema cada vez más a los requerimientos de los usuarios.

La aplicación que elegimos como prototipo se refiere al área de Personal. Sin embargo, dentro de la empresa el hipertexto tiene potencial de aplicación en muchas

otras áreas donde se manejan en forma similar procedimientos, cartas de instrucción, planes, etc. Nos referimos a áreas como Administración de Ventas, Finanzas, Seguridad, Procesos de Manufactura, etc. en las cuales sería factible desarrollar proyectos similares.

VII.4 Conclusión.

Finalmente, considero que se cumplieron todos los objetivos de la tesis, al haber diseñado un Sistema de Hipertexto en base a un modelo conceptual elegido, para su posterior implementación utilizando una metodología estructurada de Diseño de Sistemas.

Después se comprobó la aplicabilidad del Sistema desarrollado en un caso práctico, en el cual se justificó la construcción de un hipertexto como alternativa más adecuada para el almacenamiento y consulta de información.

En términos generales, considero que se desarrolló un programa innovador y con un nivel adecuado para competir en el mercado de computadoras personales.

Referencias Bibliográficas.

- [01]. Shneiderman, B.(1989): "Reflections on authoring, editing and managing hypertext". The Society of Text, MIT Press, Cambridge, MA.
- [02]. R.L.Mack, J. Nielsen(1987): "Software integration in the professional work environment: Observations on requirements, usage and interface issues". Technical Report RC-12677, IBM T.J. Watson Research Center.
- [03]. Bigelow, J. (1988): "Hypertext and CASE", IEEE Software 5,2
- [04]. Conlin J., Begeman M.L.(1988): "gIBIS: A hypertext tool for exploratory policy discussion", ACM Trans. Office Information Systems 6,4
- [05]. Feiner, S., Nagy, S., van Dam, A.(1982): "An experimental system for creating and presenting interactive graphical documents", ACM Trans. Graphics 1,1
- [06]. Raymond, D.R. and Tompa, F.W. (1988): "Hypertext and the Oxford English Dictionary". Communications of the ACM 31,7
- [07]. De Young, L. (1989): "Hypertext challenges in the auditing domain". Proc. ACM Hypertext'89 Conf. (Pittsburgh, PA. 5-8 November)
- [08]. Andersen M.H., Nielsen J., Rasmussen H. (1989): "A similarity-based hypertext browser for reading the Unix network news". Hypermedia 1,3
- [09]. MacPhail, Wayne (1990): "Hypertext and Journalism: Towards a New Mass Medium", Metaphor-The Hypermedia Group, Inc.
- [10]. Campbell B. and Goodman J.M. (1988): "HAM: A general purpose hypertext abstract machine", Communications of the ACM 31,7
- [11]. Kretzberg, C.R. and Shneiderman B. (1989): "Restructuring knowledge for an electronic encyclopedia". Proc. Intl. Ergonomics Association 10th Congress (Sydney, Australia, 1-5 Agosto)
- [12]. Kirby M.R. and Mayes J.T. (1989) : "Towards intelligent hypertext". in McAleese, R. : Hypertext Theory into Practice, Ablex
- [13]. Toulmin, S.E. (1989) : "The Uses of Argument". Cambridge University Press.
- [14]. Balasz, F.G. y Schwartz, M.(1990): "The Dexter hypertext reference model", Proc NIST Hypertext Standardization Workshop (Gaithersburg, MD)
- [15]. Garzotto, F. Paolini, P. (1990): "The HDM data model for hypertext applications: An informal introduction.", Tech. Report., Dept. of Electronics, Politecnico di Milano.

[16]. Berk, Emily and Devlin, Joseph (1991): "Hypertext/Hypermedia Handbook", Intertext Publications, McGraw Hill.

[17]. Frisse, Mark P., Cousins, Steve B. (1992): "Models for Hypertext", Journal of the American Society for Information Science, March.

[18]. Rada, Roy (1989): "Writing and Reading Hypertext: An Overview", Journal of the American Society for Information Science, May.

[19]. Nielsen, Jacob (1991): "Hypertext & Hypermedia", Technical University of Denmark, Academic Press, Inc.

[20]. Pressman, Roger S. (1984): Software Engineering: A Practitioner's Approach. International Student Edition. McGraw Hill.

[21]. Bradley, J. David (1984): Assembly Language Programming for the IBM Personal Computer. IBM Corporation, Prentice Hall.

[22]. Kernighan, Brian and Ritchie, Dennis (1978): El Lenguaje de Programación C. Prentice Hall.