

Autores: M. Ines Arrequi
 Patricia Harkes
 Carlina Armendano

Director: Gustavo Rossi

TES
 94/3
 DIF-02448
 SALA



UNIVERSIDAD NACIONAL DE LA PLATA
 FACULTAD DE INFORMATICA
 Biblioteca
 50 y 120 La Plata
 catalogo.info.unlp.edu.ar
 biblioteca@info.unlp.edu.ar



DIF-02448

A Nuestros Padres



INTRODUCCION

Este trabajo presenta nuestra experiencia en la definición de un Modelo de Diseño de Hypermedia (MDH) y la implementación del mismo en un prototipo al que hemos llamado **Hypermedia CIP**.

Nuestro objetivo era obtener un prototipo que solucionara la problemática del almacenamiento de grandes volúmenes de información periodística y la consulta de la misma de una forma variada y fácil de ejecutar, no solo operacionalmente sino visualmente.

Hemos utilizado al Hypertexto como base para el manejo de la información, debido a que este permite organizar y recuperar datos en forma eficiente.

Las notas periodísticas pueden pertenecer a varios medios por ejemplo texto, sonido, imagen y animación por lo cual decidimos incorporar Multimedia.

De estos dos conceptos surge **Hypermedia CIP**.

Para obtener el Modelo de Diseño de Hypermedia utilizamos conceptos existentes para la edición y diseño de Hypertextos tradicionales, así como también conceptos de Multimedia, contextos anidados y perspectivas o visiones.

Se ha incorporado el concepto de contextos anidados como una forma de organizar jerárquicamente la información de la hyperbase (estructura de la hypermedia) en nodos contexto. Esto permite una mejor recuperación de los datos, ya que éstos pueden ser anidados en cualquier profundidad. En **Hypermedia CIP** cada contexto sería un tema distinto al que puede pertenecer una nota.

Un mismo nodo puede pertenecer a distintos nodos contexto. Hemos usado a las perspectivas como contenedores de las consultas de nodos contexto, de manera



que el usuario puede fácilmente pasar de una perspectiva a otra durante la navegación de los nodos contexto de la Hypermedia. Las distintas perspectivas pueden ser combinadas de forma tal que múltiples nodos contexto pueden ser accedidos separadamente o como una unidad.

Un nodo contexto contiene los identificadores de nodos que pertenecen a ese contexto, no los datos de los nodos, evitando de esta manera duplicación de información.

Perdersse es una experiencia habitual en el uso de Hypertextos. Con los conceptos de contextos anidados y perspectivas, se disminuye la dificultad de los usuarios de navegar por la Hypermedia, ya que cuanto más se profundiza en la jerarquía de nodos, se obtiene información más específica, lo cual permite acotar la perspectiva de la consulta, siendo de esta forma más fácil encontrar información dentro de la red general de la Hypermedia.

Con respecto a la estructuración del texto que presentamos a continuación, consideramos conveniente dividirlo en 5 capítulos. En el primero de los mismos explicamos los fundamentos teóricos usados para la definición e implementación del proyecto. El capítulo 2 está relacionado con la especificación del modelo sobre el cual basamos Hypermedia CIP y las herramientas que tuvimos en cuenta para obtenerlo. En el capítulo 3 explicamos el prototipo implementado. El capítulo 4 contiene una lista de las posibles ampliaciones y mejoras que se le podrían hacer al trabajo y finalmente el capítulo 5 presenta nuestras conclusiones acerca de lo realizado.



INDICE

1. CONCEPTOS Y FUNDAMENTOS TEORICOS	
A. HYPERTEXTO	1
1.1 HISTORIA DE HYPERTEXTO	1
1.2 FUNDAMENTACION POR MEDIO DE UN EJEMPLO	2
1.3 DEFINICIONES DE HYPERTEXTO	3
1.4 COMPARACION DE HYPERTEXTO CON OTRAS ESTRUCTURAS	6
1.5 ESENCIA DEL HYPERTEXTO	8
1.6 CARACTERISTICAS DE SISTEMAS DE HYPERTEXTOS	9
1.6.1 NODOS	11
1.6.2 LINKS	12
1.6.2.1 PROPIEDADES DE LOS LINKS	14
1.6.2.2 TIPOS DE LINKS	15
1.6.3 PUNTOS Y BOTONES	16
1.7 HYPERBASE	17
1.8 NAVEGACIÓN	18
1.9 AREAS DE APLICACION	19
1.10 VENTAJAS Y DESVENTAJAS DEL USO DEL HYPERTEXTO	20
1.10.1 DESVENTAJAS	20
1.10.2 VENTAJAS	21
VENTAJAS OPERACIONALES DEL HYPERTEXTO	22
B. MULTIMEDIA	24
"EL MEDIO ES EL MENSAJE"	24
1.11 INTRODUCCION DE MULTIMEDIA	25
1.12 TRES DEFINICIONES PARA MULTIMEDIA	27
1.13 MULTIMEDIA NO INTERACTIVO?	28
1.14 BLOQUES DE CONSTRUCCION EN MULTIMEDIA	29
1.15 IDENTIFICACION DE MERCADOS DONDE SE APLICA MULTIMEDIA	
	31
1.16 LA MULTIMEDIA DEL MAÑANA	31



C. HYPERMEDIA	33
1.17 INTRODUCCION DE HYPERMEDIA	33
1.18 CARACTERISTICAS PARA LA NUEVA GENERACION DE HYPERMEDIA	35
1.19 INCREMENTANDO EL PODER DE BUSQUEDA CON CONSULTAS RELACIONALES	40
1.20 LINKEO DE ANCHORS ACTIVOS (INTERMEDIA)	41
2. MODELO PROPUESTO	
2.1 ESPECIFICACION DEL MODELO	45
2.1.1 INTRODUCCION	45
2.1.2 NODOS	46
2.1.3 LINKS	47
2.1.4 BOTONES	52
2.1.5 HYPERBASE	53
2.2 PERSPECTIVAS EN AMBIENTES DE HYPERTEXTOS	55
2.2.1 DEFINICIÓN DE PERSPECTIVAS	55
2.2.2 VENTAJAS DE LAS PERSPECTIVAS	57
2.2.3 ESTRUCTURAS VIRTUALES	58
2.3 CONTEXTOS ANIDADOS	59
2.3.1 MODELO	59
2.3.2 DEFINICION DE SUBMODELO	60
2.4 CONSULTAS A LA HYPERBASE	63
2.5 INTERFASE	65
2.6 POR QUE UNA APLICACION DE MANEJO DE NOTAS PERIODISTICAS?	65
3. PROTOTIPO IMPLEMENTADO	
3.1 LENGUAJE Y PLATAFORMA USADA - JUSTIFICACION	69
3.1.1- LENGUAJE	69
3.1.2 PLATAFORMA USADA	76
3.2 CARACTERISTICAS DEL SISTEMA	78
3.2.1 REQUERIMIENTOS	78



3.2.2 ESTRUCTURA UTILIZADA	78
3.2.3 TRATAMIENTO DE ANIMACION Y SONIDO	81
3.2.4 INTERFASE Y OPERACION	82
3.2.5 PERFIL DE USUARIO	87
4. DISCUSIONES	88
5. CONCLUSIONES	91

**CONCEPTOS Y FUNDAMENTOS
TEORICOS**



A. HYPERTEXTO

1.1 HISTORIA DE HYPERTEXTO

Vannevar Bush diseñó una máquina para estándares de hoy en día. Usaba microfilm y fotocélulas para cargar sus datos. Pero soñaba con un esquema de organización y recuperación de información semejante al de un **Hypertexto**, por lo que se lo puede considerar un pionero.

El primer investigador influenciado por Bush en los conceptos de asociación y *browsing* fue Douglas Engelbart. Su investigación en los años 60 en el Stanford Research Institute se centro en usar a las computadoras para aumentar el intelecto humano. En ese momento, comenzó a desarrollar un on-line System, o NLS, ahora llamado Augment y usado internamente para varios productos en Mc Donnell-Douglas.

Augment es un ambiente de trabajo *on-line*. En sus formas iniciales, sirvió como un receptáculo de carga para memos, notas de investigación y documentación; como una red de comunicación, desde que las conferencias *on-line* fueron posibles, como un lugar de trabajo compartido donde investigadores podían planear y diseñar proyectos.

Aún corriendo en una DEC 20, Augment cargaba información en una estructura jerárquica sofisticada permitiendo ramificaciones no jerárquicas. Ya que la velocidad era importante Engelbart inventó el *mouse*, como un dispositivo de salida. El también comenzó con el concepto de visiones filtradas. Vía filtros se puede ver una visión recortada del estado o archivo, lo cual le permite moverse rápidamente a través de la base de datos del **Hypertexto**, en búsqueda de los datos deseados solamente. Engelbart fue el primero en usar un sistema de Ayuda dependiendo del contexto, un sistema de correo incorporado, múltiples ventanas y una pantalla compartida.

Mientras estos desarrollos ayudaron a los investigadores en problemas



multidimensionales complejos, Ted Nelson tomó el concepto de Bush como un paso importante. Nelson vio el **Hypertexto** como una red de trabajo *on-line*, sustentando al mundo literario jerárquicamente.

Xanadu es el sistema de **Hypertexto** más conocido, fue creado por Nelson, quién originalmente inventó la palabra **Hypertexto** hace 23 años para significar la escritura no secuencial.

Como diseño, Xanadu fue y es un ambiente muy expandido que millones de personas pueden usar para crear, interactuar e interconectar documentos electrónicos *linkeados* y otras formas de Hypermedia, tales como video, audio, gráficos, etc. Fue diseñado para correr en paralelo en muchas redes de servidores. En un nivel básico, un manejador de carga le permitirá crear *links* entre tópicos y después volver al origen, variaciones e interconexiones de texto.

1.2 FUNDAMENTACION POR MEDIO DE UN EJEMPLO

Imagínese que está caminando por la Biblioteca Pública de Nueva York y toma un libro de Mozart. Usted comienza a leer y se entera que fue un compositor austriaco del año 1700. Usted quiere saber que pasaba en Austria entonces, por lo tanto va a un catálogo y busca un libro con la historia de Austria, va hacia la pila, ubica el libro y lo lee antes de continuar.

En este libro encuentra una referencia al viejo Salzburgo y usted quiere saber como es. Regresa al catálogo y a las pilas, a buscar un libro con imágenes. Finalmente usted regresa a Mozart y lee sobre un concierto de piano que nunca ha escuchado.

Este proceso continúa hasta que ha satisfecho los deseos de conocimiento de un sujeto o ellos han terminado con usted, que es lo que generalmente pasa.

Ahora imagínese sentado en su computadora manejando un sistema de



Hypertexto con música. Usted comienza a leer acerca de Mozart. Cuando usted desea saber acerca de la historia de Austria, simplemente resalta el texto y requiere más información con el *cliqueo* del *mouse*. Para buscar imágenes de Salzburg, realiza el mismo proceso. Y para escuchar un concierto de piano, lo mismo.

Esto es mucho más simple No?, La única restricción es que usted podrá acceder a esto si el autor del sistema de **Hypertexto** ha establecido la conexión de la información requerida y provee conocimiento adicional para que usted lo pueda recuperar.(Ref. 2)

1.3 DEFINICIONES DE HYPERTEXTO

Esta nueva tecnología abre muchas posibilidades existentes, particularmente para nuevos usos de la computadora como herramienta de comunicación y de pensamiento.

- 1 - Ted Nelson definió al **Hypertexto** como una combinación de texto en lenguaje natural con capacidad de computación para ramificaciones interactivas o muestreo dinámico de un texto no lineal, el cual no puede ser impreso en una página convencional.

- 2 - **Hypertexto** es una red de nodos conectados por *links*, los nodos pueden contener textos, gráficos y otras formas de datos.

Los nodos, y algunos sistemas de red, podrían verse como un *browsers* interactivo manipulado por una estructura de editor. Un **Hypertexto** presenta de diferentes maneras un documento tradicional.

En la mayoría de los documentos tradicionales (novela, artículos de



periodismo, especificación) la estructura física y lógica están estrechamente relacionadas; físicamente el documento es una larga secuencia de palabras que han sido divididas en líneas y páginas por conveniencia. Lógicamente el documento también es lineal, las palabras son combinadas en secuencias, las secuencias en párrafos y los párrafos en secciones.

En algunos documentos convencionales tales como enciclopedias, diccionarios y otros trabajos de referencias, la estructura lógica está separada de la estructura física.

Físicamente estos documentos son secuencias lineales de unidades independientes tales como artículos de temas específicos o entradas a palabras individuales. Lógicamente son más complejos, el lector rara vez lee el documento desde el principio hasta el fin, sino que busca en ellos encontrar el artículo o entrada de interés y entonces leer la porción secuencialmente. Sin embargo el lector está forzado a realizar varias secuencias cruzadas a otras entradas mientras lee. Para hacer esto el lector debe localizar el volumen apropiado y encontrar la entrada.

La estructura lógica de referencias es más compleja, ellos tienen una estructura secuencial que ayuda a la búsqueda, pero el camino lógico del lector es una red que puede cruzar el documento entero o conjunto de documentos desde un ítem u otro. Tales documentos son más flexibles pero también incómodos, particularmente cuando ellos aparecen en formatos grandes o multivolumenes.

Los documentos de **Hypertextos** electrónicos proveen más flexibilidad a los trabajos de referencias, así como también agregan nuevas características. Cada nodo del Hypertexto puede ser visto como una analogía a una corta sección de un artículo, de una enciclopedia o una imagen gráfica con una corta explicación. Los *links* son vistos por cada nodo como un *link* "desde" y un conjunto de *links* "hacia" que indica los

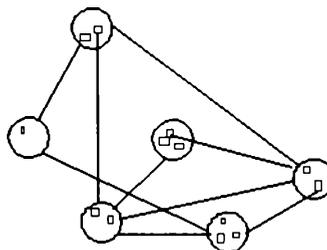


múltiples nodos que uno puede seleccionar próximamente. Por lo tanto uno se mueve de un nodo a otro seleccionando el *link* "hasta" deseado, un *link* de referencia interna o el *link* "desde" para retornar al nodo previo.

Mientras que la organización y la estructura de referencias cruzadas de documentos convencionales son fijas, los *links* y nodos del **Hypertexto** pueden ser cambiados dinámicamente.

La información a nodos puede ser actualizada, nuevos nodos pueden ser *linkeados* dentro de toda la estructura del **Hypertexto** y se pueden agregar nuevos *links* para mostrar nuevas relaciones. Los documentos de **Hypertexto** son mucho más flexibles que los documentos convencionales. Por ejemplo, uno puede leer un artículo en un Hypertexto como uno lee un artículo convencional, primero lee el nodo de presentación, después los nodos de la primera sección, la segunda sección, etc; sin embargo uno puede leer las secciones en diferente orden, por ejemplo, si la estructura jerárquica de un artículo es vista como un árbol de dos dimensiones o como un mapa, uno puede imaginar que si los "hasta" son mostrados como hijos del nodo corriente, se puede seleccionar la segunda sección antes que la primera, o saltar la primera sección entera.

Gráficamente un **Hypertexto** se podría ver de la siguiente manera:





1.4 COMPARACION DE HYPERTEXTO CON OTRAS ESTRUCTURAS

Una característica adicional común a muchos sistemas de **Hypertextos** es el uso de ventanas que tiene una correspondencia uno a uno con nodos de la base de datos.

Ciertos sistemas tienen algunos de los atributos de **Hypertexto** pero no tienen todas las propiedades de los mismos. Mientras los sistemas de ventanas tienen una funcionalidad en la interfase, y por esto algún sentido como **Hypertexto**, los sistemas de ventanas no tienen una única ventana de soporte y por lo tanto le faltaría el aspecto de base de datos sobre el cual está soportado el **Hypertexto**.

Los sistemas de archivos tampoco se los puede clasificar como **Hypertextos**; uno puede pensar que un sistema de archivos es una base de datos y que uno se mueve a través de los nodos archivos invocando un editor con su nombre. Sin embargo para calificarlo como **Hypertexto** un sistema debe usar una noción más sofisticada de *links* y debe proveer un mecanismo de soporte de estos *links* más que tipear simplemente el nombre del archivo después del *prompter* de un editor de texto.

Los sistemas de manejo de bases de datos tienen *links* de varias clases (ej: relacionales) pero les falta una única interfase coherente a la base de datos que es la marca de calidad del **Hypertexto**. Los programas de **Hypertextos** corrientes no usan registros típicos de Base de Datos y archivos estructurados, sus Bases de Datos consisten de lugares de trabajo con la dimensión de una pantalla llamados nodos. La mayoría de los **Hypertextos** implementan los nodos jerárquica o no jerárquicamente, algunos soportan ambas estructuras.

Los diseñadores vieron a los **Hypertextos** como un ambiente para interconectar escritura y carga literaria o más sofisticadamente como un ambiente de investigación multipropósito que integra el pensamiento corporativo en proyectos compartidos.

Los sistemas de **Hypertextos** están en muchos ambientes que soportan tareas variadas. El *soft* típico de **Hypertexto** consiste en un editor de texto, un editor de gráficos, base de datos, herramientas de *browsing* para visiones en 3 dimensiones (El



browser es usualmente un gráfico que es usado para orientarse dentro de una base de datos con muchos nodos), un *mouse*, ventanas, íconos y menús desplegables. Estas son todas las herramientas de los **Hypertextos** standard.

La mayoría de los sistemas de **Hypertextos** tienen una base de datos particular y no un sistema de base de datos standard.

Cuando se usa una aplicación de **Hypertexto** solamente se ve la interfase usuaria, los mecanismos de base y la base de datos están ocultas.

Como usuario del sistema se tiene acceso a un número de capacidades de indexación. Se puede, crear frases o palabras claves en un contexto o realizar búsquedas booleanas o por palabras. Algunos programas le permiten crear jerarquías indexadas, como tablas de contenidos, mientras otros le permiten crear indexaciones basadas en contenidos. Otros sistemas le permiten crear ambas.

Si se escribe una aplicación o se usa un sistema que no hace la delimitación entre el autor y el usuario, se tiene acceso a editar, *linkear* y a las herramientas para desarrollar el **Hypertexto**. Se pueden lograr aplicaciones simples o complejas dependiendo del sistema de **Hypertexto** que se usa. Muchos permiten invocar programas desde la aplicación misma a través del *mouse*. Estos programas pueden ser cortos o programas que normalmente corren desde el sistema operativo.

En el nivel más básico es un sistema que le permite conectar pantallas de información usando asociaciones. En el más sofisticado el **Hypertexto** es un ambiente de *software* de trabajos colaborativos, comunicaciones y requerimiento de conocimientos adicionales.

Hay un creciente interés en extender la noción de **Hypertexto** a un concepto más general de hypermedia, en el cual los elementos pueden ser textos, gráficos, audios, video, animación, sensaciones táctiles.(Ref. 1)



1.5 ESENCIA DEL HYPERTEXTO

El **Hypertexto** es un medio basado en computación para pensar y comunicarse.

Los textos tradicionales presentan párrafos en una sucesión lineal. Tienen marcas como comentarios parentizados, notas al pie de página, referencias interseccionales (por ej. ver capítulo 4), referencias bibliográficas; hay muchas formas para indicar que las ideas están juntas como un conjunto pero son presentadas en una secuencia lineal.

Los **Hypertextos** permiten y alientan al escritor a hacer muchas referencias y al lector a tomar sus propias decisiones de cuales *links* (o caminos) seguir y en que orden. Estos también permiten anotaciones (o referencias) para ser salvadas separadamente del documento. Por medio de los *links* puede extenderse el texto más allá de su flujo lineal.

Desde el punto de vista de la computación, el **Hypertexto** es un híbrido que corta los límites tradicionales, es un método de base de datos que provee una mejor forma de acceder a los datos, este método es bastante diferente del tradicional método de consultas.

Al mismo tiempo el **Hypertexto** es un esquema de representación, una clase de red semántica que mezcla material de texto informal con operaciones formales y mecanizadas. Finalmente el **Hypertexto** es una modalidad de interfase caracterizada por el control de botones (íconos), los cuales pueden estar arbitrariamente incorporados en el material del usuario. (Ref. 1)

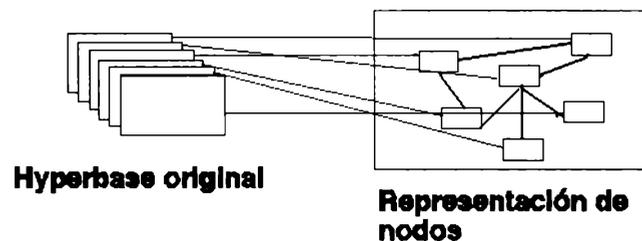
Para Janet Fiderio: " Lo único standard en los sistemas de Hypertextos es que no hay standard." (Ref.3)



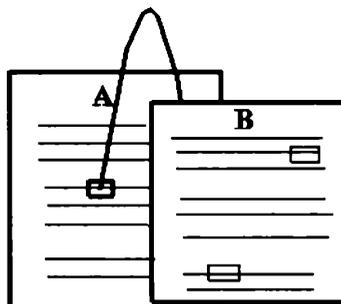
1.6 CARACTERISTICAS DE SISTEMAS DE HYPERTEXTOS

- * La base de datos es una red de nodos textuales la que puede ser pensada como una clase de hyperdocumentos.
- * Las ventanas en la pantalla tienen una correspondencia "uno a uno" con un nodo en la base de datos y cada uno tiene un nombre o título el cual es presentado en la ventana, sin embargo solo un pequeño número de nodos son abiertos en la pantalla al mismo tiempo.

RELACION DATOS - NODOS



- * Son soportadas operaciones standard en sistemas de ventanas, las ventanas pueden ser reubicadas, redimensionadas, cerradas y colocarles íconos. *Clickeando* con el *mouse* en el ícono de una ventana cerrada, causa que la ventana se abra instantáneamente.
- * Las ventanas pueden tener cualquier número de íconos que representen *links* a otros nodos en la base de datos. El ícono *link* contiene un pequeño campo de texto que identifica el nodo destino señalado por él.





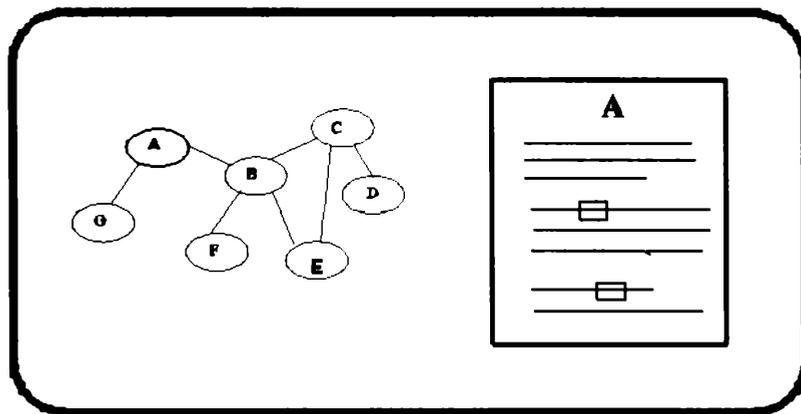
* El usuario puede fácilmente crear nuevos nodos y nuevos *links* a nuevos nodos o a nodos existentes.

* La base de datos puede ser mostrada de tres maneras:

1) siguiendo y abriendo ventanas sucesivamente para mostrar sus contenidos;

2) buscando en la red (o en una parte de ella) alguna palabra clave o valor de atributo;

3) navegando a través del hyperdocumento usando un *browsers* que muestre la red gráficamente.



A continuación se describen los componentes básicos de un **Hypertexto** como nodos, *links*, botones, hyperbase y parte de la funcionalidad del mismo.



1.6.1 NODOS

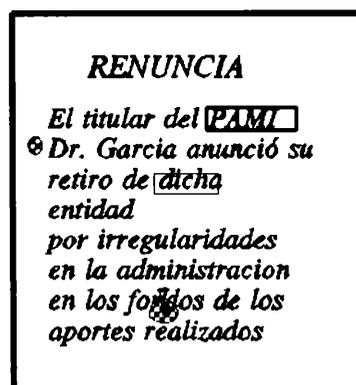
La gente no piensa en términos de pantallas, piensan en términos de ideas, hechos y evidencias. Los **Hypertextos**, vía la noción de nodos como expresiones individuales de ideas, proveen un vehículo que respeta esta forma de pensar y de trabajar.

Para usar un sistema de **Hypertexto** se debe particionar la información en unidades pequeñas, o nodos, el cual consiste en un concepto simple o ideal. La información contenida en un nodo puede usualmente ser mostrada en la pantalla de una computadora. En situaciones donde se necesita más espacio, algunos permiten crear nodos más grandes que el tamaño de la pantalla que se *scrolean* desde un botón de la misma.

Los nodos pueden ser de dos clases: tipados o no tipados. Un nodo no tipado es una caja de información. No tiene *label* o descriptor, se lo puede llenar con cualquier cosa. Un nodo tipado es nombrado, y el descriptor lo ayuda a determinar el tipo de información contenida en el mismo. Los tipos ayudan a clasificar los nodos o definir operaciones especializadas para ellos.

Estos tipos de nodos pueden ser útiles particularmente si uno esta considerando darles alguna estructura interna, ya que los tipos pueden ser usados para diferenciar las distintas formas estructurales.

Gráficamente un nodo puede por ejemplo representarse de la siguiente manera:





Los **Hypertextos** permiten el ocultamiento de información, la única guía que un usuario tiene de los contenidos del nodo destino es el nombre del *link*. El que escribe no toma todas las decisiones del flujo del texto, el lector puede y debe decidir cuales *links* seguir. Por lo tanto el **Hypertexto** impone a ambos, al lector y al escritor un proceso de conocimiento previo. (Ref. 3)

1.6.2 LINKS

Para calificar a un sistema como un **Hypertexto**, este no debe requerir más de dos movimientos del *mouse* cuando el usuario sigue un *link*, es decir que la interfase debe proveer *links* que actúen como botones mágicos que transporten al usuario rápida y fácilmente en el Hyperdocumento.

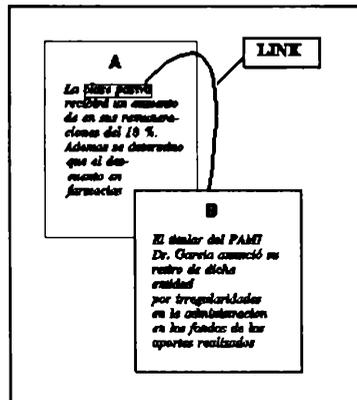
En general los *links* son utilizados para conectar nodos. Un *link* de **Hypertexto** es como una nota al pie, una nota final o una frase entre paréntesis. Es decir permiten a los lectores relacionar puntos o tópicos importantes. Los *links* de **Hypertexto** permiten asociar texto o información subordinada o dependiente.

Los *links* pueden hacer más que conectar dos nodos. Dependiendo del sistema de **Hypertexto**, los *links* pueden conectar anotaciones a un documento (incluyendo notas y comentarios) y proveer información organizacional, tal como dónde se ubica el texto en la tabla de contenidos o dónde se originó el *link*. Por lo tanto, los *links* pueden ayudar a definir relaciones de nodos a otros nodos dentro de la base de datos. Los *links* también pueden clarificar los contenidos de un diagrama o gráfico, conectando los gráficos para aumentar la información.



Los *links*, por lo tanto, son el modo de transitar la red del **Hypertexto**. Siguiéndolos, uno se puede mover entre varios nodos. Se los puede crear, editar, borrar o cambiarle los atributos.

Gráficamente se puede representar un *link*, a modo de ejemplo de la siguiente forma:



Hay dos métodos para *linkear* dos puntos en un **Hypertexto**: por referencia y por organización:

- * El método de referencia es un método no jerárquico que usa *links* referenciales que conectan puntos o regiones en el texto.

El origen del *link* es llamado *link* "fuente" y actúa como la referencia, el destino del *link* funciona como lo referenciado.

Un "*link* punto"(botón) es algún ícono que indica la presencia de un *link*, usualmente muestra el nombre del *link* y además su tipo, o puede mostrar el nombre y/o el tipo del nodo destino. Un "*link* región" es un conjunto de caracteres contiguos que son considerados como una unidad.

Se debe proveer un mecanismo para mostrar una región seleccionada al usuario; funciones para copiar, mover, modificar, borrar la región y los *substrings* dentro de ella.



- * Los organizacionales difieren de los anteriores en que implementan una jerarquía de información. Estos conectan un nodo padre con sus hijos y forman un subgrafo del grafo general del **Hypertexto**.

Una de las ventajas de los sistemas estrictamente jerárquicos, es que los comandos del lenguaje para navegar son simples; desde cualquier nodo uno puede ir a su padre, a su hermano o a su hijo, esta simplicidad también disminuye el problema de desorientación ya que conociendo el modelo de la información será suficiente. La gran desventaja de cualquier jerarquía es que su estructura está en función de los criterios específicos con que se creó, el creador de la jerarquía debe dar a conocer el criterio más importante para poder acceder a la información. Una solución a este dilema es permitir a los elementos de información ser estructurados en múltiples jerarquías.

Una de las principales ventajas de la carga de texto en una computadora es la habilidad de buscar grandes y complejos documentos por un conjunto de caracteres o palabras claves. Desde el punto de vista funcional, el seguimiento de un *link* y la búsqueda son similares, cada uno es una forma de acceder a nodos destinos de posible interés. El seguimiento de un *link* usualmente alcanza un único nodo, en cambio la búsqueda puede alcanzar muchos, por lo tanto una palabra clave es una clase de *linkeo* implícito.

1.6.2.1 PROPIEDADES DE LOS LINKS

Los *links* pueden ser usados para:

- * Asociar una referencia al documento.
- * Crear un comentario o anotación al texto que se está editando.



- * Proveer información organizada (por ej. establecer relaciones entre dos partes de texto).
- * Conectar dos partes sucesivas de texto o una parte de texto a todos sus sucesores inmediatos.
- * Conectar entradas en una tabla.

1.6.2.2 TIPOS DE LINKS

En HDM (Ref. 4) se hace diferencia entre tres tipos de *links*: Estructurales, de Aplicación y de Perspectiva:

- * Los LINKS ESTRUCTURALES conectan componentes pertenecientes a la misma jerarquía. Las entidades prevén un contexto de navegación, por lo tanto los *links* estructurales permiten al lector examinar diferentes partes de la misma cosa. Por ejemplo en el diccionario de inglés de Oxford, si una entidad es "*Entry*", cada significado sería una componente y los *links* estructurales conectarán todos los significados de la misma entrada, a través de la navegación entre ellos. Por lo tanto los *links* estructurales capturan la semántica estándar.
- * En cambio los LINKS de APLICACION determinan la semántica de las aplicaciones en el dominio. Es decir que ellos representan relaciones entre entidades que el autor considera significativas en el sentido que esta relación determina alguna asociación entre conceptos útiles al usuario del hyperdocumento. Colocando un *link* de aplicación, el autor hará natural al usuario el acceso a alguna información relacionada con el documento que esta siendo leído en ese momento - es suficiente con atravesar ese *link*.
- * Dado que los componentes determinan una abstracción de varias

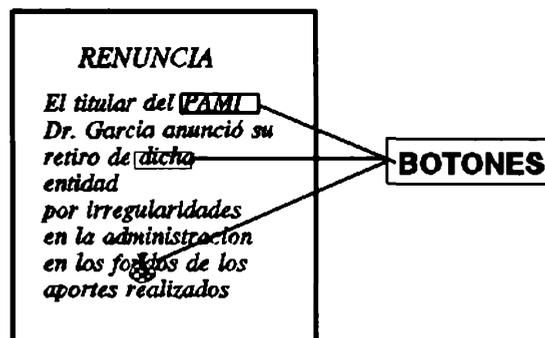


perspectivas de un mismo nodo, los LINKS de PERSPECTIVAS (o de contextos) permiten al lector moverse entre diferentes perspectivas de una misma componente.

1.6.3 PUNTOS Y BOTONES

Un botón es un carácter simple, *token* o ícono que señala un *link* en el documento. Usualmente son identificados o por el nombre del nodo destino, el *link* o un *string* arbitrario.

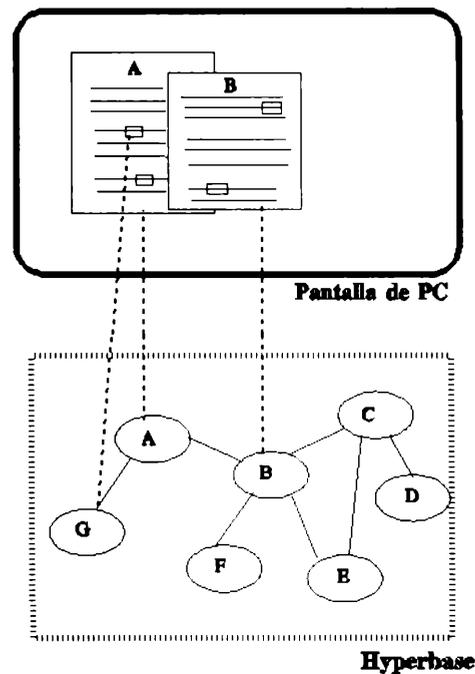
Los botones pueden disparar la muestra de información adicional, atravesar un *link* o activar un programa. Pueden ser representados por texto, íconos o ser invisibles.





1.7 HYPERBASE

Un cuidadoso análisis de las aplicaciones existentes y planeadas, han mostrado que en general son aplicaciones hechas con dos componentes: la Hyperbase y sus mecanismos de acceso. La Hyperbase es el espacio de información estructurada, siendo la esencia de la aplicación.



Los mecanismos de acceso proveen la forma de recuperar información de la Hyperbase y comenzar la navegación en ella. En HDM se han clasificado cuatro tipos diferentes de acceso:

1 - Mecanismos de *path*: provee un conjunto de líneas dirigidas permitiendo el acceso en una secuencia preestablecida a unidades de la Hyperbase. El paradigma de la navegación básica es la secuencia ordenada (típicamente dada con botones próximo, anterior, primero y último).

2 - Indexados: Permite al lector seleccionar los tópicos que le interesan a través de una serie de indexados, generalmente organizados jerárquicamente. El resultado de la selección será una entidad simple o un *path* lineal limitado.



3 - Querys: Permiten especificar al lector su interés a través de formulaciones precisas usando palabras claves o atributos previamente asociados con entidades (o componentes), similares a los mecanismos de *querys* para las bases de datos. El resultado de ejecutar tales *querys* es típicamente un viaje guiado lineal el cual ha sido armado dinámicamente durante la evaluación de la consulta.

4 - Asociativos: Es similar a los *querys*, la diferencia está en que la información deseada es especificada usando patrones en el contenido de las entidades o componentes, o en las propiedades estructurales de la Hyperbase. La ventaja es la gran flexibilidad; la desventaja es una menor precisión en la identificación de la información deseada.

1.8 NAVEGACIÓN

La base de datos es un conjunto de nodos conectados a través de los cuales los usuarios viajan rápidamente. Los usuarios navegan de un nodo a otro colocando el cursor del *mouse* en el botón que *linkea* a otro nodo.

Dependiendo del producto en particular que se usa, los comandos y características de navegación pueden variar. Algunos **Hypertextos** usan un *path* para ayudar a ubicarse en un lugar de la red. Los *paths* son rutas por defecto de la base de datos; ellas guían o dan una lista ordenada de nodos. Cuando se sigue un *path* el autor lo está guiando al próximo nodo lógico.

Otra posibilidad en la navegación es considerar visiones filtradas. Una visión filtrada es otra herramienta interesante de **Hypertexto**. Filtrando el más bajo nivel del contenido de un nodo, se puede buscar rápidamente la información que se necesita.

Para usar un sistema de **Hypertexto** correctamente, un usuario debe darse cuenta que tiene que adaptarse al uso del mismo. Los productos de **Hypertexto** no piensan por él. Los sistemas de **Hypertexto** no tienen inteligencia artificial.



Los sistemas de **Hypertexto** podrían ayudar al usuario a clasificar, y llevarlo a través de su información, pero el usuario tiene el control. Este determina qué incluir en la base de datos y qué tipo de *links* crear, y cómo organizar los nodos.

Si se usa un sistema de documentación de **Hypertexto on-line**, debe decidir cuáles nodos acceder y cuáles *links* seguir. Si se siguen *paths* oscuros, el usuario puede encontrar pesado localizar información. Si sus poderes de asociación son débiles y se crean *links* no significativos se puede obtener una base de datos inútil.

1.9 AREAS DE APLICACION

Hay cuatro áreas de aplicación para las cuales los Hypertextos han sido desarrollados:

* Sistemas de Macro literatura: el estudio de tecnologías para soportar grandes librerías, en las cuales son armados varios documentos.

En las visiones tempranas de **Hypertextos** se enfocó a la integración de grandes volúmenes de información para ser leídos por medio de una interfase simple y consistente. La red total del sistema es dinámica ya que puede ser enriquecida por lectores sin desfasar los documentos originales. De esta manera se disminuye la diferencia entre autores y lectores.

* Herramientas para la exploración de problemas: herramientas para soportar el pensamiento no estructurado de un problema donde muchas ideas desconectadas vienen a la mente, por ejemplo durante la resolución de un problema, la programación y el diseño.

* Sistemas de *browsing*: sistemas similares a los sistemas de macro literatura pero más pequeños en escala (para la enseñanza, para referencia de la información pública, donde la facilidad de uso es crucial).



* Tecnologías de **Hypertextos** en general: diseño de sistemas de propósitos generales para permitir experimentación con un rango de aplicaciones de **Hypertextos**.

1.10 VENTAJAS Y DESVENTAJAS DEL USO DEL HYPERTEXTO

1.10.1 DESVENTAJAS

Los dos problemas más peligrosos y que de hecho limitan la utilidad de los **Hypertextos**, son: la desorientación y el conocimiento adicional (*cognitive overhead*).

Con el poder de organizar la información compleja surge el problema de saber en qué lugar de la red del **Hypertexto** se encuentra el usuario y cómo llegar a conseguir lo deseado de otro lugar donde se sabe(o piensa) que existe, al que se denomina desorientación. Por supuesto que este problema también puede ocurrir en documentos lineales, pero en este caso el lector tiene solamente dos opciones: realizar la búsqueda del texto deseado hacia adelante o hacia atrás en el mismo.

Los **Hypertextos** ofrecen mayor nivel de detalle, más dimensiones en las cuales moverse y por lo tanto, un mayor potencial para llevar al usuario a la desorientación.

Hay dos soluciones para este inconveniente: *browsers* gráficos y mecanismos de búsqueda/consulta. Los *browsers* cuentan con un alto desarrollo visual. Los nodos y los *links* se ubican en espacios de dos o tres dimensiones.

El otro problema fundamental en el uso de **Hypertextos** es la dificultad para llegar a acostumbrarse al conocimiento adicional que se requiere para crear, editar y nombrar *links*.



Hypertexto es una tecnología inmadura con muchos problemas aún por resolver. Sin embargo la parte más difícil en la creación de un **Hypertexto** no es construir la interfase usuaria sino crear los modelos de datos subyacentes que son mantenidos completos.

Desde que los sistemas de **Hypertexto** necesitan ser mantenidos, los diseñadores del sistema deberían ver los *linkeos* incontrolables, los cuales podrían traer problemas de mantenimiento. Así como los grandes programas de *software* con muchos parches pueden llegar a códigos "spaghetti", los sistemas de **Hypertexto** pueden caer en una ciénaga vacía o sin sentido, conexiones y referencias oscuras. Los sistemas de **Hypertexto** deben permitir, por lo tanto, editar y borrar nodos fácilmente. Otro problema para ciertos usuarios, se presenta cuando algunos sistemas de **Hypertexto** le dan el control, por lo cual, necesitan una guía. Pueden por ejemplo perderse siguiendo *links* oscuros antes de llegar al área que están tratando de investigar.

Cuando el usuario está leyendo un texto impreso, un buen autor guiará a través de una red interrelacionada de puntos relevantes. Con los **Hypertextos**, el usuario se guía a sí mismo y hace sus propias asociaciones.

Estructuras virtuales (nodos, o composiciones) podrían ser útiles para resolver esta situación. Ellos cambiarían dinámicamente cuando se agrega o borra nodos dependiendo de sus descripciones. Las estructuras virtuales son similares a las visiones de base de datos relacionales.

1.10.2 VENTAJAS

En los métodos tradicionales de lectura de textos se presentan al lector los siguientes problemas: un lector no puede ver fácilmente dónde un libro específico o un artículo está referenciado en un documento, ni puede el autor de un artículo saber quién lo ha referenciado; cuando el lector navega por las referencias no debe perder de vista cuales documentos ha visitado y que ha



hecho con ellos; el lector debe apuntar anotaciones en los márgenes o en documentos separados; finalmente seguir una pista referencial a través de documentos requiere un esfuerzo físico y pérdida de tiempo, aún más, si el lector está trabajando en una librería con gran stock. La importancia de los hipertextos es que estas referencias están soportadas de una manera fácil de acceder y controlarlas.

También ofrecen nuevas posibilidades para acceder a grandes -o complejas- fuentes de información. Un documento lineal solo puede ser fácil de leer en el orden lineal en el cual el texto aparece en el libro; la ventaja esencial de un texto no lineal es la habilidad de organizar el texto de diferentes maneras dependiendo de diferentes puntos de vista.

VENTAJAS OPERACIONALES DEL HIPERTEXTO

- * Facilidades de referencias intertextuales: mecanismo de soporte vía *links*, significa que todas las referencias son igualmente fáciles de seguir hacia adelante o hacia atrás.
- * Facilidades para crear nuevas referencias: los usuarios pueden aumentar su propia red o hacer comentarios de sus documentos sin cambiar los documentos referenciados.
- * Documentos reutilizables: segmentos de texto pueden ser tratados juntos de varias maneras permitiendo al mismo documento servir para múltiples funciones.
- * Modularización de información: ya que el mismo texto puede ser referenciado desde varios lugares, las ideas pueden ser expresadas con menos duplicaciones.



- * Consistencia de información: las referencias son colocadas en el texto y si el texto es movido a otro documento, se puede *linkear* a la misma información.(Ref. 1)



B. MULTIMEDIA

"EL MEDIO ES EL MENSAJE" (Marshall McLuhan)

El mundo de la impresión fue revolucionario porque le dio vida eterna a la información, hizo posible su traslado en el espacio así como en el tiempo.

Las historias orales requieren que los que escuchan estén en el rango de la voz del que habla: el mundo impreso no tiene tales restricciones.

La tinta en el papel es un ejemplo del acceso directo al medio, dice Walter Bender. No hay nada entre usted y el papel excepto quizás los lentes. Aunque se traslade a través de la información, ella permanece inalterable y se puede volver al origen o a cualquier otro punto del texto.

La radiodifusión fue la siguiente revolución que golpeó al mundo de la información. La radio trajo eventos que fueron angustiantes y alentadores. Al mismo tiempo, sin embargo, lo saca de la información. La radio fue el primer ejemplo real de un acceso indirecto a la información, dice Bender. Había una caja o un procesador de señales entre el usuario y la información, pero los controles que se tienen en una radio afecta esencialmente a las características de la señal, se puede cambiar el volumen o la estación, pero no se puede hacer nada con el contenido actual de la información.

La computación tiene un rol en la última revolución de la información. Lo que es revolucionario con las computadoras, dice Bender, es que permiten una rica interacción con el contenido del material. Bender compara el correo electrónico con un fax impreso. En esencia hay varias formas de conseguir información, lo que es significativo es la forma en la cual esta se accede y lo que se puede hacer con ella.

Un fax llega y se lo puede leer y nada más, un documento de correo electrónico llega en una forma que permite leerlo directamente como un fax, pero también le



permite comenzar a hacer ciertos tipos de manipulaciones. Tales manipulaciones incluyen edición y chequeo de ortografía (diccionario): donde se altera actualmente el contenido del mensaje.

La idea de juntar impresión, radiofonía (sonido), video y computación ha capturado la imaginación de industrias gigantescas como Microsoft, Apple e IBM.

Este fenómeno se conoce hoy en día con el nombre de **Multimedia**.

"**Gracias a las computadoras, no hay más ciencia ficción**" Ben Nelson, Gobierno del estado de Nebraska.

1.11 INTRODUCCION DE MULTIMEDIA

La definición de **Multimedia** parece, en su origen, simple; **Multimedia** es la utilización conjunta de todas las formas posibles en la presentación de datos.

Es **Multimedia** una propuesta tecnológica o una nueva forma de comunicación?

Ambas cosas. El fenómeno **Multimedia** puede estudiarse tanto desde el punto de vista de evolución tecnológica como un nuevo elemento llamado a revolucionar la comunicación entre individuos. Sin embargo, tan sólo se puede considerar propiamente como tecnología **Multimedia**, aquellas partes que se refieren a la integración de los elementos. Cualquier elemento estudiado individualmente no debería ser considerado como **Multimedia**.

Sin embargo, conviene no olvidar que los nuevos desarrollos en video digital, por ejemplo, no pertenecen propiamente a **Multimedia**, sino al tratamiento digital de la imagen. Lo mismo ocurre con el sonido, la animación o cualquier otro medio. Todos y cada uno de ellos se desarrollan en ambientes distinguibles, con métodos y especialistas definidos y diferenciados. Sólo en el momento de la integración conjunta



en un único mensaje adquieren categoría **Multimedia**.

El hecho fundamental de **Multimedia** no radica tanto en la tecnología fragmentada que da acceso a este nuevo medio, como en las nuevas expectativas y posibilidades que abre en el campo de la comunicación. En una palabra su aplicación práctica y cotidiana.

Definiríamos correctamente **Multimedia** como la aplicación conjunta y en un solo mensaje de texto, gráficos, sonido e imágenes, tanto fijas como animadas?

Quizá podemos considerarla como una definición acertada, pero no completa, pues depende del factor tiempo. Podríamos decir que esta es ahora una definición ajustada, pero quizá no lo sea en un futuro a corto plazo. Para demostrar este hecho basta recordar que las formas de presentación de datos o informaciones conocidas y disponibles en la actualidad son las anteriormente mencionadas. Nadie puede atreverse a excluir la definición de nuevas formas de presentación de los datos que, forzosamente, deberán ser incluidos en **Multimedia** tan pronto como sea posible y sin las que **Multimedia** será un concepto incompleto.

Para algunos autores una presentación **Multimedia** debería ser completa en todos y en cada uno de sus elementos. Ello no quiere decir que todos los mensajes deban incluir todas las formas de presentación. Pero si una aplicación no dispone en ningún caso de alguna de ellas, quizá no debería ser denominada **Multimedia**.

La explicación es sencilla. Si aceptamos por convenio que una aplicación que no precise sonido (por ej: una visita guiada a través de una exposición de pintura) puede ser catalogada de **Multimedia**, deberemos aceptar también que cualquier combinación que excluya uno o varios elementos debería ser considerada como **Multimedia**. **Multimedia puede ser definida sólo de dos maneras: Como la integración de todas las formas de presentación de datos o como la integración de más de una forma de presentación de datos.**



Llevando esta consideración al extremo habríamos de elevar a la categoría de aplicaciones **Multimedia** a hojas de cálculo convencionales, que incorporan datos y gráficos, procesadores de texto que ofrecen texto e imágenes, o los arcaicos y tradicionales juegos "matamarcianos", compuestos por gráficos y rudimentarios "beeps". (Ref. 10)

1.12 TRES DEFINICIONES PARA MULTIMEDIA

Pueden existir tantas definiciones como formas de entender **Multimedia**, todas ellas ciertas y ajustadas. Y, si hasta el momento no se ha llegado a una definición única y unívoca es porque **Multimedia** es un fenómeno múltiple y evolutivo. Y no se puede pretender una única definición para este tipo de conceptos múltiples.

En primer lugar existe la tecnología **Multimedia**, que ya hemos definido como **todas las posibles formas de presentación de la información de forma simultánea y en un único contexto**; la presentación de un determinado mensaje bajo todas sus formas de presentación posible, ya sea en serie o simultáneamente. Esto es tecnología multimedia y se produce gracias al trabajo conjunto de una serie de elementos. Pero, del mismo modo que un violín es parte fundamental de un cuarteto, nunca puede pretender ser cuarteto. O, más claramente, no hay placa de sonido ni tarjeta de captura de imágenes que, por sí sola, merezca la denominación de **Multimedia**.

Como segunda definición, **Multimedia** es una nueva herramienta de **comunicación**, el fenómeno **Multimedia** puede representar una auténtica revolución tanto en la presentación de la información como en la estructura misma de la comunicación, comparable con la sufrida con la aparición de la imprenta.

Por último, **Multimedia** es un proceso en constante evolución basado en el **tratamiento conjunto e integral de la información desde todas sus formas de presentación**. Según esta visión, multimedia como desarrollo definitivo y cerrado nunca podrá ser definida, porque se trata de un proceso evolutivo.



Aunque las actuales propuestas pueden ser consideradas **Multimedia**, están en sus primeros estadios.(Ref.11)

1.13 MULTIMEDIA NO INTERACTIVO?

Es posible imaginar un sistema de **Multimedia** no interactivo? o, quizá mejor, es la interactividad un elemento determinante de la calificación de un determinado sistema como **Multimedia**? Esta es otra de las cuestiones en la que no todos los especialistas se ponen de acuerdo.

Una vez realizada la triple definición del concepto **Multimedia**, consideramos esta cuestión dentro de la segunda acepción, puesto que se refiere tanto a un supuesto técnico como comunicativo. La esencia de la tecnología **Multimedia** es la presentación de la información en forma conjunta y en todas sus posibles modalidades, mientras que la revolución representada por la PC se basa en su capacidad de almacenar datos, procesarlos y ofrecerlos en la forma que al usuario le resulte más útil en cada momento. Ello significa que prefijar una estructura rígida en la recuperación de datos e informaciones va en contra del tiempo y la paciencia del individuo, al no poder prescindir de todo aquello que no le sea de su interés.

Para justificar la explicación anterior, le propondríamos la lectura de "La Guerra y La Paz" con la única salvedad de que cada vez que abra el libro inicie la lectura desde la primera página hasta el punto en que abandonó la lectura en la anterior sesión. El ejemplo es lo suficientemente desalentador como para no querer iniciar la lectura.

El concepto de **Multimedia interactiva** representa una plataforma de acceso a la información deseada desde cualquier punto y en cualquiera de sus formas. Ello significa que el usuario debe poder optar por toda o parte de esta información, y el sistema se la debe poder presentar de la forma más rápida posible, evitando datos redundantes o no deseados. Bajo esta acepción, un sistema **Multimedia** será más útil



en tanto permita al usuario mayor libertad de acceso desde cualquier punto del sistema a aquella información que desea obtener prescindiendo del resto.

Es difícil imaginar una herramienta de presentación **Multimedia** que no admita interacción. Quizás algún tipo de sistema educativo necesite de alguna estructura inviolable, que obligatoriamente ha de dar determinada información en un determinado orden. Incluso en este caso el concepto de interactividad no es viable. **Multimedia** debe ser, ineludiblemente interactivo. Cualquier otra forma de desarrollo no será más que una réplica de otros sistemas de comunicación cerrados y bloqueados, desaprovechando todo el potencial de ajustar la salida del sistema a la entrada general. Y si se da el supuesto que la eficiencia del sistema exija la no interactividad, existen otras vías de comunicación audiovisual no interactivas eficientes.

Una de las definiciones más completas y que mejor se ajustan a todo lo descrito hasta este punto, fue enunciado por la Asociación Española de Empresas de **Multimedia** (ASEDEM), que definió **Multimedia** como el conjunto de técnicas y materiales necesarios para generar un "libro" o "documento" Multimedia, definiendo posteriormente este concepto como "aquel documento basado en un ordenador donde se combinan informaciones en forma interactiva"; es decir, donde el lector decide qué parte y en qué secuencia desea recibir la información, convirtiéndose en un elemento activo del proceso de lectura y/o aprendizaje.

1.14 BLOQUES DE CONSTRUCCION EN MULTIMEDIA

Es difícil conseguir **Multimedia** ya que no es un producto simple o una plataforma simple, sino que es una colección de tecnologías. Los bloques básicos para construir una multimedia son: texto, gráfico, sonido, animación y video.



Texto:

Mucha gente ignora que la impresión es parte de una mezcla de **Multimedia**. Las características de tipos de letras junto con las características de impresión y video parte de **Multimedia** porque los tipos de letras también se usan para presentación en los video.

Gráficos

Su importancia se debe a que las imágenes *scaneadas* y dibujadas son otra característica importante de **Multimedia**.

Sonido

Lo que está siendo aceptado actualmente en **Multimedia** es voz y sonido. Agregarle voz a las anotaciones, a las hojas de cálculo o a memos es como darle un toque de emoción que no se puede lograr en una carta o una ilustración cuando está siendo utilizada.

Video

Este es el componente más difícil de implementar.

Animación

Se ve comúnmente en entretenimientos y comerciales.



1.15 IDENTIFICACION DE MERCADOS DONDE SE APLICA MULTIMEDIA

Educación y capacitación

Algunos analistas ven a éstos como mercados distintos pero se considera que es un único mercado porque tienen objetivos similares, el mercado de la educación y capacitación es corrientemente, uno de los principales mercados de **Multimedia**. La **Multimedia** es natural para la capacitación porque las posibilidades de voz y movimiento en la pantalla hacen una interfase más similar a la humana (*human like*). La información esta más y mejor presentada y la comprensión es mayor.

Presentaciones

Esta es otra área importante. De Salum, una empresa que vende pisos, usa **Multimedias** interactivas para ayudar a sus clientes en la elección de los pisos de acuerdo a sus necesidades, permitiéndole interactivamente seleccionar un piso y ubicarlo dentro de habitaciones con una apariencia similar a sus ambientes.

Entretenimientos

Los entretenimientos fueron una de las primeras industrias que mostraron que la **Multimedia** era posible y aún es el mercado más importante.

1.16 LA MULTIMEDIA DEL MAÑANA

Según W. Bender la **Multimedia** no va a incrementar la interactividad tanto como la visualización. La forma en que **Multimedia** acrecentará la productividad en



Hypermedia CIP

el ámbito del trabajo, no es dándole a la gente más cosas para cortar y pegar sino dándole más poder en la ayuda para visualizar información. La visualización es la clave de la **Multimedia**.(Ref.12)



C. HYPERMEDIA

1.17 INTRODUCCION DE HYPERMEDIA

Algunos ambientes de Multimedia también tienen capacidades de **Hypermedia** o la habilidad para *linkear items* en una forma no lineal, esto le da a los usuarios de multimedia control sobre la secuencia y el contenido de la información que ellos reciben.

Ej: Los lectores pueden conseguir la información que les interesa y saltar el resto, y si ellos no pueden, hay un tutorial ilustrado que le dice como *scrolear* a través del texto de un artículo, como ir de una pantalla a otra y cómo retornar a la tabla de contenidos. También se usan medios como gráficos, animaciones y sonido que realzan el contenido de los artículos; que no pueden ser introducidos en un artículo impreso.

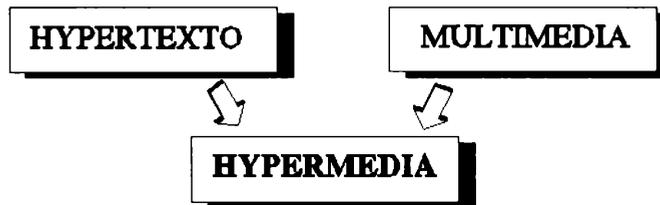
Mientras que las publicaciones convencionales están limitadas a textos o gráficos, los nodos de Hypertexto ofrecen sonido, animación, así como también programas que comienzan a correr cuando los nodos en los cuales ellos están cargados son seleccionados.

La **Hypermedia** es una generalización del concepto de Hypertextos. Los sistemas de Hypertexto e **Hypermedia** se pueden presentar las siguientes características:

- * La información es ubicada en pequeñas unidades llamadas *frames* o nodos. Las unidades pueden contener información textual. En los sistemas de **Hypermedia** las unidades pueden contener también otra forma de información como gráficos, imágenes que mapean bits, sonido y animación.
- * Las unidades de información son mostradas una por ventana.



- * Las unidades de información son interconectadas por *links*, los usuarios navegan en la base de datos de **Hypermedia** para ir de una unidad a otra.
- * Creando, editando y *linkeando* unidades, los usuarios construyen estructuras de información para varios propósitos.



Los sistemas de **Hypermedia** se diferencian por la forma de considerar las siguientes tres dimensiones:

- * **Alcance:** los sistemas de **Hypermedia** pueden construirse:
 - a) como un mecanismo para cargar y manipular grandes volúmenes de información.
 - b) como un espacio de información común para grupos de programadores en grandes proyectos de *soft*.
 - c) como una herramienta para grupos pequeños de trabajos que crean y procesan ideas.

Aunque estos tres propósitos comparten la noción de que la información debe ser organizada en redes de nodos y *links*, difieren en la escala, por ejemplo, en la dimensión necesaria para la información.



- * **Browsing vs. authoring:** en sistemas diseñados para *browsing*, la red de **Hypermedia** es cuidadosamente creada para proveer un espacio de información a ser explorada por usuarios casuales, las herramientas de presentación de información y exploración son cuidadosamente desarrolladas y no así las herramientas de creación y modificación de la red.

En diseños para autoría, la red de **Hypermedia** sirve como una estructura de información para usuarios que la crean y la modifican continuamente, y las herramientas de creación y modificación son cuidadosamente desarrolladas (ej.: sistemas de **Hypermedia** para procesar ideas, para desarrollo de *soft*).

- * **Dominio:** muchos sistemas de **Hypermedia** han sido desarrollados para soportar una tarea específica, otros proveen facilidades generales para ser usados en una variedad de aplicaciones, sin embargo son diseñados con un dominio de tareas en mente.

1.18 CARACTERISTICAS PARA LA NUEVA GENERACION DE HYPERMEDIA

- 1- La habilidad para navegar a través de la red siguiendo los *links* de un nodo a otro, es una característica definida de **Hypermedia**, es esta habilidad la que hace que Hypermedia sea una poderosa herramienta para manejar información estructurada. Pero el acceso por navegación no es suficiente, una forma efectiva de recuperar la información cargada en la red requiere un mecanismo basado en consultas para completar la navegación. Hay una variedad de aplicaciones para las cuales el acceso por navegación es problemático, estas aplicaciones se caracterizan por estar estructuradas en redes grandes, no conocidas y heterogéneas, el usuario puede llegar a perderse mientras busca una información determinada. Varias veces los usuarios pueden describir exactamente qué información están buscando, pero no la pueden encontrar fácilmente en la red.

Una solución fundamental para mejorar la navegación es un mecanismo de



acceso basado en consultas, de forma tal, el usuario puede formular una consulta describiendo la información deseada y confiar en que el sistema localizará la información en la red.

Hay dos clases de mecanismos búsqueda/consulta:

a) Búsqueda por contenido: todos los nodos y *links* en una red son considerados entidades independientes y son examinados individualmente para ver si coinciden con una consulta dada. Es una recuperación de información standard aplicada a la base de información de **Hypermedia**. Esta ignora la estructura de la red de **Hypermedia**.

b) Búsqueda por estructura: examina específicamente la estructura de **Hypermedia** para determinar las subredes que coincidan con el patrón dado. El desarrollo de este mecanismo involucra dos subtaréas interrelacionadas, la primera es diseñar un mecanismo de consulta preparado para describir estructuras de redes de **Hypermedia**. Una aproximación a esto último, sería desarrollar expresiones regulares que encierren patrones arbitrarios de la red. Este lenguaje patrón necesitará incluir operadores de expresiones regulares standard, tales como el asociativo, concatenación, negación y en adición el lenguaje necesitará alguna forma de expresar bifurcaciones.

Una importante consideración en el diseño de este lenguaje es la necesidad de una interfase simple que sea accesible al usuario de hypermedia, a quien probablemente no le es fácil entender el lenguaje de patrones.

La segunda subtaréa principal es la implementación del mecanismo de búsqueda para satisfacer las consultas expresadas en el nuevo lenguaje.

Una de las característica críticas para la próxima generación de **Hypermedia** es definir *pattern matching* para que puedan ser fáciles de implementar.

Las facilidades de búsqueda y consulta van mas allá de la tarea de localizar



información, en particular puede ser usado como un mecanismo de filtrado en la interfase de **Hypermedia**. En este caso los usuarios especificarán una consulta para describir la información que les interesa, la interfase entonces mostraría solamente aquellos aspectos de la red que coincidan con esta consulta, filtrando de esta manera la información irrelevante.

- 2- Composición aumentando el modelo de nodos básicos y de *links*: el mecanismo existente es ineficiente, se necesita un mecanismo de composición, es decir una forma de representar y de proceder con un conjunto de nodos como entidades únicas separadas de sus componentes.

El usuario piensa al nodo raíz como una representación de composición, pero el sistema no entiende de composiciones, ya que los nodos raíz son usados para juntar otros nodos dentro de un grupo pero son limitados como mecanismo de composición.

En los sistemas existentes, se falla al considerar las diferencias entre las relaciones de referencia y las de inclusión. Una relación de inclusión implica una relación total o parcial, en la cual las características y operaciones sobre el total afectan a las partes. La relación de referencia implica una relación más libre, en la cual las entidades participantes aluden a cada una de las otras pero permanecen independientes.

El archivo en los sistemas actuales se hace a través de relaciones de dependencia, pero la interfase y la documentación alientan al usuario a archivar mediante una relación de inclusión.

La solución a este problema es agregar composición al modelo básico de **Hypermedia**. La inclusión debería ser implementada en todos los sistemas de **Hypermedia**.



- 3- Estructuras virtuales para proceder ante cambios de información: los sistemas de **Hypermedia** tienen dificultades con los cambios rápidos de información, esta dificultad proviene por la naturaleza estática y fragmentaria del modelo de datos de **Hypermedia**. Por definición **Hypermedia** codifica información en colecciones de nodos independientes interconectados en una red estática; esta red no cambia a menos que sea explícitamente editada por el usuario, en particular la red no puede configurarse a si misma en respuesta a cambios en la información que contiene, esta falta de mecanismos dinámicos limita la utilidad de **Hypermedia**.

El modelo de **Hypermedia** necesita ser extendido aplicando la noción de estructuras virtuales o determinadas dinámicamente.

En el modelo corriente, los nodos y los *links* son definidos especificando la identidad exacta de sus componente, en contraste las estructuras virtuales son definidas especificando la descripción de sus componentes. Las componentes exactas de una estructura virtual, están determinadas por un procedimiento de búsqueda siempre que la estructura es accedida o instanciada.

La noción de estructuras virtuales para **Hypermedia** es una adaptación directa del concepto de visiones en el mundo de los sistemas de bases de datos relacionales. En las bases de datos relacionales, una visión es una tabla construida en tiempo de instanciación aplicando una definición visual a los datos cargados explícitamente en las tablas de la base.

El objetivo es hacer tablas basadas en visiones idénticas a las tablas de la base desde una perspectiva usuaria. Cualquier operación posible en una entidad de la base de **Hypermedia**, sería aplicable a estructuras virtuales.

La noción de estructuras virtuales en **Hypermedia** sería posible desarrollar



sólo en un sistema que soporte un mecanismo sustancial de búsqueda/consulta en la red de **Hypermedia**.

La definición de las componentes en estructuras virtuales son en realidad consultas. Instanciar una estructura virtual involucra que se satisfagan estas consultas y que se construya una entidad dinámica desde los resultados, aunque no es estricto; tendría sentido que el lenguaje de consulta usado para descripciones de estructuras virtuales sea el mismo que el usado para búsqueda e interfaces filtradas.

Las estructuras virtuales son un poderoso mecanismo cuando se combina con la noción de composición. Una composición virtual permite al usuario crear nodos que son dinámicamente construidos en tiempo de acceso desde otros nodos, o composiciones que son cargadas en la red. Tales composiciones son verdaderas entidades de **Hypermedia** y no simplemente una muestra de resultados desde una consulta, por lo tanto el usuario puede agregar , propiedades o descripciones estáticas adicionales. Los *browsers* por ejemplo, pueden ser implementados como construcciones de composiciones virtuales desde los resultados de una estructura de consulta.

Los *links* virtuales en **Hypermedia** conectan el nodo mostrado a nodos que el usuario ha visitado recientemente, esto permite a los usuarios moverse rápidamente desde donde ellos venían.

- 4- Computación en redes de **Hypermedia**: Los sistemas de **Hypermedia** son generalmente programas de carga y de recuperación pasiva, ellos proveen herramientas a los usuarios para definir, cargar y manipular una red de **Hypermedia**. Para lograr este objetivo, ellos hacen algún procesamiento de la red y de la información que contiene. Los sistemas de **Hypermedia** sin embargo no activan directamente la creación o modificación de la red o la información contenida en ella.

Uno podría diseñar un sistema de **Hypermedia** que automáticamente procese



la información cargada en la red, en este caso su sistema de **Hypermedia** podría funcionar más como un sistema de Inteligencia Artificial basado en conocimiento que cargue y procese activamente la información.

La elección entre un sistema de **Hypermedia** activo o pasivo está determinada por las aplicaciones y performance necesarias en las mismas.(Ref.5)

1.19 INCREMENTANDO EL PODER DE BUSQUEDA CON CONSULTAS RELACIONALES

La búsqueda en un sistema de **Hypermedia** es comúnmente hecha vía un mecanismo textual, en el cual el contenido de un elemento de información es examinado para algún *string* destino. En adición la mayoría de los mecanismos de *string* incluyen algunos operadores lógicos para componer algunas consultas largas desde consultas simples.

La búsqueda de *string* está limitada ya que solamente puede localizar elementos de acuerdo a su contenido textual. Una mayor facilidad de búsqueda en Hypertexto se logra usando un lenguaje de consultas relacionales y una base de datos relacional derivada automáticamente. El mayor poder de esta aproximación sobre la búsqueda de *string* se debe a que las propiedades físicas y lógicas de un elemento y su representación pueden guiar una búsqueda y también permiten actualizar la información de un Hypertexto por medio de características comunes de los lenguajes de consultas relacionales.

Las **Hypermedias** pueden ser utilizados como un mecanismo de interfase de base de datos para facilitar la dificultad en la formación de consultas. Los sistemas de Hypertextos pueden ser implementados usando bases de datos relacionales en lugar de representación de *linkeos* explícitos. El poder de búsqueda y actualización del modelo de datos relacional es aplicado al modelo de datos de la red (lo cual es más útil en los dominios de información compuestos por gráficos y textos largos). La



cooperación es lograda por la automática conversión de la información en forma de red relacional y viceversa.

El objetivo final de consultar una **Hypermedia** es producir en forma útil una lista de objetos que coincidan con condiciones dadas, no simplemente construir nuevas relaciones. Una consulta, retorna una lista de información a cerca de los objetos, generalmente ella contiene a los identificadores de los objetos. En qué se usa esta lista? Uno de los usos del resultado de una consulta es tomar los objetos en forma de tabla de base de datos y usarlos en subsecuentes consultas relacionales (SQL); esta forma es la representación más general, y puede ser fácilmente manipulada adecuándose a las necesidades de los autores y lectores de **Hypermedia**.

1.20 LINKEO DE ANCHORS ACTIVOS (INTERMEDIA)

Con el advenimiento de la tecnología de Multimedia, los Hypertextos han evolucionado hacia **Hypermedia** extendiendo su alcance para incluir otros medios que no sean textos, algunos de estos medios son estáticos mientras otros son dinámicos. La información en medios estáticos es estacionaria, mientras que la información en medios dinámicos cambia a través del tiempo. Ejemplos de medios estáticos son los textos, los gráficos y de medios dinámicos son la animación, la grabación de video-discos y recuperación desde base de datos.

Los sistemas de **Hypermedia** permiten a los autores asociar información por *linkeo* de tal manera que los usuarios pueden recuperar información relacionada atravesando *links*. Los *links* son creados entre un par de puntos, los cuales pueden ser documentos enteros o parte de documentos. El *linkeo* en sistemas de **Hypermedia** permite a los usuarios navegar desde un punto fuente a un punto destino, mientras que esto es suficiente para aplicaciones estáticas, necesita ser extendido para aplicaciones orientadas a acción.

En ambientes de **Hypermedia** es posible que coexistan aplicaciones estáticas y orientadas a acción. Desde que el *linkeo*, en ambientes de Multimedia, alienta las



transiciones rápidas desde un documento a otro, los usuarios no están forzados a conocer los detalles de operación que utilizan. Seguir un *link* de un documento de animación no trae el documento sino que activa la animación.

Un *anchors* es un punto final de un *link*. Un *anchors* estático es un punto de inserción, una alocaión coordinada, una colección de objetos o un nodo entero en una aplicación estática. Un *anchors* activo incorpora información dinámica en una aplicación orientada a acción. El *linkeo* de un *anchors* estático resulta en navegación, el *linkeo* de un *anchors* activo también resulta en navegación pero despliega la información dinámica asociada con ese *anchors*.

Muchos sistemas existentes permiten al usuario especificar acciones, pero para hacerlo los usuarios deben escribir *scripts* (programas) y conectarlos al fuente de un *link*.

Las aproximaciones *scripting* ofrecen una rica funcionalidad pero no ayudan al usuario a crear *links* para *anchors* activos. Requieren un cierto nivel de programación. En lugar de ofrecer una aplicación que provea herramientas para que el autor defina el alcance de un *anchors* activo, el mecanismo de *scripting* fuerza al autor a conocer los detalles operacionales de una aplicación para definir el alcance; además los *scripts* son vinculados al fuente del *link*, mientras que la información dinámica esta en el destino. Alterar la información en el destino puede requerir actualizar el *script* en cada fuente del *link*.(Ref.6)

Características de Diseño de Anchors Activos

Cuando los usuarios atraviesan el *link* de un *anchors* estático, Intermedia abre el documento destino y el *anchors* destino es visible y con sus contenidos resaltados, esta funcionalidad es inadecuada para aplicaciones orientadas a acciones. El estudio de aplicaciones comerciales orientadas a acción determinó una clasificación compuesta por tres paradigmas:



- * Aplicaciones que siguen el paradigma de *playback* y dan una visión dinámica de un conjunto de *items* de datos secuenciales.
- * Aplicaciones que siguen el paradigma de *recording* y transmiten información compuesta por ítems de datos desde algún sensor externo o dispositivo de grabación.
- * Aplicaciones siguiendo el paradigma de *query* que presentan la información desde una condición que satisface un conjunto de criterios definidos por el usuario. En este paradigma no es la información en sí misma dinámica sino bien la recuperación de la información.

Cuando los usuarios atraviesan un *link* hacia un *anchors* activo, qué ocurre cuando el documento destino se abre?

Esto depende de la información que el autor quiere comunicar a los usuarios y del tipo de aplicación contenida en el *anchors* destino.

En Intermedia, se han definido dos opciones básicas de lo que podría ocurrir cuando un usuario atraviesa un *link* hacia un *anchors* activo. En la primera opción el documento destino puede abrirse sin ejecutar ninguna acción, como en el caso de las aplicaciones estáticas, entonces sería el usuario el que, interactuando con la aplicación, inicie cualquier acción. La segunda opción es que el documento destino se abra y ejecute una acción usando la alocaión del *link* como punto de comienzo. Decidir si un *link* dispara o no una acción depende del autor del *link*.

El significado de ejecutar una acción depende de la aplicación. El diseñador de la aplicación debe decidir que acción se dispara cuando el *link* se activa.

Para definir el resultado de ejecutar una acción para aplicaciones *query*, distintos tipos de acciones pueden ocurrir, por ejemplo: el que desarrolla debería elegir mostrar la especificación de la consulta en el seguimiento del *link* y entonces automáticamente



ejecutar la consulta, presentando los resultados cuando la consulta se completa. Otra opción es ejecutar la consulta y presentar los resultados sin mostrar la especificación de la misma.(Ref.6)

Cuando los usuarios atraviesan un *link* para un *anchors* activo. Qué opciones tendrían los usuarios después que el documento destino se halle abierto?

Si el *link* para un *anchors* activo no realiza ninguna acción, entonces los usuarios podrían desarrollar todas las acciones disponibles para los *anchors* estáticos. Estas incluyen edición de datos, seguimiento de *links* existentes y creación de nuevos *links*.

Los usuarios podrían controlar la acción en sí misma. A diferencia de seguir y de crear *links*, los usuarios deben tener conocimiento de algunos detalles específicos de la aplicación para interactuar con aplicaciones orientadas a acciones. Desde que Intermedia es un ambiente de edición y *browsing*, los autores se dan cuenta que ocultando herramientas y controles, ayudan a mantener las pantallas sin desorden.

Los autores varias veces optan por ocultar estas herramientas y controles una vez que ellos han terminado de editar los documentos. Aún si el usuario no está familiarizado con una aplicación en particular, ellos estarían habilitados a encontrar las herramientas y los controles.(Ref.6)

Qué modificaciones se necesitarán hacer en el mecanismo de carga para permitir el uso de *anchors* activos?

En Intermedia la carga de información del *link* y *anchors* es separada de la carga del documento en sí mismo, las modificaciones necesarias para soportar *anchors* activos representan una evolución en el ciclo de vida del mecanismo de carga más que un rediseño y reflejan una evolución de la funcionalidad de **Hypermedia**.(Ref.6)

MODELO PROPUESTO



2.1 ESPECIFICACION DEL MODELO

2.1.1 INTRODUCCION

Para implementar el prototipo **Hypermedia CIP**, definimos inicialmente un modelo de diseño para especificar los distintos conceptos que se debían tener en cuenta. Al modelo obtenido lo llamamos **MDH** (Modelo de diseño de Hypermedia).

MDH es un modelo global o independiente es decir que para su definición no se tuvo en cuenta ninguna aplicación determinada.

Para llegar a definir **MDH** analizamos distintas herramientas existentes:

HDM - Modelo para el diseño de Aplicaciones de Hipertexto

KMS - Sistema de Manejo de Conocimientos

INTERMEDIA - Sistema de Edición de Hipertexto

HYPERCARD - Sistema de Hipertexto Programable

LINKWAY - Sistema para creación de Hipertexto

Ninguna metodología de las analizadas satisfacían totalmente las expectativas, por lo cual tomamos de ellas los conceptos que considerábamos útiles para la definición del modelo de diseño **MDH**. Incorporamos además los siguientes conceptos por considerarlos adecuados para nuestro diseño, como una forma de mejorar la visualización y consulta de la información:

- Perspectivas o Contextos
- Contextos Anidados



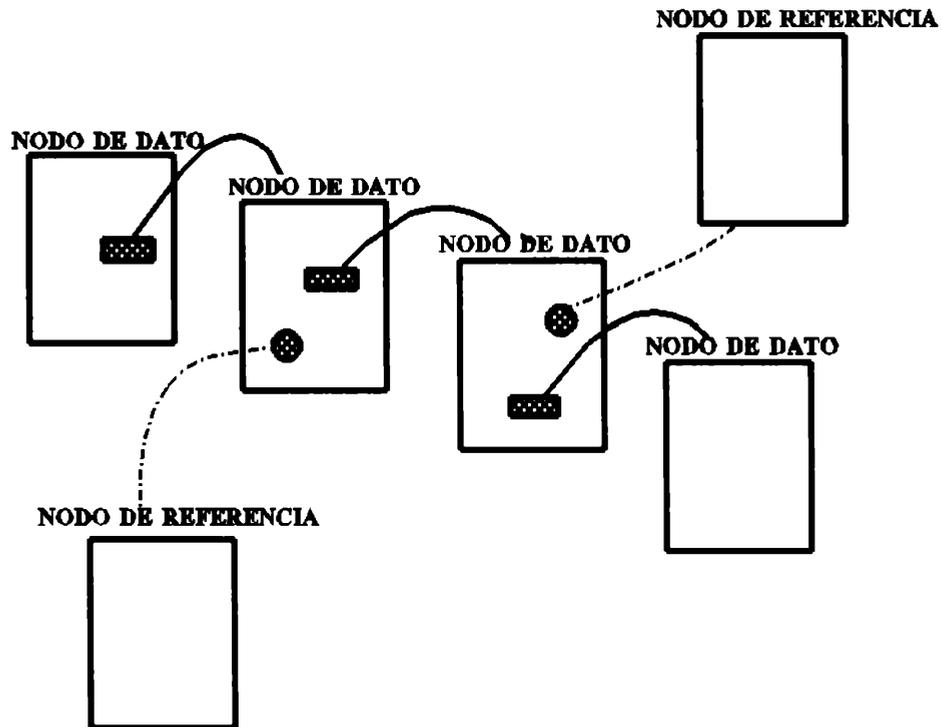
Vamos a comenzar a especificar a **MDH** determinando la forma en que serán considerados en este modelo los bloques esenciales de construcción de un **Hypertexto**: nodos, *links* y botones; ya que usaremos al **Hypertexto** como estructura fundamental sobre la cual basaremos el diseño (por las características del mismo ya explicadas en la parte 1 de este informe), así como también características para el manejo y visualización de la información.

2.1.2 NODOS

Para definir el diseño de los nodos en **MDH** usaremos el concepto de nodo de **KMS**, definido como un espacio de trabajo en la pantalla, en el cual el usuario puede ubicar un texto, gráfico, imagen o animación. Cada uno de éstos puede ser *linkeado* a otro nodo.

En **KMS** se considera útil definir distintos tipos de nodos, particularmente si se considera darles una estructura interna. En **MDH** se definen dos tipos de nodos: nodos de datos y nodos de referencias. Los nodos de datos son los que pertenecen a la red del **Hypertexto**, con posibilidad de *linkeo* a otros nodos del mismo tipo o a nodos del tipo referencia; los nodos terminales, amplían la información contenida en los nodos de datos y no permiten *linkeo* a otros nodos.

Como en **MDH** vamos a considerar distintos medios (por tratarse de una **Hypermedia**), vamos a tomar el concepto acerca del tratamiento de los medios activos de **KMS**, este considera que una forma simple de conseguir medios activos es considerarlos como nuevos tipos de contenido, sin modificar la estructura de los nodos. Esta solución, que satisface los requerimientos de muchas aplicaciones, no requiere ninguna modificación importante en el **KMS**; de la misma forma en **MDH**, lo único que cambia es el tipo del contenido del nodo de datos o referencia y la forma de recuperar cada uno de ellos.



2.1.3 LINKS

La principal ventaja de un modelo de Hypertexto es la organización de la información en una forma no lineal. Esto significa que piezas de información pueden estar relacionadas a través de *links* asociados a ellas.

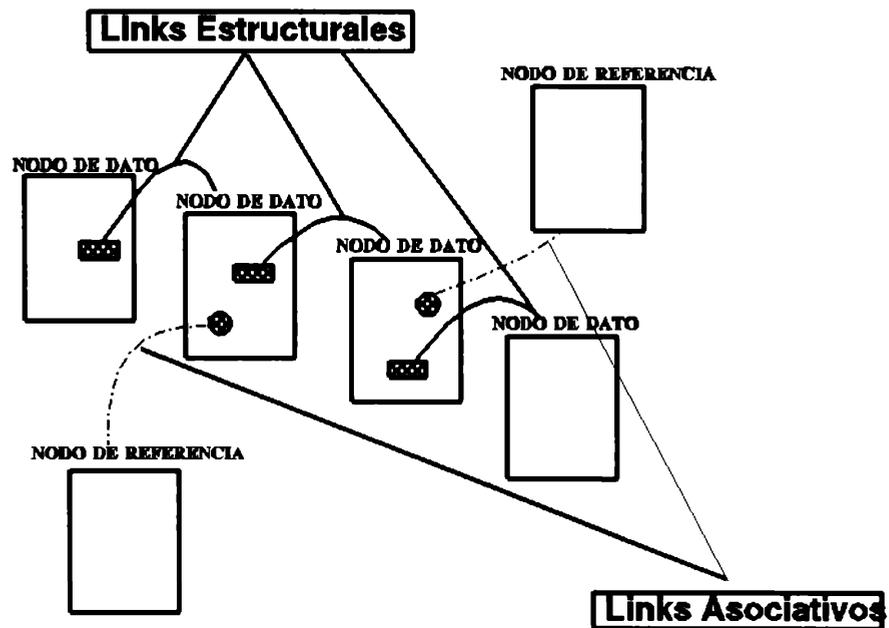
En KMS, el destino para un *link* es un nodo completo, en cambio otros sistemas de Hypermedia como Intermedia usan un punto individual o una región dentro del nodo. En KMS no se necesario esto, probablemente porque un nodo es una unidad lógica suficientemente chica, de tal forma que el nodo completo puede servir como *link* destino; el documento destino puede abrirse sin ejecutar ninguna acción, como en el caso de las aplicaciones estáticas, entonces sería el usuario el que, interactuando con la aplicación, inicie cualquier acción. Adoptamos este diseño para los *links* destino en **MDH**.



La mayoría de los sistemas de Hypermedia proveen distintos tipos de *links*. En algunos sistemas hay un conjunto predefinido de tipos de *links*, en otros pueden ser definidos por el usuario.

El principal propósito de usar tipos de *links* es proveer a los usuarios y a los programas de más información a cerca del destino del mismo.

En MDH se ha adoptado el criterio de tipos de *links* de KMS; donde se definen dos tipos de *links*: "Items de Arbol" e "Items de Anotación". Los "Items de Arbol" son los que permiten *linkear* a un nodo dentro de una jerarquía. Los "Items de Anotación" señalan material periférico, tales como comentarios y referencias cruzadas. Tener estos 2 tipos de *links* distingue entre relaciones estructurales y relaciones puramente asociativas.

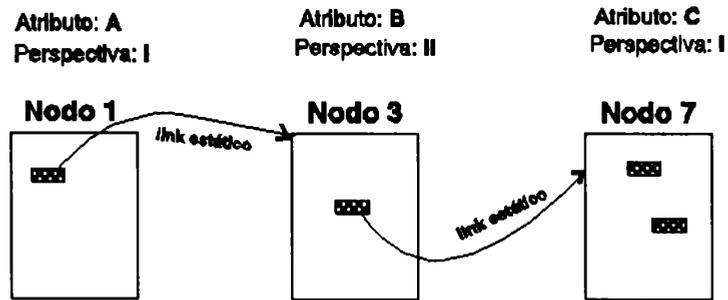




La diferencia en MDH es que, dentro de los *links* estructurales tenemos dos subtipos, los de "navegación" y los de "orden". Los *links* de navegación son clasificados a su vez con respecto a la creación y atributos del *link* en "estáticos" y "virtuales".

* Los *links* de navegación "estáticos" son los creados por el usuario, que *linkean* a otra nota ya incorporada en la jerarquía de nodos. Es decir que ellos representan relaciones entre nodos que el autor considera significativos, en el sentido que esta relación determina alguna asociación entre conceptos útiles al usuario de la Hypermedia. Colocando un *link* de navegación estático, el autor hará natural al usuario el acceso a alguna información que esté relacionada con la información que esta siendo leída en ese momento (es suficiente atravesar ese *link*, para obtener dicha información).

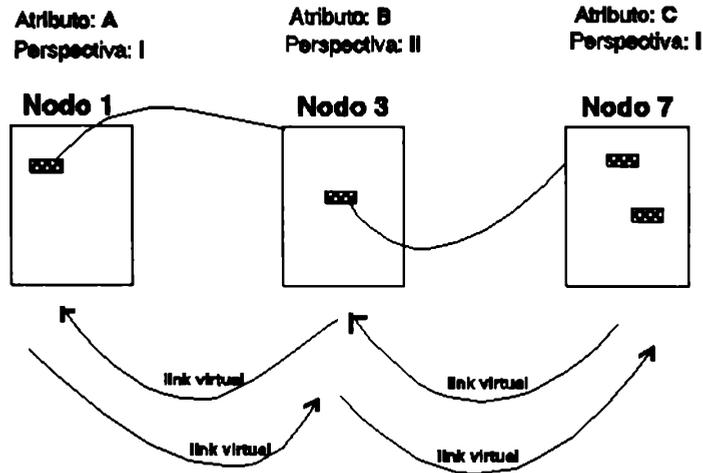
Links Estáticos





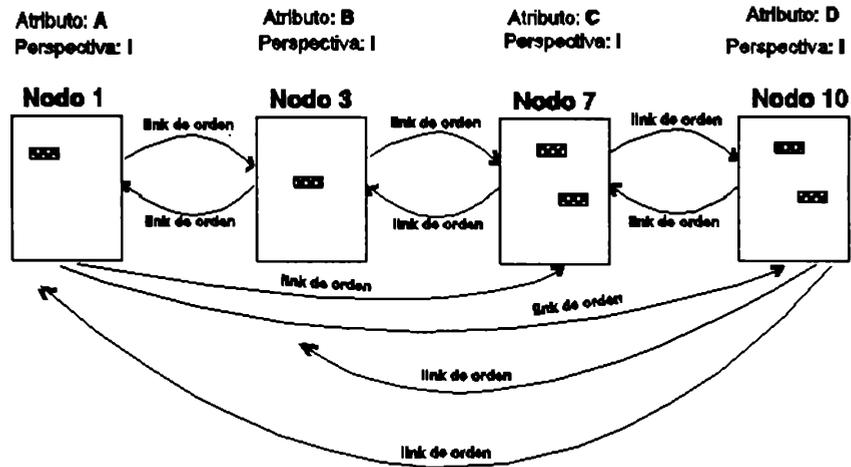
* Los *links* de navegación "virtuales" son los que se crean durante la navegación de la Hypermedia atravesando *links* de navegación estáticos o de orden.

Links Virtuales



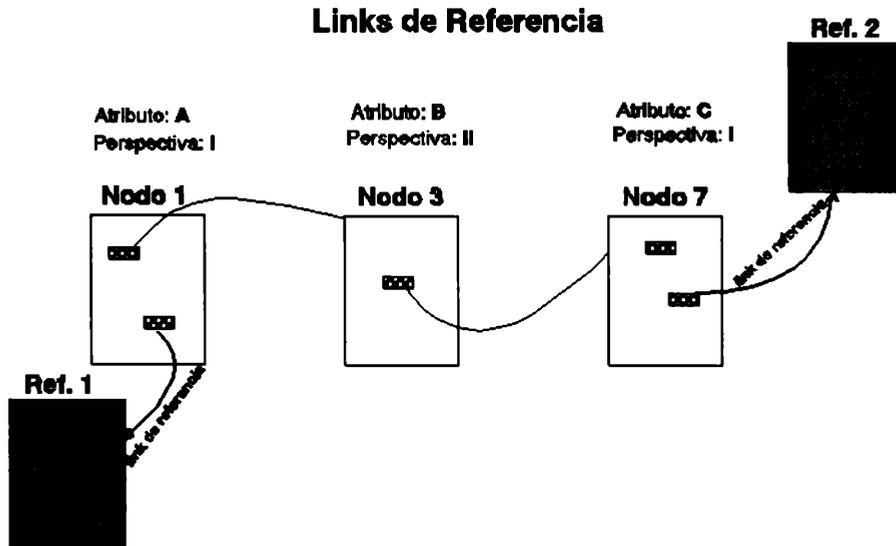
* Los *links* de orden están relacionados con la ubicación de la nota dentro de la perspectiva, de acuerdo al valor de algún Atributo de Nodo.

Links de Orden

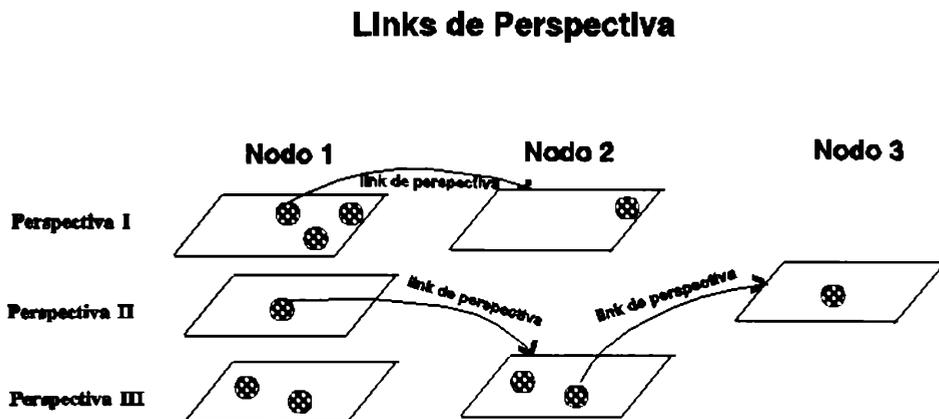




* Los "Items de anotación" de KMS son denominados "*Links de referencia*" en MDH. Estos *link*ean a Nodos de Referencia, ampliando o detallando el contenido del nodo.



En MDH se agrega otro tipo de *links* definido en HDM que son los "*links de Perspectivas*". Estos permiten al lector moverse entre diferentes perspectivas de una misma componente. Se activan al cambiar de perspectiva o contexto dentro de una nota.





En algunos sistemas los *links* son objetos con estructura interna, que provee más información a cerca del destino del *link*. En **MDH** al igual que en **KMS** un *link* no es un objeto sino más bien una propiedad de un botón. Los *links* no tienen ninguna otra estructura interna que no sea el identificador del nodo que representa el destino del *link*. Un botón *linkeado* provee información a cerca del destino del *link*, esto reduce la necesidad de mecanismos de edición de estructura interna de los mismos.

2.1.4 BOTONES

En los Hypertextos, las conexiones entre piezas de información son usualmente visualizadas por medio de botones. Los botones son áreas bien identificables de la pantalla que señalan la existencia de una conexión y pueden ser seleccionados por el lector para llegar al *link* destino. El mecanismo de botones también permite al autor ocultar selectivamente algunos *links* para algunas perspectivas.

En **MDH** se distinguen los siguientes tipos de botones, los "estáticos" y los "dinámicos".

- * Los botones estáticos son los creados por el autor de la hypermedia y permiten navegar a través de los *links* de navegación virtuales y los *links* de orden.

- * Los botones dinámicos son los creados por el usuario de la hypermedia. Estos pueden ser representados por medio de íconos, botones standard y botones invisibles, permitiendo navegar a través de los *links* de navegación estática y los de referencia.



2.1.5 HYPERBASE

Para definir la hyperbase de **MDH** hemos usado parte del esquema de la base de datos de Hypercard.

Hypercard considera las siguientes cinco características físicas:

- * **CARD** son los contenedores básicos de información de los datos incorporados al Hypertexto. En **MDH** se los denomina "NODOS DEL HYPERTEXTO" y contienen el conjunto de los nodos que componen el Hypertexto.

- * **BUTTON** son puntos en los nodos para *linkear* o mandar mensajes a otros objetos. En **MDH** se los denomina "BOTONES".

- * **FIELD** campos que son contenedores de información, usualmente texto. En **MDH** se los denomina "ATRIBUTO DE NODO"; contienen información de cada uno de los atributos de los nodos definidos por el usuario. Al conjunto de Atributos de un Nodo se lo denomina "NODO DE DATOS".

- * **STACK** es una lista ordenada de nodos que soportan operadores "próximo", "anterior". Contiene todas las cartas (nodos en MDH) que son secuencialmente *linkeadas*. En **MDH** se la denomina "PILA DE LINKS DINAMICOS".

También fue necesario definir en **MDH** una estructura, que soporte los nodos de referencia, ya que la información necesaria para los mismos es diferente de la de los nodos de datos.

- * **NODOS DE REFERENCIA** son los contenedores de los datos de las referencias.



Como **MDH** soporta el concepto de perspectivas, se adicionan dos nuevas características físicas:

* **CONTEXTO-BOTONES O PERSPECTIVA** es el conjunto de contextos y botones asociados al mismo contexto para cada identificador de **NODO DE DATOS**.

* **ARBOL DE CONTEXTOS** es un árbol que contiene la información de los contextos y los identificadores de los **NODOS DE DATOS** asociados a cada uno de ellos.

Otra nueva característica física a considerar es la creación de descriptores o índices para filtrar la información contenida en la Hyperbase y así recuperar sólo la que se necesite o sea relevante para la consulta realizada por el usuario.

Cada estructura índice consiste en un identificador de clave primaria dada por un atributo de nodo y el conjunto de los identificadores de nodo de datos que coinciden con el valor de ese atributo.

El esquema usado para modelar contiene las siguientes siete estructuras:

NODOS-DEL-Hypertexto (id_nodo_dato, nodo_de_datos)

BOTONES (contenido_botón, contexto_link_destino, formato_botón, icono_botón, link_destino, punto_inicio, punto_fin, tipo_botón)

ATRIBUTO-DE-NODO(atributo_nodo...)

PILA-DE-LINKS-DINAMICOS(id_nodo_dato)

CONTEXTO-BOTONES(id_nodo_dato, id_contexto, conjunto_botones)

ARBOL-DE-CONTEXTOS(id_contexto, id_contexto_padre, conjunto_contextos_hijos, conjunto_id_nodo_dato)

NODOS-DE-REFERENCIA(id_nodo_ref, titulo_ref, contenido_ref)



La estructura de cada uno de los índices tiene la siguiente forma:

INDICE-ATRIBUTO(atributo_nodo, conjunto_id_nodo_dato)

En Hypercard las propiedades de interconexión de nodos se modela con la tabla LINK. Esta representa cada *link* como un par ordenado de identificadores de carta, destino y fuente. En MDH no hemos necesitado definir esta tabla ya que se *linkea* a un nodo completo, por lo tanto lo único que se necesita es el *link* destino (la información del mismo se encuentra detallada en la estructura BOTONES).

2.2 PERSPECTIVAS EN AMBIENTES DE HIPERTEXTOS

2.2.1 DEFINICIÓN DE PERSPECTIVAS

En los últimos años los Hypertextos han sido usados para sistemas que representan y manejan información, donde el énfasis está en compartir y colaborar con más información, los Hypertextos que caen en estos ambientes deben estar habilitados para soportar organizaciones de datos que cambien a través del tiempo. En muchos casos las organizaciones de datos cargadas en Hypertextos varían entre usuarios y aplicaciones.

Debemos permitir mantener varias configuraciones de la red de acuerdo a las necesidades de un usuario en particular, proyecto o ambiente destino.

El grupo de Intermedia está concentrado en proveer al usuario la forma de manejar la complejidad creciente en el ambiente de Hypertexto. Ellos consideran que múltiples *links* que salgan del mismo punto del documento, pueden confundir al lector, su alternativa es tener un ícono *link* en el material, que pueda rápidamente consultar vía *mouse* los *links* específicos, sus nombres y sus nodos destinos.

Ellos proponen una construcción llamada red (*web*), para implementar la



muestra de *links* dependientes del contexto, cada *link* pertenece a una o más redes, y es solamente visible cuando una de estas redes está visible.

Para ver documentos que pertenecen a una red particular, un usuario abre una red entonces abre también uno o más de sus documentos. Aunque otras redes pueden también referenciar el documento, solamente los que están en la corriente red son mostrados, por lo tanto, el usuario no puede filtrarse a través de las conexiones hechas en diferentes contextos. En **MDH** existe la posibilidad de acceder a otros contextos atravesando los *links* de perspectivas. Las perspectivas aumentan la flexibilidad del Hypertexto permitiendo la construcción y el manejo de múltiples redes.

En **MDH** una perspectiva define un contexto semántico, por ejemplo una versión, una visión, etc. Un contexto sería un conjunto de nodos con *links* que relacionan a los nodos. En un contexto dado los identificadores de nodos son únicos, no puede haber dos nodos con el mismo identificador de nodo. Los botones mostrados son únicamente los botones que corresponden al contexto corriente y señalan un *link* a un nodo destino que puede pertenecer al contexto corriente o a otro contexto.

Las perspectivas o los contextos pueden compartir identificadores de nodos, los nodos que en diferentes perspectivas comparten un identificador común son llamados compatibles. Es importante notar que un nodo que pertenece a más de una perspectiva no necesita ser copiado, las perspectivas pueden ser simplemente listas de punteros a nodos del sistema.

La ventaja clave de las perspectivas es la flexibilidad, los usuarios pueden organizar su material de una manera informal con nodos que contienen muchos tipos de contextos y expresan varias clases de relaciones.



2.2.2 VENTAJAS DE LAS PERSPECTIVAS

La navegación es la única gran dificultad para los usuarios de Hypertextos, perderse es una experiencia habitual en el uso de los mismos. Cualquiera que ha usado un mapa puede testificar qué dificultoso es encontrar algo especialmente si el mapa contiene mucha información. Los diseñadores de mapas han tratado de aliviar este problema produciendo numerosos mapas del mismo área, cada uno conteniendo diferente información.

De una manera similar, un sistema de Hypertexto puede ser menos problemático para sus usuarios si éste presenta solamente las piezas relevantes de la información mientras oculta el resto. Las perspectivas son particularmente adecuadas para esta clase de información oculta; así como la gente tiene diferentes mapas del mismo área, los usuarios de Hypertextos pueden tener un número de perspectivas donde cada una presente diferentes aspectos de la misma información.

Las perspectivas son también adecuadas para sistemas de Hypertextos basados en nodos, ya que cuando se baja en la jerarquía de nodos, el detalle de la información disponible crece.

Asignando una perspectiva diferente a cada nivel de detalle, podemos incrementar la profundidad de la información presentada simplemente agregando más perspectivas.

Es posible usar perspectivas como contenedores de estructuras de tal manera que el usuario pueda fácilmente cambiar de una a otra durante la navegación. Las perspectivas pueden ser combinadas de tal manera que múltiples estructuras pueden ser accedidas separadamente o como una unidad.



2.2.3 ESTRUCTURAS VIRTUALES

Las perspectivas pueden ser usadas para implementar estructuras virtuales. La noción de estructuras virtuales para Hypermedia es una adaptación directa del concepto de visiones en el mundo de los sistemas de bases de datos relacionales. En las bases de datos relacionales, una visión es una tabla construida en tiempo de instanciación aplicando una definición visual a los datos cargados explícitamente en las tablas de la base.

El objetivo es hacer tablas basadas en visiones, idénticas a las tablas de la base, desde una perspectiva usuaria. Cualquier operación posible en una entidad de la base de Hypermedia, sería aplicable a estructuras virtuales.

La noción de estructuras virtuales en Hypermedia sería posible aplicando solamente en un sistema que soporte un mecanismo sustancial de búsqueda/consulta en la red de Hypermedia.

La definición de las componentes en estructuras virtuales son en realidad consultas. Instanciar una estructura virtual involucra que se satisfagan estas consultas y que se construya una entidad dinámica desde los resultados. Aunque no es estricto, tendría sentido que el concepto de consulta usado para descripciones de estructuras virtuales, sea el mismo, que el usado para búsqueda e interfaces filtradas.

Al implementar estructuras virtuales hemos definido *links* virtuales, que son contruidos siempre que un nodo es accedido y mostrado. Conectan el nodo mostrado a nodos que el usuario ha visitado recientemente, esto permite a los usuarios moverse rápidamente desde donde ellos venían. Esto ayuda al usuario a no perderse en la navegación del Hypertexto.



2.3 CONTEXTOS ANIDADOS

2.3.1 MODELO

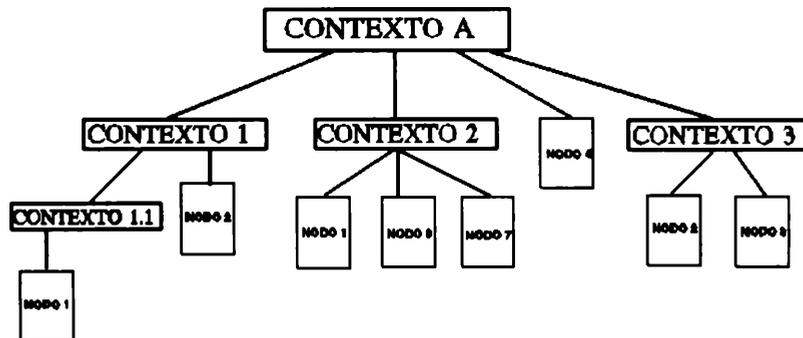
En **MDH** también se ha adoptado el concepto de modelo de contextos anidados (Ref. 9) para la definición, presentación y *browsing* (lectura) de documentos. El modelo de definición introduce los conceptos básicos para estructurar documentos, a saber el nodo y el *link* (como en los sistemas de Hypertextos estándares). El modelo combina cuidadosamente los *links* de Hypertextos con un tipo especial de nodos llamados **Nodos Contexto**, usados para agrupar conjuntos de nodos incluyendo nodos de contexto. Los nodos contextos pueden ser anidados en cualquier profundidad y generalizan la clásica jerarquía de organización de documentos.

Los submodelos de presentación y navegación definen una abstracta y flexible interfase de programas de aplicación. El primero de ellos introduce el concepto de la presentación de un nodo documento, que de alguna manera captura la idea de que diferentes aplicaciones puedan observar el mismo nodo en diferentes formas, esto también define el concepto de *estado* (*state*) que de un modo general corresponde a un modelo para un conjunto de documentos parcialmente recuperados.

Finalmente el submodelo de navegación define la navegación primitiva y las operaciones de navegación del más alto nivel. Esto ofrece un rico conjunto de operaciones para explorar la doble estructura de un hyperdocumento que está definido por los *links* e inducido por la anidación de nodos contexto.



CONTEXTOS ANIDADOS



2.3.2 DEFINICION DE SUBMODELO

La definición de hyperdocumentos en el modelo de contextos anidados está basado en dos conceptos: nodos y *links*. Los nodos son fragmentos de información y los *links*, interconexiones entre nodos en redes de nodos relacionados.

El modelo llega a ser más amplio y distingue dos clases básicas de nodos llamados nodos terminales y de contexto, estos últimos serían el concepto central del modelo.

Intuitivamente un nodo terminal contiene datos cuya estructura interna, si la tiene, depende de la aplicación y no sería parte del modelo. La clase de nodos terminales podría estar especializada en otras clases (texto, voz, imagen, etc), según lo requieran las aplicaciones. La clase del nodo determina sus atributos (los nombres de atributos y operaciones podrían ser las mismas es decir dos clases pueden tener los mismos nombres de atributos y de operaciones), que contiene información definida por el usuario o implementación dependiente de la información.

Independientemente de la clase, cada nodo N debe tener un atributo "contenido" que describe el dato asociado con el nodo y un atributo "id" que



asigna un único identificador de nodo a nodo. Los valores de estos atributos serán referenciados simplemente como contenido e identificador de nodo.

Un nodo contexto agrupa un conjunto de nodos terminales o de contexto, recursivamente. El concepto de nodos contexto de esta manera permite organizar, jerárquicamente o no, conjuntos de nodos y ofrece un mecanismo para definir diferentes visiones de un mismo documento ajustándose a diferentes aplicaciones o clases de usuarios.

Si N es un nodo contexto, entonces sus contenidos deben definir un par (N,L) donde N es un conjunto de nodos y L es el conjunto de *links*, cuyos nodos finales pertenecen a N . Nosotros decimos que N contiene un nodo M , sí y sólo sí, M está en N , y que N contiene un link l , sí y sólo sí, l está en L . Se podría aclarar que un nodo M puede estar contenido en más de un nodo contexto pero todo cambio en cualquiera de los valores de los atributos de M llegaría a ser visible en todos los nodos contextos que contienen a M , en cambio un *link* es interno al nodo contexto, esto es, los *links* no son compartidos por nodos contexto. Decimos también que N contiene recursivamente al nodo M , sí y sólo sí, N contiene a M o N contiene un nodo contexto que recursivamente contiene a M . Para ejemplificar esto último, consideremos en el gráfico de la página 58, a M como nodo1 y a N como contexto1. Si bien nodo1 está definido como nodo terminal del contexto 1.1, éste está contenido dentro del nodo contexto1.

Un nodo contexto define un Hyperdocumento en nuestro modelo Una Hyperbase es un conjunto de nodos H , tal que para cualquier nodo contexto N en H , si N contiene un nodo M , entonces M esta en H . La Hyperbase es consistente, sí y sólo sí, no hay dos nodos que tengan el mismo identificador, el proceso de agregar un Hyperdocumento a una Hyperbase H consiste entonces en agregar un nodo contexto h a H y recursivamente agregar a H todos los nodos contenidos recursivamente en h que no estén ya presentes en H .

Un *link* pertenece al nodo contexto que contiene sus nodos finales, por



Finalmente nosotros decimos que un *link* (L,l) es visible en N desde una perspectiva P para N, sí y sólo sí, hay un nodo contexto M, tal que M está presente en P, M contiene L y N está presente en uno de los *links* (representación del origen de un *link*) de L. Nosotros denotamos el conjunto de *links* visibles en N desde P por $V(P)$.

La representación de los contextos anidados en **MDH** justifica la utilización de la estructura **ARBOL-DE-CONTEXTOS**.

Para implementar el concepto de perspectivas se utiliza la estructura **CONTEXTO-BOTONES**, la cual refleja la posibilidad de representar un nodo en distintos contextos, ya que se cuenta con un identificador de nodo y cada uno de ellos tiene asociados los identificadores de contexto para los contextos a los que pertenece el nodo y para cada identificador de contexto los botones (representación del origen de un *link*) definidos para el mismo.

2.4 CONSULTAS A LA HYPERBASE

En este punto detallaremos los mecanismos de acceso a la Hyperbase para recuperar la información a partir de una condición que satisfaga un conjunto de criterios definidos por el usuario.

En **MDH** para definir el resultado de ejecutar una acción para aplicaciones *query*, se ha adoptado el criterio de ejecutar la consulta y presentar los resultados sin mostrar la especificación de la misma.

Las aplicaciones *query* se basan en dos componentes: La Hyperbase y sus mecanismos de acceso. En HDM se ha definido a la Hyperbase como el espacio de información estructurada que puede ser descrita vía sus primitivas estándar; y los mecanismos de acceso que proveen la forma de recuperar información de la Hyperbase y comenzar la navegación en ella.



En HDM se han clasificado cuatro tipos diferentes de acceso:

1 - Mecanismos de *path*: provee un conjunto de líneas dirigidas permitiendo el acceso en una secuencia preestablecida a unidades de la Hyperbase. El paradigma de la navegación básica es la secuencia ordenada (típicamente dada con botones próximo, anterior, primero, último). En **MDH** hemos definido a este tipo de acceso como CONSULTAS DE ORDEN, las cuales nos permiten navegar por las distintas componentes dentro de un contexto establecido por el usuario.

2 - Indexados: Permite al lector seleccionar los tópicos que le interesan a través de una serie de índices, generalmente organizados jerárquicamente. El resultado de la selección será o una entidad simple o un *path* lineal limitado. Este es el caso de las CONSULTAS POR CONTEXTO en **MDH**, donde se puede seleccionar un contexto y obtener como respuesta un conjunto de Nodos de Datos que pertenecen a dicho contexto, pudiéndose mover linealmente dentro del mismo por medio de los Botones de Orden.

3 - *Querys*: Permiten especificar al lector su interés a través de formulaciones precisas usando palabras claves o atributos previamente asociados con nodos (o componentes), análogos a los mecanismos de *querys* para las bases de datos. Estos mecanismos necesitan información formateada (atributos) para ser vinculada a nodos. El resultado de ejecutar tal *query* es típicamente un *tour* guiado lineal, el cual ha sido armado dinámicamente durante la evaluación de la consulta. En **MDH** se ha denominado a esta consulta como CONSULTA POR ATRIBUTOS, donde se permite al usuario armarla seleccionando los atributos deseados, y de esta forma obtener como resultado el conjunto de Nodos de Datos que satisfacen la consulta. Así se llegan a obtener distintas visiones de acuerdo a la consulta realizada y se logra filtrar mayor cantidad de nodos obtenidos, así como realizar una mayor especificación de los datos buscados.

4 - Asociativos: Es similar a los *querys*, la diferencia está en que la información deseada es especificada usando patrones en el contenido de los nodos o



componentes, o en las propiedades estructurales de la Hyperbase. La ventaja es la gran flexibilidad; la desventaja es una menor precisión en la identificación de la información deseada. En **MDH** podría realizarse este tipo de consulta especificando un atributo del Nodo de Datos como atributo clave y que por medio de la consulta devuelva todos los nodos que tengan el valor de dicho atributo.

2.5 INTERFASE

En cuanto al estilo de la interfase usuaria, en **MDH** hemos tratado de crear un ambiente en el cual el usuario tenga pocos conceptos que aprender. Por ej. usar únicamente dos tipos de nodos (de datos y de referencia) y restringir el número de *links* explícitos a dos (de navegación y de referencia).

Consideramos adecuado para disminuir la complejidad separar la edición de los datos, de la consulta de los mismos. Así como también hacer uso del *mouse* para reducir el tiempo de operación.

En cuanto a la representación de los nodos en **MDH**, cuando un usuario selecciona un botón *linkeado* a otro nodo, el nodo corriente es reemplazado por el nuevo nodo en el caso de los *links* de navegación estática, en el caso de los *links* de referencia se abrirá un nuevo nodo sobre el nodo desde el que se *linkea*, volviendo al nodo de inicio al salir del nodo de referencia.

2.6 POR QUE UNA APLICACION DE MANEJO DE NOTAS PERIODISTICAS?

Un archivo de noticias en una editorial es un conjunto de artículos recortados de diarios y revistas, o cassettes con grabaciones de notas radiales o televisivas,



clasificados siguiendo algún criterio. Estos son guardados de acuerdo a su clasificación. Es obvio que para mantener un archivo en forma completa y actualizada, se tendrán volúmenes de artículos tan grandes que hacen que su manejo sea complejo y demande mucho tiempo.

Una consulta al archivo involucra conocer por parte del usuario el criterio de clasificación de los artículos.

Cuando una consulta se lleva a cabo, el resultado obtenido es un conjunto de artículos que responden a una misma clasificación, no pudiendo relacionarse con otros artículos que, a pesar de tener distinta clasificación contienen información que es valiosa para la consulta, así como tampoco ampliar el contenido del mismo.

Planteada esta problemática consideramos que el manejo de un archivo de noticias sería un ámbito adecuado para explotar el modelo **MDH** propuesto en el punto anterior por varias razones:

1 - La información que guarda un archivo de notas periodísticas proviene de distintos medios: gráfico, sonido, imágenes y textos. Los nodos propuestos en **MDH** permiten almacenar información de distintos medios.

2 - **MDH** propone al Hypertexto como forma de representación de los datos, como ellos están estructurados por una red de nodos, cada nodo del Hypertexto puede ser visto como un analogía a un artículo. Cada uno de ellos contiene un conjunto de *links* que los relacionan permitiendo al usuario navegar a través de ellos. Además los nodos y *links* pueden ser cambiados rápidamente, la información de los nodos pueden ser actualizada, nuevos nodos pueden ser agregados dentro de la estructura del Hypertexto, así como también nuevos *links* para mostrar nuevas relaciones, lo cual es indispensable en el manejo de notas periodísticas, debido a que al incorporar artículos uno puede querer relacionarlos con otros ya existentes. Por otro lado si bien un artículo puede pertenecer a un determinado contexto por ej: deporte, éste también puede tener



connotaciones de otros contextos por ej: social, político, etc.; por lo que el artículo puede pertenecer a varios contextos y de acuerdo al mismo relacionarse con distintos artículos.

3 - Puede suceder que si bien un artículo pertenece a un determinado contexto por ejemplo: deporte, también tenga connotaciones en otros contextos, por ejemplo: social, político, etc; por lo cual este artículo también pertenecería a varios contextos, y de acuerdo a cada uno de ellos relacionarse con distintos artículos. Por lo cual utilizar el concepto de perspectivas definido en **MDH** solucionaría esta problemática.

4 - Los artículos en un archivo de notas periodísticas no son un conjunto de información clasificada estáticamente, no todos los artículos se relacionan entre sí por el vínculo de la clasificación(contexto), sino que existen otras relaciones que se establecen entre conceptos comunes en los artículos, por lo cual utilizar un Hypertexto para representar la información guardada en una hyperbase, sería lo más apropiado ya que la característica principal como ya lo expresamos en el punto 1.1 es permitir la organización no lineal de la información, con las ventajas de relacionarla de diferentes formas dependiendo de distintos puntos de vista, y ampliar la información a través de referencias.

Como un archivo de noticias fundamentalmente sirve para realizar consultas a la información almacenada en el mismo; si ésta se encuentra almacenada en una forma lineal; el resultado de la misma es un conjunto de artículos de los cuales no se puede obtener más información de la que está contenida en los mismos, ya sea por referencias o por relaciones con otros artículos. Por lo cual tener los artículos relacionados por otros vínculos (autor, fecha, palabra clave, fuente) que no sea, su clasificación, permitiría obtener información relevante para la consulta de los artículos que no pertenezcan a la misma categoría.

5 - Además los artículos resultantes de una consulta pueden no ser todos útiles,



debido a que para distintos contextos un artículo puede ser relevante o no, por lo cual se pensó que la posibilidad de contextos anidados propuestos en el modelo **MDH** permitiría resolver este problema.

PROTOTIPO IMPLEMENTADO



3.1 LENGUAJE Y PLATAFORMA USADA - JUSTIFICACION

3.1.1- LENGUAJE

Hemos seleccionado al Smalltalk V R.2.0 bajo Windows como el lenguaje de base de nuestra aplicación, y por pertenecer éste al Paradigma orientado a objetos, vamos a explicar la teoría de objetos y ventajas de la misma.

A- Introducción a la Teoría de Objetos (Ref.7)

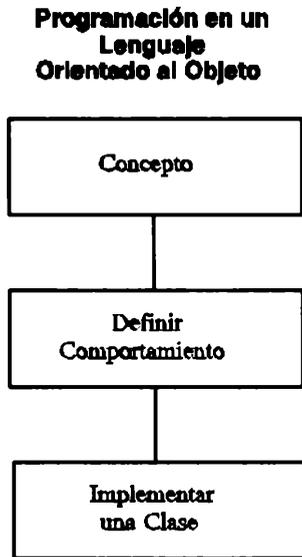
Vamos a definir inicialmente la diferencia que existe entre la programación orientada a objetos y la programación tradicional.

La programación orientada a objetos es fundamentalmente diferente de la procedural tradicional. La programación Orientada a Objetos describe un sistema en términos de los objetos involucrados. La programación tradicional describe sistemas en términos de sus funcionalidades.

En la programación tradicional se refina la solución del problema algorítmicamente paso a paso, cada paso en el proceso describe una solución al problema en un cierto nivel de abstracción. Los sistemas refinados de esta forma son más fáciles de describir usando un diagrama donde los módulos principales son organizados jerárquicamente y donde cada módulo representa una función o subproblema en la solución. Un diseño producido usando una descomposición funcional se adecua a las aproximaciones procedurales donde las subrutinas, procedimientos o subprogramas son los mecanismos predominantes para estructurar el código. Hay un directo mapeo entre los módulos funcionales en el diseño y procedimientos en el código.



En cambio en la Programación Orientada a Objetos inicialmente los objetos tienen que ser identificados, la próxima tarea es identificar sus características y definir las relaciones entre ellos.



B- Características de la Programación Orientada a Objetos

Un sistema orientado a objetos puede ser descrito como un conjunto de objetos comunicándose con otro para alcanzar algún resultado. Cada objeto puede ser pensado como una pequeña máquina virtual con su propio estado (o memoria) y su propio conjunto de operaciones (o conjunto de instrucciones). La funcionalidad se logra por el envío de mensajes a objetos. Cuando un objeto recibe un mensaje éste determina si es una operación apropiada o método que le permita responder al mensaje. La definición del método describe como reaccionará el objeto después de recibir el mensaje. En la terminología orientada a objeto nos referimos a la colección de operaciones que definen el comportamiento de un objeto como el protocolo soportado por el objeto.

Método = Sinónimo de Operaciones. Invocado cuando un mensaje es recibido por un objeto



Protocolo = Conjunto de mensajes al cual un objeto responde.

La programación orientada a objetos tiene mucho en común con la programación con tipos de datos abstractos. De hecho la programación orientada a objetos puede ser pensada como dentro de este tipo de programación, y extendida sobre dos nociones adicionales de programación - polimorfismo y herencia -.

La idea de la "abstracción de datos" y "tipos abstractos de datos" indica que es posible definir entidades, y separar la implementación de esa entidad del uso de la misma. Se puede pensar que hay un proveedor y un cliente en términos de programación. Por ejemplo, alguien que implementa una clase (un tipo abstracto de datos) y alguien que lo utiliza, que pueden ser dos personas totalmente diferentes. Lo que se logra cuando se establece una separación de este tipo es la independencia entre ambos. El nexo que se establece permite que el uso del tipo de datos abstractos se mantenga, aún cuando cambie la implementación del mismo. Entonces, todo el trabajo de programación, análisis, etc; que se haga luego usando este tipo abstracto de datos, permanecerá aún cuando cambien los detalles de implementación.

En Programación Orientada a Objetos esto se logra direccionando todo el uso de la clase a través de una interfase única coherente y que tiene distintos nombres según el lenguaje de programación con el que se trabaje. En Smalltalk la terminología habitual es la de mensaje. Existen objetos que son instancias de tipos abstractos de datos a las que se les envía un mensaje.

Los objetos (y los tipos de datos abstractos) se adhieren a un principio importante para la estructuración de sistemas de *soft* - ocultamiento de la información-. La idea de la información oculta es que los usuarios de un objeto no necesitan acceder a su representación o a la implementación de sus operaciones. La principal ventaja de esta aproximación es que permite al que implementa, modificar la implementación de un objeto en una manera que es transparente al usuario. Los usuarios o clientes del objeto no tienen



por que ser notificados del cambio. Esta separación de la interfase de los usuarios de los objetos de su implementación es esencial para la producción de *software* reusable y mantenible.

Un objeto puede ser como una abstracción del dominio del problema con un estado privado caracterizado por el protocolo de mensajes que él soporta.

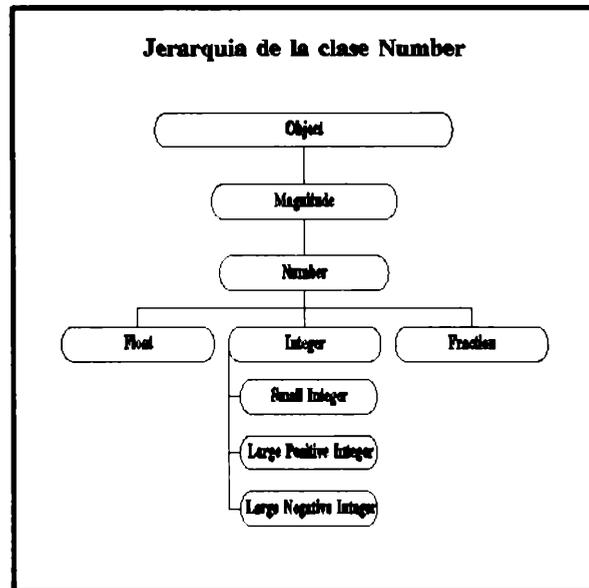
Una vez definida una clase, interesa que cada vez que se tenga que enunciar un concepto relacionado con el anterior no haya que especificar todo nuevamente. Por ejemplo, si se definió una clase vehículo con una cantidad de propiedades del mismo, es deseable que si se tiene que incorporar un objeto camión o automóvil, que es un caso específico de un vehículo, haya que especificar solamente las diferencias y no todo de nuevo, que es lo que habitualmente pasa con la programación tradicional.

Por ejemplo cuando se aprende zoología se van conociendo las cosas a través de especificaciones más detalladas. Por ejemplo, los animales se dividen en multicelulares y unicelulares, después en vertebrados e invertebrados y ya se sabe, cuando se habla de mamíferos, que es un vertebrado y por ser vertebrado tiene una cantidad de características conocidas. Esto ahorra mucha información y detalles al transmitir conocimientos.

En Programación Orientada a Objetos esto se conoce como **herencia**. Dada una clase, puede surgir la necesidad de implementar o definir una subclase que tenga todas las características que tenía la anterior, más otras que le son propias. A su vez, esta clase puede haber heredado una cantidad de información de su superclase.



Gráficamente tomando como ejemplo la clase *Number* de Smalltalk podemos ver la herencia de la siguiente forma:



Volviendo al ejemplo de zoología, se puede estar hablando de los vertebrados, los mamíferos, los osos y tal vez siguiendo esta descomposición con más nivel de detalle, se tiene la familia de los panda, polar, etc. Es decir, cada vez se va especializando la definición de una clase. En síntesis, el objetivo es permitir definir nuevas clases, nuevos conceptos en función de otros existentes, no repitiendo todo el trabajo hecho. La ventaja importante de la reutilización es que se aprovecha lo que se hace, no se está reimplementando cada vez.

Otra de las ventajas de la herencia es la centralización. Si se implementa una clase determinada y existe un error en dicha implementación, la corrección del error se realizará sobre la clase errónea.

Por ejemplo, todo lo referente a los vertebrados va a estar definido en la clase vertebrados. Si hay un error en la implementación de la clase, no se necesita modificar las 50.000 especies de vertebrados que existen en la



realidad y que se han definido a lo largo de los años. Solo se corregirá el problema dentro de la clase vertebrados.

Otra de las características más importantes de la programación orientada a objetos es el **polimorfismo**, que es la manipulación común en operaciones genéricas. Por ejemplo, si para una aplicación gráfica se crea una clase figura, la misma podrá tener varias figuras geométricas que se desplegarán en una pantalla. Estas figuras son: polígonos, rectángulos, círculos, rectas, etc. Si se considera la operación de rotación de todas las figuras, cada una de ellas va a tener su manera de rotar. Un círculo no tiene que hacer nada, rotar un círculo es la operación nula. Un cuadrado rotará de una manera, una recta de otra, etc. Pero si todas estas clases son subclases de la clase figura, en un lenguaje orientado a objetos se puede usar la técnica llamada polimorfismo que consiste en tratar a todas como figuras y, si se requiere la operación de rotación, cada una responderá a la operación en función de la figura en sí.

En síntesis cada objeto se comporta en función de lo que es. Su comportamiento es similar a un llamado a procedimiento, sólo que la ligadura al mismo se efectúa posteriormente, por lo tanto, hasta que el mensaje no sea enviado a un objeto (o figura) específico no se va a determinar el método real a invocarse; esta correspondencia entre el objeto y la posible operación que se determina en tiempo de ejecución, es lo que se conoce con el nombre de **binding** dinámico.

Esto puede hacerse en un lenguaje convencional, utilizando sentencias de flujo de control como el "if"; pero la diferencia es que dichas estructuras de control estarán repartidas por todo el código, no centralizadas en los objetos. Por lo tanto cuando el usuario debe realizar una modificación, ésta se debe efectuar en todos sus programas en lugar de hacerlo en un solo lugar.



C- Ventajas de la Programación Orientada a Objetos

La implementación de sistemas basándose en el paradigma de objetos presenta varias ventajas sobre los métodos tradicionales:

* **Reducción de la brecha semántica:** es más natural observar al mundo como compuesto por entidades que por procedimientos y datos, ya que existe una correspondencia biunívoca entre las entidades y los objetos definidos en el sistema.

La definición de clases permite agrupar objetos con características similares con un cierto nivel de abstracción apropiado. Posteriormente con la especialización (herencia) se llega a definiciones de objetos más específicos, cubriéndose de ese modo todo el espectro de posibilidades de análisis, de los objetos más generales a los más particulares.

* **Encapsulamiento:** el hecho de asociar a cada objeto con su comportamiento (conjunto de mensajes que el objeto puede recibir) provee una gran modularidad, disminuyendo la superficie de contacto entre las partes del sistema.

* **Efecto limitado de los cambios y mantenimiento simplificado:** al separarse claramente el "que" (interfase externa de los objetos) del "como" (estructura interna e implementación de los mensajes), un objeto puede modificarse sin que este cambio afecte el funcionamiento global del sistema.

* **Extensibilidad:** un sistema puede crecer en forma natural, agregándosele nuevas clases sin necesidad de modificar el código existente.

* **Seguridad:** en relación con el punto anterior, los objetos proveen una herramienta para diseñar código seguro, debido a que soportan "información *hiding*" (información oculta) y abstracción de datos. La estructura de un objeto sólo puede accederse desde los métodos del objeto mismo, esto lo



hace resistente a cambios inadvertidos.

* **Reusabilidad:** Es posible observar que el concepto de herencia provee un modo natural de reusar código (en este caso el "código" está compuesto por un aspecto declarativo, los atributos de una clase; y por otro aspecto más procedural, su comportamiento). Al desarrollar un sistema se pueden aprovechar las clases ya definidas, especializándolas para adaptarlas a la aplicación que se necesita.

Particularmente en el sistema Hypermedia CIP, el concepto ya mencionado de herencia (diferencia más importante entre programación orientada a objetos y programación procedural), nos permitirá obtener reusabilidad de código, ya que ocurren eventos similares (diferenciables por pocos datos), sobre entidades (objetos) que cumplen el mismo rol tanto en el sistema como en el mundo real.

3.1.2 PLATAFORMA USADA

El lenguaje utilizado corre bajo la plataforma Windows 3.1 (Microsoft), la cual comenzó como un entorno gráfico que corría sobre DOS en el año 1985. A partir del año 1991 comenzó a ser considerado como un Sistema Operativo.

Optamos realizar el sistema bajo Windows ya que se necesitaba una plataforma que soportara diversos dispositivos de *hardware* y archivos de distintos formatos. Permitiendo combinar (por medio de un sistema de hiperenlaces) texto, gráficos, sonido y animación.

Windows provee un paquete adicional Multimedia Extensions (Extensiones



Multimedia), creado para resolver las necesidades de los autores de multimedias, al ofrecer una vía estándar para incluir sonido y movimiento en las aplicaciones para PC. Estas extensiones son capaces de trabajar directamente con los elementos *hardware* de Multimedia: discos CD-ROM, dispositivos de audio (entrada-salida), etc.

Las Extensiones Multimedia consisten en un conjunto de librerías de enlace dinámico (DLL-Dynamic Link Libraries) que proporcionan el enlace entre las aplicaciones Windows y el *hardware* de Multimedia. Las Extensiones Multimedia traen un cierto número de *drivers*, pero el sistema está diseñado para que admita nuevos drivers a medida que sean creados; por lo tanto Windows con Extensiones Multimedia proporciona una plataforma estándar para aplicaciones multimedia.



3.2 CARACTERISTICAS DEL SISTEMA

3.2.1 REQUERIMIENTOS

Para poder ejecutar Hypermedia CIP será necesaria una PC-Multimedia, definida como la configuración mínima de *hardware* basada en una PC compatible capaz de hacer correr las aplicaciones multimedia y las Extensiones Multimedia de Windows.

Detallaremos los elementos básicos de una PC-Multimedia. Se necesita como mínimo una PC-386 SX con 4Mb de memoria RAM. El sistema operativo Windows con Extensiones Multimedia. Tanto audio como gráficos requieren una gran cantidad de espacio para almacenamiento, por lo tanto se necesita gran cantidad de Mb disponibles.

Deberá contar también con una placa de video capaz de presentar con calidad y rapidez la información gráfica sobre pantalla. Una placa Super-VGA sería adecuada.

Para incorporar audio con calidad mínima que incluya digitalización de audio y conexiones MIDI para crear y editar su propia música será necesaria una plaqueta de sonido.

En cuanto al *software* será necesario instalar en la PC Windows 3.1 y Smalltalk V versión 2.0.

3.2.2 ESTRUCTURA UTILIZADA

Para poder armar la Multimedia CIP hemos definido las siguientes estructuras de acuerdo al modelo **MDH**:



NODOS DEL Hypertexto (id_número_nota, nodo_de_datos)

- * id_número_nota será el número con que se identificará a la nota dentro del Hypertexto, este es único;
- * nodo_de_datos que contiene el conjunto de atributos que caracterizan al nodo que corresponde a dicho número.

BOTONES (contenido_botón, icono_botón, link_destino, contexto_link_destino, formato_botón, punto_inicio, punto_fin, tipo_botón)

- * Contenido_botón contendrá el texto o imagen que identificará al botón en la pantalla;
- * icono_botón ;
- * link_destino contiene el número de la nota o referencia a la que se *linkea*;
- * contexto_link_destino tendrá el contexto al que pertenece la nota a la que se *linkea*;
- * formato_botón contiene el tipo de visualización del botón (común, icono, invisible);
- * punto_inicio tendrá las coordenadas de origen del botón;
- * punto_fin tendrá las coordenadas de finalización del botón;
- * tipo_botón contiene el tipo al que pertenece, que puede ser referencia o navegación.

NODO DE DATO (título, copete, autor, fecha, fuente, medio, palabras_claves, contenido)

- * Medio contiene el tipo de medio al que pertenece la nota, que puede ser imagen, sonido, texto o animación;
- * palabras_claves contiene el conjunto de palabras claves definido por el usuario para esa nota.



PILA DE LINKS DINAMICOS (id_número_nota)

Es la pila que se usa para guardar los números de las notas visitadas durante la navegación.

CONTEXTO_BOTONES (id_número_nota, id_contexto, conjunto_botones)

- * `id_contexto` tendrá uno de los contextos definidos para ese identificador de nota;
- * `conjunto_botones` es una colección de botones para el identificador de contexto definido anteriormente.

ARBOL DE CONTEXTOS (id_contexto, id_contexto_padre, conjunto_contextos_hijos, conjunto_id_número_nota)

- * `id_contexto` contiene un contexto definido por el usuario;
- * `id_contexto_padre` contiene el contexto al que pertenece el nodo inmediato superior.
- * `conjunto_contextos_hijos` contiene a los contextos de los cuales el contexto actual es contexto padre.
- * `conjunto_id_número_nota` contiene la colección de números de notas que pertenecen a dicho contexto.

NODOS DE REFERENCIA (id_número_ref, título_ref, contenido_ref)

En esta estructura se almacenan los datos de las referencias.

También se crearon descriptores para autor, fecha, fuente y palabras claves.



3.2.3 TRATAMIENTO DE ANIMACION Y SONIDO

Las Extensiones Multimedia soportan directamente tres fuentes separadas de información sonora: forma de onda (.wav), utilizada en nuestro prototipo, MIDI y audio CD. Las formas de onda de audio proporcionan la manera de almacenar sonido en forma digital sobre un sistema, en base a muestrear el sonido al menos 11.000 veces por segundo, y almacenar 8 bits de información por cada muestra. Como resulta fácil imaginar, esto llevaría a llenar una gran cantidad de espacio en disco, del orden de 1Mb por minuto para señal estéreo. Las Extensiones Multimedia incluyen programas de Windows que permiten reproducir sonido y realizar algunas funciones sencillas de edición. También proporcionan nuevas funciones que otras aplicaciones pueden usar para grabar y reproducir ficheros en forma de onda.

Los ficheros de música digital MIDI (Musical Instrument Digital Interfase) consisten en partituras para sintetizadores musicales, en lugar de almacenar la muestra de los sonidos. Como resultado, estos ficheros son mucho más pequeños que los que contienen forma de onda de la misma duración; tan solo se necesitan unos 30 Kb de información por minuto.

La tercera forma de sonido no utiliza el espacio en disco. Las Extensiones Multimedia hacen posible reproducir pistas de audio de un CD-ROM que utiliza el mismo formato que un CD-Audio. Esto significa que puede reproducir cualquier CD o pista de audio que esté incluida en un CD-ROM. con ellos se puede seleccionar la pista a reproducir, comenzar, hacer una pausa o parar el sonido tal como se quiera.

Las Extensiones Multimedia proporcionan nuevas funciones de llamada para los sistemas de autor que se pueden emplear para cargar uno de estos ficheros de sonido, comenzar desde el principio o desde cualquier otro punto, y pararlo en cualquier instante deseado. También se puede instruir a Windows para que "coloque y se olvide" de un fichero de sonido permitiéndole que arranque y



luego siga ejecutando otros comandos mientras que suenan las pistas de música.

Otro de los componentes principales de una presentación multimedia es la animación. Hay varias formas de tomar imágenes que se muevan en la pantalla; la más sencilla es tomar una o varias de las imágenes gráficas de tipo *bitmaps*, y moverlas a lo largo de la pantalla dibujándolas y borrándola alternativamente, desplazando la posición en pequeños incrementos cada vez, y grabando la secuencia ("*path animation*"). Cambiando las imágenes *bitmaps* utilizada para las imágenes secuenciales también se logra la ilusión de una animación dentro de la imagen en movimiento ("*frame animation*").

3.2.4 INTERFASE Y OPERACION

El sistema que realizamos es un prototipo que permite incorporar y consultar notas periódicas que pueden pertenecer a varios medios.

Vamos a explicar el uso de este prototipo, describiendo su interfase y sus funciones.

Desde el punto de vista de la funcionalidad, cuando se ingresa al sistema se presenta la pantalla principal del mismo, en la cual el usuario tiene dos opciones principales para seleccionar:

- * Actualizar
- * Consultar

Si selecciona **Actualizar**, puede realizar las siguientes tres acciones:

- * Incorporar una nota



- * Modificar una nota

- * Borrar una nota

Al **Incorporar una nota**, se abrirá una ventana que le permite ingresar los datos de una nota: título, copete, autor, fuente, palabras claves, fecha y medio. Los medios que puede seleccionar son: texto, imagen, sonido y animación; seleccionando uno de ellos se abrirá otra ventana que permitirá ingresar el contenido de la nota y definir las distintas perspectivas para la misma. Cada perspectiva se define al ingresar un nuevo contexto para la nota y los botones para el mismo. Mediante la incorporación de contextos y subcontextos en el ARBOL DE CONTEXTOS se va creando la estructura para el manejo de Contextos Anidados.

Al seleccionar **Modificar una nota**, aparecerá una ventana, que le permitirá seleccionar la nota a modificar, eligiendo el contexto y título de la misma. Una vez ubicada la nota, se abre una nueva ventana con las mismas características que la ventana para incorporar una nota, que permite modificar los datos de la misma; en ella se encuentra un botón "Contenido" que habilita por medio de la apertura de otra ventana modificar el contenido, así como también agregar y borrar botones en las perspectivas ya existentes o incorporar nuevas perspectivas.

Para **Borrar una nota**, se debe localizar la nota de la misma manera que en **Modificar una Nota** y confirmar su eliminación de la red de la Hypermedia o del contexto seleccionado.

Si se selecciona la opción de la ventana principal **Consultar Notas**, se podrán seleccionar las siguientes opciones:

- * Consulta por Contexto

- * Consulta por Palabra Clave



* Consulta por Atributo

Si se opta por **Consultar por Contexto**, se abrirá una ventana que le permitirá seleccionar un contexto determinado, luego aparecerán los títulos de las notas correspondientes al contexto elegido y seleccionando uno de estos títulos se mostrará el copete y se abrirá la ventana que permite iniciar la navegación en la Hypermedia.

Si se selecciona la opción **Consultar por Palabra Clave**, se abrirá una ventana que le permitirá seleccionar una palabra clave determinada, luego aparecerán los títulos de las notas correspondientes a la palabra clave elegida, seleccionando uno de ellos se mostrará el copete y se abrirá la ventana que permite iniciar la navegación de la Hypermedia.

Si se elige **Consulta por Atributo**, se abrirá una ventana en la cual por medio de la selección de operadores lógicos, atributos y valores de atributos el usuario arma la consulta deseada. Como respuesta a esta consulta se mostrarán los títulos de las notas que responden a la misma, seleccionando uno de ellos se mostrará el copete, por medio del ícono mostrar se abrirá la ventana que permite iniciar la navegación de la hypermedia.

La ventana que permite al usuario iniciar la navegación en la Hypermedia presenta el contenido de la nota con los botones de referencia y navegación estáticos correspondientes al contexto actual de la misma, también se encuentran los íconos que muestran los atributos, si se *clikea* en el ícono "Contexto" se podrá seleccionar otro contexto que permitirá ver la nota en otra perspectivas; así como también los iconos de navegación dinámica y los de orden.

Desde el punto de vista de la interfase hemos tratado de traducir las acciones a una forma simple de aprender posible.

Elegimos una interfase icónica, por sus características de sencillez de uso,



velocidad de aprendizaje y facilidad en la traducción de una acción deseada a una acción disponible. Como contrapartida, las interfases icónicas limitan la superficie de pantalla disponible y por lo tanto necesitan ser bien diseñadas.

Detallamos a continuación la interfase de Hypermedia CIP, describiendo las pantallas principales del sistema.

La pantalla inicial es la que engloba toda la funcionalidad del sistema, a partir de ella el usuario podrá realizar las siguientes acciones: actualizar, o consultar los datos del sistema; en esta pantalla también se podrán seleccionar los íconos Ayuda (que permitirá al usuario obtener información a cerca de la funcionalidad del sistema) y Salir (que le permitirá salir del sistema).

Otra de las pantallas es la de incorporación de los datos de un artículo, ésta tiene una serie de campos en los que el usuario puede ingresar el valor de los atributos de los Atributos de Nodo, también se encuentran los botones Cancelar (que permite salir de la pantalla sin salvar la nota) y Grabar (que salva el contenido del artículo). Seleccionando el Medio, se desplegará una nueva pantalla en la que el usuario podrá incorporar el contenido de la nota, los contextos, así como también los botones y los *links* deseados para cada contexto. También se encuentran los botones Grabar (que permite grabar la nota en un contexto, seleccionando SI), Ayuda y Salir (que graba el contenido de la nota).

Las pantallas de Consulta presentan distintas características de acuerdo al tipo de consulta a realizar. En la pantalla de Consulta por Contexto se permite seleccionar un contexto y de acuerdo al mismo se mostrarán los títulos de los artículos correspondientes a ese contexto; seleccionado uno de ellos se mostrará el copete de la nota y se abrirá la pantalla que mostrará los datos de la misma.

La pantalla de consulta por Palabras Claves es igual a la pantalla de Consulta por Contextos, con la diferencia que en vez de mostrar los contextos, muestra



las palabras claves definidas en el sistema.

La ventana de consultas por Atributos permite, por medio de los íconos que representan a los operadores lógicos, los atributos y los datos armar, la consulta. Por medio del ícono Procesar (representado por una flecha) se permitirá al usuarios obtener la respuesta a la consulta, dando una lista de los títulos que corresponden a la misma.

En la pantalla que muestra los datos de las notas se muestra el contenido de la nota, con los botones definidos por el usuario, ya sea de referencia o de navegación, así como también los íconos estáticos (definidos por el autor de la Hypermedia) que permitirán al usuario navegar en la Hypermedia.

A continuación se muestran las pantallas principales del prototipo como una forma de ejemplificar de una manera gráfica la interfase.

HYPERMEDIA CIP

MENU PRINCIPAL

ACTUALIZAR

CONSULTAR

incorporar nota

modificar nota

borrar nota

AYUDA

SALIR

INCORPORAR UNA NOTA

Titulo

Los serbios-bosnios votan el plan de paz de la ONU y la C

Copete

Un contingente de 60.000 hombres, entre ellos 20.000 nort

Autor

Thelma Luzzani

Fuente

Clarín

Palabras Claves

ONU

Medios

Texto

Fecha (mmddaa)

05/05/93



 ACTUALIZAR

 CONSULTAR

 consulta por contexto

 consulta por pal. clave

 consulta por atributos

 AYUDA

 SALIR

CONTEXTOS

TITULOS

Política 

Menem anunció la separación de Alicia 

COPETE

Con el presidente Menem a la cabeza, distintas figuras del Gobierno salieron a "sepultar" definitivamente las esperanzas de Alicia Saadi de conservar su banca en la Cámara de Senadores.



MENU PRINCIPAL

Menen anunció la separación de Alicia Saadi del Senado

CONTEXTO: Política

 ACTUALIZAR

 CONSULTAR

 consulta por contexto

 consulta por pal. clave

 consulta por atributos

 AYUDA

 SALIR

La senadora catamarqueña **Alicia Saad** califico de 'irresponsable' y 'autoritario' al gobernador de Catamarca, Arnoldo Castillo, con quien polemizo publicamente y agrego que le fue 'imposible' comunicarse con el ministro del Interior, Gustavo Belliz, para explicar su punto de vista en relacion a su propia eleccion anticipada. Castillo, a su vez, durante un dialogo radial, empleo la palabra 'mafioso' para calificar al hermano de la senadora, Ramon Saadi, diputado nacional por el justicialismo. Castillo, ademas, insistio en que el ahora **detenido presidente** provisional del Senado catamarqueño, **Martin Giordani** aprovechando un descuido de la custodia policial, y disfrazado de mujer, abandono su domicilio, donde se encontraba detenido' Por su parte, Alicia Saadi insistio en la legitimidad de los actos por los cuales fue elegida senadora hasta el 2001.



Primero

Anterior

Posterior

Ultimo



MENU PRINCIPAL

ACTUALIZAR

CONSULTAR

consulta por contexto

consulta por pal. clave

consulta por atributos

AYUDA

SALIR

Consulta por Atributos

PREGUNTA

Operadores Logicos



Atributos



Autores

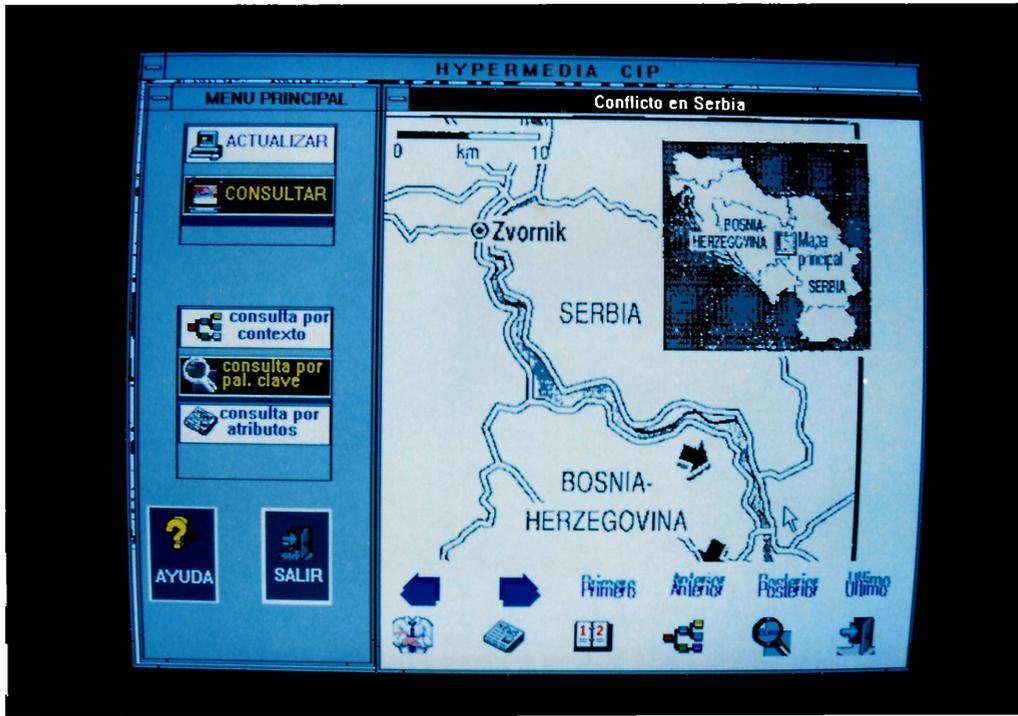
THELMA LUZZANI

RESPUESTA

Titulos

Los serbios-bosnios votan el plan de paz de la C





 ACTUALIZAR

 CONSULTAR

 consulta por contexto

 consulta por pal. clave

 consulta por atributos

 AYUDA

 SALIR

AYUDA

-  Uso de la interfase del Sistema
-  Funciones del Sistema
-  Diccionario de terminos tecnicos
-  Acerca del Sistema





3.2.5 PERFIL DE USUARIO

Teniendo en cuenta las características del sistema prototipado, definimos dos perfiles de usuario:

Usuario con conocimientos periodísticos.

Es el usuario especializado en el tema, quien tiene la responsabilidad de incorporar la información. De este usuario depende la eficiencia de la funcionalidad del Sistema.

Usuario común.

Dentro de este perfil se incluye al usuario con conocimientos mínimos de PC, quien solamente podrá consultar la información almacenada.

DISCUSIONES



4. DISCUSIONES

A continuación enunciamos algunas de las posibles mejoras y extensiones que pueden ser aplicadas al prototipo presentado.

1. Almacenamiento masivo

En caso de trasladar este prototipo a un sistema real, la cantidad de información con la cual podrá operar aumentará en forma considerable, por lo tanto se deberá proveer a la PC-Multimedia de un mecanismo de almacenamiento masivo. En la actualidad, una de las posibilidades que podemos considerar es la incorporación de discos ópticos.

2. Incorporación de Video

El medio que no ha sido incorporado en la Hypermedia CIP es video. Su incorporación se podría realizar a través de la instalación de drivers MCI (Media Control Interfase) de las Extensiones Multimedia de Windows, y de la definición de la forma de capturar y manejar las imágenes de video en Smalltalk.

3. Interfase para archivo

Se debe analizar la posibilidad de creación de archivos ajenos al Smalltalk (archivos de audio, imagen, texto) desde Hypermedia CIP.

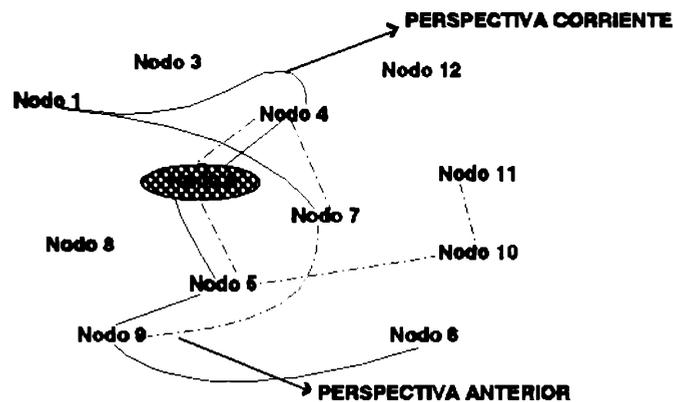


4. Interfase con base de datos relacional

Para poder implementar este concepto se necesita definir el enlace entre Smalltalk y SQL Win u otra base de dato relacional, así como también redefinir la estructura de los datos y la recuperación de los mismos.

5. Browser de Perspectivas

Uno de los inconvenientes en este tipo de sistemas es perderse al navegar por la estructura. Por esta razón es ideal proveer al sistema de una herramienta que le permita al usuario conocer el lugar donde está ubicado en cada momento. Lo cual se podría lograr implementando un mecanismo de *browser* que en esencia sería, acceder en forma *on-line* a los datos que representan la totalidad de la hyperbase, con el grafo de los datos que reproducen la perspectiva corriente. El inconveniente de la misma es que sería difícil de visualizar en caso de que la hyperbase sea demasiado grande.



Otra posibilidad es mostrar el grafo que represente la perspectiva corriente, pero consideramos que un cambio de perspectiva podría llegar a desorientar al usuario ya que podría perder la ubicación dentro de la Hyperbase.



6. Estructuras Virtuales

Las estructuras virtuales son un poderoso mecanismo cuando se combina con la noción de composición. Una composición virtual permite al usuario crear nodos que son dinámicamente construidos en tiempo de acceso desde otros nodos, *links* o composiciones que son cargadas en la red; tales composiciones son verdaderas entidades de hypermedia y no simplemente una muestra de resultados desde una consulta, por lo tanto el usuario puede agregar *links*, propiedades o descripciones estáticas adicionales. Los *browsers* por ejemplo, pueden ser implementados como construcciones de composiciones virtuales desde los resultados de una estructura de consulta.

CONCLUSIONES



5. CONCLUSIONES

De acuerdo a nuestra experiencia podemos concluir que consideramos haber logrado el objetivo propuesto. El resultado es **Hypermedia CIP**, una herramienta de hypermedia que brinda la posibilidad de archivar y consultar notas periodísticas de varios medios.

Para obtenerlo se realizó un análisis de la problemática, es decir cómo archivar las notas periodísticas de una forma que sea útil para su consulta y qué tipos de consultas proveer para cubrir todo el espectro posible. Luego se estudió la estructura que serviría de base para la misma, llegando a la conclusión que lo ideal era usar un Hypertexto por las características del mismo. Como las notas pueden pertenecer a varios medios (animación, sonido, imagen y texto) también se tuvo que tener en cuenta la incorporación de los mismos, de lo cual surgió el uso de Hypermedia para soportar Hypermedia CIP.

Inicialmente hicimos una prueba usando el utilitario Linkway de IBM, (herramienta para la creación de Hypertextos elementales), que no cumplía con nuestras expectativas; pero el prototipo logrado en el mismo nos sirvió para resolver ciertas dudas que se nos plantearon durante la creación de nuestra Hypermedia.

Ninguna de las herramientas existentes para la creación de Hypermedia, estudiadas, satisfacían los requerimientos de nuestro objetivo, por lo cual necesitábamos definir un modelo de diseño de Hypermedia, al que llamamos **MDH**.

Consideramos que la filosofía del Hypertexto plantea un traslado inmediato al paradigma de objeto; dado que los nodos, los *links* y los botones (que son los bloques base de los Hypertextos) pueden ser considerados objetos con atributos distintivos, y la navegación en el Hypertexto estaría dada por envío de mensajes entre los objetos mencionados. Por lo cual elegimos al Smalltalk como el lenguaje de programación del prototipo. Si bien habíamos tenido ya una experiencia con el mismo (Proyecto de



Hypermedia CIP

Software), la temática en la implementación cambió con su versión bajo Windows, lo que requirió un tiempo de aprendizaje superior al que habíamos estimado inicialmente.

La experiencia fue positiva ya que nos permitió investigar herramientas de avanzada. Si bien nuestro intento fue un prototipo, creemos que es útil para demostrar que lo que nos está brindando actualmente la informática permite resolver los problemas de una forma más amigable y natural para el usuario.



BIBLIOGRAFIA

- 1 - Hypertexto: Introducción y Perspectiva
Jeff Conklin.

- 2 - Senior Technical Editor, In Depth
Jane Morril Tazelaar

- 3 - Una Visión General
Janet Fiderio

- 4 - HDM - A model for the Design of Hypertext Applications
Franca Garzotto, Paolo Paolini - Dipartimento di Elettronica, Politecnico di Milano, Italy.
Daniel Schwabe - Department of Informatics Pontificia Universidade Católica, Brasil.
(Dic. 1991)

- 5 - KMS - Un sistema distribuido para manejo de conocimiento en las organizaciones.
Donald Mc. Crocken, Robert Akscyn.

- 6 - Incrementando el poder de búsqueda en el Hypertexto por medio de consultas relacionales. N. Yankelovich, N. Meyrowitz, A. Van Dan.



- 7 - Inside Smalltalk - Volumen I
Wilf R. Lalonde - School of Compute Science Carleton University.
John R. Pugh - School of Compute Science Carleton University.

- 8 - Human-Computer Interaction - Interact'90
D. Diaper - Department of Computer Science University of Liverpool; D. Gilmore - Department of Psychology University of Nottingham; G. Cockton -Department of Computing University of Glasgow; B. Shackel -HUSAT Research Institute and Department of Human Sciencies Loughbrough Univeresity of Technology. (1990)

- 9 - The Nested Context Model for Hyperdocuments.
Marco A. Casanova, Luiz Tucherman - Centro Científico Rio, IBM Brasil.
María Julia D. Lima, José L. Rangel Netto, Noemi Rodríguez, Luiz F.G. Soares - Departamento de Informática Pontificia Universidade Católica do RJ (Dic. 1991)

- 10 - Informe Multimedia
Pc Magazine (Mayo 1992)

- 11 - Multimedia - La primera Generación mientras llega Multimedia
Pc Magazine (Mayo 1992)

- 12 - Multimedia - Software de Autor
Pc Magazine (Mayo 1992)

- 13 - When Worlds collide: Demystifying Multimedia.



Jackie Fox - Pc Today (Julio 1991)

14 - Integración de Medios: Authoring Software Makes it Happen
Lori Beckmann - Pc Today (Junio 1991)



Agradecimientos

Agradecemos a Cacho, Claudia, Silvia, María José, Marta, Marisa, Mauricio, Carolina, Mario, Pablo por la colaboración prestada en la realización de este proyecto.