



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL**

Unidad Zacatenco  
**Departamento de Computación**

Modelo y Arquitectura para Exploración Meta-conceptual  
Dinámica en Bases de Datos con Información Biológica

Tesis que presenta:  
**M. en C. Joaquín Sergio Zepeda Hernández**

para obtener el Grado de:  
**Doctor en Ciencias**

en la Especialidad de:  
**Ingeniería Eléctrica**

Opción:  
**Computación**

Director de Tesis: Dr. Sergio Víctor Chapa Vergara

*México, D.F.*

*Junio 2009*



## Resumen

La recuperación e integración de los datos contenidos en una base de datos científica es una tarea no trivial; si adicionalmente la base contiene información biológica el diseño, implementación e interacción involucra una mayor complejidad. Los sistemas Web que mantienen interacción con estas bases, regularmente tienen un interfaces poco intuitivas y predecibles, por lo que el usuario necesita ser un experto para encontrar información específica relacionada a los datos biológicos. Con el surgimiento de bases de datos científicas y públicas en la Web, son necesarios nuevos modelos de interacción, exploración y visualización que permitan una mejor experiencia al usuario.

En esta tesis se propone un nuevo modelo de interacción que denominamos como: *Modelo Meta-conceptual Dinámico (Modelo MD)*, el cual es orientado a sistemas Web que interactúan con bases de datos científicas, e integra diferentes enfoques usados en la exploración de datos como: Consultas visuales, Consultas Dinámicas, Metadatos, Semántica y Mapas conceptuales. La integración es alcanzada a través de capas separadas y cuatro niveles de interacción, que permiten un acceso transparente a la información por medio de una estructura conceptual de los datos y una interface con interacción visual-dinámica.

Adicionalmente se propone una nueva arquitectura genérica e independiente de plataforma que es una extensión de la tradicional cliente-servidor y denominamos como *Arquitectura Meta-conceptual Dinámica (Arquitectura MD)*, las capas que le pertenecen son descritas a detalle para indicar la función específica que tienen en la interacción con los usuarios. Esto con el objetivo de permitir la implementación del modelo MD para su uso en sistemas reales.

El potencial de este nuevo modelo de interacción se presenta a través de un caso de estudio y la implementación de un prototipo sobre una base de datos con información biológica. El prototipo muestra algunas de las ventajas que ofrece este nuevo modelo con respecto a sistemas Web tradicionales, la funcionalidad presentada por el modelo MD promete una nueva generación de sistemas Web más interactivos.



## Abstract

The recovery and integration of data in a scientific database is not a trivial task, if the database contains additional biological information, the design, implementation, and interaction involves more complexity. Web systems that maintain interaction with scientific databases, usually have poor design and unpredictable interfaces, so the user needs to be an expert to find specific information related to biological data. With the emergence of several scientific databases with public information on Web, there is a need of new models for data interaction, exploration and visualization to enable a better user experience.

In this thesis, we propose a new model of interaction for Web systems with scientific databases online, which is called *Dynamic Meta-conceptual Model (DM Model)*. This model is used to develop Web-oriented systems that interact with scientific databases, it integrates various approaches used in the exploration on database, such as Visual query, Dynamic query, Metadata, Semantic and Concept maps. The integration is achieved through separate layers and four interaction levels, which allows transparent access to information through a conceptual structure of data and interface with a dynamic visual interaction.

Additionally, we propose an extension of the traditional client-server architecture to enable the implementation of the DM Model for use in real systems. This new architecture is generic and platform independent, named *Dynamic Meta-conceptual Architecture (DM Architecture)*, the layers of the DM architecture are described in detail to indicate the specific role they have in the interaction with users.

The potential of our approach is shown through a prototype implementation for exploration on a database with biological information. The prototype demonstrates advantages of DM model compared to traditional Web systems, the functionality provided by the DM Model promises a new generation of interactive Web systems.



*A mis padres*

*A mi esposa y a mi hija*

*A mis sobrinas*



## Agradecimientos

A mi asesor el Dr. Sergio V. Chapa Vergara, por el apoyo y animo brindado en el desarrollo de esta tesis.

Al personal de la Colección Nacional Mexicana de Cultivos Microbianos, Profa. Jovita Martínez Cruz, Biólogo Juan Carlos Estrada Mora y técnico Armando Sánchez Chavarría, por su asesoría y apoyo en la parte de microbiología, para la realización de este proyecto de investigación.

A la Dra. Esther Orozco Orozco por todo el apoyo que en su momento tuvo bien a darme, además del reconocimiento a todo su esfuerzo por difundir y apoyar la ciencia en todos sus niveles.

A Charles J. C. Elling Espejel (Charly), por su entusiasmo y ayuda prestada en diversos momentos académicos y de trabajo.

Al Dr. Octavio Martínez de la Vega por todo el conocimiento e ideas que compartió, con el animo de desarrollar nuevas herramientas computacionales orientadas a la biología.

A los revisores por sus valiosos comentarios y propuestas para enriquecer esta tesis, Dra. Sonia Mendoza Chapa, Dra. Xiaou Li, Dr. José Manuel Gómez Soto, Dr. Saul Villa Treviño y Dr. Jorge Buenabad Chavez.

A los doctores que conforman el Departamento de Computación, en especial al Dr. Adriano de Luca Pennachia, Dr. Carlos A. Coello Coello, Dra. Ana María Martínez Enríquez, Dr. Pedro Mejía Alvarez y al Dr. José Guadalupe Rodríguez García.

A Osvaldo Ibáñez Sandoval por las provechosas discusiones y la amistad otorgada durante varios años.

A mis compañeros José Antonio Coria Fernández, René Rodríguez Zamora, Amilcar Meneses Viveros y Mayté Ramírez, por las horas de plática dedicadas al intercambio de conocimiento para mejorar nuestros trabajos.

A las secretarias del Departamento de Computación, Sofía Reza Cruz, Flor Córdova González y Felipa Rosas López, por su ayuda incondicional en su labor administrativa.

Al personal de Servicios Escolares y Becas por el apoyo brindado en todos los trámites requeridos durante mis estudios.

Al CINVESTAV por darme la oportunidad de realizar mis estudios de posgrado y del cual me siento orgulloso de ser egresado.

A CONACYT por todo su apoyo económico otorgado a través de la beca con registro 166542, que permitió la conclusión satisfactoria de mis estudios de doctorado.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	2
1.2. Descripción del Problema . . . . .	3
1.3. Objetivos . . . . .	4
1.4. Organización de la Tesis . . . . .	4
<b>2. Estado del Arte</b>	<b>7</b>
2.1. Ciencia y Computación . . . . .	7
2.1.1. Tendencias hacia el 2020 . . . . .	8
2.1.2. Colaboración interdisciplinaria . . . . .	9
2.2. Bases de Datos en la Investigación . . . . .	12
2.2.1. Bases de datos científicas . . . . .	12
2.2.2. Bases de datos con información biológica . . . . .	14
2.3. Sistemas en la Investigación . . . . .	17
2.3.1. Sistemas con datos científicos . . . . .	17
2.3.2. Sistemas con datos biológicos . . . . .	18
2.4. Comentarios Finales . . . . .	20
<b>3. Interacción y la Web</b>	<b>21</b>
3.1. La Interfaz y los Usuarios . . . . .	21
3.1.1. Consultas visuales . . . . .	22
3.1.2. Limitación en las consultas visuales . . . . .	24
3.1.3. Consultas dinámicas . . . . .	25
3.1.4. Limitaciones en consultas dinámicas . . . . .	27
3.1.5. Usabilidad . . . . .	27
3.1.6. Interfaz multi-nivel . . . . .	29
3.2. La Evolución de la Web . . . . .	31
3.2.1. Ingeniería Web . . . . .	31
3.2.2. Web 1.0 . . . . .	32

3.2.3.	Web 2.0 . . . . .	34
3.2.4.	Web 3.0 . . . . .	36
3.2.5.	Web semántica . . . . .	36
3.3.	Interacción en la Web . . . . .	39
3.3.1.	Limitaciones en un sistema Web . . . . .	39
3.3.2.	Limitantes de una interfaz Web . . . . .	40
3.3.3.	Usabilidad en la Web . . . . .	42
3.3.4.	Aplicaciones RIA . . . . .	44
3.4.	Interacción Visual-Dinámica . . . . .	46
3.4.1.	Ventajas de interacción en sistemas de escritorio . . . . .	46
3.4.2.	Ventajas en interacción Web . . . . .	47
3.4.3.	Modelo de interacción visual-dinámica . . . . .	48
3.5.	Comentarios Finales . . . . .	50
<b>4.</b>	<b>Semántica y Metadatos</b>	<b>51</b>
4.1.	Semántica y Datos . . . . .	51
4.1.1.	Semántica en sistemas de información . . . . .	52
4.1.2.	Limitación semántica en sistemas de información . . . . .	53
4.1.3.	Semántica . . . . .	54
4.1.4.	Semántica léxica . . . . .	55
4.1.5.	Mapa conceptual . . . . .	59
4.2.	Metadatos . . . . .	61
4.2.1.	Definición de metadato . . . . .	61
4.2.2.	Uso de los metadatos . . . . .	63
4.2.3.	Metadatos como diccionario de datos . . . . .	64
4.2.4.	Metadatos en la Web . . . . .	65
4.2.5.	Limitaciones de los metadatos . . . . .	66
4.3.	Interacción con Metadatos Conceptuales . . . . .	67
4.3.1.	Relación semántica en mapas conceptuales . . . . .	67
4.3.2.	Metadatos conceptuales . . . . .	69
4.3.3.	Modelo basado en metadatos conceptuales . . . . .	70
4.4.	Comentarios Finales . . . . .	71
<b>5.</b>	<b>Modelo Meta-conceptual Dinámico</b>	<b>73</b>
5.1.	Presentación del Modelo . . . . .	73
5.1.1.	Modelo Meta-conceptual Dinámico . . . . .	74
5.1.2.	Capas en el modelo MD . . . . .	76
5.1.3.	Niveles en el modelo MD . . . . .	79
5.2.	Conceptualizando los Datos . . . . .	81
5.2.1.	Base de datos científica . . . . .	81

## ÍNDICE GENERAL

---

5.2.2. Mapa conceptual semántico . . . . .	83
5.2.3. Integración meta-conceptual . . . . .	85
5.3. Comentarios Finales . . . . .	86
<b>6. Arquitectura Meta-conceptual Dinámica</b>	<b>87</b>
6.1. Descripción de la Arquitectura . . . . .	87
6.2. Técnica de Código No Invasivo . . . . .	89
6.3. Interfaz Visual . . . . .	91
6.4. Interacción Dinámica . . . . .	93
6.5. Consultas Automáticas . . . . .	97
6.6. Comentarios Finales . . . . .	98
<b>7. Prototipo</b>	<b>99</b>
7.1. Descripción del Caso de Estudio . . . . .	99
7.1.1. Colección microbiana . . . . .	100
7.1.2. Base de datos microbiana CDBB500 . . . . .	101
7.2. Mapa de Exploración . . . . .	104
7.2.1. Mapa semántico de exploración . . . . .	104
7.2.2. Mapa meta-conceptual . . . . .	108
7.3. Prototipo . . . . .	110
7.3.1. Diseño de interfaz . . . . .	110
7.3.2. Interacción en el prototipo . . . . .	112
7.4. Resultados . . . . .	115
7.4.1. Comunicación asíncrona de la arquitectura MD . . . . .	115
7.4.2. Formato para recuperación de datos . . . . .	119
7.5. Comentarios Finales . . . . .	123
<b>8. Conclusiones</b>	<b>125</b>
8.1. Discusión . . . . .	125
8.2. Conclusiones . . . . .	128
8.3. Contribuciones . . . . .	129
8.4. Implicaciones . . . . .	130
8.5. Limitaciones . . . . .	131
8.6. Perspectivas . . . . .	132

*ÍNDICE GENERAL*

---

# Índice de figuras

2.1. Arquitectura von Neumann . . . . .	11
2.2. Gráfica de nucleótidos . . . . .	15
3.1. Sistema Lida . . . . .	23
3.2. Consultas dinámicas . . . . .	25
3.3. Diseño multi-capa . . . . .	29
3.4. La Web 2.0 . . . . .	35
3.5. Principio KISS . . . . .	43
3.6. Aplicación RIA . . . . .	45
3.7. Interacción visual-dinámica . . . . .	49
4.1. Significado como concepto . . . . .	55
4.2. Relación Hiponimia-hiperominia . . . . .	57
4.3. Relación Meronimia-holonimia . . . . .	58
4.4. Mapa conceptual . . . . .	61
4.5. Folksonomía . . . . .	65
4.6. Mapa conceptual semántico . . . . .	68
4.7. Acceso transparente de datos . . . . .	69
4.8. Modelo basado en metadatos conceptuales . . . . .	71
5.1. Modelo Meta-conceptual Dinámico . . . . .	75
5.2. Esquemas para bases de datos . . . . .	81
5.3. Esquema conceptual multinivel . . . . .	83
5.4. Mapa conceptual semántico . . . . .	84
5.5. Enlace Meta-conceptual . . . . .	85
6.1. Arquitectura Meta-conceptual Dinámica . . . . .	89
6.2. Estructura DOM . . . . .	91
6.3. Interacción en la Arquitectura MD . . . . .	95

## ÍNDICE DE FIGURAS

---

7.1. Vista del Sistema Micro500 . . . . .	101
7.2. Esquema conceptual de la base de datos CDBB500 . . . . .	103
7.3. Mapa general de exploración . . . . .	105
7.4. Concepto cepa . . . . .	106
7.5. Mapa conceptual de morfología . . . . .	107
7.6. Mapa meta-conceptual de exploración . . . . .	108
7.7. Referencias al esquema lógico . . . . .	109
7.8. Interfaz multinivel . . . . .	111
7.9. Vista inicial exploración micro500 . . . . .	113
7.10. Vista del nivel 2 . . . . .	114
7.11. Vista del nivel 3 . . . . .	115
7.12. Comunicación síncrona . . . . .	117
7.13. Recuperación de datos . . . . .	118
7.14. Comunicación asíncrona . . . . .	119
7.15. Etiquetas generadas . . . . .	120
7.16. Datos totales en la consulta . . . . .	121
7.17. Transporte total . . . . .	122

# Capítulo 1

## Introducción

Actualmente existe una inmensa cantidad de datos obtenida por los científicos, pero también existe una fuerte necesidad de contar con nuevas herramientas computacionales que permitan administrar, explorar, visualizar y procesar estos datos. Los datos científicos surgen de experimentos y contienen un cierto conocimiento relacionado a un área científica. Este tipo de datos muchas veces necesitan de un procesamiento para poder ser explotados; otras veces, necesitan una manera eficaz para acceder a ellos de manera puntual y poder compararlos con otros o simplemente para validar y verificar su calidad.

Los datos científicos tienen características muy diferentes a la de datos de negocio, sobre los cuales fueron diseñados los administradores de datos, mejor conocidos como DBMS (DataBase Manager Systems). Los datos científicos cuentan principalmente con tres vertientes que son: heterogeneidad, interpretación y significado; estas tres características hacen más difícil el diseño y desarrollo de un esquema conceptual y un esquema lógico, que ayuden a crear una base de datos que los contenga. Cuando estos esquemas logran construirse, la mayoría de las veces cuentan con un alto grado de complejidad relacional, que hace muy difícil el desarrollo de sistemas que permitan alcanzar una rica interacción con los datos; la implementación de sistemas con esta complejidad en la base de datos, se convierte en una tarea no trivial. Debido a esta situación han surgido diferentes alternativas de sistemas orientados a la exploración, que tratan de buscar una solución para poder obtener un acceso transparente a todos los datos contenidos en una base de datos con una alta complejidad relacional. Este tipo de sistemas de exploración tratan de esconder toda la complejidad sintáctica y semántica de un lenguaje de consulta, para acceder de manera puntual a datos específicos; motivando

el surgimiento de diversas líneas de investigación como lo son: las consultas visuales (*visual query*) y las consultas dinámicas (*dynamic query*). Las consultas visuales y dinámicas son algunas de las mejores soluciones encontradas, pero han sido especialmente orientadas a sistemas de escritorio; por lo que su ejecución es realizada en una máquina local donde se encuentra instalado el propio sistema.

La implementación de estos enfoques en sistemas orientados a la Web es imposible, debido a las múltiples restricciones que impone la arquitectura cliente-servidor tradicional como: falta de estados, uso obligado de navegador, comunicación en red, interacción sobre páginas Web, etc.

Estas restricciones han motivado otras líneas de investigación como el uso de metadatos y semántica<sup>1</sup>. Todos estos enfoques utilizados en sistemas orientados a la exploración de datos, no ofrecen una buena solución cuando tratamos con bases de datos científicas que contienen información biológica. El problema surge debido a que los datos biológicos tienen una amplia heterogeneidad y con una rica semántica inmersa en cada dato; esto genera una necesidad por abrir nuevas líneas de investigación en la *Ingeniería Web*, que permitan encontrar soluciones acordes a la exploración y acceso de datos biológicos.

Esta tesis está enfocada en encontrar nuevas formas de acceso y exploración de datos biológicos científicos contenidos en una base de datos, sin que el usuario requiera de: aprendizaje de un lenguaje de consulta, conocimiento previo de la estructura interna y lógica de los datos o entrenamiento especial en el manejo del software. Esta investigación presenta un interesante desafío debido a que existen muy pocas herramientas orientadas en este sentido. Consecuentemente no existen modelos y arquitecturas que se centren en la resolución de este problema en específico.

## 1.1. Motivación

La motivación inicial en este proyecto de investigación surge del llamado y el interés de diversos científicos relacionados al área biológica, que desean obtener un mejor acercamiento con científicos en computación. La complejidad que genera la administración de datos científicos con información biológica y la semántica que contienen, plantea una fuerte necesidad por nuevos modelos de interacción, esquemas de bases de datos, arquitecturas y sistemas que ayuden a la comprensión de la información biológica.

---

<sup>1</sup>Nos referimos al uso de la semántica a través de metadatos, para describir el contenido de archivos multimedia.

Por otro lado, la fuerte tendencia por migrar todo tipo de aplicaciones a la Web, motiva una mayor investigación para encontrar soluciones a todas estas necesidades en ambientes Web, para obtener todas las ventajas que permite este medio de comunicación. Descripciones más amplias de estas necesidades, podemos encontrarlas en diversos artículos científicos como: [1], [2], [3], [4], [5]. El desafío por obtener herramientas computacionales, que permitan una exploración de datos en bases de datos con información científica y biológica a través de la Web, representa una interesante proyecto de investigación en el área de la *Ingeniería Web y Acceso a bases de datos*. La necesidad de nuevos modelos, arquitecturas, técnicas y metodologías, computacionales que sean orientadas a obtener soluciones en la exploración de datos científicos biológicos, motiva nuestro interés sobre este problema en específico.

### 1.2. Descripción del Problema

El incremento exponencial de información biológica necesita de nuevas formas de interacción entre usuarios y datos, debido a la fuerte tendencia actual de poner a disposición bases de datos con información biológica en la Web. La exploración sobre estas bases de datos, es una de las principales problemáticas a la cual se enfrentan los científicos que pertenecen al área biológica. Soluciones tradicionales como el uso de formularios en las páginas Web, mantienen consultas restringidas y la navegación por la base de datos en busca de datos de interés, resulta ser una tarea imposible para los usuarios; por este motivo, el uso de formularios tradicionales como método de acceso a los datos no siempre es una alternativa viable y efectiva para sus diferentes necesidades. Tratar de obtener consultas especializadas con este enfoque sobre una base de datos con complejidad relacional, requiere de un conocimiento de toda la estructura interna de la base de datos, aprendizaje y comprensión de un lenguaje de consulta, además de evitar cometer errores sintácticos y semánticos. La inexperiencia con los lenguajes de consulta puede ocasionar que un usuario construya algunas docenas de consultas, antes de poder encontrar la información que verdaderamente requiere. La creciente necesidad de encontrar nuevas soluciones como: modelos y arquitecturas orientadas a la exploración e interacción en bases de datos científicas basados en la Web, es uno de los problemas esenciales que actualmente no cuentan con soluciones efectivas. Este problema no es sencillo de resolver, ya que esta acompañado de una serie de subproblemas que también necesitan ser resueltos como:

- Encontrar una alternativa para que usuarios inexpertos eviten el aprendizaje y entrenamiento sobre un lenguaje de consulta.
- Encontrar un solución alternativa al uso de formularios tradicionales, para el acceso a la información.
- Encontrar una manera que permita alcanzar un acceso transparente a datos especializados en una base de datos.
- Buscar una nueva manera de interacción entre el usuario y los datos.
- Encontrar una diseño controlado de acceso a los datos en la interfaz de un sistema Web.
- Generar la metodología que permita implementar las soluciones en un sistema real.

### 1.3. Objetivos

Los objetivos generales en este proyecto de investigación son:

- *Nuevo modelo para exploración de datos.*- Crear un nuevo modelo que permita integrar la semántica conceptual inmersa en una base de datos, con una interfaz con interacción visual y dinámica; que a su vez habilite una exploración y navegación de los datos de manera transparente.
- *Nueva arquitectura genérica.*- Construir un nueva arquitectura que permita establecer todos los componentes y capas que se necesitan para el desarrollo de sistemas Web, basados en el nuevo modelo creado; con las propiedades de ser genérica, abierta e independiente de plataforma para su uso en diversas aplicaciones.
- *Prototipo para la exploración de datos biológicos.*- Obtener un nuevo prototipo basado en el nuevo modelo de interacción, que permite observar las ventajas que ofrece la nueva arquitectura con respecto a las implementaciones tradicionales.

### 1.4. Organización de la Tesis

A continuación se presenta la manera en que está organizada la presente tesis para obtener una mejor comprensión de la misma.

El capítulo 2, está orientado a presentar una panorámica general sobre el trabajo interdisciplinario entre computación y biología. Después se analizará la esencia de los datos científicos y los datos científicos biológicos. Por último, se presentarán algunos de los conceptos utilizados en la implementación de sistemas orientados a la exploración y recuperación de datos.

En el capítulo 3, se presentan los distintos enfoques usados para enriquecer la interacción de los usuarios con un sistema de información. Después nos enfocamos en la esencia de los sistemas Web y describimos el área de la ingeniería Web, así como la evolución que tiene la Web actualmente. Al final de este capítulo, se obtiene un nuevo modelo básico de interacción Web, al que denominamos como *Modelo para la interacción visual-dinámica*.

En el capítulo 4, nos enfocamos en la esencia estructural de los datos y la manera en que se obtiene el encapsulamiento de la semántica a través de conceptos; así como la obtención de un mapa conceptual que sirve para la exploración de los datos. En esta parte se obtiene otro modelo básico que denominamos con el nombre de *Modelo de interacción basado en metadatos conceptuales*, que permite enlazar la estructuración semántica de datos, con el esquema lógico de una base de datos.

En el capítulo 5, se presenta el modelo final propuesto y que denominamos como *Modelo Meta-conceptual Dinámico (MD)*, el cual unifica a los dos diferentes modelos básicos anteriores y permite una funcionalidad basada en una interacción visual-dinámica, a través de mapas conceptuales semánticos en la interfaz. También se presenta la manera en que la estructuración de datos debe ser llevada a cabo para su incorporación al nuevo modelo.

El capítulo 6, presenta la arquitectura que hace posible que el modelo MD, sea implementado en sistemas Web reales. La descripción de la arquitectura muestra todas las partes internas, que permiten obtener la funcionalidad requerida.

En el capítulo 7, se presenta un caso de estudio que permite poner en práctica al modelo y la arquitectura MD. Se muestra cómo el diseño de la interfaz influye en la funcionalidad del sistema. En la parte final se obtienen algunos resultados por el proceso de recuperación de datos.

Por último en el capítulo 8, se presenta una breve discusión sobre otros sistemas de exploración y el que se obtuvo en el prototipo. Por último se presentan las conclusiones finales y las perspectivas de trabajo futuro.

*CAPÍTULO 1. INTRODUCCIÓN*

---

## Capítulo 2

# Estado del Arte

El uso de computadoras en los experimentos científicos está generando una enorme cantidad de datos, que los investigadores necesitan administrar y explotar de manera eficiente. Un desafío importante en bases de datos con datos científicos, es permitir un acceso transparente y eficiente de los datos para poder realizar una útil explotación de los mismos. Alcanzar este objetivo no es sencillo, debido a las características propias de los datos científicos. El capítulo está orientado en mostrar parte de estas características, así como los conceptos utilizados en sistemas orientados al acceso de datos.

El estado del arte es amplio en este sentido y sólo se muestran los conceptos más relevantes y de mayor impacto. El inicio de este capítulo comienza con el llamado que hacen diversos científicos de diferentes áreas de investigación, para tratar de obtener un acercamiento con investigadores en computación que puedan ofrecer nuevas alternativas en software, que ayuden a solventar algunas de las dificultades que contiene la administración y exploración de datos científicos.

### 2.1. Ciencia y Computación

Esta sección presenta una discusión general sobre cómo las herramientas de software han modificado la manera de hacer ciencia; ya que ahora se está obteniendo una enorme cantidad de datos, pero existe una fuerte necesidad de contar con nuevas herramientas que permitan el procesamiento de dichos datos. Esto ha motivado que algunos investigadores del área biológica, inviten a científicos en ciencias de la computación a trabajar en colaboración.

La primera parte de esta sección muestra el impacto que tiene la computación en la ciencia; la segunda parte presenta el creciente interés por establecer un fuerte trabajo interdisciplinario entre computación y biología.

### 2.1.1. Tendencias hacia el 2020

En julio del 2005, se reunieron un grupo de cuarenta reconocidos científicos de doce diferentes nacionalidades y diversas áreas de investigación como: biología, física, medicina, bioquímica, astronomía, genética, matemáticas y ciencias de la computación. En dicha reunión se discutió sobre el gran impacto que tiene la computación actualmente y deberá tener hacia el año 2020 en la ciencia, principalmente en áreas relacionadas con la biología.

La fuerte relación entre la computación y la ciencia afecta la manera misma de hacer ciencia en todas sus facetas, áreas y ámbitos. La inclusión de la computación en el proceso científico, ha hecho posible nuevas clases de experimentos que no podían ser posibles una década atrás. Estas nuevas clases de experimentos generan nuevas clases y tipos de datos que incrementan exponencialmente su complejidad y volumen [1].

Es fácil observar que las herramientas de software que tradicionalmente se orientan a datos de negocios, no cumplen con algunas de las expectativas de los científicos, generando una fuerte necesidad por desarrollar nuevo software orientado a administrar, explorar, consultar, procesar, analizar, describir, visualizar, filtrar y compartir los datos científicos. También es notable la necesidad por investigar sobre nuevas infraestructuras, que permitan explotar toda esta información contenida en los datos científicos de manera efectiva y constructiva, tal como lo menciona Buneman, et al. en [6]. Para cubrir estas necesidades, son necesarios nuevos modelos y arquitecturas que permitan desarrollar nuevas clases de aplicaciones orientadas hacia este rubro.

Nuevas y novedosas herramientas conceptuales, semánticas y tecnológicas, son la nueva tendencia a seguir, teniendo a corto plazo un profundo y radical impacto en la ciencia, especialmente en las ciencias biológicas.

En [1] podemos encontrar una especial discusión sobre este tema; también encontramos otros puntos de interés y relevantes para nuestro proyecto de investigación, los cuales son descritos como principales áreas de interés hacia el año 2020 entre los que se encuentran:

- Incremento en la semántica de datos.
- Amplio uso de metadatos para hacerlos extensamente disponibles y usables.
- Recuperación y búsqueda de información para novedosos tipos de datos.
- Surgimiento de nuevas clases de herramientas computacionales amigables, para los científicos.
- Integración de publicaciones científicas con bases de datos del área de investigación.
- Nuevas representaciones de sistemas biológicos para ser transportados hacia esquemas lógicos.

Algunos de estos puntos ya están generando nuevas investigaciones alrededor del mundo y vislumbran la tendencia mencionada por Shapiro, et al. en [7], donde menciona que “Las ciencias de la computación llegarán a ser fundamentales en la ciencia, particularmente en las ciencias naturales, así como las matemáticas han llegado a ser en la ciencia, en especial en las ciencias físicas”.

Con esta visión, diversos países están realizando cambios radicales en la formación de sus nuevos científicos, los cuales están siendo preparados para desarrollar trabajo interdisciplinario con una cierta formación común en ciencias de la computación, permitiendo a mediano plazo el surgimiento de una nueva ciencia con una nueva manera de hacer investigación, con novedosos institutos de investigación diferentes a los tradicionales.

### 2.1.2. Colaboración interdisciplinaria

La colaboración está volviéndose fundamental en la presente investigación científica, generando resultados exitosos cuando esta es llevada adecuadamente. Por esta razón, cada vez existe un mayor interés de diversos científicos por lograr un mayor acercamiento entre biología y ciencias de la computación. Algunos investigadores resaltan de manera especial este interés como: Jaques Cohen en [3], A. Finkelstein en [4] y Junhyong Kim en [8]. El acercamiento entre la computación y biología permite vislumbrar nuevas facetas de investigación que contribuirán de manera sustancial a las dos disciplinas; este es el principal interés de animar y motivar trabajos conjuntos entre estas dos ciencias, tal como se señala en [1], [2], [3], [4], [5], [8], [9].

Este acercamiento empieza a consolidarse en el área de la bioinformática, aunque ha sido principalmente orientada hacia la comparación de secuencias genómicas; pero la computación no sólo debe enfocarse en resolver este tipo de problemas, es necesario impulsar nuevas propuestas en todas las diversidades que ofrece la manipulación de información biológica como: nuevos modelos de datos, modelos de interacción, arquitecturas, propuestas de procesamiento, simulaciones, etc.

Por otro lado, la computación también puede ayudar a explicar diversos comportamientos biológicos tal como lo plantean Brent y Bruck en [2], donde muestran las similitudes entre un sistema biológico y la arquitectura de von Neumann, como se muestra en la figura 2.1. El trabajo conjunto tiene la tendencia de cambiar a la biología y a la computación misma en un futuro no muy lejano, para visualizar que: “*La relación entre computación y biología llegará a ser tan fundamental, como las matemáticas han llegado a ser en la física*”, como se afirma en [3], [4], [10], [1], [5], [11].

El trabajo conjunto entre científicos es descrito por Carlis de dos maneras:

1. Multidisciplinario.-*Es una coalición separada de científicos, trabajando cada uno en forma separada e independiente, en una parte específica de un problema (Carlis en [9]).*
2. Interdisciplinario.-*Es todo un equipo de científicos de diferentes disciplinas, trabajando todos en conjunto para resolver un determinado problema (Carlis en [9]).*

Carlis muestra al trabajo multidisciplinario como un trabajo aislado, dedicado y orientado solo a una pequeña parte del problema, olvidando ciertos aspectos relacionados a un todo. Esto puede conducir a errores comunes por investigadores en computación cuando trabajan de manera aislada ciertos problemas relacionados a la biología, él lo menciona como:

*“Correr complejos y sofisticados algoritmos<sup>1</sup> en datos biológicos; no tienen ningún valor y significan nada, si los datos obtenidos son irrelevantes y no tienen un buen sustento biológico (Carlis, [9])”.*

Con la afirmación de Carlis, podemos observar la necesidad de un sólido trabajo interdisciplinario muy cercano entre estas dos áreas y otras más.

---

<sup>1</sup>El autor se enfoca principalmente en algoritmos de computación desarrollados para el área genómica y trabajan sobre secuencias de nucleótidos, aminoácidos, rna, etc.

Por lo que tratar de explicar algunos fenómenos biológicos es una ambiciosa tarea que requiere la ayuda de diversas áreas como: biología, bioquímica, química, medicina, genética, matemáticas, física, entre otras más. Con el aporte de cada una de las diferentes áreas es necesario revisar, corregir y crear nuevas técnicas para objetivamente interpretar, comprender y explicar todo tipo de fenómenos físicos, fisiológicos, moleculares y morfológicos, como es observado en [12], [3], [4], [13], [14].

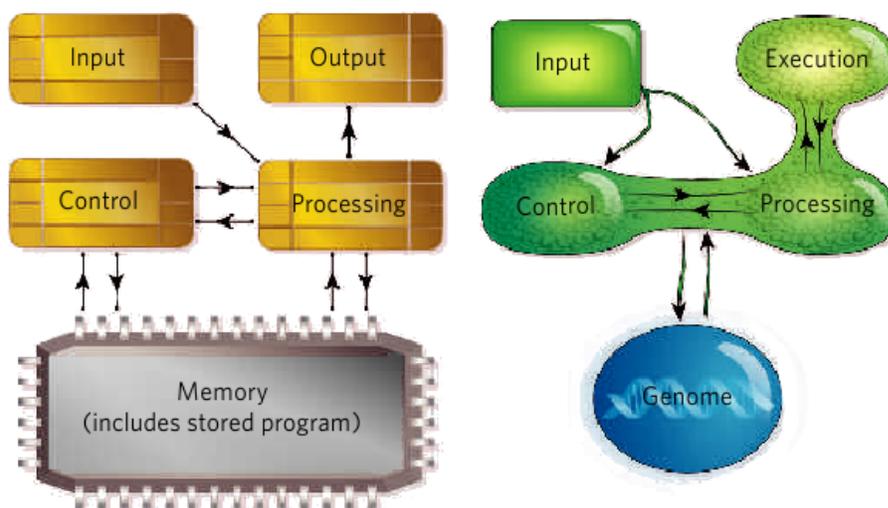


Figura 2.1: Diferencias y similitudes entra la arquitectura de von Neumann y los sistemas biológicos (figura original de Brent, et al. [2]).

Es importante señalar que el trabajo interdisciplinario ayuda a enfocar, motivar y retroalimentar hacia nuevas direcciones; donde la investigación aislada no puede llegar. Esto nos dice que nuevos descubrimientos e invenciones pueden llegar a ser realizados por la conjunta colaboración; además de buscar respuestas más argumentadas, más sólidas y con diferentes puntos de vista, para así obtener resultados que quizás nunca se obtendrían trabajando de manera separada, como se discute en [15], [4], [9], [13].

Chonacky et al. en [16], observa que las contribuciones obtenidas en un trabajo interdisciplinario, pueden ser aplicadas muchas veces a otras disciplinas, las cuales pueden encontrar diferentes representaciones o utilidades, haciéndolas más comprensibles y relevantes.

## 2.2. Bases de Datos en la Investigación

El uso masivo de computadoras en los experimentos y la investigación científica, ya ha ocasionado un incremento exponencial en la generación y volumen de datos. El almacenamiento de los vastos volúmenes de datos recolectados depende en mayor medida de Sistemas Administradores de Datos (DataBase Management System DBMS) para guardar, organizar y administrar dichos datos. Esta sección describe algunas características propias de los datos científicos en general y de los datos científicos que contienen información biológica.

### 2.2.1. Bases de datos científicas

El sistemático uso de bases de datos ha llegado a ser una parte integral, fundamental y una herramienta natural del proceso científico, como se describe en [17], [16], [18]. Muchos experimentos obtienen importantes cantidades de datos sin procesar y son guardados masivamente en bases de datos antes de poder ser estudiados.

Muggleton en [19] y Szalay et al. en [13], describen como: “*Un inmenso volumen de datos, puede generar un decremento en la comprensión de los mismos*”; este decremento es mayor cuando existen pocas herramientas computacionales para que los datos se puedan explorar, filtrar, visualizar, analizar, procesar y compararse con otros, de una manera apropiada.

En la figura 2.2 podemos observar el crecimiento exponencial de datos científicos en genómica; la gráfica ilustra la necesidad de nuevas herramientas que permitan realizar todas las tareas antes mencionadas, además de manejar grandes cantidades de datos y con nuevas formas para acceder a la información. Adicionalmente su conceptualización tiene una mayor complejidad inmersa, porque este tipo de datos contienen mayor ruido y requieren de un procesamiento especial; así como llamados rápidos para exploración, filtrado, visualización, extracción, transformación, entre otras diversas operaciones no convencionales, las cuales son referidas en [20], [21], [22].

También algunas veces, los datos científicos necesitan almacenar junto con ellos un contexto, condiciones específicas de obtención de datos o un historial con múltiples versiones de ellos, entre muchas otras características. Por todo esto, rígidos esquemas conceptuales pueden ser inefficientes, tener poca o nula utilidad, para una enriquecedora explotación científica de los datos como lo describe Zhang et al. en [18].

Esta descripción nos permite observar por qué los datos científicos han sido identificados como el mayor desafío para la tecnología de bases de datos, algunos de estos desafíos y problemas se deben a que los presentes DBMS son generalmente diseñados para aplicaciones de negocio.

Las tradicionales bases de datos orientadas a negocio son relativamente fáciles de conceptualizar, porque las relaciones entre diferentes entidades son bien entendidas y conocidas, pero en bases de datos científicas regularmente ese no es el caso, como se señala en [18], [22], [21].

El desafío de trabajar con datos científicos empieza desde el momento de tratar de obtener un esquema conceptual, que involucre muy diversos rasgos esenciales en este tipo de datos, como el contexto en que se obtuvieron los datos, así como la descripción de la semántica inmersa en ellos.

Para ejemplificar parte de esta problemática, veamos la tabla 2.1, donde podemos observar algunas de las características que deben tenerse en cuenta cuando trabajamos con datos científicos, lo cual los hace muy diferentes a datos orientados a negocio.

Datos	negocio	científicos
Ruido		X
Filtrado		X
Contexto		X
Estáticos	X	X
Dinámicos		X
Visualización	X	X
Exploración		X
Semántica		X

Cuadro 2.1: Tabla que muestra las diferencias entre los datos de negocio y los datos científicos.

Una vez mostradas algunas diferencias entre los datos de negocios y los datos científicos, podemos observar por qué los DBMS orientados a negocio no tienen el soporte suficiente para administrar la información científica, como se señala en [6] y [23].

Algunas soluciones para datos científicos han sido orientadas a *Almacenes de Datos (Data Warehouse)*, pero muchas de las técnicas utilizadas en ellos, no son transferibles a ambientes científicos con datos dinámicos, porque estos son esencialmente de naturaleza estática, careciendo del dinamismo necesario que necesitan este tipo de datos (vease [22] y [24]).

Por otro lado el impacto de la Web para una completa comunicación científica ha sido subestimado, ya que gran parte del enfoque hacia la Web ha sido orientado a la administración de artículos científicos y al acceso de algunas bases de datos con información científica en la Web.

El uso de bases de datos científicas abiertas a través de la Web empieza a motivar el intercambio de datos entre distintos laboratorios alrededor del mundo, para ser una herramienta común de comunicación, publicación y colaboración entre distintos grupos interdisciplinarios para hacer ciencia, tal como lo hace notar Shneiderman en [25], [26], [27].

Las bases de datos científicas con información pública en la Web, necesitan de nuevos modelos de interacción y exploración de este tipo de datos, para alentar una mayor colaboración científica a nivel mundial y reducir las barreras existentes entre los datos y los científicos, permitiendo que los datos puedan ser interpretados correctamente y ocuparlos para producir nuevo conocimiento.

### 2.2.2. Bases de datos con información biológica

Es innegable que entre las ciencias, la biología juega un papel muy importante en el siglo veinte y es muy probable que adquiera mayor relevancia de la prevista en los años siguientes. Uno de los principales factores de su importancia es la relación estrecha con la medicina, ya que algunos progresos innovadores en medicina, dependen del entendimiento y esclarecimiento de algunos de los misterios no resueltos por científicos en biología [3].

En investigación biológica, todos los datos que generan los diversos componentes que pertenecen a un sistema biológico, son esenciales para poder generar y construir modelos comprensibles. Existe una evidente y fuerte necesidad de investigar en la construcción de nuevos sistemas computacionales, que involucren todos estos aspectos y permitan entender y encontrar nuevas facetas biológicas desconocidas.

Kitano discute este tema en [12] y lo explica de una manera más detallada haciendo una analogía entre un avión y un sistema biológico; en esta analogía explica por qué no se debe concentrar toda la atención en sólo una pequeña parte y señala que es necesario tener en cuenta un todo, para así generar un modelo comprensible que pueda explicar un comportamiento biológico.

Tratando de explicar diversos comportamientos biológicos, los científicos en esta área han reunido una enorme cantidad de datos. Un ejemplo puede observarse en la figura 2.2, donde se muestra una inmensa cantidad de datos relacionados a nucleótidos. Todos estos datos son valiosos y es necesario contrastarlos con una diversa cantidad de datos homogéneos y heterogéneos, para poder encontrar y generar nuevo conocimiento útil. Al tratar de llevar esto a la práctica, es evidente la falta de herramientas que puedan explotar de manera eficaz todos estos datos.

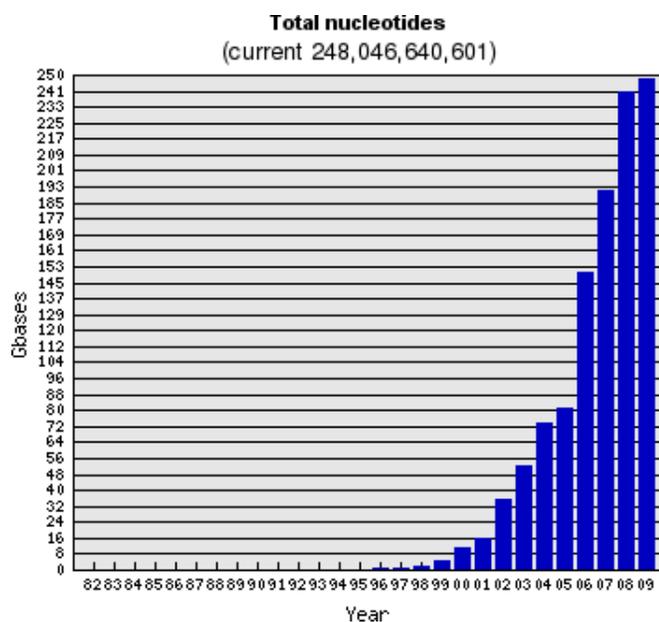


Figura 2.2: Gráfica que muestra el comportamiento exponencial que tiene la adquisición de nucleótidos, reporte generado por EMBL (European Molecular Biology Laboratory).

En la parte computacional, el desarrollo de herramientas o sistemas de computación que permitan administrar e interactuar con la diversa información biológica, involucra una tarea nada trivial; porque además de las diferentes características mostradas en la tabla 2.1, es necesario adicionar otras muy propias de este tipo de datos, las cuales hacen más difícil una flexible representación, especificación, organización, conceptualización y mantenimiento.

Algunas de estas características son:

- Semántica inmersa
- Heterogeneidad en los datos
- Dependencia de datos
- Variabilidad de valores
- Complejidad sintáctica
- Captura de contexto
- Variabilidad contextual
- Dependencia contextual

Philippi en [28], presenta algunas de estas particularidades de este tipo de información y también muestra algunas adicionales como los problemas técnicos y de definición de políticas con respecto a la seguridad y confidencialidad de la información.

Philippi resalta el problema semántico (ver [28]), por la interpretación y significado que puede tener un proceso biológico; y resalta cómo un mismo dato puede ser descrito con diferentes términos y enfoques, haciendo más difícil identificar diversas relaciones, hechos o reglas existentes entre conjuntos de datos. Estos hechos o reglas son difíciles de observar, debido a la inflexible estructuración que presentan los sistemas actuales y los rígidos diseños bajo los cuales fueron construidos.

Otro problema con esta información, radica en que los archivos planos son el estándar para el intercambio de datos biológicos, generando diversos problemas de acceso a la información, ya que la mayoría de las veces la información no puede ser accesada de manera puntual. Ejemplos de sistemas que administran su información de esta manera pueden verse en [29], [30], [28] y en [32]. Por último, es necesario puntualizar el creciente interés que existe en poner toda esta información a disposición del público en la Web. Por consiguiente se están buscando nuevas maneras de poder llevar a cabo esta tarea de la manera más eficiente. Sistemas con esta orientación pueden verse en [29], [31], [32], [33], [34].

## 2.3. Sistemas en la Investigación

Una vez discutidas algunas de las características que hacen diferentes a los datos de negocio de los datos científicos, ahora es necesario conocer la manera en que los sistemas mantienen la interacción entre los usuarios y la base de datos. En esta sección se presentan de una manera muy general, algunos de los conceptos utilizados en la implementación de sistemas con una amplia variedad de información científica.

### 2.3.1. Sistemas con datos científicos

El diseño e implementación de las bases de datos científicas es una difícil tarea; el desarrollo de sistemas que permitan un eficaz acceso a este tipo de datos, también resulta ser una tarea complicada. El tema de interés en esta sección, es la manera en que el sistema tiene acceso a los datos en la base de datos. Existen diferentes formas de hacerlo y las de mayor interés para nuestro propósito son:

- Metadatos
- Semántica
- Interfaz visual

A continuación describimos cada una de ellas:

1. *Uso de Metadatos.*- En sistemas basados en Metadatos (datos sobre datos), los datos son etiquetados y esta etiqueta funciona como un metadato. Luego estos metadatos funcionan como índices de acceso y localización de datos o archivos. Los metadatos describen en la mayoría de los casos: una característica, el contenido de un dato o archivo. Sistemas con este enfoque pueden verse en [35], [36], [37], [38].
2. *Uso de Semántica.*- Estos sistemas se basan en el uso de ontologías para caracterizar los datos y sus relaciones con otros similares. Esto permite eliminar ciertas ambigüedades para obtener descripciones más consistentes de los datos o archivos. Las estructuras formadas pueden generar diversos tipos de relaciones que son validadas por expertos del área de estudio. El significado inmerso en el uso de la semántica es orientado hacia lenguaje natural. Ejemplos de sistemas de este tipo pueden verse en [39], [40].

3. *Uso de Interfaz Visual.*- La interfaz visual es un potente medio que ayuda a eliminar complejidad de uso en ambientes computacionales; a través de la interfaz podemos interactuar fácilmente con los sistemas y realizar diversos procesamientos de la información. Sistemas con esta orientación requieren de capas intermedias, que permitan percibir las acciones del usuario en la interfaz para poder generar una respuesta automática y dinámica según corresponda. Algunos sistemas tratan de mapear los datos de la base de datos en gráficos que permitan un tipo de interacción, ejemplos de estos sistemas pueden verse en [41], [42]. Otro enfoque utilizado es el uso de lenguajes visuales de consulta, una de las desventajas de este enfoque es que el usuario necesita aprender a utilizar el lenguaje visual, ver [43], [44]. Uno de los enfoques más interesantes es el de las consultas dinámicas (*Dynamic queries*), concepto introducido por Ben Shneiderman en 1992 en [45]. Shneiderman plantea hacer un amplio uso de elementos gráficos como barras de deslizamiento y botones, para realizar consultas de manera interactiva, ver [46], [47].

Cada uno de estos enfoques contiene ciertas ventajas unos respecto de otros, dado su uso y la manera en que son implementados en sistemas reales. Cada uno de estos enfoques es de gran interés en nuestra investigación y son mostrados con más detalle en los capítulos 3 y 4, porque a partir de una descripción más amplia se propone una nueva manera de utilizarlos.

### 2.3.2. Sistemas con datos biológicos

El gran impacto que tuvo el anuncio de la secuenciación del genoma humano, por un grupo de científicos (ver [48], [49]), causó mucho interés en la comunidad científica. Consecuentemente el anuncio motivo una carrera por la obtención de nuevos datos genómicos de diferentes seres vivos, desde seres microscópicos hasta mamíferos.

Todo esto generó una urgencia por el desarrollo de nuevos sistemas que permitirán administrar toda esta información genómica. Es por ello, que los sistemas más representativos en información biológica están orientados a la genómica.

Actualmente existen muchas investigaciones orientadas a la genómica, pero debemos tener en cuenta como lo mencionamos en la sección 2.2.2, que es necesario conjuntar diversos componentes biológicos, para poder obtener modelos comprensibles de ciertos comportamientos biológicos. Parte de esta concepción es utilizada en el caso de estudio presentado en el capítulo 7.

Ahora mostramos algunos de los enfoques utilizados en algunos sistemas implementados, orientados a la información biológica.

- *Uso de Data Warehouse.*- Los Data Warehouses fueron diseñados para aplicaciones de negocios por lo que su consistencia es estática. Los datos biológicos muchas veces necesitan cambios dinámicos y necesitan mantener historiales de datos para comparación, procesamiento y cambio de contexto; un Data Warehouse puede brindar cierta ayuda, pero no tiene el soporte suficiente para este tipo de datos. Aunque son utilizados cuando se obtienen datos externos desde diferentes fuentes heterogéneas. Un Data Warehouse puede ocupar diferentes niveles o capas en una arquitectura implementada, algunos ejemplos de su utilización pueden verse en [28], [50], [51].
- *Uso de archivos de texto plano.*- Muchos sistemas que contienen información biológica utilizan archivos de texto plano para la administración y presentación de datos, esto genera sistemas bastante rígidos y con alta complejidad en su mantenimiento. Por otro lado existen ambigüedades en los datos y algunas veces no se tiene un acceso a un dato en particular, por lo que es necesario leer los datos contenidos y buscar manualmente la ubicación de los mismos, ejemplos de estos sistemas son [29], [30].
- *Interfaces Web rígidas.*- Muchas de las interfaces para sistemas orientados a la Web presentan formularios HTML, con diversas cajas de texto que llenar para poder realizar una consulta. Algunas veces los formularios soportan operaciones AND, OR y NOT, en la consulta como puede verse en [30], [51]. Algunas interfaces más complejas requieren introducir sentencias SQL puras para poder realizar las consultas, pero para hacerlo es necesario conocer la estructura interna de la base de datos.
- *Interfaces Web complejas.*- Los sistemas más avanzados y potentes, contienen complejas y sofisticadas interfaces que son orientadas a personas con un amplio conocimiento en cierta área de estudio, ya que contienen una variedad amplia de opciones especializadas. Por lo cual son inaccesibles a cualquier usuario no especializado en esa área. Otras veces, son tantas las páginas de navegación a las que lleva un procesamiento de datos, que es muy fácil perderse y no saber dónde se está ubicado en el sistema. Algunas interfaces con estas características podemos verlas en [30], [32], [52].

Estos son algunos de los principales enfoques usados en sistemas orientados a información biológica. Los dos últimos puntos son los más interesantes en nuestra investigación; estos puntos son tratados con más detalle en el siguiente capítulo, ya que nuestro principal interés es obtener un modelo para sistemas en la Web.

## 2.4. Comentarios Finales

En esta sección hemos observado la importancia que tiene actualmente la computación en la investigación científica. Hemos destacado el llamado que existe por diversos investigadores de diferentes áreas de investigación y principalmente del área biológica, por ejercer un acercamiento con científicos en computación. Esto parece ser el antecedente de una nueva manera de realizar investigación en el futuro, con científicos trabajando de manera interdisciplinaria en todas las áreas de estudio.

También en este capítulo hemos observado algunas de las diferencias entre los datos de negocio y los datos científicos; más aún, se mostraron algunas de las particularidades de los datos biológicos. Estas particularidades son importantes, porque son parte fundamental de nuestro proyecto de investigación, ya que trabajamos con este tipo de datos en particular.

Por último, se han mostrado algunos de los enfoques usados por diferentes sistemas orientados a información científica, algunos de ellos son de nuestro interés como se ha señalado y serán mostrados con más detalle en los dos siguientes capítulos.

Como conclusión final, podemos decir que los datos biológicos están abriendo nuevas preguntas en el área de computación, muchas de ellas basadas en el manejo de este tipo de datos de manera eficiente. Los métodos tradicionales no son suficientes en la gran mayoría de las veces, esto hace necesaria la investigación para proponer nuevos modelos y arquitecturas orientadas a este tipo de información, así como lo han señalado diversos científicos siendo el principal propósito de esta tesis.

## Capítulo 3

# Interacción y la Web

La interacción que realiza un usuario por medio de la interfaz de un sistema de información, puede definir si un sistema es útil, usable y exitoso para el propósito que fue desarrollado. El diseño de la interfaz es una de las partes más importantes en el desarrollo de un sistema; los elementos contenidos en la interfaz de un sistema pueden generar una cierta facilidad o complejidad en su uso. El tiempo de respuesta a las acciones del usuario en la interfaz, también representa un importante factor en la interacción.

En este capítulo se muestra como es llevada a cabo la interacción en sistemas de escritorio y en sistemas Web. En principio se muestran algunas soluciones para sistemas de escritorio, tal como lo son los lenguajes visuales y las consultas dinámicas. Después se muestra como es llevada a cabo la interacción en sistemas Web y la evolución que están teniendo actualmente. Al final del capítulo en base a lo descrito, se obtiene un modelo para la interacción en sistemas Web orientados a la exploración de datos científicos. Este nuevo modelo sirve como preámbulo para la construcción de otro más robusto en el capítulo 5.

### 3.1. La Interfaz y los Usuarios

La computadora es un gran recurso que nos permite realizar múltiples tareas, pero desafortunadamente muchas personas tienen diversas experiencias frustrantes cuando la usan. Una de las principales causas de este problema es la presentación de una interfaz confusa o compleja al usuario, la cual puede requerir de un entrenamiento especializado para su correcto uso.

El diseño de la interfaz puede hacer a un sistema: bastante exitoso, o no tan exitoso pero algo funcional o llevarlo a un completo fracaso.

Los malos diseños de una interfaz, pueden hacer fracasar a un sistema por no cumplir con algunas de las expectativas de los usuarios como: facilidad de uso, corto tiempo de respuesta, diseño intuitivo, obtención de resultados requeridos, etc. A su vez, la interfaz puede ser difícil de comprender, complicada de usar, limitada, estática y requerir de software adicional de terceros para su correcto funcionamiento. La esencia de este problema, radica en que muchos usuarios tienen poca experiencia en el manejo y uso de software; su conocimiento adquirido es muy limitado y por lo consiguiente fácilmente se frustran, esto dificulta severamente la efectividad y el uso de la tecnología, tal como lo describe Shneiderman en [53], además de causar preocupación y desánimo en los usuarios como lo comenta Bevan en [54].

Algunas investigaciones muestran que al menos 5.1 horas por semana, o un tercio del total de tiempo enfrente de una computadora, es perdido tratando de usar un programa. Shneiderman en [55], menciona que un 45 % del tiempo es gastado en experiencias frustrantes relacionadas con menús confusos, cajas de diálogo indecifrables y desorientación para encontrar algunas funciones; parte de este problema se debe a los malos diseños de interacción con el usuario como se describe en [53], [56]. Una interfaz compleja, puede hacer que características clave diseñadas para hacer más atractivo el producto, no sean utilizadas, ya sea por desconocer su existencia, por no saber cómo utilizarlas, o simplemente no entenderlas.

Muchas investigaciones se han enfocado a este problema y han surgido diversas soluciones. Las más interesantes para el proyecto son las relacionadas a consultas visuales y consultas dinámicas, debido a que han sido orientadas a tratar de ofrecer una amplia interacción con las bases de datos.

Es indispensable señalar que este tipo de consultas son para sistemas de escritorio. A continuación se describen estas soluciones.

### 3.1.1. Consultas visuales

Los *Sistemas de Consultas Visuales (VQS Visual Query System)* están basados esencialmente en el uso de representaciones visuales y amigables interfaces, que sirven para expresar consultas relacionadas a un cierto dominio de interés en una base de datos, facilitando la interacción de los usuarios. Los VQS son orientados a una gran variedad de usuarios con habilidades técnicas limitadas, inexperiencia en el uso de un sistema, poca experiencia

### CAPÍTULO 3. INTERACCIÓN Y LA WEB

con computadoras, poco o nulo conocimiento de la estructura interna de la base de datos. Algunos ejemplos de VQS podemos verlos en [57], [58], [59]. Para alcanzar este enfoque, los VQS adoptan un amplio rango de representaciones visuales y estrategias, que aumentan la interacción de los usuarios con las bases de datos. Esto quiere decir que se tiene una exploración o manipulación interactiva de los esquemas de bases de datos a través de una representación gráfica, tal como puede verse en [44].

El propósito de adoptar una representación visual para una consulta en un sistema, sirve para comunicar de manera eficaz al usuario, la información contenida dentro de la base de datos. De esta manera, el usuario se puede concentrar en rasgos esenciales y omitir los detalles que son innecesarios [57]. Las interfaces visuales de un VQS consta de dos partes: la primera ofrece una variedad de funcionalidades para facilitar la interacción y la segunda provee un lenguaje para expresar las consultas en forma gráfica, el cual es denominado *Lenguaje de Consulta Visual (VQL Visual Query Language)*.

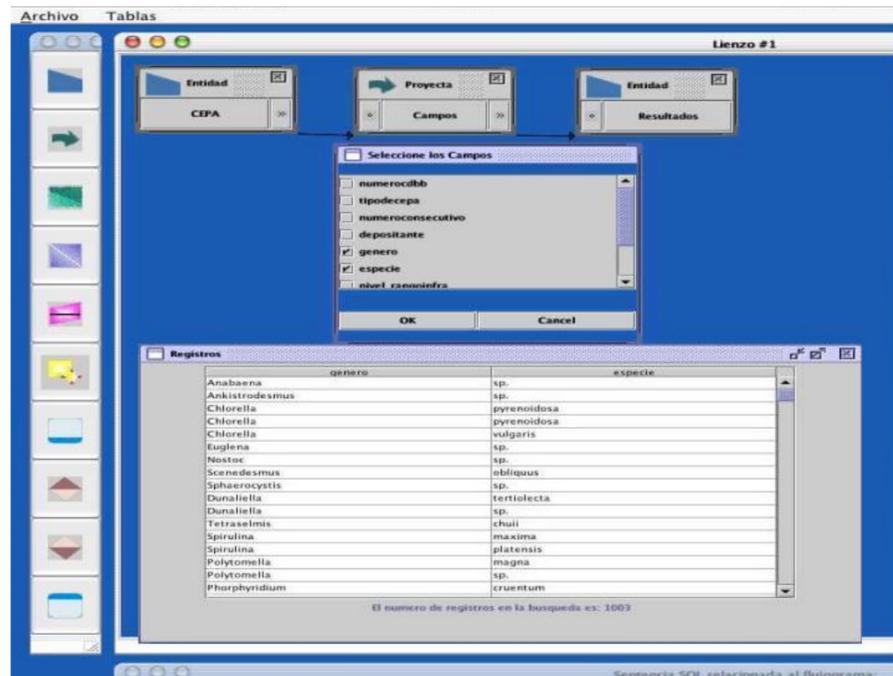


Figura 3.1: El sistema Lida utiliza iconos para construir consultas a la base de datos, ver [43].

Un VQL puede implementar consultas con una predefinida semántica [60], [58]. Para hacer estas consultas visuales, sólo es necesario construir una consulta a partir de ciertos elementos gráficos como iconos o figuras, estos elementos representan a las entidades y las propiedades contenidas en la base de datos. A su vez, también se pueden representar las operaciones y condicionantes que contiene un lenguaje de consulta a bases de datos. Estos elementos gráficos pueden expresar toda clase de datos y modelos de conocimiento de la estructura interna de la base de datos, permitiendo liberar al usuario de la complejidad de la base de datos, ya que esta es representada de una manera visual, como se describe en [60], [43], [61], [59], [62], [58].

En la figura 3.1, podemos observar al sistema Lida (ver [43]), el cual es un sistema orientado a consultas visuales. El sistema contiene iconos que representan entidades y operaciones a realizar en la base de datos; pero estos deben mantener un cierto orden lógico para que la consulta pueda ser realizada con éxito.

### 3.1.2. Limitación en las consultas visuales

Aunque los lenguajes visuales son una buena solución, también es necesario resaltar algunas de sus limitantes en su implementación como:

- *Uso para aplicaciones de escritorio.*- Un VQL es principalmente orientado a aplicaciones de escritorio, para poder implementar toda la interfaz de manera visual y el tiempo de respuesta sin restricciones.
- *Abuso de elementos gráficos.*- Requerir de muchos elementos gráficos como iconos, flechas, líneas, entre otros, causa una saturación de memoria de la computadora, ocasionando que el sistema sea lento y su rendimiento no sea el esperado. Si la implementación es en lenguaje de bajo nivel, se convierte en dependiente de un sistema operativo en especial.
- *Aprendizaje de un nuevo lenguaje.*- Un lenguaje visual requiere de un previo aprendizaje o entrenamiento de uso; también requiere de ciertas reglas para poder ser usado de manera adecuada (ver [59], [43]).
- *Falta de soporte Web.*- La interfaz visual impone muchas limitantes para su implementación en sistemas Web, porque la interfaz depende de un navegador, además de producir una saturación en memoria al tener tantos elementos gráficos y datos del diccionario de datos para realizar una consulta.

### 3.1.3. Consultas dinámicas

Actualmente grandes bases de datos han llegado a ser ampliamente disponibles para los usuarios, nuevas herramientas son necesitadas para procesar y filtrar la información de manera fácil, rápida y eficaz. Esta necesidad ha motivado el surgimiento de nuevos paradigmas para la consulta a bases de datos; uno de los más interesantes es el relacionado a la consulta por manipulación directa descrito en [63].

Las consultas son generalmente expresadas en lenguajes de alto nivel como SQL y trabajan muy bien para muchas aplicaciones. Pero para usuarios inexpertos y no cercanos al área de informática, los usuarios pueden necesitar aprender a formular consultas en el lenguaje SQL, lo cual les puede llevar mucho tiempo y es muy probable que incluyan una multitud de errores en sintaxis y semántica, porque no tienen un preciso conocimiento de cómo funciona una base de datos y como es su estructura interna [46].

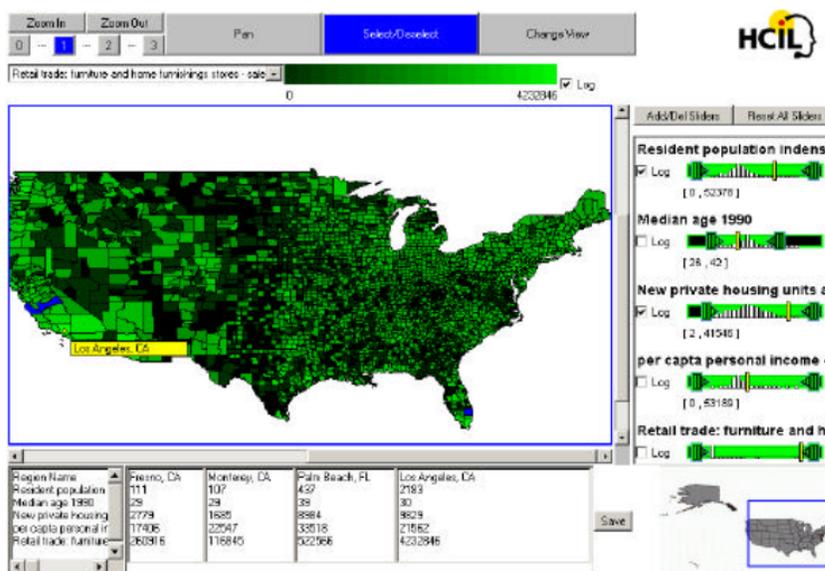


Figura 3.2: Ejemplo de un sistema orientado a consultas dinámicas, el sistema es un desarrollo del Human-Computer Interaction Laboratory HCIL, dirigido por Shneiderman [64].

Las *Consultas Dinámicas* (*DQ Dynamic Query*) surgen como respuesta a esta problemática y son una innovación con respecto a las consultas tradicionales.

Una DQ involucra el control interactivo, así como manipulación directa del usuario para generar consultas a través de movimientos en controles como: *Barras deslizantes (Sliders)*, *Botones (Buttons)* y *Casillas de verificación (Check Box)*. Estos controles de interacción contienen acotaciones, rangos, intervalos y límites en los datos, como se describe en [45], [46], [65].

Una de las principales ventajas de las DQ es que permiten a los usuarios generar consultas específicas, sin requerir que los usuarios memoricen toda la sintaxis y semántica que requiere un lenguaje de consultas; es decir, una DQ permite explorar una base de datos sin requerir una consulta específica o lenguaje en mente.

Las DQ muestran una vista general de la base de datos, por medio de una representación visual en un panel de visualización de datos; esta forma de representación permite obtener rápidas actualizaciones a través del ajuste y la manipulación directa de barras deslizantes. Todos los datos recuperados son mostrados en el panel de visualización a través de una representación gráfica como puntos, líneas, barras, rectángulos, círculos, áreas irregulares, etc. Las representaciones pueden ser de diferentes colores y pueden estar una encima de otra; así mismo, los datos pueden ser seleccionados directamente del panel de visualización, para su consulta particular y observar su contenido.

La manipulación a través de barras deslizantes permite explorar y filtrar convenientemente la información no deseada de la base de datos, como se muestra en [46], [47]. Las DQ han ganado cierta popularidad y los principales y más representativos sistemas basados en DQ han sido desarrollados en sistemas de escritorio. Esto se debe a que los sistemas han sido programados en lenguaje C y parte en lenguaje ensamblador, para permitir ofrecer un tiempo de respuesta casi inmediato en la consulta.

Actualmente existe un fuerte incremento en el interés de trasladar este concepto a sistemas orientados a la Web. Uno de estos ejemplos puede verse en la figura 3.2, donde podemos observar un panel con diferentes barras deslizantes, las cuales contienen un rango de valores de consulta a realizar. Algunos de estos sistemas han sido implementados con Java Applets, Flash y con HTML Dinámico (DHTML), como puede verse en [64], [66]; pero estas implementaciones sobrecargan el lado cliente y ofrecen una lenta respuesta.

### 3.1.4. Limitaciones en consultas dinámicas

Los sistemas basados en DQ tienen muchas ventajas, pero también presentan algunas limitantes importantes que debemos tomar en cuenta como:

- *Rango limitado.*- Las barras deslizantes (*Sliders*) son los principales elementos de interacción en las consultas dinámicas. Pero las barras sólo pueden contener un número finito o rango limitado de valores; por lo cual, las consultas sólo son válidas para un rango predeterminado de valores, además no hay información sobre si una actualización en la base de datos es reflejada de manera inmediata en la interfaz.
- *Actualización de valores.*- La principal característica de una DQ es una inmediata respuesta en la interfaz; para alcanzar este alto rendimiento en la respuesta es necesaria una programación de bajo nivel. Esto hace al sistema dependiente de un sistema operativo específico.
- *Pobre soporte Web.*- Las DQ han fracasado al ser llevadas a la Web, ya que han tenido que utilizar software no estándar como los *plugins*, Java Applets, Flash, los cuales hacen que la respuesta en la interfaz del sistema Web sea lenta. Otra alternativa fue usar DHTML, pero la falta de estados en los sistemas Web resultó ser un grave problema; a lo cual, la opción fue saturar la aplicación de capas ocultas al usuario y activarlas según la exploración. La precarga hace lento al sistema originando una saturación de memoria y generando un riesgo de inestabilidad del sistema, ver [66].

Todas las limitantes señaladas son importantes y deben de tomarse en consideración en un desarrollo basado en DQ. La principal característica de una DQ es su tiempo de respuesta y proponemos incluirla en un sistema Web. La implementación de esta característica es presentada en el capítulo 6.

### 3.1.5. Usabilidad

Las alternativas para una mayor interacción como los lenguajes visuales y las consultas dinámicas, responden también a la necesidad de brindar una mayor usabilidad a los sistemas. La *Usabilidad* está estrechamente relacionada con la facilidad de uso, la cual es la manera más común en que este término es usado. Pero en ese sentido; uno puede hablar de un buen diseño de interfaz, pero que no cumple con el objetivo para el cual un sistema fue construido, lo cual significa que puede ser muy usable pero no útil, como lo afirma Bevan en [67].

No es fácil hablar sobre la usabilidad de un sistema, debido a que muchas veces depende del contexto en el que el software es usado. Cambiar algún aspecto relevante en el contexto de uso, puede cambiar la usabilidad del producto; así, un producto puede ser usable para usuarios entrenados, pero no usable para usuarios sin ese entrenamiento [54].

El término *Usabilidad* no tiene una definición única; en la práctica puede significar diferentes cosas para diferentes personas y puede ser vista de muy diversas maneras para diferentes propósitos.

Algunas definiciones de *Usabilidad* son:

- ISO/IEC 9126.- Se refiere a la capacidad de un software de ser comprendido, aprendido, usado y atractivo para el usuario en condiciones específicas de uso.
- ISO/IEC 9241.- Es la efectividad, eficiencia y satisfacción, con la que un producto permite alcanzar objetivos específicos en un contexto de uso específico.

Pero la definición que nosotros ocuparemos en este proyecto de investigación, es la de Nigel Bevan en [54], donde él define a la usabilidad de la siguiente manera:

**Definición 1.** Usabilidad.- *Es la habilidad de un producto a ser usado con efectividad, eficiencia y satisfacción por usuarios específicos, para alcanzar metas en ambientes particulares.*

La definición presentada es importante porque parte de la propuesta es obtener sistemas de exploración de datos, con rasgos de usabilidad inmersa en ellos.

La usabilidad no sólo es una propiedad que deben tener los sistemas de escritorio; la usabilidad también está presente en los sistemas Web, pero presenta diferentes rasgos debido a que toda la interacción es a través de una página Web presentada al usuario. Parte de este tema se presenta en la siguiente parte de este capítulo.

Pero antes, es imprescindible para nuestro objetivo en esta tesis presentar una propuesta de diseño de interfaz no muy conocida, pero que contiene una idea interesante que podemos retomar y modificarla para darle un uso muy específico en nuestra propuesta final.

### 3.1.6. Interfaz multi-nivel

En 2003 Shneiderman presentó una propuesta sobre diseño en una interfaz llamada *Diseño de Interfaz Multicapa (Multi-Layer Interface Design)*, a la cual denominaremos en esta tesis de aquí en adelante como *Diseño de Interfaz Multi-nivel*, para evitar confusiones con las capas del modelo que se propone más adelante. Shneiderman afirma que este diseño es importante en la investigación sobre interacción humano-computadora y el incremento de interés de la usabilidad universal.

La usabilidad universal es definida por Shneiderman como: “*Los servicios de información y comunicación para ser usables por todos los ciudadanos*”. Shneiderman menciona que la usabilidad universal debe ser direccionada a usuarios que tengan diferentes lenguajes, culturas, inexpertos, intermedios, expertos, literatos, jóvenes, adultos mayores, o con discapacidades como lo menciona en [56], [25] y [55].

También afirma en [27], que la usabilidad universal debe inspirar nuevos avances en investigación sobre cómo proveer mejores experiencias de usuario. Una de sus contribuciones hacia la usabilidad universal, es precisamente su diseño de interfaz Multi-nivel, del cual ofrece una representación visual mostrada en la figura 3.3.

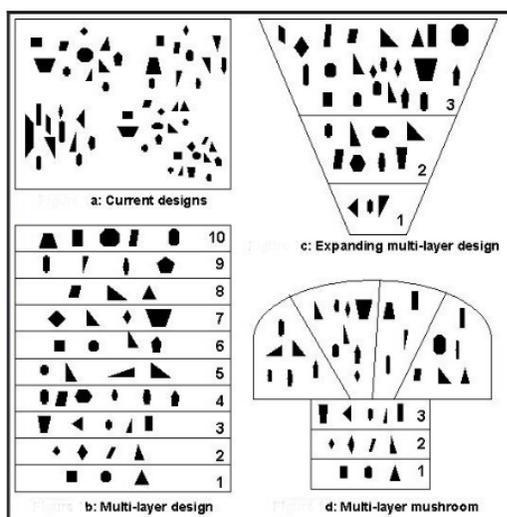


Figura 3.3: Propuesta de diseño multi-nivel de Shneiderman, el diseño debe permitir una interacción basada en niveles según la experiencia del usuario (Figura original de Shneiderman en [55]).

Este enfoque puede ser definido como:

**Definición 2.** Diseño de interfaz multi-nivel.- *Este diseño utiliza diferentes niveles de interacción con el usuario, para habilitar a los usuarios de sistemas complejos en la obtención de un fácil y progresivo avance, según sus necesidades.*

El diseño multi-nivel ofrece solo cierto control a los usuarios sobre un pequeño grupo de elementos disponibles en un primer momento; a medida que el usuario interactúa con el sistema a través de la interfaz, puede irse moviendo hacia niveles más altos de interacción, como lo muestra la figura 3.3, en las opciones (b), (c) y (d). Shneiderman presenta dos casos de estudio, los cuales son un procesador de palabras y una herramienta de exploración geoespacial. En estas herramientas de software, se observa como el sistema restringe el acceso a niveles más altos en base a las necesidades del usuario; pero hay que señalar que estas dos herramientas son orientadas como sistemas de escritorio.

Shneiderman visualiza este diseño como una promesa a futuro para nuevos desarrollos. Aunque también reconoce que existen muchas dificultades en una implementación para un uso real, debido a la dificultad para definir los niveles y la definición de la interacción entre ellos, por lo que no queda determinada su eficacia para aplicaciones reales.

Una vez mostradas algunas soluciones en aplicaciones de escritorio, veamos a continuación como la interacción es llevada a cabo en sistemas Web, así como los cambios vertiginosos que están ocurriendo por su evolución misma.

## 3.2. La Evolución de la Web

La Web está sufriendo una rápida transformación con el surgimiento de nuevos conceptos, términos, técnicas y aplicaciones que la transforman día a día. La Web está evolucionando muy rápidamente, incluso en la interacción con los usuarios; esto nos dice que debemos prepararnos para una nueva generación de sistemas de información basados en la Web. Esta sección inicia con la descripción del área de investigación llamada ingeniería Web, para después hacer una descripción de los cambios que están ocurriendo en la Web, así como la interacción con los usuarios y hacia donde se dirige.

### 3.2.1. Ingeniería Web

Desde el surgimiento de la Internet, se ha generado toda una expectativa por querer contar con sistemas orientados a la Web. La frenética carrera ha ocasionado un incremento en este tipo de sistemas, pero el bajo rendimiento de algunos y la tendencia a tener fallas de todo tipo ha generado una enorme cantidad de problemas. Muchos de estos problemas han sido generados por los pobres desarrollos, deficientes implementaciones y malos diseños por parte de los desarrolladores.

Algunos de estos problemas motivaron el surgimiento de lo que hoy conocemos como *Ingeniería Web* (*Web engineering*, en inglés).

La ingeniería Web es establecida en 1998 como una nueva disciplina de investigación en la universidad de Western Sydney. Desde entonces ha despertado un fuerte interés por parte de desarrolladores Web, agencias de gobierno, académicos e investigadores. Adicionalmente, ha atraído a profesionales de áreas relacionadas como: ingeniería de software, sistemas distribuidos, recuperación de información, entre otros.

La mayoría de las veces el término de ingeniería Web es concebido como un sinónimo de ingeniería de software, causando mucha confusión en la utilización de este término; debido a esto, Ginige y Murugesan en [68], hacen una aclaración con respecto a esta confusión:

- “Contrariamente a la percepción que tienen algunos desarrolladores profesionales de software, la ingeniería Web no es un clon de la ingeniería de software, aunque ambos involucran programación y desarrollo de software (Ginige y Murugesan en [68])”.

Adicionalmente a esta aclaración, ellos también explican a detalle muchos de los propósitos y alcances que tiene la ingeniería Web y en base a esto se obtiene la siguiente definición:

**Definición 3.** Ingeniería Web.- *Es un área multidisciplinaria que trata con todos los aspectos de concepción, desarrollo, implementación, rendimiento, mantenimiento e involucra diversas áreas como: análisis y diseño de sistemas, ingeniería de software, ingeniería hipermedia e hipertexto, ingeniería de requerimientos, interacción humano-computadora, ingeniería de la información, desarrollo de interfaz de usuario, recuperación de la información, pruebas, modelado, simulación, administración de proyectos, presentación y diseño gráfico (Ginige y Murugesan en [68]).*

La ingeniería Web se ha convertido en una área de estudio de fuerte necesidad, debido a que los sistemas basados en la Web crecen y cambian más rápido en sus requerimientos, contenidos y funcionalidad durante su ciclo de vida, que el tradicional software de escritorio. El desarrollo de sistemas Web es significativamente diferente del desarrollo de tradicional software y presenta muchos desafíos adicionales. Existen diversos atributos a tomar en cuenta en el desarrollo de sistemas Web como: facilidad de navegación, accesibilidad, escalabilidad, mantenibilidad, usabilidad, compatibilidad, interoperabilidad, fiabilidad y seguridad (ver [69]).

### 3.2.2. Web 1.0

La *Web* (World Wide Web) es un sistema de documentos de *Hipertexto* (*Hypertext*) dinámicamente relacionados, enlazados y accesibles a través de una serie de enlaces llamados *Hipervínculos* (*Hypertlinks*). Los hipervínculos habilitan la referencia cruzada y distribuida, cruzando el espacio de información a través de *Internet*.

*Internet* es un conjunto descentralizado de redes de comunicación que utilizan el protocolo TCP/IP, que garantiza que redes físicas heterogéneas puedan comunicarse. La navegación de los documentos es a través de lo que llamamos *Navegador* (Browser), la mayoría de ellos pueden descargarse de manera gratuita.

Parte importante del éxito de la Web 1.0 es su simplicidad, debido a que ofrece una interfaz uniforme para distribuir datos y tiene un potencial de extensión ilimitado. En la Web 1.0, la comunicación siempre es síncrona y utiliza la arquitectura clásica cliente-servidor. La estructura interna está constituida por un *Lenguaje de Etiquetado de Hipertexto* (*HTML Hypertext Markup Language*) y los documentos construidos con este lenguaje son conocidos como páginas Web o páginas HTML [70].

Las páginas Web creadas en HTML pueden ser de dos tipos:

- *Páginas Estáticas.*- Son páginas que no cambian en el tiempo y cada vez que se les solicita muestran el mismo contenido; a su vez; pueden contener archivos multimedia, animaciones y utilizar lenguajes cliente como JavaScript, ActionScript, etc.
- *Páginas Dinámicas.*- Estas páginas son creadas en tiempo de ejecución por parte de un servidor Web. El servidor Web construye una página y su contenido completamente para dar respuesta a una petición de una computadora cliente. La petición puede requerir la ejecución de una aplicación o el acceso a una base de datos. Actualmente la implementación de sistemas Web con páginas dinámicas utiliza lenguajes scripts en el servidor como: PHP, JSP, ASP, ColdFusion, entre otros.

Las páginas estáticas son orientadas a sistemas Web simples y las páginas Web dinámicas son orientadas a sistemas Web avanzados.

Algunas de las diferencias entre estos sistemas son presentadas en [68] y son mostradas a continuación:

#### *Sistemas Web simples*

- Página simple y presenta información textual.
- La información no cambia.
- Navegación simple.
- Sistemas independientes
- No se requiere alto rendimiento
- Desarrollado por una persona o un pequeño equipo.
- Usado para diseminación de información.

#### *Sistemas Web avanzados*

- Páginas Web complejas (HTML con código script servidor).
- Información dinámica que cambia según la necesidad de los usuarios.
- Dificultad en navegar y encontrar la información deseada.
- Integración con bases de datos u otros sistemas.
- Requiere alto rendimiento y disponibilidad continua.

- Requiere un gran equipo de desarrollo con experiencia en distintas áreas.
- Desarrollado para aplicaciones de misión crítica.

Los sistemas avanzados Web han forzado la evolución de navegadores, lenguajes script servidor, técnicas de desarrollo, comunicaciones, etc.

Pero también la Web misma está cambiando la interacción entre las personas y está provocando el surgimiento de nuevas comunidades virtuales de manera vertiginosa. Nuevas técnicas y conceptos para innovar la Web están surgiendo y los cambios son tan rápidos que muchas veces no nos damos cuenta.

### 3.2.3. Web 2.0

El término Web 2.0 fue acuñado por Tim O'Reilly en 2004, existiendo una cierta confusión inicial de la verdadera interpretación o significado que debía darsele a este término. Actualmente la Web 2.0 es considerada como un cambio de paradigma de la Web, ya que pasa de medio de publicación a medio de interacción y participación, como se precisa en [71], [72], [73], [74].

Con la Web 2.0 los usuarios ya no son meramente usuarios pasivos, se convierten en participantes activos; en otras palabras, la Web 2.0 es marcada por la fuerte integración y participación de los usuarios. Los usuarios se convierten en productores y proveedores de contenido, formando diversas redes sociales y comunidades virtuales en una amplia red global [75], [76], [77], [72],[73].

Con el advenimiento de la Web 2.0, también ha surgido una cantidad sin igual de nuevos términos y conceptos que hasta hace muy poco eran totalmente desconocidos como: Wikis, Blogs, Ajax, Folksonomía, Mashups, entre algunos otros (ver figura 3.4). Muchos de estos nuevos conceptos han tenido gran aceptación y actualmente gozan de mucha popularidad y uso, hasta hace poco tiempo no imaginable.

La Web 2.0 ya es conocida ampliamente como la Web Social y de ella han surgido una serie de nuevos *Servicios de colaboración social* entre los cuales se encuentran: Foros de discusión (Blogs), Administradores de contenido de autoría colaborativa (Wikis), Aplicaciones Web mezcladoras de contenido externo o Aplicaciones Web Híbridas (Mashups), Sindicación Web o Redifusión de contenido (RSS Really Simple Syndication), entre otros (ver [72], [78], [79], [71]).

Wikipedia, Blogger, eBay, son algunos ejemplos de sitios orientados a la Web 2.0, donde los usuarios contribuyen con contenido y servicios; generalmente interactúan unos con otros, sin involucrar de manera directa al proveedor del servicio. Servicios como estos deberán llegar a ser un importante enfoque, para que los científicos en un futuro próximo puedan organizar, guardar, obtener y descubrir información; construyendo y extendiendo vastas redes sociales colaborativas de científicos en línea. [75].



Figura 3.4: En la Web 2.0 han surgido nuevos términos y conceptos.

La Web 2.0 también es considerada como una revolución social en el uso de las tecnologías Web, combinando técnicas maduras de implementación para aplicaciones Web dinámicas y re-descubriendo JavaScript como medio potente de interacción a través de nuevos usos y funcionalidades complementándolo con modernas técnicas de implementación.

Un ejemplo del nuevo uso de JavaScript es el surgimiento del término de *Ajax*, (*Asynchronous JavaScript and XML*). Con este nuevo concepto podemos crear interfaces con una rica interacción dinámica (*Rich Interaction*), muy similar a aplicaciones de escritorio. Ejemplos de este tipo de aplicaciones son: Google Maps, Google Docs, Google Suggest, Gmail, Yahoo Mail, Windows Live, etc. [77], [71], [72], [80], [81], [70].

Mucho del éxito de la Web social ha sido en base a la facilidad de uso y soluciones simples [82]. La Web 2.0 ha creado una nueva generación de aplicaciones y servicios Web como: YouTube, Flickr, Google Maps, del.icio.us, Wikipedia, Amazon, entre otras [76], [79], [83], [84].

### 3.2.4. Web 3.0

El término *Web 3.0* fue acuñado por John Markoff en una publicación del New York Times en 2006, causando gran confusión y controversia con respecto a la relación entre la Web semántica y la Web 3.0, porque el término fue usado como sinónimo de *Web 3.0*. Parte de esta discusión podemos verla en [85], [73], [74].

Actualmente no existe un consenso sobre la definición o significado que tiene el término de *Web 3.0*, aunque diversos autores coinciden en que en esta nueva etapa de la Web, el significado será parte de la nueva Web, pero no hay un acuerdo sobre cuáles son o serán los caminos más apropiados para su desarrollo y futuro.

Diferentes investigadores ven a la Web 3.0, como una meta a alcanzar en un corto plazo y como parte de una evolución de la Web 2.0; contrario a la meta de la Web Semántica, que se considera ser alcanzada a largo plazo.

### 3.2.5. Web semántica

Al igual que existió una confusión con el término de Web 3.0 y el de Web semántica, anteriormente hubo otra confusión entre el término de Web 2.0 y el de Web semántica; parte de la confusión surgió por referirse a ellos como sinónimos.

Al ser acuñado el término Web 2.0 en 2004 por Tim O'Reilly, este fue fuertemente asociado con el de *Web Semántica*; porque en ese mismo año, el RDF (*Resource Description Framework*) y el lenguaje OWL (*Ontology Web Language*) obtienen el estándar de la W3C.

La confusión fue dada, porque el término de Web 2.0 no fue claramente definido y la Web Semántica se presentó como la nueva tendencia a seguir, como puede verse en [74], [77]. Pero actualmente, el término de Web 2.0, ya es definitivamente asociado al de *Red Social* como puede verse en [82], [78].

El término de *Web Semántica* fue acuñado por Tim Berners en el año 2001. Berners describe a la *Web Semántica* como “Dar un significado bien definido a la información, para mejorar la habilidad de las computadoras y personas a trabajar en cooperación” [82], [74].

La *Web Semántica* basa su idea en habilitar a las computadoras para que puedan comprender documentos semánticos y datos. Para que esto sea posible; es necesario que las computadoras tengan acceso a colecciones estruc-

turadas de información y diferentes conjuntos de reglas de inferencia usadas para conducir un razonamiento automatizado [86]. Todo este proceso debe ser llevado a cabo por agentes de manera transparente para el usuario final [85], [86].

Berners explica en [86], como debería ser el funcionamiento ideal de la Web semántica en el famoso *Caso de Lucy*, que presentó en el año 2001.

Hacer realidad el caso de Lucy, todavía necesita un largo camino por recorrer, porque la Web actual ha generado una inmensa cantidad de información en una Web orientada a documentos [87].

La Web semántica actual está basada en el uso de metadatos semánticos que describen el contenido, el significado y la relación entre datos, de manera que puedan ser evaluados automáticamente. Para alcanzar este objetivo se hace uso de RDF (*Resource Description Framework*), el cual es un lenguaje para la definición de ontologías y metadatos en la Web. El elemento de construcción básica en RDF es el triplete o sentencia, que consiste en dos nodos llamados sujeto y objeto, unidos por un arco llamado predicado, donde los nodos representan recursos y los arcos propiedades y relaciones. A través del RDF, se definen jerarquías de clases de recursos, especificando todas las propiedades y las relaciones que se admiten entre ellas; todas las clases y relaciones son recursos y por lo tanto se pueden examinar y recorrer.

La implementación RDF también necesita de un lenguaje que permita la construcción consultas. Para ello surgió OWL (*Ontology Web Language*), el cual es la fusión de dos lenguajes similares DAML (*DARPA Agent Markup Language*) y OIL (*Ontology Inference Language*). De esta manera, OWL se considera una extensión de RDF.

Para desarrollar aplicaciones basadas en RDF-OWL o lenguajes similares, es necesaria la utilización de diferentes librerías externas para leer y procesar las ontologías definidas en estos lenguajes. Escribir en lenguajes como RDF y OWL resulta sumamente difícil y alto riesgo de cometer errores. Tratando de solventar parte de este problema se recurre a entornos gráficos para visualizar y construir ontologías. Las páginas generadas por la Web semántica son documentos XML y todos los datos contienen etiquetas que tratan de describir parte del significado del dato. El cambio de una Web orientada a documentos a una Web semántica requiere un radical cambio de paradigma y en muchas maneras este debe ser revolucionario; pero como se menciona en [70], una revolución también puede finalizar en un desastre.

La transición entre los diferentes paradigmas no se sabe como ocurrirá, aunque Bernes asegura que ya se está formando el embrión de la futura Web Semántica.

El nombre de *Web Semántica* también ha causado una cierta controversia. Recientemente Berners declaró en una entrevista, que quizás él debió llamarla *La Web de Datos* y definirla como: “La Web capaz de interpretar e interconectar un número mayor de datos”, dado que la palabra *Semántica* es utilizada para significar diferentes cosas.

Una vez dados los principales enfoques sobre los que trabaja cada versión de la Web, es necesario conocer cómo es la interacción con los usuarios en este tipo de sistemas.

Los sistemas orientados a Web tienen una variedad de restricciones y condicionantes, que no tiene una aplicación o sistema de escritorio. La siguiente sección muestra algunas de estas condicionantes y permite obtener diferentes enfoques orientados a la interacción con el usuario.

### 3.3. Interacción en la Web

El interés de diseñadores e investigadores está enfocado actualmente en migrar sus aplicaciones de escritorio a la Web, pero esta no es una tarea sencilla e involucra diversos aspectos a tomar en cuenta para tratar de alcanzar este propósito. Gran parte de las dificultades son las diferentes restricciones y limitantes que impone un sistema orientado a la Web, ya que se depende de la interfaz mostrada por un navegador para la interacción. Algunas de estas limitaciones son presentadas a continuación.

#### 3.3.1. Limitaciones en un sistema Web

Obtener una interacción eficaz con el usuario, es una importante meta cuando se trata de desarrollar software científico orientado a la Web. Algunas características limitantes que deben tomarse en cuenta en este tipo de sistemas son:

- *Pobre acceso a datos.*- La inmensa mayoría de sistemas Web, usan formularios para obtener datos del usuario. Los formularios regularmente están limitados a unas pocas consultas específicas y planeadas, limitando la capacidad del usuario para explorar todos los datos inmersos, en la base de datos.
- *Interfaces confusas.*- Una tendencia en el diseño Web es la implementación de una interfaz decepcionante, impredecible y con una navegación confusa [53]. Muchos factores pueden afectar el diseño de la interfaz como: uso de tecnologías no estándar, sistema operativo exclusivo, navegador exclusivo, versión del navegador, resolución de la pantalla de la computadora, entre otras, ver [88].
- *Software propietario no estándar.*- El uso de software propietario que no se ajusta a los estándares, provoca sistemas dependientes de sistemas operativos y navegadores específicos para un funcionamiento correcto. El vencimiento de licencias para cierto software propietario puede inhabilitar el uso de un sistema.
- *Software Adicional.*- Algunos sitios necesitan software adicional para poder trabajar correctamente. El software adicional puede consistir en: plugins, máquinas virtuales, librerías externas, librerías propietarias o de terceros. La inexperiencia de algunos usuarios, puede impedir instalar u obtener este software adicional.

Esto causa una severa frustración, porque no se puede interactuar con este sitio, al no obtener o instalar de alguna forma este software.

- *Arquitectura rígida.*- Una arquitectura rígida presenta muchos problemas cuando un sistema necesita adicionar extensiones o características no contempladas en el diseño original. Esto puede originar que sea necesario reconstruir todo el sistema en su totalidad, para incorporar estas nuevas capacidades.

Estas son algunas de las limitantes a tomar en cuenta en el desarrollo de un sistema Web. Pero existen otras limitantes relacionadas a la interfaz, que no permiten un acceso transparente a los datos, las cuales son mostradas a continuación.

### 3.3.2. Limitantes de una interfaz Web

La interfaz de un sistema Web necesita ser interpretada y ejecutada por diferentes navegadores, además de necesitar un control de todos los estados del sistema para que puedan ser reflejados en la misma interfaz. Existen diferentes tipos de limitantes a considerar en una interfaz Web, algunas de las cuales son mostradas a continuación:

- *Dependencia de un navegador.*- Uno de los principales problemas con una interfaz Web, es que dependen de cómo un navegador interpreta su presentación al usuario; también depende de cómo el navegador interprete los eventos relacionados a la interfaz y las diferentes instrucciones para que los diferentes navegadores puedan tener el comportamiento requerido.
- *Componentes de interacción restringidos.*- Los únicos componentes que se pueden usar son los que pueden ser interpretados por un navegador. Aunque muchos de ellos son muy eficaces, algunos de ellos son insuficientes para una rica y amplia interacción con el usuario.
- *Velocidad de actualización.*- Interfaces exitosas en el Web tienen una velocidad de respuesta aceptable, esto implica restringir la cantidad de datos o información no deseada en cada solicitud.
- *Falta de estados.*- Una de las desventajas en el Web es la falta de control de estados de las aplicaciones, existen algunas técnicas como cookies o sesiones para obtener estos estados, pero algunas veces son insuficientes para sistemas de información con más complejidad.

- *Diseños complejos.*- Diversos sistemas de información Web tienen una interfaz compleja, donde el usuario fácilmente se pierde y se confunde en la interacción. Para aprender el uso correcto se requiere de un previo y lento aprendizaje; pero aún con todo este entrenamiento puede seguir usándose de manera incorrecta.
- *Diseños sobrecargados.*- Muchos diseños de interfaz contienen una cantidad de datos o información no necesaria; con la reciente popularidad de acceso a la Web desde teléfonos móviles, se observa una clara necesidad de obtener una interfaz muy ligera que respete el estándar de la W3C, ya que los navegadores en móviles son restringidos.
- *Uso de tecnologías no estándar.*- El uso de diverso software propietario puede restringir su amplio uso, debido a incompatibilidades con los sistemas de los usuarios. Además de tener que lidiar con licencias de uso y requerimientos de sistemas operativos específicos, para uso del software.
- *Falta de diseños para exploración de datos.*- La inmensa mayoría de sistemas Web, contiene una serie de consultas prediseñadas basadas en formularios para ingresar los datos. Alcanzar consultas más complejas requiere de conocer la estructura interna de la base de datos y de aprender un lenguaje de consulta como SQL.

Todas estas limitantes descritas hacen más difícil el diseño de una interfaz para un sistema Web. Pero las limitantes pueden ser agrupadas en cuatro problemas principales los cuales son:

1. Dependencia de interacción a través de un navegador.
2. Control de los estados del sistema.
3. Recuperación de datos.
4. Transmisión de datos.

Adicionalmente a estos cuatro problemas, también es necesario considerar la usabilidad que debe tener el sistema. La usabilidad para sistemas Web, tiene algunas diferencias a considerar de la definición dada anteriormente, como se muestra a continuación.

### 3.3.3. Usabilidad en la Web

La usabilidad en la Web también ha sido un tema central de investigación en el diseño de interfaces de usuario. Jakob Nielsen es una de la autoridades más respetadas en el ámbito internacional sobre usabilidad en la Web.

Nielsen define a la usabilidad como: “Un atributo de calidad, que evalúa la facilidad de uso de una interfaz Web”. Es decir, una interfaz Web usable es aquella que permite interactuar de una manera sencilla, fácil, cómoda, segura y de manera inteligente con el usuario.

También destaca algunas características que debe contener una interfaz como:

- Facilidad de uso
- Entendible
- Novedosa
- Comprensible
- Inteligente
- Atractiva

Nielsen centra a la usabilidad hacia sitios Web estáticos<sup>1</sup> y la facilidad de navegación para encontrar en el menor tiempo posible lo que un usuario busca; esto es, alcanzar un diseño centrado en el usuario.

Nielsen ha publicado diversas pautas para alcanzar la usabilidad en un sitio Web; pero por otro lado, también ha recibido diferentes críticas, algunas de ellas por afirmaciones como: “*Usar flash<sup>2</sup> es 99 % malo*”, o “*El uso de archivos pdf para lectura en línea, es uno de los peores errores de diseño Web*”. Algunas otras críticas están orientadas en el sentido de que Nielsen basa toda la usabilidad en sitios Web estáticos y no brinda pautas para sistemas Web dinámicos.

El diseño de la interfaz para sistemas Web dinámicos, es uno de los grandes problemas a los que se enfrenta un equipo de desarrollo, ya que se debe de tener muy claro el objetivo para el cuál el sistema es desarrollado.

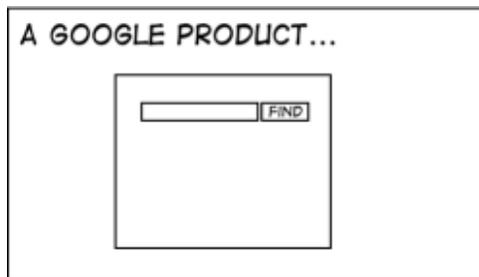
La gestión de proyectos contempla dos problemas relacionados a los alcances en un proyecto de desarrollo de sistemas.

---

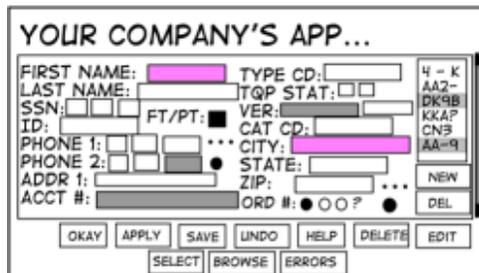
<sup>1</sup>Ver, descripción de página Web estática y la descripción de sistemas Web simples, descritos en la sección dedicada a Web 1.0

<sup>2</sup>Software de Macromedia usado para generar gráficos animados en una página Web.

El primero se denomina como *Scope creep*, el cual se refiere a rebasar el propósito original y brindar características no deseadas en un software. El segundo es denominado como *Instruction creep* que se refiere a un incremento en la complejidad y tamaño de un software, hasta hacerlo totalmente inmanejable. Estos problemas directamente impactan en el diseño de la interfaz de un sistema, pero una manera de evitarlo es aplicar el principio conocido como *KISS (Keep It Short and Simple)*, que puede ser traducido al español como *Manténgalo breve y simple*.



(a)



(b)

Figura 3.5: Principio *KISS* se basa en tratar de obtener una interfaz simple y precisa, que permita ocultar toda la complejidad interna del sistema (Figura original de Eric Burke).

Esto significa en términos informáticos, ocultar toda la complejidad del sistema substrayéndola de la vista del usuario. En otras palabras, esto puede significar: “*Cuanto más simple hacia afuera, más complicado por dentro*”. Aunque el principio es simple, en algunas ocasiones puede ser una de las cosas más complicadas del mundo. Eric Burke conocido autor de libros especializados en Java en O’Reilly, describe en la figura 3.5 la simplicidad y usabilidad en (a) y la complejidad sin pensar en la usabilidad en (b).

### 3.3.4. Aplicaciones RIA

Una de las principales características del software interactivo es que la interfaz provee un incremento en la accesibilidad y por consecuencia un incremento en la funcionalidad.

La Web actualmente ha generado una explosión de interés en el acceso en línea de diversos tipos de datos. Los sistemas Web tradicionales contienen diversas limitaciones para interactuar con la información dinámica.

Debido al incremento en la complejidad de los sistemas Web, los límites de la interacción y usabilidad son los principales problemas a resolver en este tipo de sistemas. La diferencia de experiencia que percibe un usuario entre un sistema de escritorio y un sistema Web, es básicamente por la rapidez en que una interfaz es actualizada. Los sistemas Web son más lentos debido a que dependen de que toda una página sea construida en el servidor y enviada a través de la red; sin olvidar que la interacción es interrumpida hasta que la nueva página es obtenida por el cliente.

Todos estos problemas han originado la búsqueda de nuevas soluciones para interactuar en línea con los datos. Una de estas soluciones ha dado origen a lo que ahora se conoce como una aplicación *RIA* (*Rich Internet Application*), la cual podemos traducir al español como aplicación Web altamente interactiva. El término RIA fue acuñado por Jeremy Allaire en 2002, al cual podemos definir de la siguiente manera:

**Definición 4.** Aplicación RIA.- *Unifica las propiedades de las tradicionales aplicaciones de escritorio con las propiedades de las aplicaciones Web. Esto quiere decir, una RIA combina la rica interactividad de una aplicación de escritorio, con el amplio alcance de una aplicación Web.*

La descripción gráfica de esta definición es mostrada en la figura 3.6, donde podemos observar la unificación entre las aplicaciones de escritorio y las aplicaciones Web. Algunas de las características de las aplicaciones RIA dadas por Bozzon en [89] son:

- Las aplicaciones RIA deben de soportar parte del procesamiento en el cliente, para reducir la comunicación con el servidor al mínimo.
- Deben soportar diferentes niveles de persistencia en el lado cliente y en el lado servidor.
- El procesamiento, creación, modificación y filtrado de datos puede ocurrir en ambos lados cliente y servidor.

- El uso en la Web y fuera de la Web debe ser posible.

Compañías dedicadas a la investigación como Garnet, IDC, Integration New Media, entre otras como Macromedia han destacado su importancia y esperan un amplio crecimiento y aceptación entre desarrolladores.

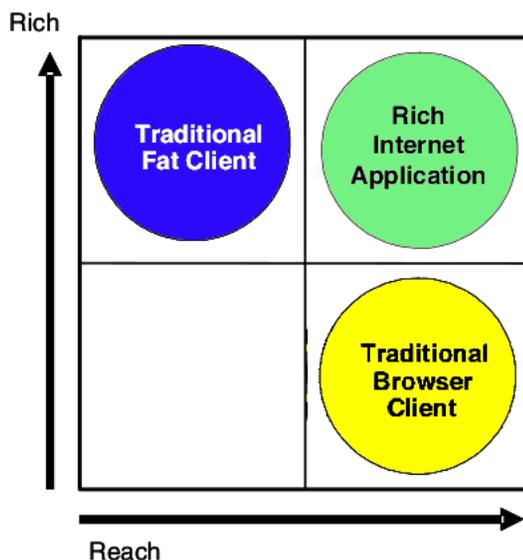


Figura 3.6: Representación visual de una aplicación RIA, en esta representación podemos observar cómo la unificación de una aplicación de escritorio (Fat client) y las tradicionales aplicaciones Web (Browser client), permiten obtener un nuevo tipo de aplicación conocida como RIA (Figura original de Driver en [90]).

Pero el crecimiento de estas aplicaciones enfrenta diversos problemas, ya que este tipo de aplicaciones presenta mucha complejidad en todas las fases de desarrollo, porque se carece de modelos, arquitecturas, técnicas, estandarización y metodologías de desarrollo. Razón por la cual, diversas aplicaciones de este tipo presentadas como RIA, en realidad sólo son aplicaciones Web disfrazadas de aplicaciones RIA, ya que no contienen ninguna de las características mencionadas anteriormente. Se espera en corto tiempo que la investigación en ingeniería Web, permita obtener algunas soluciones para el desarrollo de las aplicaciones RIA.

### 3.4. Interacción Visual-Dinámica

Una vez que se han mostrado las diferentes maneras de interacción en sistemas de escritorio, en esta sección desarrollamos una nueva propuesta de para interactuar con los sistemas Web, utilizando como base algunos de los conceptos descritos anteriormente. Como primer punto mostramos algunas características que son de nuestro interés adicionar en nuestro modelo básico y después describimos una nueva manera de interacción que será parte de nuestra propuesta final en esta tesis.

#### 3.4.1. Ventajas de interacción en sistemas de escritorio

En esta sección mostramos algunas de las ventajas que ofrecen los diversos enfoques utilizados para la interacción en sistemas orientados a escritorio, pero que queremos incorporar a un sistema orientado a la Web. Las ventajas están separadas en base a los diferentes enfoques, que van a ser usados en el nuevo modelo.

Las ventajas a utilizar son:

##### Consultas Visuales

- *Elementos visuales.*- Las representaciones visuales como: imágenes, iconos, botones, entre otras. Permiten la creación de una amigable interfaz facilitando la interacción.
- *Manipulación interactiva.*- La semántica de ciertos elementos visuales permiten representar entidades o propiedades de una base de datos, haciendo posible realizar diversas operaciones o consultas.
- *Secuencia de elementos.*- Los elementos de una consulta visual tienen una secuencia definida y necesaria para poder construir una consulta correctamente.

##### Consultas dinámicas

- *Actualización instantánea.*- La principal característica en las consultas dinámicas, es la actualización instantánea de la interfaz cuando una consulta es realizada.
- *Manipulación directa.*- La manipulación directa ofrece la ventaja a los usuarios de no tener que aprender una sintaxis o semántica que requiere un lenguaje de consulta.

- *Generación automática de consultas.*- Una consulta dinámica permite la construcción de una consulta a través de diversos controles como barras deslizantes, botones o casillas de verificación.

#### **Interfaz Multi-nivel**

- *Capas de interacción.*- La interfaz es dividida en diferentes niveles de especialización para la interacción. El usuario puede escalar cada una de las capas según sus propias necesidades.
- *Control de interacción.*- La división por niveles permite mantener un control sobre la interacción y el acceso del usuario al sistema.

Estas son las ventajas que nos interesa incorporar en nuestro nuevo modelo de interacción, por la parte de sistemas orientados a escritorio. Ahora veamos otras ventajas a adicionar por parte de la interacción de sistemas Web.

#### **3.4.2. Ventajas en interacción Web**

Como hemos observado las principales limitantes de los sistemas Web radican en la interfaz y el control de estados. La interfaz tiene una dependencia total de los navegadores, que a su vez mantienen diferencias entre ellos; generando diversos problemas, por lo cual las características que deseamos en nuestros sistemas Web para permitir adoptar nuestro nuevo modelo de interacción son:

- *Interfaz intuitiva.*- Es necesario evitar la complejidad de la interfaz y evitar la saturación de controles. Por ello es recomendable usar el principio *KISS* descrito anteriormente, así como la posible implementación de una interfaz multi-capa.
- *Interfaz ligera.*- Una interfaz ligera evita la sobrecarga en el lado cliente y evita que la interfaz contenga una cantidad importante de elementos no necesarios e introduzcan una complejidad no necesaria.
- *Software estándar.*- El uso de software estándar evita diferentes problemas de presentación y comportamiento en la interfaz. Además es posible evitar el uso de plugins, máquinas virtuales u otro software adicional externo, para su correcto funcionamiento en diferentes navegadores.
- *Control de estados.*- Mantener el control de los estados del sistema en el momento de la interacción, para saber el comportamiento que tendrá el sistema en la interacción.

- *Arquitectura abierta.*- Una arquitectura genérica y abierta, habilita una implementación con software que el desarrollador conoce y utiliza; evitando condicionar el uso de un software en particular en la implementación.

Estas son algunas de las principales características a mantener en los sistemas Web hacia los cuales está orientado nuestro modelo, el cual se presenta a continuación.

### 3.4.3. Modelo de interacción visual-dinámica

El nuevo modelo de interacción propuesto está enfocado a la parte de la interfaz Web y el usuario. La representación visual de este modelo computacional para la interacción puede verse en la figura 3.7, donde podemos observar las diferentes capas que se necesitan implementar para obtener algunas de las ventajas mencionadas en esta sección.

La figura 3.7 muestra en primera instancia a un usuario, el cual accede a una interfaz que puede contener diversos elementos visuales como: botones, barras de deslizamiento, iconos, entre otros.

Estos elementos visuales contienen una cierta semántica inmersa puntual, la cual es definida por medio de representaciones conceptuales que permiten el acceso a conjuntos específicos de datos. La semántica que pueden contener las diversas representaciones conceptuales es discutida en el capítulo siguiente.

La capa visual en la interfaz Web presenta los elementos visuales para que los eventos sean obtenidos a través de ellos. La capa dinámica mantiene procesos de escucha, captura y procesamiento de eventos en los elementos visuales. Un evento puede lanzar una solicitud de datos de forma interna o externa. Cuando se requiere una actualización interna, esta es dada solamente por los datos que ya están en el lado cliente y las modificaciones son hechas sin requerir de ninguna solicitud al servidor. Las actualizaciones internas pueden ser de creación, modificación, eliminación o presentación de los elementos en la capa visual.

Cuando es requerida una actualización de datos externos, entonces es necesaria una solicitud al servidor. La figura 3.7 muestra el lado servidor en la capa *Consulta A. (Consultas automáticas)*, esta capa verifica y valida las solicitudes para crear una consulta de manera automática según los parámetros recibidos.

Cuando la consulta es creada entonces se obtienen el acceso a la base de datos y se recupera la información, la cual es retornada a la capa dinámica para que pueda ser incrustada en la interfaz, a través de la generación de nuevos elementos visuales y ubicandolos en un área específica determinada.

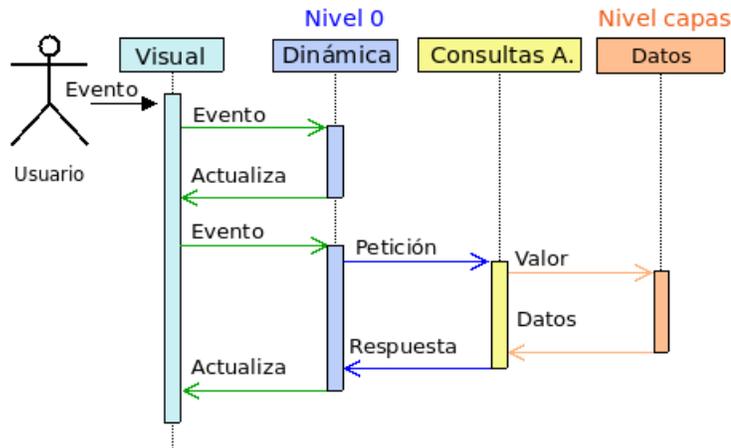


Figura 3.7: Modelo para interacción visual-dinámica, que permite capturar los eventos y realizar consultas de manera automática en una base de datos, después la interfaz es actualizada de manera dinámica.

Utilizando el concepto multi-nivel de Shneiderman, la actualización de datos interna es referida como *Nivel 0* y la interacción con datos del servidor en este momento es referida como *Nivel capas*, aunque más adelante tendremos nombres más específicos para diferentes capas por anexar.

Toda esta interacción debe mantener la interacción por medios visuales (consultas visuales), una actualización inmediata de datos (consultas dinámicas), interfaz multi-nivel que permita acceder a nueva información, mantener una interfaz ligera, software estándar, arquitectura abierta, independencia de navegador, librerías o software externo y sistema operativo.

Todas estas propiedades son casi imposibles de obtener en un sistema Web, pero para que esto pueda ser posible, también es necesario obtener una reestructuración en la administración de los datos que permitirán facilitar la interacción.

La administración de los datos comprende una nueva manera de explotar el esquema conceptual y el esquema lógico de la base de datos y mapearlos en

la interfaz de manera que permita obtener una representación visual, que permita un acceso transparente a los datos, parte de este proceso es descrito en el siguiente capítulo.

### 3.5. Comentarios Finales

En este capítulo se han descrito algunos enfoques que son usados en la interacción de los sistemas y que son de nuestro interés en este proyecto de investigación como: las consultas visuales, consultas dinámicas y la interfaz multi-nivel, todas ellas para sistemas de escritorio y además de presentar algunas de sus limitantes para usarlas en los sistemas Web.

También se mostró porqué la usabilidad es importante en los diseños de la interfaz para permitir crear un sistema exitoso. Por otro lado, al lector se le acercó al mundo de los sistemas Web y su evolución, para conocer cómo la interacción con los usuarios está cambiando rápidamente. Después nos enfocamos en la interacción Web y observamos algunas limitantes.

Por último, damos un modelo que toma las ventajas de todos los enfoques, para hacerlos trabajar de manera conjunta. El modelo presentado es un poco abstracto en este momento, pero será complementado con algunas capas de estructuración y acceso a datos en el siguiente capítulo.

## Capítulo 4

# Semántica y Metadatos

Para hacer realidad la interacción visual-dinámica en la Web, son necesarios algunos cambios estructurales en los datos. Una de las propiedades de las consultas visuales es que las representaciones contienen una cierta semántica. Parte de este enfoque lo usamos pero de manera diferente, además de orientarlo exclusivamente a datos. El concepto de metadato también es tratado en este capítulo, ya que lo utilizamos como enlace entre la semántica y el esquema lógico de una base de datos; para alcanzar este propósito tomamos el concepto de metadato y adicionamos algunas propiedades que permiten este enlace.

La primera parte de este capítulo describe el concepto de semántica y la segunda parte nos acerca al concepto de metadato. En la última parte del capítulo, describimos y definimos nuestro nuevo tipo de metadato al cual denominamos con el nombre de *metadato conceptual*, con el cual podemos habilitar un nuevo tipo de interacción con el usuario.

### 4.1. Semántica y Datos

Uno de los principales problemas con los datos científicos y biológicos, es que los datos contienen una semántica inmersa, esta semántica es difícil de extrapolar más allá del esquema conceptual; obtener una representación en la interfaz con cierta semántica es uno de los desafíos más difíciles de resolver. Esta sección está enfocada en describir la naturaleza de la semántica, así como algunas propiedades de relación y representación. La sección inicia con una breve discusión sobre la semántica en la informática, después analizamos con mayor profundidad algunos aspectos para la representación de la semántica.

#### 4.1.1. Semántica en sistemas de información

El término de *Semántica* se refiere a los aspectos del significado o interpretación del significado de un determinado dato, símbolo, palabra, lenguaje o representación formal como se describe en [91].

La semántica puede tener variaciones y ser versátil dependiendo del contexto en que se encuentra, además puede tener diferentes tipos de aplicaciones. La semántica también puede estar en diferentes niveles dentro de un sistema de información, por ejemplo puede estar en los datos [28], en archivos [39], en la estructura de datos [92], en las bases de datos [91], [93], en las consultas [40], en capas de interacción [94], en metadatos [95], etc.

La asignación de una cierta semántica a un dato es una tarea no trivial porque el mensaje semántico asignado a un dato puede ser muy versátil y para un humano puede ser captado de inmediato, pero para una computadora eso está lejos de ser cierto. Aún entre humanos el mensaje semántico dado originalmente, puede ser interpretado de una manera completamente diferente, ya sea en un mismo o en un diferente contexto. Esta discrepancia es comúnmente referida como: *Diferencia semántica (Semantic gap)*. Una descripción más detallada de este término la podemos ver en [39] y la podemos definir como:

**Definición 5.** Diferencia semántica.- *Es la falta de coincidencia entre la información que uno puede extraer de un dato y la interpretación que el mismo dato tiene para un usuario en una situación dada (Smeulders en [96]).*

Esto quiere decir que la descripción semántica de un dato puede tener una diferente interpretación a la dada originalmente; por lo que la obtención de una descripción lingüística de un dato puede ser en algunos casos una tarea imposible; esta es la descripción del problema fundamental de recuperación basada en la semántica de datos [96]. Muchas investigaciones están orientadas a tratar de obtener soluciones para este problema, pero son orientadas principalmente hacia la recuperación de archivos multimedia que contienen música, imágenes, vídeos, etc. Este enfoque trata de usar la semántica de diversas formas, básicamente a través de metadatos para definir o describir el contenido o ciertos rasgos distintivos de los archivos, como puede verse en [39], [40], [97], [98]. Esta manera de hacerlo, crea diversas limitaciones para tratar de describir la semántica contenida en los datos, algunas de estas limitaciones son mostradas a continuación.

#### 4.1.2. Limitación semántica en sistemas de información

Diversos sistemas de información han tratado de incluir a la semántica como parte del sistema. Este uso ha mostrado algunas limitaciones, las cuales son mostradas a continuación:

- *Respuesta a eventos.*- Existe un nulo soporte para que una representación semántica responda a los eventos del usuario, ya que básicamente su interacción es a través de formularios que realizan una consulta tradicional.
- *Descripción semántica.*- Obtener descripciones lingüísticas de un dato o información es una tarea muy difícil, en la inmensa mayoría de los sistemas de información se usan los metadatos como una información semántica y relacional (ver [39]).
- *Diferencia semántica.*- La interpretación de un dato dada por distintos usuarios, puede ser muy diversa y se pueden obtener interpretaciones distintas de cada uno de ellos. Factores como el contexto actual, contexto de obtención del dato o conocimiento previo, pueden variar en diferentes maneras su significado (ver [96]).
- *Mensaje semántico.*- La semántica dada a un dato o información puede tener diferentes interpretaciones y una computadora simplemente no lo entiende si no cuenta con un software que lo interprete; también entre humanos puede existir cierta discordancia en la interpretación, como se muestra en [96].
- *Sinónimos y homónimos semánticos.*- Cuando un término, información o dato contiene una semántica, pueden existir términos sinónimos que representen lo mismo y puede que el sistema no tenga relaciones explícitas entre estos dos términos. Por el otro lado, pueden existir términos idénticos pero que contienen una diferente semántica, pudiendo obtener y ocasionar relaciones anómalas (ver [39]).
- *Riqueza semántica.*- Es referida a cierto tipo de dato o información que contiene una amplia semántica inmersa, esto quiere decir que no puede ser restringida a una semántica específica porque esta es muy diversa. Muchos datos científicos en especial los relacionados a biología, son partícipes de este concepto como puede verse en [28].

Una vez mostradas algunas de las limitaciones de la semántica implementada en sistemas de información, mostramos a continuación una descripción de

la naturaleza misma de la semántica para obtener una nueva manera de aplicación en una interfaz. La semántica por si sola no puede resolver el problema de la exploración de datos, pero si aporta parte de la solución.

### 4.1.3. Semántica

La palabra semántica (*del griego semantikos “lo que tiene significado”*), se refiere a los aspectos del significado, interpretación del significado o por decirlo de otra manera, se refiere al estudio del significado.

La semántica nació como una disciplina empírica que estudiaba los distintos lenguajes naturales y estudiaba el modo en que los significados se atribuían a ciertas palabras, así como las modificaciones para un nuevo significado. *Saussure* que es considerado fundador de la lingüística moderna, define como significado al contenido mental que es dado a un signo lingüístico, el cual contiene dos componentes principales, el *significante* (designador) y el *significado* (imagen mental del significante).

El significado de un signo lingüístico puede tener muy diversas variaciones, Moreno en [99], describe como el significado puede cambiar dependiendo de una variación contextual, ya que el significado de las palabras puede variar en mayor o menor medida según el contexto en que aparezca; el significado puede verse como un concepto y es definido de esta manera:

**Definición 6.** Significado como concepto.- *El significado consiste en la relación entre las palabras y los conceptos o ideas o representaciones mentales; son estos conceptos los que se relacionan con los objetos de la realidad mediante la identificación de un conjunto de rasgos o propiedades compartidas (Moreno en [99]).*

Esta definición cuenta con una representación gráfica que permite tener una idea más clara de esta definición, esta representación se muestra en la figura 4.1, donde podemos observar el vínculo bidireccional que existe entre una palabra, un concepto y un objeto. Este vínculo nos permite mantener una relación conectiva del concepto, del significante y de la realidad. La representación de la figura 4.1, nos permite observar la conexión que permite establecer un concepto con objetos de una realidad. Esta vinculación es de suma importancia porque, como se muestra más adelante, nos permite mantener un enlace entre una semántica por medio de un concepto y datos en una base de datos.

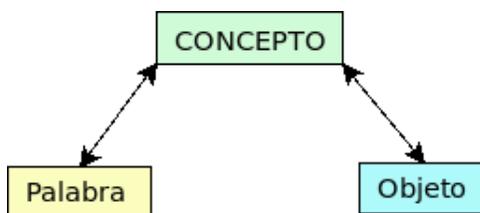


Figura 4.1: Representación gráfica de la definición 6, donde los conceptos se relacionan con los objetos de la realidad (figura original de Moreno en [99]).

Un vez descrito el significado como concepto, también debemos decir que el estudio del significado se divide en dos grandes bloques de estudio, los cuales son:

1. Semántica Léxica
2. Semántica Composicional

El primer bloque estudia el significado de las palabras aisladas y el segundo el significado de toda una oración. En nuestro caso, nos interesa en detalle el estudio de la semántica léxica, ya que nuestra investigación está orientada a tener una representación de esta semántica en una interfaz, para permitir una interacción transparente con los datos. A continuación describimos la semántica léxica con un poco más de detalle.

#### 4.1.4. Semántica léxica

La semántica léxica es el estudio de lo que denotan las palabras en una lengua natural, donde las palabras denotan entidades físicas del mundo o conceptos. La semántica léxica contiene unidades de significado, las cuales son conocidas como unidades léxicas. Es importante deducir si el significado contenido en una unidad léxica, es determinado por su posición y relación en un red semántica o este es localmente contenido en la misma unidad.

Moreno en [99], describe que las principales tareas de la semántica léxica son:

- Caracterizar el significado de las palabras.
- Dar cuenta de las relaciones de significado entre palabras
- Caracterizar distintos tipos de significado.

- Definir la variación contextual del significado.
- Describir el cambio semántico.

La semántica léxica contiene lexemas los cuales son definidos como:

**Definición 7.** Lexema.- *Un lexema es una unidad de significado que subyace a las palabras (Moreno en [99]).*

Esto quiere decir, que un lexema es la unidad de significado que contienen diversas palabras relacionadas morfológicamente. También esta unidad de significado puede no tener solo una palabra en concreto, esto nos observa una diferencia entre palabra y lexema. Esta diferencia es resaltante cuando se estudia la manera en que una palabra nos puede conducir a una ambigüedad léxica y llevarnos a fenómenos de *polisemia* y *homonimia*.

La polisemia está dada por la diversificación del significado que una cierta palabra puede adquirir, mientras que la homonimia expresa una confluencia o igualdad a nivel fónico y ortográfico, con significados diferentes no relacionados entre sí. Esta igualdad puede darse de las siguientes tres maneras: fónica/ortográfica, fónica/no-ortográfica y ortográfica/no-fónica, una discusión más detallada sobre este tema es dada por Moreno en [99].

Dentro de la semántica léxica existen diferentes tipos de relaciones léxicas entre lexemas mismos. Las relaciones de interés para nuestra investigación son: relaciones de sinonimia-antinomia, hiponimia-hiperonimia y holonimia-meronimia. Las definiciones de cada una de estas relaciones se establecen a continuación:

**Definición 8.** Sinonimia-antinomia.-*La sinonimia se establece cuando diferentes lexemas presentan significados idénticos en un mismo contexto y la antinomia es el significado opuesto a un cierto lexema.*

Esto nos dice que dos lexemas son sinónimos si la sustitución de uno por otro no altera el significado de la expresión, ya que presentan idénticos significados en los mismos contextos. La sinonimia puede no ser absoluta y es necesario verificar que el significado de un lexema sinónimo no cambie cuando el contexto sufra alguna variación o modificación. Por otra parte, un lexema también puede establecer un significado totalmente opuesto con respecto a otro lexema, a través de la antinomia.

Otra relación de gran interés en esta investigación es la relación de hiponimia-hiperonimia, con la cual podemos establecer ciertas relaciones de inclusión entre lexemas y la definimos de la manera siguiente:

**Definición 9.** Hiponimia-hiperonimia.- *Se trata de una relación de inclusión entre conjuntos y subconjuntos o entre clases y subclases. Al conjunto o clase incluyente lo denominamos hiperónimo y al conjunto o clase incluida, hipónimo (Moreno en [99]).*

De esta manera podemos observar que una hiperonimia ocurre cuando el significado de una palabra incluye los significados de otras. Es decir, una palabra hiperónima puede ser utilizada para referirse a una categoría más general, en tanto que las palabras hipónimas comparten un hiperónimo común, como si se tratase de una relación entre una clase y una superclase. La representación gráfica de esta relación la podemos ver en la figura 4.2, donde se observa como el hiperónimo en este caso el lexema microorganismo, actúa como un conjunto más general, del cual parten otros subconjuntos de microorganismos más específicos.

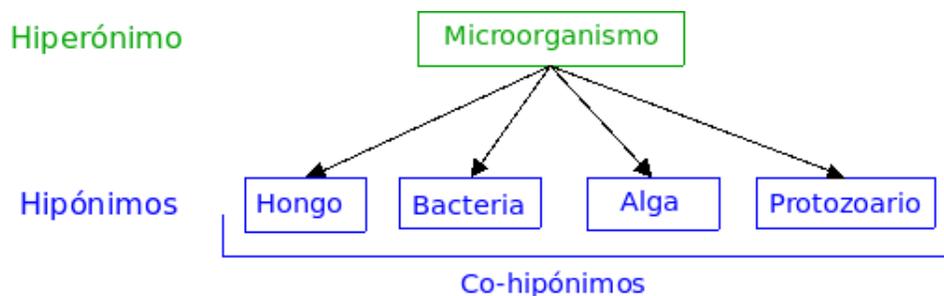


Figura 4.2: La figura muestra como existe una relación entre los co-hipónimos (parte inferior) y un hiperónimo (término más general).

Existe otra relación que aunque puede parecer muy similar es estrictamente diferente en su naturaleza, esta relación es conocida por el nombre de holonimia-meronimia y está basada específicamente en la inclusión y es definida de la siguiente manera:

**Definición 10.** Meronimia-holonimia.- *Esta relación está basada en una relación parte-todo. La relación es de inclusión, por lo cual una meronimia designa partes que están contenidas en una holonimia.*

Esta relación es más estricta, ya que podemos hablar de que un lexema X (merónimo), es una parte del lexema Y (holónimo) o que el lexema Y (holónimo) tiene X (merónimo). Esta relación establece una diferencia entre los merónimos y los hipónimos, ya que los merónimos son por definición partes de algo restringiendo la relación a la inclusión, de tal modo que no se puede hablar de subclases o subconjuntos como en la hiponimia. La figura 4.3, describe las partes merónicas de una holonimia como en este caso es una bacteria.

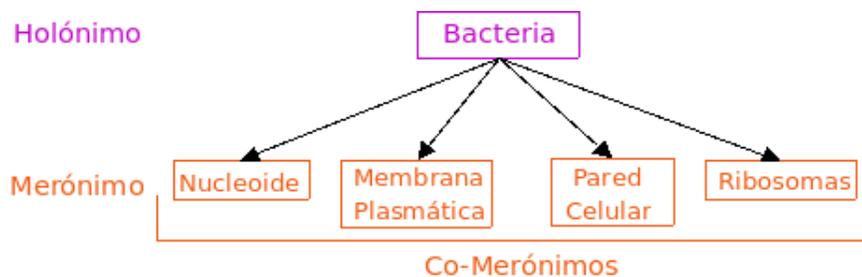


Figura 4.3: La figura muestra como una meronimia es parte de un holonimia, esta es una relación conocida como parte-todo.

Adicionalmente al estudio del significado de las palabras, la semántica léxica también incluye propuestas de clasificación por medio de jerarquías como las taxonómicas y las meronímicas. Las jerarquías taxonómicas están basadas en establecer diferentes categorías, con distintos niveles de especialización, mientras que las jerarquías meronímicas son aquellas en que la relación de dominancia expresa una relación parte-todo, sin expresar niveles de categorización.

Una vez descritas algunas de las relaciones de importancia en la semántica léxica, vamos a analizar otra herramienta que sirve en la representación de conocimiento, la cual lleva el nombre de mapa conceptual. De esta manera podemos obtener la representación de conocimiento con una relación de semántica en su interior. La siguiente parte está enfocada en describir la naturaleza de los mapas conceptuales.

#### 4.1.5. Mapa conceptual

Los mapas conceptuales tienen su origen en el departamento de educación de la universidad de Cornell en 1972, por Joseph D. Novak.

Los mapas conceptuales fueron basados en la teoría del aprendizaje significativo de David Ausubel, que trata sobre el aprendizaje de nuevos conceptos y conocimiento, los cuales son asimilados a través de la relación de conocimiento previamente adquirido.

Para entender que es un mapa conceptual, es necesario primero dar una definición sobre una de sus partes elementales más importantes y que es conocida como *concepto*, su definición es la siguiente:

**Definición 11.** Concepto.- *Es una unidad básica de conocimiento que puede contener una idea abstracta, una representación mental o representación abstracta de una experiencia, una estructura de información o razonamiento en nuestra mente.*

La manera en que un concepto es relacionado a un agrupamiento de información es a través de una etiqueta, la cual puede ser una palabra o símbolo. Los conceptos son utilizados para construir los mapas conceptuales los cuales son definidos como:

**Definición 12.** Los Mapas Conceptuales.- *Son herramientas gráficas para la organización y representación de conocimiento, que incluyen conceptos usualmente encerrados en círculos o algún tipo de cajas, las relaciones entre conceptos son indicadas a través de una línea que permite conectar dos conceptos (Novak en [100]).*

Un mapa conceptual es un instrumento que permite representar el conocimiento de una manera sencilla, práctica y sirve para transmitir con claridad mensajes conceptuales complejos, facilitando la comprensión, el aprendizaje y la enseñanza. Inicialmente los mapas fueron concebidos para el uso de la representación de conocimiento, pero llegó a ser una útil herramienta en la enseñanza, el aprendizaje y con un uso extendido en la investigación en: educación, negocios, industria, milicia, entre otros.

Los mapas conceptuales están formados por tres elementos fundamentales los cuales son: concepto, palabra de enlace y preposición. Los conceptos encerrados en un círculo o caja son llamados nodos, los cuales están unidos a través de una línea a otro concepto y señalan su relación con otro concepto

a través de una palabra de enlace; por último, las preposiciones están constituidas por los conceptos y las denominadas palabras de enlace, formando lo que se conoce como una unidad semántica.

Una de las principales características de los mapas conceptuales dadas por Novak, es que los mapas tienen una representación jerárquica, donde los conceptos más generales e inclusivos ocupan los lugares superiores en la estructura gráfica, mientras que los conceptos subordinados a conceptos más amplios deben estar estructuralmente abajo (ver figura 4.4).

La estructura jerárquica para un particular dominio de conocimiento depende del contexto, en el cual ese conocimiento es considerado. Por lo cual, la mejor construcción de un mapa conceptual, puede ser dada para responder a una pregunta en particular, en una situación o evento que estamos tratando de comprender a través de una cierta organización de conocimiento.

La figura 4.4 muestra un mapa conceptual, donde podemos observar como los conceptos más generales incluyen a otros conceptos más específicos y particulares. Cada concepto actúa como nodo que es enlazado por medio de palabras de enlace, que permiten establecer proposiciones para establecer una relación semántica entre conceptos.

La relación en un mapa conceptual es muy similar a la relación de hiponimia-hipernomia, pero esta relación semántica está basada en conjuntos y subconjuntos, en cambio la relación básica en un mapa conceptual utiliza conceptos que incluyen a conceptos más específicos, con lo cual la semántica representada es diferente en cada relación.

Todas las relaciones semánticas descritas y la relación de conceptos mostrada en el mapa conceptual, son una valiosa herramienta que nos permitirá mapear diferente tipo de semántica y conocimiento de toda la información contenida en una base de datos científica.

Toda la estructuración, representación y descripción de conocimiento, no sirve de mucho si no existe un enlace entre el usuario y la información. Para alcanzar este enlace podemos hacer uso de los metadatos, pero necesitamos utilizarlos de una manera ligeramente diferente.

En la siguiente sección describimos el uso de los metadatos actualmente y después en la sección final mostramos un nuevo tipo de metadato y cómo actúa como enlace.



Figura 4.4: Ejemplo de un mapa conceptual, la figura muestra los conceptos (nodos) y las palabras de enlace que señalan la relación entre nodos y las relaciones presentan una jerarquía.

## 4.2. Metadatos

Los metadatos son una de las partes más importantes en las bases de datos, en los sistemas y en los datos mismos. Pero los metadatos no tienen una única definición ya que dependen de su uso, el cual algunas veces puede ser muy variable.

Esta sección está enfocada en definir el término *Metadato* y describir algunas ventajas y limitaciones en diversos sistemas de recuperación de información. La primera parte de esta sección se enfoca en la definición y el uso práctico del concepto de metadato en diferentes sistemas de información, desarrollados bajo este concepto. En la parte final, describimos algunas de las limitaciones de los metadatos cuando trabajan como único enfoque en un sistema de información.

### 4.2.1. Definición de metadato

Los metadatos surgieron de la necesidad de describir, identificar y caracterizar diferentes objetos de información (datos, documentos, todo tipo de archivo, etc.) a través de etiquetas que facilitan su administración, gestión, búsqueda, localización y acceso entre otros usos.

El término *Metadato* no tiene una definición única; diferentes autores otorgan una definición a metadato según el contexto y propósito en el cual es utilizado en un sistema.

A continuación, mostramos algunas de las diferentes definiciones dadas al término de *Metadato* en la literatura científica; aunque las definiciones son muy similares, en un análisis profundo ya no lo son tanto.

Algunas de estas definiciones son:

- Datos sobre datos (ver [6], [36], [101], [102]).
- Información sobre información (Steinacker, et al. en [103]).
- Datos sobre el formato y el contenido de los datos (Günther en [104]).
- Datos estructurales sobre datos (Deng en [105]).

Otras definiciones más amplias orientadas a la administración lo describen como:

- Datos que describen un recurso físico o electrónico y que pueden ser utilizados para administrar colecciones de documentos, imágenes u otro tipo de información de un repositorio, tal como un archivo o museo (McClelland en [106]).
- Información descriptiva sobre recursos para el propósito de administrar, encontrar, usar y reutilizar estos recursos más efectivamente (Yin, et al. en [107]).

Pero la definición de *Metadato* que más se acerca a nuestro propósito en esta investigación es:

**Definición 13.** *Metadato.*- *Es una estructura de información que describe, explica, localiza, o de otra manera hace fácil de recuperar, usar o administrar un recurso de información. Además asegura que el recurso debe sobrevivir y continuar siendo accesible en el futuro (Smith, et al. en [98]).*

Como podemos observar, los metadatos están definidos por el uso práctico que se le da en la implementación de un sistema de información. Por ello, existen diferentes clases de metadatos y algunas veces pueden ser clasificados según su uso y pueden ser: *Descriptivos, Administrativos, Estructurales, Contextuales, etc.*

### 4.2.2. Uso de los metadatos

Hoy día, el uso de metadatos ha cobrado vital importancia y tienen amplio uso en dominios comerciales, académicos, educacionales, entre otros. En sistemas de información, los metadatos han logrado difundirse y han alcanzado una amplia aceptación y popularidad.

Los metadatos actualmente han logrado una estrecha relación con todo tipo de información digital, por ello son usados para obtener diccionarios de datos, administrar datos o hacer referencia a todo tipo de archivos digitales como por ejemplo: archivos de texto, música, imágenes, vídeos, etc. Algunos de estos ejemplos los podemos ver en [108], [98], [39], [36], [107], [109].

Actualmente existen diversos esfuerzos para tratar de obtener un estándar de metadatos. Pero su uso es ampliamente heterogéneo, lo que ha derivado en diversos tipos de estándares para diferentes propósitos como el intercambio. Por ejemplo para objetos de aprendizaje tenemos el estándar *Dublin Core* (ver [106]) y LOM (*Learning Object Metadata*, ver [103]), para intercambio de imágenes tenemos el estándar EXIF (*Exchangeable Image File*, ver [98]), ó el IMM (*Information Interchange Model*, ver [98]), por mencionar solo algunos. Aunque estos estándares pueden proveer algunas ventajas, estos no son utilizados en la inmensa mayoría de las veces, según los propósitos para los cuales fueron creados.

Hay que observar que la cantidad de metadatos que podemos adicionar a un dato, información o archivo es ilimitada; en otras palabras, podemos incluir una cantidad ilimitada de metadatos, que a su vez pueden ser tan variables y heterogéneos como el dato, archivo o información permita definirlo. Esto significa, que los metadatos pueden causar y generar más complejidad administrativa o descriptiva, que la generada por los datos mismos a los cuales se refieren. También puede implicar un mayor esfuerzo y generar complejidad en desarrollar un ambiente adicional administrativo de metadatos, adicional y diferente al desarrollado para la información contenida a la cual hacemos referencia.

Generar muchos metadatos puede dificultar en gran medida la distinción explícita entre los datos y los metadatos, porque los metadatos son datos en sí mismos. Además muchas veces también es posible, imprescindible o necesario crear metadatos sobre los metadatos. Algunas veces la obtención de metadatos sobre metadatos puede parecer exagerado o incluso absurdo,

pero estos metadatos son muy útiles para ciertos tipos de problemas, ya que pueden representar índices, relaciones, estructuras, conjuntos, descriptores semánticos, contextuales, o aplicados para otros usos, como puede verse en [109], [110], [111], [36], [35], [112]. Una de los principales usos de los metadatos en el área de bases de datos, es la construcción de diccionarios de datos; este uso es descrito a continuación.

#### 4.2.3. Metadatos como diccionario de datos

Uno de los primeros usos y aplicaciones de los metadatos en la informática, fue en la construcción de diccionarios de datos. Este uso es de mucha utilidad ya que ayuda a obtener parte de un control del sistema, de los datos y de la base de datos. El diccionario de datos en realidad es un conjunto de metadatos del cual podemos obtener una definición como:

**Definición 14.** Diccionario de datos.- *Es un conjunto de metadatos, que contiene diversas propiedades, detalles y descripción de los datos o elementos que son parte de un sistema.*

Los diccionarios de datos pueden ser construidos durante la fase de análisis del flujo de datos, diseño de la base de datos, diseño de interfaz y en diversas partes del desarrollo del sistema. Una de las principales aplicaciones del diccionario de datos, es establecer una completa compatibilidad entre todos los elementos que forman parte del comportamiento, adquisición, procesamiento, validación, seguimiento y almacenamiento de los datos.

Algunos de los beneficios de un diccionario de datos son:

- Información centralizada de los datos como nombre, tipo y relación.
- Definición del significado de los datos o elementos del sistema.
- Integridad de los datos e identificación de redundancias.

Una vez presentada la utilidad de un diccionario de datos, vamos a mostrar el uso de los metadatos en los sistemas Web y su reciente evolución en la clasificación de información.

#### 4.2.4. Metadatos en la Web

Los metadatos en la Web han basado mucha de su popularidad en la búsqueda y localización de datos, información o diversos tipos de archivos. Algunos ejemplos de este enfoque en sistemas Web son: IEEE Explore, Amazon, Wikipedia, por solo mencionar algunos.

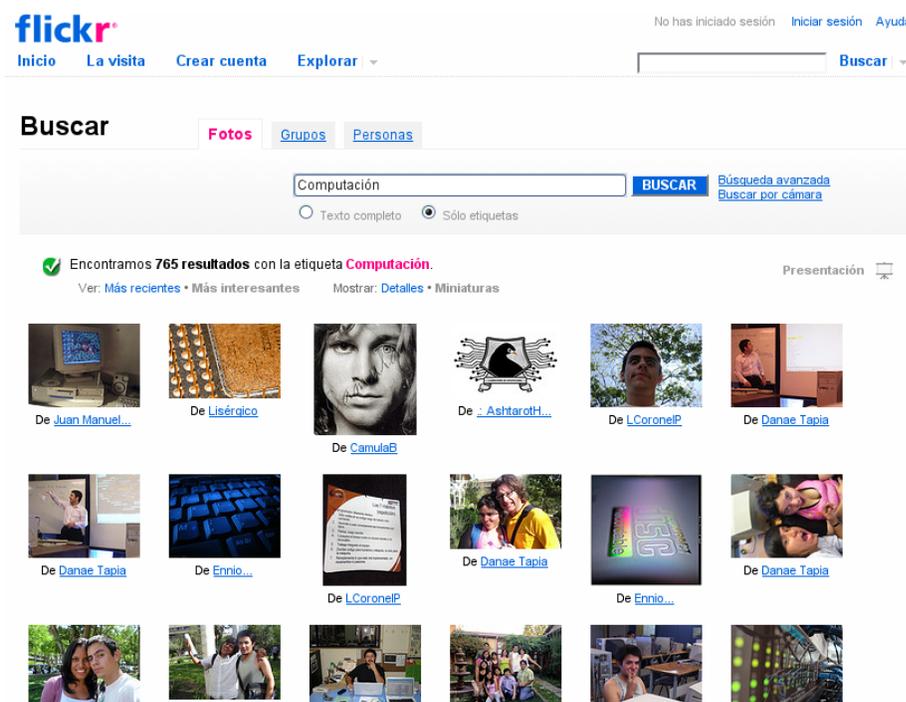


Figura 4.5: Uso de folksonomía en la exploración.

En el auge de la Web 2.0, muy diversos sistemas han implementado un sistema administrativo, basado en metadatos que los usuarios pueden incorporar de manera libre. Estos metadatos son usados para referenciar cualquier tipo de archivo o información, que ellos depositan para su acceso público a través de la Web. A su vez, los usuarios que localizan y encuentran esta información, también pueden adicionar más metadatos que ellos creen convenientes según el contexto en el que se encuentran. La incorporación de metadatos dada por diferentes usuarios de esa información, ha dado lugar al término conocido como *Folksonomía*.

El término *Folksonomía* ó *Folcsonomía* (*Folksonomy*), es atribuido a Thomas Vander Wal. La figura 4.5 muestra el concepto en uso.

Este término consiste en categorización y clasificación abierta y libre, sin seguir ningún orden o estructura jerárquica. Esto quiere decir que todos los metadatos (*etiquetas*) están a un mismo nivel de importancia y pueden crecer de manera espontánea, caótica, descoordinada y generando una cantidad enorme de relaciones.

Este nuevo paradigma de clasificación esta teniendo mucha popularidad en la Web social. Ejemplos de su utilización pueden ser encontrados en sitios Web como Flickr, del.icio.us. y YouTube (ver [73], [82], [84]).

Una vez obtenidos algunos de los usos prácticos de los metadatos y parte de su actual importancia en la Web, es necesario conocer algunas de sus limitaciones, las cuales son descritas de manera breve a continuación.

#### 4.2.5. Limitaciones de los metadatos

Algunas de las principales limitaciones que tienen los Metadatos son:

- *Limitación descriptiva.*- Los metadatos que son utilizados para tratar de describir contenido, muchas veces solo dan una idea muy vaga sobre ese contenido. Por ejemplo, la cantidad inmensa de metadatos que pueden ser necesarios para describir el contenido de fotos o vídeo, da una muestra de qué tan difícil es obtener las características descriptivas y conceptuales de este material. Además, diferentes personas otorgan diferentes metadatos a una misma información, haciendo más difícil una descripción única (ver [84], [98]).
- *Metadatos ilimitados.*- Es siempre posible adherir más y más metadatos útiles y no hay un límite. Obtener muchos metadatos de un cierto dato o archivo, puede generar o necesitar un complejo sistema administrativo para manejar una cantidad quizás innecesaria de metadatos, los cuales pueden no ser utilizados en ciertos contextos (ver [109], [102]).
- *Limitación referencial.*- Cuando los metadatos son usados como referencia, no es posible conocer relaciones explícitas entre el contenido y los datos. Muchas veces la referencia por metadatos está dada por los usuarios bajo un contexto determinado y puede no tener ninguna relación referencial del metadato, para otro usuario en un diferente contexto (ver [110], [111]).

- *Limitación semántica.*- Los metadatos contienen información, pero no tienen un significado explícito; esto puede generar cierta confusión sobre cómo es interpretada la información del metadato, por los diferentes usuarios (ver [35]).

Después de conocer algunas de las limitaciones de los metadatos, ahora podemos dar una nueva propuesta de uso que permita librar algunas de estas limitantes.

### 4.3. Interacción con Metadatos Conceptuales

Una vez obtenido una descripción de la semántica léxica, junto con el concepto de mapa conceptual y de metadatos, nuestro siguiente paso es conjuntar algunas de las mejores características de cada concepto, para de esta manera crear una nueva forma de interacción entre una semántica, una representación conceptual y un conjunto de metadatos.

La primera parte de la sección está enfocada en describir diferentes tipos de relaciones semánticas a través de un mapa conceptual. Después definimos un nuevo tipo de metadato, que permitirá la creación de un nuevo modelo de interacción. Por último se describe este nuevo modelo y como puede permitir un acceso transparente y guiado de datos.

#### 4.3.1. Relación semántica en mapas conceptuales

Como se describió anteriormente, un mapa conceptual es una herramienta de organización y representación de conocimiento, que utiliza conceptos para alcanzar esta representación. Cada uno de esos conceptos es una unidad básica de conocimiento, que también puede tener representaciones mentales las cuales están descritas en las definiciones 6 y 11. Los mapas conceptuales establecen una relación de inclusión y representación jerárquica donde los conceptos más generales contiene a otros menos inclusivos. La relación de hiponimia-hiperonimia de la semántica léxica establece una relación similar, pero la inclusión está basada en subconjuntos.

La idea principal en este punto es obtener una semántica con ciertas relaciones encapsuladas en conceptos y representadas u organizadas en mapas conceptuales. De esta manera, podemos establecer diversas relaciones semánticas y describirlas por medio de conceptos relacionados al área de estudio a la que pertenecen los datos.

Los conceptos utilizados deben contener una jerarquía que permita establecer una relación de inclusión y permitir el acceso a la información a través de una descripción de la información contenida en una base de datos.

La representación visual de esta idea es mostrada en la figura 4.6, donde podemos observar la estructura básica de un mapa conceptual, donde existe una jerarquía entre conceptos, pero en esta idea los conceptos pueden contener en su interior diversos tipos de relaciones semánticas, dadas por un área de especialización de los datos.

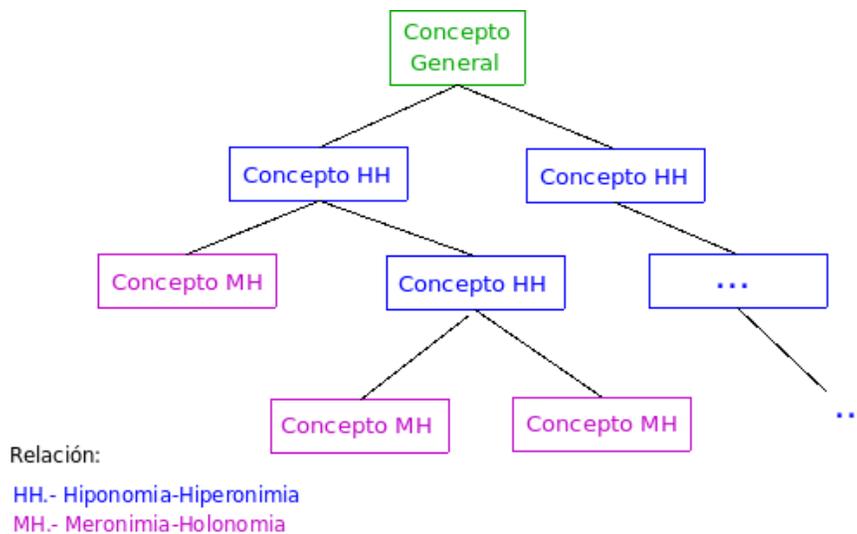


Figura 4.6: Mapa conceptual semántico, que contiene una relación entre diferentes conceptos, donde cada concepto tiene una relación semántica inmersa en su interior.

La figura 4.6 muestra en este caso dos tipos de conceptos, uno denominado como concepto HH que representa una relación de Hiponimia-hiperonimia y los conceptos en los nodos finales representados como conceptos MH, que representan la relación Meronimia-holonimia, donde tenemos el concepto de parte-todo. Es decir, se llega a un nodo final que nos mostrará partes (datos puntuales en la base de datos), relacionadas a un concepto en específico. Esto habilita una manera controlada para explorar los datos, pero necesita de una manera práctica de enlazar el mapa conceptual con los datos reales en una base de datos; nuestro enlace puede ser un metadato, pero con diferentes propiedades como se muestra a continuación.

### 4.3.2. Metadatos conceptuales

Como se mencionó, un metadato es definido según el propósito y uso que tiene en un cierto sistema. Dada la definición 13 sobre metadato, observamos que aunque esta definición es muy completa necesitamos algunos cambios según nuestro propósito, el cual está orientado a formar un enlace entre el mapa conceptual y los datos en una base de datos.

Para alcanzar este propósito, es necesaria la creación de un nuevo tipo de metadato que denominamos con el nombre de *Metadato conceptual*, del cual damos la siguiente definición:

**Definición 15.** Metadato Conceptual.- *Es un metadato que sirve como enlace y referencia para explorar y recuperar datos, a partir de conceptos contenidos en un mapa conceptual semántico y catálogos (diccionario de datos) contenidos en una base de datos, permitiendo un acceso transparente a datos especializados.*

El metadato conceptual nos permite crear un enlace entre la descripción semántica conceptual, que puede ser mapeada a un usuario en una interfaz y una base de datos científica.

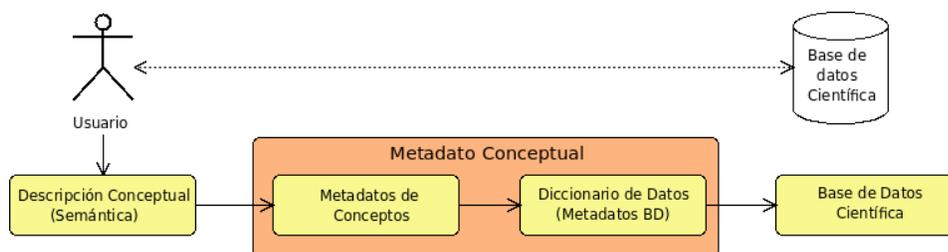


Figura 4.7: El usuario percibe un acceso transparente de los datos, debido al enlace que realizan los metadatos conceptuales.

La figura 4.7, nos muestra como el metadato conceptual habilita un acceso transparente a los datos, ya que el usuario tiene interacción a través de los conceptos, los cuales son referenciados internamente a catálogos y datos de una base de datos científica.

Los metadatos conceptuales pueden actuar como una caja negra a la vista del usuario, aunque internamente funcionan como metadatos de los conceptos y como metadatos de los datos contenidos en la base de datos. De manera abstracta podemos obtener un modelo de interacción basado en estos metadatos conceptuales, como se describe a continuación.

### 4.3.3. Modelo basado en metadatos conceptuales

Una vez que hemos obtenido una representación semántica y conceptual, con un enlace a través de los metadatos conceptuales, es conveniente regresar a nuestra definición 2, donde se presenta a la interfaz multinivel, propuesta por Shneiderman.

Debemos recordar que esta interfaz solo fue implementada en sistemas de escritorio, pero ahora nosotros deseamos extenderla hacia sistemas orientados a Web. Para alcanzar este propósito, necesitamos dividir la representación semántica en un primer nivel, los metadatos conceptuales en un segundo nivel y un tercer nivel dado por la base de datos.

Si nosotros podemos llevar el mapa conceptual obtenido a la interfaz y hacer que responda a ciertos eventos lanzados por el usuario, entonces se puede habilitar un acceso puntual a diversos datos especializados y referenciados por un concepto MH en un nodo terminal del mapa conceptual.

Para mostrar esta manera de acceso a los datos, creamos y proponemos un modelo de interacción basado en el uso de metadatos conceptuales; la representación gráfica de este modelo, podemos observarla en la figura 4.8. El modelo está dividido en diferentes niveles de interacción con el usuario, el primer nivel contiene el mapa conceptual con representación de conocimiento; este nivel contiene cierta semántica de la información en la base de datos. El segundo nivel contiene a los metadatos conceptuales, a este nivel lo denominamos como nivel *meta-conceptual*. Desde este nivel de metadatos, podemos obtener acceso directo a ciertos catálogos y datos de la base de datos, también podemos hacer una selección de ciertos datos de interés, los cuales son representados en el modelo como metadatos conceptuales. Después de seleccionados estos datos, se deben de generar consultas especializadas de manera automática a la base de datos científica, la cual se encuentra en el tercer nivel de interacción.

Para que toda esta interacción pueda llevarse a cabo, es necesario extrapolar una representación del mapa conceptual obtenido a la interfaz, la cual debe reaccionar a los eventos que ocurren en la representación del mapa.

El usuario debe primero obtener una interacción con el primer nivel, donde solo navega entre los conceptos hasta llegar a un nodo terminal; este finalmente cambia el nivel y muestra datos reales contenidos en la base de datos; al obtener una selección de ellos se pueden realizar consultas de manera automática a la base de datos.

El modelo basado en metadatos conceptuales propuesto parece que resuelve algunas de nuestros principales objetivos, tal como facilitar la navegación y exploración de datos científicos; pero llevarlo a una implementación en un sistema real, puede ser una tarea imposible. Para que el modelo pueda ser incorporado son necesarias algunas modificaciones que permitan su viabilidad en un desarrollo Web. La propuesta final de nuestro modelo es presentada en el siguiente capítulo.

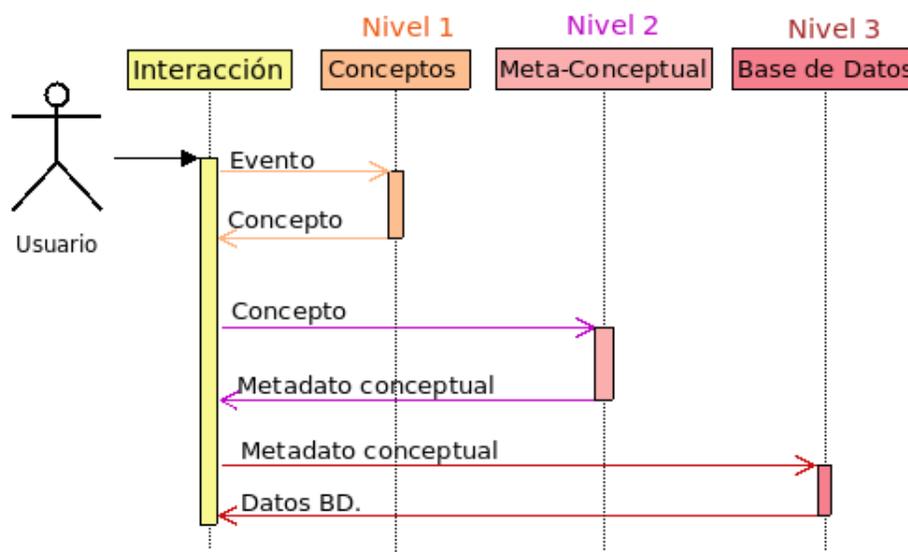


Figura 4.8: El modelo de interacción basado en metadatos conceptuales, habilita una exploración puntual y controlada de los datos en un base de datos.

#### 4.4. Comentarios Finales

El presente capítulo mostró inicialmente una descripción sobre el uso de la semántica en sistemas informáticos, así como las limitaciones que tiene este enfoque en la exploración. Después nos sumergimos a las entrañas de la semántica léxica, donde obtuvimos relaciones entre lexemas que son de suma utilidad en nuestro proyecto. Estas relaciones sirven para representar relaciones semánticas, que se encuentran en la información de diferentes áreas de especialización científica. Para describir esta semántica, analizamos algunas de las ventajas que nos ofrece un mapa conceptual.

La jerarquía de conceptos que caracteriza a un mapa conceptual, permite obtener una estructura de navegación de la información controlada, pero la estructura necesita de un enlace entre la conceptualización y los datos reales. Para este propósito definimos un nuevo tipo de metadato al que denominamos como metadato conceptual. Con este metadato logramos un enlace entre el usuario y los datos, permitiendo una navegación por niveles, que logra un acceso puntual a cierta información. Todo este proceso es explicado con un modelo de interacción basado en metadatos conceptuales. Pero este modelo, no es lo suficientemente completo para poder ser implementado en sistemas reales.

Nuestro siguiente objetivo es extender el modelo por medio de una integración con el modelo visual-dinámico, el cual también ha sido propuesto en esta investigación en el capítulo anterior. La fusión de estos dos modelos, nos permite obtener otro nuevo modelo más robusto y con amplias capacidades; ya que fusiona la parte estructural semántica de los datos, con la parte de amplia interacción dinámica con los usuarios. El siguiente capítulo está orientado a definir y describir con mayor profundidad este nuevo modelo.

## Capítulo 5

# Modelo Meta-conceptual Dinámico

Una vez dedicado un capítulo a la interacción en los sistemas y otro que muestra cómo una estructuración semántica puede permitir un mejor acceso a la información en una base de datos. Ahora en el presente capítulo proponemos una nueva forma de interactuar con la información a través de tres componentes principales: rica interacción de datos, conceptos semánticos y metadatos conceptuales, los cuales son la esencia de una nueva propuesta que denominamos como *Modelo Meta-Conceptual Dinámico (Modelo MD)*, el cual está orientado a obtener una consistente interacción y exploración en una base de datos científica. El nuevo modelo surge de la integración de los dos modelos propuestos en los capítulos anteriores. La primera parte está enfocada en definir y describir las capas y niveles que lo integran, en la segunda parte nos acercamos a la estructuración y conceptualización de la información para la interacción con los usuarios.

### 5.1. Presentación del Modelo

Esta sección presenta un nuevo modelo de interacción y exploración de datos, la sección pretende obtener una profunda descripción de todas sus partes internas, así como el comportamiento interno dado por los eventos realizados por un usuario en la interfaz. La sección está dividida en tres partes. La primera presenta la esencia del modelo, la segunda describe las capas que lo integran y la tercera parte describe cómo se obtienen los diferentes niveles de interacción.

### 5.1.1. Modelo Meta-conceptual Dinámico

En el capítulo 3 se presentó una descripción de la usabilidad y el enfoque de interacción que presentan diversos sistemas, de los cuales obtuvimos algunas características de nuestro interés. La integración de estas características nos permitió obtener un nuevo modelo que denominamos como: *Modelo de interacción visual-dinámica*, que conjunta algunas de las propiedades de las aplicaciones RIA y trata de obtener rasgos esenciales de tres tipos de consultas: visuales, dinámicas y automáticas. A partir de esta integración el modelo de interacción visual-dinámica obtiene tres principales componentes los cuales son: *elementos visuales de interacción*, *actualización instantánea* y *construcción de consultas automáticas*, pero el componente faltante es la estructuración interna de los datos y su representación visual a través de una interfaz.

Por otro lado, en el capítulo 4 nos enfocamos en obtener una manera para representar ciertas relaciones semánticas. En la semántica léxica encontramos algunas relaciones semánticas entre lexemas, que contienen una amplia capacidad de representación semántica. Después presentamos cómo los mapas conceptuales nos ayudan a estructurar el conocimiento por medio de conceptos. Esto nos ayudó a develar otro modelo basado en la estructuración de la información y que necesitaba de una herramienta de enlace, la cual definimos y denominamos como *Metadato conceptual*.

El modelo nuevo que obtenemos de este otro enfoque, lo denominamos como *Modelo basado en metadatos conceptuales* y pretende hacer una interacción basada en mapas conceptuales semánticos presentados al usuario, pero no específica como puede llevarse la extrapolación del mapa conceptual a la interfaz. Podemos observar que cada modelo presenta diferentes puntos de vista, uno basado en la interacción e interfaz y el otro en la estructuración semántica de los datos.

La integración de éstas dos diferentes perspectivas permiten la creación de un nuevo enfoque en la interacción de sistemas Web y que denominamos como *Modelo Meta-conceptual Dinámico (Modelo MD)*, que muestra ser robusto, completo y con amplias ventajas de utilización para el desarrollo de potentes sistemas de exploración. También su enfoque es obtener un acceso transparente a los datos, sin importar la complejidad semántica inmersa de la base de datos. Para alcanzar este objetivo, hace uso de un mapa conceptual que contiene conceptos semánticos que son enlazados por medio de los metadatos conceptuales.

A su vez, los mapas deben mantener una representación en la interfaz y responder de manera inmediata a los eventos del usuario en la interfaz. La definición que damos de este modelo es la siguiente:

**Definición 16.** Modelo MD.- *El modelo MD integra la semántica conceptual que contienen los datos de una base de datos, con una rica interacción visual-dinámica en la interfaz, obteniendo una manera eficaz, transparente y controlada, para navegar y explorar entre los diversos tipos de datos contenidos en una base de datos.*

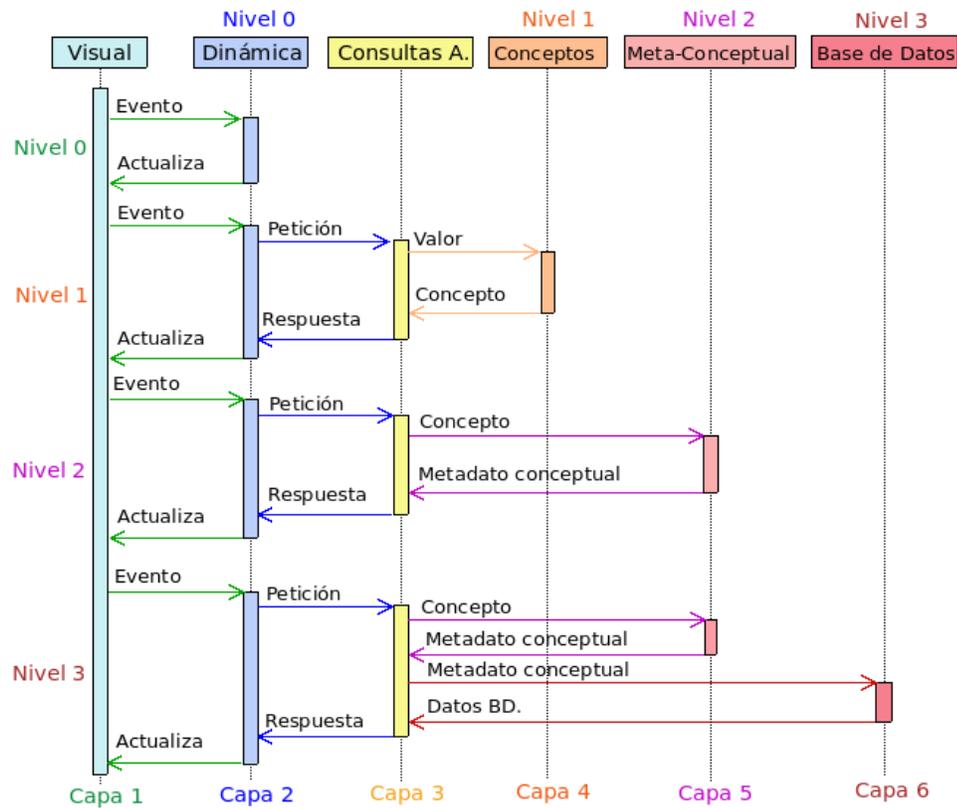


Figura 5.1: Diagrama de Secuencia que muestra todas las capas y niveles de interacción, que pertenecen al Modelo MD.

La representación gráfica del modelo es presentada en la figura 5.1, donde podemos observar las diferentes capas y niveles que están interactuando entre sí para responder de manera inmediata a los eventos de un usuario.

El modelo MD cuenta con seis capas co-dependientes, las primeras tres capas pertenecen al modelo visual-dinámico y las otras tres al modelo basado en metadatos conceptuales. Es fácil observar en la figura 5.1, la integración de los dos modelos mencionados.

El modelo MD está enfocado a sistemas orientados a la Web, por lo que las primeras dos capas denominadas como visual y dinámica pertenecen al lado cliente, mientras que las otras cuatro restantes pertenecen al lado servidor. Por otro lado, también cuenta con cuatro diferentes niveles de interacción, donde cada nivel representa un diferente tipo de acceso que es controlado desde la interfaz; los niveles están relacionados con el acceso a los diferentes tipos de abstracción que representan cada una de las capas, por lo que la esencia y descripción de cada una de ellas es dada a continuación.

### 5.1.2. Capas en el modelo MD

Como se ha descrito, el modelo MD está dividido en diferentes capas que tiene funciones muy específicas, pero permiten un comportamiento armonico entre ellas, con el fin de facilitar la interacción de los usuarios. Las capas mantienen una comunicación y colaboración entre sí, manteniendo un acceso controlado y estructurado de la información en la base de datos.

Cada capa debe ser implementada de manera diferente, debido a que realizan diferentes cosas con diferentes propósitos. Obviamente contienen una lógica interna diferente, además de ser implementadas con diferente software y lenguaje. Con todo esto; es importante mantener el traslado de datos de manera eficiente e integral en todo momento. A continuación; se presenta una descripción interna de las capas en base a su función interna dentro del modelo:

- *Capa Visual.*- Esta es la primera capa del modelo MD, esta capa debe contener elementos visuales y gráficos que permitan la interacción con el usuario. Los elementos visuales pueden ser iconos, botones, barras deslizantes, casillas de verificación, imágenes, etc.

Los elementos visuales de esta capa, deben ayudar a representar el mapa conceptual semántico que contiene la capa llamada conceptos. El mapeo de estos datos en la interfaz puede ser por medio de texto, imágenes, iconos, etc. La capa visual está encargada de presentar una interfaz intuitiva y responder a los eventos del usuario permitiendo la presentación de los datos obtenidos.

- *Capa Dinámica.*- Esta capa es la encargada de mantener escuchas en todos los elementos de la capa visual. También es la encargada de la actualización interna de todos los elementos y componentes visuales en la interfaz. Cuando los eventos son capturados de elementos que tienen escuchas, la capa debe verificar el nivel de interacción y acceso que le corresponde; es decir, la capa mantiene el comportamiento del sistema. Esta capa controla la actualización de los datos de manera local o por medio de peticiones externas (al servidor). Cuando la actualización es de manera local, sólo utiliza los datos contenidos en el cliente y los modifica según el requerimiento solicitado. Si la actualización necesita de datos externos entonces hace una petición asíncrona al servidor, obtiene los datos y los formatea para su inclusión en la interfaz. Esta inclusión puede necesitar de la creación de nuevos elementos visuales, los cuales son creados en tiempo de ejecución y generados sus respectivos escuchas.

El nivel de interacción 0 en esta capa, nos indica que los eventos del usuario solo requieren una actualización local; cuando el nivel de interacción capturado es 1,2 o 3, entonces necesitamos una actualización con datos externos los cuales son requeridos al servidor.

- *Capa Consultas Automáticas.*- Esta capa se encuentra en el lado servidor y recibe peticiones de interacción externas solicitadas por la capa dinámica. En realidad esta es un administrador de consultas, que dependiendo del nivel de interacción solicitado construye consultas a las capas correspondientes, también mantiene un sistema de validación de peticiones y verificación de parámetros y permite el acceso a las tres capas restantes: conceptos, meta-conceptual y base de datos.

Esta capa mantiene comunicación con la capa dinámica desde donde recibe diferentes solicitudes de datos y retorna los datos obtenidos las consultas realizadas. Además actúa como enlace entre el lado cliente y servidor, por lo cual debe contar con diferentes algoritmos de verificación, validación, e integridad de los parámetros enviados, para que se puedan construir las consultas de manera automática con otros algoritmos de acceso a las diferentes capas.

- *Capa Conceptos.*- El nivel de interacción de esta capa es el 1 y puede ser vista como una exploración de alto nivel, ya que la navegación por los conceptos nos permite acceder a datos muy puntuales. Esta capa contiene uno o varios mapas conceptuales basados en nodos con conceptos semánticos.

Cada mapa conceptual requiere de una representación gráfica, la cual puede ser obtenida a través de un lenguaje de modelamiento como UML, donde se pueden utilizar diagramas de clases, asociaciones, entre otros. Con estos diagramas se pueden obtener las diferentes representaciones semánticas y conceptuales, de la información contenida en la base de datos. La naturaleza de cada una de estas representaciones es un mapa conceptual, ya que utilizamos conceptos en cada nodo y con ellos permitir una navegación y exploración de datos. Los nodos finales de cada rama del mapa conceptual, deben mantener una correspondiente relación de holonomía en el nodo final y de meronimia en los catálogos correspondientes a la base de datos, de la capa meta-conceptual. La representación gráfica puede mapearse a una base de datos o en su caso es necesario crear un sistema administrativo de los mapas conceptuales.

Cada mapa conceptual debe mantener una representación en la capa visual y obtener actualizaciones de la capa conceptos, donde se encuentran todos los elementos de los diferentes mapas conceptuales. Cada mapa conceptual obtenido, debe mantener una correspondencia con el diseño del esquema conceptual de la base de datos, para una mejor integración con el modelo MD.

- *Capa Meta-conceptual.*- Esta capa mantiene el nivel de interacción 2 y representa una exploración de nivel medio, ya que en esta capa se obtiene el acceso a los catálogos de la base de datos. La manera en que se accede a estos datos, es por medio de los metadatos-conceptuales, los cuales son el enlace entre los mapas conceptuales y los catálogos del esquema lógico de la base de datos. Los metadatos conceptuales actúan como apuntadores y referencias semánticas cruzadas, entre los conceptos y la estructura interna de la base de datos. Esta capa es el núcleo principal de todo este modelo, porque mantiene todo el enlace entre la representación visual, la representación semántica y la base de datos.
- *Capa Base de Datos.*- Esta capa mantiene el nivel de interacción 3 y puede ser vista como una exploración a bajo nivel. Esta capa es la fuente de datos que contiene la base de datos donde necesitamos hacer la exploración y navegación de los datos. Esta capa se encarga de responder a peticiones muy específicas realizadas por la capa de consultas automáticas, para que estas consultas puedan hacerse de manera automática; la capa meta-conceptual obtiene la construcción válida de

esta consulta a través de los metadatos-conceptuales, el acceso a esta capa es controlado por la capa de consultas automáticas, ya que la validación en esta capa habilita el acceso a los datos. La base de datos contenida en esta capa debe mantener un esquema lógico basado en catálogos de datos, ya que estos catálogos facilitarían la integración con el modelo MD. El esquema conceptual aunque complejo debe estar bien definido, porque a partir de este esquema se debe obtener una abstracción en conceptos, para que pueda ser usado por la capa del mismo nombre.

Cada una de las capas debe mantener una comunicación transparente y eficaz, ya que deben reaccionar de acuerdo a los diferentes eventos realizados por el usuario en la parte visual; algunos de estos pueden disparar acciones en cascada entre las diversas capas pertenecientes al modelo MD, como lo muestra la figura 5.1. Las capas intermedias se comunican mediante el envío de parámetros para solicitar información y realizar procesos con la información obtenida. La interacción entre las diversas capas en el lado servidor y el lado cliente adiciona al modelo MD, una mayor capacidad de interacción y respuesta. Una vez descritas las capas, es necesario describir los distintos niveles de interacción.

### 5.1.3. Niveles en el modelo MD

Los niveles de interacción tienen una correspondencia muy cercana con la interfaz, en el modelo MD tenemos 4 diferentes niveles de interacción y cada uno de estos tiene un reflejo en la interfaz. De esta manera, cada nivel representa diferentes estados de navegación e interacción por parte del usuario y los datos.

Los diferentes tipos de niveles son:

- *Nivel 0.*- Este nivel de interacción sólo es realizada con los datos que se encuentran en la parte cliente. Todas las actualizaciones son internas sin requerir de datos externos y son ejecutadas por un lenguaje script cliente.
- *Nivel 1.*- Este nivel requiere de datos externos que se encuentran en el servidor. Estos datos requeridos son conceptos que contienen una semántica y una descripción del concepto. Este nivel es usado para navegar de conceptos muy generales a conceptos más específicos y concisos.

- *Nivel 2.*- Este nivel presenta catálogos de datos contenidos en la base de datos, estos son seleccionados a partir de conceptos semánticos en el nivel anterior. En este nivel 2, se seleccionan los valores de búsqueda para construir las consultas específicas a la base de datos. Los parámetros seleccionados permiten una eficaz exploración de datos, sin necesidad de conocer la estructura interna de la base de datos.
- *Nivel 3.*- Este nivel contiene y muestra los resultados de las consultas a la base de datos. Todos los resultados permiten nuevas consultas para obtener detalles más específicos de la información.

Cada uno de estos niveles debe tener un lugar y una representación en la interfaz. Por lo cual, la interfaz debe ser simple (ver figura 3.5), intuitiva y ligera. La intención de incorporar el diseño multinivel de Shneiderman, es para mantener un acceso muy específico a ciertos datos especializados y difíciles de encontrar en una base de datos. La utilidad de este diseño en la interfaz sera evidente en nuestro caso de estudio, en el capítulo 7.

Por otra parte, podemos decir que el modelo que usamos y definimos en esta sección, tiene un contexto de uso mucho más amplio y puede ser usado para diversos tipos de procesamiento e interacción con bases de datos como: exploración de datos, filtrado de datos, procesamiento de datos, simulación, minería de datos, etc.

Aunque todavía no existen aplicaciones con el modelo MD para estos tipos de procesamiento de información; podemos afirmar que esto es posible, basándonos en nuestra experiencia adquirida en este proyecto de investigación orientado a la exploración de datos.

## 5.2. Conceptualizando los Datos

El modelo MD necesita de una conceptualización eficaz de los datos para su buen funcionamiento. Como se explicó en la sección anterior, es necesario construir mapas conceptuales que contengan conceptos con cierta semántica en sus nodos, algunas veces puede no ser una tarea fácil obtener los mapas conceptuales de esta manera. Esta sección se enfoca en describir como puede ser generados los mapas conceptuales, partiendo desde el esquema conceptual de una base de datos, después se muestra como estos mapas pueden ser integrados por medio de los metadatos conceptuales, para lograr el acceso transparente de datos.

### 5.2.1. Base de datos científica

La construcción de una base de datos con información científica, puede tener diferente grado de complejidad y depende de las características de los tipos de datos contenidos como: la semántica de los datos, relaciones directas, relaciones indirectas, variabilidad de los datos, histórico de datos, descripciones contextuales, complejidad en la nomenclatura, etc. Todas estas características repercuten directamente en el diseño de la base de datos, la cual debe capturar las propiedades *estáticas* (contenido de la información) y *dinámicas* (operaciones de procesamiento) (ver [113]). Las propiedades estáticas son las que nos interesan en este momento y para capturarlas es necesaria la abstracción a través de un esquema conceptual, que permite capturar una cierta semántica del mundo real (ver [114]).

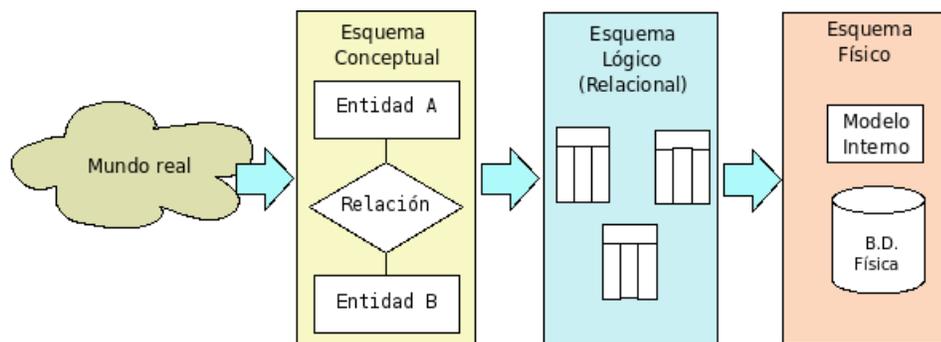


Figura 5.2: Diferentes tipos de esquemas necesarios, para la construcción de una base de datos, a partir de una abstracción del mundo real.

En la figura 5.2, podemos observar cómo la abstracción del mundo real modelada en un esquema conceptual, nos permite capturar cierta semántica relacional de alto nivel. Castaño et al. señala en [114], que mucha de esta semántica es perdida en el esquema lógico, porque las entidades y las interrelaciones se transforman en relaciones<sup>1</sup>, de manera que ya no es posible distinguir unas de otras.

Para base de datos de negocio el esquema conceptual de un solo nivel de abstracción puede ser suficiente. Pero para bases de datos con información científica, puede no ser el caso. Información científica puede requerir de un esquema conceptual con diversos niveles de abstracción, generando un esquema conceptual con mayor complejidad inmersa, ver figura 5.3.

El modelo MD está orientado a esquemas conceptuales que contienen diferentes niveles de abstracción e información que contenga una amplia riqueza semántica. La implementación de nuestro modelo requiere de un análisis profundo al diseño del esquema conceptual de la base de datos. Como primer paso, es necesario obtener un esquema con entidades conceptuales de alto nivel, que a su vez pueden contener conjuntos de subentidades relacionadas entre sí; cada una a su vez, puede contener otras subentidades hasta lograr alcanzar a subentidades específicas y especializadas, desde las cuales podemos definir el esquema lógico.

Un ejemplo de lo mencionado anteriormente, se muestra en la figura 5.3, donde podemos observar en la parte de arriba dos grandes entidades conceptuales con una relación trivial, cada entidad a su vez contiene un conjunto de subentidades, las cuales pueden contener otras subentidades hasta lograr alcanzar una entidad específica. Desde esta entidad específica, se pueden construir los catálogos y tablas junto con sus respectivos atributos; en el esquema lógico estas tablas pueden contener distintos tipos de relaciones con otras entidades pertenecientes a otra entidad conceptual.

Obtener un esquema conceptual de esta manera es muy complejo, pero lo que debemos hacer es obtenerlo por niveles; ya sea de arriba hacia abajo o viceversa. Cada una de las entidades conceptuales puede actuar como una entidad incluyente que contiene otros esquemas conceptuales en su interior. De esta manera, las subentidades contienen un esquema conceptual más específico, hasta permitir obtener una pequeña entidad final. Obtener diversos esquemas conceptuales dentro de otros, puede parecer innecesario, pero no lo es; porque los esquemas sirven para crear el mapa conceptual requerido.

---

<sup>1</sup>En el modelo relacional, sólo existe la relación para representar ambos tipos de objetos.

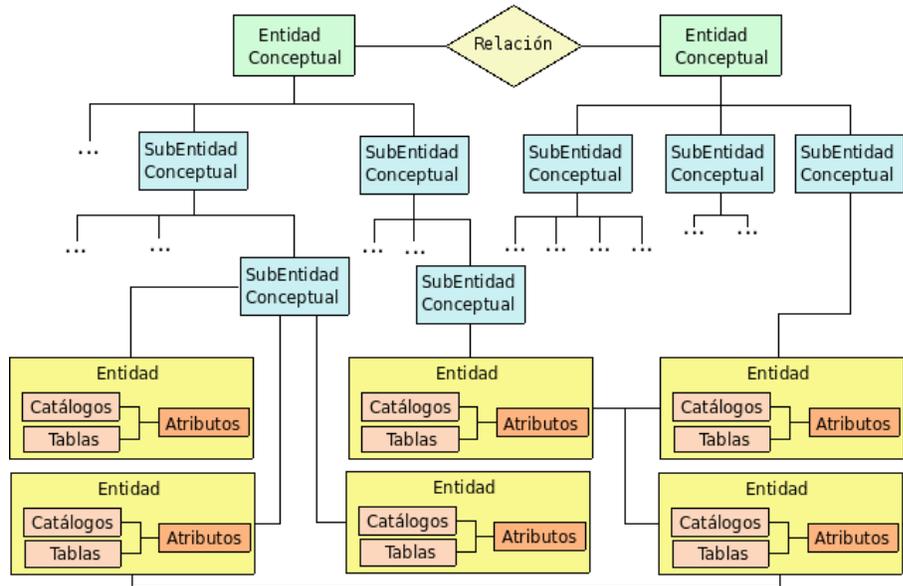


Figura 5.3: Esquema conceptual complejo, que cuenta con diversos niveles de abstracción y relaciones no triviales entre entidades.

### 5.2.2. Mapa conceptual semántico

Las pequeñas entidades obtenidas por la contención entre entidades más generales, nos permiten establecer un borrador de la estructuración de la información. Dependiendo de las relaciones encontradas en cada pequeño esquema conceptual, se pueden encontrar diversas relaciones semánticas de hiponimia, hiperonimia, holonomía y meronimia. Cada una de estas relaciones puede ser representada por conceptos que signifiquen esta relación. Los conceptos designados varían según el área de estudio a la que pertenecen los datos; de esta manera, podemos encontrar varios de ellos con alto valor de referencia para los usuarios. Todos los conceptos encontrados para especificar estas relaciones necesitan ser clasificados en niveles de inclusión; en esta clasificación pueden surgir distintos conceptos que sirven como referencia y contención de otros.

La figura 5.4, muestra como los conceptos semánticos son clasificados en el mapa conceptual partiendo desde uno muy general, que contiene otros

subconceptos generales, los cuales pueden contener más subconceptos hasta llegar a los subconceptos específicos. Los nodos finales que representan datos específicos se unen por medio de metadatos conceptuales a la base de datos.

La figura 5.4, muestra cómo podemos obtener un mapa conceptual que puede contener diversos mapas conceptuales hijos en cada nodo.

Esta jerarquía entre conceptos permite en la implementación una amplia filtración de datos no deseados y obtener un acceso directo a los datos de interés. En otras palabras, el mapa conceptual elimina la complejidad del esquema lógico de la base de datos, al dirigir al usuario a datos muy puntuales sin requerir conocer nada de la estructura interna.

Es fácil ver que la administración de conceptos pertenecientes al mapa conceptual es sencilla, porque sólo se obtienen muy pocos conceptos de esta estructura semántica, por lo que se puede utilizar una base de datos o crear un sistema administrativo propio.

De cualquier forma debemos habilitar una manera dinámica para adicionar, eliminar, actualizar, modificar y reestructurar los conceptos. También en cualquiera de los dos casos, es necesario anexar descripciones de su significado a cada concepto; las cuales serán utilizadas por la capa visual para evitar la diferencia semántica.

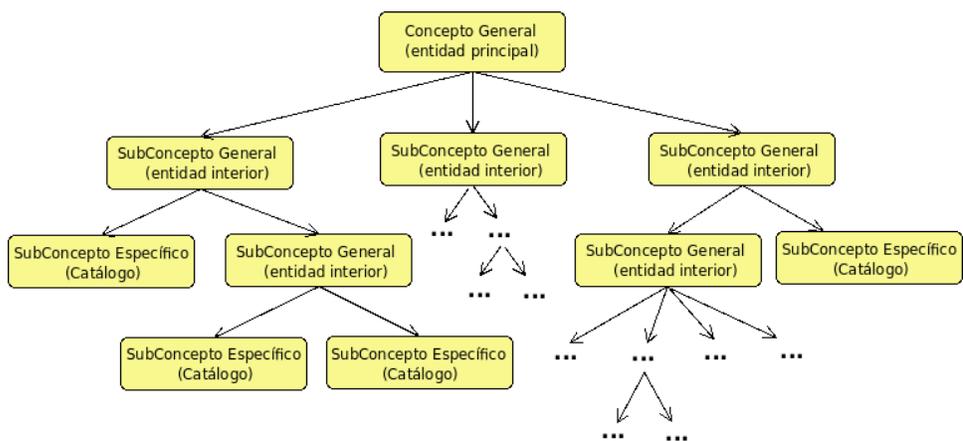


Figura 5.4: Representación de un mapa conceptual semántico, que contiene diversos conceptos incluyentes. Este mapa también puede incluir a otros submapas conceptuales en algunos nodos.

### 5.2.3. Integración meta-conceptual

Una vez que tenemos un mapa conceptual semántico de exploración, es necesario conectarlo con la información contenida en la base de datos. Para alcanzar este objetivo es necesario contar con un diccionario de datos (metadatos de la base de datos) y referenciarlo con metadatos conceptuales del mapa semántico de exploración.

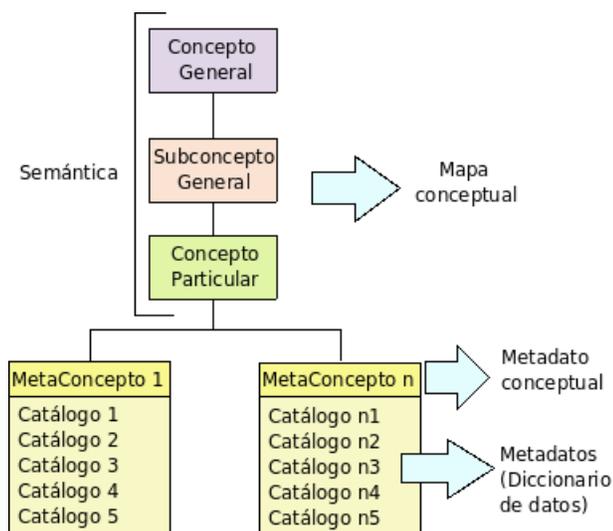


Figura 5.5: Mapa meta-conceptual que permite conectar los conceptos semánticos, con catálogos de datos de una base de datos real.

La figura 5.5, muestra cómo el mapa meta-conceptual se conecta con el diccionario de datos por medio de los metadatos conceptuales, los cuales contienen un conjunto de catálogos de la base de datos a explorar. Los catálogos de acceso contienen tablas llave; estas tablas nos permiten acceder directamente a otras tablas que tienen valores relacionados a ellas.

Cuando se obtienen los valores de los catálogos, se pueden construir las consultas de manera automática desde la capa de consultas. Las consultas se construyen de manera automática, por medio del diccionario de datos y los metadatos conceptuales. De esta forma, podemos decir que alcanzamos el acceso transparente de datos por parte de los usuarios.

En la figura 5.5, observamos cómo el usuario navega a través de los conceptos y percibe un acceso guiado y transparente a la información de su interés, cuando se llega a nodos finales se muestran los catálogos pertenecientes a un concepto muy específico. Al igual que en el mapa conceptual semántico, la estructura de metadatos conceptuales obtenida necesita de un administrador que permita controlar todas estas referencias, podemos hacerlo a través de una base de datos o utilizar un sistema administrativo propio, ya que lo importante es garantizar de cierta forma que se pueda tener un rasgo dinámico, para actualizar o modificar este tipo de metadato.

### 5.3. Comentarios Finales

En este capítulo se ha presentado un nuevo modelo de interacción orientado a la exploración de datos en sistemas Web, al cual denominamos como Modelo MD (Meta-conceptual dinámico). Este se encarga de conjuntar la rica interacción visual-dinámica y la estructuración conceptual semántica, por lo que contiene diferentes capas que mantiene una amplia comunicación y diferentes niveles de acceso a la información. También nos enfocamos en describir la esencia del modelo desde un punto de vista conceptual y aunque en este momento tenemos una idea de como funciona en su interior, es necesario diseñar una arquitectura cliente-servidor que permita llevar el modelo MD a una implementación real. En el siguiente capítulo nos enfocamos en detallar una arquitectura específica para este nuevo modelo de interacción.

## Capítulo 6

# Arquitectura Meta-conceptual Dinámica

Una vez presentado nuestro modelo MD y la conceptualización de sus datos, es necesario describir la manera en que el modelo incluye la capacidad de la interacción visual-dinámica. La manera en que lo describimos es a través de una extensión de la arquitectura tradicional cliente-servidor, donde incorporamos las capas del modelo MD y las partes internas que las componen. La arquitectura resultante es lo suficientemente genérica y abierta, para ser usada en diferentes problemas orientados a la exploración de datos. En la primera parte del capítulo se ofrece una descripción a un alto nivel, para después mostrar como puede implementarse cada una de las capas que permitan facilitar la interacción con el usuario.

### 6.1. Descripción de la Arquitectura

La implementación del modelo MD en un sistema real, necesita de una sólida arquitectura que le permita obtener toda la funcionalidad contenida en este modelo. Algunas de las características deseadas en una arquitectura orientada a sistemas Web son: flexibilidad, independencia de plataforma, independencia de software, independencia de navegador, eficiencia, viabilidad y facilidad en su implementación. Para alcanzar este objetivo, extendemos la conocida arquitectura tradicional cliente-servidor, a la cual integramos las seis capas de nuestro modelo MD, con esto obtenemos nuestra nueva *Arquitectura Meta-conceptual Dinámica (Arquitectura MD)*, principalmente orientada a la implementación de nuestro modelo en sistemas reales.

Sobre nuestra nueva arquitectura podemos obtener la siguiente definición:

**Definición 17.** Arquitectura MD.- *Esta arquitectura está basada en extender la arquitectura cliente-servidor para incorporar las capas del modelo MD, además de especificar la manera de implementar la parte visual-dinámica para su uso en sistemas reales.*

Para representar nuestra nueva arquitectura hacemos uso de UML (Unified Modeling Language) para obtener una vista de alto nivel, donde las capas y componentes son representados a través de paquetes, así como Conallen lo propone en [115]. La representación resultante se muestra en la figura 6.1, donde podemos observar las capas representadas por paquetes en blanco para sistemas Web avanzados, ya descritos en la sección 3.2.2, dedicada al Web 1.0. Por otro parte, se muestra la parte cliente y la parte servidor, donde los paquetes en color representan cada una de las capas del modelo MD y las diferentes interacciones que tienen entre ellas.

Las capas que se encuentran en la parte servidor, han sido descritas en el capítulo anterior a excepción de la capa de consultas automáticas. La capa visual y la capa dinámica se encuentran en la parte cliente. Aunque estas dos capas se muestran de manera separada, éstas interactúan y están integradas a través de la interfaz de una página Web. También debemos observar que las capas encontradas en la parte servidor, siempre tienen un reflejo en la interfaz visual de la página.

La arquitectura en capas mostrada describe cómo un usuario desde una página XHTML, obtiene una representación visual que está ligada a una capa dinámica que puede retornar una respuesta o enviar una solicitud a la capa de consultas en el lado servidor. La capa de consulta verifica qué tipo de solicitud es requerida y la capa correspondiente envía sus datos para ser mostrados por la capa visual. Todas las capas deben reflejar cambios en la capa visual. La capa meta-conceptual mantiene un enlace entre los conceptos y la base de datos, administrando la relación existente entre la capa concepto y base de datos, cuando estas obtienen actualizaciones o cambios.

La arquitectura MD es genérica, por lo cual puede ser utilizada en diversos tipos de aplicaciones orientadas a la exploración de datos en la Web. También la arquitectura es abierta, esto posibilita su implementación con cualquier tecnología que mantiene el estándar de la W3C y puede ser libre o propietaria.

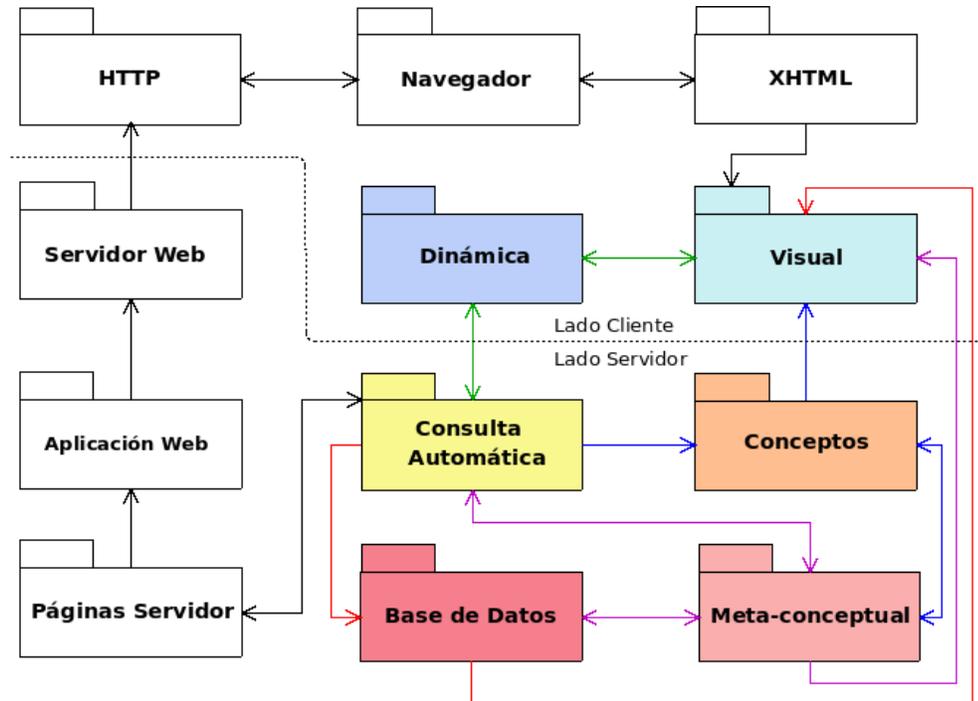


Figura 6.1: La arquitectura MD permite habilitar la rica interacción en los sistemas Web orientados a la exploración de datos.

## 6.2. Técnica de Código No Invasivo

La inmensa mayoría de páginas Web contienen una mezcla de lenguajes de programación Web, entre ellos podemos encontrar HTML, JavaScript, CSS, Flash, entre otros. Cuando se observa el código de las páginas Web, es común encontrar todos ellos en una misma página. Además, cuando tenemos páginas dinámicas como las descritas en la sección 3.2.2, entonces las páginas deben de incluir un lenguaje script servidor.

De esta manera, en la programación original de desarrollo este lenguaje también es incluido en la página y puede verse en la parte servidor, más no puede verse en la parte cliente. Existen técnicas como el uso de Servlets de Java donde el código servidor es separado, pero la parte cliente sigue presentando estos problemas de mezcla de código.

Este tipo de programación ampliamente usada, requiere de una técnica que permita la separación e independencia de código en una página Web.

La técnica de separación de código recomendada para la arquitectura MD, es la técnica conocida como *Unobtrusive Scripting*, de la cual existen diferentes traducciones al español, pero la traducción que mejor se acerca a la esencia de esta técnica es la de *Técnica de código no invasivo*, descrita como:

**Definición 18.** Técnica de código no invasivo.- *Esta técnica consiste en obtener los diferentes aspectos de una página Web de una manera independiente y separada en tres partes esenciales que son: contenido, presentación y comportamiento (Zeldman en [116]).*

Esta técnica de código no invasivo permite obtener código puro<sup>1</sup> en archivos independientes anexos a una página Web, y aunque está separado debe poder integrar su comportamiento correspondiente a la página.

Esta separación permite restar complejidad a la programación y permite una rápida localización de errores, también ofrece un mantenimiento sencillo de cada una de las partes al ser independientes unas de otras.

Cuando la técnica de código no invasivo es bien aplicada, permite una amplia disminución de código eliminando mucha de su complejidad, porque sólo se programa código con propósitos bien definidos, eliminando código basura y sub-utilizado. La técnica permite la creación de páginas más ligeras y eficientes aumentando el rendimiento del sistema Web.

Esta técnica también permite el uso de estándares abiertos en programación Web, ayudando a la implementación de otro tipo de técnicas, tal como la *Técnica de Cross-Browser* que es descrita en [116] y en [117].

La técnica de *Cross-Browser*, permite que un sistema Web no pierda su diseño, accesibilidad y funcionalidad por el uso de diferentes navegadores en el acceso. El éxito de esta técnica es alcanzado mediante el uso de estándares abiertos y el uso de código especial el cual es conocido como *Hacks*.

Los *Hacks* son usados especialmente en algunos navegadores que no siguen el estándar del consorcio W3C, por ejemplo: *Internet Explorer 5*, *Internet Explorer 6* y *Safari*. Los *Hacks* regularmente son usados para la presentación y el acceso a elementos de una página Web.

---

<sup>1</sup>Se refiere a la capacidad de obtener código en un solo lenguaje, sin mezclar ningún otro lenguaje como sucede en la programación común de páginas Web.



Los elementos contienen texto e imágenes, que sirven como vista principal a los usuarios y permiten el diseño de nuevas interfaces que brindan una mayor usabilidad.

Los elementos en la interfaz visual de la arquitectura MD, no son ligas a otros documentos de hipertexto, son elementos que capturan eventos de usuarios en la interfaz; los elementos activan funciones de actualización que requieren datos internos o externos y utilizan una sola interfaz para muchas funcionalidades comunes. Las funcionalidades varían conforme los niveles de interacción de usuario las vayan requiriendo.

Los elementos requieren de una presentación, la cual es dada a través de las *Hojas de Estilo (CSS Cascading Style Sheets)*. Estas últimas son una técnica que permite asignar la presentación de los elementos en la interfaz como puede verse en [117]. El estilo dado a los elementos permite especificar el color, tamaño, ancho y ubicación de los elementos en una página Web y pueden usarse diversas técnicas, como se muestra en [118], [119], [120]. Las hojas de estilo deben ser anexadas en archivos separados con codificación estándar CSS, para definir el estilo a través de clases o identificadores de los elementos en la página.

Para identificar y obtener clases de elementos en la página, es necesario describirlos como un conjunto de objetos; ello se logra por medio del uso del *Modelo de Objetos del Documento (DOM Document Object Model)*, el cual permite ver a los elementos de un documento Web de una manera diferente.

El consorcio W3C define a DOM como:

**Definición 19.** Modelo de Objetos del Documento.- *Es una plataforma y una interfaz en lenguaje neutral, que permite a los programas y scripts, acceder y actualizar dinámicamente el contenido, la estructura y el estilo de los elementos del documento (Edwards et al. en [116] y Langridge en [117]).*

La arquitectura MD utiliza DOM para obtener toda una serie de diversas propiedades como: estilo, acceso, modificación, creación, eliminación y escucha de eventos, en todos los elementos que conforman la estructura interna del documento Web, como puede verse en la figura 6.2.

DOM es utilizado por la capa de interacción dinámica, para producir todos estos cambios en la interfaz visual, como se describe en la siguiente sección.

## 6.4. Interacción Dinámica

La capa de interacción dinámica define el *Comportamiento* del sistema el cual es reflejado en la interfaz visual y la exploración. La capa está conformada por cuatro componentes internos que son: *Escucha*, *Actualización*, *Formateo* y *Motor asíncrono*, como lo muestra la figura 6.3.

Todos los componentes son codificados con un solo lenguaje Script Cliente como: JavaScript, Visual Basic Script (VBScript), JScript, ActionScript o Flex. En este proyecto hacemos uso del lenguaje *JavaScript* que contiene diversas características propias, que permiten implementar de manera abierta diversos algoritmos para la manipulación de la interfaz y usar diversas técnicas, entre ellas la técnica de *Cross-Browser*.

JavaScript hace un amplio uso de DOM para manipular el comportamiento del sistema y su interfaz. Para manipular este comportamiento es necesario obtener una comunicación cercana para recibir parámetros o datos entre los diversos componentes, que forman la capa dinámica.

La implementación del componente *Escucha* puede ser realizada de dos diferentes maneras, a través de manejadores DOM 0 que permiten una implementación rápida pero con algunas desventajas o con el estándar W3C de *Escucha de Eventos*, que contiene diversas ventajas con respecto a DOM 0. El uso de escuchas de eventos de la W3C, debe controlar una serie de etapas en los eventos como: *Captura*, *Objetivo*, *Burbujeo* y *Propagación*.

Los escuchas de eventos W3C pueden ser implementados con soporte para Cross-Browser y obtener igual comportamiento en diferentes navegadores, mucha información amplia sobre la implementación de escuchas de eventos, puede verse en [116], [117].

Cuando los eventos son capturados, se pueden activar ciertas funciones de actualización con datos internos o externos; si la actualización es con datos internos, los datos están contenidos en el documento cliente y son modificados y manipulados de manera inmediata. Cuando la actualización necesita datos externos al documento, es necesario obtener datos del servidor de manera asíncrona. Una vez que los datos son obtenidos se recurre a funciones de actualización de interfaz para incorporar los nuevos datos.

El modelo clásico Cliente-Servidor utiliza la comunicación síncrona, esto quiere decir que el cliente no puede continuar su trabajo hasta recibir una respuesta del servidor, generando muchos tiempos de espera innecesarios.

La comunicación asíncrona permite lanzar peticiones y obtener datos, sin necesitar bloquear la interacción del usuario; una amplia explicación de este proceso podemos encontrarlo en [81]. Para implementar el motor asíncrono es necesario hacer uso del objeto *XMLHttpRequest* el cual fue implementado alrededor de 1999, pero fue ampliamente conocido hasta que el término AJAX (Asynchronous JavaScript and XML) fue acuñado por Jesse James Garret en 2005 (ver [121], [122], [123], [124], [125]).

El objeto *XMLHttpRequest* es parte fundamental del motor asíncrono y contiene cuatro estados de transición que interactúan con diferentes métodos y propiedades del objeto, las transiciones son:

- |   |                 |   |
|---|-----------------|---|
| 0 | No Inicializado | El método <code>Open</code> no ha sido llamado.               |
| 1 | Cargando        | Se llama al método <code>Open</code> .                        |
| 2 | Cargado         | Se llama al método <code>Send</code> .                        |
| 3 | Interactivo     | La propiedad <code>responseText</code> tiene datos parciales. |
| 4 | Completo        | La propiedad <code>responseText</code> tiene todos los datos. |

Estos métodos y propiedades son los que permiten la comunicación asíncrona con el servidor. Las peticiones utilizan el *Protocolo de Transferencia de Hipertexto (HTTP, HyperText Transfer Protocol)* y se espera el código de respuesta 200. Junto con este código de respuesta, debemos verificar que la transición alcanzó el estado 4 y después activar una función de filtrado y formateo de los datos. Los datos obtenidos pueden tener diferentes formatos como: Texto Plano, XML, JSON, imágenes, etc.

El algoritmo que provee la comunicación asíncrona es pequeño y potente, una muestra de su implementación puede verse en [126], [125].

Aunque existen bibliotecas externas que proveen este algoritmo, en esta investigación optamos por una implementación propia y orientada al trabajo de exploración. La implementación propia permite eliminar una importante cantidad de código innecesario que contiene este software externo. También podemos modificar el código según la funcionalidad requerida y generar una activación en base a parámetros de paso y obtener algún formato de dato que facilite el procesamiento de la información.

Por otra parte, el formato de datos que usamos en este proyecto son texto plano y estructuras XML, según el nivel de interacción en que se encuentre el usuario. Una vez que los datos son obtenidos del servidor, es necesario analizar la estructura y los formatos de los datos y procesarlos para que puedan ser integrados a la interfaz del documento.

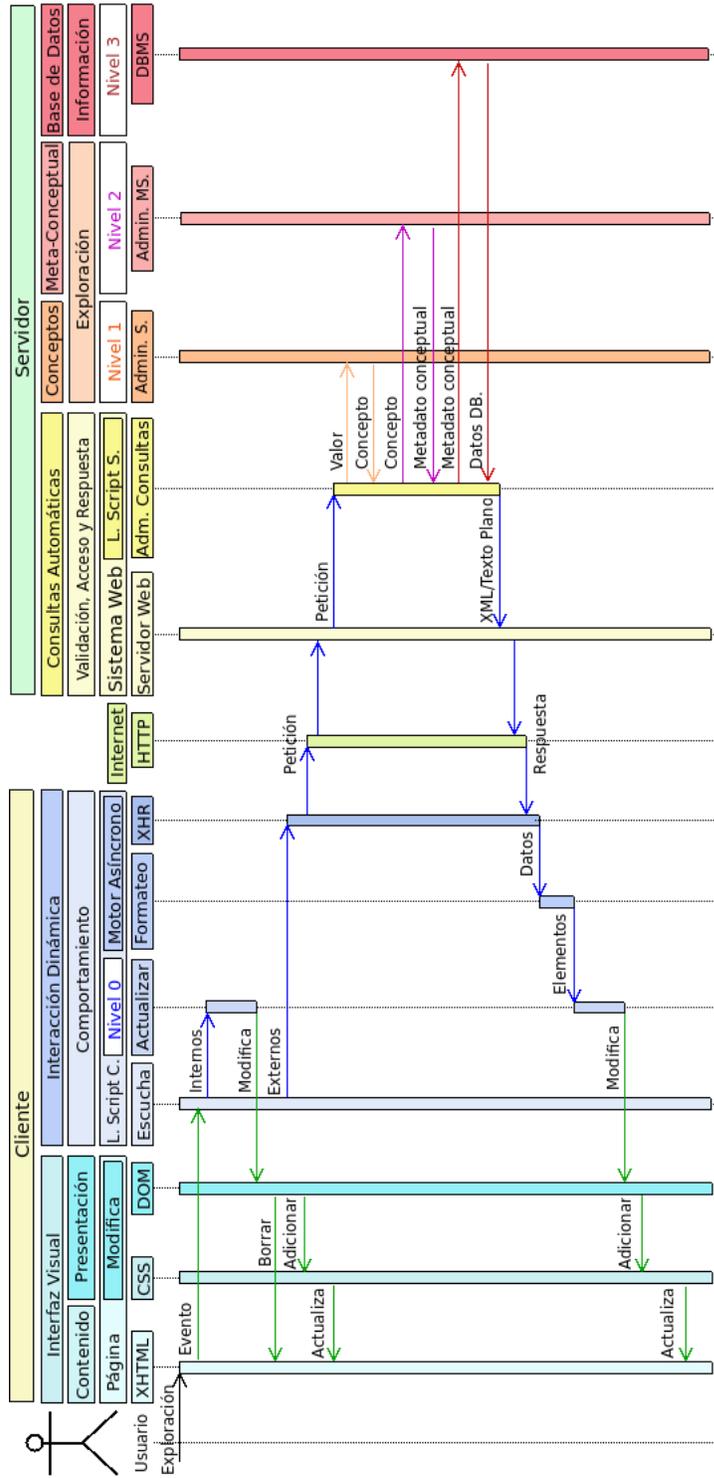


Figura 6.3: Diagrama de secuencia que muestra la interacción entre todos los componentes y capas en un sistema que utiliza una arquitectura MD. Los eventos lanzados por el usuario disparan un comportamiento de actualización de la interfaz visual. La actualización depende del nivel de interacción en que se encuentre el usuario.

## CAPÍTULO 6. ARQUITECTURA META-CONCEPTUAL DINÁMICA

Esto implica modificar y actualizar la estructura interna del documento y puede realizarse a través del uso de DOM. Las funciones que permiten la actualización, dependen en gran medida del estado en que se encuentra la interfaz para incorporar los nuevos datos, ver figura 6.3.

Todo el proceso anteriormente descrito, también puede verse como una secuencia de instrucciones a seguir cuando se produce un evento en la interfaz gráfica, el cual es descrito como:

```
Capturar Evento en la Interfaz
  Identificar evento, elemento, nivel de interacción
    Modificar parámetros internos
      Activar funciones de actualización
        Activar Petición Asíncrona
          Solicitar Datos al Servidor
            Obtener Datos del Servidor
              Analizar y Formatear Datos
                Crear elementos acordes a la estructura
                  Asignar identificadores y escucha de eventos
                    Asignar estilo a los elementos
Incorporar elementos a la interfaz
```

Cada una de estas instrucciones generales involucra la implementación de una serie de diferentes algoritmos para llevar a cabo cada instrucción. Aunque la serie de instrucciones puedan parecer demasiadas, en realidad el tiempo de ejecución de cada una de ellas es muy corto; pensar en la implementación de cada una de ellas puede parecer una enorme tarea, pero la implementación es facilitada en gran medida por la técnica de código no invasivo. Por lo que se obtiene en realidad muy poco código y este es realmente utilizado y explotado por la aplicación.

Existen diversas ventajas en esta implementación, algunas de las más importantes son el tiempo de respuesta y la actualización casi instantánea de la interfaz, si la técnica de programación usada es correcta y eficiente. En la parte cliente, esta capa es la más importante ya que define todo el comportamiento del sistema. La capa se encarga de escuchar eventos, obtener los datos, modificar los elementos e incorporar a los nuevos datos en los elementos, como lo muestra la figura 6.3.

## 6.5. Consultas Automáticas

La capa de consultas automáticas se encuentra en la parte servidor por consecuencia, debemos de hacer uso de un lenguaje de programación script servidor como: *JSP (Java Server Pages)*, *PHP (Hypertext Pre-Processor)*, *ASP (Active Server Pages)*, *ColdFusion*, *Perl*, *Ruby*, etc.

El lenguaje a utilizar es indiferente, pero es imprescindible verificar que el lenguaje seleccionado pueda enviar datos al cliente en los formatos necesitados por la capa de interacción dinámica. Algunos lenguajes se ejecutan directamente en el servidor Web Apache y otros necesitan un software servidor independiente que interactúa con el servidor, por ejemplo Tomcat. Otros en cambio son propietarios y tienen un servidor Web propio de ejecución.

La más importante característica en esta capa, es la validación de datos y el acceso a los datos. Es importante tener un algoritmo riguroso, que pueda analizar los datos de entrada y pueda deducir en base a parámetros correctos y muy específicos las consultas a realizar. Esto garantiza que el uso de técnicas de SQL injection, no afecten de ninguna forma el sistema, para de esta manera evitar el uso de caracteres diferentes a los necesitados y validar cada tipo de dato de entrada. Es importante este punto porque desde esta capa, se construyen las sentencias de consulta a las bases de datos. Es importante usar la *Técnica de Parametrizar Consultas* y hacer un encapsulamiento de las referencias reales. Es decir, no tener nombres reales de acceso a datos en la parte cliente.

Cuando los datos son recibidos, deben ser validados y modificados para construir una consulta real, todo hecho en base a un modelo bien definido de validación y acceso a datos. A su vez, las bases de datos deben tener usuarios con permisos restringidos de acceso a datos reales; si las modificaciones son parte de la necesidades del sistema, es necesario construir reinos de acceso en el sistema mismo a nivel de aplicación, una explicación detallada del concepto de reinos de acceso, la podemos ver en [127].

Los lenguajes orientados a objetos, permiten la reutilización de código con la construcción de clases con propósitos específicos. Esto nos permite crear clases para validación, construcción de sentencias, acceso a datos y respuesta al cliente.

El uso de objetos nos puede permitir implementar de manera general las cuatro funcionalidades y ocupar los objetos en diversos niveles de interacción

## CAPÍTULO 6. ARQUITECTURA META-CONCEPTUAL DINÁMICA

con un cambio de parámetros en los objetos (instancias de clases) creados en la ejecución. Las instrucciones generales para esta implementación son:

Recibir petición

    Ejecutar Aplicación Servidor

        Validar petición

            Identificar nivel de acceso

                Crear nueva petición especial según nivel

                    Enviar Petición a una BD. según nivel

                    Obtener Datos

                Formatear respuesta

            Enviar Respuesta a Cliente

Esperar nueva petición

Debemos de recordar que la parte servidor actúa como servicio Web y no necesita construir una página Web, sólo debe enviar los datos requeridos al cliente en el formato especificado en la capa dinámica; esto permite incrementar el tiempo de respuesta en la interfaz.

### **6.6. Comentarios Finales**

Una vez que hemos descrito nuestra nueva arquitectura MD, así como las capas que le brindan la funcionalidad deseada, es necesario observar que las capas contienen distintos lenguajes de programación estándar; si los desarrolladores desean usar software propietario pueden hacerlo, sólo es necesario mantener la interacción descrita entre capas. Este capítulo ya nos permite desarrollar el modelo MD en un sistema real, la pregunta que nos hacemos en este momento es si realmente funcionará o solo sirve como caso teórico. En nuestro siguiente capítulo nos concentramos en un caso de estudio, que contiene datos biológicos y una amplia semántica en los datos. Este caso de estudio nos permite obtener un prototipo con eficaz exploración de datos, sin requerir conocer nada de la compleja estructura interna de la base de datos, ni del sistema. El prototipo nos muestra el amplio potencial de este modelo y de esta arquitectura con fines prácticos.

## Capítulo 7

# Prototipo

Una vez descrito el modelo y la arquitectura MD, es necesario dar a conocer las ventajas que ofrece su implementación en un sistema real. Como se ha mencionado el modelo MD está orientado a la exploración en bases de datos científicas. De esta manera el caso de estudio que se presenta en este capítulo, es una base de datos científica que contiene datos biológicos sobre microorganismos; la cual en un candidato ideal, ya que la información cuenta con las características descritas en la sección 2.2.2 y 2.3.3, del capítulo 2. Este capítulo en un inicio describe la esencia de la base de datos de microorganismos; después se presenta la manera en que se obtienen los mapas conceptuales semánticos de navegación, posteriormente se describe la implementación de un prototipo construido a partir del modelo y la arquitectura MD; por último se muestran resultados sobre la recuperación de datos en el lado cliente y servidor.

### 7.1. Descripción del Caso de Estudio

Como se describió en la sección 2.2.2, el desarrollo de herramientas o sistemas orientados a bases de datos científicas con información biológica no es una tarea trivial, debido a las características especiales que tienen estos datos. El caso de estudio seleccionado contiene información biológica especializada sobre microorganismos, una de sus principales necesidades es la exploración sobre una compleja estructura relacional. Esta sección se enfoca en describir la esencia de esta base de datos científica usada como caso de estudio.

### 7.1.1. Colección microbiana

México cuenta con la *Colección Nacional de Cultivos Microbianos*, la cual fue creada en el Centro de Investigación y de Estudios Avanzados del I.P.N. (Cinvestav), con el fin de obtener un acervo de cultivos puros para su aplicación en la docencia y la investigación.

En 1977 la colección fue reconocida y aceptada por el WDCM (World Data Centre for Microorganisms) con el acrónimo *CDBB* y el número de registro *500* para su identificación internacional. En 1994 surge el proyecto interdisciplinario *P-153 "Colección de Cultivos Microbianos del CINVESTAV fase 1, Base de Datos"*, entre la Sección de Computación del Departamento de Ingeniería Eléctrica, y la Colección Nacional de Cultivos Microbianos, el cual es apoyado por CONABIO.

Los principales objetivos del proyecto fueron:

- Administrar y poner a disposición de la comunidad la información de la base de datos CDBB500.
- Desarrollar un sistema que facilite el intercambio de información microbiana, por medio de una red nacional entre diversas instituciones.
- Implementar proyectos asociados que permitan la investigación y el desarrollo tecnológico en computación y microbiología.
- Animar la colaboración entre diversos institutos y laboratorios especializados en cepas microbianas.

El proyecto obtuvo como resultado la primera versión del *Sistema de Información CDBB-500 (Versión 1.1)* desarrollado en Access, después diferentes versiones fueron liberándose hasta obtener la versión que actualmente es denominada como *Sistema Micro500* y que permite consultas públicas de información de cepas microbianas en la Web, ver figura 7.1.

Este sistema tiene una gran importancia nacional y mundial, por lo que existen ligas al sistema desde instituciones nacionales como CONABIO e internacionales como REMIB (*Red Mundial de Información sobre Biodiversidad*), WDCM (*World Data Center for Microorganisms*), ARS (*Agricultural Research Service, USA*) y en 2008 fue incorporada a una federación mundial de bases de datos microbianas, sobre el portal *StrainInfo.net*, perteneciente a la Universidad de Gent en Bélgica.

La colección de microorganismos es reconocida como una autoridad internacional en el mantenimiento, depósito y conservación bajo resguardo de cepas

microbianas puras; esto la hace ser valiosa en el desarrollo de la biotecnología en México y en el mundo. La base de datos de la colección microbiana contiene información científica y biológica con una alta riqueza semántica, esto la convierte en un buen candidato para nuestro caso de estudio; información más amplia sobre el proyecto Micro500 puede verse en [128] y [129]. A continuación se describe el tipo de información contenida en esta base de datos, como primer paso de implementación de nuestro modelo.

CDBB	Nombre	Acrónimo
978	<i>Bacillus amyloliquefaciens</i>	[ NRRL B-645 ]
1105	<i>Bacillus atrophaeus</i>	[ ATCC 9372 ] [ CIP 77.18 ] [ DSM 675 ] [ IFO 13721 ] [ NCIB 8058 ]
949	<i>Bacillus cereus</i>	[ ATCC 6464 ]
1019	<i>Bacillus cereus</i>	[ ATCC 11778 ] [ ATCC 9634 ] [ NCTC 10320 ]
988	<i>Bacillus cereus</i>	[ ATCC 6464 ]
34	<i>Bacillus cereus subespecie mycoides</i>	[ ND W-27 ]
29	<i>Bacillus circulans</i>	[ ND WL-12 ]
549	<i>Bacillus circulans</i>	[ IGC 1074 ]
28	<i>Bacillus circulans</i>	[ ND WL-5 ]
31	<i>Bacillus licheniformis</i>	[ ATCC 11945 ]
30	<i>Bacillus licheniformis</i>	[ ATCC 102 ] [ ATCC 4527 ] [ ATCC 8243 ] [ ATCC 9789 ] [ NCTC 2586 ] [ NCTC 6346 ]

Figura 7.1: Vista actual del sistema Micro500, el cual contiene información pública en la Web sobre microorganismos.

### 7.1.2. Base de datos microbiana CDBB500

Actualmente la colección nacional mexicana de microorganismos contiene información diversa y relacionada de aproximadamente 3000 cultivos microbianos de diferentes áreas geográficas del territorio nacional y de diferentes colecciones microbianas de otros países. Cada cultivo genera una diversidad de datos biológicos especializados que son utilizados en distintos ámbitos como industria, medicina, agricultura, investigación y docencia. El nombre de base de datos es *CDBB500*, donde CDBB es el acrónimo otorgado por la WDCM y 500 el número de registro y cuenta con 4 tipos de cepas las cuales son: bacterias, hongos filamentosos, levaduras y microalgas.

Cada una de las cepas contiene diversas propiedades y características de su propia naturaleza. Pero todas ellas comparten conceptualmente hablando, las mismas áreas de estudio como: taxonomía, conservación, reproducción, medios de cultivo, aplicaciones, referencias científicas, etc. Aunque nuevamente expresamos, cada una de estas áreas son específicas para cada tipo de cepa, algunos de los atributos de cada cepa son muy especializados y dependientes del contexto sobre el cual son extraídos; esto incrementa de manera considerable la complejidad relacional de la información, porque presentan una alta variabilidad según el contexto. Además también es necesario sumar la alta diversidad de características muy particulares que un mismo microorganismo puede contener, con respecto a otros muy similares o parecidos.

Por otro lado, podemos decir que existen diversos atributos compartidos en la información sobre microorganismos, que corresponden con datos particulares como: género, especie, acrónimos, etc. Estos atributos ayudan a obtener ciertos esquemas conceptuales genéricos que ayudan a estructurar estos datos biológicos.

El trabajo de varios años en el proyecto Micro500, ha permitido obtener un esquema conceptual a un alto nivel de la base de datos CDBB500 y que es mostrado en la figura 7.2. En este esquema podemos observar una serie de entidades muy generales y la manera en que se relacionan unas con otras; cada una de estas entidades en su interior esta relacionada a diversas tablas, generando una relación más compleja.

El esquema conceptual de la figura 7.2, ha sido dividido visualmente a través de diferentes colores, para mostrar la separación entre las diferentes áreas de especialización de la información; también permite observar la complejidad relacional entre las entidades que muchas veces actúan como co-dependientes unas de otras.

En la sección 5.2.1, señalamos lo afirmado por Castaño et al. en [114], donde menciona que mucha de la semántica obtenida por el esquema conceptual es perdida en el esquema lógico, de esta manera necesitamos alguna forma de conservarla y proyectarla a la interfaz con el fin de acceder a la información de manera puntual. El modelo MD trata de hacer este mapeo mediante la construcción de un mapa conceptual de exploración para relacionarlo con el diccionario de la base de datos a través de los nodos terminales.

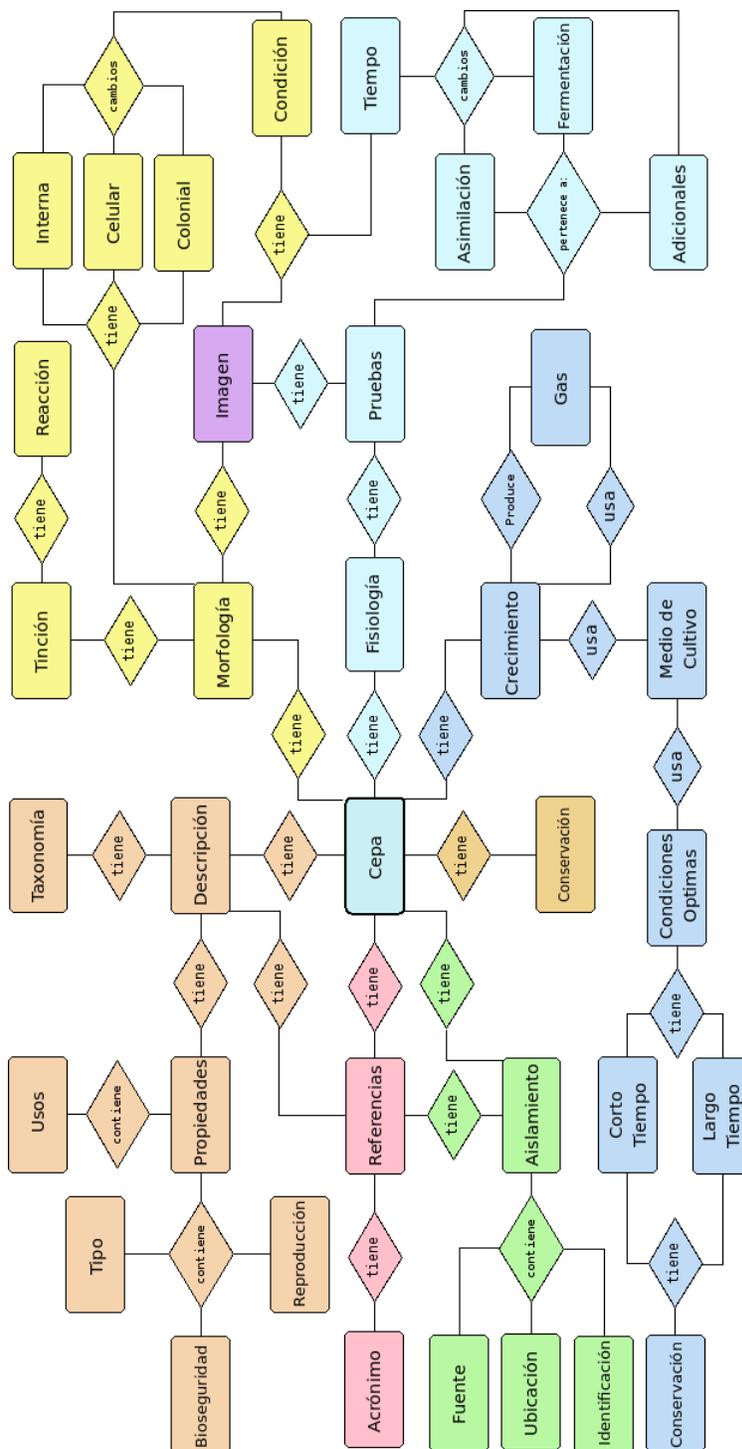


Figura 7.2: Esquema conceptual de alto nivel, que presenta parte de la complejidad inmersa en la base de datos CDBB500. El modelo MD rescata parte de esta semántica, para ser utilizada en la interfaz de un sistema Web.

## 7.2. Mapa de Exploración

La construcción de un mapa conceptual para la exploración de los datos, es la parte vital del modelo MD. La construcción del mapa puede ser llevada a cabo de manera sencilla y mediante un lenguaje de modelado, en este caso de estudio en específico utilizamos UML para obtener su representación. Cada concepto general es representado con un paquete y las relaciones son modeladas a través de asociaciones (homonimia-hiperonimia) y composiciones (meronimia-holonomia). De esta manera, la sección presenta la construcción de estos mapas de navegación para nuestro caso de estudio.

### 7.2.1. Mapa semántico de exploración

Las relaciones entre entidades del esquema conceptual de la figura 7.2, tienen una alta complejidad inmersa que se incrementa de manera sustancial en el esquema lógico. Implementar un sistema de exploración que pueda manejar de manera eficiente todas estas relaciones y ofrezca un acceso interactivo a los datos, es una tarea no trivial. El modelo MD está orientado a este tipo de problemas y permite la creación de sistemas de información Web, con amplia interacción orientada a la exploración en bases de datos.

El primer paso para crear la capa conceptual de navegación, es obtener una estructuración jerárquica partiendo de conceptos generales a subconceptos específicos, ver figura 5.4. Para realizar este mapeo es necesario tener una buena comprensión de los datos y la información de la base de datos.

También necesitamos seguir los siete pasos descritos por Quesada en [130], que nos ayudan bastante en este proceso:

1. Identificar y seleccionar los conceptos o ideas principales.
2. Escoger el concepto más importante.
3. Ordenar a partir del concepto seleccionado los demás, según su grado de generalidad.
4. Estructurar entre sí los conceptos y elegir palabras que represente de mejor manera el tipo de relación.
5. Buscar las relaciones posibles aún entre conceptos lejanos.
6. Elaborar el mapa conceptual jerárquico.

7. Elaborar mapas conceptuales diferentes para los distintos temas o subtemas.

El paso número uno nos indica que se deben identificar los conceptos o ideas principales, en nuestro caso de estudio estamos hablando de diferentes tipos de cepas con características como: fisiología, morfología, crecimiento, conservación, aislamiento, descripción, imágenes, referencias, etc. El principal elemento en esta información es el tipo de cepa ya sea bacteria, hongo, alga, levadura o protozooario. De esta manera seleccionamos al concepto *cepa* que es definido como un conjunto de microorganismos de un mismo género y especie con iguales características intrínsecas que crecen en una colonia o cultivo.

Siguiendo estos pasos podemos crear nuestro primer mapa conceptual<sup>1</sup>, obteniendo como resultado el de la figura 7.3 que está en un alto nivel y es muy general, donde cada concepto es representado por un paquete que contiene internamente otros submapas conceptuales, como el punto 7 lo señala.

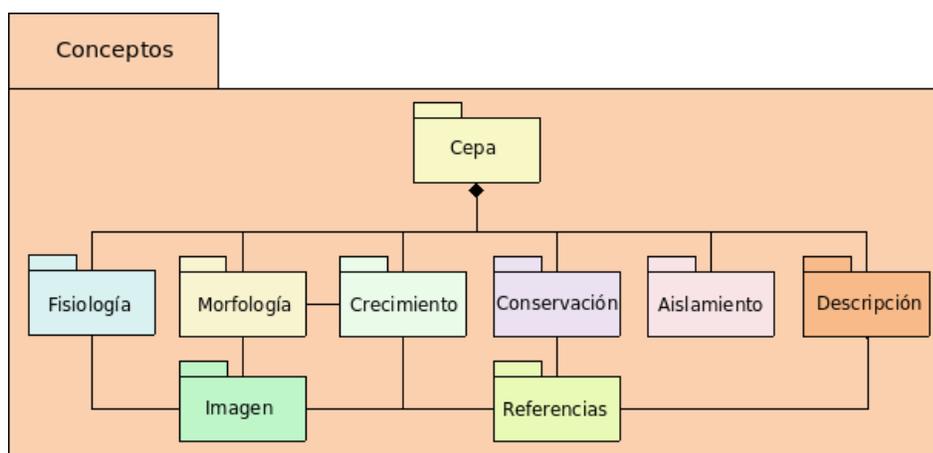


Figura 7.3: Mapa general de exploración, el cual contiene las entidades conceptuales más generales, para la exploración en la base de datos.

Las relaciones que muestra en este mapa conceptual (figura 7.3), están representadas para un primer acceso a los datos. Cada uno de estos conceptos necesita ser representado, mapeado y responder a diferentes tipos de eventos en la interfaz a través de botones, barras deslizantes, gráficos u otros elementos.

<sup>1</sup>Los mapas de exploración son resultado del trabajo interdisciplinario con *Jovita Martínez y Juan Carlos Estrada*, personal de la Colección de Cultivos Microbianos.

Por medio de estos elementos, deberemos poder navegar entre los mapas conceptuales interiores de cada concepto. Las relaciones encontradas en el mapa conceptual (figura 7.3), presentan relaciones aparentemente triviales, pero a medida que descendemos en el mapa las relaciones pueden hacerse más complejas, como se muestra en la figura 5.3.

El principal concepto encontrado en la capa conceptos es el denominado como *Cepa*. Este concepto está representado por un paquete y está asociado a cinco diferentes tipos de microorganismos. El mapa conceptual interior de este concepto es presentado en la figura 7.4.

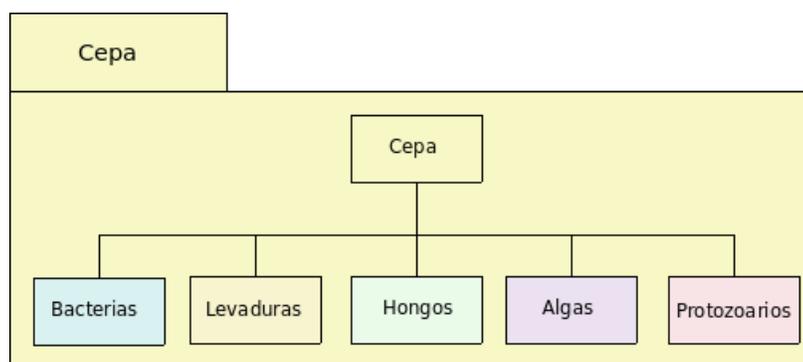


Figura 7.4: El concepto de cepa contiene una relación de hiperonimia con los diferentes tipos de microorganismos, cada tipo contiene catálogos diferentes en el esquema lógico de la base de datos.

El concepto cepa relaciona a los subconjuntos de diferentes tipos de cepa encontrados en nuestra base de datos. Dado un subconcepto seleccionado, los catálogos habilitados serán diferentes porque cada tipo mantiene una especialización de sus datos. Es decir, que aunque comparten la misma semántica conceptual del mapa de la figura 7.3, los datos a los que hacen referencia son totalmente diferentes. Por ejemplo, los datos que son referenciados a una morfología<sup>2</sup> de una bacteria, son completamente diferentes a la morfología de un hongo. La morfología aunque representa el mismo concepto de descripción física, cada tipo mantiene datos especializados según el tipo de microorganismo al que hacemos referencia.

La figura 7.5 presenta de manera gráfica cómo un mismo concepto mantiene

<sup>2</sup>La morfología hace referencia a las características micro y macroscópicas de un microorganismo, en este caso de estudio.

relación hacia los diferentes tipos de cepas; también muestra al concepto holónimo morfología y los subconceptos merónimos a excepción del subconcepto tinción<sup>3</sup>, todos los demás subconceptos mantienen una relación conceptual de la morfología de microorganismos.

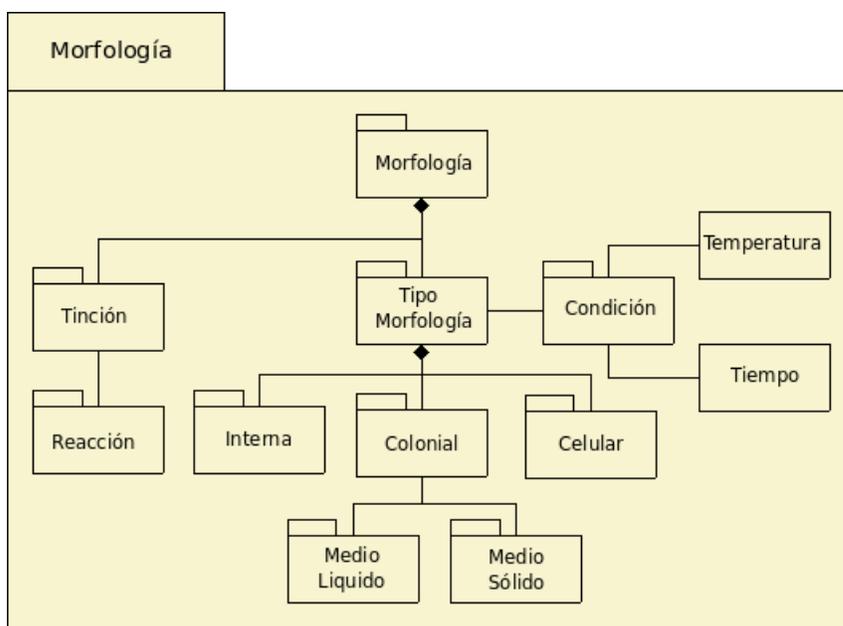


Figura 7.5: Mapa conceptual obtenido para el concepto de morfología y que es común a todos los tipos de cepa, pero según el tipo de microorganismo al que hacemos referencia, el nodo final obtiene distintos catálogos en la base de datos.

El subconcepto condición mantiene los nodos terminales temperatura y tiempo. Estos subconceptos específicos pueden mantener un rango de valores comunes, donde el nodo de temperatura mantendrá grados celsius y el nodo de tiempo es dividido en horas y días. El nodo tipo de morfología contiene los subnodos interna, celular, colonial; donde colonial es dividido en otros dos subnodos de medio líquido y sólido. Todos estos nodos son comunes, pero las referencias y los datos son diferentes según el tipo de cepa que seleccionemos. El enlace de los nodos terminales con la base de datos es nuestro siguiente punto a describir.

<sup>3</sup>La tinción es un proceso regularmente orientado a la identificación de bacterias y puede no ser muy útil para otro tipo de microorganismo.

### 7.2.2. Mapa meta-conceptual

Una vez que llegamos a nodos terminales a través del mapa conceptual de exploración, es necesario el enlace entre los conceptos y los datos en la base de datos. En este enlace necesitamos identificar los catálogos específicos a los cuales se tiene la referencia. Suponiendo que en el mapa conceptual de morfología, navegamos hasta el nodo terminal denominado como *Celular*<sup>4</sup>, es necesario conocer cuales son los respectivos datos a los que se debe tener acceso.

La capa Meta-conceptual actúa como núcleo referencial entre semántica conceptual y los datos. De esta manera, a través del concepto *Celular* y el tipo de cepa seleccionado, obtenemos los metadatos de referencia a la base de datos.

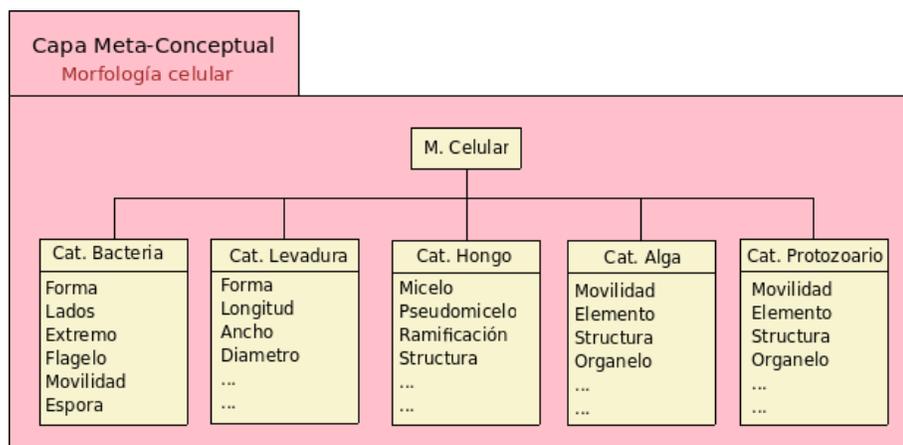


Figura 7.6: Mapa Meta-conceptual de exploración, para el subconcepto terminal de morfología celular de cepas.

La figura 7.6 muestra las referencias de los distintos catálogos a los que se puede tener acceso. La entidad conceptual morfología celular hace referencia a cinco diferentes tipos de catálogos. Cada uno de ellos está compuesto de otros catálogos especializados para los diferentes tipos de microorganismos. Para una navegación desde bacteria hacia su morfología celular, deberemos tener referencia puntual al catálogo de bacteria que contiene las propiedades de forma, lados, extremo, flagelo, etc. Cada uno de estos contiene datos únicos y especializados para bacterias.

<sup>4</sup>La morfología celular se refiere a la morfología que tiene individualmente un microorganismo en particular.

Es necesario señalar que aunque existan nombres iguales en las propiedades de otros catálogos, estos son totalmente diferentes porque la referencia es hacia catálogos distintos en la base de datos.

La figura 7.7, muestra la funcionalidad dual de un metadato conceptual, porque trabaja como metadato del concepto semántico y como metadato de referencia hacia el esquema lógico; esta funcionalidad dual, nos permite enlazar un concepto a datos en la base de datos. En la figura 7.7 podemos observar cómo cada propiedad está enlazada a un catálogo diferente, con sus respectivos datos reales.

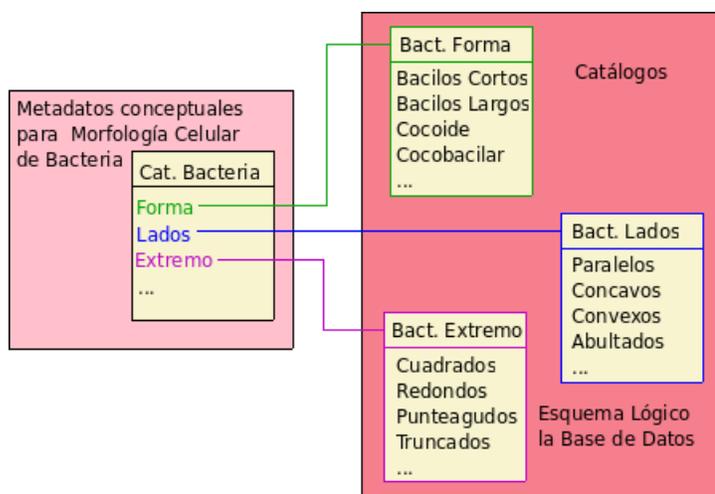


Figura 7.7: Cada propiedad del catálogo bacteria es un metadato que actúa como referencia del diccionario de datos al esquema lógico de la base de datos.

Una vez obtenido el dato y el catálogo correspondiente en la navegación, la capa de consultas obtiene los datos correspondientes y después es habilitada a realizar la consulta de manera automática y dinámica.

Todo este proceso es oculto a la vista del usuario y es activado por medio de los mapas conceptuales de navegación; los cuales deben ser representados en la interfaz del sistema, permitiendo el acceso a los datos de manera transparente, filtrando toda la información que no es de interés para el usuario. Una vez obtenida esta estructuración de la información, es necesario mostrarla en una interfaz que permita su acceso en la Web, como a continuación se presenta.

## 7.3. Prototipo

Una vez obtenida la parte estructural por medio de mapas conceptuales, es necesaria la implementación de un prototipo que permita verificar la funcionalidad del modelo y arquitectura MD. El prototipo debe facilitar la exploración de esta base de datos científica e integrar todos los puntos descritos en los capítulos anteriores. La primera parte de la sección presenta el diseño de la interfaz; en la segunda parte se presenta el prototipo y cómo ocurre la navegación de los usuarios en el sistema.

### 7.3.1. Diseño de interfaz

El diseño de interfaz para un sistema de exploración de datos no es una tarea sencilla. Cuando la interfaz es desarrollada para un sistema de escritorio, podemos hacer un uso más extenso de los recursos que ofrece una computadora para proporcionar una amplia interacción. Cuando desarrollamos la interfaz de un sistema Web, el desafío es mayor; porque la interacción es más restringida, necesita datos externos, además de requerir un navegador. Debemos recordar que la interfaz puede definir si un sistema es útil y exitoso para el propósito que fue creado.

Anteriormente señalamos que muchas de las experiencias frustrantes son debido a complejos, malos y deficientes diseños de la interfaz que no permiten explotar toda la utilidad de un sistema. Algunos de los aspectos que se deben considerar para el diseño de una interfaz son:

- Tipo de usuario a quien está dirigido el sistema.
- Interfaz intuitiva para poco o nulo entrenamiento especializado.
- Rica y dinámica interacción con el usuario.
- Eficiencia y transparencia en el acceso a los datos.
- Recursos con los que cuenta para su funcionamiento.
- Recursos a los que necesita acceder para interactuar.
- Obtención de software estándar, ligero, eficaz, reusable e independiente de plataforma.

Cada uno de estos puntos ayuda a obtener un diseño de interfaz con una amplia usabilidad del sistema, integrando eficiencia y efectividad para permitir alcanzar la satisfacción de los usuarios.

La interfaz de nuestro prototipo debe contener muchos de los aspectos antes mencionados, además de considerar que debe ser mostrada a través de un navegador.

El diseño de la interfaz de la página Web está basado en el *principio KISS*<sup>5</sup> y el diseño de interfaz multinivel<sup>6</sup> de Shneiderman; con ello, se busca construir una interfaz usable, ligera y de acceso controlado a los datos. En la figura 7.8, observamos en primera instancia el diseño que incluye la separación de los distintos niveles de interacción con el usuario.

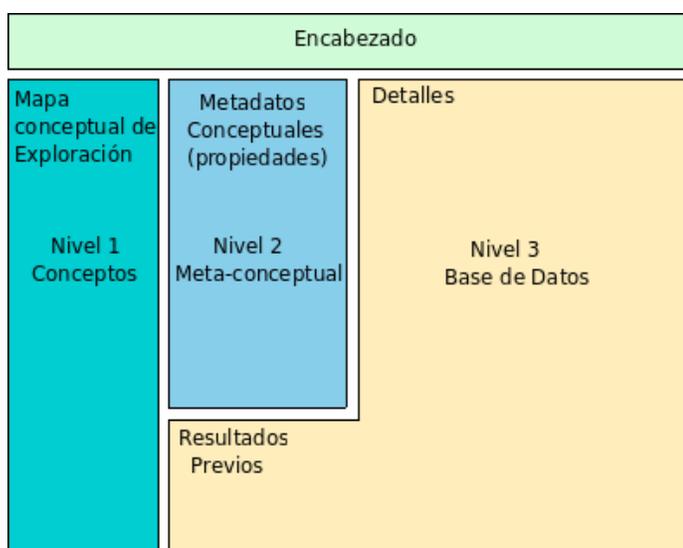


Figura 7.8: Diseño de la interfaz multinivel que es mostrada en el navegador y permite un acceso controlado a los mapas de exploración y a los datos.

El diseño de interfaz mostrado en la figura 7.8, presenta los tres niveles de interacción denominados como *Mapa Conceptual de Exploración*, *Metadatos Conceptuales (Propiedades)* y *Detalles*. El nivel uno se encarga de mostrar los mapas de exploración para los diferentes conceptos dominantes, así como los submapas contenidos en el interior de cada uno de ellos. La interfaz debe permitir una navegación de manera gradual, hasta lograr alcanzar un concepto muy específico o nodo terminal.

---

<sup>5</sup>En la sección 3.3.3, se describe este concepto.

<sup>6</sup>Recordemos que en la sección 3.1.6, damos este nombre al diseño de interfaz multicapa propuesto por Sheiderman.

En el nivel dos de la interfaz denominado *Propiedades*; muestra el mapa meta-conceptual donde podremos observar los catálogos a los que hace referencia el nodo terminal. En este nivel podemos seleccionar los diferentes valores de interés, para que sea realizada una consulta. Por último, el nivel tres denominado como *Resultados Previos*, es usado para mostrar todos los resultados obtenidos; cada uno de ellos puede ser explorado de una manera particular en la parte denominada como *Detalles*.

Los tres niveles de interacción mostrados en la interfaz necesitan de datos en la parte servidor y conforme se van obteniendo los datos, la interfaz debe actualizarse de manera automática en el nivel correspondiente, sin requerir una actualización de toda la página, porque el servidor solo nos devuelve como respuesta datos muy concretos y específicos.

El nivel 0 de interacción mostrado en la capa dinámica, se refiere exclusivamente a actualizaciones inmediatas locales, sin requerir datos del servidor. Algunos ejemplos del nivel de interacción 0 son el cambio de color, ubicación, borrado o modificación de un elemento existente en la interfaz. Todas las actualizaciones en este nivel deben ser realizadas a través de DOM, desde donde podemos modificar toda la estructura interna del documento.

### 7.3.2. Interacción en el prototipo

Una vez que fueron definidas las regiones para cada nivel de interacción es necesario definir cómo llevar a cabo la interacción visual, ya que puede ser realizada de muy diversas maneras; en este proyecto y para optimizar el tiempo de respuesta en la interfaz usamos botones de formularios, aunque debemos resaltar que se pueden utilizar otros elementos gráficos.

Los botones usados son creados de manera local en el lado cliente, lo único que se necesita es recuperar del servidor el nombre que deberán mostrar. De esta manera, sólo requiere del servidor unos pocos datos de texto, lo que significa pocos bytes; esto proporciona un transporte ligero de datos del servidor al cliente.

La figura 7.9 muestra la vista inicial, donde podemos ver dos botones iniciales en la parte de arriba, que nos indican el lenguaje de exploración, que en este caso puede ser español o inglés. Después de seleccionar el idioma, inmediatamente surgen las opciones que contiene la entidad cepa.



Figura 7.9: Vista de la interfaz inicial, después de seleccionar el idioma para la exploración, las opciones de cepa aparecen en el idioma indicado.

Cuando seleccionamos alguno de los botones mostrados, se entrará a otro nivel de exploración, hasta lograr alcanzar un concepto específico en un nodo terminal. Cada vez que es presionado un botón o elemento, se disparan eventos que necesitan ser escuchados y capturados. Dependiendo del nivel de interacción en el que un evento es capturado, se obtiene el contenido del elemento y se analiza para validar una petición asíncrona al servidor. El servidor responde a la petición y los datos son incorporados al nivel de interacción correspondiente, anexando escuchas a los elementos que sirven a la interacción. Si alcanzamos la parte final de un nivel, se debe pasar inmediatamente al siguiente, para crear todos los elementos necesarios en ese nuevo nivel.

La figura 7.10, nos muestra la navegación en el mapa conceptual hasta llegar a un concepto final, cuando presionamos en la entidad celular, nuevos datos son adquiridos para ser incorporados al nivel 2 *Meta-conceptual*. A todos los elementos integrados se les asignan escuchas de manera automática para que puedan responder a las acciones de los usuarios. Los elementos en este nivel contienen catálogos relacionados al concepto semántico y en ellos podemos seleccionar los valores de búsqueda para ese concepto.

La figura 7.11 muestra los resultados de la búsqueda para bacteria y morfología celular, con algunas opciones de los catálogos.



Figura 7.10: Vista que muestra el nivel 2 de interacción, donde se seleccionan las propiedades para la consulta.

La región asignada para resultados en el nivel 3, muestra todos los datos recuperados que cumplen las condiciones seleccionadas. Desde esta parte únicamente es necesario presionar en los números de identificación de cada resultado mostrado y los detalles concernientes a las cepas con el número de identificación, son mostrados en el nivel 3 de *Detalles*<sup>7</sup>.

Si deseamos cambiar los valores de búsqueda o concepto semántico, sólo navegamos nuevamente en el nivel 1 y podemos realizar una nueva búsqueda con entidades concepto y valores diferentes. Si se desea cambiar de lenguaje de exploración, sólo se debe presionar en los botones de lenguaje y los niveles se vacían para volver a empezar otra exploración de datos.

La arquitectura de este sistema de exploración fue implementada con XHTML estricto para las páginas Web, CSS 1.0 para la presentación de los elementos, DOM para el acceso a los elementos y JavaScript en la parte dinámica del lado cliente. En el lado servidor usamos el servidor Web Apache 2.2, como contenedor de objetos java usamos a Tomcat 6.0.18 y como administrador de bases de datos a PostgreSQL 8.3.5. Cabe recordar que la arquitectura genérica presentada en el capítulo anterior, nos permite usar diferente software al mostrado para este ejemplo.

<sup>7</sup>Los detalles y fotos de la figura 7.11, sólo son para pruebas de recuperación de información y no corresponden a los datos reales de esa cepa en particular.



Figura 7.11: Vista que muestra el nivel 3, donde se puede explorar cada registro obtenido de la base de datos.

## 7.4. Resultados

El tiempo en que los datos son recuperados en un sistema puede impactar directamente en su éxito o fracaso. El prototipo mostrado cuenta con una alta eficiencia en la recuperación de los datos, porque utiliza diversas técnicas de implementación y obtiene amplias ventajas con respecto a implementaciones tradicionales. La presente sección muestra algunas de las ventajas obtenidas por la implementación de nuestro prototipo bajo la arquitectura MD.

En la primera parte se presentan algunas ventajas sobre la transmisión asíncrona de datos; después se muestran las diferencias existentes en el uso de los distintos formatos de los datos.

### 7.4.1. Comunicación asíncrona de la arquitectura MD

La naturaleza de la arquitectura MD es la comunicación asíncrona de los datos. La capa dinámica utiliza un motor asíncrono para la solicitud y recuperación de los datos (ver figura 6.3). Como sabemos, una de las partes más importantes en la arquitectura es la que concierne a la recuperación de la información; cuando la recuperación de los datos es eficiente, impacta directamente en el rendimiento de un sistema.

Existen dos maneras de recuperar los datos, una es a través de la comunicación síncrona, que es la manera tradicional y que ocupan la inmensa mayoría de los sistemas Web; la otra opción es por medio de la comunicación asíncrona. Ahora tratemos de entender las diferencias entre estas dos maneras de comunicación entre el cliente y el servidor.

Los sistemas tradicionales Web usan la comunicación síncrona en sus páginas estáticas y en sus páginas dinámicas; cuando un usuario presiona una liga o envía datos, necesita esperar toda una nueva página con la información que es solicitada. Las páginas estáticas no cambian en el tiempo y su navegación en otras páginas es por medio de ligas entre páginas.

Las páginas dinámicas trabajan de manera diferente, porque son construidas en tiempo de ejecución en el servidor. Las páginas dinámicas usan formularios como medio de interacción y la respuesta necesita de la creación de una nueva página Web de manera automática; esto significa generar una nueva página en tiempo de ejecución e incorporar los nuevos datos y enviarla como respuesta al cliente que la solicitó; el cliente no puede seguir interactuando hasta que la página completa es obtenida. Cada solicitud realizada debe construir y generar nuevas páginas en el lado servidor y ser transportadas en su totalidad para ser mostradas al cliente.

La comunicación asíncrona<sup>8</sup> usada en nuestra arquitectura MD, trabaja de manera diferente. La interfaz espera por eventos a través de escuchas en los elementos contenidos en la página, cuando estos eventos son capturados por funciones de escucha, debe verificar si se necesitan realizar peticiones externas; si este es el caso, entonces se generan peticiones asíncronas al servidor. Cuando el servidor recibe una solicitud, verifica los datos y entonces se conecta a la base de datos para recuperar los datos solicitados.

Una vez que son recuperados los datos, se les anexa un carácter que marca su relación con los datos solicitados. Después los datos son enviados al motor asíncrono que realizó la solicitud. Cuando los datos son obtenidos en la parte cliente, estos son incorporados a la página en el lado cliente a través de la generación automática de sus respectivas etiquetas XHTML. Es necesario señalar que el servidor, no necesita la construcción ni el envío de toda una página completa, únicamente envía los datos solicitados.

---

<sup>8</sup>El motor que permite la comunicación asíncrona, fue explicado con mayor detalle en la sección 6.4

## CAPÍTULO 7. PROTOTIPO

---

La eliminación en la generación y transportación de una página completa en cada solicitud, permite eliminar bastante carga de trabajo al servidor. Para mostrar esta diferencia, obtenemos una comparación sobre la realización de una consulta por el método tradicional de implementación Web y el método asíncrono usado por nuestra arquitectura MD.

La tabla siguiente muestra la cantidad de bytes recuperados y transportados en cada solicitud al servidor, por medio de la comunicación síncrona y asíncrona dinámica usada en nuestro prototipo.

Solicitudes	Síncrona	Asíncrona
1	4144	4144
2	4359	67
3	4936	132
4	5147	48
5	7038	482
6	9293	504
7	10519	524

Si obtenemos la gráfica de la siguiente tabla, podemos obtener una mejor vista de lo que ocurre en la transmisión de los datos.

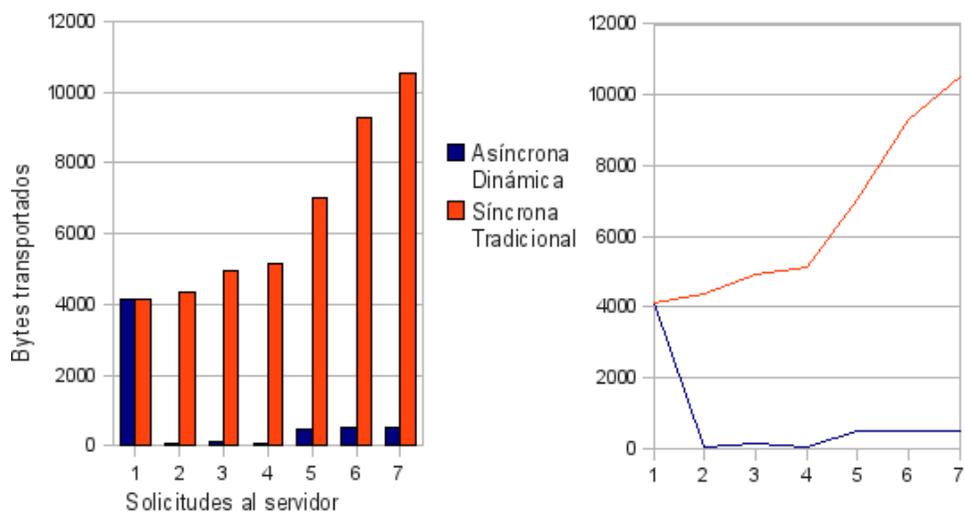


Figura 7.12: La gráfica muestra la cantidad de datos requeridos para la interacción por comunicación asíncrona dinámica usada en nuestro prototipo, respecto a la comunicación síncrona usada en sistemas Web tradicionales.

Podemos observar que la descripción entre la comunicación síncrona y asíncrona es bastante similar, pero la eficiencia obtenida en la recuperación de datos es muy diferente.

La gráfica mostrada en la figura 7.12, presentada en dos estilos diferentes, ayuda a visualizar la diferencia de datos recuperados y transportados entre las dos diferentes maneras de comunicarse con el servidor. Podemos observar que en la primera solicitud se recupera la misma cantidad de información, pero después existe una recuperación diferente de datos.

Cuando las solicitudes son hechas de manera síncrona, la página necesita transportar toda la estructura interior de una página e incorporar la información obtenida de la base de datos; además el servidor debe construir y enviar la estructura de la página todas las veces.

Por otro lado, cuando las solicitudes son hechas a través de la comunicación asíncrona dinámica sólo son recuperados los datos requeridos. Esto evita una transportación de datos completamente innecesaria, ambigua y redundante. Esto muestra evidencia de la diferente cantidad de trabajo realizado por el servidor, ya que existe una fuerte separación en la cantidad de datos administrados y enviados hacia el lado cliente que hace la solicitud. Si este proceso es repetido miles de veces, observaremos una separación exponencial entre las diferentes cantidades de datos administrados y enviados.

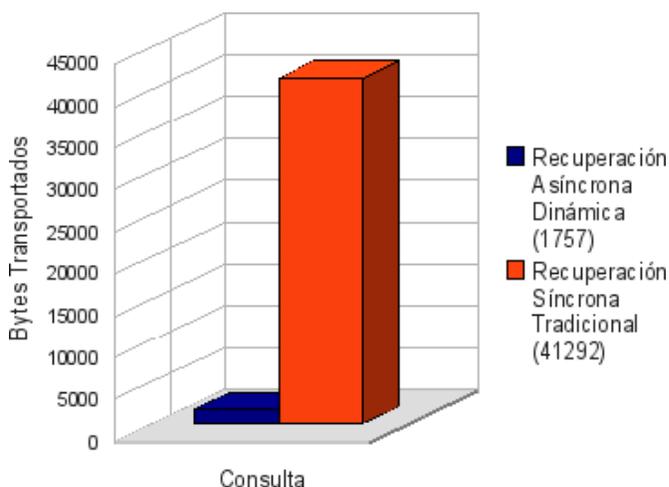


Figura 7.13: La gráfica muestra la cantidad total de datos recuperados en las siete peticiones realizadas, por el método asíncrono dinámico y el método tradicional; podemos observar una amplia diferencia.

La figura 7.13, presenta una gráfica donde podemos observar una enorme diferencia en la cantidad de datos transmitidos, ya que la parte síncrona necesitó recuperar 41292 bytes, esto significa una diferencia de 39535 bytes adicionales, para mostrar exactamente los mismos datos. Esto nos muestra que la eficiencia en la recuperación asíncrona dinámica implementada en nuestro sistema prototipo es muy superior, respecto a implementaciones tradicionales. Esto explica en gran medida la percepción de la actualización inmediata en la interfaz del sistema.

### 7.4.2. Formato para recuperación de datos

Aún cuando la recuperación asíncrona permite eficiencia en la recuperación de datos, esta recuperación puede hacerse de formas diferentes, ya que la información obtenida puede tener un cierto formato de recuperación. Estos formatos pueden ser a través de recuperación de estructuras en XML, el formato JSON, etiquetas XHTML o texto plano.

La recuperación de datos en la comunicación asíncrona, es realizada de forma común en formato XML, pero debemos de recordar que una estructura XML contiene toda una especificación en su construcción, la cual necesita de etiquetas de descripción de la información por cada dato contenido.

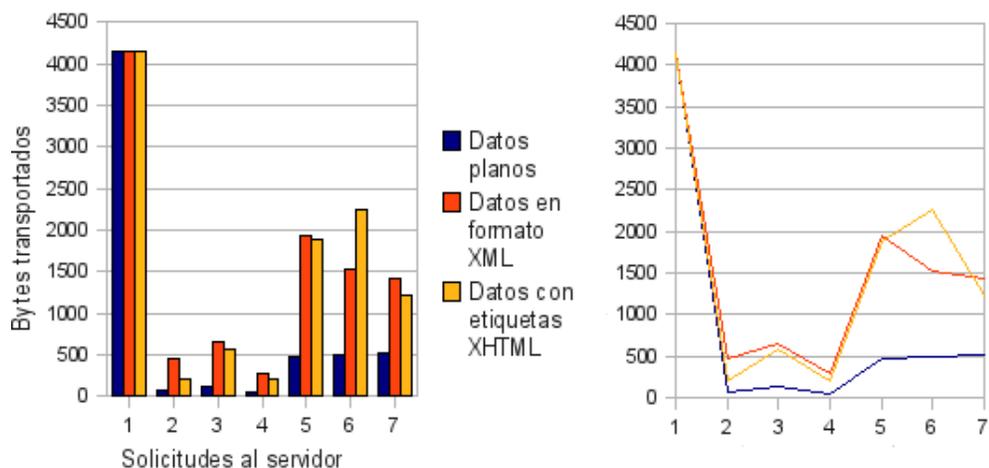


Figura 7.14: La gráfica muestra la cantidad de datos transportados en las diferentes consultas, a través de diferentes formatos para recuperación de datos asíncronos.

Esto nos dice que cada dato necesita estar contenido en dos etiquetas que lo describan, una de inicio y una etiqueta final. Cada dato contiene estas dos etiquetas y otras dos más para cada serie de datos diferentes obtenidos en la respuesta, sin olvidar otras etiquetas de estructura general de los datos.

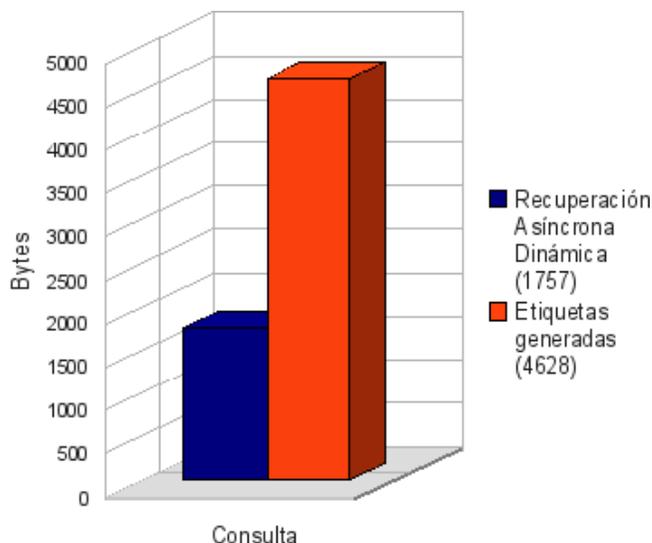


Figura 7.15: La gráfica muestra la cantidad de etiquetas generadas en el lado del cliente y que no fueron transportadas en las solicitudes al servidor.

Al igual que en la parte síncrona, nos indica una innecesaria cantidad de datos a transportar, ya que únicamente nos interesan los datos y no sus etiquetas en este caso muy específico. Lo mismo sucede con el formato JSON que sólo presenta ligeras variaciones con respecto al formato XML, pero este no lo consideramos porque necesita de librerías externas de software, tanto en la parte cliente como en la parte servidor. Otra forma es construir las etiquetas en XHTML directamente en el servidor, para que puedan ser incluidas de manera directa en la página Web. Al igual que en XML las etiquetas en XHTML necesitan cumplir ciertas especificaciones como el inicio y terminación de etiquetas, lo cual nos regresa al mismo problema de transportación innecesaria de datos. Además esto significa una mayor cantidad de procesamiento por parte del servidor y aunque esto todavía es eficiente podemos eficientar al máximo el traslado de datos por medio del transporte de texto plano con unos pocos caracteres marcadores dentro de la información recuperada, como se muestra en la siguiente tabla.

## CAPÍTULO 7. PROTOTIPO

Datos	XML	XHTML	T. Plano
Dato1	<nombre>Dato1</nombre>	<span>Dato1</span>	Dato1
5	22	18	6

En la tabla podemos observar la cantidad de caracteres que necesitan ser transportados en diferentes formatos por cada dato. Podemos observar que el dato mismo consta de 5 caracteres, pero al incorporarlo a un formato específico, la cantidad de caracteres para obtener el formato de recuperación superó ampliamente al dato mismo en cantidad. Por decirlo de otra forma, del total de la información recuperada, sólo una mínima parte corresponde a los datos; mientras que la gran totalidad pertenece a etiquetas que hacen referencia a esos pocos datos, donde en la inmensa mayoría de los casos estas referencias no se utilizan.

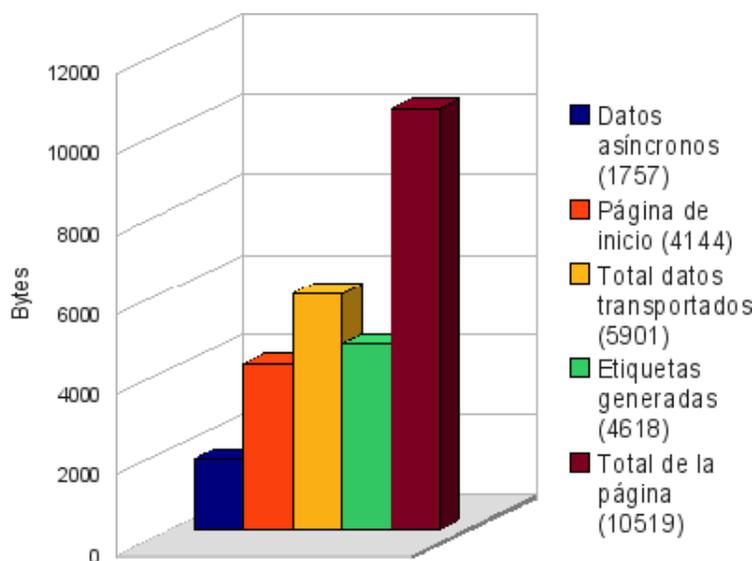


Figura 7.16: La gráfica muestra los datos totales obtenidos en la consulta, hasta alcanzar el total de la página

La figura 7.14, permite observar las diferentes cantidades de datos a transportar, según el formato elegido para obtener los datos; claramente podemos observar que el texto plano con un carácter de separación, obtiene una mayor eficiencia y menor cantidad de datos innecesarios. Por consiguiente también es claro observar que el servidor necesita realizar una menor cantidad de trabajo al tener una menor cantidad de datos para procesar, preparar y enviar.

El prototipo trata de eliminar todas estas etiquetas innecesarias, por lo que utiliza el texto plano; pero adiciona un carácter que permite indicar la separación entre datos, como se muestra en la tabla anterior. Una vez obtenido este dato la capa dinámica del modelo MD, se encarga de separar el código y crear de manera dinámica la etiqueta XHTML que permita la incorporación del dato en la página. La capa dinámica no sólo se encarga de construir la etiqueta, también asigna un identificador, escuchas para los eventos que necesite y un estilo correspondiente, según el elemento DOM donde se incorporará el dato. Esto elimina mucho trabajo en la parte servidor y delega una buena parte al lado cliente.

La gráfica en la figura 7.15, nos permite mostrar la carga de trabajo que es eliminada de la parte servidor, con esto se logra una mayor respuesta en la interfaz del sistema prototipo.

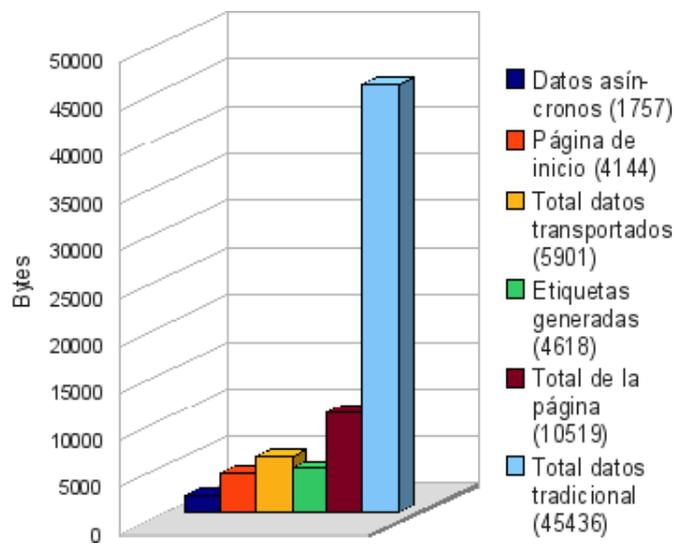


Figura 7.17: La gráfica muestra los bytes totales producidos por las diferentes etapas y comparada con los datos del método tradicional.

Cuando observamos la cantidad total de los datos en la página Web para permitir mostrar los resultados finales, podemos observar que la cantidad de datos es mayor a la transmitida y recuperada, esto se debe a que la totalidad de las etiquetas es generada en la parte cliente. La gráfica de la figura 7.16 muestra de manera visual este proceso; donde la primera barra representa los datos asíncronos transmitidos junto con el total de la página inicial,

la tercera columna nos muestra todos los datos transmitidos los cuales son 5901 bytes en total. Pero la página final nos muestra un total de 10519 bytes representados por la última columna. Pero la diferencia es generada de una manera automática en el lado cliente, lo cual representa 4618 bytes.

Con todos estos datos es fácil observar que de la totalidad en bytes de la página mostrada al usuario por el prototipo, únicamente se transmitió un 56 % por la red, el otro 44 % restante fue generado en la parte cliente de una manera automática. La gráfica de la figura 7.17, presenta que la comunicación asíncrona dinámica en relación 5901 a 45436, sólo transmite un 13 % de bytes de los necesitados por la comunicación síncrona, es decir que se transmitió 87 % menos código, etiquetas y datos por la red. La relación inversa nos señala que la aplicación síncrona necesita 770 % más bytes en relación a los bytes necesitados por nuestro prototipo. Esto hace evidente que la comunicación síncrona usada por las implementaciones tradicionales Web, son ineficientes comparadas con la comunicación asíncrona dinámica que es usada por la arquitectura MD.

### 7.5. Comentarios Finales

En este capítulo se presentó una base de datos de microorganismos, que contiene una cierta complejidad semántica. La base de datos funciona muy bien como caso de estudio y permite aplicar todas las características que contiene el modelo MD. El prototipo en primera instancia necesitó de una estructuración de los datos para obtener los mapas conceptuales de navegación. Después se diseñó la parte de la interfaz, aplicando la parte visual-dinámica del modelo. La arquitectura MD, permitió llevar a una implementación real del prototipo.

El prototipo mantiene una interfaz basada en navegación de mapas conceptuales, que mantienen una respuesta inmediata a las solicitudes del usuario. La manera en que alcanza una respuesta inmediata en la interfaz es mostrada en la sección de resultados, donde podemos observar la ventaja en la transición de datos para obtener esta propiedad de respuesta inmediata en la interfaz. La necesidad de la implementación de este prototipo fue con el fin de demostrar que el modelo y la arquitectura MD, pueden ofrecer soluciones reales a problemas reales.



## Capítulo 8

# Conclusiones

Una vez obtenido el prototipo y mostrada la parte de estructuración, interacción y comunicación con el servidor, necesitamos una discusión sobre el estado que mantienen otros sistemas de exploración en la literatura científica, con respecto al prototipo. Una vez realizada esta discusión se presentan las conclusiones finales, así como las contribuciones, implicaciones y limitaciones del modelo y la arquitectura MD.

### 8.1. Discusión

Diversos sistemas han sido implementados con la finalidad de facilitar la exploración de datos, pero muy pocos han sido orientados a datos científicos. Algunos de estos sistemas son un poco rígidos y sólo hacen uso de formularios para el acceso a sus datos, como puede verse en [30], [51].

Otros sistemas de exploración más sofisticados hacen uso de metadatos; para ello crean una base de metadatos la cual utiliza un agente que localiza los datos, pero estos son estáticos y de un único tipo como en [38], [35], diferentes a la naturaleza de los datos biológicos que muchas veces son heterogéneos, variables y dependientes del contexto.

El uso de semántica presenta el mismo caso; la exploración es principalmente usada para archivos multimedia como fotos o vídeos, sobre los cuales se da una descripción de un mismo tipo de dato, estos sistemas necesitan un administrador de relaciones, ejemplos de este uso pueden verse en [97], [39].

El paradigma de manipulación directa parece ser una buena opción para la exploración de datos, pero los sistemas que son eficientes bajo este concepto únicamente han sido implementados en sistemas de escritorio, como puede verse en [62], [41]. Bajo este concepto hay un sistema de escritorio muy interesante que utiliza regiones semánticas para la exploración, aunque sólo está orientado a la exploración de fotos, podemos verlo en [94].

Un acercamiento al desafío de la exploración de datos en una compleja base de datos, hace uso de un framework que identifica todas las rutas de un esquema lógico. Esto genera esquemas con muchas relaciones cruzadas y está orientado a que el usuario pueda comprender la estructura interna de una base de datos, para que pueda realizar consultas a una base de datos, como puede verse en [131].

Por la parte Web, existen pocos sistemas orientados a la exploración. Un sistema de interés podemos verlo en [61], el cual utiliza el paradigma de interfaz de escritorio y crea una exploración a través de un explorador de archivos contenido en los sistemas operativos; los directorios representan tablas y sus relaciones, pero como se menciona en el mismo documento, presenta mucha redundancia y poca integridad relacional entre entidades.

Los mejores acercamientos para este tipo de sistemas en la Web han sido implementados por Ben Sheiderman y su orientación de consultas dinámicas; aunque estos sistemas tienen el inconveniente de sobrecargar el lado cliente. Uno de sus ejemplos usa un Applet de java, pero la aplicación es muy pesada y en ciertos casos resulta ineficiente, puede verse en [64]. Otro ejemplo que trata de evitar esta sobrecarga por la aplicación en el lado cliente, hace uso de DHTML como fuente de interacción y evitando el software adicional, pero tiene el inconveniente de sobrecargar la parte cliente con cientos de capas de datos, que permanecen a la espera de poder ser mostradas con un cambio de estilo. Esta interesante aplicación puede verse en [66].

Es claro observar que existen diversas investigaciones orientadas a la exploración de los datos y que es un área que necesita de nuevas propuestas, que puedan resolver varias de las desventajas mostradas.

Nuestro modelo es una de estas propuestas que presenta ventajas frente a conceptos tradicionales como metadatos, lenguajes visuales, etc. Todos las investigaciones orientadas a la exploración de datos, nos dejan ver una fuerte necesidad de buscar nuevas formas de realizar la exploración en bases de datos.

El nuevo modelo MD utiliza una integración de los conceptos tradicionales como: consultas dinámicas, metadatos, semántica y consultas visuales. El modelo MD ofrece una integración a través de capas, que se comunican entre sí, para que los diferentes conceptos puedan trabajar de manera conjunta y con funciones muy específicas. Desde este punto de vista, es fácil observar que obtenemos muchas ventajas sobre los sistemas de exploración que se han descrito anteriormente.

La orientación inicial de nuestro modelo es la de permitir la exploración de datos de manera transparente, sin la necesidad de conocer la compleja estructura interna o datos de la compleja base de datos. El modelo hace uso de una semántica, no para describir los datos o el contenido de estos, sino para navegar en mapas conceptuales semánticos, que nos direccionan hacia catálogos puntuales en la base de datos. Los metadatos son modificados en su esencia, por lo que extendemos su aplicación y los denominamos como metadatos conceptuales. Esta nueva clase de metadatos une la semántica inmersa en los mapas conceptuales, con el esquema lógico de la base de datos.

De esta manera, observamos que aunque utilizamos los mismos conceptos como los metadatos y la semántica en implementaciones tradicionales, la manera de uso en este modelo es muy distinta y con un propósito definido. Con toda esta estructura actuando de manera oculta para el usuario, es necesaria una nueva arquitectura que permita a toda esta implementación un uso práctico. La arquitectura MD soporta completamente al modelo MD, para que pueda ser usado en sistemas Web.

Por otra parte, la interfaz Web requiere de un diseño que adapte los conceptos de usabilidad, interacción y actualización dinámica. El modelo de interacción multi-niveles que es propuesto por Sheiderman es adherido al concepto de interacción visual, para proporcionar un uso casi nulo del teclado. Pero esto no es suficiente, ya que la parte dinámica permite alcanzar la amplia utilidad del sistema; para lograrlo se manipula la estructura interna del documento y se actualiza de manera constante en modo asíncrono, enriqueciendo la interacción con los datos.

Para evitar la sobrecarga en el lado cliente con capas de datos o software externo, se utilizan diversas técnicas sobre tecnologías abiertas para la implementación Web.

De este modo, técnicas como *código no invasivo*, *escucha y manipulación de eventos*, *manipulación DOM*, *manipulación de estilos*, *petición asíncrona*,

*cambio de parámetros, re-uso de clases, envío simplificado, formateo de cadenas, Cross Browser*, nos permiten obtener un sistema de información Web muy ligero en el lado cliente, sin requerir ningún tipo de software adicional o sobrecarga de datos.

La parte cliente también es beneficiada porque se baja la carga de procesamiento, ya que no se requiere de construir las páginas completas, únicamente se requiere el paso de los datos; de esta manera evitamos el transporte de código innecesario o redundante, haciendo eficiente la comunicación al transportar una cantidad reducida de datos.

Todas estas ventajas nos ofrece una nueva manera de desarrollar sistemas Web, que pueden permitir una nueva generación de sistemas orientados a la exploración en bases de datos científicas.

La arquitectura MD nos permite implementar sistemas híbridos que contienen una estructuración de datos, basada en metadatos y semántica, con un ingrediente adicional, una interfaz visual-dinámica que ofrece una rica interacción con los usuarios. Una discusión sobre los sistemas híbridos en base a la integración de diferente conceptos, puede ser encontrada en [82].

## 8.2. Conclusiones

La información científica y biológica necesita de nuevas maneras para interactuar con los datos. Diversos científicos hacen el llamado a científicos en computación para realizar trabajo interdisciplinario, que permita el desarrollo de nuevas herramientas computacionales. La propuesta del nuevo modelo MD surge precisamente de este llamado y la necesidad de explorar los datos en bases de datos científicas.

El trabajo interdisciplinario entre el Departamento de Computación y la Colección Nacional Mexicana de Cepas Microbianas, permitió comprender la esencia del problema de la exploración de datos en bases de datos científicas con información biológica. La comprensión de este problema necesitó de la revisión de los diferentes conceptos utilizados en el desarrollo de sistemas de exploración; pero ninguno tenía el soporte suficiente para el tipo de información a explorar. Esto motivó en gran parte el desarrollo de un nuevo modelo de interacción que permitiera ofrecer una solución al problema central de la exploración de datos, debido a que las implementaciones tradicionales no permiten ofrecer una solución práctica a este problema.

La idea principal en este nuevo modelo, está basada en permitir la integración de diferentes características de los diferentes conceptos usados en la exploración de datos. Alcanzar esta integración resultaría imposible sin la adopción de su implementación a través de diferentes capas, con cada uno de estos conceptos interactuando y trabajando de manera coordinada.

El nuevo modelo MD pretende ser genérico para que pueda ser adoptado en la resolución de diferentes problemas, aunque este puede parecer complejo en un inicio; la comprensión de cómo cada concepto es utilizado elimina mucha de esta complejidad facilitando su adopción. No podemos afirmar que el nuevo modelo sirve para todas las situaciones, porque inicialmente ha sido orientado a la exploración de datos, pero podemos decir en base a la experiencia obtenida, que puede ser extendido para diferentes tipos de aplicaciones.

Para la implementación del modelo en sistemas reales, fue necesario obtener la arquitectura MD que soporta este modelo. La arquitectura puede presentar algunos aspectos con ciertas deficiencias iniciales, pero a medida que se madura en la experiencia de este tipo de aplicaciones, pueden ser subsanadas en corto tiempo. La arquitectura y la implementación del prototipo, prometen una nueva generación de aplicaciones Web y la interacción de los usuarios con los datos, especialmente en el área científica, así como la adopción de nuevos paradigmas de interacción para sistemas basados en la Web, que cuenten con interacción visual y una respuesta dinámica.

### 8.3. Contribuciones

En esta sección se describen las principales contribuciones, que se aportan en este trabajo de investigación.

- *El modelo MD*: Este modelo es único en su tipo, ya que integra el trabajo conjunto de diversos conceptos como: mapas conceptuales, semántica, metadatos, construcción automática de consultas, consultas visuales y consultas dinámicas. Este modelo ha sido principalmente orientado a la exploración de datos, aunque tiene un amplio potencial de uso para otras aplicaciones prácticas.
- *La arquitectura MD*: Esta arquitectura es una extensión de la arquitectura cliente-servidor, porque incorpora nuevas capas adicionales que le permiten ofrecer un rica interacción con el usuario y mantienen una óptima recuperación de datos.

La nueva arquitectura propuesta es abierta, genérica, con independencia de software, plataforma y navegador, lo cual permite su amplio uso en sistemas Web.

- *Metodología de implementación:* La incorporación de nuevas capas necesita una metodología para su implementación. De esta manera, la estructuración semántica y conceptual de los datos es presentada, así como la manera en que la interfaz visual mantiene una respuesta inmediata a las acciones del usuario.
- *Prototipo:* El prototipo de exploración de datos muestra diversas ventajas frente a otros sistemas como: eficiencia en el transporte de los datos, eficiencia en la actualización, exploración de datos sin que el usuario requiera conocer la estructura interna de la base de datos, acceso transparente de datos, poca cantidad de código en la implementación y uso interactivo por parte del usuario.

## 8.4. Implicaciones

El modelo y la arquitectura MD se diseñaron de manera genérica, para permitir un uso versátil en una amplia variedad de problemas. Este primer acercamiento es hacia una base de datos científica con datos biológicos, pero puede ser adaptada a la interacción con otro tipo de bases de datos científicas, que quizás no contengan datos biológicos.

Las ventajas ofrecidas por el modelo y la arquitectura MD en el caso de estudio son diversas frente a otros conceptos utilizados en arquitecturas tradicionales cliente-servidor. El modelo y la arquitectura MD son nuevos pero la experiencia con el prototipo desarrollado, muestra que puede ser adaptado para diferentes tipos de procesamiento de información. Esto por supuesto requiere de mejores diseños de interfaz y creación de elementos visuales orientados a facilitar la experiencia del usuario. El uso de técnicas como arrastrar y soltar (drag and drop), pueden permitir una mayor interacción para cierto tipo de datos como: minería de datos, búsqueda de conocimiento, simulación de cambios en datos o búsqueda visual distribuida, pueden ser algunas de la aplicaciones inmediatas de nuestra arquitectura. Otra aplicación de interés puede ser en bases de datos con información genómica, para la búsqueda de genes relacionados a funciones específicas, simulación de mutaciones o supresión en cadenas cortas de RNA (Ácido Ribonucleico), localización de nucleótidos o aminoácidos en secuencias, etc.

El amplio uso de esta arquitectura puede generar también un cambio radical en los métodos tradicionales de búsqueda y evitar el problema del *Web oculto*<sup>1</sup> (*hidden Web*), con agentes que puedan navegar en las bases de datos de los sistemas a través de conceptos.

La unión de semántica conceptual, metadatos de referencia, interacción visual y dinámica, nos presenta una nueva manera de crear sistemas Web más efectivos en la exploración de datos. Pero esto también provoca una nueva generación de sistemas Web híbridos, que permitirán la creación de nuevas herramientas en la Web futura.

## 8.5. Limitaciones

La implementación de la arquitectura semántica observa algunas limitaciones como:

- La arquitectura es dependiente de la Web y del uso de un navegador en la interacción.
- Requiere de desarrollo de mapas de exploración en la capa de conceptos y la capa meta-conceptual.
- Necesita de administradores para el manejo de la capa conceptos y meta-conceptual.
- Requiere del uso de diversas nuevas técnicas de desarrollo Web, para obtener un comportamiento eficiente.
- El modelo y la arquitectura son nuevos y requieren mayor investigación para la creación de sistemas más maduros.

---

<sup>1</sup>Para obtener más información sobre este problema, ver la referencia [132].

## 8.6. Perspectivas

Tenemos dos tipos de perspectivas para este trabajo de investigación, las cuales son las generales orientadas a la investigación misma y las particulares orientadas al prototipo desarrollado. Las perspectivas son:

*Generales:*

- Tratar de automatizar la construcción y desarrollo de sistemas basados en la arquitectura MD.
- Crear un framework que permita generar los mapas de exploración de manera automática, para ser usados por un sistema de este tipo.
- Generar un administrador de mapas conceptuales semánticos y meta-conceptuales, que soporte el nuevo framework.
- Desarrollar nuevas aplicaciones con el uso de la técnica de arrastrar y soltar (drag and drop), para diferentes tipos de procesamiento de datos.
- Uso de la arquitectura MD, para el desarrollo de un sistema basado en búsqueda de conocimiento a partir de procesos visuales.

*Particulares del prototipo:*

- Convertir el prototipo en una versión final para uso de la comunidad científica.
- Ampliar las capacidades del prototipo, para permitir obtener consultas con una mayor complejidad.
- Adicionar nuevas capacidades visuales para mejorar la interacción con los datos.
- Utilizar la arquitectura MD, para el desarrollo de la vista de altas de nuevos datos.

# Bibliografía

- [1] The 2020 Science Group, “Towards 2020 Science,” *Towards 2020 Science*, Microsoft Research, pp. 1-82, 2006.
- [2] Roger Brent, Jehoshua Bruck, “Can Computers Help to Explain Biology?,” *Nature*, Vol. 440, pp. 416-417, March, 2006.
- [3] Jaques Cohen, “Bioinformatics An Introduction for Computer Scientists,” *ACM Computing Surveys*, Vol. 36, No 2, pp. 122-158, Jun. 2004.
- [4] A. Finkelstein, J. Hetherington, Linzhong Li, “Computational Challenges of Systems Biology,” *Computer*, pp. 26-33. May, 2004.
- [5] Marjory Blumenthal, “Frontiers at the Interface between Computing and Biology,” *IEEE Intelligent Systems*, pp. 11-13, March-April, 2002.
- [6] Peter Buneman, Serge Abiteboul, Alex Szalay, Andre Hagehülsman, “Semantics of Data,” *Towards 2020 Science*, Microsoft Research, pp. 15-16, 2006.
- [7] Ehud Shapiro, David Harel, Christopher Bishop, Stephen Muggleton, “The Fundamental Role of Computer Science Concepts in Science,” *Towards 2020 Science*, Microsoft Research, pp. 24-25, 2006.
- [8] Junhyong Kim, “Computers are from Mars, Organisms are from Venus,” *Computer*, pp. 25-32, July 2002.
- [9] John V. Carlis, “Pitfalls and Paths to Success in Interdisciplinary Scientific Database Research,” *IEEE Engineering in Medicine and Biology*, Nov-Dec, 2002.
- [10] Lawrence Hunter, Richard H. Lathrop, “Computer Science and Biology: An Unlikely Pair,” *IEEE Intelligent System*, pp. 8-10, March-April, 2002.
- [11] Yongwang Zhao, Dianfu Ma, Chunyang Hu, Min Liu, Yonggang Huang, “SOCOM: A Service-Oriented Collaboration Middleware for Multi-User Interaction with Web Services based Scientific Resources,” *Sixth International Symposium on Parallel and Distributed Computing (ISPDC'07)*, 2007.
- [12] Hiroki Kitano, “System Biology: A Brief Overview,” *Science*, Vol. 295, pp. 1662-1664, March, 2002.
- [13] Alexander Szalay, Jim Gray, “Science in an Exponential World,” *Nature*, Vol. 440, pp. 413-414, March, 2006.
- [14] John H. Reif, “The Emerging Discipline of Biomolecular Computation in the US,” *New Generation Computing*, 20, pp. 217-236, 2002.
- [15] Ian Foster, “A Two Way Street to Science’s Future,” *Nature*, Vol. 440, 419. March, 2006.

- [16] Norman Chonacky, Dante Choi, "Science and Engineering Databases in an Open-Source Software World," *Computing in Science and Engineering*, May, pp. 10-14. 2003.
- [17] Wolfgang Emmerich, M. Andy Parker, José Blakeley, Clemens Szypersky, "Computational Science," *Towards 2020 Science*, Microsoft Research, pp. 14-15, 2006.
- [18] Mingwu Zhang, Daisuke Kihara, Sunil Prabhakar, "Tracing Lineage in Multi-version Scientific Databases," *IEEE Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering, (BIBE'07)*, pp. 440-447. 2007.
- [19] Stephen H. Muggleton, "Exceeding human limits," *Nature*, Vol 440, pp. 409-410, March, 2006.
- [20] M. Ivanova, N. Nes, R. Goncalves, M. Kersten, "MonetDB/SQL Meets SkyServer: the Challenges of a Scientific Database," *19th International Conference on Scientific and Statistical Database Management, (SSDBM'07)*, 2007.
- [21] Yang Liu, Zina Ben Miled, Omran Bukhres, Michael Bem, "Efficient Schema Design for a Pharmaceutical Data Repository," *Proceedings 13th IEEE Symposium on Computer-Based Medical Systems, (CBMS 2000)*, 2000.
- [22] Terence Critchlow, Krzysztof Fidelis, Madhavan Ganesh, Ron Musick, Tom Slezak, "DataFoundry: Information Management for Scientific Data," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 4, No. 1, March. 2000.
- [23] S. Abiteboul, R. Agrawal, P. Bernstein, "The Lowell Database Research Self-assessment," *Commun. ACM*, 48(5):pp. 111-118, 2005.
- [24] John L. Pfaltz, "What Constitutes a Scientific Database?," *19th International Conference on Scientific and Statistical Database Management, SSDBM'07*, 2007.
- [25] Ben Shneiderman, "Science 2.0," *Science*, Vol. 319, pp. 1349-1350, March, 2008.
- [26] Ben Shneiderman, "Web Science: A Provocative Invitation to Computer Science," *Communications of the ACM*, Vol. 50, No. 6, June, 2007.
- [27] Maria Klawe, Ben Shneiderman, "Crisis and Opportunity in Computer Science," *Communications of the ACM*, Vol. 48, No. 11, November, 2005.
- [28] Sthepan Philippi, "Light-weight Integration of Molecular Biological Databases," *Bioinformatics*, Vol. 20, no.1, pp. 51-57, 2004.
- [29] Cornelia B. Osmond, "The Universal Virus Database ICTVDB," *Computing in Science and Engineering*, pp. 16-25, May, 2003.
- [30] D. Wheeler, T. Barret, D. Benson, S. Bryan, "Database Resources of the National Center for Biotechnology Information," *Nucleic Acids Research*, Vol. 34, Database issue, D173-D180, 2006.
- [31] Bahram Parvin, Qing Yang Gerald Fontenay, Mary Helen Barcellos Hoff, "BioSig: An Imaging Bioinformatic System for Studying Phenomics," *IEEE Computer*, pp. 65-71, July, 2002.
- [32] T. Hubbard, D. Andrews, M. Caccamo, "Ensembl 2005," *Nucleic Acids Research*, Vol. 33, Database issue, D447-D453., 2005.
- [33] Vernor Vinge, "The Creativity Machine," *Nature*, Vol. 440, 411. March, 2006.
- [34] Lincoln Stein, "A DNA Sequence Class in Perl," *Dr. Dobbs Journal*, June, pp. 50-60, 1999.

## BIBLIOGRAFÍA

---

- [35] Andrés Rodríguez, Nicolás Guil, David M. Shotton, Oswaldo Trelles, “Automatic Analysis of the Content of Cell Biological Videos and Database Organization of Their Metadata Descriptors,” *IEEE Transactions on Multimedia*, Vol. 6, No. 1, February, 2004.
- [36] Jean Marc Pierson, Ludwig Seitz, Hector Duque, “Metadata for Efficient, Secure and Extensible Access to Data in a Medical Grid,” *Proceedings 15th International Workshop on Database and Expert Systems Applications*, pp. 562-566, 2004.
- [37] Wendy Chang, Deepak Murthy, Aidong Zhang, Tanveer F., Syeda Mahmood, “Global Integration of Visual Databases,” *Proceedings 14th International Conference on Data Engineering*, pp. 542-549, 1998.
- [38] W. Chang, G. Sheikholeslami, J. Wang, A. Zhang, “Data Resource Selection in Distributed Visual Information System,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 10, No. 6, pp. 926-929, Nov.-Dec., 1998.
- [39] Cees G. M. Snoek, Bouke Huurnink, Laura Hollink, Maarten de Rijke, Guus Schreiber, Marcel Worring, “Adding Semantics to Detectors for Video Retrieval,” *IEEE Transactions on Multimedia*, Vol. 9, No. 5, pp. 975-986, August. 2007.
- [40] Gholamhosein Sheikholeslami, Wendy Chang, and Aidong Zhang, “SemQuery: Semantic Clustering and Querying on Heterogeneous Features for Visual Data,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 5, September-October, 2002.
- [41] Daniel A. Keim, Hans Peter Kriegel, “VisDB: Database Exploration Using Multidimensional Visualization,” *IEEE Compute, Graphics and Applications*, Vol. 14, issue 5, pp. 40-49, 1994.
- [42] Janak Mulani, Arun Bahulkar, “A Graphical Navigator for Viewing Databases,” *Software and Practice*, Vol. 26(4), pp. 411-426, 1996.
- [43] Adriana. Hernández, Sergio V. Chapa, “Lida/Rec: Visual Language for Databases Interface PostgreSQL,” *2nd International Conference on Electrical and Electronics Engineering*, pp. 27-31, 2005.
- [44] Bogdan Czejdo, David Embley, Venugopal Reddy, “A Visual Query Language for an ER Data Model,” *IEEE Workshop on Visual Languages*, pp. 165-170, 1989.
- [45] Ahlberg, Christopher, Williamson, Shneiderman, “Dynamic queries for Information Exploration: An Implementation and Evaluation,” *Proc. ACM (CHI'92): Human Factors in Computing Systems*, pp. 619-626, 1992.
- [46] Ben Shneiderman, “Dynamic Queries for Visual Information Seeking,” *IEEE Software*, Vol. 11, No.6, pp. 70-77. 1994.
- [47] Gunjan Dang, Chris North, Ben Shneiderman, “Dynamic Queries and Brushing on Choropleth Maps,” *Proceedings Fifth International Conference on Information Visualization*, pp. 757-764, July, 2001.
- [48] J. Craig Venter, Mark D. Adams, Eugene W. Myers, Peter W Li, Richar J. Murai, “The Secuence of the Human Genome,” *Science*, Vol. 291, February, pp. 1304-1351, 2001.
- [49] Eric S. Lander, Lauren M. Linton, Bruce Birren, Chad Nusbaum, Michael C. Zody, “Initial Sequencing and Analysis of the Human Genome,” *Nature*, Vol. 409, February, pp. 860-921, 2001.

- [50] Dawynt P, Vancanneyt M, Janssens D. "Knowledge Management Systems for Microbial Information," *BCCM Newsletter*, Edition 19, Spring 2006.
- [51] Arek Kasprzyk, Damian Keefe, Damian Smedley, Darin London, "EnsMark: A Generic System for Fast and Flexible Access to Biological Data," *Genome Research*, 14:160-169, 2004.
- [52] J. L. Ashurst, C. K. Chen, "The Vertebrate Genome Annotation (Vega) Database," *Nucleic Acids Research*, Vol.33, Database issue, D-459-D465, 2005.
- [53] J. Lazar, K. Bessiere, I. Ceaparu, J. Robinson, B. Shneiderman, "Help! I'm Lost: User Frustration in Web Navigation," *IT and Society*, Vol. 1, issue 3, pp. 18-26, 2003.
- [54] Nigel Bevan, Miles Macleod, "Usability Measurement in Context," *Behaviour and Information Technology*, Vol. 13, pp. 132-145, 1994.
- [55] Ben Shneiderman, "Promoting Universal Usability with Multi-Layer Interface Design," *ACM, Conference on Universal Usability*, pp. 1-8, 2003.
- [56] Ben Shneiderman, "Universal Usability," *Communications of the ACM*, Vol. 43, No. 5, May, 2000.
- [57] Tiziana Catarci, Maria F. Costabile, "Visual Query System for Databases: A Survey," *Journal of Visual Languages and Computing*, Vol. 8, pp. 215-260, 1997.
- [58] Hong Yang Shiguang, Xinyi Li, Zengjin Qian, "Interpreting System for Visual Query Language CQL," *Proceedings of the Third International Conference on Image and Graphics (ICIG'04)*, pp. 564-567, 2004.
- [59] Lerina Aversano, Gerardo Canfora, Andrea De Lucia, "Understanding SQL through Iconic Interfaces," *Proceedings 26th Annual Computer Software and Applications*, pp. 703-708, 2002.
- [60] Ming Xu, Yong-Jun Wang, Yi jie Wang, "The Visual Query Language for An Object-Oriented Knowledge Based System," *International Symposium on Database Applications in Non Traditional Environments*, pp. 407-410, 1999.
- [61] Stavros Polyviou, George Samaras, Paraskevas Evripidou, "A Relationally Complete Visual Query Language for Heterogeneous Data Sources and Pervasive Querying," *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pp. 471-482, 2005.
- [62] Sudhir R. Kaushik, Elke A. Rundensteiner, "SEE: A Spatial Exploration Environment Based on a Direct Manipulation Paradigm," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 13, No. 4, July-August, 2001.
- [63] Stacie Hibino, Elke A. Rundensteiner, "Processing Incremental Multidimensional Range Queries in a Direct Manipulation Visual Query Environment," *Proceedings 14th International Conference on Data Engineering*, pp. 458-465, Feb. 1998.
- [64] Norman, Kent L., Zhao, H., Shneiderman, Ben and Golub, E., "Dynamic Query Choropleth Maps for Information Seeking and Decision Making," *Proceedings of the Tenth International Conference on Human-Computer Interaction*, pp. 1263-1267, 2003.
- [65] Qing Li, Chris North, "Empirical Comparison of Dynamic Query Sliders and Brushing Histograms," *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'03)*, 2003.
- [66] Evan Golub, Ben Shneiderman, "Dynamic Query Visualization on World Wide Web Clients: A DHTML Solution for Maps and Scattergrams," *International Journal of Web Engineering and Technology*, pp. 1-14, May, 2003.

## BIBLIOGRAFÍA

---

- [67] Nigel Bevan, "Usability is Quality of Use," *Proceeding of the 6th International Conference on Human Computer Interaction*, Anzai and Ogawa (eds), Elsevier, July, 1995.
- [68] Athula Ginige, San Murugesan. "Web Engineering: An Introduction," *IEEE Multimedia*, pp. 14-18, 2001.
- [69] Athula I. Vakali, Georgios I. Papadimitriou, "Web Engineering The Evolution of New Technologies," *Computing in Science and Engineering*, pp. 10-11, 2004.
- [70] Danny Ayers, "The Shortest Path to the Future Web," *IEEE Internet Computing*, pp. 76-79, Nov.-Dec., 2006.
- [71] Kwei Jay Lin, "Building Web 2.0," *IEEE Computer*, pp. 101-102, May, 2007.
- [72] San Murugesan, "Understanding Web 2.0," *IEEE IT Professional*, pp. 34-41, July-August, 2007.
- [73] Ora Lassila, James Hendler, "Embracing Web 3.0," *IEEE Internet Computing*, pp. 90-93, May-June, 2007.
- [74] Jim Hendler, "Web 3.0: Chicken Farms on the Semantic Web," *IEEE Computer*, pp. 106-108, January, 2008.
- [75] Timo Hannay, "Transforming Scientific Communication," *Towards 2020 Science*, Microsoft Research, pp. 18-20, 2006.
- [76] Susanne Boll, "MultiTube Where Multimedia and Web 2.0 Could Meet," *IEEE Multimedia*, Vol. 14, Issue 1, pp. 9-13, January-March, 2007.
- [77] Mark Greaves, "Semantic Web 2.0," *IEEE Intelligent System*, Vol 22, issue 2, pp. 94-96, March-April, 2007.
- [78] Ilya V. Yakovlev, "Web 2.0: Is It Evolutionary or Revolutionary," *IEEE IT Professional*, pp. 43-45, Nov-Dec., 2007.
- [79] George Lawton, "Web 2.0 Creates Security Challenges," *IEEE Computer*, pp.13-16, October, 2007.
- [80] Linda Dailey Paulson, "Building Rich Web Applications with Ajax," *Computer*, Vol. 38 issue 10, pp. 14-17, Oct. 2005.
- [81] J. Sergio Zepeda, Sergio V. Chapa, "From Desktop Applications Towards Ajax Web Applications," *4th International Conference on Electrical and Electronics Engineering*, pp. 193-196, 2007.
- [82] Alexander Mikroyannidis, "Toward a Social Semantic Web," *Computer*, pp. 113-115, November, 2007.
- [83] Danny Ayers, "From Here to There," *IEEE Internet Computing*, Volume 11, issue 1, pp. 85-89, Jan.-Feb., 2007.
- [84] James Hendler, "The Dark Side of the Semantic Web," *IEEE Intelligent System*, Vol. 22, issue 1, pp. 2-4, Jan.-Feb., 2007.
- [85] R. Benjamins, M. Greaves, J. Davies, J. Gómez, R. Baeza, J. Contreras, P. Mika, "Near-Term Prospects for Semantic Technologies," *IEEE Intelligent Systems*, pp. 76-88, January-February, 2008.
- [86] Tim Berners Lee, James Hendler, Ora Lassila, "The Semantic Web," *Scientific American Magazine*, May 17, 2001.
- [87] Martin Hepp, "Semantic Web and Semantic Web Services," *IEEE Internet Computing*, pp. 85-88, March-April, 2006.

- 
- [88] Gary Perlman, "Achieving Universal Usability by Designing for Change," *IEEE Internet Computing*, pp. 46-55, March-April, 2002.
- [89] Alessandro Bozzon, Sara Comai, "Conceptual Modeling and Code Generation for Rich Internet Applications," *ACM International Conference on Web Engineering*, pp. 363-360, 2006.
- [90] Mark Driver, Ray Valdes, Gene Phifer, "Management Update: Rich Internet Applications Are the Next Evolution of the Web," *Gartner Research*, Gartner Inc., May, pp. 1-5, 2005.
- [91] Horst Biller, Erich j. Neuhold, "Semantics of Databases: The Semantics of Data Models," *Information System*, Vol. 3, pp. 11-30, 1978.
- [92] A. Chan, S. Danberg, S. Fox, A. Nori, "Storage and Access Structures to Support a Semantic Data Model," *Proceedings of the 8th International Conference on Very Large Database*, pp. 122-130, 1982.
- [93] Nicholas Roussopoulos, John Mylopoulos, "Using Semantic Networks for Data Base Management," *Proceedings of the First International Conference on Very Large Databases*, pp.144,172, 1975.
- [94] Hyunmo Kang, Ben Shneiderman, "MediaFinder: An Interface for Dynamic Personal Media Management with Semantic Regions," *Conference on Human Factors in Computing Systems (CHI'03)*, pp. 764-765, 2003.
- [95] Peter Buneman, Serge Abiteboul, Alex Szalay, Andre Hagehülsman, "Semantics of Data," *Towards 2020 Science*, Microsoft Research, pp. 15-16, 2006.
- [96] Arnold W.M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, Ramesh Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1349-1380, Vol. 22, No. 12, December, 2000.
- [97] A. Axelrod, J. Golbeck, B. Shneiderman, "Generating and Querying Semantic Web Environments for Photo Libraries," *Human Computer Interaction Laboratory*, Universidad de Maryland, College Park, February, 2005.
- [98] Jhon R. Smith, Jelena Tesic, "Metadata Practices for Consumer Photos," *IEEE Multimedia*, pp. 86-92. July-September, 2005.
- [99] Norberto Moreno Quibén, "Comprensión y Producción: Semántica Lexica," *UCLM-CEU Talavera*, 2007.
- [100] Joseph D. Novak, Alberto J. Cañas, "The Theory Underlying Concept Maps and How to Construct and Use Them," *Technical Report IHMC Cmap Tools*, Institute for Human and Machine Cognition, pp. 1-32, 2008.
- [101] L. Murphy, "Digital Document Metadata in Organizations: Roles, Analytical Approaches, and Future Research Directions," *31th Annual Hawaii International Conference on System Sciences*, Volume 2, pp. 267-276, Jan. 1998.
- [102] S. Cammarata, I. Kameny, J. Lender, "A Metadata Management System to Support Data Interoperability, Reuse, and Sharing," *Journal of Database Management*, Vol. 5, No. 2, pp. 30-40, 1994.
- [103] Achim Steinacker, Amir Ghavam, Ralf Steinmetz, "Metadata Standards for Web-Based Resources," *IEEE Multimedia*, pp. 70-76, January-March, 2001.

## BIBLIOGRAFÍA

---

- [104] Oliver Günther, “*Environmental Information Systems*,” Springer Verlag GmbH and Co Publishers, Berlin, Germany, 1998.
- [105] Yu Deng, “The Metadata Architecture for Data Management in Web-based Choropleth Maps,” *Human Computer Interaction Laboratory*, Universidad de Maryland, College Park, pp. 1-27, February, 2002.
- [106] Marilyn McClelland, “Metadata Standards for Educational Resources,” *Computer*, pp. 107-109, November, 2003.
- [107] Zheng Yin, Zhengfang Xu, A. Saddik, “Study of metadata for advanced multimedia learning objects,” *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE03)*, Vol. 2, pp. 1099-1102, May, 2003.
- [108] Tiziana Catarci, Harald Kosch, Laszlo Boszormenyi, Mario Doller, Mulugeta Libsise, “The Life Cycle of Multimedia Metadata,” *IEEE Multimedia*, pp. 80-86, January-March, 2005.
- [109] P. Missier, P. Alper, O. Corcho, I. Dunlop, C. Goble, “Requirements and Services for Metadata Management,” *IEEE Internet Computing*, pp. 17-25, Sep-Oct, 2007.
- [110] Willian Ku, Mohan S. Kankanhalli, Hoo Hwee Lim, “A Collection Oriented Metadata Framework for Digital Images,” *IEEE International Conference on Multimedia*, pp. 761-764, July, 2006.
- [111] Atsutoshi Imai, Shuichi Yukita, “RDF Model and Relational Metadata,” *17th International Conference on Advanced Information Networking and Applications*, pp. 534-537, March, 2003.
- [112] Ruixin Yang, Menas Kafatos, X. Sean Wang, “Managing Scientific Metadata Using XML,” *IEEE Internet Computing*, pp. 52-59, July-August, 2002.
- [113] Michael L. Brodie, Frank Manola, “Database Management: A Survey,” *Fundamentals of Knowledge Base Management Systems*, Springer Verlag, New York, 1988.
- [114] A. Miguel Castaño, Mario G. Piattini, “*Fundamentos y Modelos de Bases de Datos*,” Alfaomega RA-MA editores, 2a. Edición, 1999.
- [115] Jim Conallen, “*Building Web Applications with UML*,” Addison Wesley Object Technology Series, Addison Wesley Longman Inc., March, 2000.
- [116] James Edwards, Cameron Adams, “*JavaScript Anthology*,” Sitepoint, First Edition, 2006.
- [117] Stuart Langridge, “*DHTML Utopia: Modern Web Design Using JavaScript and DOM*,” Sitepoint, First Edition, 2005.
- [118] D. Shafer, R. Andrew, “*HTML Utopia: Designing Without Tables Using CSS*,” Sitepoint, Second Edition, 2006.
- [119] Eric Meyer, “*More Eric Meyer on CSS*,” New Riders, First Edition, 2004.
- [120] Eric Meyer, “*Eric Meyer on CSS*,” New Riders, First Edition, 2003.
- [121] Keith Smith, “Simplifying Ajax Style Web Development,” *Computer*, Vol. 39, No. 5, pp. 98-102, May, 2006.
- [122] Lihui Lei, Zhenhua Duan, “Integrating Ajax and Web Services for Cooperative Image Editing,” *IT Professional*, pp. 25-29, May-June, 2007.
- [123] Eric J. Bruno, “Ajax: Asynchronous JavaScript and XML,” *Dr. Dobb’s Journal*, Vol. 31, No. 2, pp. 32-35, February, 2006.

- [124] Steven Douglas, “*Ajax on Java*,” O’Reilly, First Edition, 2007.
- [125] Frank W. Zammetti, “*Practical Ajax Projects with Java Technology*,” Apress, First Edition, 2006.
- [126] Bruce W. Perry, “*Ajax Hacks*,” O’Reilly, First Edition, 2006.
- [127] Vivek Chopra, Amit Bakore, Jon Eaves, Ben Galbraith, Sing Li, “*Professional Apache Tomcat 5, Programmer to Programmer*,” Wiley Publishing, First Edition, 2004.
- [128] J. Sergio Zepeda, Sergio V. Chapa, “Micro500, Un Sistema para búsqueda de Conocimiento en Microorganismos” *32nd Latin America Conference on Informatics (CLEI06)*, Santiago de Chile, 2006.
- [129] J. Sergio Zepeda, Sergio V. Chapa, “Diseño e Implementación del Sistema Micro500 basado en una Arquitectura con Extensión de Capacidades,” *9na. Conferencia de Ingeniería Eléctrica*, Cinvestav, México, 2003.
- [130] Rocío Quesada Castillo, “*Ejercicios para Elaborar Mapas Conceptuales*,” Editorial Limusa, Grupo Noriega Editores, 2008.
- [131] Cecilia M. Procopiuc, Divesh Srivastav, “Database Exploration Using Join Paths,” *International Conference on Data Engineering*, pp. 1331-1333, 2008.
- [132] Zhiguo Gong, Jingbai Zhang, Qian Liu, “Hidden Web Database Exploration,” *Proceedings of the Sixth International Conference on Intelligent System Design and Applications (ISDA06)*, pp. 838-843, 2006.
- [133] Ben Sheiderman, “Designing for Fun: How Can We Design User Interfaces to Be More Fun?,” *Interactions*, pp. 48-50, Sep-Oct, 2004.
- [134] Josep Schuller, “*Sams Teach Yourself UML in 24 Hours*,” Sam Teach Yourself, 3rd Edition, 2004.
- [135] Peter Dawyndt, Marc Vancanneyt, Hans De Meyer, Jean Swings, “Knowledge Accumulation and Resolution of Data Inconsistencies during the Integration of Microbial Information Sources,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 8, August, 2005.
- [136] John Worsley, Joshua Drake, “*Practical PostgreSQL*,” O’Reilly, First Edition, 2002.
- [137] J. Sergio Zepeda, “*Genómica y Computación*,” 2o. Diplomado en Investigación Genómica, Tesina, Universidad Autónoma de la Ciudad de México, 2006.
- [138] J. Mylopoulos, M. L. Brodie, “Readings in Artificial Intelligence and Databases,” *Morgan Kaufmann*, San Mateo, CA, 1988.
- [139] Michael Y. Galperin, “The Molecular Biology Database Collection: 2007 Update,” *Nucleic Acids Research*, Vol. 35, Database issue, D3-D4, December, 2006.
- [140] Stefan Agne, Christian Reuschling, Andreas Dengel, “DynaQ Dynamic Queries for Electronic Document Management,” *10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW’06)*, 2006.
- [141] Stephen G. Eick, “*Data Visualization Sliders*,” A&T Bell Laboratories, Naperville, 1993.
- [142] Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, “The Web Semantic Web: The Roles of XML and RDF,” *IEEE Internet Computing*, pp. 63-74, Sep.-Oct., 2000.

## BIBLIOGRAFÍA

---

- [143] Michel Klein, "XML, RDF, and Relatives," *IEEE Intelligent Systems*, pp. 26-28, 2001.
- [144] M. Kate Beard, Vyjayantu Sharma, "Multilevel and Graphical Views of Metadata," *IEEE Proceedings on Research and Technology Advances in Digital Libraries*, pp. 256-265, April, 1998.
- [145] Jeremy Allaire, "Macromedia Flash MX-A next-generation rich client," *Macromedia White Paper*, Macromedia Inc. March. 2002.
- [146] Andrea Simmons, "Rich Internet Applications 101: A Primer for Marketing Agencies and Multimedia Developers," *Integration New Media*, INM Inc., pp. 1-14, 2007.
- [147] Kevin Mullet, "The Essence of Effective Rich Internet Applications," *Macromedia White Paper*, Macromedia Inc., Nov, pp. 1-28, 2003.
- [148] Matias Urbieto, Gustavo Rossi, Jeronimo Ginzburg, "Designing the Interface of Rich Internet Applications," *Latin American Web Congress*, pp. 144-152, 2007.
- [149] Ron Rogowski, "Financial Institutions Need Rich Internet Applications," *Forrester Research, Inc.*, pp. 1-6, 2007.
- [150] Jose A. Lay, Ling Guan, "Retrieval For Color Artistry Concepts," *IEEE Transactions on Image Processing*, Vol. 13, No. 3, March, 2004.
- [151] Yingxu Wang, "On concept Algebra and Knowledge Representation," *5th IEEE International Conference on Cognitive Informatics*, pp. 320-330, 2006.
- [152] Huang Ying Hui, Li Guan Yu, Sheng Xing, "Concept Markup and Concept System Used to Construct Semantic Web," *4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-6, 2008.
- [153] Ana Arruarte, Urko Rueda, Jon A. Elorriaga, "Organizing The Learning Resources Related to the subject Introduction to Artificial Intelligence Through Concept Maps" *38th IEEE Frontiers in Education Conference*, F1C17-F1C22, 2008.
- [154] Debopriyo Roy, "Using Concept Maps for Information Conceptualization and Schematization in Technical Reading and Writing Courses: A Case Study for Computer Science Major in Japan," *IEEE International Professional Communication Conference*, pp. 1-12, July, 2008.
- [155] T. Hubbard, D. Barker, E. Birney, G. Cameron, Y. Chen, "The Ensembl Genome Database Project," *Nucleic Acids Research*, Vol 30, No. 1, 2002.
- [156] Evelina Lamma, Fabrizio Riguzzi, Sergio Storari, "Discovering Validation Rules from Microbiological Data," *New Generation Computing*, 21, pp 123-133, 2001.
- [157] Naren Ramakrishnan, Ananth Y. Grama, "Mining Scientific Data," *Advances in Computers by Academic Press*, Vol 55, pp. 119-163. 2001.
- [158] Dennis A. Benson, Ilene Karsch-Mizrachi, David J. Lipma, "GenBank," *Nucleic Acids Research*, Vol. 33, D34-D38, Database issue. 2005.