

03063
9



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

UNA APLICACIÓN WEB PARA LA OBTENCIÓN DE
INDICADORES BIBLIOMÉTRICOS EN CIENCIA Y
TECNOLOGÍA

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRA EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:

MA. DE JESÚS MADERA JARAMILLO

DIRECTORA DE TESIS: DRA. AMPARO LÓPEZ GAONA

MÉXICO, D.F.

2003



A



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

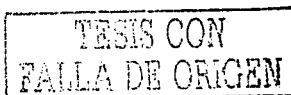
DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Índice general

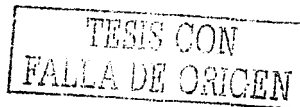
Agradecimientos	5
Introducción	7
1. Tecnología de la Web	11
1.1. De los mainframes a Internet	11
1.2. Sistemas para acceder la información de Internet	13
1.2.1. Archie	13
1.2.2. Gopher	13
1.2.3. WAIS	14
1.2.4. La World Wide Web	14
1.3. El protocolo de comunicación HTTP	17
1.4. Publicación de documentos estáticos	18
1.5. Publicación de documentos dinámicos	19
1.6. Aplicaciones Web y sus herramientas de desarrollo	20
1.6.1. CGIs	20
1.6.2. ASP	22
1.6.3. JSP	22
1.6.4. Applets	22
1.6.5. Servlets	23
1.6.6. JavaScript	23



1.6.7. JavaBeans	23
1.6.8. ActiveX	23
1.7. Seguridad en las aplicaciones Web	24
1.8. Arquitectura de las aplicaciones Web	25
1.8.1. Cliente Web débil (Thin Web client)	25
1.8.2. Cliente Web robusto (thick Web client)	28
1.8.3. La Web como mecanismo de entrega (Web delivery)	32
2. Indicadores de ciencia y tecnología	35
2.1. Antecedentes del análisis del desarrollo científico	35
2.2. Generación del conocimiento científico	37
2.3. Importancia de los indicadores bibliométricos	39
2.4. Indicadores de ciencia y tecnología	40
2.4.1. Indicadores de calidad científica	40
2.4.2. Indicadores de producción científica	41
2.4.3. Indicadores del impacto de los trabajos	42
2.4.4. Conexión entre trabajos y autores científicos	42
2.4.5. Indicadores de impacto de las fuentes	42
2.5. Fuentes de información (bases de datos bibliográficas)	44
2.6. Estandarización de los datos	45
2.7. Los Indicadores de Ciencia y Tecnología en la World Wide Web	46
2.8. Selección de fuentes de datos	47
3. UML y las aplicaciones Web	49
3.1. Extensión UML para aplicaciones Web	49
3.2. Requerimientos del software	56
3.3. Análisis	57
3.3.1. Modelo de objetos	58
3.3.2. Diccionario de clases	60

TESIS CO
FALLA DE ORIGEN

3.4. Diseño	60
3.4.1. Diagramas de secuencia para el diseño	62
3.4.2. Páginas servidor, enlaces, redireccionamientos y formas	63
3.4.3. Frames	66
3.4.4. Diseño de la base de datos	66
4. Herramienta para obtención de indicadores bibliométricos	69
4.1. Requerimientos funcionales	69
4.2. Requerimientos no funcionales	92
4.3. Modelo de objetos	93
4.4. Diccionario de clases	93
4.5. Diseño	98
5. Resultados, Conclusiones y trabajos futuros	109
5.1. Resultados	109
5.2. Conclusiones	111
5.3. Trabajos futuros	112
Bibliografía	113
A. Interfaz de Usuario	119



TESIS CON
FALLA DE ORIGEN

Agradecimientos

Al *International Centre for Theoretical Physics (ICTP)* por la beca otorgada para el desarrollo del sistema SIESTA.

A Amparo López Gaona por escucharme y ayudarme a poner en orden las ideas; además por la paciencia que tuvo en mis tropiezos.

A Jane Russell por proporcionarme gran parte del material utilizado en el tema de indicadores de ciencia y tecnología, y por aclararme las dudas en este tópico, nuevo para mí.

A mi familia por creer en mí.

A mis amigos y mis hermanas Bertha y Ana por presionarme para que llegara a la meta.

TESIS CON
FALLA DE ORIGEN

SECRET
SALA DE GIUSTIZIA

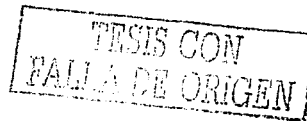
Introducción

En la actualidad la mayoría de las empresas e instituciones académicas están haciendo uso del avance tecnológico para difundir información en forma rápida, fácil y a bajo costo. Esto se debe a que la *Internet* proporciona los mecanismos que apoyan el proceso de difusión.

A pesar de que *Internet* nació en la década de los 60s, no fue sino hasta los 90s que se hizo popular. Dicha popularidad se debe al surgimiento de la *World Wide Web* (WWW o Web), la cual fue creada, por Tim Berners-Lee del CERN (*Conseil European pour la Recherche Nucleaire*), originalmente para intercambiar información entre investigadores del campo de la física de gran energía y en la segunda mitad de la década de los 90s su uso creció en forma acelerada debido a que las empresas e instituciones académicas hacían uso de ella para difundir información y realizar transacciones.

El uso de la *Web* no solamente cambió la forma en que los individuos intercambiaban información, también tuvo un impacto en el desarrollo de sistemas. Debido a los requerimientos de las empresas para realizar transacciones a través de la *Web* se tuvo la necesidad de desarrollar aplicaciones que cubrieran dichos requerimientos, en este punto es donde se empieza a hablar de aplicaciones *Web*.

Las aplicaciones *Web* son aquellas que utilizan tecnologías que permiten que el contenido de las páginas *Web* sea dinámico y que los usuarios puedan realizar transacciones a través de ella, es decir, que sea posible cambiar el estado del servidor. Es importante mencionar que aquellas aplicaciones que solamente permiten realizar consultas no son consideradas aplicaciones *Web* ya que en ningún momento cambian el estado del servidor.



Una de las principales ventajas en el desarrollo de aplicaciones *Web* radica en el hecho de que éstas permiten que la información difundida a través de ella se actualice en forma constante y pueda ser utilizada por una gran cantidad de individuos. A finales de la década de los 90s este tipo de aplicaciones atrajo la atención de los encargados de analizar y evaluar la producción científica. La evaluación de la producción científica se hace a través de indicadores bibliométricos, ya que éstos proporcionan información que permite conocer los avances e impacto de la ciencia. Es importante que éstos estén al alcance de los empresarios y del gobierno para que los recursos económicos sean asignados en el desarrollo de la ciencia de una forma más adecuada.

La aparición de las aplicaciones *Web* también tuvo impacto en uno de los lenguajes utilizados para el modelado de sistemas. Debido a que este tipo de aplicaciones presentan nuevos elementos, a uno de los lenguajes más utilizados en la actualidad para realizar el modelado de sistemas (UML) se le hizo una extensión para que facilitara dicho proceso. La extensión propuesta por Conallen [11], permite modelar elementos presentes en las aplicaciones *Web* que no se encuentran en las aplicaciones tradicionales. Las aplicaciones *Web* pueden ser modeladas sin utilizar la extensión antes mencionada, pero los diagramas obtenidos no son tan claros.

En la presente tesis, se muestra el análisis y diseño de la aplicación *Web* denominada *Software Interface for the Evaluation of Scientific Technological Activity (SIESTA)*. Esta aplicación surgió de la necesidad de un grupo italiano que se encontraba realizando un estudio sobre la producción científica de dicho país [4]. Para realizar tal estudio, se tomaban los datos de la bases de datos *Current Contents (CC)*, la cual es una base de datos producida por el *Institute for Scientific Information (ISI)*, disponible en siete ediciones multidisciplinarias (*Life sciences, Agriculture, Biology & Environmental sciences, Engineering, Computing & Technology, Physical, Chemical*

& Earth Sciences, Social & Behavioral Sciences, Arts & Humanities, y Clinical Medicine). Cada edición cubre un grupo de disciplinas interrelacionadas, diseñadas para proveer la más amplia cobertura de revistas. Los datos extraídos de la base de datos CC no están estandarizados. El grupo antes mencionado realizaba la estandarización de los datos en forma "manual", es decir, utilizando el procesador de palabras *Word* y la hoja de cálculo *Excel*. De aquí la necesidad de contar con una herramienta que permitiera realizar dicha estandarización y el cálculo de indicadores por número y distribución de las publicaciones en forma automática.

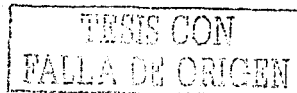
El objetivo de esta tesis es: utilizando la extensión UML, modelar una aplicación *Web* que sirva como herramienta de apoyo para la obtención de indicadores bibliométricos en ciencia y tecnología en términos de disciplina, área geográfica, y organizaciones.

Para cumplir con tal propósito la tesis se estructura de la siguiente manera:

En el capítulo uno, se menciona cómo ha evolucionado la forma de difundir la información a través de la *Web* hasta llegar al desarrollo de aplicaciones *Web*, así como la arquitectura de tales aplicaciones.

En el capítulo dos, se proporciona información que permitirá conocer cuáles son los indicadores más comunes en la evaluación de la productividad científica, así como los problemas que deben ser resueltos para que los datos extraídos de bases de datos bibliográficas puedan ser utilizados para la obtención de indicadores de ciencia y tecnología.

En el capítulo tres, se muestran la extensión UML, así como la forma en que ésta se utiliza en la fase de diseño para modelar aplicaciones *Web* con arquitectura cliente *Web* débil.



En el capítulo cuatro, se presenta el modelado de la aplicación *Web SIESTA*, que sirve de apoyo en la obtención de indicadores de ciencia y tecnología de Italia. El sistema se utiliza para la obtenciones de indicadores italianos, ya que fue patrocinado por el *International Centre for Theoretical Physics (ICTP)* .

En el capítulo cinco, se presenta los resultados de la aplicación *Web SIESTA*, las conclusiones y los trabajos futuros. En el apéndice A se muestra algunas de las pantallas de la interfaz gráfica de usuario del presente sistema.

COPIA CON
EL ORIGEN

Capítulo 1

Tecnología de la Web

El uso de la *World Wide Web* (Web o WWW) ha crecido en forma acelerada, entre otras cosas, debido a que las transacciones a través de *Internet* se han convertido en una parte esencial de las empresas. Este crecimiento ha conllevado una evolución de la forma de hacer uso de la *Web* para presentar información o realizar transacciones a través de ella. En el presente capítulo se mencionará la evolución que han tenido las aplicaciones para la *Web*, así como las herramientas que se utilizan para el desarrollo de las mismas.

1.1. De los mainframes a Internet

En la actualidad la mayoría de las empresas e instituciones académicas están haciendo uso del avance tecnológico para difundir información en forma rápida, fácil y a bajo costo. La utilización de las comunicaciones automatizadas en estas instituciones han pasado, en poco menos de medio siglo, por diversas etapas.

Los primeros sistemas informáticos, en la década de los 60s, se basaban en computadoras centrales, *mainframes*, que proporcionaban soporte a una red de terminales locales o remotas. Fue la era de los departamentos de proceso de datos centralizados.



En los inicios de la década de los 80s surgieron dos nuevos conceptos que intentaban eliminar el control y centralización de los *mainframes*: la informática distribuida y la aparición y despliegue de las computadoras personales o PCs. Algunas de las aplicaciones de *mainframes* se trasladaron a las computadoras personales, en un intento de mejorar la flexibilidad, la eficiencia y la productividad de los usuarios. Después de esto se vio la necesidad de comunicar las PCs mediante redes para desarrollar funciones como el intercambio de mensajes, aplicaciones y recursos compartidos, etcétera. Estas redes, de alta velocidad y bajo costo, se denominan redes de área local o LANs.

A pesar de todo, este tipo de redes no han sustituido, ni mucho menos, a los *mainframes*. La elección entre ambas ha de considerarse desde la perspectiva que cada una de las instituciones tiene de sí misma, y de sus necesidades de información y comunicación. La mayor ventaja que proporciona el modelo de *mainframe*, la centralización de los procesos, se ha conseguido con la integración de las aplicaciones basadas en redes locales (LANs) en un entorno corporativo mediante las redes de área amplia (WANs) que proporcionan acceso a los datos del sistema central.

Además de todos estos desarrollos de comunicación *dentro* de las instituciones, y en conjunción con el enfoque sistemático que gobierna el mercado actual, se han desarrollado redes a nivel más global, siendo su máximo exponente lo que conocemos como *Internet*. La expansión de esta "red de redes" ha traído grandes ventajas a las instituciones privadas y académicas. Una de estas ventajas es que la información disponible en *Internet* puede ser compartida con un gran número de individuos y esto genera nuevas experiencias, conocimiento, y nuevas formas de colaboración.

TESIS CON
FALLA DE ORIGEN

1.2. Sistemas para acceder la información de Internet

En *Internet* existen diferentes sistemas que hacen posible el acceso e intercambio de la información que ésta contiene. Estos sistemas son: *Archie*, *Gopher*, *Veronica*, *WAIS*, y la *WWW*.

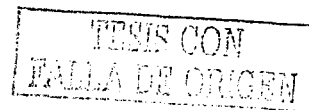
1.2.1. Archie

Archie fue desarrollado en 1989 por Peter Deutsch de la Universidad McGill de Montreal. *Archie* es una aplicación del protocolo *Prospero*¹, que organiza la información de servidores FTP en base a índices, es decir, se crea un índice que contiene los nombre de archivos y la dirección del servidor en el que se encuentran localizados. Así, este índice se instala en un servidor (Servidor *Archie*) que puede ser consultado por todos los servidores *Archie*. Este tipo de servidores pueden ser consultados a través de correo electrónico, sesión remota (vía *telnet*), y clientes *Archie*, pero el usuario debe conocer de antemano el nombre del archivo.

1.2.2. Gopher

Gopher fue desarrollado en 1991, por un equipo dirigido por Mark McCachill de la Universidad de Minnessota, para intercambiar documentos de texto y mensajes entre estudiantes y personal de la universidad. Este sistema no solo permite realizar búsquedas por nombre de archivo, sino también por su contenido. *Gopher* está basado en el concepto de menús jerárquicos, los cuales tienen una estructura semejante a la organización de los directorios y archivos de las computadoras. La maneras de consultar servidores *Gopher* son a través de entorno de texto, Windows o Xwindows, programas cliente de *Gopher* o con navegadores (como por ejemplo Netscape). Los

¹*Prospero* es un protocolo para la presentación de "directorios virtuales". Los directorios contienen archivos y otros directorios, los cuales pueden estar localizados en diferentes computadoras.



servidores *Gopher* se accesan a través de su Localizador Uniforme de Recursos (URL, *Uniform Resource Locator*)². Una vez que se establece la conexión con el servidor, se muestra un menú en el cual el usuario puede ir adentrándose en submenús que proporcionarían información más específica.

Una herramienta utilizada para facilitar la búsqueda en los servidores *Gopher* es *Veronica* (*Very Ease Rodent-Oriented Net-wide Index to Computerized Archives*). Esta herramienta, similar a *Archie*, genera índices con los contenidos de los menús de los servidores *Gopher* y es posible accederla a través de ellos.

1.2.3. WAIS

WAIS es un sistema electrónico de indexación, publicación y recuperación de información multimedia, que puede trabajar localmente o en una red. Un servidor *WAIS* procesa las consultas, consistentes en una lista de palabras clave, realiza una búsqueda en los documentos indexados y regresa una lista de aquellos que contienen las palabras claves especificadas. Al igual que *Gopher*, las búsquedas en *WAIS* se pueden realizar en entorno texto, Windows o Xwindows, programas cliente *WAIS* o mediante WWW.

1.2.4. La World Wide Web

No obstante que existían los sistemas antes mencionados, la *Internet* se hizo popular debido al surgimiento de la *World Wide Web*. La *Web* fue creada, en 1989, por Tim Berners-Lee del CERN (*Conseil European pour la Recherche Nucleaire*) y el primer uso público fue en 1992, en Ginebra. La *Web* fue creada originalmente para intercambiar información entre investigadores del campo de la física de gran energía y en la segunda mitad de la década de los 90s su uso creció en forma acelerada debido a que

²URL es la dirección electrónica de un recurso, un archivo o un programa.

las empresas e instituciones académicas hacían uso de ella para difundir información y realizar transacciones.

La *Web* es un sistema que permite el acceso a documentos multimedia enlazados entre sí y que residen en servidores situados por todo el mundo. Está disponible para todo individuo que tenga acceso a una computadora y a una red con acceso a *Internet*, desde cualquier plataforma, y las posibilidades de búsqueda son prácticamente ilimitadas. En la *Web* es posible encontrar herramientas de software libre, éstas ayudan en el desarrollo de aplicaciones *Web* y de esta manera es posible que las empresas e instituciones académicas puedan difundir e intercambiar información a bajo costo.

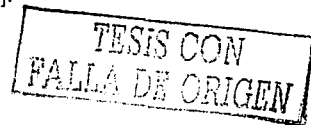
La capacidad multimedia significa que los documentos de la *Web*, archivos llamados páginas *Web*, puedan contener enlaces con texto, imágenes, sonido y vídeo, que están contenidos en otros archivos en el mismo o distinto servidor. En la *Web*, para visualizar los documentos *Web* se utilizan lenguajes de marcado³ como HTML, XML, SMIL y MathML, entre otros.

Debido a la capacidad multimedia de su información, de su facilidad de uso y navegación entre documentos de cualquier servidor, de su bajo costo, de la gran variedad de plataformas desde las que se puede acceder, y de su disponibilidad, la *Web* ha tenido un gran éxito, que lo lleva a ser el servicio más utilizado de *Internet*.

El lenguaje de marcado de hipertexto HTML

El primer lenguaje de marcado que se utilizó para expresar y visualizar las páginas *Web* fue el Lenguaje de Marcado de Hipertexto (HTML, *HiperText Markup Language*). HTML es un lenguaje de etiquetas basado en el Lenguaje Estándar Generalizado

³Los lenguajes de marcado comprenden un conjunto de reglas que definen todo aquello que es parte de un documento digital, pero que no pertenece al texto mismo [39].



de Marcado (SGML, *Standard Generalized Markup Language*)⁴. Este lenguaje es una aplicación específica de SGML que contiene etiquetas que definen cómo debe ser formateado un documento de texto en la pantalla.

EL lenguaje de marcado extensible XML

El Lenguaje de Marcado Extensible (XML, *Extensible Markup Language*) es un lenguaje para formatear texto derivado de SGML. Originalmente fue diseñado para el tratamiento de publicaciones electrónicas, pero en la actualidad se utiliza para el intercambio de una gran variedad de datos a través de la *Web*. XML es utilizado para crear especificaciones o modelos de documentos o datos con el fin de poder estandarizar el intercambio de datos.

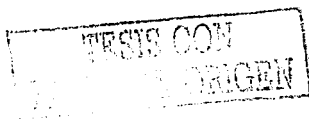
El lenguaje de marcado matemático MathML

El Lenguaje de Marcado Matemático (MathML, *Mathematical Markup Language*) es un lenguaje para describir expresiones matemáticas, éste es un derivado de XML. El objetivo de MathML es que las matemáticas sean procesadas en la *Web* como un texto HTML. Debido a su complejidad, no está previsto que este lenguaje sea utilizado directamente por los usuarios, para ello se pueden utilizar editores de ecuaciones o programas de conversión que generen código de MathML [44].

El lenguaje de integración multimedia sincronizado SMIL

EL Lenguaje de Integración Multimedia Sincronizado (SMIL, *Synchronized Multimedia Integration Language*) está basado en XML y permite la integración de objetos multimedia independientes, dentro de una presentación multimedia sincronizada en la *Web*. Con SMIL se puede describir el comportamiento temporal de la presentación, la distribución de los elementos en la pantalla y crear hiperenlaces con objetos multimedia.

⁴SGML es un metalenguaje con el cual se pueden definir lenguajes de marcado.



El lenguaje para modelar realidad virtual

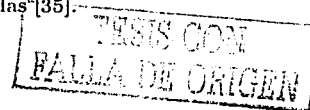
VRML (*Virtual Reality Modeling Language*) es un lenguaje basado en texto genérico diseñado para construir imágenes en tercera dimensión. Se utiliza principalmente para construir imágenes tridimensionales, no obstante, puede ser utilizado para crear texto en tercera dimensión.

1.3. El protocolo de comunicación HTTP

Un elemento muy importante en la *Web* es el Protocolo de Transferencia de Hipertexto (HTTP, *Hyper Text Transfer Protocol*), en éste se definen todas las interacciones entre un navegador cliente y un servidor. HTTP corre sobre el protocolo de comunicación TCP/IP (*Transport Control Protocol/Internet Protocol*)⁵. Una transacción HTTP consta de cuatro pasos:

1. **Establecimiento de la conexión.** Para intercambiar información entre un cliente y un servidor, primero se debe establecer una conexión TCP/IP a través del puerto 80. Este es el puerto que se utiliza normalmente, pero es posible utilizar otros puertos que no estén reservados para otros servicios.
2. **Petición.** El cliente envía un mensaje de petición hacia el servidor.
3. **Respuesta.** Después de que el servidor recibe e interpreta la petición hecha por el cliente, envía una respuesta.
4. **Final de la conexión.** Después de haber cumplido con la solicitud hecha por el cliente, el servidor tiene la responsabilidad de cerrar la conexión TCP/IP. El cliente también puede cerrar la conexión mediante el botón "Detener" del navegador.

⁵TCP/IP son un conjunto de protocolos que han conformado un estándar para *Internet* propuesto por el Departamento de Defensa de Estados Unidos. El conjunto TCP/IP permite que la comunicación ocurra entre redes heterogéneas y/o similares interconectadas^[35].



HTTP es el protocolo de comunicación más comúnmente utilizado entre un navegador cliente y un servidor, pero también pueden ser utilizados otros protocolos como el HTTPS o HTTP con *Secure Socket Layer* y el WebDAV. HTTPS es un protocolo que permite manejar confidencialidad y autenticación en *Internet*. Web-DAV (*Web-based Distributed Authoring and Versioning*) es un conjunto de extensiones del protocolo HTTP. Las áreas de aplicación de éste son las ediciones en colaboración y manipulación de archivos en servidores *Web* remotos.

1.4. Publicación de documentos estáticos

Cuando se empezó a utilizar el servicio *Web* como una herramienta para difundir información, su contenido consistía de páginas *Web* estáticas escritas usando un editor capaz de aplicar las etiquetas HTML que controla el formato de presentación de los documentos HTML, incluyendo enlaces a otros documentos o a otros servidores *Web*.

Para mostrar este tipo de documentos, lo único que se necesita es el navegador (cliente), *Intranet*⁶ o *Internet*, y el servidor *Web*. La información que proporciona el servidor *Web* ya se encuentra almacenada en páginas HTML estáticas. El proceso para mostrar una página HTML estática es: 1) El navegador (cliente) hace una petición de un documento *Web* al servidor. En este paso se establece una conexión HTTP vía TCP/IP, entre el navegador (cliente) y el servidor *Web*, 2) El servidor *Web* busca la página HTML en el sistema local de archivo, 3) El sistema local de archivos regresa al servidor el archivo solicitado, 4) El servidor envía, al navegador (cliente) la página HTML solicitada y cierra la conexión. La figura 1.1 muestra el proceso de solicitud y envío de información.

⁶Una *Intranet* es una red privada que usa estándares y protocolos de *Internet* para que los miembros de una organización se comuniquen y colaboren más eficientemente entre ellos.

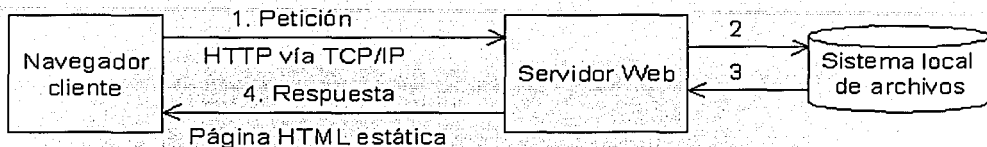


Figura 1.1: Transmisión de documentos estáticos

1.5. Publicación de documentos dinámicos

El surgimiento de las etiquetas de formateo HTML que hace posible el uso de *formas de entrada*⁷ en páginas *Web* permitió que se integraran las bases de datos y la *Web*. Con esta integración los documentos *Web* son creados en forma dinámica.

Para la generación de páginas dinámicas el proceso es el siguiente: 1) Este paso es similar al paso 1) del proceso de páginas estáticas, pero el navegador (cliente) además envía datos a través de una forma HTML, 2) El servidor *Web* invoca un proceso y le pasa los datos de la forma, 3) Se realiza el procesamiento de los datos, si es necesario se hace una conexión a la base de datos, 4) Se genera una página HTML dinámicamente, 4) La página es enviada al servidor *Web* y finalmente 5) El servidor envía esta página al navegador cliente. La figura 1.2 muestra el proceso mencionado.

En la actualidad, la tendencia para difundir información es la integración de la *Web* y las bases de datos. Esta tendencia se debe a que las bases de datos son una de las herramienta que ayudan en el proceso de generación de páginas *Web* dinámicas y el flujo de información es bidireccional. Otra de las ventajas que trajo consigo esta integración fue el desarrollo de aplicaciones *Web*.

⁷Una forma de entrada es una parte de una página *Web* que puede aceptar entradas por parte de los usuarios. Es una colección de campos que permiten que los usuarios introduzcan texto o realicen selecciones de una lista.

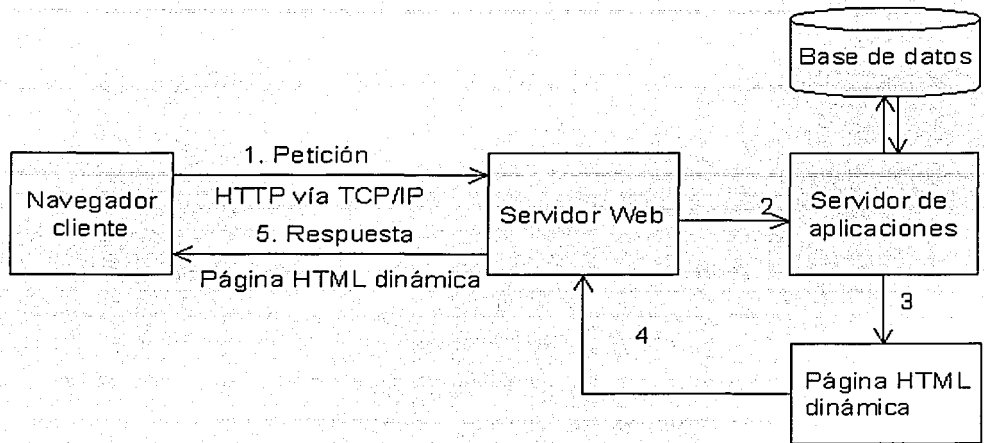


Figura 1.2: Trasmisión de documentos dinámicos

1.6. Aplicaciones Web y sus herramientas de desarrollo

Una aplicación *Web* es un sistema accesible a través de la *Web* que permite que los usuarios puedan realizar transacciones. Existen herramientas que proveen la infraestructura necesaria para el desarrollo de este tipo de aplicaciones, tales herramientas son el punto de unión entre la *Web* y las bases de datos.

1.6.1. CGIs

La primera herramienta usada para este fin fueron los CGIs (*Common Gateway Interface*). Un CGI es un protocolo que especifica la manera en cómo los *scripts* escritos por usuarios, que son ejecutados en el servidor *Web*, pueden comunicarse con el navegador (cliente).

Los *scripts* CGI son pequeños programas que siguen el protocolo CGI, éstos actúan

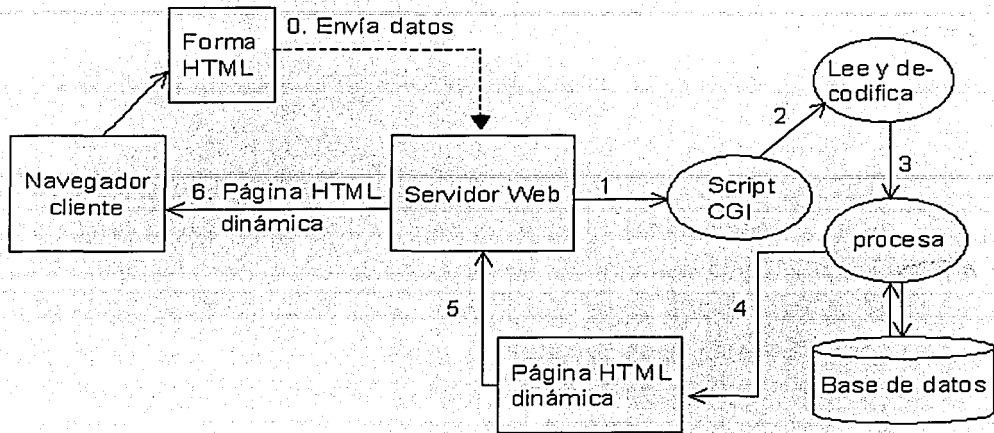


Figura 1.3: Funcionamiento de los *scripts* CGI

como interfaz entre un usuario *Web* (cliente) y el servidor *Web*, estos programas son ejecutados por el servidor. Los *scripts* CGI aparecen como enlaces dentro de una página HTML y pueden recibir datos de entrada, tomados de las formas HTML, los resultados pueden ser texto y/o gráficos formateados como documentos HTML, aunque algunas veces en lugar de generar páginas HTML son utilizados para hacer llamadas a otros *scripts* CGI encargados de generar las páginas. Perl (*Practical Extraction and Report Language*) es uno de los lenguajes más usados para desarrollar *scripts* CGI ya que tiene elementos que facilitan la manipulación de texto. La forma de operar de un *script* CGI es la siguiente: 1) el *script* es invocado por el servidor *Web*, 2) lee y decodifica las entradas de los usuarios, las cuales fueron enviadas típicamente a través de formas HTML (0), 3) realiza el procesamiento y si es necesario hace una conexión a una base de datos, 4) genera dinámicamente una página HTML, 5,6) la página HTML es enviada al navegador (cliente) a través del servidor *Web*. La figura 1.3 muestra el proceso antes mencionado.

Una de las desventajas de los CGI es que cada vez que un usuario realiza una llamada a un programa el servidor lo ejecuta y crea un proceso para cada uno de ellos, esto ocasiona que el servidor *Web* se sobrecargue de trabajo y el desempeño no sea óptimo. Algunas herramientas que fueron implementadas para solucionar este problema son módulos compilados que se ejecutan por el servidor y tienen acceso a APIs (*Application Programming Interfaces*) que proveen la información enviada por el usuario; los APIs más comúnmente usados son los implementados por Microsoft (ISAPI) y por Netscape (NSAPI) y los servlets de Java. Estas herramientas tienen mecanismos que manejan hilos de control múltiples. Otras herramientas utilizadas para desarrollar aplicaciones Web son los applets de Java, JavaBeans, ActiveX, JavaScript.

1.6.2. ASP

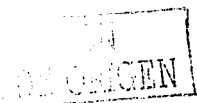
Active Server Pages (ASP) es tecnología de Microsoft para crear páginas *Web* con contenido dinámico. Los *scripts* ASP, al igual que los *scripts* escritos en Perl, son ejecutados del lado del servidor. Éstos pueden ser escritos en los lenguajes VBScript o JavaScript, pero el más utilizado es el primero. VBScript es un subconjunto del lenguaje Visual Basic.

1.6.3. JSP

Java Server Pages (JSP) es una tecnología basada en Java que facilita el desarrollo de páginas *Web* dinámicas. En los JSP se incluyen etiquetas HMTL y código Java que permite el acceso a datos que se encuentran en el servidor. JSP también es ejecutado en el servidor.

1.6.4. Applets

Los applets son aplicaciones Java que se ejecutan en el navegador (cliente), éstos son invocados mediante documentos HTML. Los applets se encuentran en el servidor,



pero cuando son invocados por un navegador (cliente) se descargan y ejecutan en éste mostrando una interfaz gráfica al usuario.

1.6.5. Servlets

Los servlets son tecnología basada en Java, son componentes que se ejecutan en el servidor. A diferencia de los applets, éstos no despliegan una interfaz gráfica al usuario, los resultados son mostrados usualmente como un documento HTML.

1.6.6. JavaScript

JavaScript es un lenguaje para escribir *scripts* que son ejecutados en el navegador (cliente). Con los *scripts* escritos en este lenguaje se realiza la verificación de los datos proporcionados por el usuario, en una forma de entrada HTML, en el mismo navegador (cliente) evitando así la sobrecarga de procesamiento en el servidor.

1.6.7. JavaBeans

Un *bean* es un contenedor que puede almacenar y obtener cierto tipo de información. Cada pieza de información es conocida como una propiedad y puede ser solicitada o extraída individualmente. Los JavaBeans son el equivalente a los controles ActiveX de Microsoft.

1.6.8. ActiveX

ActiveX, al igual que ASP, es una tecnología Microsoft para el desarrollo de páginas *Web* dinámicas. ActiveX puede ser ejecutado tanto en el navegador (cliente) como en el servidor. Un elemento importante en ActiveX son los llamados controles ActiveX. Éstos son programas o módulos que se pueden incrustar en documentos HTML. El código de tales controles es descargado en el navegador (cliente) y tiene acceso ilimitado a los recursos de éste.

Las herramientas utilizadas para el desarrollo de las aplicaciones *Web* dependerán del sistema que se desee desarrollar, de los recursos económicos, y del equipo de trabajo.

1.7. Seguridad en las aplicaciones *Web*

Una parte importante en las aplicaciones *Web* es la referente a la seguridad en el acceso a la información. Existen varios tipos de seguridad los cuales dependen de la arquitectura de la aplicación *Web*.

- Autenticación del usuario. Este tipo de seguridad lo proporciona el servidor *Web*. En la autenticación de usuario, el servidor *Web* solicita el nombre del usuario y su clave de acceso. El proceso de autenticación debe verificar que los datos proporcionados por el usuario sean correctos, en caso de que el resultado sea exitoso, el usuario podrá tener acceso a las bases de datos o a las páginas *Web*.
- Control de navegación. El control de navegación en las páginas *Web* lo proporciona la aplicación *Web*. En este tipo de seguridad se debe implementar un proceso que controle las rutas y la secuencia en la que el usuario puede visualizar la información.
- Autorización de usuario. Este tipo de seguridad también lo proporciona la aplicación *Web*. Una vez que el proceso de autenticación (si es que existe) verificó que el usuario es quien dice ser, el proceso de autorización debe verificar qué páginas, este usuario, está autorizado a visitar.
- Autorización de acceso a la base de datos. La autorización de acceso a la base de datos la proporciona el manejador de la base de datos. Este tipo de autorización se refiere a los privilegios de acceso a la base de datos, como por ejemplo insertar, eliminar, modificar o consultar los datos.

1.8. Arquitectura de las aplicaciones Web

En las aplicaciones *Web* la actividades involucradas en las etapas de obtención de requerimientos y el análisis se realizan en forma similar que las aplicaciones tradicionales. En este tipo de aplicaciones el diseño cambia ya que se involucran nuevos elementos, Conallen [11] menciona tres patrones diferentes en el diseño de aplicaciones *Web*, a estos patrones los define en base a los componentes más significativos del sistema. Los tres patrones antes mencionados son: aplicaciones con cliente *Web* débil, aplicaciones con cliente *Web* robusto, y aplicaciones en las cuales la *Web* actúa como mecanismo de entrega (*Web delivery*).

1.8.1. Cliente Web débil (Thin Web client)

Estas aplicaciones son las más comúnmente usadas, son llamadas de este modo ya que la configuración requerida por el cliente es mínima. En este tipo de aplicaciones el cliente únicamente necesita un navegador *Web* en el que se pueda trabajar con formas ya que las transacciones son ejecutadas por el servidor. Este patrón es apropiado para aplicaciones *Web* en los cuales el cliente no tiene control sobre su configuración, es muy usada en aplicaciones de comercio electrónico.

Estructura

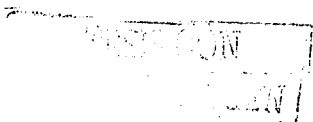
En este patrón los componentes se encuentran en el servidor, y estos son los siguientes:

- Navegador cliente. Debe ser cualquier navegador que cumpla con los estándares de HTML y que soporte formas. El usuario solamente puede solicitar páginas HTML o servidor, y aceptar y regresar *cookies*⁸; el servidor le regresa páginas que ya se encuentran totalmente formateadas para ser desplegadas en el navegador

⁸Las *cookies* son un elemento de los navegadores, estas permiten que el servidor *Web* almacene información en la computadora del usuario *Web*; son utilizadas para personalizar el sitio *Web* para usuarios frecuentes.

(cliente). La interacción entre el usuario y el sistema es a través del servidor *Web*.

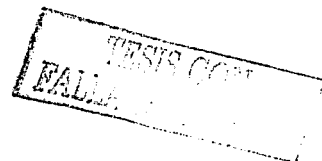
- Servidor *Web*. Este es el punto de unión entre el sistema y el navegador cliente. En esta arquitectura los navegadores (cliente) accesan el sistema solamente a través del servidor *Web*, el cual acepta peticiones de páginas *Web* (páginas HTML estáticas o servidor). Dependiendo de la petición, el servidor *Web* puede iniciar algunos procesos. Si la página *Web* solicitada es una página que se genera dinámicamente (CGI, ISAPI, NSAPI), el servidor *Web* delega la responsabilidad al intérprete o módulo ejecutable apropiado. El resultado es una página HTML formateada completamente para ser desplegada en el navegador (cliente).
- Conexión HTTP. HTTP es el tipo de protocolo de comunicación comúnmente utilizado en las aplicaciones de este tipo, aunque también es posible utilizar una conexión segura HTTP vía SSL (*Secure Sockets Layer*). Cada vez que el navegador (cliente) o el servidor *Web* envían información se establece una conexión entre ellos.
- Páginas HTML. Son páginas *Web* que no necesitan ningún tipo de procesamiento, contienen ayudas y/o formas. Estas son las llamadas páginas estáticas. Cuando un navegador (cliente) hace una petición de una página HTML, el servidor *Web* busca el archivo y lo envía hacia el cliente sin hacer ningún tipo de procesamiento.
- Páginas servidor. Son páginas *Web* generadas dinámicamente que tienen acceso a los recursos del servidor. Estas páginas son generadas por CGI, ASP, JSP, etcétera. Son procesadas por el intérprete o módulo ejecutable correspondiente en el servidor de aplicaciones. Este tipo de páginas tienen acceso a todos los recursos del servidor incluyendo el acceso a bases de datos, sistemas heredados, sistemas mercantiles, etcétera.



- **Servidor de aplicaciones.** El servidor de aplicaciones es el encargado de ejecutar las transacciones lógicas del lado del servidor. Éste servidor ejecuta el código de las páginas servidor. El servidor de aplicaciones puede localizarse en el mismo equipo donde se encuentra el servidor *Web*.
- **Persistencia.** La mayoría de las aplicaciones *Web* usan bases de datos para tener datos persistentes. Como se utilizan diferentes tecnologías para realizar la persistencia, este componente es etiquetado genéricamente como persistencia. La forma para conectar la base de datos y el sistema es permitiendo que los *scripts* de las páginas servidor tengan acceso directo a este componente.
- **Sistemas heredados.** Este componente es accesado a través del servidor de aplicaciones.
- **Sistemas mercantiles.** Los sistemas mercantiles, en este caso, son sistemas que permiten que las aplicaciones *Web* acepten y procesen pagos mediante tarjetas de crédito. Al igual que el componente anterior, éste es accesado a través del servidor de aplicaciones.

La dinámica de la arquitectura cliente *Web* débil es la siguiente:

Las transacciones lógicas son ejecutadas solamente en respuesta a páginas *Web* solicitadas por el navegador (cliente). Como se menciona en los componentes, los clientes accesan el sistema utilizando el protocolo HTTP para solicitar páginas *Web* al servidor *Web*. Si la página solicitada es una página HTML estática, el servidor *Web* únicamente busca la página y la envía al cliente. Si la página es un *script* (página servidor), el servidor delega la responsabilidad al servidor de aplicaciones. Éste interpreta el *script* mediante el interprete o módulo ejecutable correspondiente e interactúa con los recursos del servidor (bases de datos, sistemas heredados, sistemas mercantiles, etcétera).



La figura 1.4 muestra los componentes involucrados en las aplicaciones cliente *Web* débil.

1.8.2. Cliente *Web* robusto (thick *Web* client)

Las aplicaciones Cliente *Web* robusto extienden al patrón Cliente *Web* débil y son llamadas así ya que algunos procesos son ejecutados por el cliente. A continuación se mencionan los elementos que esta arquitectura agrega al patrón anterior.

Estructura

- **Scripts del cliente.** Éstos son “programas”, escritos en JavaScript o VBScript, que se encuentran incrustados en páginas HTML, estos programas son interpretados por el cliente. Para interpretar estos *scripts* se utiliza HTML y DOM (*Document Object Model*)⁹.
- **Documentos XML.** Son documentos formateados con el Lenguaje de marcado extensible.
- **Control ActiveX.** Es un objetos que puede ser referenciado en un *script* cliente y descargado en el cliente, si es necesario. Éste tipo de controles tienen acceso total a los recursos del cliente. Ya que este tipo de objetos representan un peligro en la seguridad del cliente, éste debe ser configurado para que no acepte o advierta que controles ActiveX van a ser descargados en él.
- **Applets Java.** Los applets de Java son programas que se almacenan en el servidor, pero cuando un cliente los invoca indirectamente a través de una página *Web* éstos son descargados y ejecutados en él. Los applets, a diferencia de los controles ActiveX, tienen acceso limitado a los recursos del servidor.

⁹DOM es una interfaz independiente de la plataforma y del lenguaje que permite que los programas y *scripts* accedan y actualicen dinámicamente el contenido, estructura y estilo de los documentos [12].

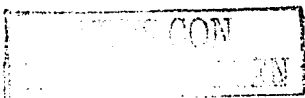
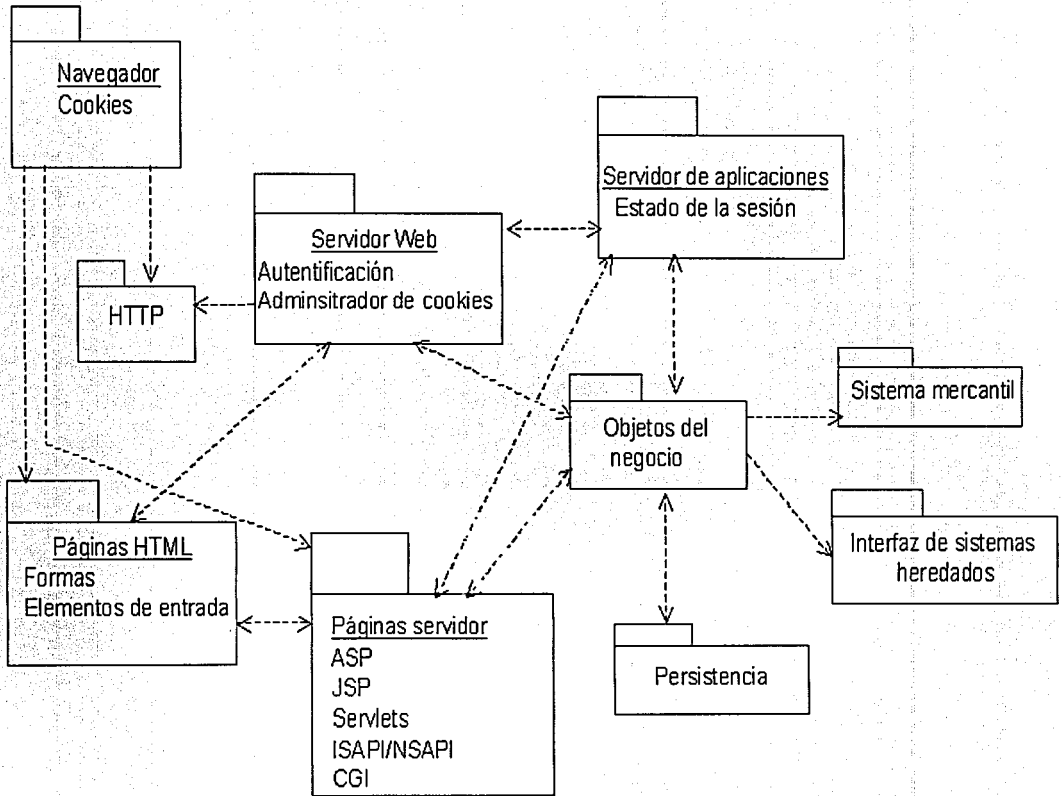


Figura 1.4: Vista lógica de las aplicaciones cliente Web débil



- **JavaBeans.** Un *bean* es un contenedor que puede almacenar y obtener cierto tipo de información. Cada pieza de información es conocida como una propiedad y puede ser solicitada o extraída individualmente. Los JavaBeans son el equivalente a los controles ActiveX de Microsoft.

La dinámica de la arquitectura cliente *Web* robusto es similar a la cliente *Web* débil. La comunicación entre el cliente y el servidor es durante la solicitud de una página, sin embargo las transacciones lógicas pueden ser ejecutadas parcialmente en el cliente a través de *scripts*, controles ActiveX, o applets. Cuando se envía una página al navegador (cliente), ésta puede contener *scripts*, controles ActiveX o applets que serán usados simplemente para realzar la interfaz de usuario o para contribuir en alguna transacción lógica (comúnmente validación de datos). Para que los applets y controles ActiveX sean usados, estos deben estar especificados en el contenido de una página HTML, pueden trabajar independientemente de cualquier *script* en la página o ser guiados por éstos. Los *scripts* en las páginas HTML pueden responder a eventos especiales enviados por el navegador (cliente), un evento puede ser que se terminó de descargar una página *Web* o que el *mouse* del usuario debe ser movido a una región específica de la página. Tales eventos tienen acceso a la interfaz DOM del navegador (cliente).

XML es utilizado como un mecanismo de intercambio de información entre el navegador (cliente) y el servidor *Web*. XML es el lenguaje que hace posible que los diseñadores creen etiquetas de propósito específico y mediante la interfaz DOM los *scripts* cliente pueden tener acceso a documentos XML. Los applets y controles ActiveX, en el cliente, pueden enviar o solicitar documentos XML.

La figura 1.5 muestra un diagrama con la vista lógica de este patrón.

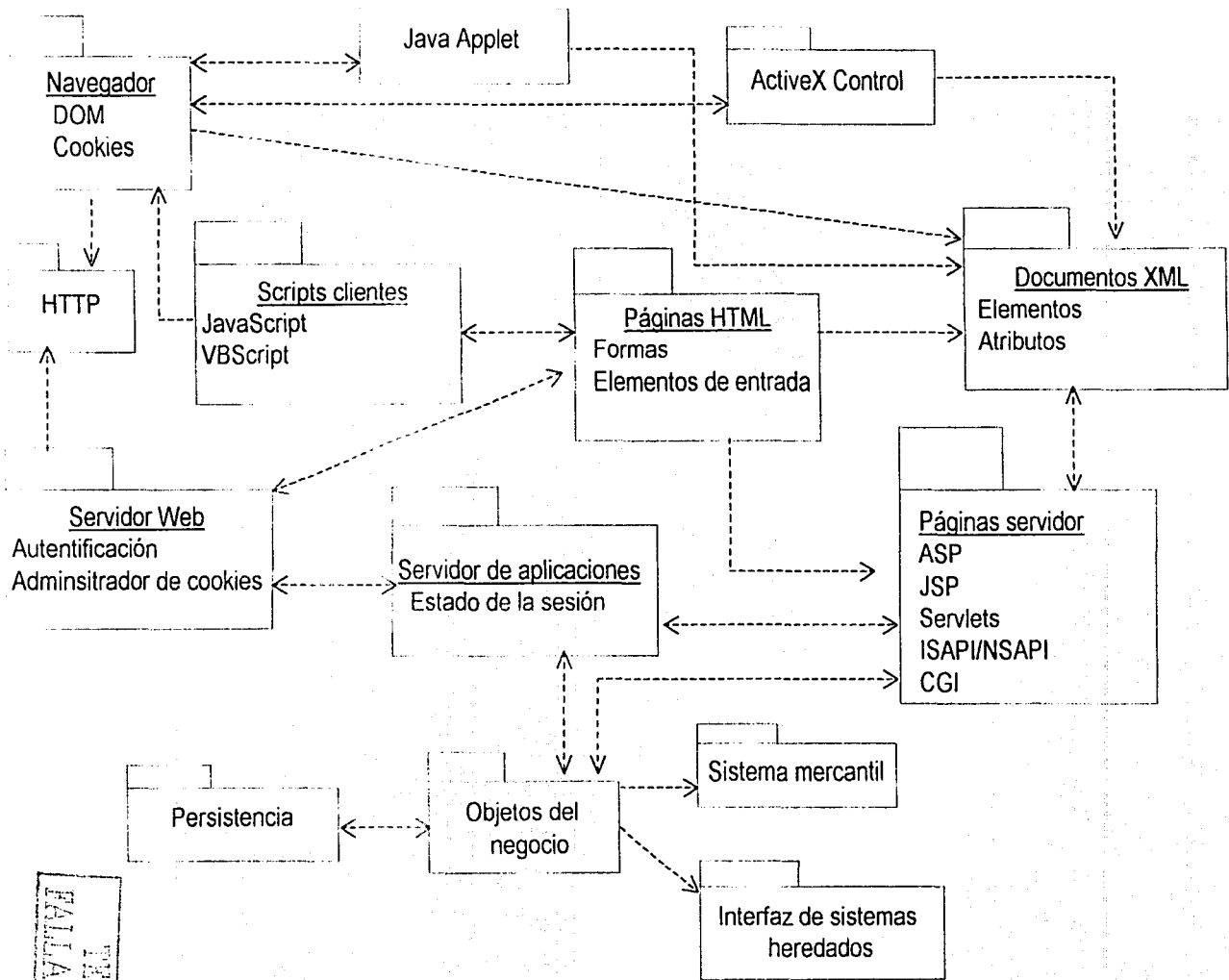


Figura 1.5: Vista lógica de las aplicaciones cliente Web robusto

TESIS CON
 FALLA DE ORIGEN

1.8.3. La Web como mecanismo de entrega (Web delivery)

Estas aplicaciones son llamadas así ya que la *Web* actúa principalmente como un mecanismo de entrega de las aplicaciones cliente/servidor de objetos distribuidas tradicionales. En las aplicaciones *Web delivery*, además de utilizarse el protocolo HTTP se utilizan otros protocolos de comunicación como IIOP, DCOM, y RMI, estos protocolos son utilizados para trabajar con sistemas de objetos distribuidos. Este patrón es utilizado en casos donde se tiene control sobre las configuraciones de la red y del cliente.

Estructura

El patrón *Web delivery* tiene los mismos elementos del patrón descrito anteriormente pero su agregado son los protocolos de comunicación utilizados para el manejo de objetos distribuidos.

- DCOM (*Distributed COM*). Es un protocolo de objetos distribuidos de Microsoft. Este protocolo permite que objetos que se encuentran en una máquina puedan interactuar e invocar métodos de objetos que se encuentran en otras máquinas.
- IIOP *Internet Inter-ORB*. Este es un protocolo que permite la interacción de objetos distribuidos a través de *Internet* o a través de cualquier red basada en el protocolo TCP/IP.
- RMI *Remote Method Invocation*. La invocación remota de métodos es el mecanismo que proporciona Java para la interacción de objetos entre diversas máquinas. El JRMP (*Java Remote Method Protocol*) es el protocolo nativo para RMI, pero este puede también ser implementado con el protocolo IIOP.

La dinámica de la arquitectura la *Web* como mecanismo de entrega es la siguiente: El navegador (cliente) es utilizado como un contenedor de la interfaz de usuario y de

algunos objetos que se comunican, independientemente del navegador (cliente), con objetos de los servidores. La comunicación entre el cliente y los objetos del servidor se hace a través de los protocolos IIOP, RMI, y DCOM. En aplicaciones de este tipo se debe tener una red sólida ya que la conexión entre los objetos cliente y servidor dura más que la proporcionada por el protocolo HTTP.

La figura 1.6 muestra el diagrama con la vista lógica que este patrón.

Conallen [11] muestra los patrones para las aplicaciones *Web* más comunes pero asevera que esta clasificación no es cerrada ya que las revoluciones tecnológicas parecen suceder anualmente.

FIN CON
FALLA DE ORIGEN

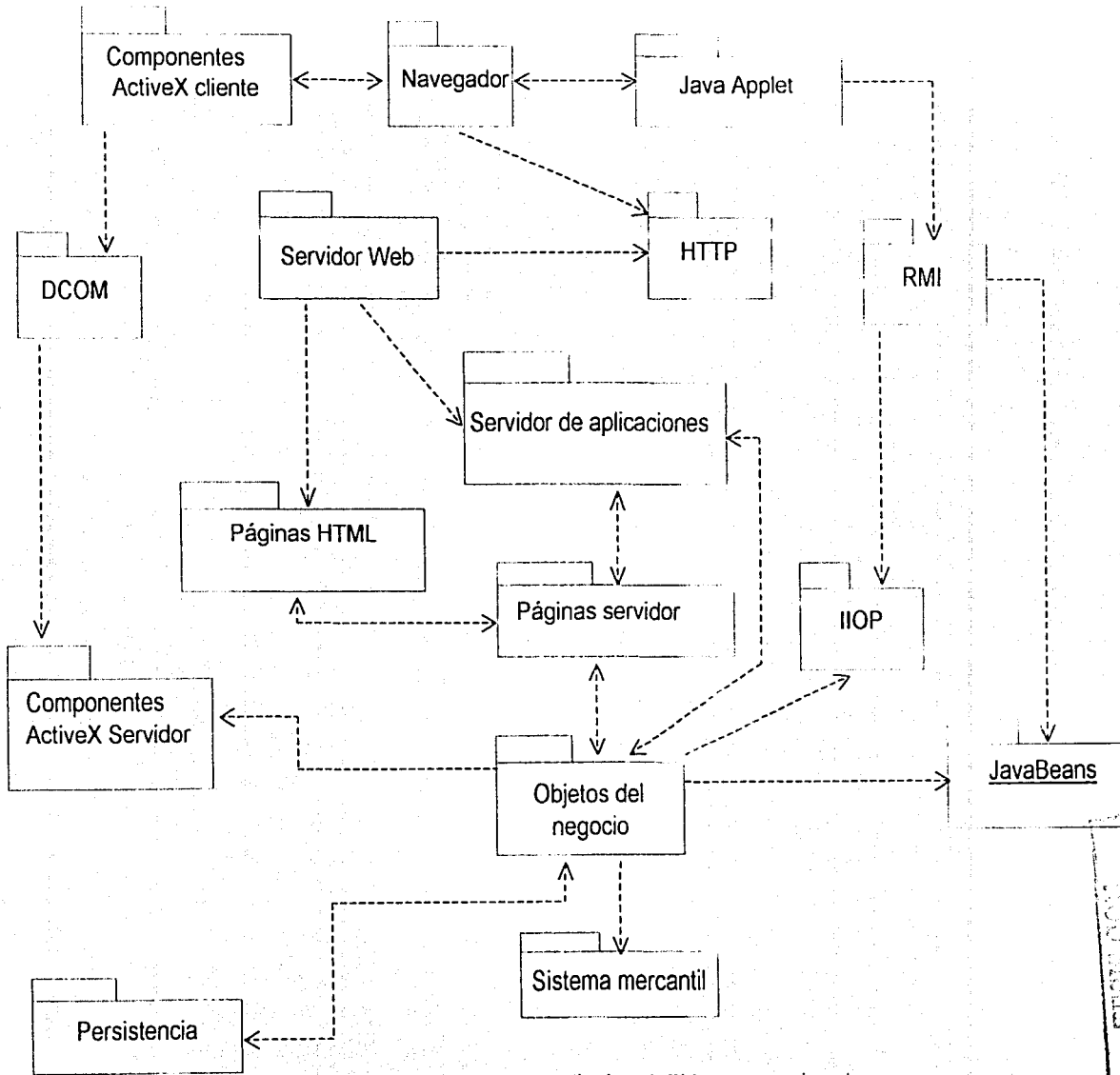


Figura 1.6: vista lógica de las aplicaciones la Web como mecanismo de entrega

INSTITUCION
 UNIVERSITARIA

Capítulo 2

Indicadores de ciencia y tecnología

El presente capítulo proporciona una base conceptual que permitirá comprender los aspectos más relevantes involucrados en el desarrollo de un proyecto que ayudará en el proceso de obtención de indicadores de ciencia y tecnología a través de bases de datos bibliográficas.

2.1. Antecedentes del análisis del desarrollo científico

La necesidad de evaluar la productividad científica ha adquirido gran importancia debido a que a partir de dicha evaluación se pueden asignar recursos económicos de forma más adecuada. El análisis del desarrollo de la ciencia se inició a principios del siglo XX, al respecto Sancho [49] asevera que “algunos de los primeros trabajos en bibliometría fueron resultado de una curiosidad innata por entender el desarrollo científico”. La misma autora cita, como primer trabajo realizado en esta área, el recuento del número de publicaciones por país, sobre los trabajos de anatomía existentes en el periodo 1543-1860.

El término usado para denominar este tipo de análisis fue variando a través del tiem-

po, Gorbea Portal [24] menciona que fue hasta 1969 cuando Alan Pritchard definió el término Bibliometría como “la aplicación de los métodos matemáticos y estadísticos a los libros y otros medios de comunicación”¹.

Otro término que comenzó a acuñarse paralelamente al de Bibliometría y que también está estrechamente relacionado con el análisis del desarrollo científico es el de Cienciometría o Ciencia de la Ciencia. Sancho [49] lo cita como “la aplicación de métodos cuantitativos a la investigación sobre el desarrollo de la ciencia considerada como proceso informativo”.

La diferencia entre la Bibliometría y la Cienciometría radica en el hecho de que la segunda está más enfocada hacia la política científica, es decir, cómo se organiza la comunidad científica y con esto conocer cómo se desarrolla el crecimiento de la ciencia. Es importante señalar que la Bibliometría ha sido y continua siendo utilizada para realizar el análisis del desarrollo de la ciencia y la tecnología, a través del establecimiento de indicadores bibliométricos.

En un amplio sentido, un indicador es un parámetro para evaluar el estado de cualquier actividad, así tenemos que un indicador bibliométrico es un parámetro que muestra el estado de la actividad científica. Bordons [6] dice que “los indicadores bibliométricos proporcionan información principalmente de la producción, comunicación y consumo de la información científica, así como de la especialización y la visibilidad de las publicaciones, y, por consiguiente de la investigación científica, de los vínculos entre la ciencia y la tecnología, de las relaciones entre diversas disciplinas científicas”. Si se realiza un estudio en base al uso de indicadores bibliométricos es posible conocer como ha ido evolucionando y desarrollándose el conocimiento y la

¹Actualmente, entre los medios más utilizados también se encuentran las revistas, específicamente los artículos. Es común que las revistas incluidas sean las que pertenecen a la corriente principal o *mainstream*.

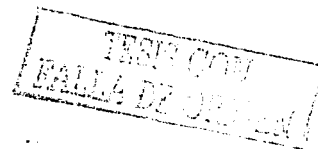
Cuadro 2.1: Tipología para la definición y clasificación de la bibliometría y cienciometría

Tipología	Bibliometría	Cienciometría
Objeto de estudio	Libros, documentos, revistas, artículos, autores y usuarios	Disciplinas, materias, campos, esferas
Variables	Números en circulación, citas, frecuencia de aparición de palabras, longitud de las oraciones, etc.	Aspectos que diferencian a las disciplinas y a las subdisciplinas. Revistas, autores, trabajos, forma en que se comunican los científicos
Método	Clasificación, frecuencia, distribución	Análisis de conjunto y de correspondencia
Objetivos	Asignar recursos, tiempo, dinero, etc.	Identificar esferas de interés; dónde se encuentran las materias; corresponder cómo y con que frecuencia se comunican los científicos

actividad científica y tecnológica. La gran utilización de este tipo de indicadores se debe a que las publicaciones son los principales resultados de la actividad científica. Un indicador cienciométrico, es una medida que provee información sobre los resultados de la actividad científica en una institución, país o región del mundo [30]. En el cuadro 2.1 se muestra un fragmento de una clasificación de los campos de aplicación de la bibliometría y cienciometría citado por Macías-Chapula [41].

2.2. Generación del conocimiento científico

Debido a que, en cualquier campo, las publicaciones son los principales resultados de la actividad científica, es importante mostrar el proceso de generación del conocimiento científico, cómo éste es difundido en la comunidad científica y posteriormente

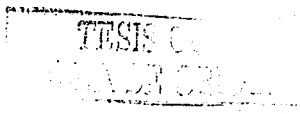


utilizado para realizar análisis de desarrollo y productividad científica. La investigación se inicia con un dato, una evidencia, información, es difundida a través de medios de comunicación formales e informales y por medio de éstos llega a ser publicada en revistas, que es el resultado esperado en la investigación científica (artículos). La publicación de artículos en este medio formal de comunicación fomenta el desarrollo de bases de datos que podrán ser accedidas para obtener indicadores bibliométricos.

En la figura 2.1 se muestra un diagrama de un modelo de la comunicación científica citado por Hurd [31], en el cual los resultados de la investigación científica, antes de su comunicación formal a través de su publicación como artículos en revistas científicas, son difundidos frecuentemente en seminarios y conferencias. Los datos de los artículos son recogidos en primera instancia por el servicio de alerta de *Current Contents* para luego ser indexados en las bases de datos bibliográficas. *Current Contents* es una de las bases de datos proporcionadas por el *Institute for Scientific Information (ISI)*² en Filadelfia, provee acceso a la información bibliográfica más completa de artículos, editoriales, cartas, comentarios, resumen de reuniones, y reseñas de libros. Esta base de datos es actualizada semanalmente, de esta forma ayuda a los usuarios a mantenerse actualizados, y les proporciona un panorama completo de la investigación global al día.

La importancia de utilizar la bases de datos *Current Contents* para la obtención de indicadores bibliométricos radica en el hecho de su frecuente actualización, así, el análisis del desarrollo de la ciencia realizado con estos indicadores es al día, en cambio el análisis que se hace utilizando las otras bases de datos bibliográficas es retrospectivo ya que éstas son generadas posteriormente a que los datos son almacenados en *Current Contents*.

²Véase <http://www.isinet.com>



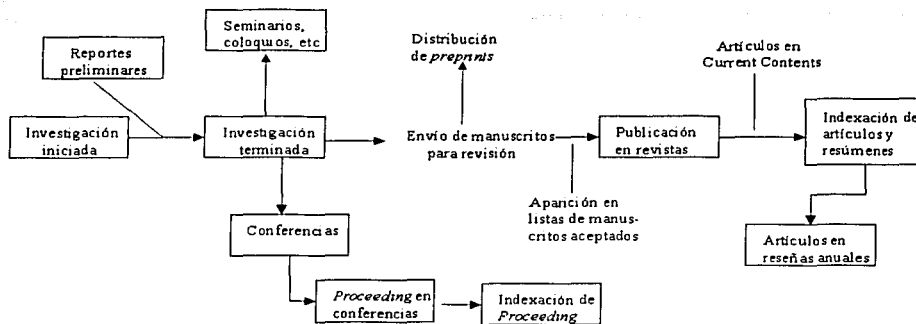


Figura 2.1: Modelo de comunicación científica Garvey-Griffith

2.3. Importancia de los indicadores bibliométricos

Actualmente los indicadores han adquirido gran importancia ya que los gobiernos y la iniciativa privada están demostrando gran interés en las áreas científicas en las que van a invertir recursos económicos.

Los indicadores bibliométricos no sólo proporcionan información que permite conocer los avances e impacto de la ciencia, sino también muestran el surgimiento de nuevas disciplinas. Así mismo a través del estudio de éstos es posible conocer la forma en que interactúan los países, organizaciones, grupos de trabajo, etcétera.

Russell y Rousseau [48] mencionan que los indicadores de investigación científica pueden dividirse en dos grandes grupos, estos son: indicadores de entrada e indicadores de salida. Los indicadores de entrada son aquellos relacionados con el dinero invertido, el equipo utilizado y el grupo de trabajo, y los indicadores de salida, como los indicadores basados en literatura, representan los efectos y resultados del proceso de investigación. También menciona que los indicadores pueden ser relativos o absolutos, define los indicadores absolutos como aquellos que se refieren a una característica en

particular de la actividad de investigación como son número de artículos publicados, número de citas o la cantidad de recursos económicos invertidos, y los indicadores relativos como aquellos que muestran la relación entre dos o más aspectos tales como el número de artículos publicados por grupo de investigación o el número de citas por artículo.

Es importante mencionar que cuando se utilizan los indicadores bibliométricos para realizar un análisis, éstos deben ser tomados en conjunto y la fuente de datos utilizada debe cubrir adecuadamente la producción científica del grupo que se desea evaluar para que dicho análisis tenga validez. La evaluación de la producción científica tiene mayor validez cuando se realiza en grupos grandes, como por ejemplo análisis de tipo nacional o por disciplina, que en análisis de grupos pequeños, como por ejemplo grupos de investigación, y esto se debe a que los indicadores estadísticos tienen mayor validez cuando se aplican a unidades grandes [6, 48]. También, la validez de estos tipos de indicadores varía de acuerdo al área que se evalúa ya que en las áreas básicas se tiene más producción de publicaciones difundidas a través de revistas de amplia circulación mundial que en las áreas tecnológicas o aplicadas donde muchas veces los resultados aparecen en informes internos de circulación restringida..

2.4. Indicadores de ciencia y tecnología

A continuación se mencionan algunos de los indicadores que se utilizan comúnmente para llevar a cabo estudios sobre desarrollo de la ciencia y tecnología.

2.4.1. Indicadores de calidad científica

Estos tipos de indicadores son de los más difíciles de obtener ya que se basan en la opinión que proporcionan los expertos en el tema, sobre un trabajo en particular, es posible aplicarlos a uno o más autores. Con éstos se trata de determinar la calidad

científica de un trabajo en cuanto a su contenido y metodología empleada.

2.4.2. Indicadores de producción científica

En este tipo de indicadores bibliométricos se tratan aspectos puramente cuantitativos, como son número y distribución de publicaciones (geográfica, temática, por institución, etcétera.), así como la productividad de los autores y su colaboración a través de la co-autoría de publicaciones.

Indicadores por número y distribución de publicaciones

Los indicadores bibliométricos por número y distribución de publicaciones son los indicadores más comúnmente usados, y son los más sencillos de obtener. Este tipo de indicadores se basa en el número de publicaciones y la distribución que ha realizado un país, una organización, un grupo de trabajo, etcétera.

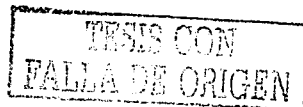
Los estudiosos del área consideran como publicación a todos aquellos documentos propagados a través de canales formales y públicos, los canales más utilizados para realizar este tipo de estudios son las revistas y las patentes.

Productividad de los autores

Este tipo de indicadores muestra el número de publicaciones y su distribución, basándose en la productividad de un autor en particular.

Colaboración en las publicaciones

Los indicadores de colaboración en las publicaciones son utilizados para conocer la cooperación que hay entre grupos de científicos, de organizaciones, de países, etcétera.



2.4.3. Indicadores del impacto de los trabajos

El impacto del trabajo de investigación en la literatura juega un papel muy importante porque se refiere al número de citas que un determinado trabajo ha recibido. Este indicador es el más utilizado y también el más polémico, ya que si el número de citas recibidas es muy alto esto no indica necesariamente que el trabajo sea de muy buena calidad y viceversa. Algunos investigadores mencionan que ciertos trabajos que han considerado más importantes y de mejor calidad son los que menos citas han recibido.

2.4.4. Conexión entre trabajos y autores científicos

Estos indicadores se basan en el análisis de las referencias contenidas en los artículos, para determinar el número de referencias que un trabajo hace hacia otro, y el número de citas que recibe el artículo al cual se hace referencia. Con este tipo de indicadores es posible identificar los trabajos anteriores que son utilizados por un autor para desarrollar su investigación.

Mediante el análisis de las referencias es posible determinar la vida media de la literatura utilizada en cualquier campo de la ciencia. También, se puede conocer la producción de los grupos que dirigen una determinada disciplina, a estos se les llama "frente de investigación", y a los autores que forman parte de ellos se les llama "colegios invisibles" [49].

2.4.5. Indicadores de impacto de las fuentes

En este tipo de indicadores se tienen el factor de impacto de las revistas, el índice de inmediatez, y la influencia de las revistas.

Factor de impacto de las revistas

Este índice muestra la frecuencia con la que un artículo promedio ha sido citado en un año específico. El factor de impacto de una revista de un determinado año se calcula sumando todas las citas recibidas por ésta en ese año, tales citas corresponden a los artículos publicados en los dos años anteriores, y se dividen por el número total de artículos publicados por dicha revista en esos dos años. Por ejemplo, si se desea calcular el factor de impacto de la revista X en el año 2001, si el número de citas recibidas en el año 2001 por artículos publicados en los años 1999 y 2000 es de 50 y el total de artículos publicados en esos dos años es de 40 entonces, tenemos que el factor de impacto de la revista X en el año 2001 es de 1.25.

Índice de inmediatez

El índice de inmediatez hace referencia a la rapidez con la cual un artículo es citado.

Influencia de las revistas

Este indicador se utiliza para conocer la influencia que tiene una revista, el cálculo se hace en base al número de referencias que una revista proporciona y el número de citas que ésta recibe. El indicador de influencia de las revistas no es tan importante como los dos anteriores ya que la influencia de las revistas depende del grado de especialización, la disponibilidad, idioma, país de origen, etcétera.

No obstante que existen otros tipos de indicadores de ciencia y tecnología, lo presentados en esta sección son los más comúnmente usados.

2.5. Fuentes de información (bases de datos bibliográficas)

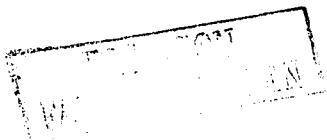
Un aspecto importante en el desarrollo de los indicadores de ciencia y tecnología es la selección las fuentes de información de las que se extraen los datos, así como un conocimiento adecuado de los tipos de problemas que se deben resolver. Los datos pueden ser extraídos de diferentes fuentes de información y dependiendo del tipo de análisis bibliométrico que se desea realizar es la complejidad del problema que se debe resolver.

Una de las principales fuentes de información son las bases de datos bibliográficas, cuya importancia radica en su finalidad bibliográfica. Estas bases de datos contienen la mayoría de datos necesarios para realizar un análisis bibliométrico³.

Garduño Vera [23] menciona que en el proceso de automatizar el control de la información bibliográfica se cuenta con formatos de almacenamiento y recuperación. Estos formatos son: de almacenamiento, consulta en línea, impresión, y de intercambio.

Los formatos de almacenamiento están basados en etiquetas para la identificación de los atributos bibliográficos y no bibliográficos. Los formatos de consulta en línea se refieren a la forma en cómo se visualizará la información, por lo general este formato se presenta bajo la forma y contenido de fichas catalográficas, aunque también la presentación puede estar basada en el formato de almacenamiento utilizado. Los formatos de impresión, como su nombre lo indica, se refieren a la forma en que los datos almacenados pueden ser impresos. Los más comunes son: tarjetas catalográficas, listados en orden alfabético de autores o títulos, bibliografías temáticas, etc. Los formatos de intercambio están diseñados para hacer intercambio de información entre

³Las bases de datos bibliográficas surgen de la necesidad de identificar de manera más rápida el conocimiento científico que se ha publicado.



distintas bases de datos de una forma sencilla para facilitar la realización de actividades bibliotecarias.

Los formatos antes mencionados, al parecer, facilitan la extracción de datos para realizar el análisis, pero son tantos los tipos de formatos bibliográficos existentes y las claves no siempre se usan en forma correcta que la tarea se vuelve compleja.

2.6. Estandarización de los datos

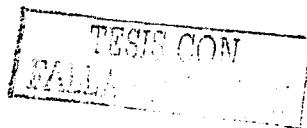
Uno de los problemas más graves en el análisis bibliométrico es la normalización de los datos bibliográficos, independientemente de las fuentes de información que hayan usado. El problema radica en el hecho de que en los datos existe inconsistencia de información debido a las fuentes de donde éstos se obtienen y también por los errores cometidos cuando se almacenan.

Algunos de los errores más frecuentemente encontrados son:

- Errores ortográficos introducidos por el capturista o por el autor de la fuente bibliográfica
- Errores en orden de las letras, introducidos por el capturista
- Omisión de datos, por ejemplo los autores no escriben sus datos, ni los datos de la institución a la que están afiliados de la misma forma.

Para realizar un estudio bibliométrico lo primero que se hace es extraer los datos de las fuentes de información seleccionadas, después se realiza una estandarización dependiendo del tipo de análisis que se desea realizar.

Algunas de los pasos que se siguen actualmente para hacer la normalización son:



- La normalización se hace de forma “artesanal”, es decir, se extrae la información contenida en los registros bibliográficos en formato texto, los datos se separan con un carácter especial y posteriormente se realiza una estandarización usando un procesador de palabras, regularmente *Word* y la hoja de calculo *Excel*. Finalmente los datos son migrados a una base de datos relacional que en su mayoría solamente cuenta con una relación.
- La normalización se realiza en forma “automática”. Se dice que la normalización es automática ya que con base en los datos que serán extraídos se diseña una base de datos relacional, se desarrollan algoritmos que permitan “separar” los datos extraídos y estos son normalizados en base a catálogos de autoridad o tesauros⁴ que fueron previamente almacenados en la base de datos.

Una vez que los datos se encuentran almacenados en una base de datos relacional se procede a realizar consultas para que, con la ayuda de herramientas de estadística y gráficas, se proceda a realizar los estudios bibliométricos.

2.7. Los Indicadores de Ciencia y Tecnología en la World Wide Web

Como se puede observar, las bases de datos que se construyen pueden ser usadas solamente por los encargados de realizar los estudios. Últimamente y debido al auge que han tenido los indicadores de ciencia y tecnología se han estado desarrollando aplicaciones que permitan acceder a este tipo de información a través de la *Web*.

Algunas de las instituciones que han desarrollado aplicaciones que permiten la obtención de indicadores de ciencia y tecnología a través de la *Web* son: la Universidad de

⁴Un tesoro contiene todos los términos de una área temática, institución, área geográfica, etc. y en algunos casos se denomina catálogo de autoridad.

Sussex de Inglaterra y el Centro de Información y Documentación Científica (CINDOC) de España.

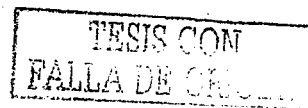
El proyecto *Bibliometric Evaluation of Sectorial Scientific Trends (BESST)* [36, 37], desarrollado por la primera, permite obtener indicadores de 6 sectores institucionales ingleses. El CINDOC en colaboración con el Instituto de Investigaciones Biomédicas (IIB) y el Departamento de Sistemas Telemáticos (DIT) de la Universidad Politécnica de Madrid llevaron a cabo el proyecto "Indicadores de Ciencia y Tecnología (INDICYT)" [18, 20]. A través de éste es posible obtener indicadores por regiones geográficas, sectores institucionales y áreas temáticas con referencia a la actividad española.

Los principales problemas detectados en este tipo de desarrollo fueron debido a que el desempeño del manejador de base de datos usado no era óptimo, ya que el volumen de información es muy grande y las consultas realizadas a la base de datos eran complejas, a pesar de que se cuenta con un diseño de base de datos capaz de responder todo tipo de consultas, no es posible diseñar una interfaz gráfica que satisfaga la mayoría de las peticiones que un usuario conocedor de análisis bibliométricos, pueda realizar.

2.8. Selección de fuentes de datos

Puesto que las fuentes de información para la extracción de datos son diversas y la estandarización que se debe realizar depende del nivel al que se desea llegar en el análisis del desarrollo de la ciencia, es importante preguntar *¿En dónde se encuentran dichas fuentes de información?, ¿Es fácil acceder a ellas?*

El acceso a las fuentes de información es otro de los grandes problemas a los que se deben enfrentar los encargados de realizar análisis de ciencia y tecnología. Existen



empresas que se dedican a recopilar los datos de revistas consideradas de nivel internacional; si las instituciones cuentan con recursos económicos para suscribirse a dichas bases de datos, estarán en la posibilidad de realizar los análisis deseados.

Las principales fuentes de datos utilizadas para realizar estudios bibliométricos son las bases de datos del ISI. Las bases de datos producidas por ISI son las más completas en cuanto a la información que proporcionan sobre cada uno de los documentos. Esta organización proporciona bases de datos que permiten, a los investigadores, realizar análisis cuantitativos de actualidad de los avances de la ciencia. También, permiten determinar la importancia relativa de las revistas dentro de las disciplinas. Proporcionan información de los *proceedings*, componentes importantes en la comunicación de la comunidad científica, ya que en ellos es donde típicamente se hace la presentación de nuevas teorías, soluciones y desarrollo de nuevos conceptos. Además, son las únicas que proporcionan los elementos para obtener indicadores del impacto de los trabajos (análisis de citas). No obstante lo anterior, existen controversias en cuanto a los criterios que esta empresa utiliza para la inclusión de las revistas.

Para la mayoría de los países desarrollados el problema de acceso a las fuentes de información no es tan grave ya que gran parte de las revistas en las que sus científicos publican se encuentran cubiertas por las bases de datos de ISI, sin embargo, si el análisis que deseen realizar es muy local deberán recurrir a una base de datos de carácter nacional o recopilar ellos mismos los datos.

Para los países en vías de desarrollo el problema se acrecenta ya que la mayoría de las revistas que se utilizan para hacer difusión no son consideradas como revistas de corriente principal o internacionales y por lo tanto no son consideradas en esas bases de datos internacionales. Por otra parte, por ejemplo, Wesley [54] menciona que estos países tienen una mayor producción en el campo de la agricultura, y las revistas de esta área son las que menos cobertura internacional poseen.

RESERVADOS LOS DERECHOS
DE TODA MANERA

Capítulo 3

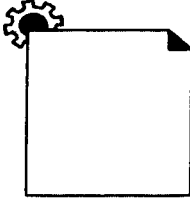
UML y las aplicaciones Web

Al igual que los sistemas tradicionales, las aplicaciones para el Web están integradas por un conjunto de fases de desarrollo, las cuales deben realizarse adecuadamente para la óptima evolución del software, dichas fases son: requerimientos del software, análisis, diseño, codificación, verificación, operación, y mantenimiento. En el presente capítulo se mostrará la extensión UML para el modelado de aplicaciones *Web*, así como la forma en que tal extensión es utilizada en las aplicaciones *Web* con arquitectura cliente *Web* débil.

3.1. Extensión UML para aplicaciones Web

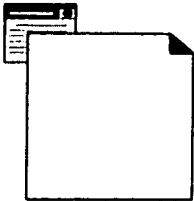
En esta sección se presentan los estereotipos, proporcionados por Conallen [11], que pueden ser usados para modelar aplicaciones *Web*.

Clase Página servidor



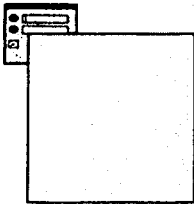
Las páginas servidor representan páginas *Web* que contienen *scripts* que son ejecutados por el servidor. Estos *script* interactúan con recursos del servidor tales como bases de datos, sistemas externos y realizan transacciones. Estas páginas solamente pueden estar relacionadas con objetos en el servidor.

Clase Páginas cliente



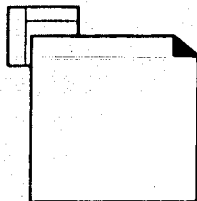
Una instancia de una página cliente es una página *Web*, además es una mezcla de datos, presentación y eventos lógicos. Las páginas clientes se localizan en el navegador (cliente) y pueden contener *scripts* que son interpretados por el mismo. Estas páginas no pueden interactuar con recursos del servidor.

Clase Formas



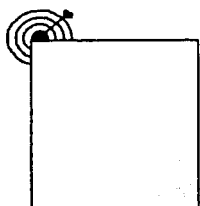
Una forma es una colección de campos de entrada que son parte de una página cliente. Éstas pueden ser vistas como agregaciones de las páginas cliente, las formas no tienen operaciones y cualquier operación que interactúe con ellas pertenece a su página cliente propietaria.

Clase Frameset



Un *frameset* es un contenedor de múltiples páginas *Web*. La ventana del navegador está dividida en rectángulos llamados *frames* y cada *frame* puede estar asociado con un *target*. El contenido de un *frame* puede ser una página *Web* u otro *frameset*. Un *frameset* es como un sinónimo de una página cliente, así que éste puede tener operaciones y atributos.

Clase Target

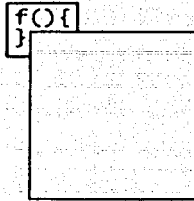


Un *target* es como se nombra al compartimento de la ventana del navegador en donde la página *Web* está localizada. Un *target* es un *frame* en una ventana definido por un *frameset*; también puede ser una nueva instancia de un navegador en vez de una ventana.

Clase JavaScript Object

Este estereotipo es utilizado para simular objetos definidos por el cliente mediante funciones de JavaScript. Las instancias de estos objetos existen solamente en el cliente y se denotan mediante `<<JavaScript Object>>`.

Clase ClientScript Object



Un objeto *clientscript* es una colección de *scripts* cliente contenidos en un archivo que puede ser bajado al navegador (cliente) mediante una solicitud específica. Estos *scripts* son utilizados por la aplicación *Web*.

Asociación Link

Un `<<link>>` es un apuntador de una página cliente a otra página. En un diagrama de clases, un *link* es una asociación entre una página cliente y otra página cliente o página servidor. Una asociación de este tipo se mapea directamente a una etiqueta HTML *anchor*. Este componente puede ir acompañado de una lista de parámetros que son pasados desde la página cliente.

Asociación Targeted Link

Un `<<targeted link>>` es una asociación similar a `<<link>>`, es un enlace cuya página asociada es enviada a otro *target*. Esta asociación se mapea directamente a una etiqueta HTML *anchor*, con el *target* especificado por el atributo *target*. Además de la lista de parámetros contiene el nombre del *target* en el cual el contenido de la página deberá ser enviado.

Asociación Frame Content

Una asociación *frame content* es una agregación de asociaciones que expresa un frame contenedor de otras páginas o *target*. Este tipo de asociación puede apuntar a otros *frameset*, lo cual indica *frames* anidados. Las etiquetas HTML asociadas son *row* y *col*.

Asociación Submit

La asociación <<*submit*>> siempre es entre una forma y una página servidor. Las formas envían los valores a ser procesados por el servidor *Web* a través de una página servidor. Esta asociación también tiene una lista de parámetros que son enviados por la página que realizó el enlace.

Asociación Builds

La asociación <<*builds*>> identifica que página servidor es responsable de la creación de una página cliente. Una página servidor puede crear muchas páginas cliente pero, una página cliente solamente puede ser creada por una página servidor.

Asociación Redirect

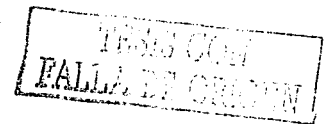
Una interrelación <<*redirect*>> es una asociación unidireccional con otras páginas *Web*, esta puede estar dirigida desde y hacia páginas cliente y servidor. Si la interrelación es originada por una página servidor, el procesamiento de la página solicitada puede continuar en otra página servidor que no necesariamente será la responsable de crear la página cliente solicitada. Si la interrelación es originada por una página cliente, esto indica que la página destino será solicitada automáticamente por el navegador.

Asociación IIOP

IIOP (*Internet Inter-ORB Protocol*) es un tipo especial de interrelación entre objetos en el cliente y objetos en el servidor. IIOP representa otro mecanismo de comunicación cliente/servidor. Usualmente, esta interrelación es entre JavaBeans del cliente y Enterprise JavaBeans del servidor.

Asociación RMI

RMI (*Remote Method Invocation*) es un mecanismo para que los applets de Java y los JavaBeans envíen mensajes a otros JavaBeans localizados en diferentes máquinas.



Usualmente, esta interrelación es entre JavaBeans o applets en un cliente y Enterprise JavaBeans en el servidor.

Atributo Input Element

Un *input element* es un atributo de un objeto `<<form>>`, se mapea directamente en una etiqueta `<input>` dentro de una forma. Este atributo es usado para almacenar una palabra o una línea de texto. El nombre del atributo está asociado a la etiqueta *name* y el valor inicial a la etiqueta *value*.

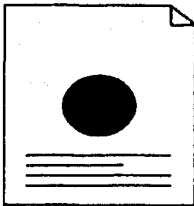
Atributo Select Element

Es una entrada de control usada en las formas. Este control permite que los usuarios seleccionen uno o más elementos de una lista. Mediante la etiqueta *size* es posible indicar el número de elementos que serán mostrados al mismo tiempo y mediante *multiple* se indica que se pueden seleccionar más de un elemento de la lista.

Atributo Text Area Element

Es un control de entrada, usado en las formas, que permite mostrar múltiples líneas de entrada. Mediante la etiqueta *row* se indica el número de líneas de texto visibles y con *cols* el número de caracteres por renglón.

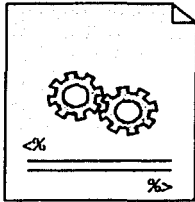
Componente Web Page



Este componente es una página *Web*. Éste puede ser solicitado por nombre por un cliente y es posible que no contenga *scripts* cliente o servidor. Usualmente son archivos de texto accedidos por el servidor *Web*, también pueden ser módulos compilados cargados e invocados por el servidor *Web*. En

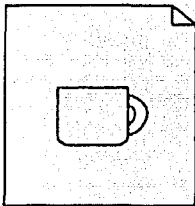
cualquiera de los casos al final se genera un documento en formato HTML que será enviado al navegador que hizo la petición. Se le debe indicar al servidor *Web* la trayectoria de la página *Web*.

Componente ASP Page



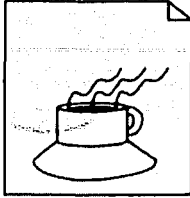
Son páginas *Web* implementadas con ASP. Este estereotipo es aplicable solamente en aplicaciones basadas en *Active Server Pages* de Microsoft. La funcionalidad es igual a la del componente *Web page*.

Componente JSP Page



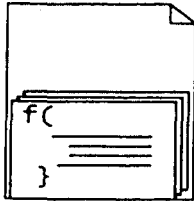
Este componente es igual a los dos anteriores pero es implementado con código JSP y es aplicable solamente en aplicaciones *Web* desarrolladas en ambientes que usan Java Server Pages.

Componente Servlet



Este componente es igual al componente *Web Pages* pero, solamente se usa en aplicaciones *Web* desarrolladas en ambientes que soportan *servlets* de Sun.

Componente Script Library



Es un componente que provee una biblioteca de subrutinas o funciones que pueden ser incluidas por otros componentes *Web page*.

3.2. Requerimientos del software

La obtención de los requerimientos del software es de gran utilidad ya que estos ayudan a delimitar las necesidades del sistema y a entender el problema, también son útiles en la planeación y control del proceso de desarrollo del sistema.

Los requerimientos de un sistema pueden ser funcionales y no funcionales. Los requerimientos funcionales son aquellos que expresan una acción que el sistema debe realizar. Los requerimientos no funcionales son aquellos referentes a la usabilidad, el rendimiento (*performance*), el hardware, la seguridad, etc.

En esta fase de desarrollo se utilizan los casos de uso ya que con ellos se captan más fácilmente los requerimientos funcionales del usuario del sistema. Un caso de uso es una secuencia de acciones que un actor realiza en un sistema para lograr una meta específica, es decir, describe el funcionamiento del sistema ante un estímulo externo. Éstos tienen rutas básicas de acciones que representan el flujo principal que debe seguir el sistema, rutas alternativas, así como condiciones de error. Aunque los casos de uso se dieron a conocer en la metodología orientada a objetos, éstos son también de gran utilidad en la programación orientada a procedimientos.

Un actor representa un rol que un usuario juega con respecto a un sistema, aunque también puede ser una entidad externa, como por ejemplo otro sistema.

Con la representación de casos de uso y actores en un mismo espacio se obtienen los diagramas de casos de uso.

3.3. Análisis

Los objetivos del análisis son la construcción de un modelo de objetos y la definición de un diccionario de datos. Esta fase implica centrar la atención en lo que se desea haga el sistema y no en cómo se hará.

3.3.1. Modelo de objetos

Para obtener el modelo de objetos, al igual que en la obtención de requerimientos funcionales, se debe suponer un ambiente de implementación ideal, para ello no deben tomarse en cuenta los siguientes aspectos: hardware, el lenguaje de programación, si el sistema será distribuido o no, requerimientos no funcionales, el manejador de bases de datos que se utilizará, entre otros.

No obstante que el sistema a modelar es una aplicación *Web*, el método para realizar el análisis no varía ya que en esta fase no se deben tomar en cuenta los elementos arriba mencionados. Al igual que en la obtención de requerimientos, en el análisis también se utilizan los casos de uso para identificar los objetos que están involucrados en el problema que el sistema que se está modelando resolverá.

Además de los diagramas de secuencia, en el modelado del presente sistema se utilizará el análisis de robustez (*robustness analysis*) de la metodología propuesta por Rosenberg [47] denominada Modelación unificada de objetos ICONIX, así como su notación. El mismo autor menciona que el concepto de análisis de robustez fue introducido al mundo de la orientación a objetos, en 1991, por Ivar Jacobson.

Este tipo de análisis involucra un análisis del texto de los casos de uso para obtener a una primera vista los objetos del dominio de problema. Una vez que se han encontrado, éstos deben ser clasificados dentro de los tres estereotipos de objetos de esta metodología.

Los estereotipos antes mencionados son: objetos de control, objetos de interfaz, y objetos del dominio. En la figura 3.1 se muestran los iconos para denotar los estereotipos antes mencionados y a continuación se definen.

- Objetos de control (control objects). Los objetos de control representan a los

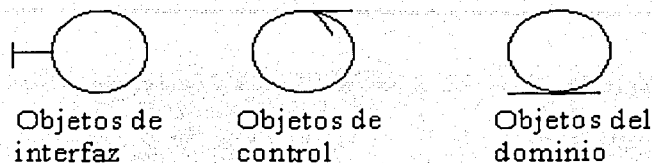


Figura 3.1: Notación de estereotipos

procesos que debe realizar el sistema. Los objetos de este tipo pueden interactuar con otros objetos de control y también con los dos tipos de objetos que se mencionan a continuación.

- **Objetos de interfaz (boundary objects).** Los objetos que representan la interfaz que hay entre un actor y el sistema. En las aplicaciones *Web* estos tipos de objetos pueden ser las páginas *Web* que utiliza el usuario para acceder al sistema. Este tipo de objetos solamente puede interactuar con objetos de control.
- **Objetos del dominio (entity objects).** Los objetos de este tipo son los objetos del dominio del problema, pertenecen al modelo de objetos y son aquellos que pueden ser persistentes, es decir, que su ciclo de vida es más largo que el tiempo de ejecución del sistema. Los objetos de este tipo solamente pueden ser accedados por objetos de control.

En el análisis de robustez se obtienen unos diagramas llamados “diagramas de robustez” que son el punto de unión entre el análisis y el diseño, son como una versión preliminar del diseño. En estos diagramas intervienen los tres estereotipos antes mencionados y los actores, se obtienen utilizando los diagramas de casos de usos, así como la descripción de cada uno de los casos de uso. Con estos diagramas se puede visualizar rápidamente la forma en cómo interactúan los actores, los procesos y los objetos involucrados en el sistema.

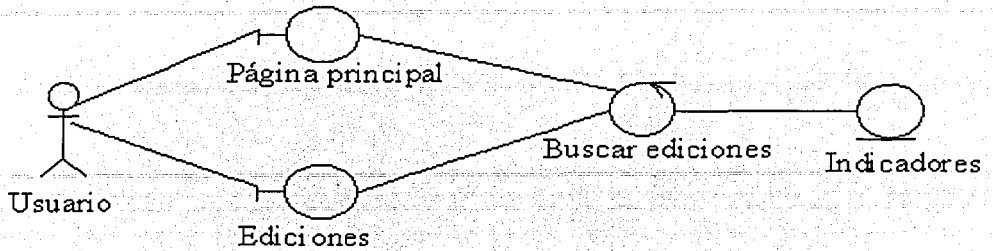


Figura 3.2: Diagrama simple de robustez

La figura 3.2 es un pequeño diagrama de robustez en el cual se muestra la forma en que un usuario interactúa con el sistema para que éste muestre el nombre de las ediciones existentes en el sistema que se modelará en el siguiente capítulo.

3.3.2. Diccionario de clases

En el diccionario de clases se debe definir cada una de las clases encontradas en el modelo de objetos, así como sus atributos. La creación es una tarea laboriosa pero necesaria ya que ésta ayuda a eliminar las ambigüedades en el sistema que se está modelando.

3.4. Diseño

La principal actividad que se realiza en la etapa de diseño es refinar el modelo obtenido durante el análisis, esto se debe a que es aquí donde se toman en cuenta los requerimientos no funcionales del sistema para que el modelo quede listo para la codificación.

Durante el diseño se continúa trabajando con los diagramas de secuencia pero de una forma más específica, además se adiciona el diagrama de componentes. Los componentes son mapeados a archivos ejecutables, bibliotecas estáticas o bibliotecas de ligado

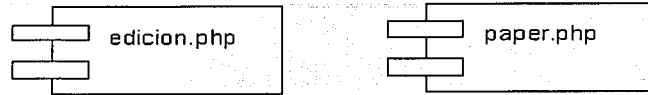


Figura 3.3: Componente

dinámico (DLL), archivos class de java, etc. Cada clase puede ser implementada en al menos un componente. La figura 3.3 muestra la representación de los componentes.

Otras de las actividades del diseño son la separación de objetos en los objetos que pertenecen al cliente y los que pertenecen al servidor, y la definición de páginas *Web*. Para realizar la separación de objetos antes mencionada se debe saber la arquitectura que será utilizada.

El diseño de aplicaciones *Web* difiere del diseño de aplicaciones tradicionales ya que en ellas intervienen otros elementos, entre éstos están las páginas *Web*. Las páginas *Web* actúan como contenedores generalizados de la interfaz de usuarios, éstas son la conexión entre los navegadores (cliente) y el sistema. Las páginas *Web* son tratadas como objetos dentro de las aplicaciones *Web* y es en ellas donde radica uno de los principales problemas ya que en algunos casos están sobrecargadas de acciones. Esto ocurre cuando las páginas *Web* contienen *scripts* que son ejecutados en el servidor, éstos interactúan con los recursos del servidor y posteriormente generan páginas que son enviadas al navegador (cliente) como una interfaz de usuario. También, pueden contener *scripts* que son ejecutados en el cliente. A este tipo de páginas Conallen [11] las llama páginas esquizofrénicas. UML no cuenta con notación suficiente para tratar casos como estos. Para realizar el diseño de aplicaciones *Web* el mismo autor propone una extensión a UML, esta extensión cuenta con iconos para denotar páginas *Web* cliente, servidor, formas, enlaces, *frames*, *scripts*, entre otros.

TESIS CON
FALLA DE ORIGEN

En la presente tesis se mostrará el uso de la extensión antes mencionada para modelar un aplicación *Web* con arquitectura cliente *Web* débil. En este tipo de arquitectura la tarea de separación de objetos cliente y servidor no es difícil ya que todos los objetos pertenecen al servidor.

Para encontrar las páginas *Web*, así como los objetos cliente y servidor se hace uso de los diagramas de secuencia.

3.4.1. Diagramas de secuencia para el diseño

Los diagramas de secuencia expresan la interacción entre un actor y el sistema, haciendo énfasis especial en el tiempo. Se pueden identificar objetos, responsabilidades, atributos y asociaciones entre clases.

Las aplicaciones *Web* cliente débil son las que tienen más restricciones en el uso de páginas *Web* ya que éstas contienen solamente elementos soportados por HTML.

En este tipo de arquitectura los actores interactúan solamente con páginas cliente y las páginas servidor solamente con los recursos del servidor. La forma de mapear los diagramas de secuencia y de robustez generados durante la fase de análisis es transformando directamente los *objetos interfaz* en páginas cliente, los *objetos de control* en páginas servidor. La figura 3.4 muestra la transformación realizada a la figura 3.2 que describen la forma en que se realiza una consulta a la base de datos para mostrar ediciones.

Este diagrama inicia cuando el actor (usuario) envía un mensaje a la página principal. El mensaje consiste en la solicitud de un servicio a “Buscar ediciones” y es implementado en la Página principal mediante la etiqueta HTML “< a href=ediciones.php>”. Buscar ediciones es una página servidor y es la encargada de realizar los procesos

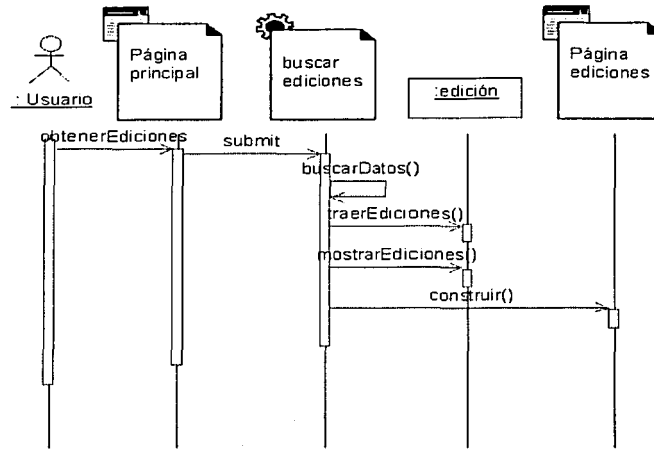


Figura 3.4: Diagrama de secuencia para mostrar ediciones

necesarios para presentar la información solicitada por el actor.

La figura 3.5 muestra las clases y componentes relacionados con la creación de una página *Web* que muestra ediciones. Este tipo de diagramas, así como los mostrados en las figuras 3.6, 3.7 y 3.8, son la ventaja que ofrece la utilización de la extensión UML para modelado de aplicaciones *Web*, ya que en ellos se integran los componentes, los objetos del dominio y la páginas *Web*.

3.4.2. Páginas servidor, enlaces, redireccionamientos y formas

Algunos de los principales elementos involucrados en las aplicaciones *Web* son los enlaces que hay entre los actores, páginas y clases. En la figura 3.6 se muestran estos elementos. En dicha figura se puede apreciar que las páginas cliente, en el tipo de patrón que se está utilizando, solamente interactúan con páginas servidor y esto se

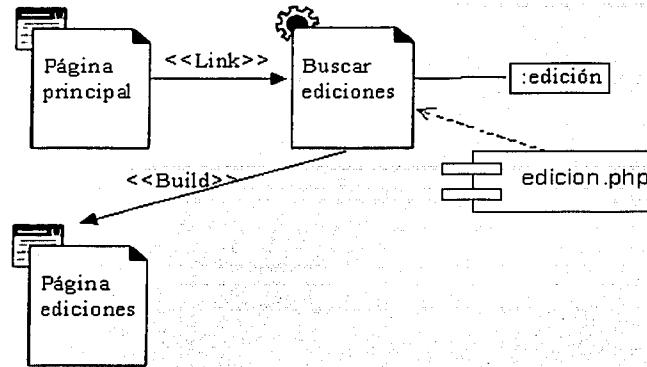


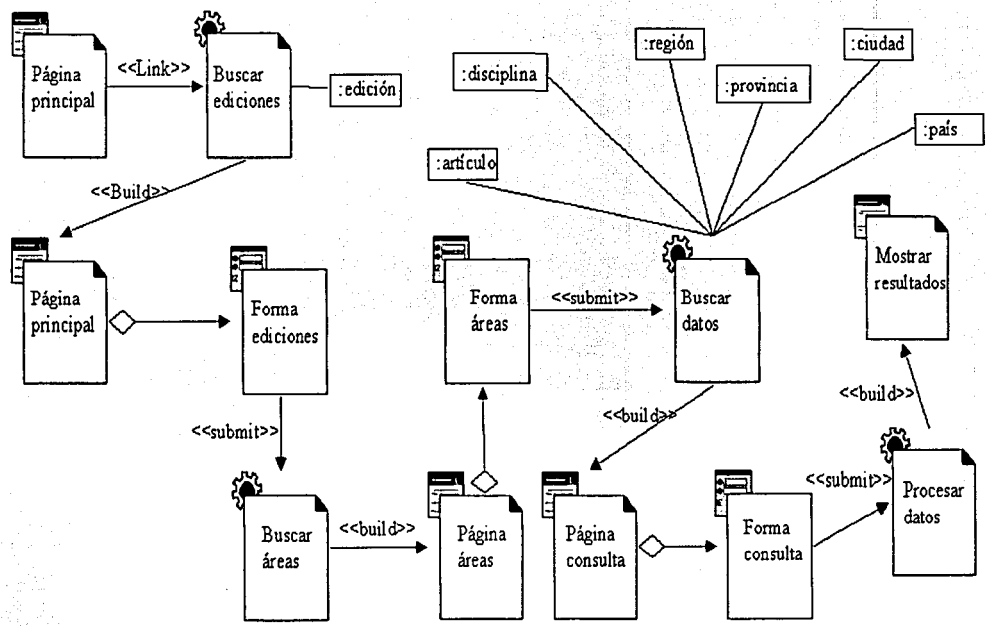
Figura 3.5: Vista lógica de las clases y componentes para mostrar ediciones

denota mediante el uso de una flecha etiquetada con `«link»`. En la misma figura se muestra que las páginas servidor son las encargadas de hacer peticiones a los objetos para obtener la información solicitada y generar una nueva página cliente, la creación de una nueva página se denota mediante una flecha etiquetada con la palabra `«build»`. Las páginas servidor no solamente pueden crear nuevas páginas, a éstas también les es posible redireccionar los resultados a otras páginas servidor encargadas de terminar de realizar los procesos para mostrar la información solicitada por el usuario. La figura 3.7 muestra un fragmento del diagrama presentado en la figura 3.6, en este caso la página servidor “procesar consulta” redirecciona su salida a la página servidor “procesar errores”. Este redireccionamiento se hace en el caso en que el actor no haya seleccionado los elementos necesarios para realizar una consulta o el resultado de la consulta haya sido vacío.

Otro elemento sobresaliente en el diseño de aplicaciones *Web* son las formas. Las formas solamente pueden estar contenidas en páginas cliente, éstas son una agregación de tales páginas. Las formas se comunican con las páginas servidor a través de una flecha etiquetada con `«submit»`. En la figura 3.6 se pueden apreciar las formas.

TESIS CON
FALLA DE CATEGORÍA

Figura 3.6: Vista lógica de las clases relacionadas con la consulta a Indicadores



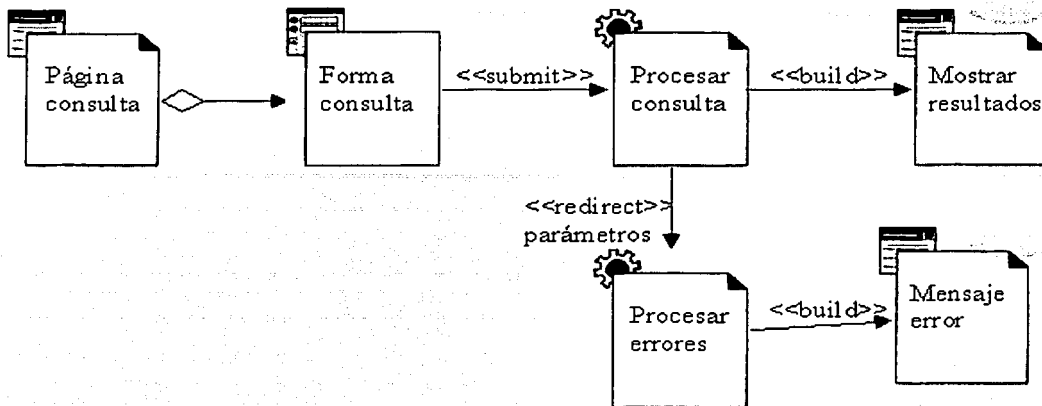


Figura 3.7: Redireccionamiento de páginas

3.4.3. Frames

Los *frames* son elementos que permiten que la ventana del navegador pueda ser dividida en áreas, cada una de estas áreas contiene una página *Web*. Por lo general la división se realiza de tal forma que en una de estas áreas contenga un índice o un menú y que otra área muestre el contenido de los elementos del índice o el menú. La figura 3.8 muestra la vista general de la interfaz de usuario que permite realizar las actividades descritas en el diagrama de casos de uso de actualizar indicadores.

3.4.4. Diseño de la base de datos

La herramienta usada para almacenar los objetos persistentes del sistema es un manejador de bases de datos relacional, por tal motivo se debe hacer un mapeo del modelo de objetos a un esquema de bases de datos relacionales.

Debido a que se utilizó el lenguaje del método unificado (UML) para modelar el sistema, el proceso de normalización de la base de datos no es complejo ya que al

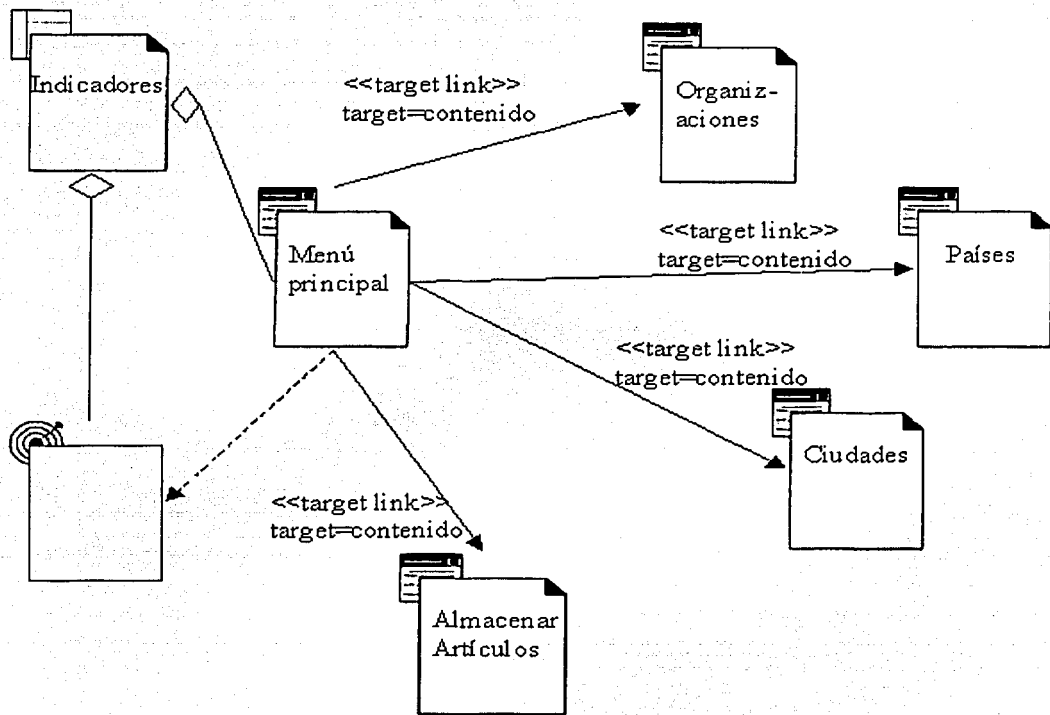


Figura 3.8: Notación utilizada para expresar los frames

TESIS CON
VALIA DE ORIGEN

realizar la transformación de dicho modelo a un esquema relacional este último queda normalizado. En muchos casos lo que debe realizarse es una desnormalización para mejorar el rendimiento (performance) del sistema.

El primer paso que se realiza es transformar las clases en relaciones y los atributos de éstas en atributos de las relaciones obtenidas. Las clases que no pertenezcan a una generalización se les crea una llave primaria; a las subclasses se les agrega una llave foránea que corresponde a la llave primaria de la clase padre. La transformación de tipos UML al esquema relacional se hace de acuerdo a los tipos que definidos en el manejador de bases de datos que se esté utilizando. En el caso de que existan asociaciones binarias 1..*, la asociación es absorbida del lado n. Las asociaciones binarias se representan por una línea que une a dos clases. En UML también es posible modelar asociaciones terciarias y mayores, éstas se representan por un diamante, para transformar este tipo de asociación al esquema relacional el diamante es promovido a una relación; esta nueva relación está formada por las llaves primarias de las clases participantes en la asociación. En los casos en donde la multiplicidad de la asociación es de muchos a muchos se genera una nueva relación cuyos atributos con las llaves primarias de las clases participantes en dicha asociación.

En el presente capítulo se mostró la forma en como es utilizada la extensión UML para modelar aplicaciones con arquitectura cliente *Web* débil. En el próximo capítulo se mostrara el modelo de una aplicación *Web*, con este tipo de arquitectura, que es utilizada como herramienta de apoyo en la obtención de indicadores bibliométricos en ciencia y tecnología.

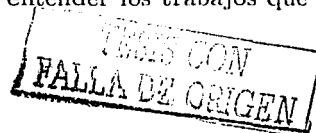
Capítulo 4

Herramienta para obtención de indicadores bibliométricos

En el capítulo 2 se mencionó la importancia que tienen los indicadores de ciencia y tecnología en el ámbito económico, y debido a esto es importante contar con una herramienta que apoye, a los estudiosos en el área, a obtener indicadores de ciencia de una forma más rápida y fácil. En el presente capítulo se mostrará la forma de utilización de la extensión UML para modelar la aplicación *Web* denominada *Software Interface for the Scientific Technological Activity (SIESTA)*. Ésta es una herramienta de apoyo en la obtención de indicadores bibliométricos en ciencia y tecnología.

4.1. Requerimientos funcionales

La aplicación SIESTA nació de la necesidad de un grupo italiano que se encontraba realizando un estudio sobre la producción científica de dicho país. El grupo antes mencionado no tenía conocimientos sólidos con respecto a los indicadores utilizados para hacer este tipo de estudio, así como tampoco de los sistemas desarrollados para tales fines. Debido a esto, el modelado de la aplicación SIESTA inició, por parte de la autora, con una búsqueda de literatura para localizar y entender los trabajos que



habían sido desarrollados previamente sobre este tópico. La fuente de datos que se utilizaría era *Current Contents*.

Después de analizar la literatura y realizar algunas entrevistas con miembros del grupo de trabajo, se obtuvieron los casos de usos y requerimientos del sistema.

A continuación se muestran los requerimientos funcionales del sistema.

Objetivos:

Desarrollar un sistema que pueda ser utilizado como herramienta de apoyo para la obtención de indicadores de ciencia y tecnología en términos de "disciplina", "área geográfica", y "organizaciones" en el contexto italiano. El sistema también deberá ser capaz de determinar la colaboración científica entre Italia y otros países.

En base a los objetivos antes mencionados a continuación se muestra una lista de las características funcionales que el sistema deberá tener.

1. Para apoyar la obtención de los indicadores antes mencionados los datos serán extraídos de la versión para CD-ROM de la base de datos *Current Contents* en formato extendido.

La base de datos *Current Contents* (CC) es producida por el Institute for Scientific Information (ISI), está disponible en siete ediciones multidisciplinarias. Cada edición cubre un grupo de disciplinas interrelacionadas, diseñadas para proveer la más amplia cobertura de revistas. En el presente proyecto se trabajará con las siguientes cuatro ediciones:

- Life Sciences (LS)
- Agriculture, Biology & Environmental Sciences (ABES)
- Physical, Chemical & Earth Sciences (PCES)
- Engineering, Computing & Technology (ECT)

2. Para crear un tesauro con los nombres estandarizados de los países se usará la lista proporcionada por ISO-3166 Maintenance agency (ISO-3166/MA).
3. Los datos que deberán ser extraídos de la base de datos CC son los siguientes:
 - Accession number. El atributo accession number contiene un número único de 9 dígitos, el cual puede ser usado para obtener un documento de la base de datos CC.
 - ISSN. El atributo ISSN contiene el Número serial estándar Internacional (International Standard Serial Number) de la revista en la cual el artículo fue publicado.
 - CC Categories. El atributo Current Contents category (CC) contiene categorías o disciplinas en las cuales CC está dividida.
 - Institution. El atributo Institution contiene la dirección y la institución a la cual los autores del artículo están afiliados. La dirección está compuesta por el nombre de la ciudad y el país, y la institución por varios nombres de institutos separados por comas.
4. El sistema debe contar con una interfaz gráfica para el Web que permita obtener indicadores bibliométricos a nivel región, provincia, ciudad, organización y generales, así como indicadores que muestren la colaboración científica entre Italia y otros países.
5. Procesar los datos obtenidos de la base de datos CC. Cada registro extraído de CC contiene el nombre del país y las instituciones que participaron en la producción del artículo, si el nombre de un país aparece más de una vez solamente se contabilizara una colaboración.
6. Procesar los registros que contengan nombres de ciudades italianas no estandarizados.



7. Estandarizar los nombres de los países extraídos de la base de datos CC y que no se encuentren en el tesoro de países.
8. Crear un tesoro de organizaciones italianas, así como los procesos necesarios para la normalización de los nombres de las instituciones italianas extraídas de la base de datos CC.

Es importante recalcar que debido a la complejidad en la normalización de los datos solamente se normalizarán los nombres de las ciudades e instituciones italianas. Las ciudades, regiones, provincias e instituciones con quien Italia tiene colaboración no serán normalizadas ya que no se cuenta con la información necesaria para la elaboración de tesoros.

Para el presente sistema se detectaron dos diagramas de casos de uso, el primero de ellos es para modelar la forma en que el usuario interactúa con el sistema para obtener los datos necesarios para el cálculo de indicadores de ciencia y tecnología antes mencionados. Éste se muestra en la figura 4.1. En el segundo diagrama se modela la funcionalidad que debe tener el sistema para procesar los datos que deberán ser usados como entradas que el usuario utilizará para la obtención de los datos que le ayuden en la obtención de indicadores de ciencia y tecnología.

Descripción del diagrama de casos de uso de consultas mostrado en la figura 4.1:

1. El usuario navega en la página principal del sistema.
2. El sistema muestra una lista de las ediciones que existen para que el usuario seleccione una de ellas.
3. El usuario selecciona una edición.
4. El sistema muestra una página con las áreas geográficas.

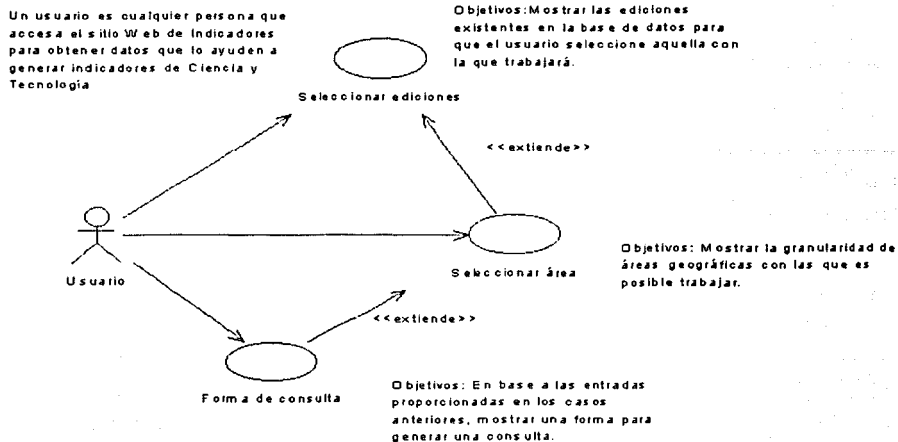


Figura 4.1: Diagrama de casos de uso de consultas

5. El usuario selecciona un área geográfica.
6. El sistema muestra una forma de consulta con todas las disciplinas y años correspondientes a la edición seleccionada, subáreas geográficas, organizaciones y países.
7. El usuario indica los criterios de consulta, es decir, selecciona las disciplinas, años, subáreas geográficas u organizaciones.
 - Alternativas. El usuario selecciona colaboración con otros países
8. El sistema muestra los resultados de la consulta solicitada por el usuario.
9. Condiciones de error:
 - Si el usuario no selecciona elementos en la subárea geográfica, en organización, o en países el sistema envía un mensaje de error.

- Si el usuario selecciona colaboración con países y además selecciona más de un elemento en la subárea geográfica u organizaciones, el sistema envía un mensaje de error.

Las figuras 4.2 y 4.3 muestran los diagramas de secuencia y robustez de este diagrama de casos de usos.

Diagrama de secuencia de Consulta Indicadores

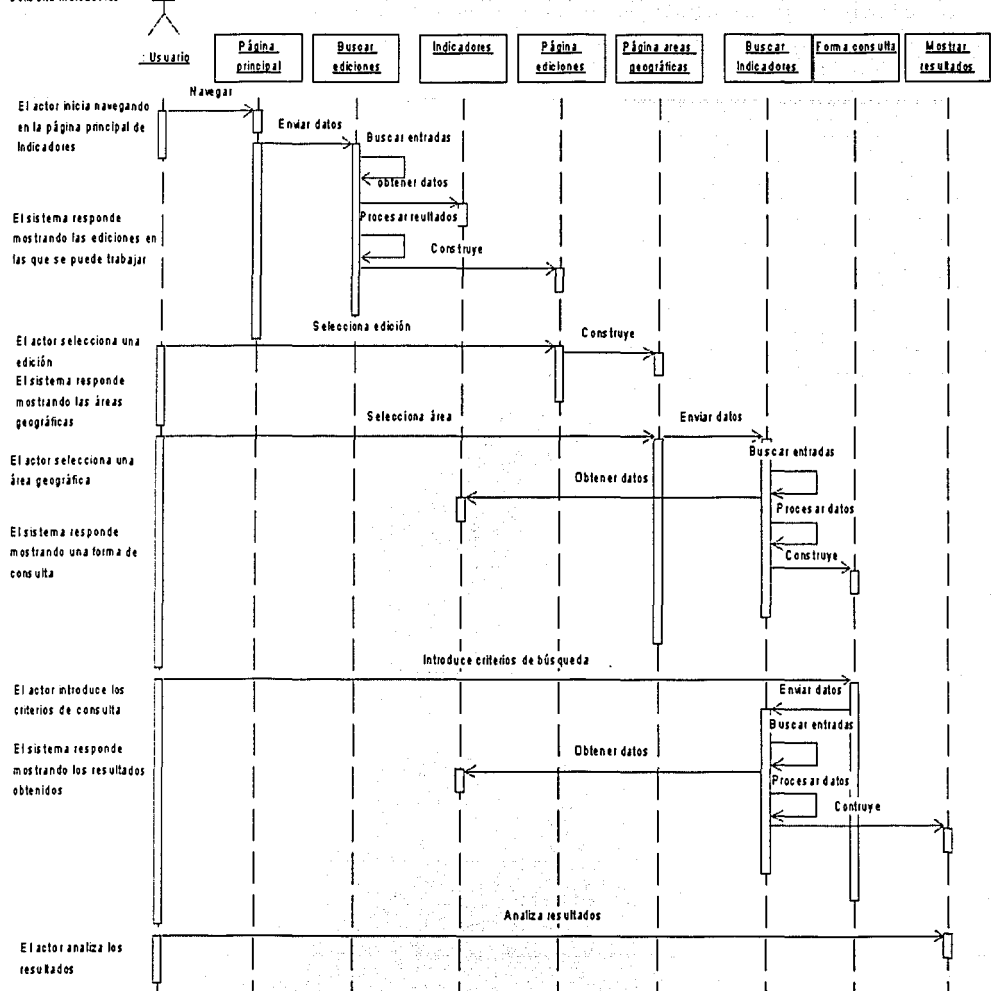
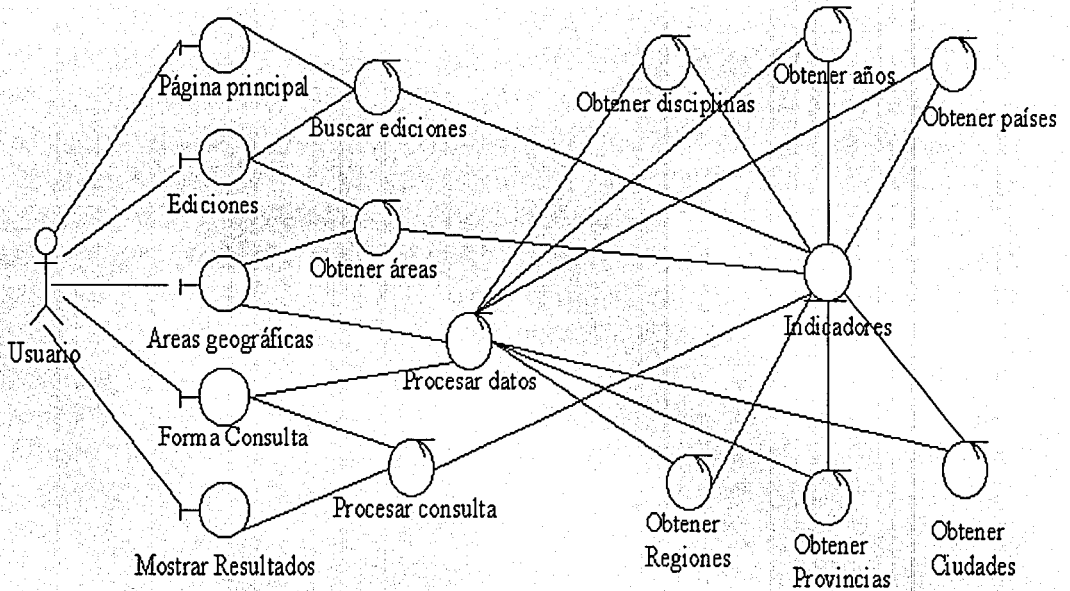


Figura 4.2: Diagrama de secuencia para consultar Indicadores

Figura 4.3: Diagrama de robustez para consultar Indicadores



La figura 4.4 muestra el segundo diagrama, en éste se modela la interacción que existe entre el sistema y el usuario para almacenar y estandarizar los datos que serán usados para el cálculo de indicadores. La descripción del diagrama antes mencionado se muestra a continuación.

Caso de uso: Actualizar Indicadores

En este caso de uso se describe la secuencia para almacenar artículos en la base de datos.

Precondiciones. El usuario debe haber extraído de la base de datos CC y con formato OVID¹, los siguientes datos:

- Accession number
- ISSN
- CC disciplines name/edition
- Institutions.

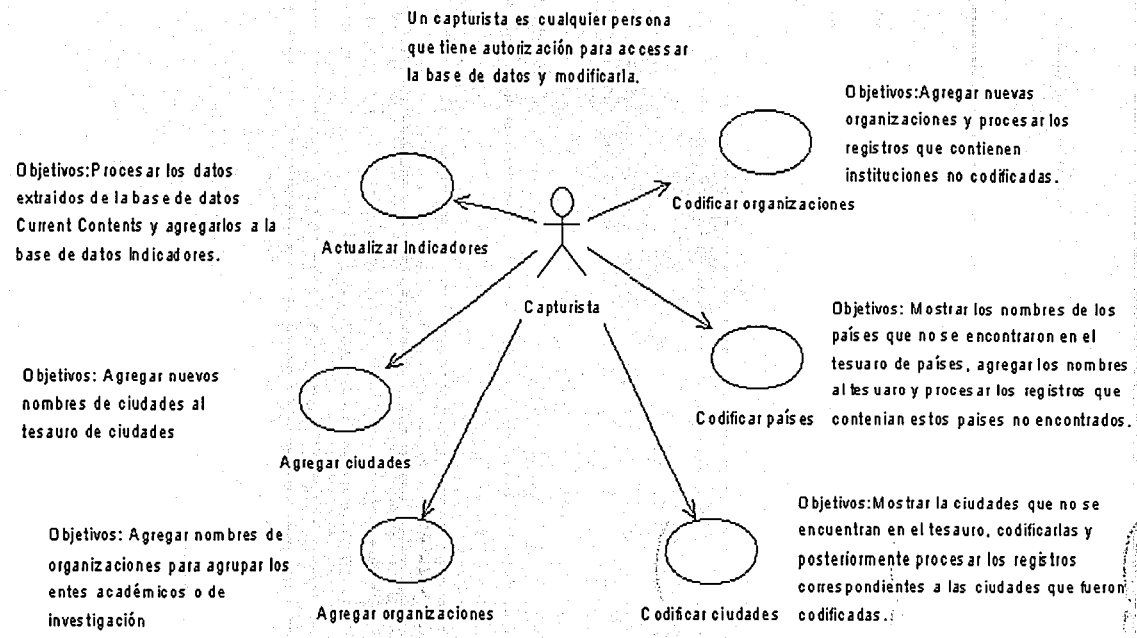
En el cuadro 4.1 se muestra un fragmento de los datos en formato OVID extraídos de la base de datos CC.

1. El capturista navega en la página principal de las actualizaciones.
2. El sistema muestra un menú de actualizaciones.
3. El capturista selecciona la opción de agregar artículos a la base de datos.
4. El sistema muestra una forma para que el capturista seleccione la edición y el año que desea procesar.
5. El capturista selecciona los datos.

¹OVID es la interfaz gráfica que permite hacer consultas a la base de datos CC, esta interfaz tiene cuatro formatos para almacenar la información en archivos. Uno de estos es el OVID, este formato permite una mejor manipulación de los datos

TESIS
ADIC

Figura 4.4: Diagrama de casos de uso para almacenar y estandarizar datos



Cuadro 4.1: Formato OVID de la base de datos CC

< 8 >

Accession Number

142LT-0053

ISSN

0040-6090

CC Categories

Applied Physics/Condensed Matter/Materials Science in Current

Contents(R)/Physical, Chemical & Earth Sciences.

Institution

Reprint available from:

Neri F

Univ Messina, Dipartimento Fis mat & TecnoI Fis Avanzate

Sta Sperone 31,Vill S Agata

I-98166 Messina

Italy

Univ Messina, Dipartimento Fis mat & TecnoI Fis Avanzate

I-98166 Messina

Italy

Univ Messina, INFM

I-98166 Messina

Italy

CNR, Ist Tecn Spettroscopiche

I-98166 Messina

Italy

Instruments ISA Italia

Milan

Italy

6. El sistema busca el archivo y procesa los artículos contenidos en él. Este proceso debe codificar los nombres de los países, y de las ciudades italianas en base a tesauros almacenados previamente. Los artículos que contengan nombres de países o ciudades que no se encuentren en los tesauros deberán ser almacenados como registros erróneos para ser procesados posteriormente. Una vez terminado este proceso, se deberá indicar el número de artículos procesados.
7. Condiciones de error: Si el capturista no escribe el año, o es incorrecto, o si no se tiene el archivo del año indicado, el sistema muestra un mensaje de error.

Las figuras 4.5 y 4.6 muestran los diagramas de secuencia y robustez, respectivamente, del caso de uso de actualizar indicadores.



Diagrama de secuencia de Actualizar Indicadores

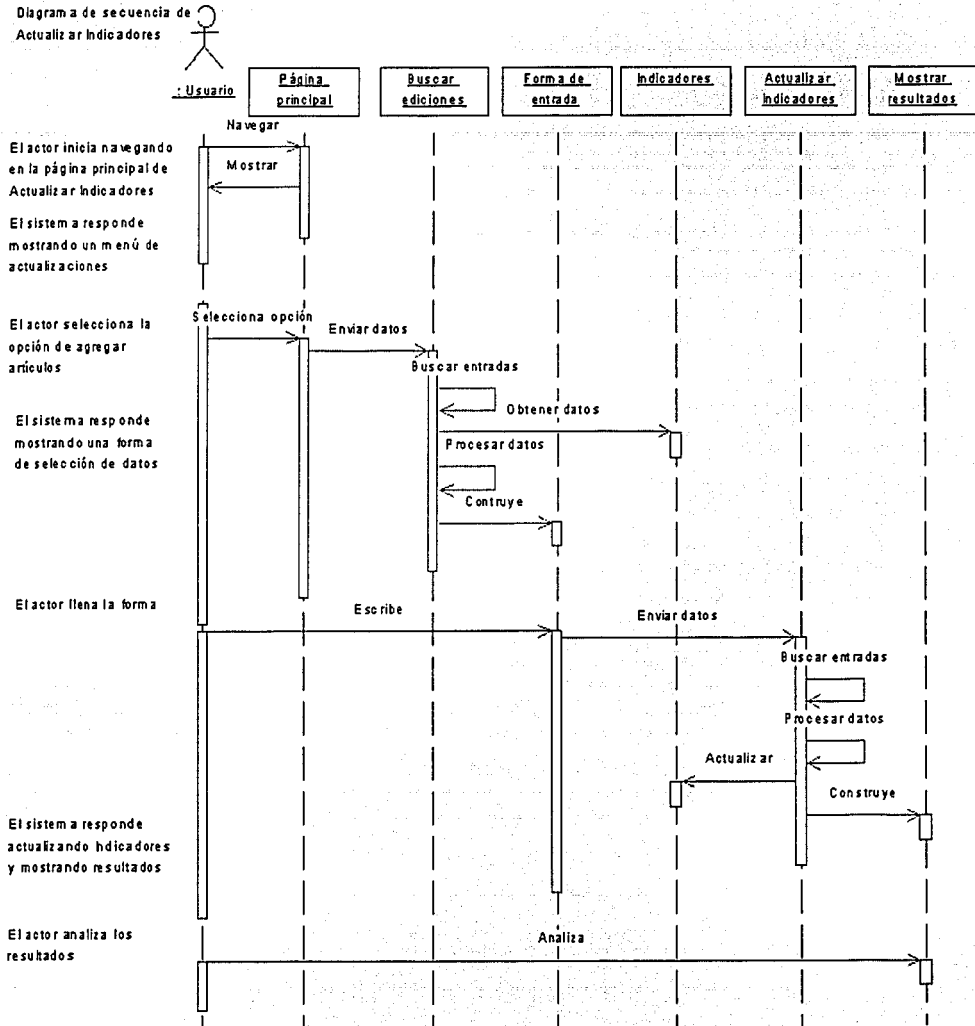


Figura 4.5: Diagrama de secuencia para actualizar Indicadores

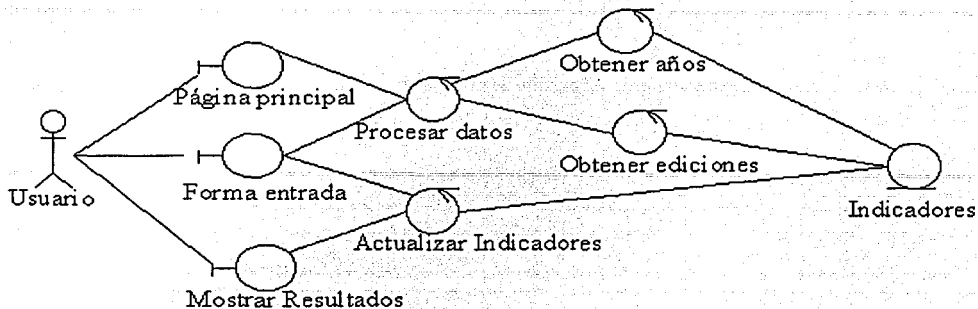
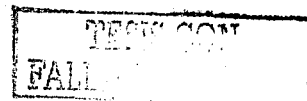


Figura 4.6: Diagrama de robustez para actualizar Indicadores

Caso de uso: Codificar ciudades:

Este caso de uso se utiliza para modelar la funcionalidad del sistema que permite procesar los registros que contenían algunos errores en el nombre de la ciudad.

1. El capturista navega en la página principal de las actualizaciones.
2. El sistema muestra un menú de actualizaciones.
3. El capturista selecciona la opción de codificar ciudades.
4. El sistema muestra una forma para que el capturista seleccione la edición y el año que desea procesar.
5. El capturista selecciona los datos.
6. El sistema muestra una lista con nombres no estandarizados de ciudades, es decir, nombres que no se encontraron en el tesauro de ciudades.
7. El capturista selecciona uno de estos nombres.
8. El sistema muestra una lista estandarizada de nombres de ciudades que inician con la primera letra o contienen el nombre completo de la ciudad seleccionada en el paso anterior.



9. El capturista selecciona la ciudad con nombre estandarizado

10. El sistema registra el nombre no estandarizado de la ciudad como un sinónimo y procesa el registro. Muestra los resultados del proceso.

Las figuras 4.7 y 4.8 muestran los diagramas de secuencia y robustez, respectivamente, del caso de uso de codificar ciudades.

Diagrama de secuencia de Codificar ciudades

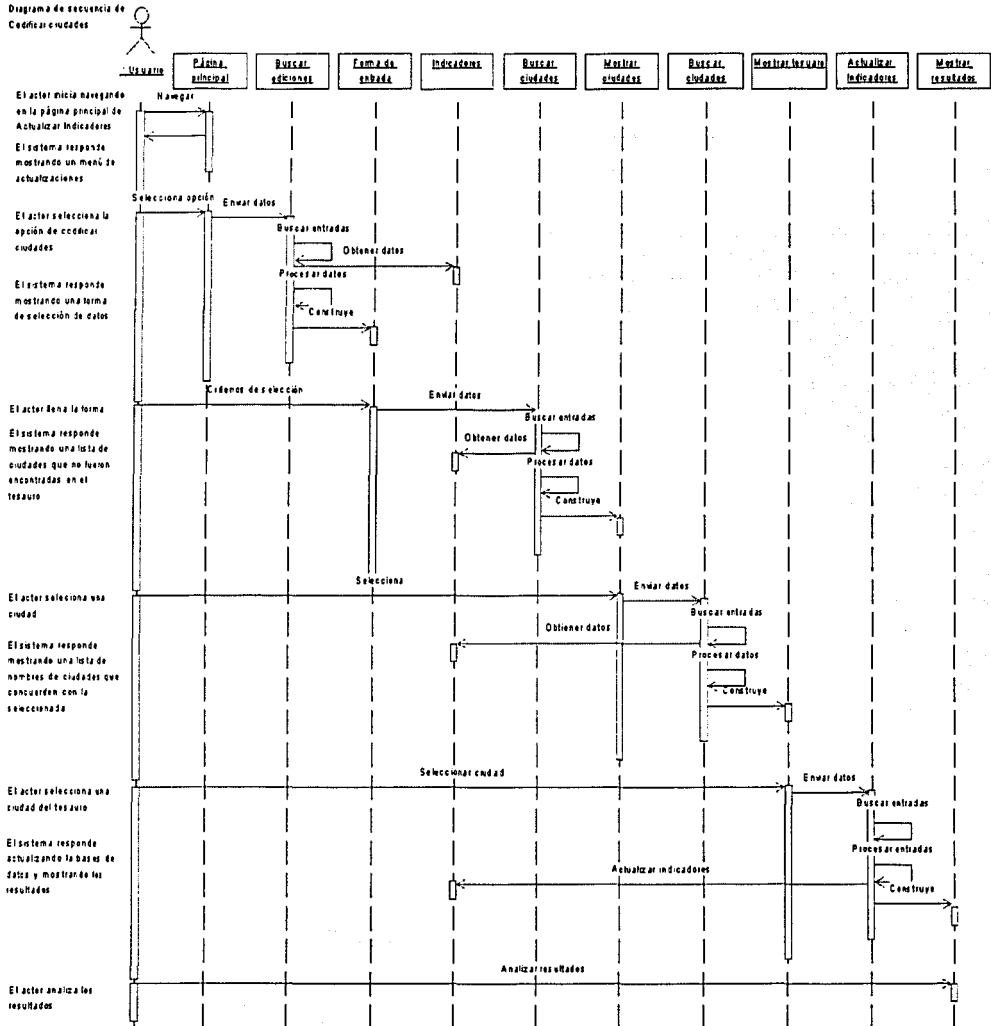
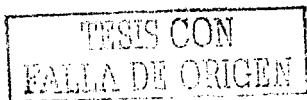


Figura 4.7: Diagrama de secuencia para codificar ciudades



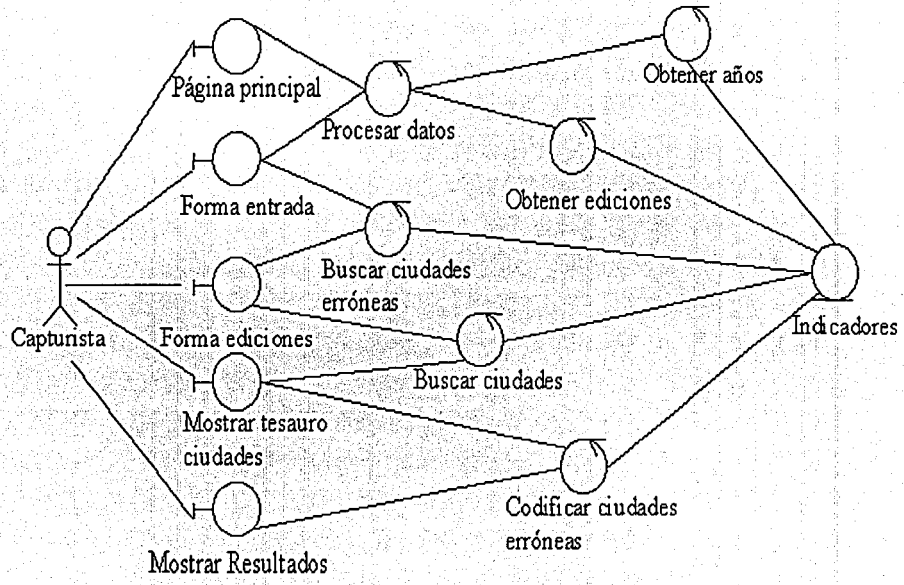


Figura 4.8: Diagrama de robustez para codificar ciudades

Caso de uso: Codificar países

En este caso de uso es para mostrar la funcionalidad que debe realizar el sistema para codificar los registros que contenían algún error en el nombre de los países. Este caso de uso no se describe ya que es similar al de codificar ciudades. Los diagramas de secuencia y robustez de este caso de uso no se muestran ya que son similares a los diagramas de codificar ciudades.

Caso de uso: Codificar organizaciones

La finalidad de este proceso es la de agrupar institutos en organizaciones. Algunos ejemplos de ese tipo de organizaciones son: Universidad, CNR (*Consiglio Nazionale della Ricerca*), ICTP, etc.

1. El capturista navega en la página principal de las actualizaciones.
2. El sistema muestra un menú de actualizaciones.
3. El capturista selecciona la opción de codificar organizaciones.
4. El sistema muestra una forma para que el capturista seleccione la edición, el año, y el atributo que desea procesar.
5. El capturista selecciona los datos.
6. El sistema hace una liga entre los institutos y las organizaciones, y muestra los resultados.

Las figuras 4.9 y 4.10 muestran los diagramas de secuencia y robustez, respectivamente, del caso de uso de codificar organizaciones.



Diagrama de secuencia de
Codificar organizaciones

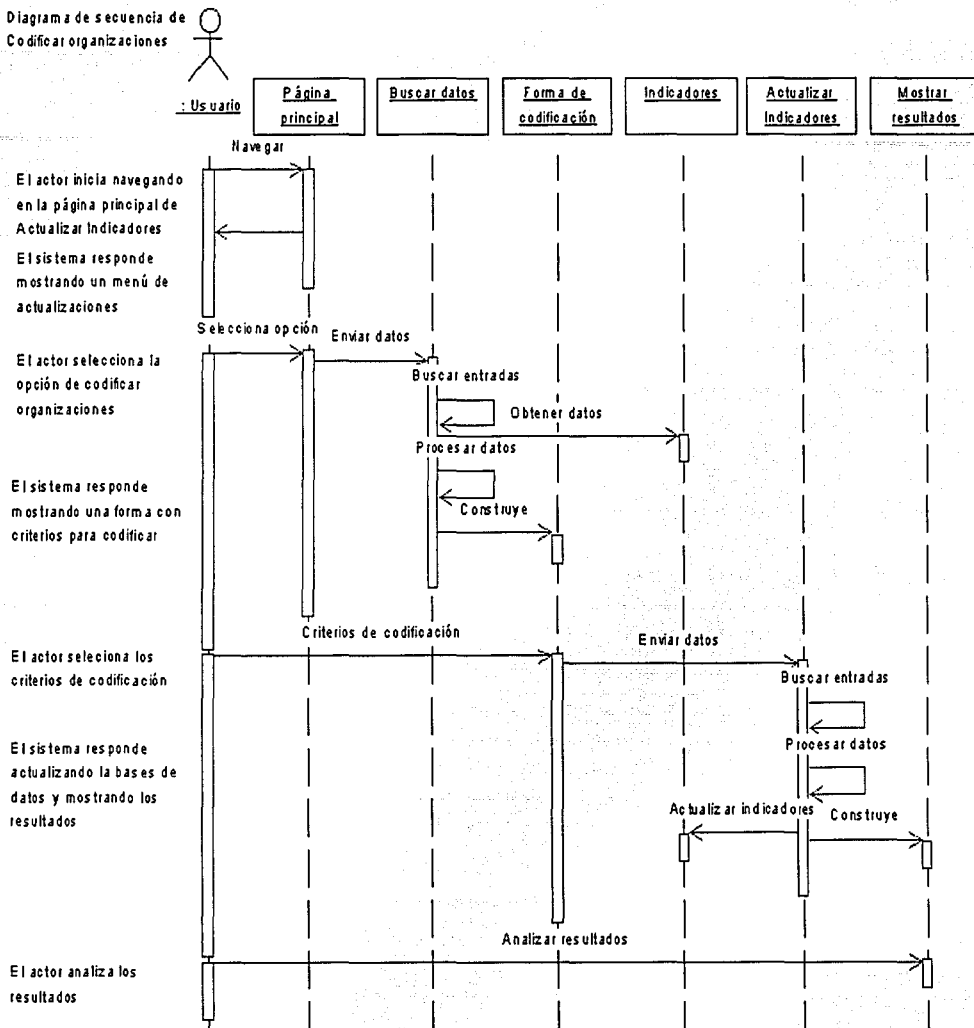


Figura 4.9: Diagrama de secuencia para codificar organizaciones

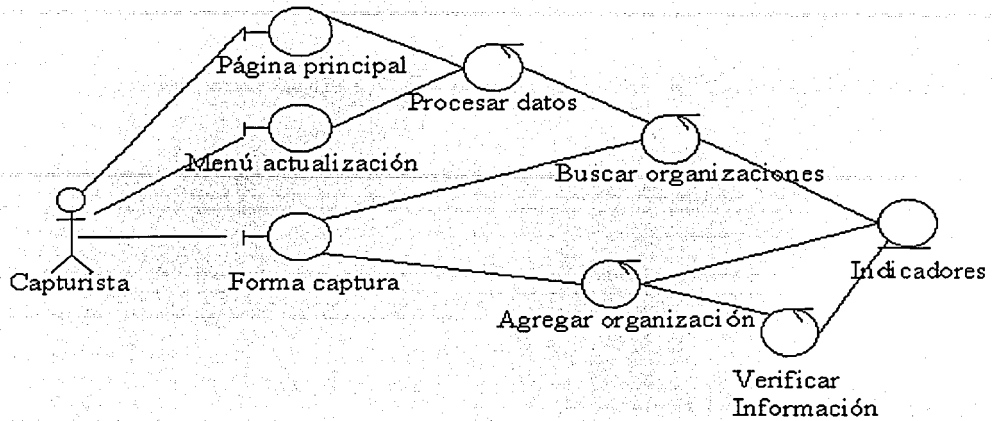


Figura 4.10: Diagrama de robustez para codificar organizaciones

Caso de uso: Agregar ciudades

La finalidad de este caso de uso es mostrar la secuencia para agregar el nombre de una ciudad al tesauro de ciudades.

1. El capturista navega en la página principal de las actualizaciones.
2. El sistema muestra un menú de actualizaciones.
3. El capturista selecciona la opción de agregar una ciudad.
4. El sistema muestra una forma para que el capturista escriba el nombre de la ciudad, el código postal y seleccione la provincia.
5. El capturista escribe los datos solicitados.
6. El sistema verifica los datos, los agrega al tesauro de ciudades y muestra los resultados del proceso.
7. Condiciones de error. Si no escribe el nombre de la ciudad o el código postal, o si el código postal no tiene 5 dígitos muestra un mensaje de error.

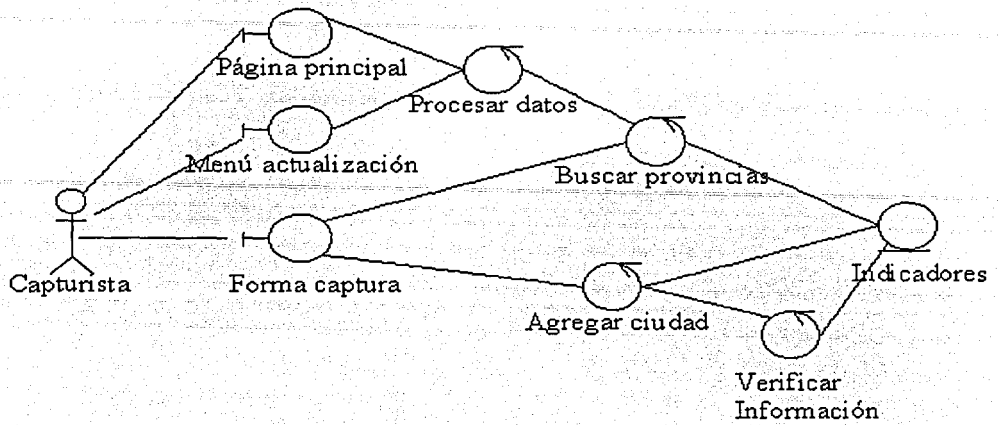


Figura 4.11: Diagrama de robustez para agregar ciudades

Las figuras 4.12 y 4.11 muestran los diagramas de secuencia y robustez, respectivamente, del caso de uso de agregar ciudades.

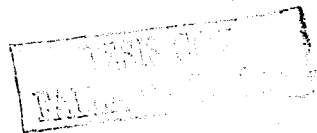


Diagrama de secuencia de agregar ciudades

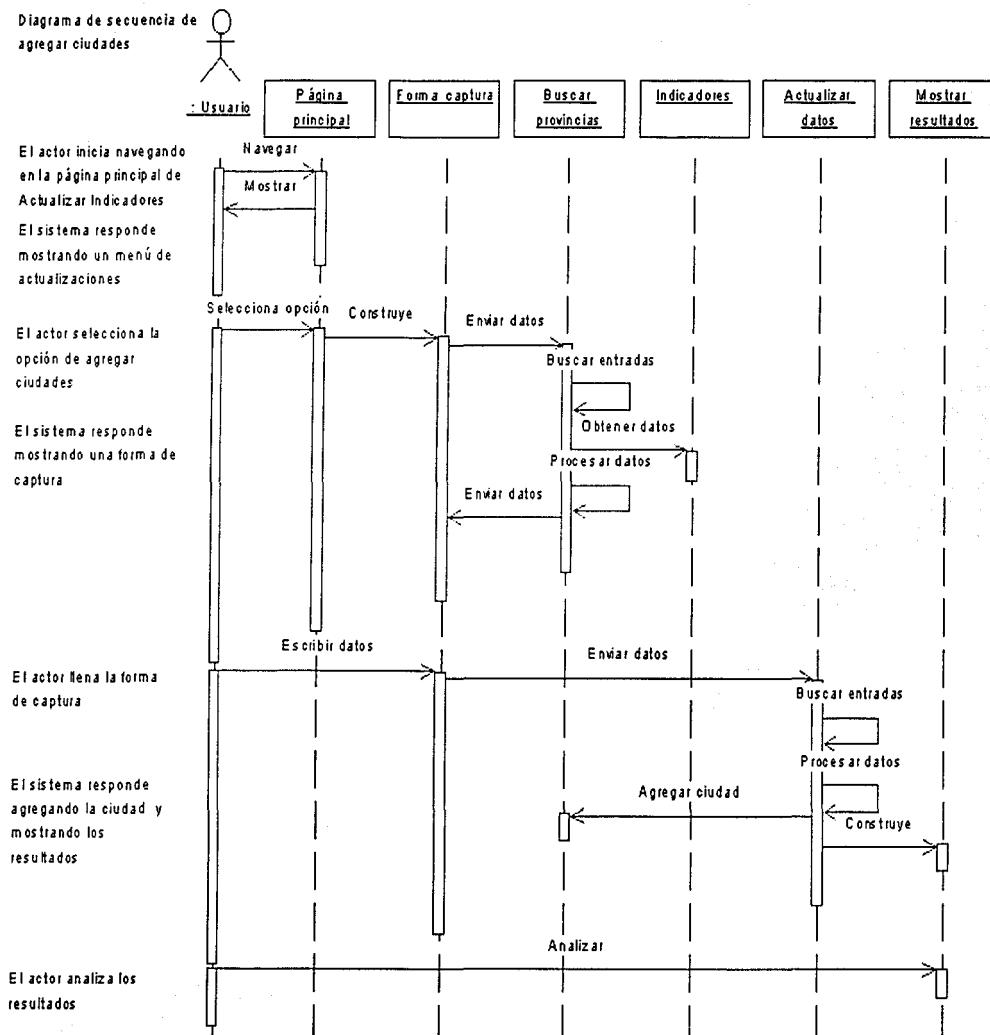


Figura 4.12: Diagrama de secuencia para agregar ciudades

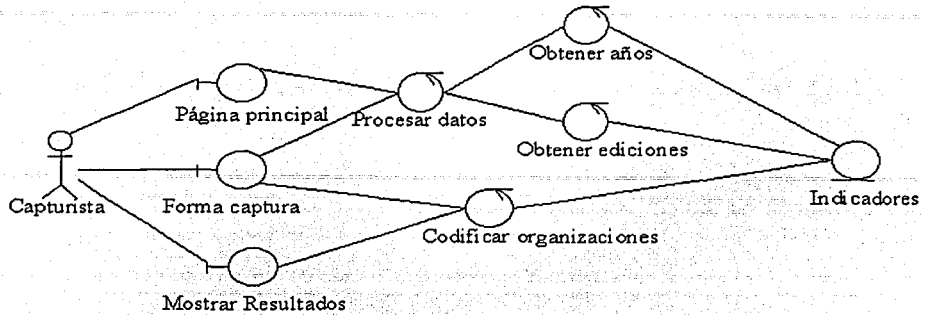


Figura 4.13: Diagrama de robustez para agregar organizaciones

Caso de uso: Agregar organizaciones

1. El capturista navega en la página principal de las actualizaciones.
2. El sistema muestra un menú de actualizaciones.
3. El capturista selecciona la opción de agregar una organización.
4. El sistema muestra una forma para que el capturista escriba las siglas de la organización que desea agregar, también muestra una lista de las organizaciones existentes en la base de datos.
5. El capturista escribe los datos solicitados.
6. El sistema verifica los datos, los agrega al tesoro de organizaciones y muestra los resultados del proceso.
7. Condiciones de error. Si no escribe el nombre de la organización, o ésta ya existe muestra un mensaje de error.

Las figuras 4.13 y 4.14 muestran los diagramas de secuencia y robustez, respectivamente, del caso de uso de agregar organizaciones.

Diagrama de secuencia de agregar organizaciones

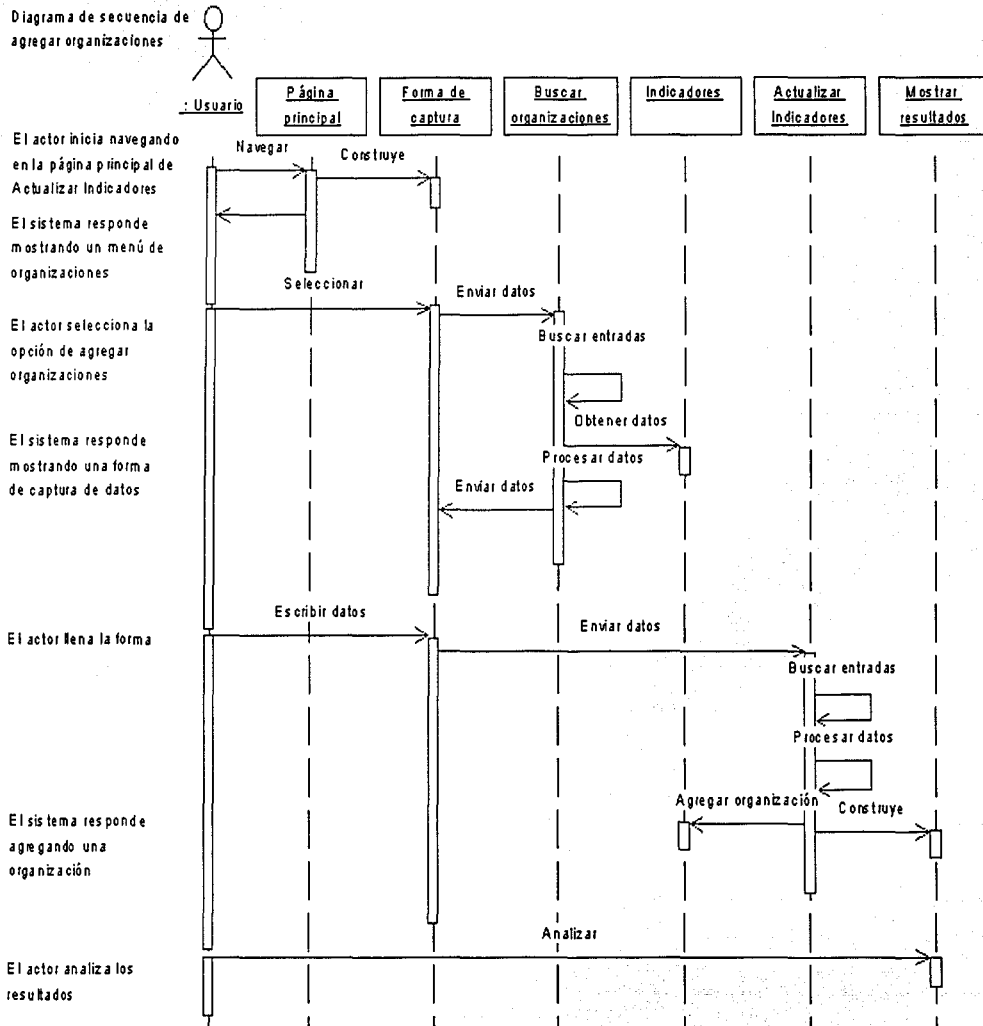
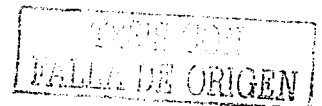


Figura 4.14: Diagrama de secuencia para agregar organizaciones



4.2. Requerimientos no funcionales

Los requerimientos no funcionales para el sistema en desarrollo son los siguientes:

- Usabilidad. Los requerimientos de usabilidad se refieren a los aspectos generales de la interfaz entre el usuario y el sistema. Este tipo de requerimientos están frecuentemente relacionados a estándares de la interfaz de usuario y en el caso de que la aplicación sea una aplicación *Web* estos requerimientos se refieren a la configuración mínima del navegador que será utilizado o a los elementos de HTML que serán utilizados [11]. En esta aplicación, el navegador puede ser Explorer o Netscape, pero deberá soportar el uso de *frames* y *formas*.
- Seguridad. La seguridad que será utilizada en esta aplicación es la proporcionada por el servidor Web (autenticación básica) y la proporcionada por el manejador de bases de datos.
- Hardware. El sistema será implementado en un servidor con las siguientes características:
 - Pentium 4
 - RAM 256 MB
 - 200 MB de espacio en disco para almacenar los archivos de entrada, la base de datos y el sistema.
- Software. El software que debe ser instalado en el servidor antes mencionado es:
 - Sistema operativo RED HAT ver. 7.0 o posterior
 - Servidor Web Apache
 - Manejador de bases de datos MySQL
 - PHP ver. 4.0 o superior.

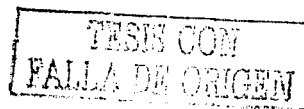
4.3. Modelo de objetos

En base a los casos de uso y con los diagramas de secuencia se detectaron los siguientes objetos: Paper, Organization, Edition, Discipline, Geographic Area, City, CityCC, CountryCC, y SynonymOrganization. El nombre de las clases y atributos se muestran en inglés para homogeneizar el idioma del grupo que participó en algunas de las fases de desarrollo. La figura 4.15 muestra el modelo de objetos. En el diagrama que se obtuvo en la presente sección no se muestran las asociaciones existentes entre los objetos ya que éstas, así como su multiplicidad serán modeladas en la fase de diseño.

4.4. Diccionario de clases

A continuación se describe cada una de las clases del sistema.

1. **Paper.** Documento escrito por un especialista para ser publicado en algún canal formal de difusión. Los atributos que definen a los objetos de esta clase son:
 - **acceNumber.** Contiene un número único de 9 dígitos que identifica a cada artículo.
 - **ISSN.** Contiene el número serial estándar internacional de la revista en la cual el artículo fue publicado.
 - **year.** Contiene el año en el cual el artículo fue ingresado a la base de datos CC.
2. **Institution.** Una institución es una universidad, instituto, centro, etcétera a la cual está afiliado un investigador.
 - **mainInstitution.** Contiene el nombre del primer instituto de la información contenida en el atributo Institutions de la base de datos CC.



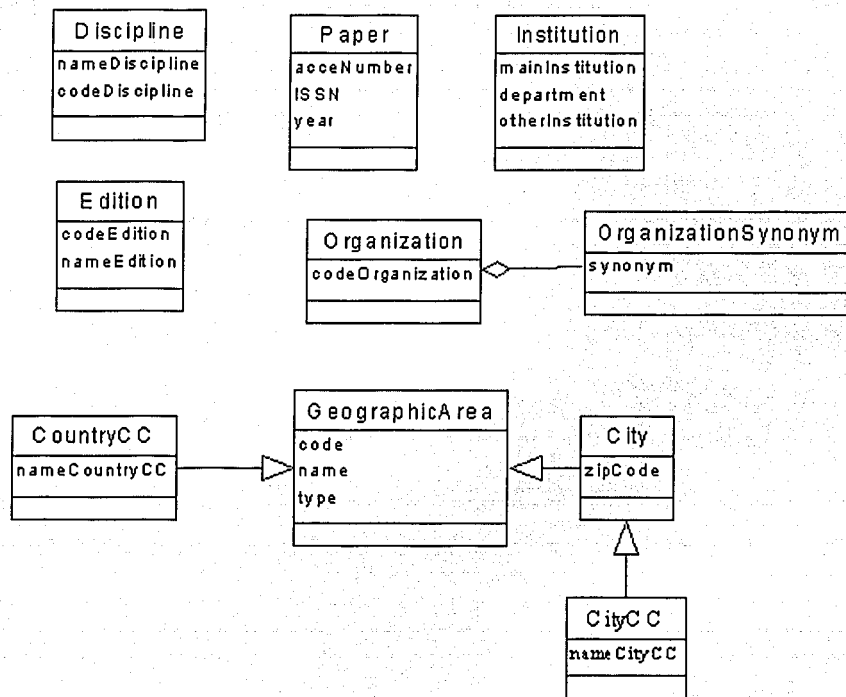


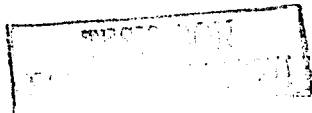
Figura 4.15: Modelo de objetos



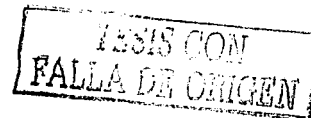
- **department.** Contiene el segundo nombre de instituto de la información contenida en Institutions de la base de datos CC.
 - **others.** Contiene la información contenida en el atributo Institutions de la base de datos CC, desde la tercera posición hasta la antepenúltima.
3. **Organization.** Una organización es un ente que agrupa instituciones académicas del mismo tipo. Por ejemplo, todas las universidades están agrupadas en la organización UNIVERSIDAD.
- **codeOrganization.** Contiene una cadena de caracteres alfabéticos para identificar una organización.
4. **OrganizationSynonym.** Ésta es un ente que indica mediante sinónimos las diferentes instituciones que forman una organización.
- **synonym.** Contiene una cadena de caracteres alfabéticos para identificar una institución.
5. **OrganizationAbbreviation.** Ésta es un ente que indica frases claves mediante las cuales se puede identificar a una institución.
- **abbreviation.** Contiene una cadena de caracteres alfabéticos para identificar una institución.
6. **GeographicArea.** Una área geográfica es una división política. Esta es una clase abstracta.
7. **Country.** Un país es una área geográfica que forma una unidad política independiente.
- **nameCountry.** Contiene una cadena alfabética para indicar el nombre de un país.

TESIS CON
FALLA DE ORIGEN

- codeCountry. Contiene una cadena alfabética de dos elementos para identificar de manera única a un país.
8. **Region.** Es la división política más grande de un país.
- nameRegion. Contiene una cadena para almacenar el nombre de una región.
 - codeRegion. Contiene una cadena alfabética de 6 elementos para identificar de manera única a una región.
9. **Province.** Una provincia es la división política más grande de una región.
- nameProvince. Contiene una cadena para almacenar el nombre de una provincia.
 - codeProvince. Contiene una cadena alfabética de dos elementos para denotar de forma única a una provincia.
10. **City.** Una ciudad es la división política más grande de las provincias.
- nameCity. Nombre de la ciudad.
 - codeCity. Código de la ciudad.
 - zipCode. Contiene un número de 5 dígitos.
11. **Edition.** Es un ente que agrupa un conjunto de disciplinas interrelacionadas.
- codeEdition. El atributo clave contiene una cadena alfabética que identifican de manera única a una edición.
 - nameEdition. Este atributo contiene el nombre completo de la edición.
12. **Discipline.** Las disciplinas son las categorías en las cuales CC divide a cada edición.
- nameDiscipline. Contiene el nombre completo de la disciplina.



- codeDiscipline. Contiene una cadena alfabética que identifican de manera única a una disciplina.
13. **CityCC.** Las ciudadesCC son las ciudades que se encuentran almacenadas en CC pero que no se encuentran en el tesoro de ciudades.
- nameCityCC. Este atributo contiene el nombre de una ciudad extraído de CC.
14. **CountryCC.** Son los países que se encuentran almacenadas en CC pero que no se encuentran en el tesoro de países.
- nameCountryCC. Este atributo contiene el nombre de un país extraído de CC.
15. **CityMistake.** Esta clase es usada para agrupar los artículos en los cuales el nombre de la ciudad no se encontraba en el tesoro de ciudades.
- acceNumber, year. Ver definición en Paper.
 - codeEdition. Ver definición en Edition.
 - nameCity. Es el nombre de una ciudad que no se encuentra ni en City, ni CityCC.
 - mainInstitution, department, other. Ver definición en Institution.
 - status. Es una bandera para indicar si el nombre de la ciudad ya fue agregada al tesoro de ciudades o como sinónimo.
16. **CountryMistake.** Esta clase es usada para agrupar los artículos en los cuales el nombre del país no se encuentra ni en Country o CountryCC.
- acceNumber, year. Ver definición en Paper.
 - nameCountry. Es el nombre del país que no se encuentra almacenado en los tesauros de países.



- `codeEdition`. Ver definición en `Edition`.
- `status`. Es una bandera para indicar si el nombre de la ciudad ya fue agregada en los tesauros de países.

4.5. Diseño

La arquitectura de la aplicación SIESTA es del tipo cliente *Web* débil. Se seleccionó esta arquitectura debido a que se deseaba que los usuarios que consultarían el sistema no tuvieran la necesidad de instalar, ni configurara nada en especial de su navegador. Lo único necesario era que el navegador (cliente) soportara HTML, la aplicación no cuenta con sistemas heredados, ni con sistemas mercantiles, para manejar la persistencia se utiliza el manejador de bases de datos MySQL, el lenguaje de programación es PHP, los *scripts* CGIs escritos con este lenguaje se ejecutan del lado del servidor.

En el modelo de la figura 4.15 se tenía la clase *GeographicArea* con un atributo llamado *type*, con este atributo se hacía referencia a los diferentes tipos de áreas geográficas que son: País, Región, Provincia, Ciudad; para facilitar el mapeo del modelo de objetos a un esquema de bases de datos relacionales, en esta fase de desarrollo se transformó la clase antes mencionada en sus subclases. La figura 4.16 muestra la versión final del modelo de objetos del sistema que se está modelando. Durante el diseño se continúa trabajando con los diagramas de secuencia pero de una forma más específica, además se adiciona el diagrama de componentes.

El presente sistema está compuesto por los componentes Páginas Web y Componentes del servidor. El componente Páginas Web contiene todas las clases y colaboraciones que son implementadas por el servidor Web, el segundo componente contiene los procesos que son ejecutados por el servidor. La figura 4.17 muestra el diagrama de componentes del sistema que se está modelando.

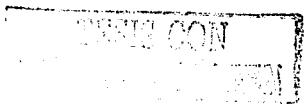
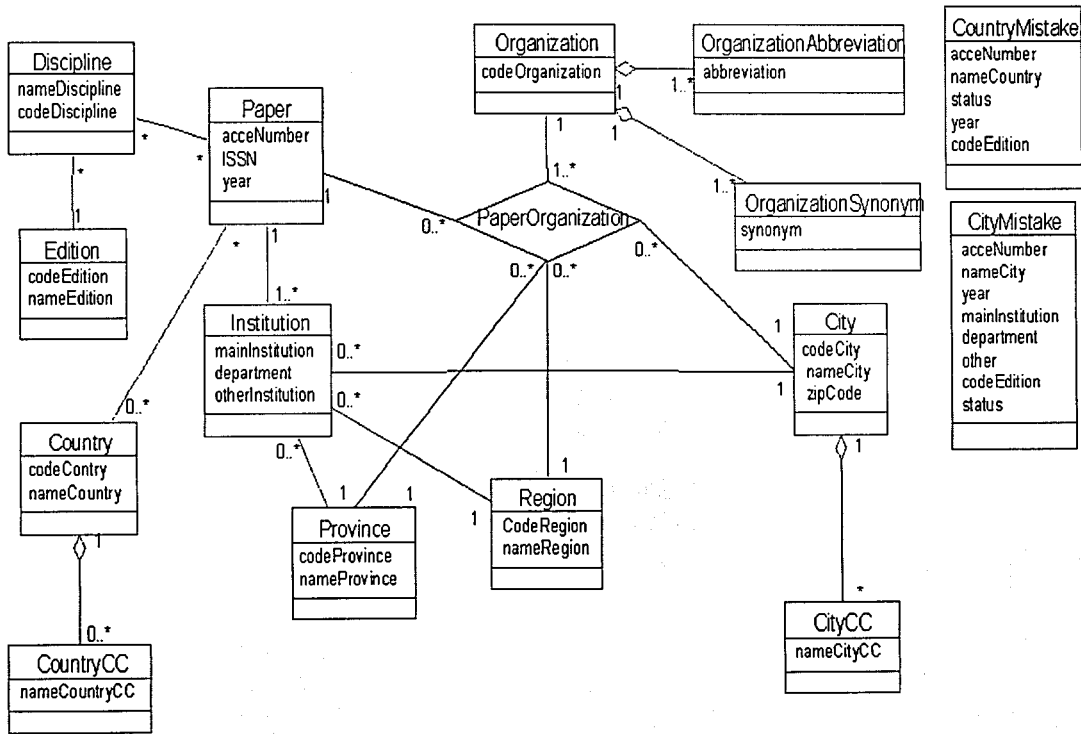


Figura 4.16: Modelo de objetos del diseño



TESIS CON
 FALTA DE ORDEN
 NUMERO DE PAGINA

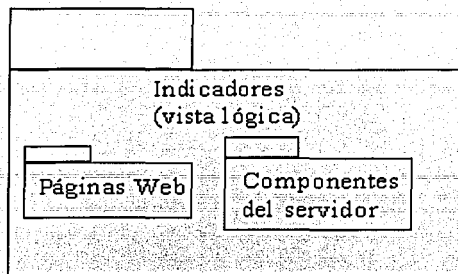


Figura 4.17: Vista lógica de los componentes del sistema

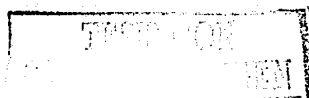
Diagramas de secuencia para el diseño

La figura 4.18 muestra la transformación realizada a las figuras 4.2 y 4.3 que describen la forma en qué se realiza una consulta a la base de datos para obtener los datos necesarios que ayudaran en el cálculo de indicadores.

Este diagrama inicia cuando el actor (usuario) envía un mensaje a la página principal. El mensaje consiste en la solicitud de un servicio a “Buscar ediciones” y es implementado en la Página principal mediante la etiqueta HTML “< a href=ediciones.php>”. Buscar ediciones es una página servidor y es la encargada de realizar los procesos necesarios para presentar la información solicitada por el actor.

Vista lógica del proceso de consulta, panorama de las páginas Web para consultar y actualizar la base de datos, esquema de la base de datos

La figura 4.19 muestra la vista lógica de las clases que intervienen en el proceso de consulta para obtener indicadores bibliométricos. No se muestran los diagramas de la vista lógica de la parte de actualización de la base de datos ya que en su elaboración



101
 TERN CON
 FALTA DE OMBREIN

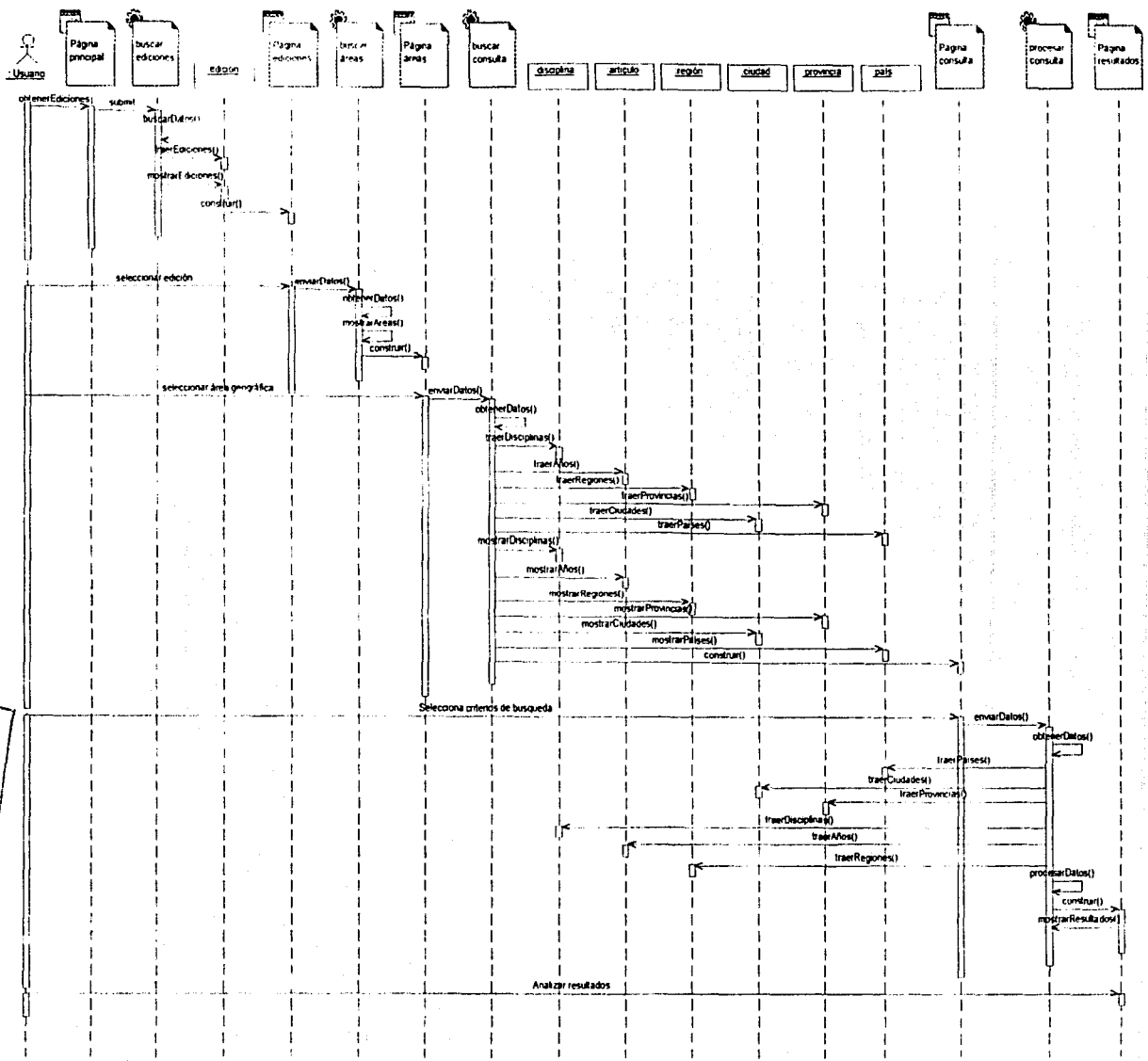


Figura 4.18: Diagrama de secuencia de diseño para procesar consultar Indicadores

intervienen los mismos elementos UML. En la figura 4.20 se muestra un panorama de las páginas *Web* cliente que forman la interfaz de usuario para la obtención de indicadores bibliométricos. Así mismo, en las figuras 4.21 y 4.22, se muestran las páginas cliente que intervienen en la interfaz de usuario para actualizar la base de datos. La figura 4.23 muestra el diseño de la base de datos.

Notación del diagrama de la base de datos

La notación utilizada en el esquema de la figura 4.23 es la siguiente:

- Las relaciones son representadas por rectángulos que contienen en la parte superior los atributos que forman parte de la llave primaria y en la parte inferior el resto de los atributos.
- La interrelación modelada mediante una línea indica que la llave foránea forma parte de la llave primaria.
- Una línea punteada indica que la llave foránea solamente es un atributo más de la relación que la absorbió.

No obstante que en el presente trabajo solo se muestran algunos de los diagramas del modelo del sistema, éste se desarrolló completamente y es utilizado por personal de la Biblioteca del *Consiglio Nazionale della Ricerca* de Boloña, Italia.

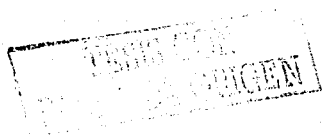


Figura 4.19: Vista lógica de las clases relacionadas con la consulta a Indicadores

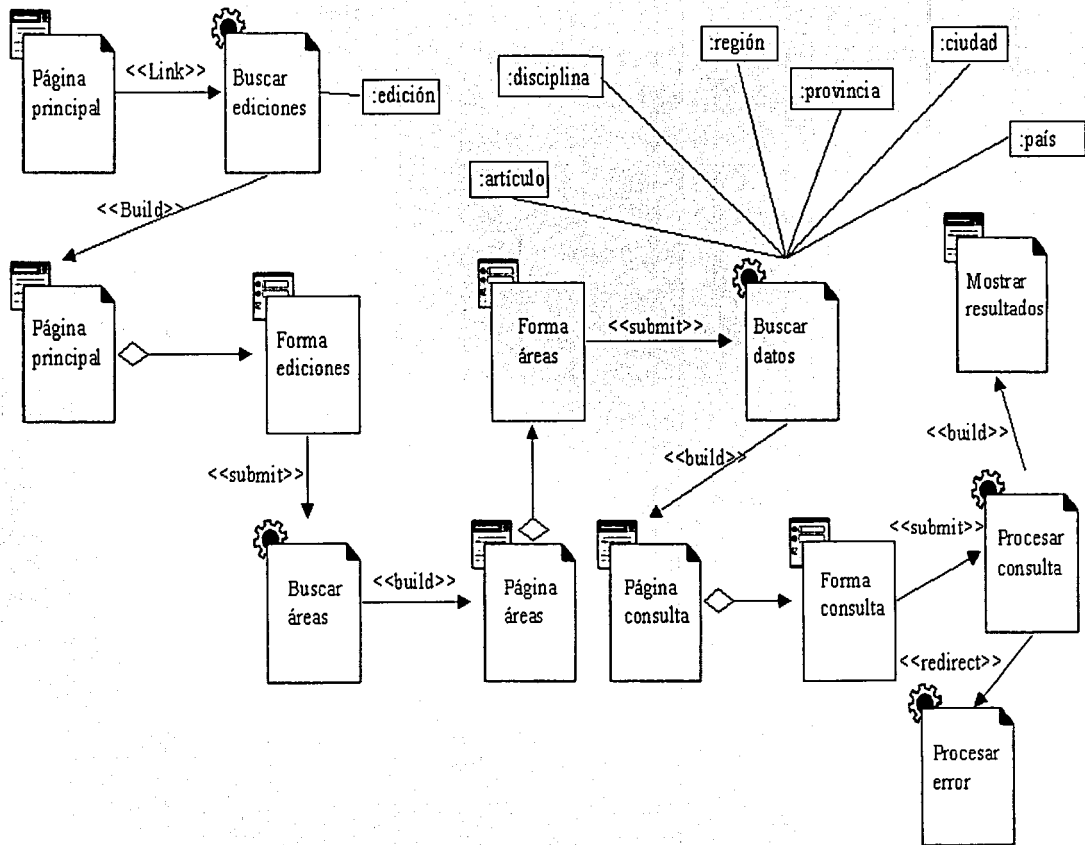


Figura 4.20: Vista general de la interfaz de usuario para consultar a indicadores

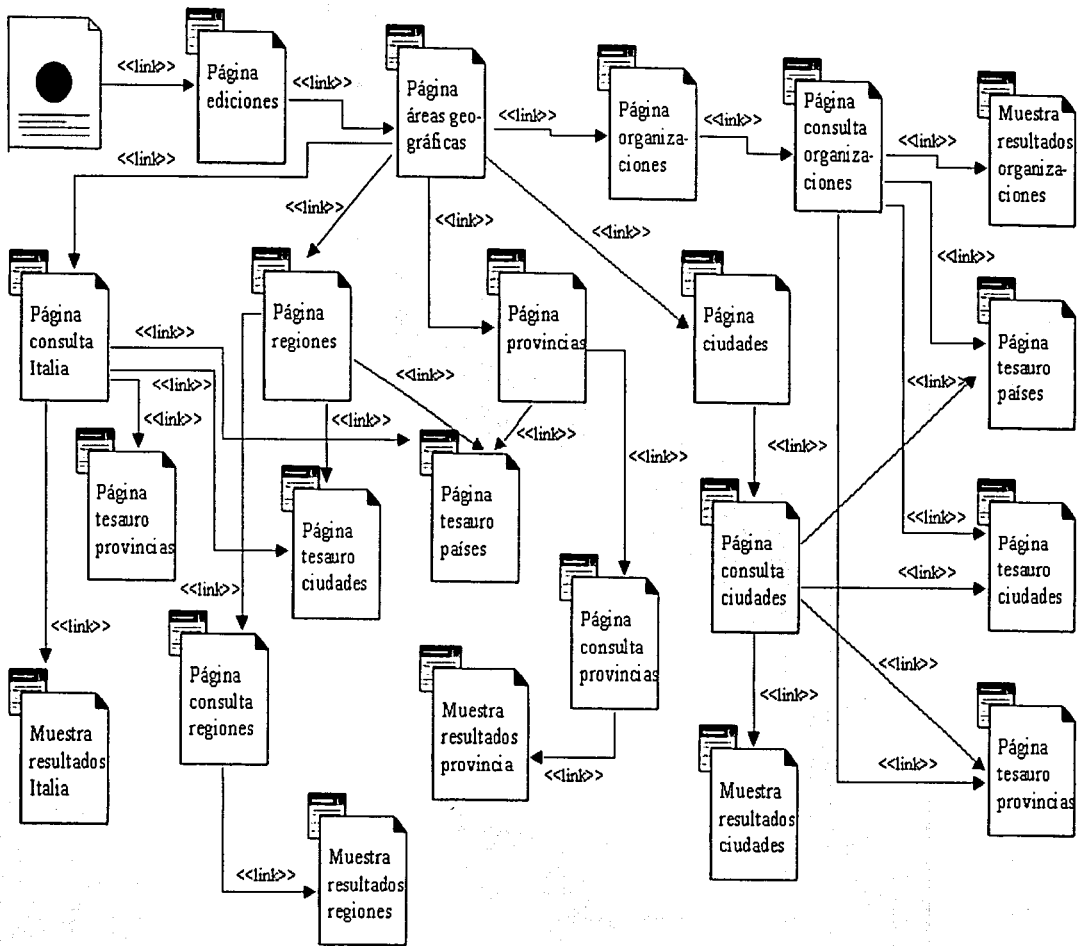


Figura 4.21: Vista general de la interfaz de usuario para actualizar la bases de datos indicadores

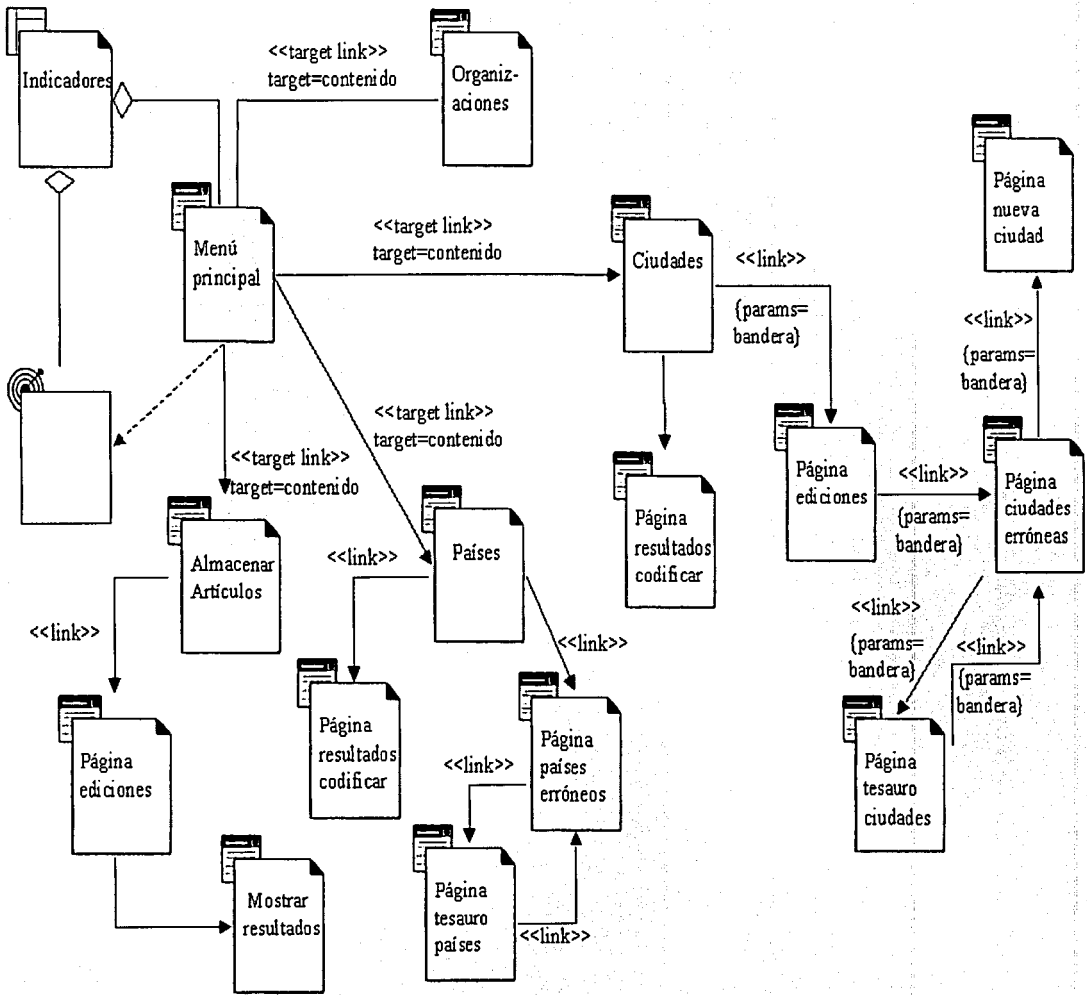
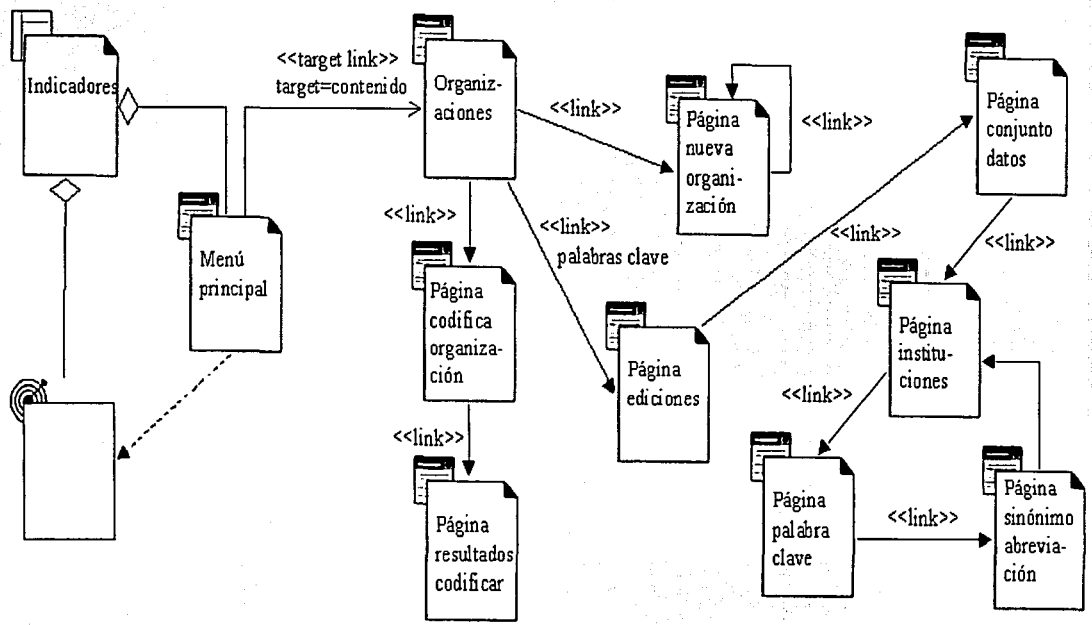


Figura 4.22: continuación de la figura anterior



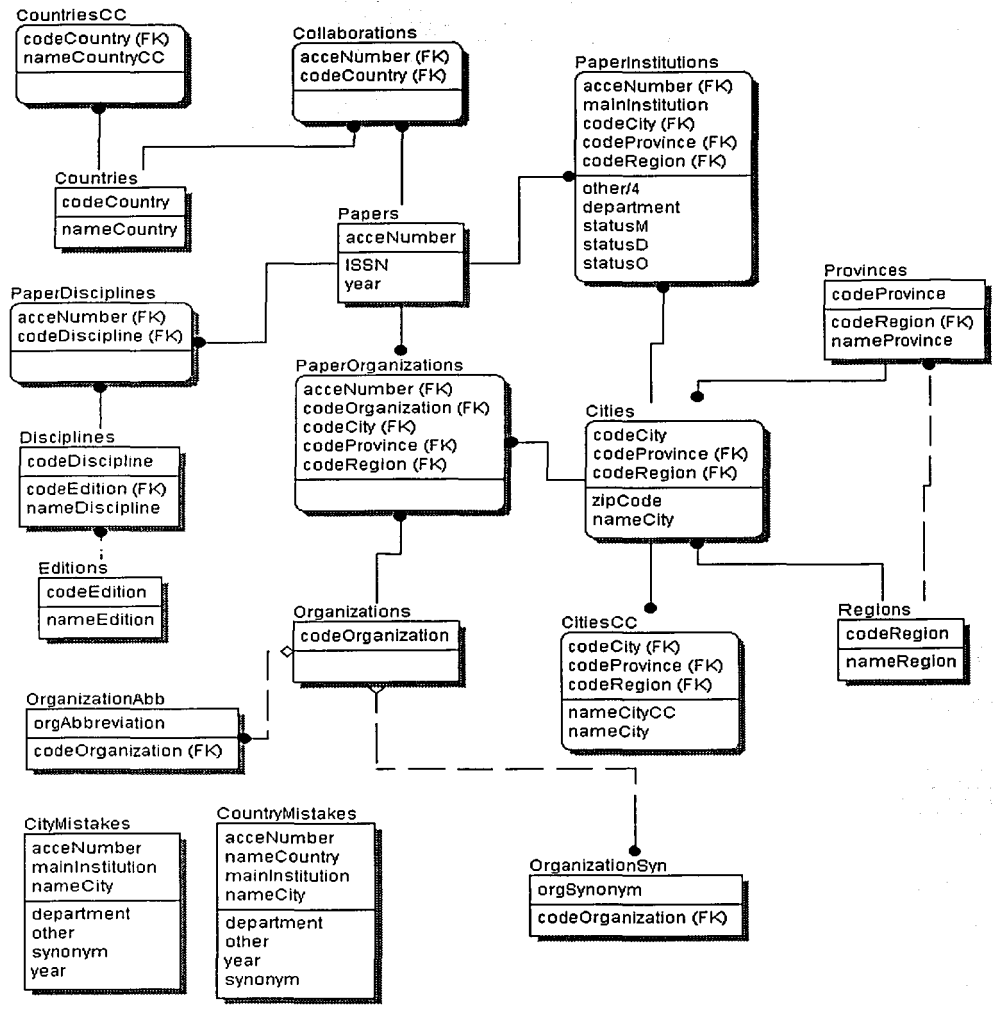


Figura 4.23: Diseño de la base de datos

TESIS CON FALLA DE ORIGEN

RECEIVED
FEBRUARY 1968

Capítulo 5

Resultados, Conclusiones y trabajos futuros

5.1. Resultados

La aplicación SIESTA permite determinar la producción científica en Italia por región, ciudad, año, disciplina, y organización. También se puede conocer la colaboración científica entre Italia con otros países. El personal de la biblioteca del CNR de Boloña realizó pruebas para verificar la validez de los resultados obtenidos con la interfaz de consulta proporcionada por el sistema. Los mismos determinaron que en el peor de los casos el sistema mostraba un error del 3%, pero aun así era mejor que los resultados obtenidos en la forma que lo venían haciendo.

Los datos extraídos de la base de datos CC contienen una multitud de errores debido a que la información no está estandarizada. La resolución de este problema, aunado a la ausencia de reglas para ser aplicadas a la identificación y agrupación de nombres de ciudades y organización consumió mucho tiempo durante el desarrollo de software. Durante el desarrollo se formularon reglas para ser aplicadas en el proceso de inserción de artículos y agrupación de ciudades, países, y organizaciones. Debido a que en el

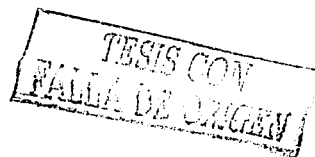
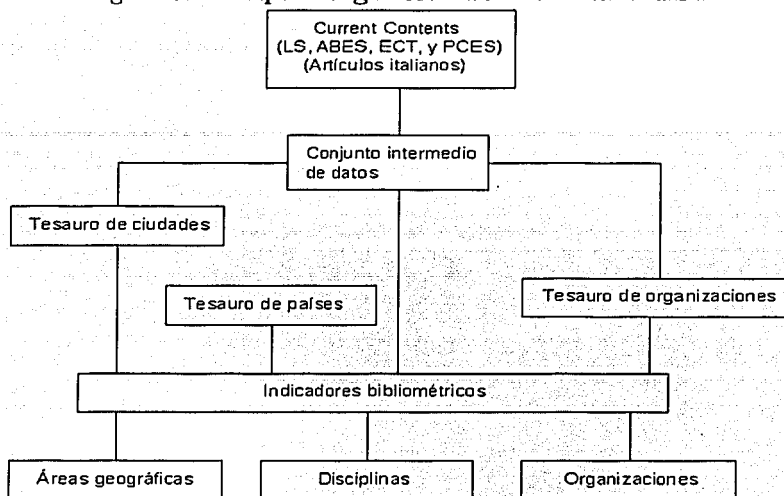


Figura 5.1: Esquema general del sistema SIESTA



sistema SIESTA era de gran importancia la verificación y validación de los datos, no fue posible desarrollar módulos de seguridad y se utilizó la autenticación de usuario proporcionada por el servidor *Web*.

No obstante que el sistema se desarrolló para que fuese consultado por cualquier usuario interesado en conocer la producción científica italiana, éste solo puede ser utilizado por los miembros que participaron en el desarrollo. Éstos, al final del desarrollo del proyecto, determinaron que por políticas del CNR no era posible dejarlo abierto a todos los usuarios.

La figura 5.1 muestra un panorama general del proceso utilizado por el sistema para la obtención de indicadores bibliométricos. Se extraen los artículos italianos correspondientes a cada una de las cuatro ediciones, se crea un conjunto intermedio de datos. De este conjunto se extraen los datos de los artículos y de las instituciones

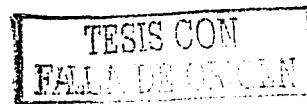
participantes, se generan los tesauros de ciudades, países, y organizaciones. Con los tesauros y los datos de los artículos se obtienen indicadores bibliométricos por número y distribución de las publicaciones (áreas geográficas, disciplinas, y organizaciones).

5.2. Conclusiones

En el modelado de SIESTA se utilizaron elementos proporcionados por una extensión de UML para modelar aplicaciones *Web*. La extensión propuesta por Conallen [11] para aplicaciones *Web* permite modelar elementos presentes en este tipo de aplicaciones. Éstas pueden ser modeladas sin utilizar dicha extensión, pero los diagramas obtenidos no son tan claros, ya que en ninguno de ellos es posible integrar las páginas *Web*, los componentes y los objetos del dominio del problema.

Aunque el lenguaje de programación usado para la codificación no es orientado a objetos, UML es una herramienta de gran utilidad en el modelado de sistemas que al final resultan ser codificados como sistemas no orientados a objetos. También es de gran utilidad para la obtención del esquema conceptual de la base de datos cuando el sistema manejador no es orientado a objetos.

Se menciona que SIESTA es utilizado para la obtención de indicadores bibliométricos en ciencia y tecnología italianos, pero éste puede ser utilizado para conocer indicadores de México si se le agregan los módulos necesarios para procesar los datos de otras fuentes de información ya que la fuente utilizada para el caso de Italia proporciona una cobertura mínima de la producción científica mexicana. Sin embargo, los datos obtenidos sin agregar ningún módulo, tendrían gran utilidad y valor para los tomadores de decisiones en política científica mexicana ya que corresponden a la producción nacional más visible a nivel internacional, la llamada *mainstream*. Otra

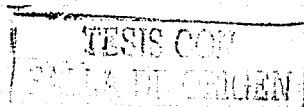


de las ventajas que proporciona SIESTA es que es uno de los primeros desarrollos de este tipo que se hacen para ser consultados a través de la *Web*.

5.3. Trabajos futuros

Como se mencionó en la sección anterior, a SIESTA se le pueden agregar módulos para obtener indicadores bibliométricos en ciencia y tecnología mexicanos. Éstos módulos son aquellos que ayudarían en el proceso de estandarización de los datos extraídos de fuentes de datos que tengan cobertura de la producción científica nacional.

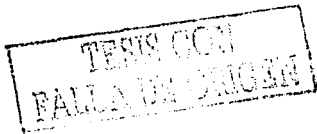
A SIESTA se le pueden agregar nuevos módulos para obtener otros de los indicadores mencionados en el capítulo dos como por ejemplo, el factor de impacto de las revistas el cual es una herramienta muy útil para determinar el valor de las publicaciones, en términos de la frecuencia con la cual sus artículos están citados en trabajos posteriores.



Bibliografía

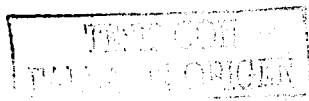
- [1] Internet protocols; [en línea]. <http://www.odur.let.rug.nl/bert/PROSA/rep-protocols.html>, [Consulta: Marzo de 2003].
- [2] Alvarez, MA. Qué es ActiveX; [en línea]. <http://www.desarrolloweb.com/articulos/993.php>, [Consulta: Marzo de 2003].
- [3] Arvanitis, A., Russell, J.M. y Rosas A. Experiences with the National Citation Reports database for measuring national performance: the case of Mexico. *Scientometrics*, 35(2):247-255, 1996.
- [4] Balestri, M.G., Mangiaracina, S. y Nobili, D. Bibliometric S&T indicators to comply with users' needs. *Research Evaluation*, 10(1):5-12, 2001.
- [5] Basu, A. y Vinu Kumar, B.S. International collaboration in Indian scientific papers. *Scientometrics*, 48(3):381-402, 2000.
- [6] Bordons, M. Aspectos metodológicos en la obtención de indicadores bibliométricos. *Cuadernos de Indicios. Indicadores bibliométricos en Iberoamérica. Red de Indicadores de Ciencia y tecnología-RICYT*, (1):17-26, 2001.
- [7] Callaway, D. *Inside Servlets*. Addison-Wesley, Massachusetts, 1999.
- [8] Caravantes, A. (adaptación web). Gopher, Veronica y Wais; [en línea]. <http://www.eumed.net/grumetes/govewa.htm>, [Consulta: Marzo de 2003].
- [9] Cockburn, A. *Writing effective use cases*. Addison-Wesley, New Jersey, 2001.

- [10] Conacyt. Producción científica y tecnológica y su impacto económico; [en línea]. <http://www.conacyt.mx/dccyt/index-frame.html>, [Consulta: Febrero de 2001].
- [11] Conallen, J. *Building web applications with UML*. Addison-Wesley, New Jersey, 2000.
- [12] Consorcio World Wide Web. Document object model; [en línea]. <http://www.w3.org/DOM/>, [Consulta: Marzo de 2003].
- [13] Consorcio World Wide Web. Extensible markup language (XML); [en línea]. <http://www.w3.org/XML/>, [Consulta: Marzo de 2003].
- [14] Cuervo, V. Introducción al VBScript; [en línea]. <http://www.aulambra.com/ver.asp?id=43>, [Consulta: Marzo de 2003].
- [15] Davis, A. M. *Software requirements objects, functions and states*. Prentice Hall, New Jersey, 1993.
- [16] Delgado, A.L. SMIL. Un lenguaje para multimedia en Internet; [en línea]. <http://www.arquired.es/users/aldegado/doc/smil/>, [Consulta: Marzo de 2003].
- [17] El rincón del programador. JSP; [en línea]. <http://www.elrincondelprogramador.com/default.asp?pag=/secciones/jsp/jsp.asp>, [Consulta: Marzo de 2003].
- [18] Fernández, M.T. INDICYT: Indicadores de ciencia y tecnología en red. *Cuadernos de Indicios. Indicadores bibliométricos en Iberoamérica. Red de Indicadores de Ciencia y tecnología-RICYT*, (1):101-109, 2001.
- [19] Fernández, M.T. et al. Relational database: Constructing a relational database for bibliometric analysis. *Research Evaluation*, 3(1):55-62, 1993.
- [20] Fernández, M.T. et al. INDICYT Science and technology indicators in Spain: development of an application for interactive search on the Internet. *Research Evaluation*, 10(2):83-88, 2001.

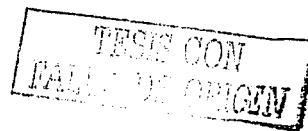


- [21] Fournier, R. *A methodology for client/server and web application development*. Prentice Hall, New Jersey, 1998.
- [22] Franco, A. Introducción a los JavaBeans; [en línea]. <http://www.sc.chu.es/sbweb/fisica/cursoJava/applets/javaBeans/fundamento.htm>, [Consulta: Marzo de 2003].
- [23] Garduño Vera, R. *Estudio sobre los formatos internacionales de intercambio de registros bibliográficos y su función en el control bibliográfico universal*. El autor, Facultad de Filosofía y Letras, México, D.F., 1994.
- [24] Gorbea Portal, S. Principios teóricos y metodológicos de los estudios métricos de la información. *Investigación bibliotecológica*, 8(17):23-32, 1994.
- [25] Gorbea Portal, S. y Castro Thompson, A. Cuib-metric: Un sistema integral para el análisis métrico de la información bibliográfica. *Investigación bibliotecológica*, 8(16):27-31, 1994.
- [26] Greer, T. *Understanding Intranets*. Microsoft Way, Washington, 1998.
- [27] Greg Stein. Welcóm to WebDAV resources; [en línea]. <http://webdav.org>, [Consulta: Marzo de 2003].
- [28] GSyC Grupo de Sistemas y Comunicación. Introducción a WebDAV; [en línea]. <http://gsyc.escet.urjc.es/docencia/asignaturas/tsai/transpas/node11.html>, [Consulta: Marzo de 2003].
- [29] GSyC Grupo de Sistemas y Comunicación. SMIL; [en línea]. <http://www.gsys.escet.urjc.es/docencia/asignaturas/tsai/transpa/node7.htm>, [Consulta: Marzo de 2003].
- [30] Hernández, E. *Desarrollo de los indicadores bibliométricos-cienciométricos de salida (producción e impacto) para el departamento de física del CINVESTAV-*

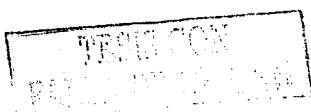
- IPN:1963-1998*. El autor, Escuela Nacional de Biblioteconomía y Archivistomía, México, D.F., 2002.
- [31] Hurd, JM. The transformation of scientific communication: A model for 2020. *Journal of the American Society for Information Science*, 51(14):1279-1283, 2000.
- [32] James, E. Collaborative authoring on the web: introducing WebDAV; [en línea]. <http://www.asis.org/Bulletin/Oct-98/webdav.html>, [Consulta: Marzo de 2003].
- [33] James, E. y Goland, Y. A network protocol for remote collaborative authoring on the web; [en línea]. <http://www.ics.edu/ejw/papers/dav-ecscw.pdf>, [Consulta: Marzo de 2003].
- [34] James, E. y Wiggins, M. WebDAV:IETF standard for collaborative authoring on the web; [en línea]. <http://ftp.ics.uci.edu/pub/ietf/webdav/intro/webdav-intro.pdf>, [Consulta: Marzo de 2003].
- [35] Jansa, K., Lalani, S. y Weakley, S. *Programación para el web*. Mc Graw Hill, México, D.F., 1998.
- [36] Katz, J.S. y Hicks, D. Desktop scientometrics. *Scientometrics*, 38(1):141-153, 1997.
- [37] Katz, J.S. y Hicks, D. Indicators for systems of innovation, a bibliometrics-based approach; [en línea]. *IDEA paper series* 12, Reino Unido: Step Group, <http://www.sussex.ac.uk/Users/sylvank/pub/Idea12.pdf>, [Consulta: Marzo de 2001]. 1998.
- [38] Kulak, D. y Guner, E. *Use cases requirements in context*. Addison-Wesley, New Jersey, 2000.
- [39] Lafuente López, R. y Rosas, A. La publicación electrónica: ¿Un paradigma de organización documental digital? *Investigación bibliotecológica*, 12(25):164-196, 1998.



- [40] Learn the Net.com. Glosario. <http://www.learnthenet.com/spanis/glossary/>, [Consulta: Marzo de 2003].
- [41] Macías-Chapula, CA. Papel de la informetría y la ciencia-metría y su perspectiva nacional e internacional; [en línea]. <http://bvs.sld.cu/revistas/aci/vol9s01/scisus01.pdf>, [Consulta: Abril de 2003].
- [42] Michalas, A. et al. A comparison of multimedia application development platforms towards the object web. *Computer Standard & Interfaces*, 22:13–26, 2000.
- [43] Mönz, S. ActiveX y HTML; [en línea]. <http://www.selfhtml.com.ar/introduccion/tecnologias/activex.htm>, [Consulta: Marzo de 2003].
- [44] Morales, P. MatML; [en línea]. <http://www.tejedoresdelweb.com/307/fo-article-5809.pdf>, [Consulta: Abril de 2003].
- [45] Muller, R.J. *Database design for smarties: Using UML for data modeling*. Morgan Kaufmann Publishers, San Francisco, 1999.
- [46] Pursnani, V. Introducción a la programación de java servlets; [en línea]. <http://www.acm.org.crossroads/espanol/xrds8-2/servletsProgramming.html>, [Consulta: Marzo de 2003].
- [47] Rosenberg, D. y Scott, K. *Use cases driven object modeling with UML a practical approach*. Addison-Wesley, New Jersey, 2001.
- [48] Russell, J. M. y Rousseau, R. Bibliometrics and institutional evaluation. *Encyclopedia of life systems*, UNESCO, París, 2002.
- [49] Sancho, R. Indicadores bibliométricos utilizados en la evaluación de la ciencia y la tecnología. revisión bibliográfica. *Revista Española de Documentación Científica*, 13:842–865, 1990.



- [50] Sancho, R. Indicadores bibliométricos en latinoamérica. *Cuadernos de Indicadores. Indicadores bibliométricos en Iberoamérica. Red de Indicadores de Ciencia y tecnología-RICYT*, (1):9-15, 2001.
- [51] Sun Microsystems, Inc. JavaServer Pages; [en línea]. <http://java.sun.com/products/jsp/>, [Consulta: Marzo de 2003].
- [52] Universidad de Cuenca. JSP; [en línea]. <http://www.rai.ucuenca.edu.ec/proyectos/snib/Tecnologia/JSP/jsp.html>, [Consulta: Marzo de 2003].
- [53] Welling, L. y Thomson, L. *PHP and MySQL web development*. Sams, Indianapolis, 2001.
- [54] Wesley Shrum. View from afar: 'visible' productivity of scientists in the developing world. *Scientometrics*, 40(2):215-235, 1997.
- [55] Zator Systems. Notas sobre Internet; [en línea]. http://www.zator.com/Internet/A9_1.htm, [Consulta: Marzo de 2003].



Apéndice A

Interfaz de Usuario

Las figuras que se presentan a continuación muestran la interfaz de usuario para almacenar artículos en la base de datos, estandarizar y codificar los datos de acuerdo a tesauros. La figura A.1 muestra el menú principal de la interfaz antes mencionada.

La figura A.2 muestra el submenú para procesar registros que tenían algún error en el nombre de la ciudad, dar de alta nombres de ciudades en el tesauro agregar sinónimos.

En la figura A.3 se muestra la ventana en la que se debe seleccionar la edición o el año que se procesará.

La figura A.4 muestra una lista de nombres de ciudades que no fueron encontrados en el tesauro, así como una serie de datos que pueden ayudar en el proceso de estandarización de ésta.

La figura A.5 muestra una lista de posibles ciudades que pueden ser usadas para codificar el nombre de la ciudad seleccionada anteriormente.

La opción *Add new city* es utilizada para agregar nuevos nombres de ciudades al te-

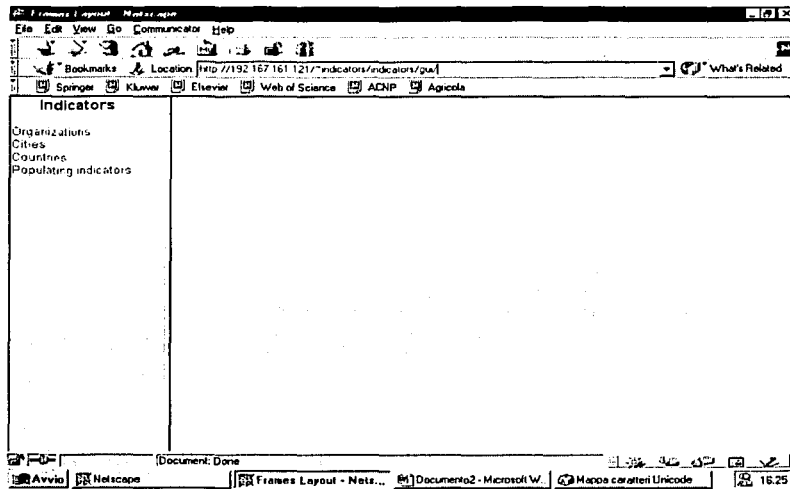


Figura A.1: Menú principal

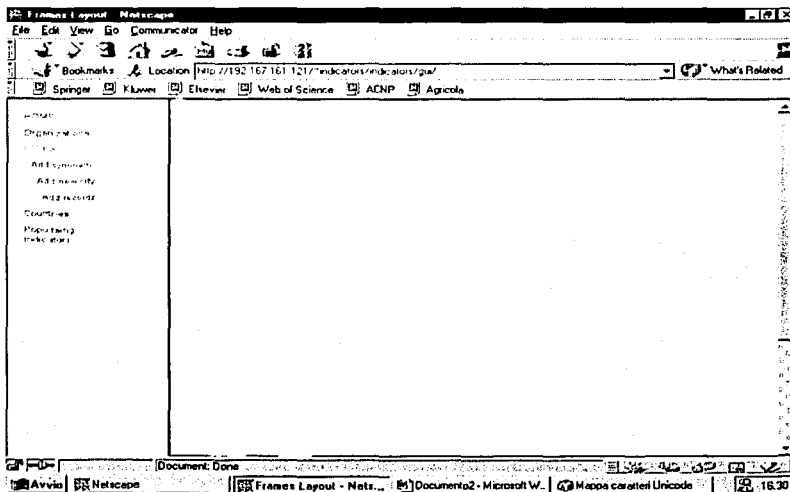
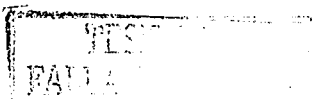


Figura A.2: Submenú ciudades



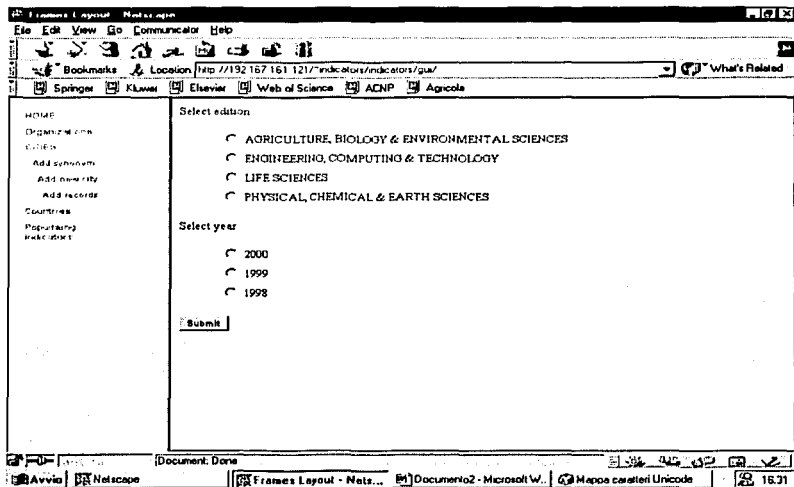


Figura A.3: Seleccionar edición y año

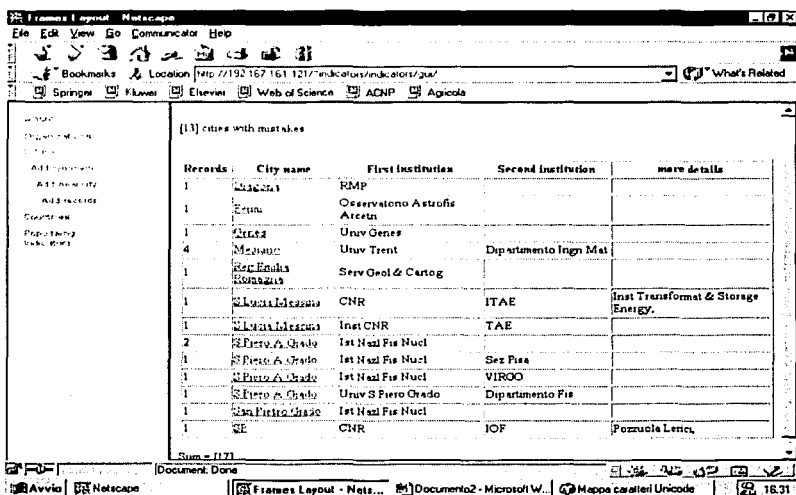
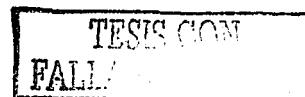


Figura A.4: Listado de nombre de ciudades no estandarizados



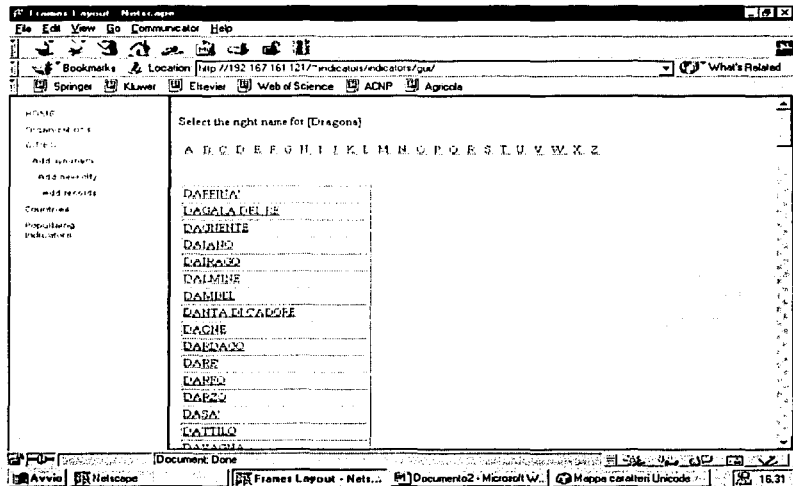


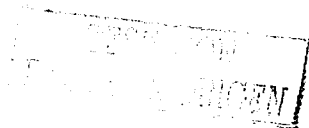
Figura A.5: Seleccionar el nombre estandarizado

sauro. La figura A.6 muestra una lista de nombres de ciudades que no se encuentran en el tesoro y la figura A.7 se tienen los datos que deben ser capturados para insertar una nueva ciudad en él.

Una vez que se han agregado nuevos nombres de ciudades y sinónimos, se procede a utilizar la opción *Add records* para procesar los registros que fueron almacenados como erróneos. Ver figura A.8.

La figura A.9 muestra el submenú de las opciones que son utilizadas en el proceso de agrupar institutos en organizaciones. Algunos ejemplos de este tipo de organizaciones son CNR, INFM, INFN, ICTP, etcétera.

La ventana que se muestra en la figura A.10 es utilizada para agregar nuevas organizaciones.



Frames Layout - Netscape

File Edit View Go Communicator Help

Bookmarks Location <http://192.167.161.121/indicadores/indicadores/ga/> What's Related

Springer Kluwer Elsevier Web of Science ADNP Agricola

13) cities with mistakes

Records	City name	First Institution	Second Institution	more details
1	DRAGONA	RMP		
1	ESUN	Observatorio Astrofis Arcehn		
1	GRITA	Univ Genes		
4	MEGARE	Univ Trent	Departamento Ingh Mat	
1	Rep Enahis Ponzagis	Serv Geol & Cartog		
1	SILOSA MIZAPUIS	CNR	ITAE	Inst Transform & Storage Energy
1	SILOSA MIZAPUIS	Inet CNR	TAE	
2	SI PIERO A ORADO	1st Natl Fis Nucl		
1	SI PIERO A ORADO	1st Natl Fis Nucl	Sez Pisa	
1	SI PIERO A ORADO	1st Natl Fis Nucl	VIROO	
1	SI PIERO A ORADO	Univ SI Piero Orado	Departamento Fis	
1	SI PIERO A ORADO	1st Natl Fis Nucl		
1	SI PIERO A ORADO	CNR	IOF	Pozzuola Lenc

Sum = 171

http://acnp.cib.unibo.it/cgi-bin/stat/it/cnv/tp.htm?m040-ACNP6m040-TEMP&action=ind

Avvio Netscape Frames Layout - Nets... Documento2 - Microsoft W Mapa caratelli Unicode 16.32

Figura A.6: Listado de nombre de ciudades no estandarizadas

Frames Layout - Netscape

File Edit View Go Communicator Help

Bookmarks Location [Go to your personal start page](http://192.167.161.121/indicadores/indicadores/ga/) <http://192.167.161.121/indicadores/indicadores/ga/> What's Related

Springer Kluwer Elsevier Web of Science ADNP Agricola

Diagona

City name

Zip Code

Province

Go to your personal start page

Avvio Netscape Frames Layout - Nets... Documento2 - Microsoft W Mapa caratelli Unicode 16.32

Figura A.7: Datos de la ciudad

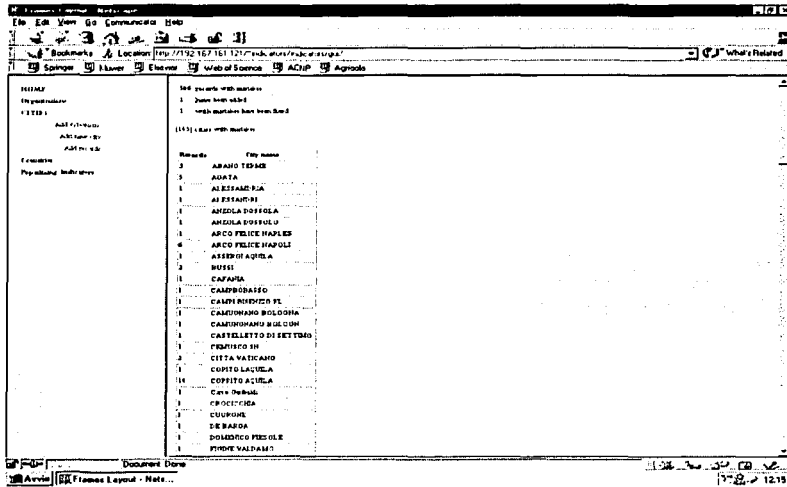


Figura A.8: Procesando registros con nombre de ciudades no estandarizadas

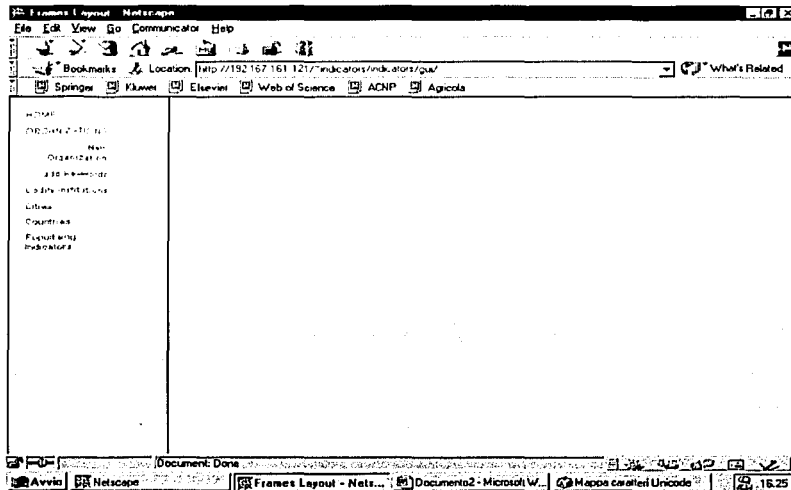


Figura A.9: Submenú organizaciones

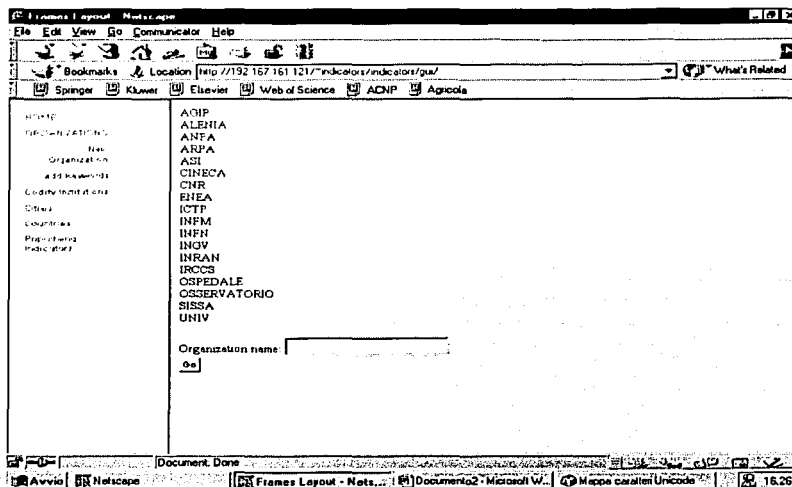


Figura A.10: Agregar una nueva organización

Las figuras A.11, A.12, A.13, A.14 y A.15 muestran la secuencia que se sigue para dar de alta sinónimos o abreviaciones que pueden ser usados para realizar la agrupación de institutos en organizaciones.

Una vez dadas de alta nuevas organizaciones, sinónimos y abreviaciones, la opción *Codify institutions* es utilizada para realizar el agrupamiento de institutos en organizaciones. Ver figura A.16.

TESIS CON
FALLA DE ORIGEN

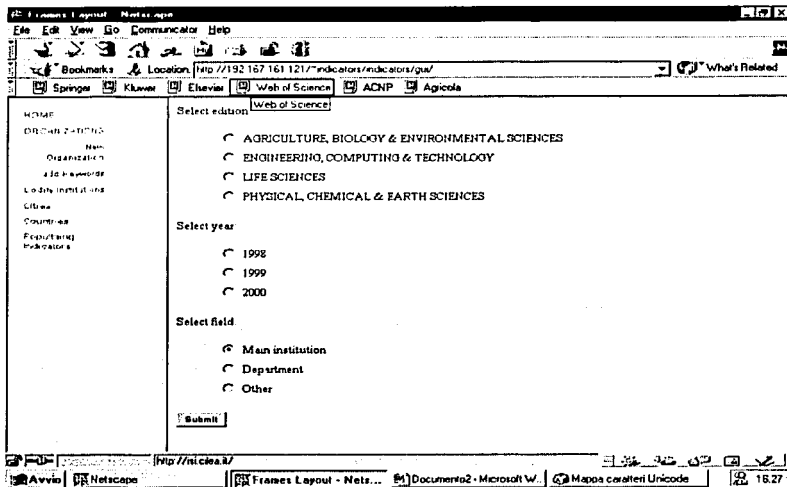


Figura A.11: Selección de edición, año y atributo

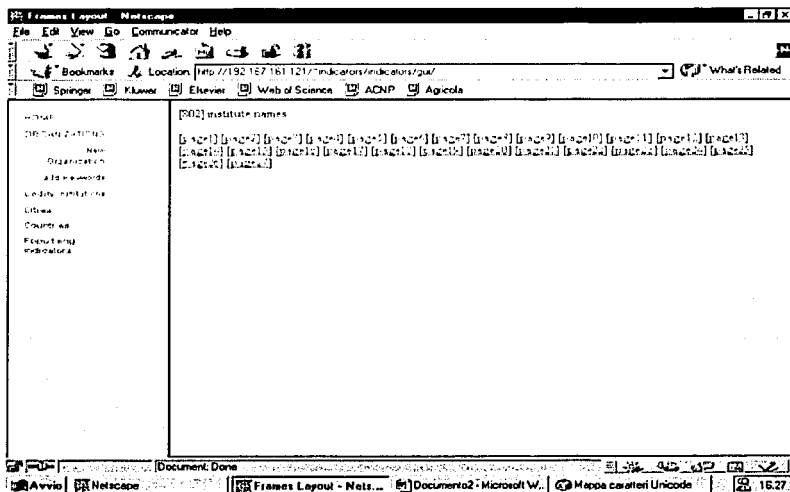


Figura A.12: Selección de página

Num. Records	First organization	Second organization	more details	City
1	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		ROMA
2	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		GENOVA
3	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		FIRENZE
4	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		GENOVA
5	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		FIRENZE
6	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		TRIBIANO
7	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND	ZAMMELLI CURIONI, Sestini	MODENA
8	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND	UNION CHIM. Ambientale 1, Dipartimento Prov	FIRENZE
9	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND	INT. COOPERAZIONE SCIENTIF. INTERNAZIONALE	LEGHARO
10	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		ROMA
11	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		FIRENZE
12	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		VIMERCATE
13	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		GENOVA
14	AGENZIA AEREA CIVILE ITALIANA	ENR MISTRAND		GENOVA

Figura A.13: Lista de nombre de institutos no estandarizados

ENR MISTRAND

KeyWord:

Organizations:

City:

Figura A.14: Escribir palabra clave

TESIS CON
FALLA DE ORIGEN

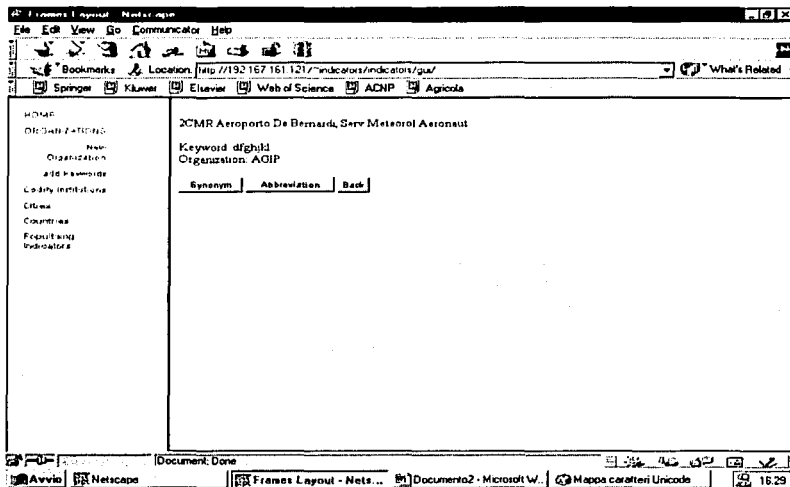


Figura A.15: Seleccionar sinónimo o abreviación

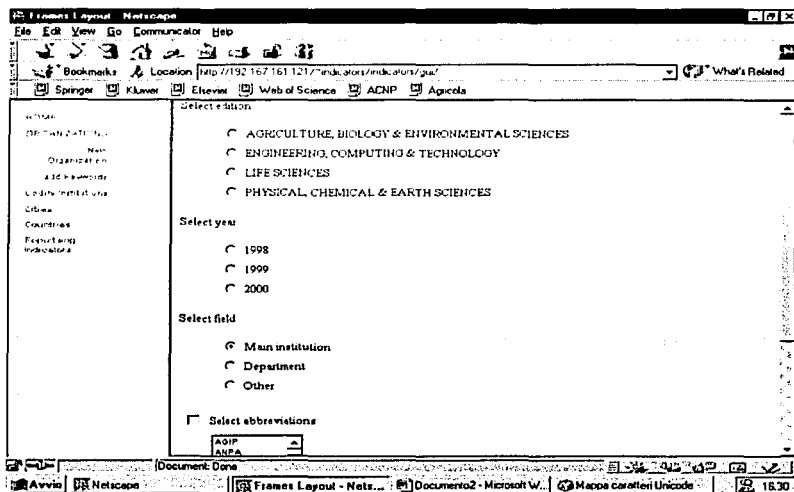


Figura A.16: Codificar organizaciones

TESIS CON
FALLA DE CALIDAD