

01132
69



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA
CIUDAD UNIVERSITARIA

**DESARROLLO DE UN SISTEMA PARA
CONSULTAR Y ALIMENTAR UN DICCIONARIO
TERMINOLÓGICO EN WEB**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

MUÑOZ CORTÉS GABRIELA

TESIS CON
FALLA DE ORIGEN

DIRECTOR: M. en C. GABRIEL CASTILLO HERNÁNDEZ

MÉXICO, D. F.

Se autoriza a la Dirección General de Bibliotecas
UNAM a difundir en formato electrónico e impreso
el contenido de este trabajo.
NOMBRE: MUÑOZ CORTÉS
GABRIELA
FECHA: 8 Julio-2003
FIRMA: GABRIELA MUÑOZ C.

2003



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.

INTRODUCCIÓN	1
CAPÍTULO 1 Fundamentos Teóricos	
1.1 Introducción	3
1.2 Ingeniería de software	5
1.2.1 Definición	5
1.2.1.1 Análisis	5
1.2.1.2 Diseño	6
1.2.1.3 Desarrollo e Implantación	8
1.2.1.4 Mantenimiento	
1.3 Bases de datos	8
1.3.1 Definición de Bases de Datos	8
1.3.2 Objetivos de los Sistemas de Bases de Datos	9
1.3.3 Abstracción de la Información	9
1.3.3.1 Nivel Físico	9
1.3.3.2 Nivel conceptual	9
1.3.3.3 Nivel de visión	10
1.3.4 Instancias y Esquemas	
1.4 Modelo de datos	10
1.4.1 Definición de Modelo de Datos	10
1.4.2 Modelos lógicos basados en objetos.	10
1.4.2.1 Modelo Entidad-Relación	10
1.4.3 Modelos lógicos basados en registros.	12
1.4.3.1 Modelo Relacional	12
1.4.3.2 Modelo de Red	12
1.4.3.3 Modelo Jerárquico	13
1.5 Manejador de bases de datos	13
1.5.1 Definición de DBMS	13
1.5.2 Estructura general de un Sistema Manejador de Base de Datos	14
1.5.3 Administrador de bases de datos	16
1.5.4 Lenguaje de definición de datos (DDL Data Definition Language)	16
1.5.5 Lenguaje de manipulación de datos (DML Data Manipulation Language).	16
1.6 El servidor microsoft SQL Server	17
1.6.1 Descripción de Microsoft sql server 2000	17
1.6.2 Requerimientos de Hardware	18
1.7 Modelo relacional y bases de datos relacionales	18
1.7.1 características de los modelos relacionales	18
1.7.2 Estructura de las bases de datos relacionales.	19
1.8 Lenguajes de consulta de bases de datos	20
1.8.1 Lenguajes formales	20
1.8.2 Lenguajes de consulta comerciales	20

ÍNDICE

1.9	SQL (structured query language)	21
	1.9.1 Definición	21
	1.9.2 Componentes de SQL	21
	1.9.2.1 Comandos	21
	1.9.2.2 Cláusulas	22
	1.9.2.3 Operadores Lógicos	22
	1.9.2.4 Operadores de comparación	23
	1.9.2.5 Consultas	23
1.10	Internet	24
	1.10.1 Definición	24
	1.10.2 Arquitectura Cliente / Servidor	24
	1.10.3 TCP / IP	25
1.11	Servidor Web y Microsoft Internet Information Server	26
	1.11.1 Respuesta del servidor Web	26
	1.11.1.1 Páginas estáticas	26
	1.11.1.2 Páginas dinámicas	26
	1.11.1.3 Listas de directorios	27
	1.11.2 Microsoft Internet Information Server	27
	1.11.2.1 Definición	27
	1.11.2.2 Como funciona Internet Information Server	28
1.12	CGI (Common Gateway Interface)	30
	1.12.1 Definición de CGI	30
1.13	El lenguaje de programación Perl	32
	1.13.1 Definición de Perl	32
	1.13.2 Sintaxis básica de un programa en Perl	33
	1.13.3 Perl y SQL Server	33
1.14	Programación de CGI's para Web en Perl	34
	1.14.1 Biblioteca cgi.pm	35
1.15	Microsoft Windows 2000 Server	35
	1.15.1 Descripción de Microsoft Windows 2000 Server	35
	1.15.2 Requerimientos de Hardware	35
1.16	Representación esquemática Del Sistema, Escenarios	36
1.17	Descripción de fichas técnicas	40
1.18	Recursos	41
1.19	Resumen del Capítulo	42

CAPÍTULO 2 Análisis y Diseño

2.1	Marco Histórico	42
2.2	Requerimientos y Necesidades en la elaboración del Sistema	44

C

TESIS CON
FALLA DE ORIGEN

ÍNDICE

2.3	Consideraciones y Expectativas en la elaboración del Sistema	48
2.4	Herramientas a utilizar	49
2.5	Diseño Modular del Sistema	51
2.5.1	Diagrama jerárquico funcional del sistema	51
2.5.1.1	Diseño lógico de la base de datos	54
2.5.1.2	Modelo conceptual de datos	54
2.5.1.3	Modelo entidad - relación	54
2.5.1.4	Diccionario de datos	56
2.5.2	Escenarios del sistema	58
2.6	Resumen del Capítulo	73
CAPÍTULO 3 Desarrollo, implantación y resultados		
3.1	Estructura de los Scripts en Perl	74
3.1.1	Scripts	74
3.1.1	Páginas estáticas	76
3.1.2	Páginas dinámicas	77
3.2	Servidor de base de datos	80
3.2.1	Estructura física de la base de datos	80
3.2.2	Scripts de creación de la base de datos	81
3.2.3	Stores procedures	83
3.3	Servidor Web	87
3.4	Sistema de administración	87
3.5	Pruebas del sistema	89
3.5.1	Pruebas Unitarias de Integración	89
3.5.2	Pruebas Globales	90
3.6	Ejemplo de navegación del sistema	91
3.7	Resumen del Capítulo	111
CONCLUSIONES		112
BIBLIOGRAFÍA		114

INTRODUCCIÓN

El presente trabajo describe el proceso de desarrollo de un Sistema para explotar un diccionario terminológico en Web, mostrando la metodología aplicada en las diferentes etapas desde el análisis hasta la implantación y resultados del mismo.

Este sistema nace como una herramienta que permitirá capturar, recopilar y consultar las bases de datos terminológicas en el área de lingüística, así como de las bases de datos que sean requeridas por el grupo de Ingeniería Lingüística del Instituto de Ingeniería.

El sistema contempla un módulo de Administración que permitirá dar de alta nuevos usuarios al sistema, así como modificar o eliminar permisos a las bases terminológicas de los usuarios a través del Web.

En el capítulo uno se presenta los fundamentos teóricos que dan pauta para el desarrollo del sistema. Aquí se describen conocimientos teóricos, tecnologías y herramientas. También se describe la metodología utilizada para el desarrollo de la interfaz Web, fichas técnicas y los recursos disponibles para la implementación del sistema.

El capítulo dos muestra un panorama general de la Ingeniería Lingüística, así como del Grupo de Ingeniería Lingüística. Se plantean los requerimientos y necesidades del sistema, así como las consideraciones y expectativas, las herramientas a utilizar y la justificación de la elección de la misma, el diseño modular del sistema, diseño de la base de datos, incluyendo los diagramas E-R así como el diccionario de datos. También se muestra parte de los escenarios para la interfaz Web.

En el capítulo tres se muestra parte de las fichas técnicas de las páginas estáticas, páginas dinámicas y los Scripts desarrollados. Se describe el servidor de bases de datos, algunos de los Scripts de creación de bases de datos y parte de los Store Procedures. Posteriormente se habla sobre el servidor Web.

En este mismo capítulo, se describe el sistema de administración del sistema y un esbozo del mismo. Se mencionan las pruebas efectuadas al sistema y los resultados obtenidos, y finalmente se muestra un ejemplo de navegación sencillo del sistema ya implantado.

Como parte final, se presentan las conclusiones y los resultados obtenidos.

Cabe señalar que se cuenta con un manual Técnico del sistema, el cual contiene en su totalidad los escenarios de la Interfaz Web, las fichas técnicas de las páginas estáticas, las páginas Estáticas, los Scripts, los Procedimientos almacenados y los Scripts de creación de la base de datos así como las restricciones de la base de datos que permiten establecer la integridad referencial.

CAPÍTULO 1 FUNDAMENTOS TEÓRICOS

1.1 Introducción

El análisis, diseño y desarrollo de los sistemas de cómputo se basa en la aplicación de un conjunto de técnicas y tecnologías. El trabajo presentado en este documento no es la excepción, en este capítulo se presentan los fundamentos teóricos sobre los cuales descansa el sistema desarrollado, el cual está organizado de la siguiente manera:

Introducción general de Ingeniería de software, las etapas necesarias a llevar a cabo en el desarrollo de un sistema y una breve descripción de cada una de ellas.

Breve descripción de bases de datos, que son y cuáles son los objetivos principales de una base de datos, así como los diferentes niveles de interacción entre el sistema de base de datos y los usuarios/programadores.

Modelos de Datos, su definición y una breve descripción de los modelos más utilizados.

La definición de un sistema manejador de bases de datos (DBMS), sus principales características y como es un DBMS de forma general.

El servidor de bases de datos utilizado (Microsoft SQL Server), sus principales características y los requerimientos de hardware para su instalación.

Modelo Relacional –modelo utilizado en el desarrollo del sistema presentado-, sus características y las bases relaciones

Lenguajes de Consulta, su Clasificación y en que consiste cada uno de ellos.

SQL (Structured Query Lenguaje) Definición y principios básicos del lenguaje.

TESIS CON
FALLA DE ORIGEN

Internet. Definición, direcciones IP, DNS, etc.

Servidor Web. Definición de Servidor, servicios que puede ofrecer un servidor Web, diferentes tipos de respuesta del servidor, mención de los principales servidores, cómo funciona Microsoft Internet Information Server, etc.

CGI (Common Gateway Interface) , en qué consiste y cómo funciona.

Perl (Practical Extraction and Report Language). Principales características del lenguaje, librerías existentes para interactuar con SQL Server, etc.

Programación de CGI´s con Perl, La biblioteca CGI.pm de perl 5 para crear formularios Web, etc.

Microsoft Windows 2000 Server. Principales características y requerimientos de Hardware para su instalación.

Posteriormente, se definirá la simbología que se utilizará para hacer la representación esquemática de la interfaz Web.

Se definirá lo que es una ficha técnica y se describirá lo que cada página estática, dinámica, Script ó Store Procedure contendrá.

Recursos y requerimientos mínimos para la utilización del sistema a desarrollar.

Conclusiones del capítulo

1.2 Ingeniería de Software

1.2.1 DEFINICIÓN

Una definición de Ingeniería de Software es la presentada en el libro de IEEE titulado "Standard Glossary of Software Engineering Terminology", en el cual se define como el enfoque sistemático para el desarrollo de operación, mantenimiento y eliminación del software donde software se define como aquellos programas, procedimientos, reglas y documentación posible asociada con la computación, así como los datos pertenecientes a la operación de un sistema de cómputo.

El desarrollo de software va unido a un **ciclo de vida**, compuesto por una serie de etapas que comprenden todas las actividades, desde el momento en que surge la idea de crear, hasta aquel en el que el software necesita mantenimiento o evolucionar.

Para el desarrollo del sistema presentado, se llevaron a cabo las etapas que se describen brevemente a continuación:

1.2.1.1 Análisis

Durante esta etapa, se deben de tener en claro los requerimientos y necesidades de los usuarios del sistema a desarrollar, mediante pláticas y entrevistas con los responsables y usuarios del proyecto.

Es necesario determinar qué elementos intervienen en el sistema a desarrollar, así como su estructura, relaciones, evolución en el tiempo, detalle de sus funcionalidades, etc., que van a dar una descripción clara de qué sistema se va a construir, qué funcionalidades aportará y qué comportamiento va a tener.

También es necesario definir la funcionalidad del sistema, que comportamiento va a tener y formalizar los requerimientos, que definan de forma precisa el sistema requerido.

De hecho, el documento inicial puede modificarse repetidas ocasiones hasta que se vea satisfecho completamente las necesidades del usuario, ya que por una parte, no siempre se sabe desde el principio las necesidades reales del sistema por parte de los usuarios, y por otra parte, pueden surgir cambios que supongan requerimientos nuevos o modificaciones de los ya contemplados.

TESIS CON
FALLA DE ORIGEN

1.2.1.2 Diseño

Tras la etapa anterior, ya se tiene claro qué debe hacer el sistema, ahora es necesario determinar cómo hacerlo (cómo debe ser construido el sistema).

En el diseño se determinan las condiciones y restricciones de las herramientas a utilizar necesarias para la funcionalidad del sistema. Aquí se seleccionará el lenguaje más adecuado, el sistema gestor de bases de datos a utilizar, configuraciones de hardware y software, redes, etc., según sea el caso.

1.2.1.3 Desarrollo e Implantación

Durante esta etapa se lleva a cabo el proceso de verificar e instalar nuevo equipo en el caso de ser necesario y la instalación de las aplicaciones que van desde la creación de los esquemas de bases de datos y sus archivos vacíos (en el caso de que exista información en una base de datos tal vez se requieran rutinas de conversión para modificar el formato de los datos y así poder almacenarlos en la nueva base de datos) en el DBMS (Sistema Manejador de Base de Datos. Ver más adelante para una mayor descripción) comercial seleccionado. En esta etapa también se incluye la programación necesaria, así como las pruebas del sistema.

El proceso de pruebas es uno de los componentes de un conjunto de actividades que permiten asegurar la calidad del sistema.

Uno de los principios básicos de pruebas es que éstas han de ser llevadas a cabo por personas distintas a los desarrolladores, tanto para evitar una simple verificación de que el programa funcione correctamente, como para probar que el sistema ha sido interpretado correctamente.

Los casos de prueba deben ser escritos tanto para condiciones de entradas válidas o inesperadas, como para condiciones válidas y esperadas.

Para cada sistema se realizan diferentes tipos de pruebas:

- **Pruebas Unitarias:** Todos los componentes del sistema que se desarrollen individualmente para comprobar su correcto funcionamiento.
- **Pruebas de integración:** Se prueba la integración entre los componentes del sistema para demostrar que se puede encajar correctamente.
- **Pruebas de sistema:** Se prueba el sistema globalmente.
Existen dos tipos de Prueba:

**TESIS CON
FALLA DE ORIGEN**

Pruebas del tipo CAJA BLANCA, que permite examinar la estructura interna del sistema.

Pruebas del tipo CAJA NEGRA, donde los casos de prueba se diseñan considerando exclusivamente las entradas y las salidas del sistema, sin preocuparse por la estructura interna del mismo.

Pruebas unitarias de integración

Las pruebas unitarias se realizan al crear cada módulo o componente individual del sistema. Las pruebas de integración son realizadas sobre estos componentes individuales, de que son llamados cuando es necesario y que los datos que se transmiten entre dichos componentes son los requeridos.

Se cuenta con dos tipos de pruebas:

a) Desarrollo Incremental

El tipo de prueba incremental consiste en agregar cada módulo o componente individual al conjunto de componentes existentes y el resultado resultante de prueba. Esto reduce la necesidad de crear módulos conductores y permite además, examinar en detalle las interfases. Además, cuando las pruebas unitarias y de integración se realizan separadamente, es difícil examinar los componentes individuales o módulos que causan resultados incorrectos.

b) Estrategias de Integración

Para una estructura de programas organizada jerárquicamente, se puede enfocar el problema de la prueba utilizando varias estrategias diferentes:

- * Estrategias de arriba a abajo (Top-Down)
- * Estrategias de abajo a arriba (Bottom-Up)
- * Estrategias combinadas
- * Comparación de estrategias
- * pruebas del sistema y de aceptación

Pruebas del sistema y de aceptación

a) Pruebas globales

Una vez que se han probado los componentes individuales y se han integrado, se ha de probar el sistema global. En esta etapa pueden distinguirse los siguientes tipos de pruebas, cada uno con un objetivo claramente definido:

- Pruebas funcionales
- Pruebas de comunicación

TESIS CON
FALLA DE ORIGEN

- Pruebas de rendimiento
- Pruebas de volumen
- Pruebas de sobrecarga
- Pruebas de disponibilidad de datos
- Pruebas de facilidad de uso
- Pruebas de operación
- Pruebas de seguridad

b) Pruebas de aceptación

Son aquellas pruebas que realiza el usuario con el objeto de comprobar si es el sistema es aceptable para él. Estas pruebas son del mismo tipo que las mencionadas anteriormente, pero son determinadas por el usuario en lugar de serlo por el equipo de desarrollo.

La etapa de desarrollo termina hasta la puesta en marcha del sistema.

1.2.1.4 Mantenimiento

El sistema resultante se encuentra ya en fase de producción (en funcionamiento para los usuarios, cumpliendo ya los objetivos para los que ha sido creada). A partir de este momento se entra en la etapa de mantenimiento, que supondrá pequeñas operaciones tanto de mejora del sistema así como de otras de mayor importancia.

1.3 Bases de Datos

1.3.1 DEFINICIÓN DE BASES DE DATOS

Una base de datos es una colección de archivos interrelacionados que son creados con un DBMS. El contenido de una base de datos engloba a la información concerniente a una organización, de tal manera que los datos estén disponibles para los usuarios en toda la organización; los usuarios pueden extraer información sin considerar las fronteras de los archivos.

Un objetivo importante de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema debe esconder ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo, para que el sistema sea manejable, los datos se deben extraer eficientemente.

1.3.2 OBJETIVOS DE LOS SISTEMAS DE BASES DE DATOS

Los principales objetivos de los sistemas de bases de datos es disminuir los siguientes aspectos:

- Redundancia e inconsistencia de datos
- Dificultad para tener acceso a los datos
- Aislamiento de los datos
- Anomalías del acceso recurrente
- Problemas de seguridad
- Problemas de integridad

1.3.3 ABSTRACCIÓN DE LA INFORMACIÓN

Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema: interno, conceptual y externo, estos niveles corresponden al punto de vista de almacenamiento físico, del usuario y del programador respectivamente.

1.3.3.1 Nivel Físico

Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en cómo se almacenan los datos en los dispositivos de almacenamiento.

1.3.3.2 Nivel conceptual

El siguiente nivel más alto de abstracción, describe qué datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos.

Consta de las siguientes definiciones:

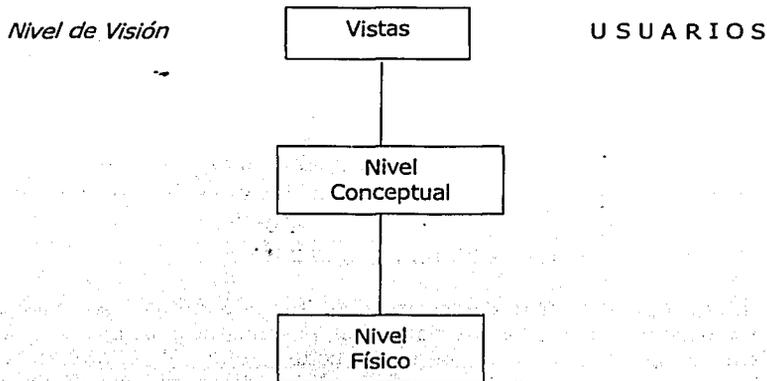
1. **Definición de los datos:** se describen el tipo de datos y la longitud de campo de todos los elementos bidireccionales en la base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos y registros conceptuales (entidades).
2. **Relaciones entre datos:** se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

En el nivel conceptual la base de datos aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario.

1.3.3 Nivel de vision

Es el nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la base de datos al usuario acreditado para verla. El sistema puede proporcionar muchas visiones de la misma base de datos para diferentes usuarios

La interrelación entre estos tres niveles de abstracción se ilustra en la siguiente figura.



1.3.4 INSTANCIAS Y ESQUEMAS

Se llama Instancia al estado que presenta una base de datos en un tiempo dado. Por otro lado, un Esquema es la descripción lógica de la base de datos, proporciona los nombres de las entidades y sus atributos especificando las relaciones que existen entre ellos.

TESIS CON
FALLA DE ORIGEN

1.4 Modelo de datos

1.4.1 DEFINICIÓN DE MODELO DE DATOS

Un modelo es una representación de la realidad que contiene las características generales de algo que se va a realizar. En base de datos, esta representación se elabora de forma gráfica.

Un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Los modelos de datos se dividen en dos grupos:

- Modelos lógicos basados en objetos
- Modelos lógicos basados en registros

1.4.2 MODELOS LÓGICOS BASADOS EN OBJETOS.

Se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo representamos los datos de tal forma como nosotros los captamos en el mundo real; tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen seis tipos de modelos lógicos: Modelo de entidad relación, Modelo orientado a objetos, Modelo binario, Modelo semántico de datos, y Modelo funcional de datos, pero el más utilizado, por su sencillez y eficiencia, es el modelo entidad-Relación.

1.4.2.1 Modelo Entidad-Relación

Denominado por sus siglas como E-R. Este modelo representa a la realidad a través de **entidades**, que son objetos que existen y que se distinguen de otros por sus características.

Las entidades pueden ser de dos tipos:

➤ **Tangibles:**

Son todos aquellos objetos físicos que podemos ver, tocar o sentir.

➤ **Intangibles**

Todos aquellos eventos u objetos conceptuales que no podemos ver, aún sabiendo que existen.

Las características de las entidades en base de datos se llaman **atributos**. A su vez, una entidad se puede asociar o relacionar con más entidades a través de **relaciones**.

La estructura lógica global de una BD puede expresarse gráficamente por el diagrama E-R:

- Rectángulos: representan conjunto de entidades.
- Elipses: representan atributos.
- Rombos: representan relaciones entre conjuntos de entidades.
- Líneas: que conectan atributos a conjuntos de entidades a relaciones.

1.4.3 MODELOS LÓGICOS BASADOS EN REGISTROS.

Se utilizan para describir datos en los niveles conceptual y físico. Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de la implementación.

Los tres modelos de datos más ampliamente aceptados son:

- Modelo relacional
- Modelo de Red
- Modelo Jerárquico

1.4.3.1 Modelo Relacional

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro. Más adelante se dan más detalles de este modelo.

1.4.3.2 Modelo de Red

Este modelo representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias.

TESIS CON
FALLA DE ORIGEN

1.4.3.3 Modelo Jerárquico

Este modelo es similar al modelo de red en cuanto a las relaciones y datos, ya que éstos se representan por medio de registros y sus ligas. La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias.

1.5 Manejador de Bases de Datos

1.5.1 DEFINICIÓN DE DBMS

Un DBMS (Sistema Manejador de Base de Datos) es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica. El objetivo principal de un sistema manejador de Bases de datos es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfaz entre los usuarios y la base de datos.

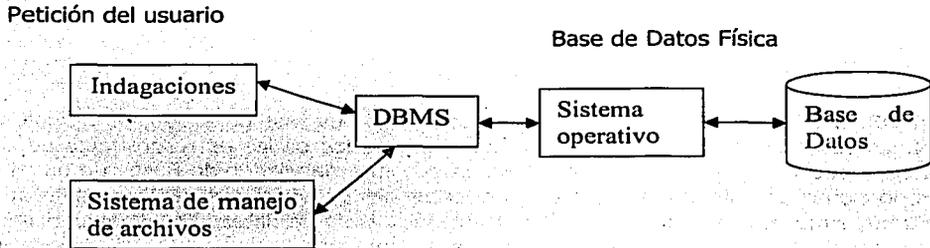
Las funciones principales de un DBMS son:

- Crear y organizar la base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos
- Interacción con el manejador de archivos.
Esto a través de las sentencias en DML al comando del sistema de archivos. Así el manejador de la base de datos es el responsable del verdadero almacenamiento de los datos.
- Respaldo y recuperación .
Consiste en contar con mecanismo implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia
Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.

TESIS CON
FALLA DE ORIGEN

➤ Seguridad e integridad

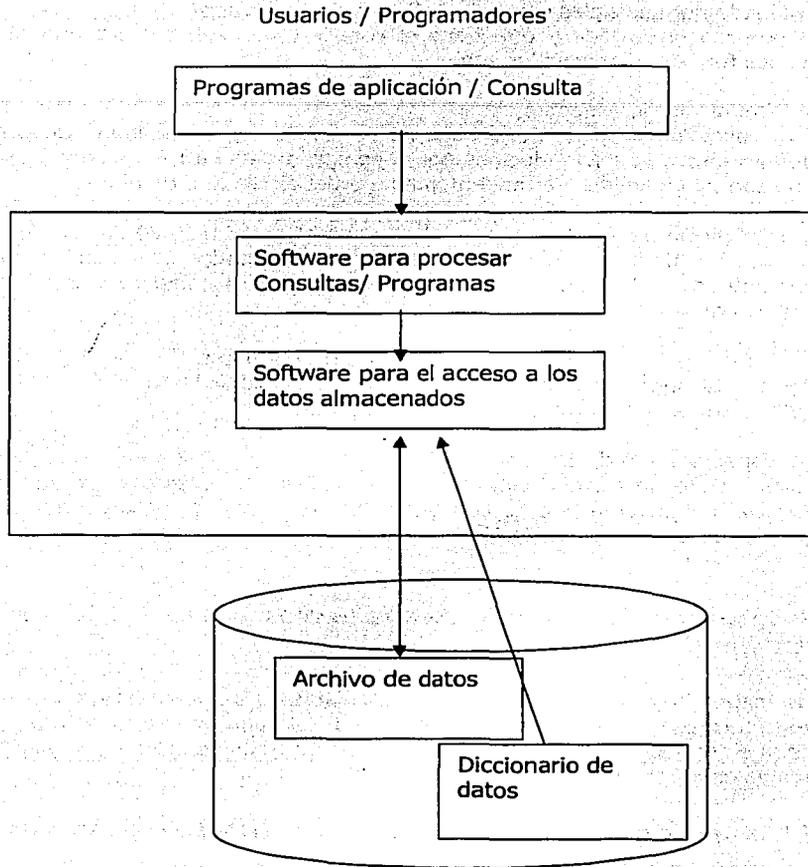
Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que éstos se vean perjudicado por cambios no autorizados o previstos.



La figura muestra el DBMS como interfase entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

1.5.2 ESTRUCTURA GENERAL DE UN SISTEMA MANEJADOR DE BASE DE DATOS.

Un sistema de base de datos se encuentra dividido en módulos, cada uno de los cuales controla una parte de la responsabilidad total del sistema. En la mayoría de los casos, el sistema operativo proporciona únicamente los servicios más básicos y el sistema de la base de datos debe partir de esa base y controlar además el manejo correcto de los datos. Así el diseño de un sistema de base de datos debe incluir la interfaz entre el sistema de base de datos y el sistema operativo.



1.5.3 ADMINISTRADOR DE BASES DE DATOS

Denominado por sus siglas como DBA: DataBase Administrator

Es la persona encargada y que tiene el control total sobre el sistema de base de datos, sus funciones principales son:

➤ **Definición de esquema**

Es el esquema original de la base de datos. Se crea escribiendo un conjunto de definiciones que son traducidas por el compilador de DDL a un conjunto de tablas que son almacenadas permanentemente en el diccionario de datos.

➤ **Definición de la estructura de almacenamiento del método de acceso**

Estructuras de almacenamiento y el acceso adecuados se crean escribiendo un conjunto de definiciones que son traducidas por el compilador del lenguaje de almacenamiento y definición de datos.

➤ **Concesión de autorización para el acceso a los datos**

Permite al administrador de la base de datos regular las partes de las bases de datos que van a ser accedidas por varios usuarios.

➤ **Especificación de limitantes de integridad**

Es una serie de restricciones que se encuentran almacenados en una estructura especial del sistema que es consultada por el gestor de base de datos cada vez que se realice una actualización al sistema.

1.5.4 LENGUAJE DE DEFINICIÓN DE DATOS (DDL DATA DEFINITION LANGUAGE).

Nos permite definir el esquema de la base de datos (el esquema es la estructura por la que está formada la base de datos) por medio de una serie de definiciones que se expresan en un lenguaje especial, y el resultado de estas definiciones se almacenan en un archivo especial llamado diccionario de datos.

1.5.5 LENGUAJE DE MANIPULACIÓN DE DATOS (DML DATA MANIPULATION LANGUAGE).

La manipulación de datos se refiere a las operaciones de insertar, recuperar, eliminar o modificar datos; dichas operaciones son realizadas a través del lenguaje de manipulación de datos, que es quien permite el acceso de los usuarios a los datos. Existen básicamente dos tipos de lenguajes de manipulación de datos

➤ **Procedimentales**

Los DML requieren que el usuario especifique qué datos se necesitan y cómo obtenerlos.

➤ No Procedimentales

Los DML requieren que el usuario especifique qué datos se necesitan y no es necesario que se especifique cómo obtenerlos.

1.6 El servidor Microsoft SQL Server

1.6.1 DESCRIPCIÓN DE MICROSOFT SQL SERVER 2000

Microsoft SQL Server 2000 es un sistema manejador de bases de datos relacionales (RDBMS) basado en la arquitectura Cliente / Servidor.

Microsoft SQL Server 2000 tiene como principales características:

Facilidad de uso. Permite la administración de varios servidores con una única consola; ejecución de trabajos basados en eventos y generación de alertas; seguridad integrada y procedimientos de comandos para realizar tareas administrativas.

Ampliable, fiable. Almacenamiento de datos

SQL Server 2000 admite un amplio intervalo de acceso de usuarios al mismo tiempo. Una instancia de SQL Server 2000 incluye los archivos que crean un conjunto de bases de datos y una copia del software DBMS. Las aplicaciones que se ejecutan en equipos diferentes utilizan un componente de comunicaciones de SQL Server 2000 para transmitir comandos a través de una red a la instancia de SQL Server 2000. Cuando una aplicación se conecta a una instancia de SQL Server 2000, puede hacer referencia a cualquiera de las bases de datos de esa instancia a la que el usuario tenga acceso. El componente de comunicación permite también la comunicación entre una instancia de SQL Server 2000 y una aplicación que se esté ejecutando en el mismo equipo. Puede ejecutar varias instancias de SQL Server 2000 en un único equipo.

SQL Server 2000 está diseñado para admitir el tráfico de los sitios Web de mayor tamaño o sistemas de procesamiento de datos empresariales. Las instancias de SQL Server 2000 que se ejecutan en servidores multiprocesador de gran tamaño pueden admitir conexiones de miles de usuarios al mismo tiempo. Los datos de las tablas de SQL Server se pueden dividir entre varios servidores, de manera que varios equipos multiprocesador pueden cooperar para satisfacer los requisitos de procesamiento de bases de datos de sistemas extremadamente grandes. Estos grupos de servidores de base de datos se denominan federaciones.

Aunque SQL Server 2000 está diseñado para funcionar como motor de almacenamiento de datos para miles de usuarios que se conectan a través de una red simultáneamente, puede funcionar también como base de datos independiente directamente en el mismo equipo de una aplicación. Las características de escalabilidad y facilidad de uso de SQL Server 2000 le permiten trabajar eficazmente en un único equipo sin consumir demasiados recursos y sin que sean necesarias tareas administrativas por parte del usuario independiente. Las mismas características permiten a SQL Server 2000 adquirir de forma dinámica los recursos necesarios para admitir miles de usuarios, al tiempo que minimizan la administración y la optimización de bases de datos. El motor de base de datos relacional de SQL Server 2000 se ajusta a sí mismo de forma dinámica para adquirir o liberar la cantidad adecuada de recursos del equipo necesarios para admitir una carga variable de usuarios que tienen acceso a una instancia de SQL Server 2000 en un momento determinado. El motor de base de datos relacional de SQL Server 2000 dispone de características para evitar los problemas lógicos que tienen lugar si un usuario intenta leer o modificar datos que están utilizando otros usuarios.

El dialecto de SQL compatible con Microsoft SQL Server se llama Transact-SQL (T-SQL). T-SQL es el lenguaje principal utilizado por las aplicaciones de Microsoft SQL Server.

1.6.2 REQUERIMIENTOS DE HARDWARE

Computadora con Procesador Intel o compatible, Pentium a 166 MHz o superior, Pentium PRO o Pentium IV.

Memoria RAM mínimo 64 MB

Utiliza 180 MB para su instalación en el disco duro

Como mínimo para Enterprise Edition se requiere Microsoft Windows NT Server 4 que tenga instalado Service Pack 5 o posterior; aunque para disponer de algunas características de SQL Server 2000 es necesario tener Microsoft Windows 2000 Server.

1.7 Modelo Relacional y Bases de Datos Relacionales

1.7.1 CARACTERÍSTICAS DE LOS MODELOS RELACIONALES

En un modelo relacional los datos se almacenan, al menos conceptualmente, de un modo en que los usuarios entienden con mayor facilidad. Los datos se almacenan como tablas y las relaciones entre las filas y las tablas son visibles en los datos.

Las características más importantes de los modelos relacionales son:

- Las entradas en la tabla tienen un solo valor; no se admiten valores múltiples, por lo tanto la intersección de un renglón con una columna tiene un solo valor, no un conjunto de valores.
- Todas las entradas de cualquier columna son de un solo tipo. Cada columna posee un nombre único, el orden de las columnas no es de importancia.
- Cada atributo tiene un dominio, que es una descripción física y lógica de valores permitidos.
- No existen dos filas en la tabla que sean idénticas.
- La información en las bases de datos es representada como datos explícitos, no existen apuntadores o ligas entre las tablas.

1.7.2 ESTRUCTURA DE LAS BASES DE DATOS RELACIONALES.

La arquitectura relacional se puede expresar en términos de tres niveles de abstracción: nivel interno, conceptual y de visión.

La arquitectura relacional consta de los siguientes componentes:

1. Modelo relacional de datos

En el nivel conceptual, el modelo relacional de datos está representado por una colección de relaciones almacenadas. Cada registro de tipo conceptual en un modelo relacional de datos se implanta como un archivo almacenado distinto.

2. Submodelo de datos

Los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos; cada uno consta de uno o más escenarios (vistas) para describir los datos requeridos por una aplicación dada. Un escenario puede incluir datos de una o más tablas de datos.

3. Esquema de almacenamiento

En el nivel interno, cada tabla base se implanta como un archivo almacenado.

4. Sublenguaje de datos

Es un lenguaje de manejo de datos para el sistema relacional, el álgebra relacional y cálculo relacional.

1.8 Lenguajes de consulta de bases de datos

Los lenguajes de consulta son los lenguajes en el que los usuarios solicitan información de la base de datos.

1.8.1 LENGUAJES FORMALES

El álgebra relacional es un lenguaje de consulta de procedimientos, existen cinco operaciones fundamentales en el álgebra relacional que son: elegir, proyectar, producto cartesiano, unión y diferencia. Todas ellas producen como resultado una nueva relación.

El calculo relacional es un lenguaje de consulta sin procedimientos, donde se da una descripción formal de la información deseada sin especificar como obtenerla. Los operadores son: Para todo x tal que $f(x)$ y Existe al menos un x tal que $f(x)$. En el calculo relacional existen dos variantes: el calculo relacional de tuplas y el calculo relacional de dominios. El mas común en lenguajes de consulta es el de tuplas.

Los lenguajes formales son la base de los lenguajes comerciales, sin embargo, los lenguajes que se usan en la practica son los comerciales.

1.8.2 LENGUAJES DE CONSULTA COMERCIALES

Son lenguajes de bases de datos que permiten un nivel de consulta mas amigable para el usuario, tres ejemplos de ellos son: SQL, Quel, QBE. QBE, esta basado en el calculo relacional de dominios. Quel se basa en el calculo relacional de tuplas y SQL es una combinación del álgebra relacional y construcciones del calculo relacional.

El nombre SQL esta formado por las iniciales en ingles Structured Query Language (lenguaje de consulta estructurado). Este lenguaje proporciona formatos y sintaxis para la manipulación y definición de los datos.

1.9 SQL (Structured Query Language)

1.9.1 DEFINICIÓN

SQL (Structured Query Language - Lenguaje estructurado de consultas) es un lenguaje de alto nivel para sistemas de bases de datos relacionales. Desarrollado originalmente por el Laboratorio de Investigación de IBM en San José a finales de los años 70, SQL ha sido adoptado y adaptado en muchos sistemas de administración de bases de datos relacionales. Ha sido aprobado como norma oficial para lenguajes de consultas relacionales por parte del American National Standards Institute (ANSI) y la International Organization for Standardization (ISO).

1.9.2 COMPONENTES DE SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

1.9.2.1 Comandos

Existen dos tipos de comandos SQL:

- los DDL que permiten crear y definir nuevas bases de datos, campos e índices.
- los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comandos DDL

Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Comandos DML

Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

1.9.2.2 Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Cláusula	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

1.9.2.3 Operadores Lógicos

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

TESIS CON FALLA DE ORIGEN

1.9.2.4 Operadores de comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor ó Igual que
>=	Mayor ó Igual que
=	Igual que
BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo
In	Utilizado para especificar registros de una base de datos

1.9.2.5 Consultas

Una consulta es el proceso de solicitar datos de la base de datos y recibir resultados en respuesta. Este proceso también se conoce como recuperación de datos. Todas las consultas SQL se expresan mediante la instrucción SELECT. Las consultas pueden usarse para realizar selecciones, que recuperan un subconjunto de filas de una o más tablas, y proyecciones, que recuperan un subconjunto de columnas de una o más tablas.

Esta es una versión simplificada de la instrucción SELECT:

```
SELECT select_list
FROM table_list
WHERE search_conditions
```

La cláusula SELECT especifica las *columnas* que desea recuperar. La cláusula FROM especifica las *tablas* de donde se deben extraer las columnas. La cláusula WHERE especifica las *filas* de las tablas que se desean ver.

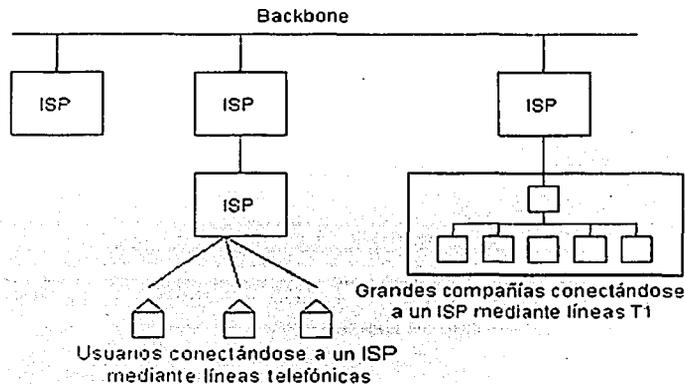


1.10 Internet

1.10.1 DEFINICIÓN

El Internet es una gigantesca colección de millones de computadoras que están unidas mediante una Red Computacional, también llamada Network. Esta red permite que todas las computadoras se comuniquen entre sí. Un computador casero es conectado usualmente a Internet utilizando una línea telefónica normal y un módem que se comunica con un ISP. Un computador de empresa o universidad posee un NIC que lo conecta directamente a una LAN dentro de la empresa. Toda la entidad conecta su LAN a un ISP (Internet Service Provider: Proveedor de Servicios de Internet) utilizando una línea telefónica en la mayoría de los casos.

Los ISP se conectan a otros ISP más grandes, y éstos mantienen conexiones de fibra óptica llamados "backbones" (columna vertebral), Infraestructura de la transmisión de datos en Internet) para una nación o región. Los backbones están conectados alrededor del mundo mediante cables submarinos o conexiones satelitales. De esta forma cada computadora en Internet está conectada a los demás.



1.10.2 ARQUITECTURA CLIENTE / SERVIDOR

La arquitectura Cliente / Servidor es la organización de un sistema de cómputo en tres partes:

TESIS CON
FALLA DE ORIGEN

- Servidor
- Cliente
- Red

Internet funciona con la estrategia "Cliente/ Servidor lo que significa que en la red hay ordenadores servidores que dan una información concreta en el momento que se solicite, y por otro lado están los ordenadores que piden dicha información, los llamados clientes.

Existe una gran variedad de "lenguajes" que usan los ordenadores para comunicarse por Internet. Estos "lenguajes" se llaman protocolos. El protocolo más utilizado en Internet es TCP/IP.

1.10.3 TCP / IP

Es un conjunto de protocolos conformado por un modelo de cuatro partes o capas:

Interfase de Red. Es responsable de poner y recuperar los paquetes del medio físico.

Red. Es responsable de las funciones de direccionamiento, empaquetamiento y ruteo. Un protocolo en esta capa es IP, que se encarga de rutear (ruteador: Sistema constituido por hardware y software para la transmisión de datos en Internet. El emisor y el receptor deben utilizar el mismo protocolo) y direccional paquetes entre nodos y redes

Transporte. Esta capa es capaz de proveer la comunicación entre dos nodos. Uno de los protocolos es TCP, el cual es un protocolo orientado a conexión; Establece comunicaciones confiables para aplicaciones que transfieran una gran cantidad de datos al mismo tiempo o requieran una confirmación de los datos recibidos.

Algunos de los números de puertos TCP más conocidos:

FTP: 21

Telnet: 23

SMTP (Simple Mail Transfer, para correo): 25

HTTP: 80

Aplicación. Esta capa está en la parte superior del modelo OSI (OSI, modelo de referencia universal para protocolos de comunicación, norma creada por la ISO: Organización Internacional para Estandarización, fundada en 1946, es una federación internacional que unifica normas en unos cien países)

TESIS CON
FALLA DE ORIGEN

1.11 Servidor Web y Microsoft Internet Information Server

En general, todas las máquinas en Internet pueden ser clasificadas en dos tipos: servidores y clientes. Las máquinas que proveen servicio (como Servidores Web, Servidores FTP, etc.) a otras máquinas son servidores. Las máquinas que se utilizan para conectarse a esos servicios son los clientes.

Un servidor puede brindar uno o más servicios en Internet. Por ejemplo, una máquina servidor debe tener un software que le permita actuar como un Servidor Web, un Servidor de correo y un Servidor FTP. Los clientes que se conectan a un Servidor hacen lo mismo con una tarea específica, así que los clientes envían sus peticiones directamente al software del Servidor que funciona en la Máquina Servidor.

Cada servidor Web en Internet utiliza el protocolo HTTP (Hypertext Transfer Protocol. Protocolo de transferencia de hipertextos que permite transferir información en archivos de texto, video, audio, etc.).

El comando básico que un servidor HTTP entiende es este: GET. Si se conecta a un servidor que entiende el protocolo HTTP y le dice "GET filename", el servidor responderá enviando el contenido del archivo especificado y desconectándose.

1.11.1 RESPUESTA DEL SERVIDOR WEB

Un servidor Web responde a la petición de un explorador de Web devolviendo una página HTML. La página devuelta puede ser de tres tipos: una página HTML estática, una página HTML dinámica o una página de lista de directorio.

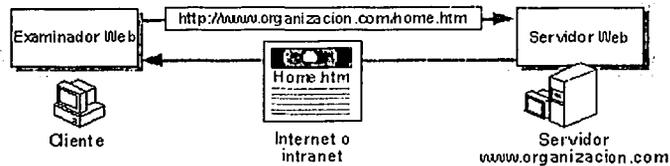
1.11.1.1 Páginas estáticas

Las páginas estáticas son páginas HTML fijas ya preparadas antes de la petición. El servidor Web devuelve las páginas HTML al usuario, pero no realiza ninguna acción especial. El usuario pide una página estática escribiendo una dirección URL (<http://www.organizacion.com/home.htm>, en la siguiente figura) o haciendo clic en un vínculo que apunte a una dirección URL. La petición de dirección URL se envía al servidor y éste responde devolviendo la página HTML estática.

1.11.1.2 Páginas dinámicas

Las páginas dinámicas se crean como respuesta a las peticiones de los usuarios. Un explorador de Web recopila información presentando una página con cuadros de texto, menús y casillas de verificación que el usuario completa o activa. Cuando el usuario

hace clic en un botón de un formulario, los datos procedentes del formulario se envían al servidor Web. El servidor pasa los datos a un archivo de comandos o una aplicación para su procesamiento, o bien consulta o envía datos a una base de datos. Entonces, el servidor devuelve los resultados al usuario en forma de una página HTML.



1.11.1.3 Listas de directorios

Si los usuarios quieren enviar consultas sin especificar un archivo en concreto, puede crear un documento predeterminado para un sitio Web o para un determinado directorio, o puede configurar su servidor para explorar los directorios. Si no se crea ningún documento predeterminado para un directorio y está configurada la exploración de los directorios, se devuelve al usuario una lista de directorios (una versión hipertexto de una lista de un Explorador de Windows o del Administrador de archivos) en forma de página HTML. Después, el usuario puede ir al archivo apropiado haciendo clic en él dentro de la lista de directorios.

Los servidores Web más populares son:

- Apache Web Server
- Microsoft Internet Information Server
- Java Web Server
- Freeware HTTP Server for Windows NT

1.11.2 MICROSOFT INTERNET INFORMATION SERVER

1.11.2.1 Definición

Microsoft Internet Information Server es un servidor Web que permite publicar información en una intranet de la organización o en Internet. Internet Information Server transmite la información mediante el Protocolo de transferencia de hipertexto (HTTP), y puede configurarse también para proporcionar servicios de Protocolo de transferencia de archivos (FTP) y gopher.

TESIS CON
FALLA DE ORIGEN

El servicio FTP permite que los usuarios transfieran archivos a y desde su sitio Web. El servicio gopher utiliza un protocolo controlado por menús para encontrar documentos. El protocolo gopher ha quedado en buena parte reemplazado por el protocolo HTTP.

Internet Information Server también proporciona otros servicios de información y es compatible con un gran número de interfaces que pueden utilizarse para desarrollar otras características para su sitio Web. Puede:

- Crear aplicaciones cliente-servidor de alto rendimiento usando la Interfaz de programación de aplicaciones servidoras de Internet de Microsoft (ISAPI).
- Personalizar el servicio WWW mediante la creación de programas ISAPI de tipo filtro que sigan las peticiones de entrada o de salida y realicen automáticamente ciertas acciones, como registros mejorados.
- Ejecutar aplicaciones o archivos de comandos de la Interfaz de puerta de enlace o gateway común (CGI).
- Transmitir o recibir archivos usando el servicio FTP.
- Publicar archivos de información, repartida en varios equipos, usando el servicio Gopher.

Internet Information Server incluye: servicios de Internet: WWW, FTP y gopher, administrador de servicios de Internet, la herramienta para administrar los servicios de Internet, Conector de bases de datos de Internet, el componente para enviar consultas a bases de datos, Administrador de claves, etc.

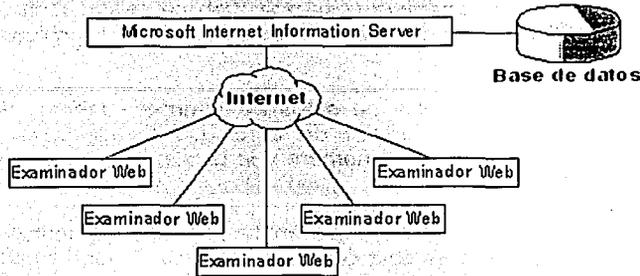
1.11.2.2 Como funciona Internet Information Server

Como se ha menciona anteriormente, el Web es fundamentalmente un sistema de peticiones y respuestas. Los exploradores de Web solicitan información, enviando una dirección URL a un servidor Web. El servidor Web responde devolviendo una página de Hypertext Markup Language (La página HTML puede ser, como se mencionó anteriormente, una página estática a la que ya se haya dado formato y esté almacenada en el sitio Web, una página que el servidor crea dinámicamente como respuesta a la información proporcionada por el usuario o una página que presente una lista de los archivos y carpetas disponibles en el sitio Web).



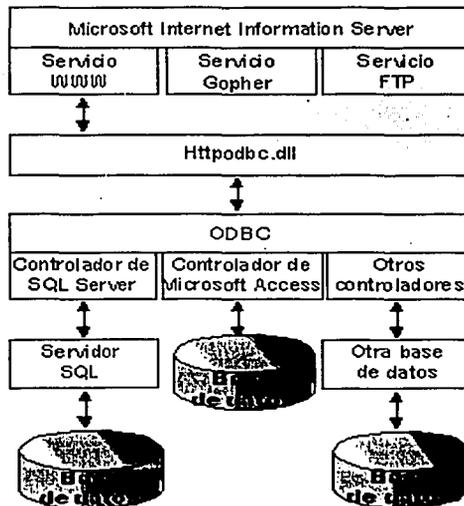
TESIS CON
FALLA DE ORIGEN

En el siguiente diagrama se muestra conceptualmente el acceso a bases de datos desde Internet Information Server.



Los exploradores de Web remiten peticiones al servidor Internet usando HTTP. El servidor Internet responde con un documento en formato HTML. El acceso a las bases de datos se realiza mediante un componente de Internet Information Server llamado Conector de bases de datos de Internet (IDC). El Conector de bases de datos de Internet, Httpodbc.dll, es una DLL ISAPI que utiliza ODBC para tener acceso a las bases de datos.

La siguiente ilustración muestra los componentes de Internet Information Server para conectar con las bases de datos.



TESIS CON FALLA DE ORIGEN

El IDC utiliza dos tipos de archivos para controlar la forma de acceso a la base de datos y el modo en que se construye la página Web de salida. Dichos archivos son archivos del Conector de bases de datos de Internet (.idc) y archivos de extensión HTML (.htx).

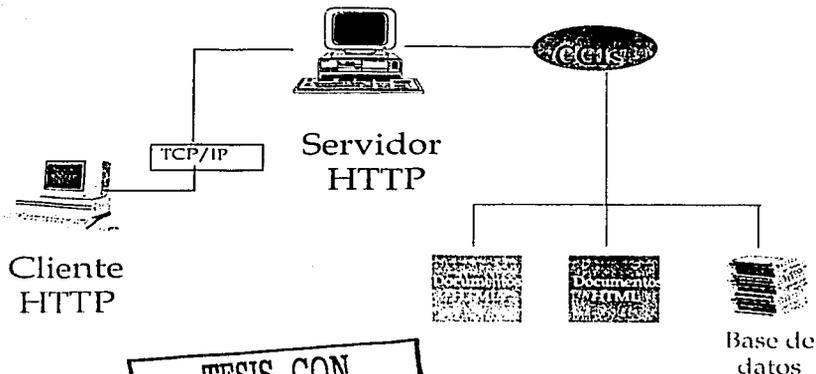
Los archivos del Conector de bases de datos de Internet contienen la información necesaria para conectar con el origen de datos ODBC adecuado y ejecutar la instrucción SQL. Además, contienen el nombre y la ubicación del archivo de extensión HTML.

El archivo de extensión HTML constituye la plantilla para el documento HTML real que se devolverá al explorador de Web cuando el IDC haya combinado su información con la base de datos.

1.12 CGI (Common Gateway Interfase)

1.12.1 DEFINICIÓN DE CGI

CGI es un estándar para programas que sirven de puente entre el Web y una aplicación. Por ejemplo, uno puede querer que una aplicación, por lo general de base de datos, pueda utilizarse a través de páginas del Web. Así, se puede hacer que un programa CGI pinte una interfaz para el usuario (forma del Web) quien llena esta solicitud completando ciertos campos de texto o seleccionando opciones de menús. Esta solicitud es recogida por el CGI quien la transforma en comandos y datos inteligibles por la aplicación; a su vez ésta le regresa al CGI los resultados de la consulta y el CGI se los despliega al usuario en una nueva página Web, tal como se muestra en la siguiente figura:



TESIS CON
FALLA DE ORIGEN

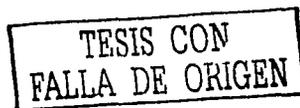
El estándar CGI surgió al mismo tiempo que el Web, para ampliar sus posibilidades al incluirle programación. Un CGI siempre corre del lado del servidor, lo que significa que cuando se accede a una página Web que contiene un CGI, éste se ejecuta automáticamente en el servidor y lo que se ve posteriormente en una página, es el resultado de esta ejecución. Esto redundará en lo siguiente:

- Aumento de la carga en el servidor, si se ejecutan simultáneamente varios CGIs complejos o que usen aplicaciones complejas.
- Por lo anterior, el cliente no sufre sobrecarga y, de hecho, el usuario puede no darse cuenta de que la página que está viendo ha sido generada por un CGI (y no proviene de un archivo HTML).
- Si no se toman las debidas precauciones al crear un CGI puede, en un momento dado, comprometer la seguridad del servidor, puesto que dentro de un CGI se puede ejecutar cualquier programa o desplegar cualquier archivo que se encuentre en dicho servidor. A pesar de lo anterior, siempre se puede deshabilitar la ejecución de CGIs y habilitar sólo ciertos directorios como mecanismo de protección. El cliente no resulta afectado en este sentido.
- Un CGI tiene a su disposición cierta información proveniente del cliente y a través de variables de ambiente se puede conocer, por ejemplo, la dirección IP y el tipo de cliente y, cuando tanto en el cliente como en el servidor se disponen de los recursos adecuados, se puede conocer el usuario de la página actual y más información, aunque esto no está disponible en el comportamiento normal, ni en todos los servidores y mucho menos en los clientes.
- Puede hacer todo lo que un programa en el servidor sea capaz de hacer, con tal de que su salida sea HTML. Por ejemplo, puede generar archivos de texto, gráficos o sonido, acceder a bases de datos, llamar a otros CGIs o HTMLs y mezclar todo ésto.
- Al igual que casi todo en el Web, los CGIs no pueden funcionar síncronamente con el usuario, pues tienen que esperar a que el cliente los mande a ejecutar.
- Por supuesto, un CGI se puede combinar con otros recursos del Web, como recarga dinámica, formas, mapas sensitivos, javascript, etc.

Para poder efectuar una transacción HTTP, existen diferentes formas, las más importantes son:

Recuperación de información del Servidor:

- Método GET. Este método es usado comúnmente en los pedidos de los navegadores, en los casos en que sólo se va a solicitar información residente en el servidor Web y no se envían datos para realizar algún proceso en el servidor.



- Método HEAD. En este método, la respuesta del servidor no incluye el contenido de la página, sino información relevante sobre ella, es decir, se indica el contenido y el tipo requerido. Si no se especifica ningún parámetro, adopta el valor predeterminado 'text/html'.

Envío de información al Servidor:

- Método POST. Este método permite que el programa CGI reciba la entrada codificada del formulario a través de STDIN (entrada estándar). Es decir, la información del formulario es agregada al propio URL.
- Método GET. Este método permite que el programa CGI reciba la entrada codificada de la variable de ambiente QUERY_STRING.

1.13 El lenguaje de programación Perl

1.13.1 DEFINICIÓN DE PERL

Perl (Practical extraction and report lenguaje) tiene muchos usos especialmente en tareas de administración en sistemas UNIX, que es donde Perl nació y se desarrolló. En la actualidad se ve a Perl como el lenguaje ideal de desarrollo para scripts de servidor Web.

Existen muchas razones por las que se desea utilizar un lenguaje como Perl. Es pequeño, eficiente, flexible y robusto, además *Perl es completamente gratuito*.

Perl es un lenguaje interpretado escrito por **Larry Wall**, que se deriva del lenguaje **C**, además de otras herramientas y lenguajes como: awk, sed, Bourne Shell, grep, etc que facilitan la manipulación de procesos, archivos y texto, así como desarrollo de utilerías para el sistema operativo, herramientas de software, tareas relacionadas con la administración de sistemas, manejo de bases de datos, programación de gráficos, redes y del World Wide Web.

Perl surgió para abreviar el tiempo de desarrollo de múltiples tareas. Perl es una utilería que pretende facilitar el proceso de grandes volúmenes de información sin limitar su rendimiento.

Al ser Perl un lenguaje interpretado y no compilado, se hace referencia a sus programas como *scripts*, siendo un script un programa interpretado al momento de ejecutarse.

TESIS CON
FALLA DE ORIGEN

1.13.2 SINTAXIS BÁSICA DE UN PROGRAMA EN PERL

Estrictamente hablando, perl no requiere de una sintaxis complicada para el desarrollo de scripts. Típicamente, perl es usado para interpretar los scripts. Los scripts se pueden ejecutar explícitamente con Perl de la siguiente forma:

```
% perl nombre_del_script  
ó  
% cat nombre_del_script | perl
```

Algunos aspectos importantes a considerar para escribir un programa en Perl son :

- Perl es un lenguaje de formato libre en el cual los espacios en blanco no importan. A mucha gente le gusta usar indentaciones, pero se puede escribir todo el script en una sola línea.
- Por convención, los archivos de perl finalizan con la extensión *.pl*, aunque esto no es una regla.
- Todos los enunciados de Perl terminan con punto y coma (;), como en lenguaje C.

Los comentarios:

- Comienzan con un signo de libra (#)
- Cualquier cosa después del #, y hasta el final de la línea es ignorada
- El # necesita estar al principio de la línea. No hay manera de poner bloques de comentarios en el código más que poniendo una # al principio de la línea. Esto es una convención tomada de los shells scripts
- Los bloques de código de Perl, tales como los ciclos de control y las condiciones siempre deben encerrarse entre corchetes ({...}).

Active Perl es un Intérprete del lenguaje PERL para Windows 95/98/NT/Me/XP, así como para Linux y Solaris.

1.13.3 PERL Y SQL SERVER

MSSQL::Dblib y MSSQL::SqlLib son dos clases en Perl para acceder Microsoft SQL Server de Perl.

MSSQL::DBlib es un modulo extendido de Perl (5.003 o superior) para llamar a Microsoft SQL Server a través de la librería de base de datos.

TESIS CON
FALLA DE ORIGEN

La mayoría de las llamadas a las librerías de bases de datos usan constantes, para parámetros o valores de retorno. Es posible importar esto de los módulos MSSQL::DBlib::Const::* . Es posible utilizar el modulo que trae las constantes necesarias, o usar MSSQL::DBlib::Const, el cual trae todos.

MSSQL::DBlib tiene un módulo: MSSQL::SqlLib, el cual reduce los Queries a SQL Server a una sola llamada, incluyèndo errores.

1.14 Programación de CGI's para Web en Perl

De inicio, es necesario suponer una maquina con un Servidor WWW disponible y un directorio en el que se pondran los Scripts definido como fuente para los Scripts. Un CGI dispone de varias variables de ambiente.

Las variables de ambiente son la forma que tiene el servidor de comunicarse con el programa CGI, la más importante es QUERY_STRING que, cuando se utiliza con formas Web y metodo GET, registra los datos introducidos en la forma. Si se usa el metodo POST, estos datos pasan al CGI por la entrada estandar y el número de caracteres enviados se registra en la variable de ambiente CONTENT_LENGTH.

Las variables de ambiente se pueden clasificar en:

Variables de ambiente basadas en el Servidor

Algunas de las variables de ambiente siempre están definidas y no son independientes de la solicitud CGI; definen típicamente el servidor en el que se ejecuta el programa CGI, por ejemplo:

SERVER_NAME

Nombre de dominio del servidor, o en su defecto, el número IP del servidor.

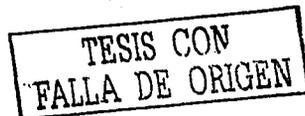
Variables de ambiente basadas en las cabeceras de solicitud

Estas variables informan al programa CGI de lo que está sucediendo durante la llamada al programa. Se definen cuando el servidor recibe las cabeceras de solicitud desde una pagina Web.

CONTENT_LENGTH

Indica la cantidad de datos adjuntos después del final de las cabeceras de solicitud.

Se usa con el método POST.



CONTENT_TYPE

Tipo de información que va a llegar.

1.14.1 BIBLIOTECA CGI.PM

La biblioteca CGI.pm de perl 5 utiliza objetos para crear formularios Web y verificar la sintaxis del contenido de los mismos. Ofrece una interfaz simple para verificar e interpretar cadenas de consulta para las secuencias de órdenes CGI. Proporciona además un conjunto de funciones para crear formas HTML. Una ventaja es que el valor de la consulta previa se emplea para inicializar la forma, lo que permite preservar el estado del formulario entre una y otra invocación.

Todo se realiza mediante un objeto CGI. Cuando se crea uno de estos objetos, este examina el entorno en busca de una cadena de consulta, la verifica y almacena los resultados, entonces se puede solicitar ya al objeto CGI que devuelva o modifique los valores de la consulta. Los objetos CGI gestionan correctamente los métodos Post y Get, además de distinguir correctamente secuencias de órdenes ejecutadas desde documentos basados en formularios.

1.15 Microsoft Windows 2000 Server

1.15.1 DESCRIPCIÓN DE MICROSOFT WINDOWS 2000 SERVER

Microsoft Windows 2000 Server es la versión siguiente de Windows NT Server 4.0. Cuenta con servicios Web de Internet Information Server 5.0 integrados que permite alojar y administrar sitios Web. Además, estos servicios permiten alojar mas sitios Web por servidor con un gran rendimiento, debido que es posible limitar la cantidad de tiempo que una aplicación o sitio Web puede utilizar la CPU.

La característica de protección de aplicaciones de IIS permite ejecutar por separado aplicaciones Web desde el propio servidor Web, lo que evita que una aplicación pueda bloquear el servidor Web.

1.15.2 REQUERIMIENTOS DE HARDWARE

CPU compatible con procesador Pentium a 166 MHZ o superior.
Memoria RAM 128 mínimo.
Monitor VGA o de mayor resolución

TESIS CON
FALLA DE ORIGEN

1.16 Representación esquemática Del Sistema. Escenarios

En el desarrollo de sistemas es necesario el uso de alguna metodología que permita al desarrollador entender de una mejor forma que es lo que el sistema va a realizar y como es que interactúan entre si las partes que componen al sistema, la relación que hay entre éstas y los procesos que intervienen en un momento dado para que el sistema realice las funciones que se esperan. Además, da una visión general de los Scripts o códigos que se usaran para realizar una acción específica, como validar por ejemplo. Por esta razón se considera conveniente hacer una representación esquemática del sistema.

Para ello consideraremos algún tipo de simbología, así como los conceptos de escenarios y excepciones que esquematicen un planteamiento de la interfaz Web. Comenzaremos con definir los conceptos de escenario y excepción.

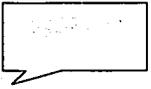
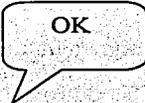
Un **escenario** nos ayuda a entender la relación que existe entre páginas Web teniendo como resultado los diferentes comportamientos que surgen al momento de interactuar de una página a otra. Entendiendo como comportamientos las distintas acciones que se pueden presentar en una página.

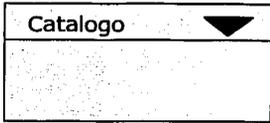
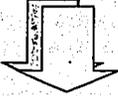
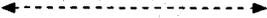
Una **excepción** es aquella que surge cuando existe algún tipo de interferencia o error inesperado que puede ser o no producido por el sistema al momento de que el usuario realice un envío de información que el mismo tecleo.

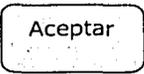
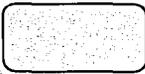
A continuación se define la simbología que utilizaremos para hacer la representación esquemática de los escenarios y excepciones para el Sistema de cómputo para consultar, alimentar y explotar un diccionario de Términos en Web

Simbología propuesta.

SÍMBOLO	DESCRIPCION
	<p>Página web estática. Es una página que presenta información estática (sin cambios) para el usuario, (formularios, imágenes, frames, etc.) o presenta opciones de que permitan navegar al usuario a través del sistema</p>
	<p>Escenario. Conjunto de páginas que nos permite analizar los diferentes comportamientos que se tienen en el sistema. Sólo aparece cuando se hace la definición del mismo.</p>

	<p>Despliegue de Liga. Representación que se hace para indicar el traslado de una página Web a otra.</p>
	<p>Mensaje Página web. Es el mensaje que presenta una página Web que puede ser o no dinámica al momento de realizar algún tipo de operación realizada por parte del usuario.</p>
	<p>Acceso de Información. Indica que la información proporcionada por el usuario entrará a la base de datos para ser dada de alta o bien para verificar si esta existe.</p>
	<p>Botón. Es una propiedad de la página Web que presenta una opción a ser ejecutada por el usuario.</p>
	<p>Mensaje de Java Script. Es un mensaje dinámico proporcionado por el sistema, si la información que se envía (mediante formularios) es válida para el sistema.</p>
	<p>Liga. Es una propiedad de la página Web que permite al usuario ir al escenario que se indica en el texto sobre la línea (hipertexto). Aparece el punto arriba de la línea si hay más de una liga, exceptuando listas de resultados y paginación.</p>
	<p>Caja de Selección. Propiedad de la página Web que permite seleccionar alguna opción de interés para el usuario.</p>

	<p>Catálogo. Propiedad de la página Web que permite al usuario elegir una sola opción a través de una cortina desplegable que presenta un conjunto de opciones que son de interés para el usuario.</p>
	<p>Propiedad de la Página. Representación que se utiliza para indicar qué elementos son parte de una página Web.</p>
	<p>Entrada de escenario con login y password. Representación que se hace para indicar al usuario que debe de introducir su login y password para poder tener acceso a una página Web.</p>
	<p>Entrada de escenario normal. Representación que se utiliza cuando el usuario desea entrar a un conjunto de páginas Web que están relacionadas entre si.</p>
	<p>Salida de escenario. Representación que se utiliza para indicar al usuario la salida hacia un escenario especificado por el texto que le sigue. Nota: Este símbolo representa una y solo una salida en un escenario.</p>
	<p>Regresar a la misma página. Es una propiedad de la página Web que permite al usuario ir a la misma página cuando se ha ejecutado alguna acción.</p>
	<p>Mensaje de Java Script. Es un mensaje que aparecerá dentro de la página Web con el fin de proporcionar algún tipo de información útil para el usuario.</p>

	<p>Botón de aceptación. Propiedad de la página Web que permite al usuario confirmar la información que se presenta en la página.</p>
	<p>Botón de cancelación. Propiedad de la página Web que permite cancelar algún tipo de información que en el instante no es de interés para el usuario.</p>
	<p>Botón de limpiar. Propiedad de la página Web que permite limpiar el formulario de una página Web después de haber metido información en este.</p>
	<p>Botón. Propiedad de la página Web que permite ejecutar una acción determinada indicada por el nombre del botón</p>
	<p>Botón opcional. Propiedad de la página Web que aparecerá sólo si se cumple la o las condiciones especificadas por el texto aunado.</p>
	<p>Ejecución de un Proceso. Se ejecuta un Script o código que realiza alguna búsqueda, elección, inserción o modificación en la base de datos del sistema.</p>
	<p>Página Web Dinámica. Es una página que puede generar y presentar con un mismo código, varios resultados, dependiendo de los valores que le hayan sido dados.</p>

TESIS CON
FALLA DE ORIGEN

<table border="1" style="width: 100%; height: 100%;"> <tr> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">Autenticación Windows 2000</td> </tr> </table>		Autenticación Windows 2000	<p>Seguridad de Windows 2000. Es un proceso que realiza el servidor de Windows 2000, para autenticar login y password. Valida en el servidor la cuenta ingresada y regresa un mensaje de error en caso de no validación.</p>
	Autenticación Windows 2000		
	<p>Excepción. Se refiere a un proceso que surge cuando existe un error inesperado, debido al tecleo de datos por parte del usuario. El símbolo # representa el número de excepción por escenario</p>		
	<p>Conector. Representa la conexión que se hace con otra parte del diagrama que se encuentre ubicado en otra página. El # representa el número de conector.</p>		

1.17 Descripción de fichas técnicas

Como una forma de sintetizar la información contenida en una página Web, es necesario la utilización de descripciones breves que permitan dar información con la cual se pueda construir la página conociendo los elementos con los que cuenta y la acción que éstos generan, así como saber qué código (Scripts) puede ser reutilizable.

Una ficha técnica es la descripción general del contenido de una página Web. Sirve para mostrarnos las funciones de la página, sus acciones y su interacción con otras páginas, así como los Scripts que las generan.

Dentro de las fichas técnicas se describirán las páginas estáticas, los Scripts que generarán las páginas dinámicas. Para esto se sigue la siguiente descripción:

Una **PÁGINA ESTÁTICA** contendrá las siguientes descripciones:

Nombre: Se refiere al nombre de la página Web estática

Entrada: Hace referencia al nombre o los nombres de las páginas de donde la página estática proviene.

Propiedades: Son los elementos con los que cuenta la página Web. Se coloca el nombre de la propiedad, el nombre particular de cada elemento y la función del mismo.

TESIS CON
FALLA DE ORIGEN

Un *SCRIPT* contendrá las siguientes descripciones:

Nombre: Se refiere al nombre del Script.

Descripción: Se refiere al o los procesos que el código genera y hace referencia al nombre de la página dinámica que es generada por el Script.

Parámetros que recibe: Son los datos que el Script recibe para ejecutar el o los procesos del Script.

Proviene de: nombre de la página dinámica, página estática u origen de los parámetros recibidos

Procedimientos almacenados: aquí se menciona el nombre del procedimiento almacenado que es utilizado en este script y su descripción.

Una *PÁGINA DINÁMICA* contendrá las siguientes descripciones:

Nombre: Se refiere al nombre de la página Web dinámica.

Generada por: Hace referencia al nombre o los nombres de Scripts de donde surge la página dinámica.

Propiedades: Son los elementos con los que cuenta la página Web. Se coloca el nombre de la propiedad, el nombre particular de cada elemento y la función del mismo.

Un *PROCEDIMIENTO ALMACENADO (Store Procedures)* contendrá las siguientes descripciones:

Nombre: Se refiere al nombre del procedimiento almacenado.

Descripción: Se describe qué acción ó acciones realiza.

Parámetros que recibe: Se refiere al nombre ó nombres de las variables que recibe el procedimiento almacenado.

Generada por: Hace referencia al nombre o los nombres de Scripts que lo utilizan.

1.18 Recursos

El Hardware disponible es una computadora Acer con procesador Pentium III a 350 MHz con 256 Mb en RAM, disco duro de 4 Gb.

El sistema necesita como mínimo computadoras PC 80386 con 16 Mb en RAM y que tengan el visualizador de Microsoft Internet Explorer.

TESIS CON
FALLA DE ORIGEN

1.19 Resumen del capítulo

El sistema que se desarrollará necesita de una base de conocimientos de diversos aspectos, desde nociones teóricas: Ingeniería de Software, bases de datos, Redes, etc., hasta tecnologías para la implementación de servidores de sistema operativo, servidores de bases de datos, servidores Web, así como tecnologías para la integración de bases de Datos en el Web.

Estos aspectos han sido explicados en este capítulo de manera breve, procurando mencionar los aspectos básicos e importantes que puedan dar una idea general de las herramientas y conocimientos que se aplicarán en el desarrollo del sistema.

CAPÍTULO 2 ANÁLISIS Y DISEÑO

2.1 Marco Histórico

En el instituto de Ingeniería se ha impulsado el desarrollo de proyectos en el área de Ingeniería Lingüística, donde participan tanto lingüistas como ingenieros en computación.

La ingeniería lingüística es el área interdisciplinaria de investigación aplicada al desarrollo de sistemas computacionales para reconocer, interpretar y generar lenguaje humano. Existe una correspondencia biunívoca, de forma que la lingüística permite la creación de modelos en lenguaje natural que puedan ser utilizados por los sistemas computacionales, mientras que la ingeniería permite el desarrollo de sistemas que puedan resolver las necesidades específicas planteadas por los problemas lingüísticos.

Con el fin de desarrollar formalmente el área de ingeniería lingüística en la UNAM y a nivel nacional, el Instituto de Ingeniería abre, en septiembre de 1999, el Grupo de Ingeniería Lingüística.

El Grupo de Ingeniería Lingüística, en el seno del Instituto de Ingeniería de la UNAM, se conformó con el fin, primero, de crear una base de conocimiento relativa y concerniente a esta área de trabajo, y segundo, de formar personal especializado y comprometido con el estudio y desarrollo de las diversas áreas que ésta ofrece.

Gracias al apoyo del Instituto de Ingeniería y al patrocinio del Consejo Nacional de Ciencia y Tecnología, entre otros, y de la propia UNAM, el Grupo de Ingeniería Lingüística ha venido realizando proyectos vinculados con el procesamiento de lenguaje natural.

Actualmente, se está desarrollando un proyecto donde se elabora un diccionario integral para búsquedas semasiológicas y onomasiológicas en el área de lingüística.

Según el diccionario de la real academia Española vigésimo segunda edición, la definición de diccionario es, textualmente:

1. m. Libro en el que se recogen y explican de forma ordenada voces de una o más lenguas, de una ciencia o de una materia determinada.

2. m. Catálogo numeroso de noticias importantes de un mismo género, ordenado alfabéticamente. Diccionario bibliográfico, biográfico, geográfico.

Una definición más general menciona que un diccionario es un repertorio de unidades léxicas, que comprende el conjunto de información semántica, de noción y gramática.

De acuerdo a las definiciones anteriores, y adcuándolo a las necesidades de los proyectos enfocados a la ingeniería lingüística, podemos decir que un diccionario es un material léxico o terminológico que surge para dar respuesta a algunas necesidades lingüísticas de un grupo de usuarios definidos.

El proyecto mencionado anteriormente, como su nombre lo indica, busca desarrollar un diccionario que permita realizar búsquedas semasiológicas y onomasiológicas.

Antes de definir lo que son las búsquedas semasiológicas y búsquedas onomasiológicas, se mencionará a continuación el significado de las palabras que se utilizarán inmediatamente después de estas definiciones:

Semántica (según el diccionario de la real academia española):

Estudio del significado de los signos lingüísticos y de sus combinaciones

Semasiología (según el diccionario de la real academia española):

Estudio semántico que parte del signo y de sus relaciones, para llegar a la determinación del concepto

Onomasiología (según el diccionario de la real academia española):

Rama de la semántica que investiga los significantes que corresponden a un concepto dado.

Desde un punto de vista semántico, una búsqueda semasiológica parte de la palabra o término para conocer el significado del mismo.

Asimismo, desde este mismo punto de vista, en una búsqueda onomasiológica se parte de palabras clave que pudieran corresponder a un término determinado.

Como se sabe, un diccionario tradicional (impreso) permite al usuario que lo consulta búsquedas muy simples: sólo es posible buscar la palabra o término del cual se desea conocer su significado, por lo que se condiciona y se restringe la forma de presentación de la información.

Como una solución por la necesidad de tener búsquedas más complejas, así como de aprovechar al máximo los recursos electrónicos disponibles en estos tiempos, surgen en Internet los diccionarios en línea.

Podemos encontrar diccionarios con una interfaz muy simple: el usuario escribe la palabra que está buscando y obtiene la entrada correspondiente, es decir, de la forma tradicional, en la que es posible consultar la versión comercial de un diccionario ya existente, ya sea a través de una suscripción (como por ejemplo, la enciclopedia Británica), o de acceso público, y es posible buscar todo lo relacionado con algún término en específico.

Algunos otros diccionarios tienen esta misma estructura pero están enfocados únicamente a un área en particular, etc.

Como podemos ver, en Internet las posibilidades de consulta son muy distintas, por lo que constituye uno de los entornos que facilita más la flexibilidad de los diccionarios, y constituye actualmente una de las formas más novedosas de mostrar, actualizar y modificar información.

Para poder llevar a cabo el proyecto mencionado, y debido a que muchas personas están involucradas en el mismo, fue necesario dividir el desarrollo del proyecto en etapas.

Una de las primeras etapas del proyecto consiste en la recopilación de la terminología en el área de lingüística que se va a incluir en el diccionario, de forma tal que la información esté disponible y sirva como base de trabajo para todos los que están involucrados en el proyecto.

2.2 Requerimientos y Necesidades en la elaboración del Sistema

Debido a la necesidad de recopilar la terminología en el área de lingüística, el proyecto "Desarrollo de un sistema para alimentar y consultar un diccionario terminológico Web", pretende cubrir esta necesidad de tener una base de datos que permita recopilar los términos en el área de lingüística y sus definiciones según diversas fuentes.

Este proyecto permitirá a los usuarios introducir la terminología en el área de lingüística, así como sus definiciones, las fuentes consultadas, los sinónimos de cada término, y toda la información relevante que sea requerida por los usuarios del sistema.

Por lo tanto, es necesario definir en esta etapa de análisis los requerimientos y necesidades del sistema a desarrollar, para poder definir de manera clara y precisa el sistema requerido.

Es necesario que el sistema a desarrollar permita a los usuarios introducir la terminología en el área de lingüística de forma ágil, ya que los usuarios en ocasiones requerirán introducir en una sola ocasión bastante información a la base terminológica.

Cabe mencionar que las personas que conforman el grupo de ingeniería lingüística no pertenecen a la misma institución educativa, y tampoco cuentan con el mismo nivel de conocimientos en cuanto a Tecnologías e Internet, por lo que era necesario considerar que el sistema sea sencillo y tenga una interfaz simple, ya que a través de él se planea no sólo consultar la información contenida en la base de datos, sino que también se requiere que los participantes del proyecto puedan realizar las manipulaciones de la información necesarias.

Por lo antes mencionado, tampoco es posible contar con un lugar de trabajo común que sea el adecuado en cuanto a ubicación, por lo que es necesario plantear un sistema que sea de fácil acceso para todos los involucrados en el proyecto, y es necesario considerar un ambiente de trabajo que facilite el acceso al sistema.

El desarrollo de un diccionario electrónico permitirá tener acceso al sistema a través de la Web, tanto para consultar la terminología en el área de lingüística, así como agregar y/o modificar información a la base de datos, además de que resolver la situación planteada anteriormente.

Para que todos los usuarios que entren al sistema, puedan navegar a través de las páginas sin ninguna dificultad, la interfaz deberá de ser amigable de tal forma que permita ligas desde cualquier página donde se esté consultando al sistema para poder trasladarse así a otras ligas del sistema, sin que los usuarios lleguen a sentirse engorroso el manejo de la interfaz Web.

Para la presentación de las páginas, y para obtener una mayor facilidad de acceso a la información contenida en la base de datos, los responsables del proyecto esperan que el sistema muestre de forma separada la información contenida en la base de datos: términos, definiciones, área temática, fuentes y sinónimos, de la siguiente manera:

Para la terminología, se mostrará el término que se desea consultar, el área temática a la que pertenece, el responsable de la captura del término y la fecha de captura.

TESIS CON
FALLA DE ORIGEN

Para las definiciones, se mostrará el término del cual se desea consultar su definición o definiciones, el área a la que pertenece, la definición o definiciones, contexto de uso, la fuente o las fuentes consultadas, el responsable de la captura de cada definición y la fecha de captura.

Para el área temática se mostrará el área temática que se desea consultar, el responsable de la captura del área y la fecha de captura.

Para las fuentes, se mostrará el nombre de la fuente consultada, editor o editores, lugar de la publicación, año de la publicación, liga electrónica si cuenta con ella, el responsable de la captura y la fecha de captura.

Para los Sinónimos, se mostrará el sinónimo que se desea consultar, el término o los términos que tienen una relación sinonímica, así como el Responsable de la captura y la fecha de captura.

Además, se espera que el sistema muestre una lista con todos los términos, fuentes consultadas, así como cada parte de la información mencionada anteriormente, por separado, de manera que la información aparezca de forma ordenada y asimismo facilite la búsqueda de la información en el sistema.

Se acordaron varias reglas del sistema, considerando las necesidades planteadas en el proyecto, y en base a pláticas y entrevistas con los responsables del proyecto, las cuales se mencionan a continuación:

- Para dar de alta un término, es necesario que exista previamente el área temática a la que pertenece.
- Para dar de alta una definición, es necesario que exista previamente la fuente consultada.
- Puede haber varias definiciones con una misma fuente consultada.
- Cada término puede tener más de una definición.
- Cada término puede tener más de un sinónimo.
- Cada área temática puede pertenecer a varios términos.
- Cada definición pertenece a un solo término.
- Un sinónimo es también un término.

- Para que un término pueda ser sinónimo de otro término, es necesario que exista previamente el término.
- Al agregar un sinónimo, se forma una relación sinonímica con el otro término, pero también sucede lo mismo a la inversa (es decir, cada uno de los términos será sinónimo del otro término).

Como la página estará disponible para cualquier persona que quiera entrar a ella, es necesario que los responsables del proyecto tengan conocimiento de quién está haciendo modificaciones a la información contenida al sistema. Para asegurar que se tendrá un control sobre quién hace algún cambio y cuando, se registrará la fecha de los cambios y modificaciones, así como el responsable de éstos.

Por lo anterior, se requiere también cada participante tenga asignado sólo determinados permisos de acuerdo a lo que va a realizar en el proyecto. Los permisos van a ser asignados por los responsables del proyecto, y se les otorgará a cada participante nombre de usuario y una clave con la finalidad de controlar y supervisar las acciones de los usuarios sobre las bases.

Para facilitar el acceso, se planteó que era necesario entonces crear diferentes tipos de usuarios, de tal forma que sólo los responsables y desarrolladores del proyecto tengan acceso a ver toda la información, así como modificar o eliminar, mientras que los demás tipos de usuarios tuvieran acceso sólo a la información que les sea necesaria.

Los tipos de usuarios planteados son:

- Usuarios de consulta.
- Los usuarios o Participantes del Proyecto.
- Los responsables y desarrolladores del proyecto.

Los usuarios de consulta sólo podrán consultar la información. Este tipo de usuarios pueden modificar, agregar o eliminar alguna información, y por seguridad del sistema, al usuario de consulta no se le informará que es posible realizar dichos cambios.

TESIS CON
FALLA DE ORIGEN

Los usuarios o desarrolladores del proyecto pueden consultar, modificar y agregar información a la base de datos.

Cada Modificación que realice un usuario se registrará con su nombre de usuario y la fecha de modificación.

Como requerimiento de los responsables del proyecto, el usuario no podrá hacer modificaciones en las fuentes consultadas ni en el área temática. Si es necesario que haya algún cambio en la información, deberá informarle al responsable del proyecto para que éste tome la decisión de hacer los cambios necesarios en la base de datos.

También como solicitud de los responsables del proyecto, estos usuarios no van a poder eliminar información. En caso de que lo requieran, el sistema deberá de mandar un correo al responsable del proyecto, quien decidirá si es posible que se elimine. Esto con la finalidad de no permitir que algún usuario pueda eliminar alguna información valiosa para el sistema sin el conocimiento de los responsables del proyecto.

En cambio, los responsables del proyecto podrán tener acceso total a toda la información contenida en la base de datos. Además de consultar, modificar y eliminar información, también podrán ver quién ha hecho y cuando alguna modificación, ya que se les mostrará junto con cada término, fuente, etc., el usuario responsable de la captura, consulta, etc., junto con la fecha de modificación.

Las opciones de modificar, eliminar, así como la información del responsable de la captura o modificación junto con la fecha de dicha acción sólo estará disponible para los responsables y desarrolladores del proyecto.

2.3 Consideraciones y Expectativas en la elaboración del Sistema

El sistema permitirá al usuario capturar ágilmente la información, por lo que será posible tener en la base de datos un fácil manejo de la información que permita posteriormente transferirla para integrarla en un diccionario semasiológico.

Además, el sistema admitirá múltiples usuarios en un mismo tiempo sin restricciones, considerando que es posible entrar al sistema siempre que se cuente con una computadora personal con acceso a Internet.

TESIS CON
FALLA DE ORIGEN

Se requiere utilizar un servidor Web que permita tener la seguridad de que la información de la base de datos, los Scripts, etc., no van a ser susceptibles de admisiones sin los permisos correspondientes.

Es necesario tomar en cuenta la importancia de realizar respaldos diarios de la información, para evitar posibles pérdidas o errores en la modificación de los términos o sus definiciones.

Para contribuir con la seguridad y facilidad del sistema, se requiere desarrollar un sistema de administración de usuarios, donde el administrador podrá dar de alta a éstos, otorgándoles los permisos de consulta y/o modificación asignado a cada uno de ellos, así como la(s) base(s) a la(s) que se les da derecho a entrar.

Asimismo, para evitar que cualquiera pueda tener acceso a este sistema de administración, se utilizará otro número de puerto diferente al de la interfaz del sistema.

Se espera obtener un sistema capaz de proporcionar un fácil acceso a la base de datos a través del Web, confiable y seguro, permitiendo que sólo los usuarios autorizados alimenten la base de datos de manera simultánea, sin restricción de acceso y que a la vez, permitan la consulta a toda persona que visite la página Web.

El sistema, además, puede ser modificado en base a las necesidades específicas de algún usuario o usuarios en particular que puedan llegar a surgir, por lo que el sistema puede servir de base para el desarrollo de diccionarios en otras áreas.

Se pretende que este sistema también permita posteriormente que se puedan consultar bases de datos de otras áreas, si así lo requieren los responsables del proyecto o algún otro usuario interesado en ello.

2.4 Herramientas a utilizar

Para el desarrollo del sistema planteado, se planteó utilizar las herramientas que se mencionan a continuación. Cabe mencionar que, en cada caso, se mencionó una herramienta o herramientas a utilizar junto con una breve descripción de ésta y la justificación de la elección:

Para el servidor del sistema, se pretende ocupar el software que está disponible en el Instituto de Ingeniería, por lo que el servidor de la máquina a utilizar será Microsoft Windows 2000 Server.

TESIS CON
FALLA DE ORIGEN

Microsoft Windows 2000 Server es la versión siguiente de Windows NT Server 4.0. Y cuenta con servicios Web de Internet Information Server 5.0 integrados que permite alojar y administrar sitios Web.

Se utilizara Microsoft SQL Server 2000 como servidor de la base de datos. Para desarrollar la base de datos, se pueden utilizar varios lenguajes, de los que SQL (Structured Query Lenguaje) es el más común. El dialecto de SQL compatible con Microsoft SQL Server se llama Transact-SQL (T-SQL).

Microsoft SQL Server 2000 nos proveerá de la flexibilidad necesaria y cubre nuestras necesidades en cuanto a número de usuarios y tamaño de las bases. También cuenta con un plan de mantenimiento, que permite realizar respaldos automáticos para prever la pérdida de información.

Además, SQL Server 2000 está diseñado para funcionar como motor de almacenamiento de datos para miles de usuarios que se conectan a través de una red simultáneamente, aunque puede funcionar también como base de datos independiente directamente en el mismo equipo de una aplicación; y el motor de bases de datos relacional de SQL Server 2000 dispone de características para evitar los problemas lógicos que tienen lugar si un usuario intenta leer o modificar datos que están utilizando otros usuarios.

Se desarrollarán Procedimientos Almacenados. En este sistema, se utilizarán procedimientos almacenados para algunas opciones del sistema, como agregar, modificar y eliminar algún campo de alguno de los objetos de la base de datos.

Los procedimientos almacenados nos permitirán tener un plan de ejecución cuando el procedimiento esté corriendo, lo que permite ejecuciones rápidas aún cuando se esté trabajando en un servidor remoto.

Además, al utilizar procedimientos almacenados, se pueden detectar los errores que pueda generar el servidor de la base de datos y así mostrar de una forma amigable al usuario final el porqué no se pudo efectuar con éxito la operación que fue solicitada al sistema.

Para la realización de las páginas Web se utilizará HTML (Hyper Text Markup Lenguaje), JavaScript y un lenguaje de programación que esté relacionado con la creación de páginas Web. El lenguaje utilizado será Perl, por lo que se utilizarán Scripts en Perl que permitirán consultar, alimentar y explotar la base de datos en la Web.

La ventaja de Perl es que por su forma y por las bibliotecas y utilidades de que dispone, es ideal para la realización de CGI's, además de facilitar la manipulación de cadenas de caracteres, etc.

TESIS CON
FALLA DE ORIGEN

Para la implementación del Servidor Web, se utilizará Microsoft Internet Information Server (MIIS), que viene incluido en Microsoft Windows 2000 Server.

Microsoft Internet Information Server cubre las necesidades del sistema, ya que provee los servicios del Web necesarios, una herramienta para administrar los servicios de Internet, etc.

En dicho servidor se incluirán los directorios correspondientes a las páginas Web estáticas y un directorio de CGI'S. Cabe aclarar que estos directorios tendrán sólo ciertos permisos asignados para garantizar la seguridad de los archivos contenidos en ellos.

También se contará con un módulo de administración en donde se podrán agregar los usuarios del sistema de una manera sencilla. Este sistema de administración podrá ser utilizado a través del Web, por lo que no será necesario que el administrador o los administradores estén trabajando directamente en el servidor.

La seguridad del sistema se basará en la que viene incluida en el servidor de la base de datos, y el acceso a la página será público, considerando únicamente los permisos asignados a cada persona.

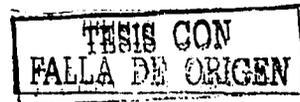
2.5 Diseño Modular del Sistema

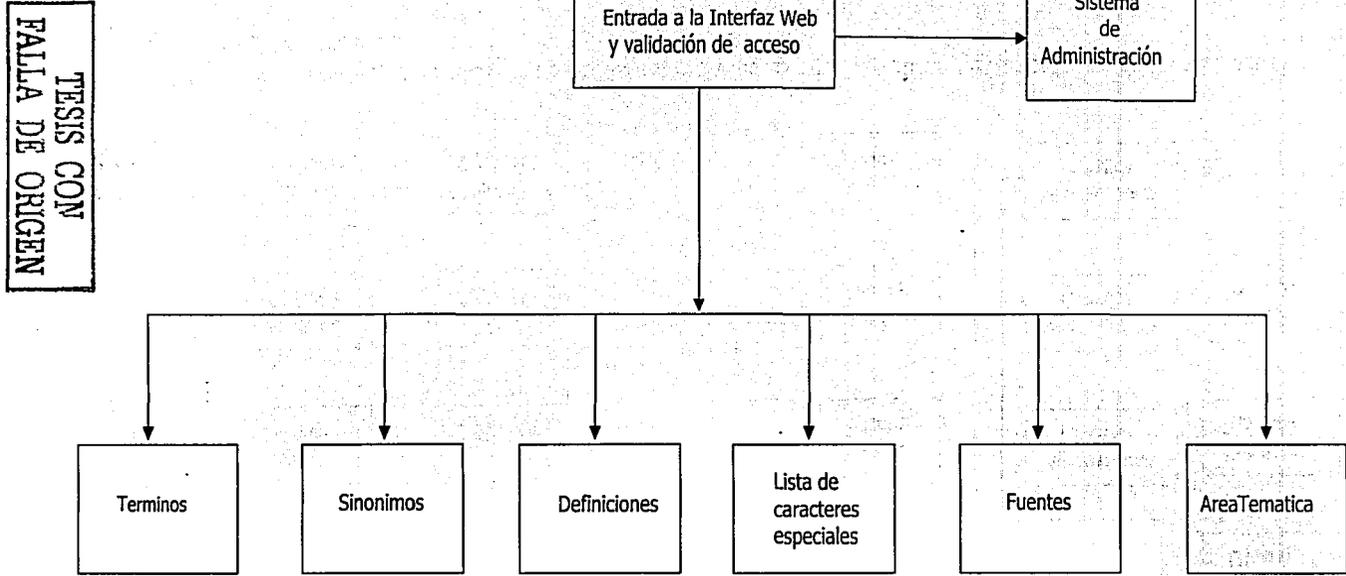
2.5.1 DIAGRAMA JERÁRQUICO FUNCIONAL DEL SISTEMA

Para explicar cómo funcionará el sistema cuando un posible usuario ingrese y realice cualquier actividad, se utiliza el diagrama jerárquico funcional del sistema.

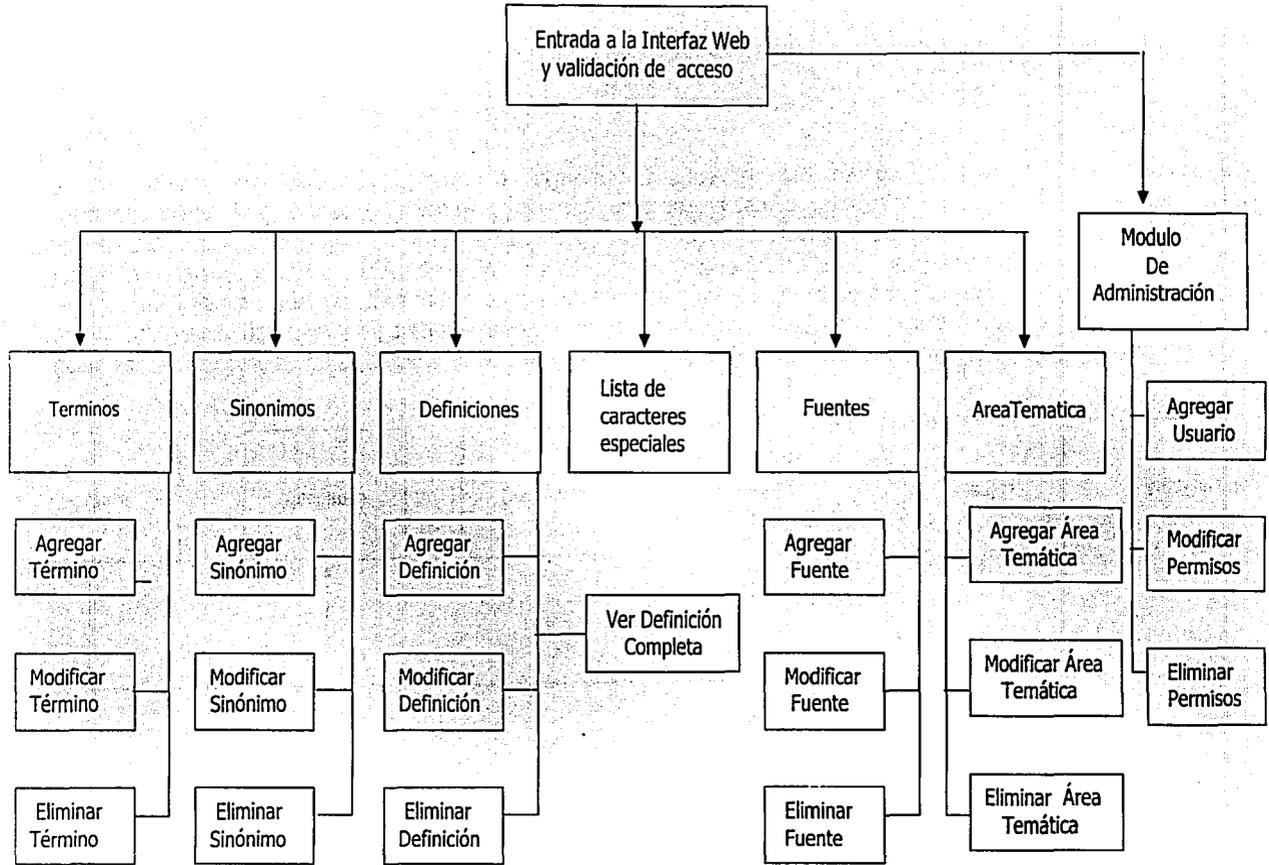
El diagrama jerárquico funcional nos va a mostrar de forma gráfica la funcionalidad del sistema; es decir, el diagrama jerárquico funcional muestra cómo están organizados los Scripts y páginas estáticas que constituyen el sistema y cómo están relacionados entre sí.

En los siguientes diagramas se muestran de manera general los módulos que conforman al sistema.





Como se puede observar, en este diagrama se muestran siete módulos importantes: el módulo de Administración, de Términos, Sinónimos, Definiciones, lista de Caracteres especiales, Fuentes y Área Temática. En los módulos de Términos, Sinónimos, Definiciones, Fuentes y Área Temática se puede insertar, modificar ó eliminar la información correspondiente a cada módulo. En el módulo de Definiciones, hay ocasiones que alguna definición de la base terminológica es demasiado extensa, por lo cual el sistema sólo permite ver una parte de ella; el módulo de definiciones agrega un módulo que permite ver a una definición de forma separada y completa. En cuanto al módulo de Administración, es posible agregar usuarios al sistema, así como modificar permisos a un usuario en determinada base ó eliminar permisos a un usuario en alguna base terminológica.



En este diagrama, se muestran los módulos principales del sistema y las acciones que puede realizar cada uno de los submódulos correspondientes a estos módulos.

TESIS CON
 FALTA DE ORIGEN

Como se puede apreciar en los diagramas mostrados anteriormente, cada módulo principal comprende a una de las partes importantes de la información contenida en la base de datos, y de cada uno de estos módulos se generan módulos independientes con los cuales es posible realizar las acciones permitidas desde la interfaz Web dependiendo del tipo de usuario.

2.5.2 DISEÑO LÓGICO DE LA BASE DE DATOS

El diseño lógico de bases de datos implica realizar el modelo conceptual de datos.

Una vez recabados y analizados todos los requerimientos que el usuario proporciona para la elaboración del sistema, el siguiente paso es crear un esquema o modelo conceptual para la base de datos mediante un modelo de datos conceptual de alto nivel.

El modelo conceptual es una descripción concisa de los requerimientos de información de los usuarios y contiene descripciones detalladas de los tipos de datos, los vínculos y las restricciones, las cuales se expresan mediante los conceptos de modelo de datos de alto nivel.

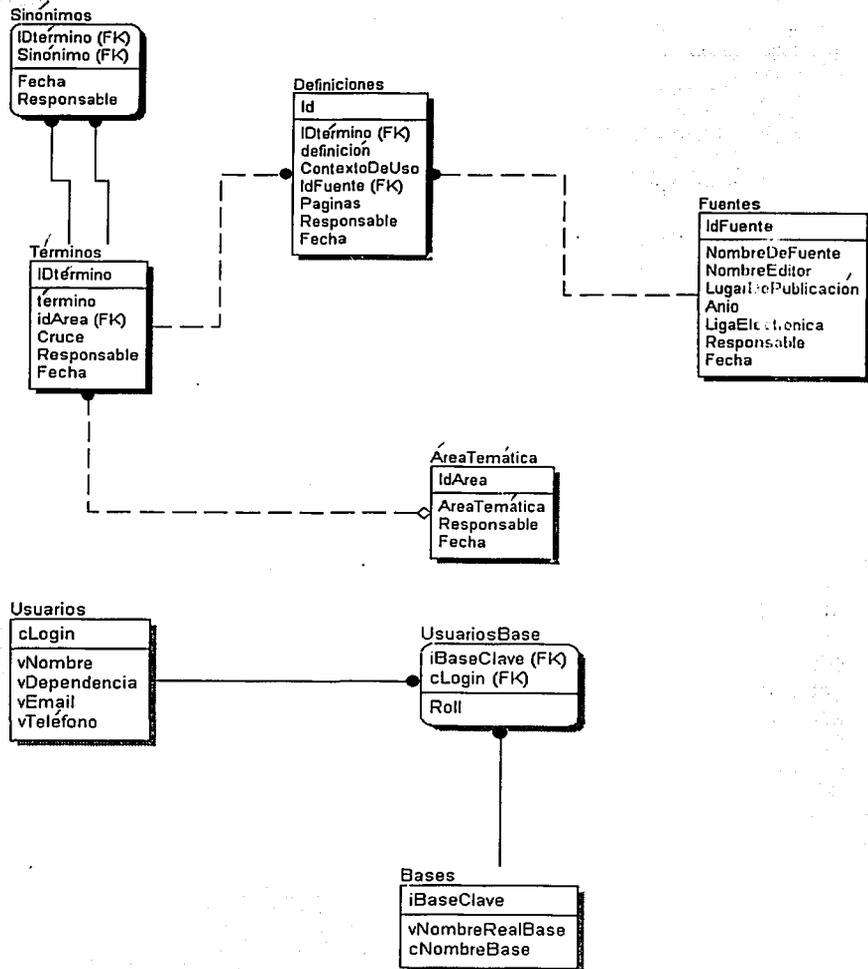
2.5.2.1 Modelo entidad - relación

El modelo entidad-relación es un modelo de datos conceptual de alto nivel muy utilizado. Este modelo se emplea en el diseño conceptual de aplicaciones de bases de datos, y muchas herramientas de diseño de bases de datos aplican sus conceptos.

Este esquema permite especificar las propiedades de los datos, sin preocuparse por detalles de almacenamiento. Su objetivo es representar la estructura de los datos y las dependencias de los mismos, garantizando la consistencia y evitando la duplicidad.

A continuación se presenta el modelo entidad- relación para el sistema planteado:





TESIS CON FALLA DE ORIGEN

2.5.2.2 Diccionario de datos

A continuación, se muestra el diccionario de datos correspondiente a la base de datos del sistema:

DEFINICIONES

Descripción	Columna	Tipo	Null/Not Null	Tipo Llave
Clave de la Definición	Id	Int	NOT NULL	Primaria
Definición	Definición	Text	NOT NULL	
Contexto de uso	ContextoDeUso	Text	NULL	
Clave del término	Idtermino	Int	NOT NULL	Foránea
Fecha de captura de la definición	Fecha	Datetime	NULL	
Clave de la Fuente	IdFuente	Char(30)	NOT NULL	Foránea
Paginas	Paginas	Char(18)	NULL	
Responsable de la captura de la Definición	Responsable	char(20)	NULL	

TÉRMINOS

Descripción	Columna	Tipo	Null/Not Null	Tipo Llave
Clave del término	Idtermino	Int	NOT NULL	Primaria
Clave del área temática	IdArea	Int	NULL	Foránea
Fecha de captura de la definición	Fecha	Datetime	NULL	
Término	termino	varchar(100)	NOT NULL	
Cruce	Cruce	varchar(100)	NULL	
Responsable de la captura de la Definición	Responsable	char(20)	NULL	

SINÓNIMOS

Descripción	Columna	Tipo	Null/Not Null	Tipo Llave
Clave del término	IDtermino	Int	NOT NULL	Foránea
Sinónimo	Sinónimo	char(100)	NOT NULL	Foránea

TESIS CON
FALLA DE ORIGEN

Fecha de captura de la definición	Fecha	Datetime	NULL	
Responsable de la captura de la Definición	Responsable	char(20)	NULL	

FUENTES

Descripción	Columna	Tipo	Null/Not Null	Tipo Llave
Clave de La fuente	IdFuente	Char(30)	NOT NULL	Primaria
Año de publicación	Anio	Smallint	NULL	
Fecha de captura de la definición	Fecha	Datetime	NULL	
Lugar de publicación de la fuente	LugarDePublicacion	varchar(100)	NULL	
Liga Electrónica	LigaElectrónica	varchar(100)	NULL	
Nombre de la fuente	NombreDefuente	varchar(100)	NOT NULL	
Nombre del editor	NombreEditor	varchar(100)	NULL	
Responsable de la captura de la Definición	Responsable	char(20)	NULL	

AREATEMÁTICA

Descripción	Columna	Tipo	Null/Not Null	Tipo Llave
Clave del área	IdArea	Int	NOT NULL	Primaria
Fecha de captura de la definición	Fecha	Datetime	NULL	
Nombre del área temática	AreaTemática	varchar(50)	NOT NULL	
Responsable de la captura de la Definición	Responsable	char(20)	NULL	

USUARIOS

Descripción	Columna	Tipo	Regla	Default	Null/Not Null	Tipo Llave
Clave del usuario	Clogin	Char(20)			NOT NULL	Primaria
Nombre del usuario	VNombre	varchar(100)			NOT NULL	
Nombre de la dependencia	VDependencia	varchar(50)			NULL	

Correo electrónico	VEmail	varchar(50)			NULL	
Teléfono	VTelefono	char(15)			NULL	

BASES

Descripción	Columna	Tipo	Regla	Default	Null/Not Null	Tipo Llave
Clave de la Base	IbaseClave	Int			NOT NULL	Primaria
Nombre de la Base en el servidor	VNombreRealBase	varchar(50)			NOT NULL	
Nombre de la Base	CNombreBase	Char(18)			NOT NULL	

USUARIOSBASE

Descripción	Columna	Tipo	Regla	Default	Null/Not Null	Tipo Llave
Clave del usuario	Clogin	Char(20)			NOT NULL	Foránea
Clave de la Base	IbaseClave	Int			NOT NULL	Foránea
Permisos del usuario en la Base	Roll	char(1)			NOT NULL	

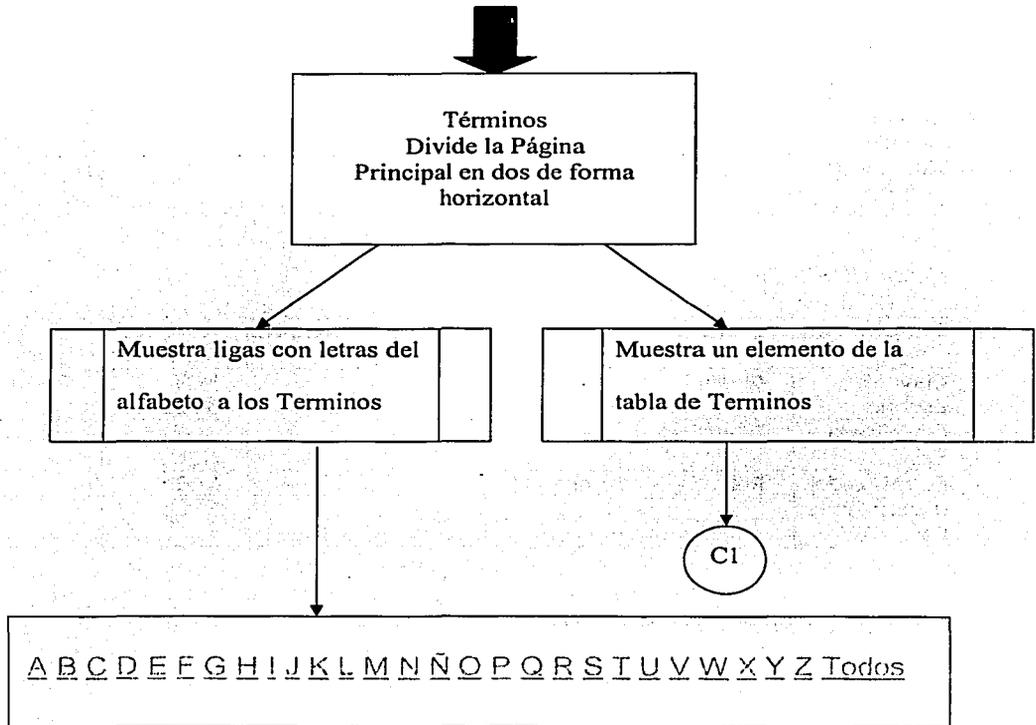
2.5.3 ESCENARIOS DEL SISTEMA

A continuación se presenta la representación esquemática de acuerdo a la simbología planteada en el capítulo uno de los escenarios más representativos del Sistema.

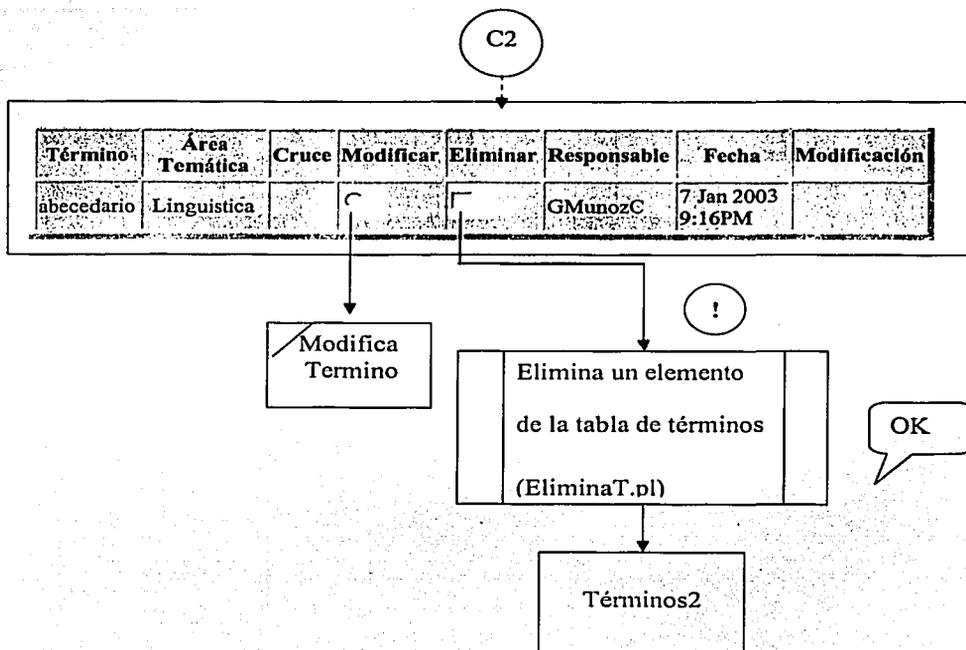
Cabe mencionar que los escenarios nos muestran todos los elementos que puede ver el responsable del proyecto, por lo que no necesariamente todo lo que se muestra lo pueden ver todos los tipos de usuarios. Todos los demás escenarios están disponibles y pueden ser consultados en el manual técnico del Sistema.

Este escenario corresponde al módulo principal de términos, este escenario muestra la página principal de esta página, que muestra los términos de la base, así como las ligas a las acciones que pueden hacerse en los submódulos.

**Escenario página Términos
(nivel 2)**



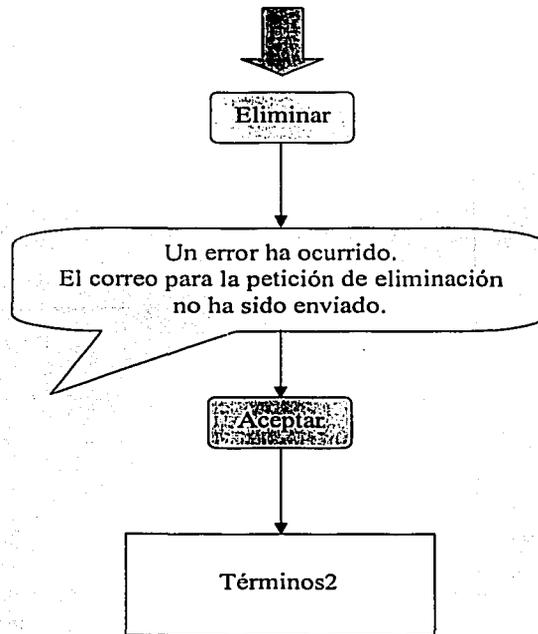
TESIS CON
FALLA DE ORIGEN



TESIS CON
FALLA DE ORIGEN

Este escenario muestra la acción que se realiza si existe un error en la página principal de Términos, ocasionado por la acción de eliminación de un término de la base.

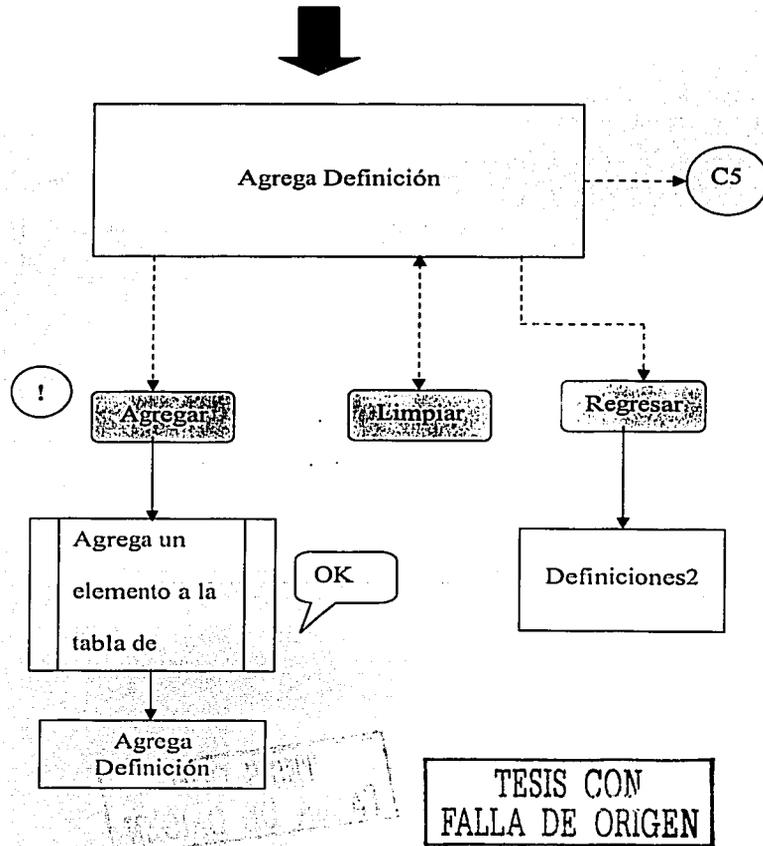
Excepción Eliminar erróneo del escenario Página Términos2

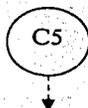


TESIS CON
FALLA DE ORIGEN

Este escenario corresponde al módulo de insertar una definición, aquí se muestra cuales son las acciones que la página permite efectuar cuando se desea agregar una definición a un término de la base.

**Escenario página Agrega Definición
(nivel 3)**





Término: abecedario

Área Temática: Lingüística

Definición:

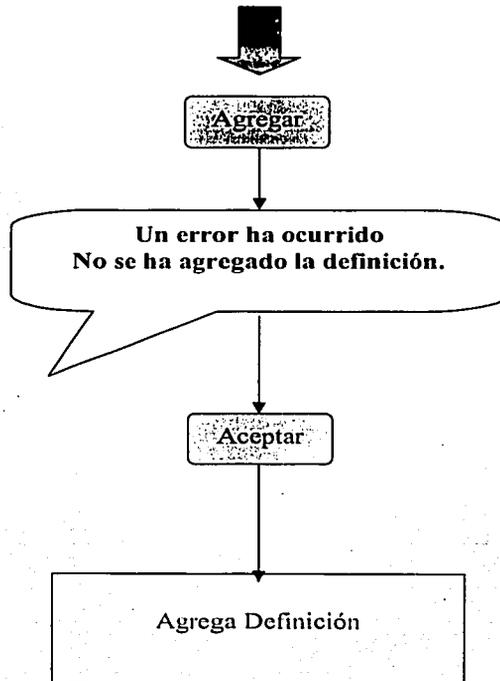
Contexto Definitorio:

Páginas:

Fuente:

Este escenario muestra la acción que se realiza si existe un error en la página de Agrega definición.

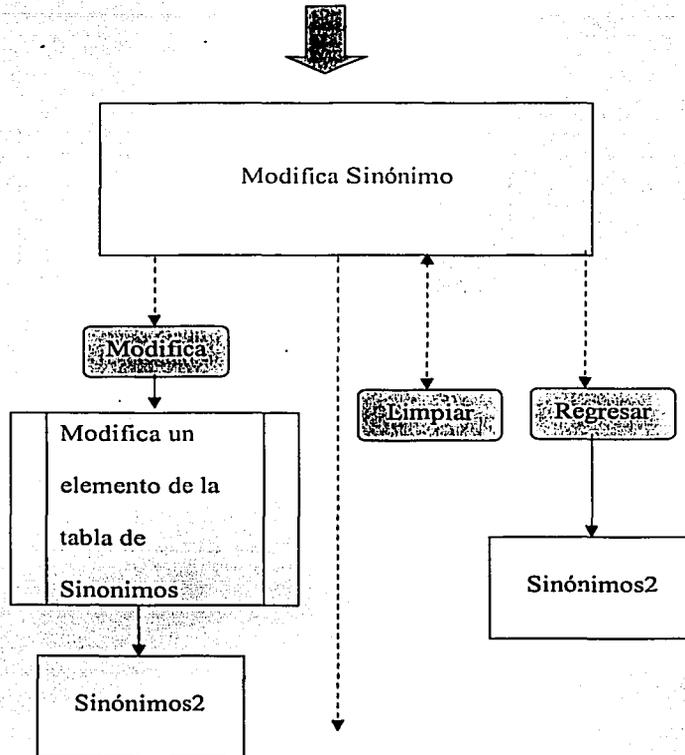
Excepción Agregar erróneo del escenario Página Agrega Definición



TESIS CON
FALLA DE ORIGEN

Este escenario corresponde al módulo de modificar un sinónimo. Aquí se muestra cuales son las acciones que la página permite efectuar cuando se desea modificar un sinónimo de la base.

**Escenario Pagina Modifica Sinónimo
(nivel 3)**



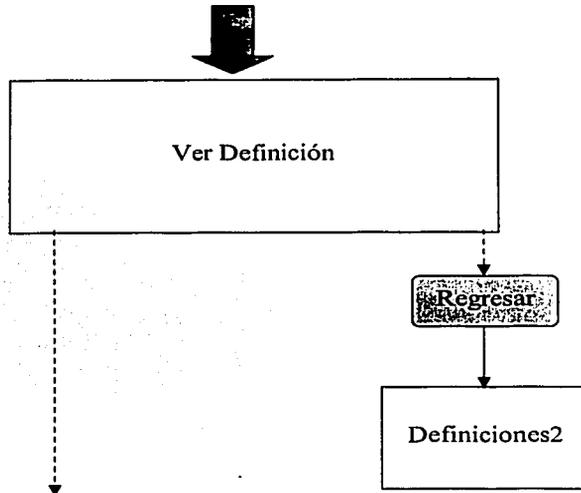
Modifique el Sinónimo:

Término: abecedario
Área Temática: Lingüística
Sinónimo:

TESIS CON
FALLA DE ORIGEN

Este escenario es un submódulo del de Definiciones. Aquí se muestra cuales son las acciones que la página permite efectuar cuando se desea ver una definición de manera separada y completa.

**Escenario Pagina ver Definición
(nivel 3)**



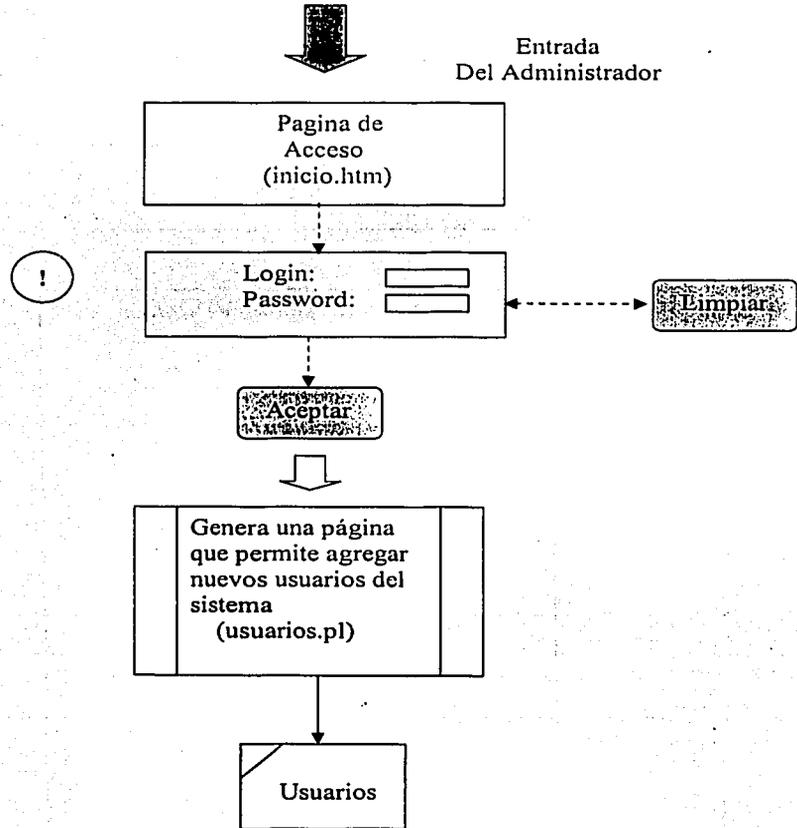
Término :	
Abecedario	
Area Temática :	
Lingüística	
Id de fuente :	
D188 Diccionario Lingüística.	
Definición :	
Serie completa y ordenada de todas las letras de una lengua.	
Contexto de Uso :	
Serie completa y ordenada de todas las letras de una lengua. Ej. El abecedario del español es: a, b, c, d, e, f, g, h, i, j, k, l, m, n, ñ, o, p, q, r, s, t, u, v, w, x, y, z. La Real Academia Española suprimió la ch y la ll desde 1995. También se le denomina ALFABETO.	
Responsable	RZacarias
Fecha	21 Nov 2002 10:29
Páginas:	

TESIS CON
FALLA DE ORIGEN

Este escenario corresponde al módulo principal de Administración. Esta página permite el acceso al sistema de Administración.

Escenario Pagina Acceso Sistema Administración

(nivel 0)

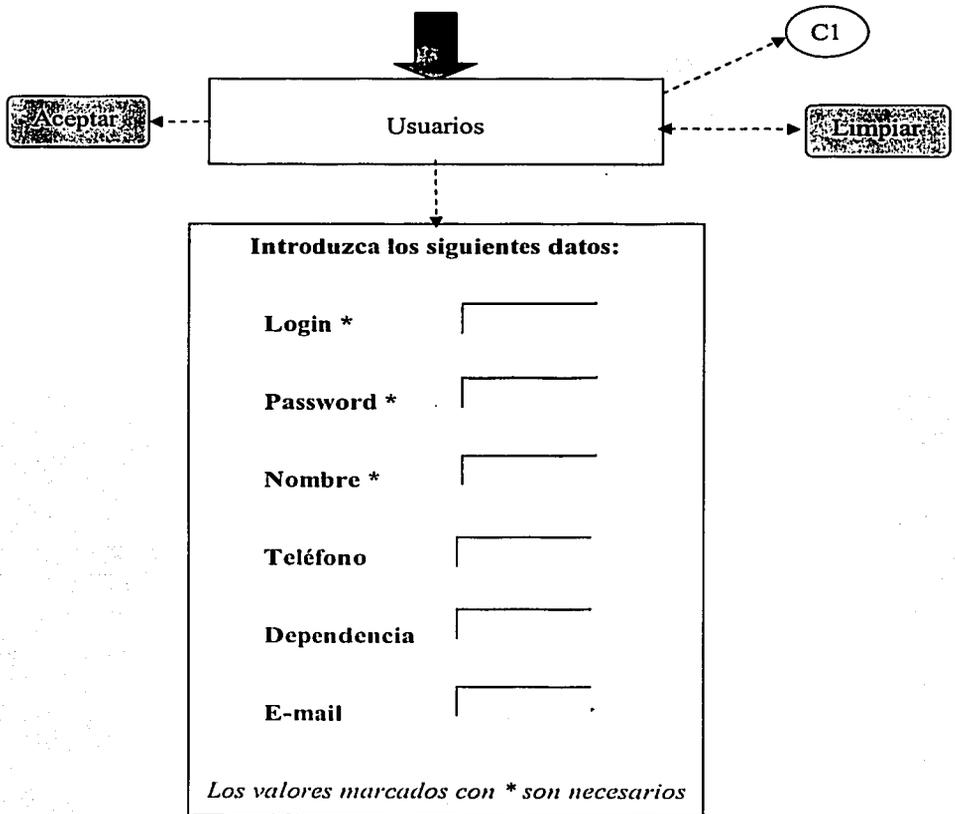


TESIS CON FALLA DE ORIGEN

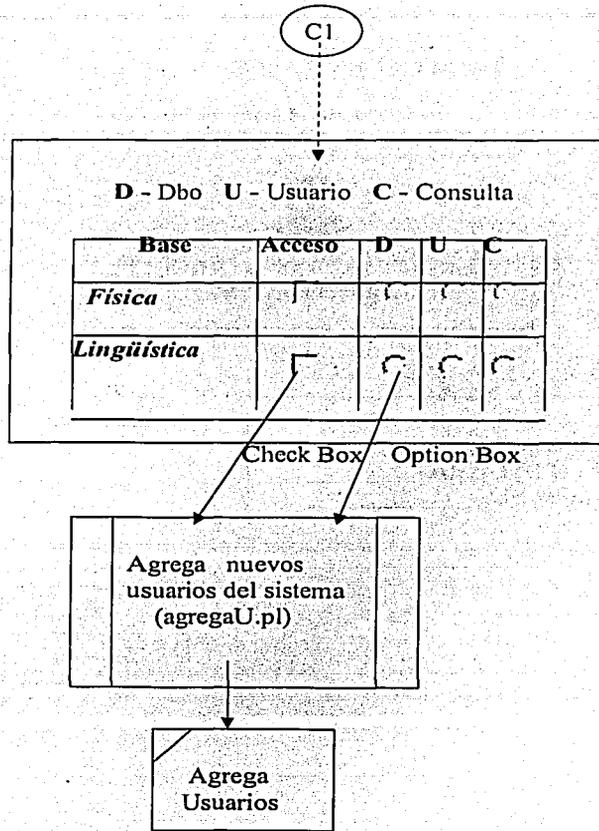
Este escenario corresponde al módulo de Usuarios, perteneciente al módulo de Administración. Aquí se muestra las acciones que se pueden efectuar en este módulo.

Escenario Pagina Usuarios

(nivel 1)



TESIS CON
FALLA DE ORIGEN

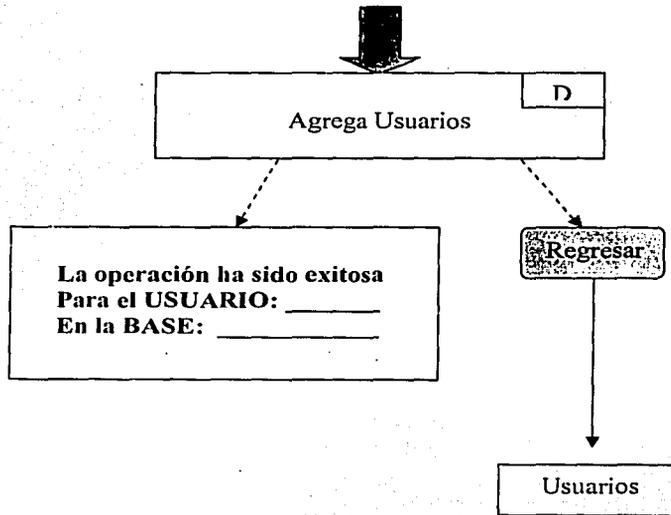


UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA EN SISTEMAS DE INFORMACION

Este escenario corresponde al módulo de insertar usuarios, perteneciente al módulo de Administración. Aquí se muestra las acciones que se pueden efectuar en este módulo.

Escenario Pagina Agrega Usuarios

(nivel 2)



TESIS CON
FALLA DE ORIGEN

2.6 Resumen del capítulo

En la elaboración de todo sistema, es muy importante la etapa del análisis, la cual permite determinar los elementos que intervendrán en el sistema a desarrollar, su estructura, etc. de forma precisa.

Esta etapa nos permitió determinar qué comportamientos se espera obtener, así como formalizar los requerimientos y las necesidades para el sistema que se va a desarrollar a través de pláticas y entrevistas con los responsables del proyecto.

Cabe mencionar que el documento fue modificado en varias ocasiones hasta que se satisfizo completamente las necesidades de los responsables, por lo que se presentó el documento final. De igual manera, se presentaron las consideraciones y expectativas planteadas para el sistema a desarrollar.

Una vez que ya se realizó el análisis del sistema a desarrollar, es necesario determinar cómo se va a construir, por lo que es necesario definir en la etapa de diseño las herramientas que se van a utilizar. Se mencionó una breve descripción de las herramientas a utilizar, así como las principales características relevantes y la justificación de la elección de las mismas.

Posteriormente, se mostraron los diagramas E-R de la base de datos del sistema a desarrollar, así como el diccionario de datos de la base.

En el final del capítulo se mostró el diagrama jerárquico funcional del sistema, y los escenarios más representativos del sistema. Cabe mencionar que las fichas técnicas se presentan en el siguiente capítulo.

CAPÍTULO 3 DESARROLLO, IMPLANTACIÓN Y RESULTADOS

3.1 Estructura de los Scripts en Perl

Como se mencionó en el análisis, se crearon CGI's en Perl que nos van a servir de interfaz entre el Web y la base de datos. Estos Scripts van a permitir a los usuarios interactuar con el sistema a través del Web.

3.1.1 SCRIPTS

A continuación se describen algunos de los Scripts más representativos. La lista completa de los Scripts se puede consultar en el Manual Técnico del sistema.

Nombre: Intro.pl
Descripción: Muestra el nombre de la base a la que se está consultando, así como la fecha y la hora.
Parámetros: login, password, base y roll
Proviene de: FramePrincipal.pl.

Nombre: Terminos.pl
Descripción: Muestra el término seleccionado, los campos y las opciones disponibles para el usuario.
Parámetros: login, password, base, roll, letra consultada, término y terminoActual
Proviene de: Terminos.html

Nombre: VerDef.pl

Descripción: Muestra el sinónimo seleccionado en Defs.pl, los campos u las opciones disponibles para el usuario.
Parámetros: login, password, base, roll, letra consultada, termino, terminoActual, areaActual, Area y el identificador de la definición seleccionada
Proviene de: Defs.html

Nombre: EliminaF.pl
Descripción: Permite eliminar un elemento de la tabla de Fuentes.
Parámetros: login, password, base, roll, letra consultada y fuente
Proviene de: Fuentes.pl
Procedimientos Almacenados:

Proc_Elimina_Fuente

Este procedimiento permite eliminar una fuente en el objeto Fuente de la base de datos.

Nombre: AreaTemática.pl
Descripción: Muestra el área temática seleccionada, los campos y las opciones disponibles para el usuario.
Parámetros: login, password, base, roll, letra consultada, termino y el identificador del área Temática seleccionada
Proviene de: Selec.pl

Nombre: AgregaS.pl
Descripción: Permite agregar un campo a la tabla de Sinónimos.
Parámetros: login, password, base, roll, letra consultada, termino, el identificador del sinónimo, area, areaActual y terminoActual
Proviene de: Sinonimos.pl
Procedimientos Almacenados:

Proc_Inserta_Sinónimo

Este procedimiento permite agregar un nuevo término al objeto Sinonimos en la base de datos.

Nombre: ModificaF.pl
Descripción: Permite modificar un elemento de la tabla de Fuentes.
Parámetros: login, password, base, roll y fuente
Proviene de: Fuentes.pl
Procedimientos Almacenados:

Proc_Actualiza_Fuente

Este procedimiento permite modificar un nuevo término al objeto Fuente en la base de datos.

A continuación, se describe una de las páginas estáticas del sistema.
Todas las descripciones de las páginas estáticas del sistema pueden consultarse en el

manual Técnico.

3.1.2 PÁGINAS ESTÁTICAS

Nombre: Default.html

Entrada: Es la página abierta por defecto

Propiedades:

A continuación, se muestran el esbozo de esta página y la descripción de cada componente:

Esbozo de la página: Default.htm

BANCO TERMINOLÓGICO

Introduzca lo siguiente:

Login:

Password:

Nombre	Componente	Función
Login	Caja de Texto (Text)	Permite introducir la clave del usuario
Password	Caja de Texto (Password)	Permite introducir la contraseña del usuario
Aceptar	Botón (Submit)	Permite enviar los datos
Limpiar	Botón (Reset)	Permite reiniciar el formulario.

A continuación, se describe algunas de las páginas dinámicas representativas del sistema. Todas las descripciones de las páginas dinámicas del sistema pueden consultarse en el manual Técnico.

3.1.3 PÁGINAS DINÁMICAS

Nombre: frmTermino

Generada por: Terminos.pl

Propiedades:

A continuación, se muestran el esbozo de esta página y la descripción de cada componente:

Esbozo de la página: frmTermino

TÉRMINOS

Seleccione un término:

Término:

Agregar
Eliminar

Término	Área Temática	Modificar	Eliminar	Responsable	Fecha	Modificación
pruebaGMC	Lingüística	<input type="checkbox"/>	<input type="checkbox"/>	GMunozC	27 Jan 2003 2:04PM	

Nombre	Componente	Función
Término	Lista de Selección(Select)	Muestra una lista con los términos que se encuentran en la base de datos
Opcion valor Agregar	Botón (Submit)	Permite agregar un nuevo término
Opcion valor Eliminar	Botón (Submit)	Permite eliminar el término seleccionado
Limpia	Botón (Reset)	Permite reiniciar el formulario.
IdTerm	Botón (radio)	Al seleccionarlo permite agregar un nuevo término a la base de datos

TESIS CON
 FALLA DE ORIGEN

IdElim	Caja de selección (checkbox)	Si está seleccionado, permite eliminar un término de la base de datos
--------	------------------------------	---

Nombre: frmAgrega

Generada por: AgregaD.pl

Propiedades:

A continuación, se muestran el esbozo de esta página y la descripción de cada componente:

Esbozo de la página: frmAgrega

AGREGAR DEFINICIÓN

Término: pruebaGMC

Área Temática: Lingüística

Definición:

Contexto Definitorio:

Páginas:

Fuente:

[Regresar](#)

Nombre	Componente	Función
Definición	Area de Texto (TextArea)	Permite introducir una nueva definición.
Contexto	Area de Texto (TextArea)	Permite introducir un nuevo contexto definitorio
Páginas	Caja de Texto (text)	Permite introducir un número de página o número de página.
Fuente	Lista de Selección(Select)	Muestra una lista con las fuentes que se encuentran en la base de datos
Agregar	Botón	Permite agregar una nueva definición.
Limpiar	Botón (Reset)	Permite reiniciar el formulario.
Regresar	Liga	Permite regresar a la página principal de Definiciones

Nombre: frmModifica

Generada por: ModificaT.pl

Propiedades:

A continuación, se muestran el esbozo de esta página y la descripción de cada componente:

Esbozo de la página: frmModifica

MODIFICA TÉRMINO

Modifique el término:

Término:

Cruce:

Área Temática:

Aceptar
Limpiar

[Regresar](#)

Nombre	Componente	Función
Término	Caja De Texto (Text)	Permite modificar el término
**cruce	Caja de Texto (Text)	Permite modificar o introducir un nuevo cruce
Area	Lista de Selección (Select)	Muestra una lista con las áreas temáticas que se encuentran en la base de datos.
Aceptar	Botón	Permite modificar un término
Limpiar	Botón (Reset)	Permite reiniciar el formulario.

3.2 Servidor de Base de Datos.

3.2.1 ESTRUCTURA FÍSICA DE LA BASE DE DATOS

Microsoft SQL Server brinda herramientas que facilitan la creación de la estructura de la base de datos.

A través de la herramienta Microsoft Enterprise Manager es posible reservar espacio para una base de datos.

Otra herramienta que nos permite crear físicamente la base de datos es Microsoft Query Analyzer, en la cual es, entre otras cosas, ejecutar múltiples Queries ó Scripts en T-SQL

Creación de la base de datos física en SQL Server Linguistica en la herramienta Microsoft Query Analyzer, y considerando las necesidades para la base de datos, se planteó que:

El archivo de datos Linguistica_Data.MDF cuenta con un espacio inicial de 10 MB, con un crecimiento del 10% y con un límite de espacio de 20 MB.

El log de Transacciones Lingüística_Log.LDF cuenta con un espacio de 4MB, con un crecimiento de 1MB y con un límite de espacio de 5MB

El archivo de datos MTF_Usuarios.MDF cuenta con un espacio inicial de 5 MB, con un crecimiento del 10% y con un límite de espacio de 10 MB.

El log de Transacciones MTF_Usuarios_Log.LDF cuenta con un espacio de 1MB, con un crecimiento de 1MB y con un límite de espacio de 3MB

TESIS CON
FALLA DE ORIGEN

3.2.2 SCRIPTS DE CREACIÓN DE LA BASE DE DATOS

Una vez que se ha reservado un espacio de memoria para la base de datos, se crean las tablas para la base de datos Lingüística.

A continuación se muestran algunos de los scripts de creación de los objetos de la base de datos. La lista completa se encuentra en el manual técnico del sistema:

Objeto: Tabla Nombre: Definiciones

```
CREATE TABLE [dbo].[Definiciones] (  
    [Id] [int] NOT NULL ,  
    [IDtermino] [int] NOT NULL ,  
    [definicion] [text] NOT NULL ,  
    [ContextoDeUso] [text] NULL ,  
    [IdFuentes] [char] (30) NOT NULL ,  
    [Paginas] [char] (18) NULL ,  
    [Responsable] [char] (20) NULL  
)
```

Objeto: Tabla Nombre: Terminos

```
CREATE TABLE [dbo].[Terminos] (  
    [IDtermino] [int] NOT NULL ,  
    [termino] [varchar] (100) NOT NULL ,  
    [idArea] [int] NULL ,  
    [Cruce] [varchar] (50) NULL ,  
    [Responsable] [char] (20) NULL  
)
```

Objeto: Tabla Nombre: Fuentes

```
CREATE TABLE [dbo].[Fuentes] (  
    [IdFuente] [char] (30) NOT NULL ,  
    [NombreDeFuente] [varchar] (100) NOT NULL ,  
    [NombreEditor] [varchar] (100) NULL ,  
    [LugarDePublicacion] [varchar] (100) NULL ,  
    [Anio] [smallint] NULL ,  
    [LigaElectronica] [varchar] (100) NULL ,  
    [Responsable] [char] (20) NULL  
)
```

Para la creación de los Constraints (un constraint es un tipo de restricción dentro de una base de datos, éste puede ser de índole primario o foráneo, que permite establecer la integridad referencial en las tablas) de la base de datos de lingüística, se hicieron cambios a las tablas. A continuación se muestran algunas de estas definiciones. Las definiciones completas pueden ser consultadas en el manual técnico del Sistema:

Definición de Constraint en la tabla: Sinonimos

```
ALTER TABLE [dbo].[Sinonimos] WITH NOCHECK ADD
  CONSTRAINT [PK_SINONIMOS] PRIMARY KEY CLUSTERED
  (
    [IDtermino],
    [Sinonimo]
  ) ON [PRIMARY]
```

Definición de Constraint entre las tablas: Definiciones y Fuentes

```
ALTER TABLE [dbo].[Definiciones] WITH NOCHECK ADD
  CONSTRAINT [FK_FUENTES_DEFS] FOREIGN KEY
  (
    [IdFuente]
  ) REFERENCES [dbo].[Fuentes] (
    [IdFuente]
  ),
  CONSTRAINT [FK_TERMINOS_DEFS] FOREIGN KEY
  (
    [IDtermino]
  ) REFERENCES [dbo].[Terminos] (
    [IDtermino]
  )
```

Definición de Constraint entre las tabla: Terminos y AreaTematica

```
ALTER TABLE [dbo].[Terminos] ADD
  CONSTRAINT [FK_AREA_TERMINOS] FOREIGN KEY
  (
    [idArea]
  ) REFERENCES [dbo].[AreaTematica] (
    [IdArea]
  )
```

3.2.3 STORES PROCEDURES

A continuación se muestran algunos de los Stores Procedures más representativos. Los demás pueden ser consultados en el manual técnico del sistema.

- Procedimiento para agregar un elemento en la tabla de Sinónimos:
proc_Inserta_Sinonimo

```
CREATE proc proc_Inserta_Sinonimo @sinonimo char(100), @termino varchar(100) ,
    @responsable char(20)
as
/** Nombre Procedimiento : proc_Inserta_Sinonimo
**
** Base de Datos      : Linguistica
**
** Uso                : Inserta en la tabla de Sinonimos
**
** Autor              : Gabriela Muñoz
**
** Parametros Entrada :
** @Sinonimo
** @termino
** @responsable
**
** Parametros Salida  : @err
**
** Valores de retorno : -500 -> se aborta el procedimiento almacenado
**                    0   -> el procedimiento almacenado se efectuo con exito
**
** Variables : @err int, -- guarda el status de error de las sentencias sql
**
** Tablas usadas: Sinonimos, Terminos
**
**/

begin tran
declare @err int, -- variable de error
        @idtermino int
-- verifica que no exista ya el sinonimo que se desea insertar
if exists(select sinonimo from sinonimos, Terminos where sinonimo=@sinonimo and
Terminos.idTermino = Sinonimos.idtermino and termino = @termino)
begin
    select @err = -500
end
end
```

TESIS CON
FALLA DE ORIGEN

```

else
begin
-- verifica que el termino exista
if exists(select * from Terminos where termino=@termino)
begin
select
@idtermino = (select idTermino from Terminos where termino=@termino)
insert Sinonimos values (@idtermino, @sinonimo, @Responsable)
select @err = @@error
end
else
select @err= -550
end
if @err != 0
begin
rollback transaction
return @err
end
commit tran
return 0

```

- Procedimiento para eliminar un elemento en la tabla de Área Temática:
proc_Elimina_AreaTematica

```

create proc proc_Elimina_AreaTematica @AreaTematica varchar(50)
as
/** Nombre Procedimiento : proc_Elimina_AreaTematica
**
** Base de Datos      : Linguistica
**
** Uso                : Elimina el area tematica correspondiente
**
** Autor              : Gabriela Muñoz
**
** Parametros Entrada : @AreaTematica -> área a borrar
**
** Parametros Salida  : @err
**
** Valores de retorno : -300 -> se aborta el procedimiento almacenado
**                    0   -> el procedimiento almacenado se efectuo con exito
**
** Variables          : @err int, -- guarda el status de error de las sentencias sql
**
** Tablas usadas      : AreaTematica, Terminos
**/

```

```
begin tran
declare @err int, --variable de error
        @IdArea int
if exists (select * from AreaTematica where @AreaTematica = AreaTematica)
begin
    select @idArea = (select idArea from Areatematica where
AreaTematica=@AreaTematica)
    if not exists (select * from Terminos where idArea = @IdArea)
    begin
        delete AreaTematica where IdArea = @IdArea
        select @err = @@error
    end
    else
    begin
        select @err = -500
    end
end
if @err != 0
begin
    rollback transaction
    return @err
end
commit tran
return 0
```

- Procedimiento para modificar un elemento en la tabla de Fuentes:
proc_Actualiza_Fuente

```
CREATE proc proc_Actualiza_Fuente @idFuente char(30), @NombreFuente
varchar(100), @NombreEditor varchar(100),
        @LugarDePublicacion varchar(100), @Anio smallint, @LigaElectronica
varchar(100), @Responsable char(20)
as
/** Nombre Procedimiento : proc_Actualiza_Fuente
**
** Base de Datos : Lingüística
**
** Uso : Modifica en la tabla de Fuentes
**
** Autor : Gabriela Muñoz Cortés
**
** Parametros Entrada :
** @idFuente
```

```
** @NombreFuente
** @NombreEditor
** @LugarDePublicacion
** @Ano
** @LigaElectronica
** @Responsable
**
** Parametros Salida : @err
**
** Valores de retorno : -500 -> se aborta el procedimiento almacenado
** 0 -> el procedimiento almacenado se efectuo con exito
**
** Variables : @err int -- guarda el status de error de las sentencias sql
**
** Tablas usadas : Fuentes
**
**/

begin tran
declare @err int -- variable de error
if not exists (select * from Fuentes where @idFuente = idFuente)
begin
select @err = -500
end
else
begin
update Fuentes set nombrefuente = @NombreFuente, nombreditor =
@NombreEditor, lugardepublicacion = @LugarDePublicacion, anio=@Anio,
ligaelectronica=@LigaElectronica, responsable = @Responsable where idFuente =
@idFuente
select @err = @@error
end
if @err != 0
begin
rollback transaction
return @err
end
commit tran
return 0
```

3.3 Servidor Web

El servidor Web que se utilizó, como se mencionó en el capítulo dos, es Microsoft Internet Information Server.

Es necesario crear un directorio virtual dentro del sitio Web por default, cuyas principales características son:

Número de conexiones: ilimitada

Cuando se conecta al servidor, se fuerza, por seguridad del sistema, a que el origen del contenido esté en un directorio del servidor especificado, con permisos de sólo lectura.

El sitio tiene permisos de ejecución de secuencias de comandos. Protección: media

Documento por default: Default.htm

En cuanto a la seguridad del directorio, se utilizó la autenticación integrada de Windows por ser la más segura, ya que el login y el password que se envía a través de las páginas se encriptan.

3.4 Sistema de Administración

Como se mencionaba en el capítulo dos, se planteó desarrollar un sistema de administración de usuarios.

Este sistema de administración de usuarios tiene asignado un número de puerto diferente al default del protocolo http (80), que es el que tiene asignado la interfaz Web del sistema para la alimentación de la base de datos. Esto se hizo para evitar que personas ajenas a los administradores tengan conocimiento de éste.

Una vez que el administrador ha introducido su login y contraseña al sistema, ingresa a una página dinámica que muestra los campos necesarios para la inserción de un nuevo usuario:

- **Login.**
Este es un campo requerido que solicita se ingrese la nueva clave del usuario que se desea agregar, ó en su defecto, la clave del usuario ya existente en la base de datos al cual se desea modificar o eliminar algún permiso.

- **Password**
Este es un campo requerido que solicita se ingrese la contraseña del usuario nuevo o del usuario ya existente en la base de datos.
- **Nombre**
Este también es un campo requerido en el cual se pide el nombre del nuevo usuario a agregar a la base de datos ó el del usuario ya existente del cual se desea hacer alguna modificación en los permisos ya asignados.
- **Teléfono**
Este no es un campo obligatorio. Aquí se solicita el teléfono del usuario a agregar en la base de datos. Si el usuario ya existe pero no tiene asignado un número telefónico, si se introduce se agrega el número telefónico. En caso de ser un usuario ya existente y ya tenga un número telefónico y éste es modificado, se guarda el número telefónico modificado al aceptar los cambios.
- **Dependencia**
Este campo es opcional. Aquí se introduce la dependencia a la cual pertenece el nuevo usuario. Como en el campo anterior, si el usuario existe en la base de datos pero no tiene asignado este campo, lo agrega. En caso de que ya exista este campo y se modifique, se guardará los cambios realizados.
- **E-mail**
Este campo tampoco es obligatorio. Aquí se introduce la dirección electrónica del usuario a agregar a la base de datos o de uno ya existente. También se guarda la modificación a este campo en un usuario ya existente.

Posteriormente de estos campos, se muestra el listado de las bases existentes de la siguiente forma:

D - Dbo U - Usuario C - Consulta				
Base	Acceso	D	U	C
<i>Física</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Lingüística</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Check Box Option Box

TESIS CON
FALLA DE ORIGEN

Es posible dar de alta a un usuario sin asignarle permisos en ninguna de las bases terminológicas existentes, con lo cual sólo podrá tener accesos de consulta. Esto significaría que en las bases existentes se seleccionó la opción de consulta únicamente.

En caso de que se desee asignarle otros permisos, es necesario tener seleccionado el check box correspondiente a cada base terminológica y seleccionar qué tipo de usuario va a ser.

Si el usuario ya existe y se desea hacer alguna modificación, se procede igual que con un usuario nuevo, es posible agregar permisos en bases en las cuales no tenga más que

permisos de consulta, modificar los permisos que tenía asignados, e incluso eliminar los permisos con los que ya contaba anteriormente.

3.5 Pruebas del sistema

Las pruebas del sistema fueron realizadas tanto por los desarrolladores del sistema, como por los usuarios de éste.

Al principio, se llevaron a cabo pruebas por parte del desarrollador del sistema, y a continuación se describe cómo se llevaron a cabo.

3.5.1 PRUEBAS UNITARIAS DE INTEGRACIÓN

Para poder llevar a cabo las pruebas unitarias, es necesario probar cada uno de los módulos del sistema. Para poder probar cada módulo individualmente, es necesario crear módulos auxiliares que simulen las acciones de los módulos indicados por el módulo que se está probando. Asimismo, para este tipo de prueba, es necesario crear módulos conductores para establecer las precondiciones necesarias al módulo objeto de la prueba para poder examinar los resultados de la prueba. Debido a esto, a menudo se combinan ambos tipos de prueba unitaria y de Integración.

Para este sistema, se utilizó el tipo de prueba unitaria de integración de tipo de desarrollo incremental que consiste en agregar cada módulo componente individual al conjunto de componentes existentes y el conjunto resultante de prueba. Esto redujo la necesidad de crear módulos conductores y permitió, además, examinar en detalle las interfases.

Cuando las pruebas unitarias y de integración se realizan separadamente, es difícil examinar los componentes individuales o módulos que causan resultados incorrectos, por

el contrario, con el tipo de prueba incremental, los errores surgen al incorporar un nuevo componente a un grupo previamente probado.

Cabe mencionar que se presentaron errores pequeños sobre todo cuando se integraba un módulo a los ya integrados, pero fue bastante más sencillo identificar los errores ya que este tipo de pruebas permite visualizar que el error está en el módulo incorporado o bien a alguna de las interfases entre éste y alguno de los otros módulos con los que interactúa.

Una vez que se integraron todos los módulos del sistema y se realizaron las pruebas unitarias de integración, se realizaron pruebas globales.

3.5.2 PRUEBAS GLOBALES

Se realizaron cuatro tipos de pruebas globales básicos:

- Prueba de volumen

Aquí se prueba que el sistema pueda manejar el volumen de actividades que ocurren cuando el sistema está en el punto más alto de su demanda de procesamiento. Esto se determina en función del número de usuarios que soporta simultáneamente el servidor de la base de datos. Este tipo de prueba no se pudo llevar a cabo de forma que se cumpliera esta condición, ya que este servidor soporta miles de usuarios simultáneamente, y los requerimientos del sistema son muy inferiores. Se probó, no obstante, que al estar utilizando el sistema simultáneamente los involucrados en el proyecto, no se presentaron fallas ni se colapsó el sistema.

- Prueba de almacenamiento

Esta prueba se realizó satisfactoriamente ya que el servidor de la base de datos tiene 1 MB de almacenamiento inicial, y se incrementa conforme se requiera; además, el servidor está dedicado únicamente para los procesos de este sistema.

- Prueba de tiempo de ejecución

Para esta prueba, fue necesario determinar si el sistema era ágil cuando respondía a una gran cantidad de información que procesar. Esta prueba, que se efectuó de forma simultánea con la primera que se menciona, permitió observar que los tiempos fueron mínimos y los tiempos de procesamiento son casi imperceptibles en la interfaz Web, aunque estos tiempos dependen en gran medida de las características de la PC que el usuario esté ocupando.

- Pruebas de factores humanos

Aquí se determinó cómo utilizarán los usuarios el sistema al procesar datos, por lo que se comprobó, gracias a las entrevistas con los usuarios del sistema, que el

sistema es amigable y de fácil operación, además, genera mensajes indicativos de ocurrencias del sistema.

Además, los usuarios, a pesar de que algunos nunca habían hecho alguna manipulación a información a través del Web, pudieron hacer uso del sistema de forma correcta.

Es necesario mencionar que para las pruebas se utilizaron dos tipos de datos de prueba, que se describen a continuación:

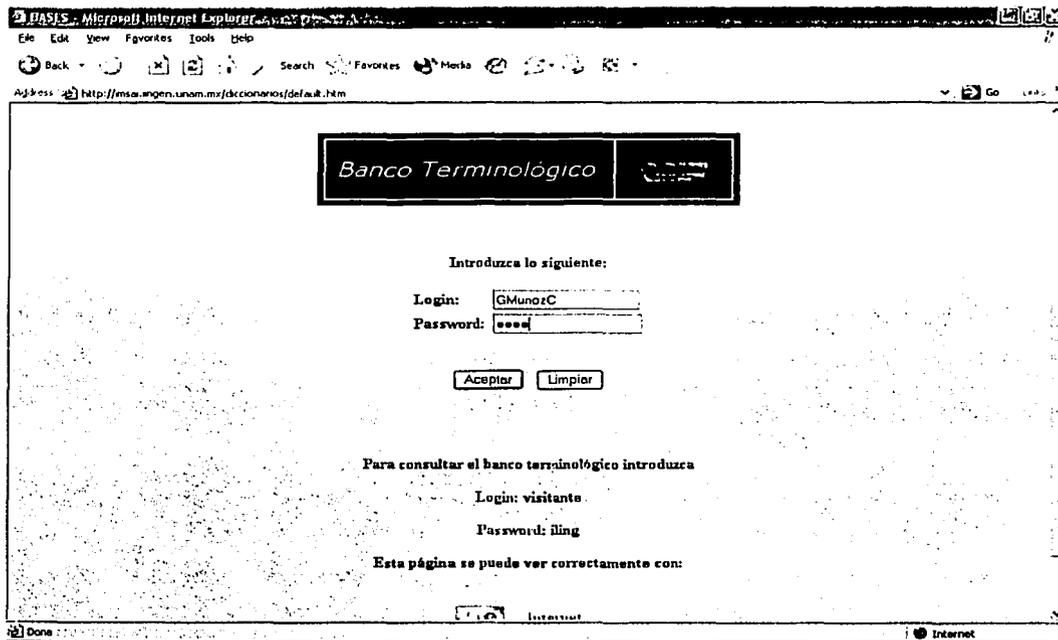
- Datos reales, los cuales permiten probar las ocurrencias y los casos reales que se pueden presentar, aunque no permiten probar otras rutas programadas porque no han sido alcanzados por los usuarios para las pruebas en un sistema en desarrollo, o también porque no han ocurrido en el periodo de prueba del sistema ya en funcionamiento.
- Datos artificiales. Son los creados artificialmente tratando de considerar todas las combinaciones y rutas posibles, y lo recomendable es que sean preparados por personas que no sean los desarrolladores del sistema.

Como se menciona en las pruebas de tipo global, se realizaron pruebas de aceptación por parte de los usuarios del sistema; estas pruebas son del mismo tipo que las tres primeras pruebas globales mencionadas anteriormente, pero con la diferencia de que son determinadas por el usuario y no por el desarrollador del sistema.

3.6 Ejemplo de Navegación del sistema

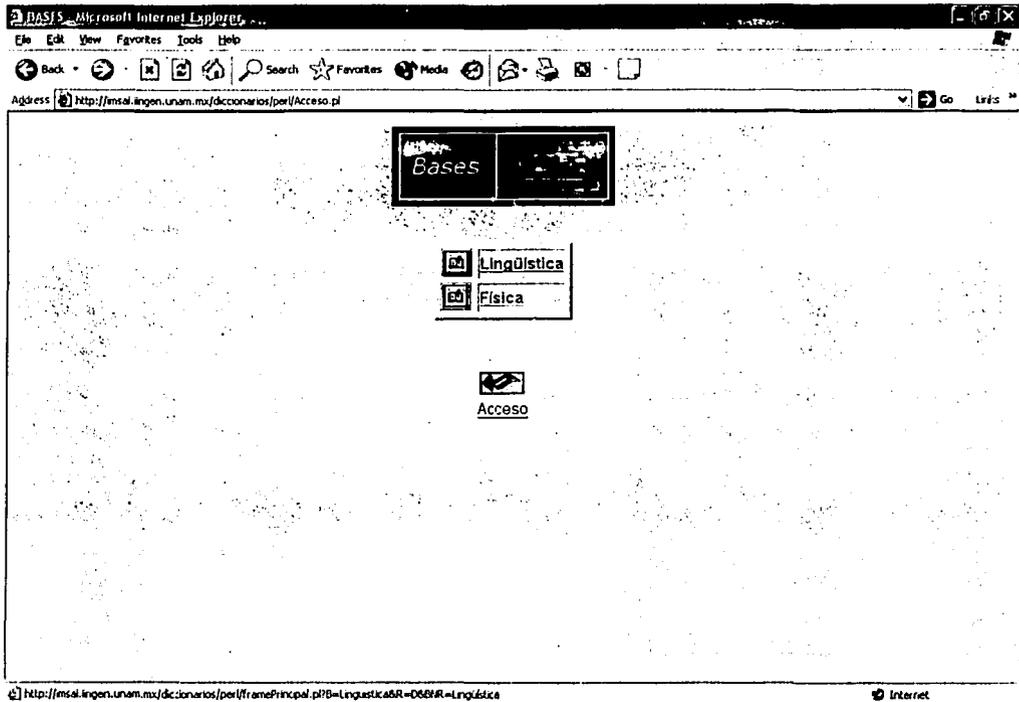
A continuación se presentan algunas de las páginas del sistema implantado con un ejemplo de navegación sencillo.

La página presentada es la página default del sistema, y permite el acceso al mismo por medio de un login y password. Si la persona que ve la página desea ingresar al sistema y no cuenta con permisos para ello, se proporciona un login y password de visitante que sólo le da permisos de consulta.



TESIS CON
FALLA DE ORIGEN

La siguiente página que presenta el sistema le muestra una liga a la base o las bases a las cuales el usuario tiene asignado algún permiso. Si se entra con el login y password proporcionados en la página anterior, el sistema mostrará todas las bases que se encuentren disponibles y tendrá permisos de consulta únicamente.



TESIS CON
FALLA DE ORIGEN

A continuación, se muestra parte del código del Script que valida el login y password del usuario y muestra la página con las ligas a las bases a las que tiene asignado permisos.

Acceso.pl

```

use MSSQL::DBlib;          # Se usara el modulo DBlib
use CGI;                   # Se usara el modulo CGI

$query=new CGI;

$cookie1 = $query->cookie(-name => "id", -value => [$query->param('Login')]);
$cookie2 = $query->cookie(-name => "pas", -value => [$query->param('Password')]);

print $query->header(-cookie=>[$cookie1, $cookie2]);

print <<ET;                # Imprime lo siguiente tal y como se muestra
<HTML>
<HEAD>
<TITLE>BASES</TITLE>
</HEAD>
<BODY TEXT="#222222" BGCOLOR="#FFFFFF" LINK="#FF6666" VLINK="#CC0000"
ALINK="#FF0000" BACKGROUND="./img/fondo.jpg" NOSAVE>

<CENTER><TABLE BORDER=6 CELLPADDING=10 bordercolor="#FF0000"
bordercolorlight="#800000" bordercolordark>
<TR>
<TD VALIGN=CENTER BGCOLOR="#A40000" align="center"><FONT FACE="Verdana, Arial">
FONTCOLOR="#C0C0C0"><FONT SIZE=+2><I>Bases</I></FONT></FONT></FONT></TD>
<TD VALIGN=CENTER BGCOLOR="#A40000" align="center"><IMG SRC="./img/logoII.GIF"
ALT="Logo" NOSAVE width="89" height="46"></TD>
</TR>
</TABLE></CENTER>
<BR>&nbsp;
<BR>
ET

$i = $query->param('Login');
$p = $query->param('Password');
$s = "RUBI";

#Conexiones a SQL

if ($X = MSSQL::DBlib->dblogin($i, $p, $s))
{
    ($X->dbuse('Usuarios') == SUCCEED )          # Usa tabla de Usuarios
    or "<center><b>No se pudo usar la base de datos de Usuarios</center></b>";

    ($X -> dbcmd("select Base,Roll from UsuariosBase where cLogin = \"\$i\"")) # En el buffer de
    comandos del cliente almacena la sentencia SQL

```

TESIS CON
FALLA DE ENTREN

```

or print "No puede meter el comando en el buffer";
($X -> dbsexec) # Envía el buffer de comandos al Servidor
or print "Fallo el select";

($X -> dbresults) # Trae los resultados
or print "Fallo el results";

$status = $X->dbretstatus;

if($status == 0) {
    $i = 0;
    while(@data = $X->dbnxtrow) {
        $data[0] =~ s/[ ]//g;
        $Bases[$i] = $data[0];
        $Roles[$i] = $data[1];
        $i++;
    }

    if((@Bases+0)>1) {
        print "<center><table cellpadding=5 border=3>";
        for($i=0;$i<(@Bases+0);$i++) {
            print "<tr><td><a href=\"framePrincipal.pl?Base=$Bases[$i]&Roll=$Roles[$i]\">
            <img SRC=\"../img/book.gif\"></a></td><td><font size=5>$Bases[$i]</font>
            </td></tr>\n";
        }

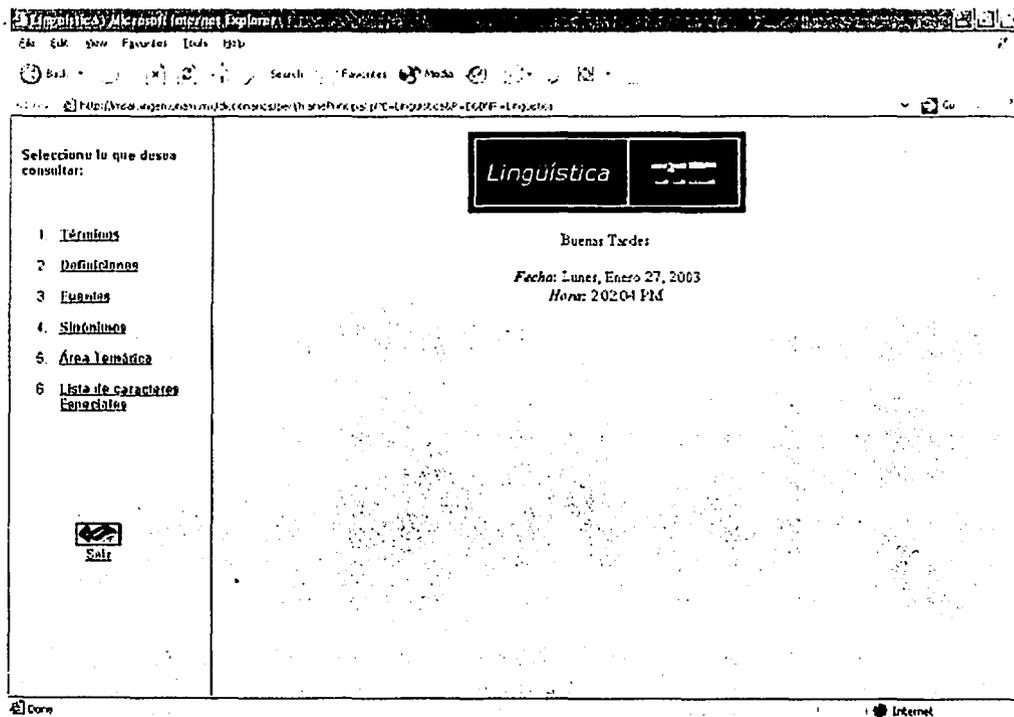
        print "</center></table>";
    }
    else {
        print "<form name=\"frmOpcion\" method='post' action=\"framePrincipal.pl\">";
        print "<INPUT type=\"hidden\" value=$Bases[0] name=\"Base\">";
        print "<INPUT type=\"hidden\" value=$Roles[0] name=\"Roll\">";
        print "<SCRIPT LANGUAGE=\"JavaScript1.2\">";
        print "document.frmOpcion.submit();";
        print "</SCRIPT>";
    }
    else {
        print "<center><b>Un error ha ocurrido contacte a su administrador</center></b>";
    }
    else {
        print "<center><b>El login y password proporcionados son incorrectos</center></b>";
    }

    print <<ET;
    </form>
    <BR>&nbsp;
    <center><a href=\"../default.htm\" target = \"_top\">
    <img SRC=\"../img/logomaspetit2.gif\"><BR>Acceso</a></center>
    ET

    print $query->end_html;

```

Posteriormente, el sistema muestra una página que nos indica la base con la que estamos trabajando y las ligas a la información que se puede consultar, o si tiene otros permisos, a las opciones que tenga habilitadas el usuario.



TESIS CON
FALLA DE ORIGEN

Si entramos a la liga de términos, se mostrará la siguiente página. Cabe señalar que el usuario es desarrollador del sistema, por lo que puede agregar, eliminar o modificar el término seleccionado.

Si el usuario sólo tuviera permisos de consulta, los botones de agregar y eliminar, así como la opción de eliminar no se mostrarían. Además, los campos de Responsable, Fecha y Modificación tampoco serían visibles a este usuario.

Selecciona lo que deseas consultar:

1. Términos
2. Definiciones
3. Fuentes
4. Sinónimos
5. Área Temática
6. Lista de caracteres especiales

ABCDEF GHIJKL MNOPQRST UVWXYZ Todos

Terminos

Selecciona un término:

Término: pruebaGMC

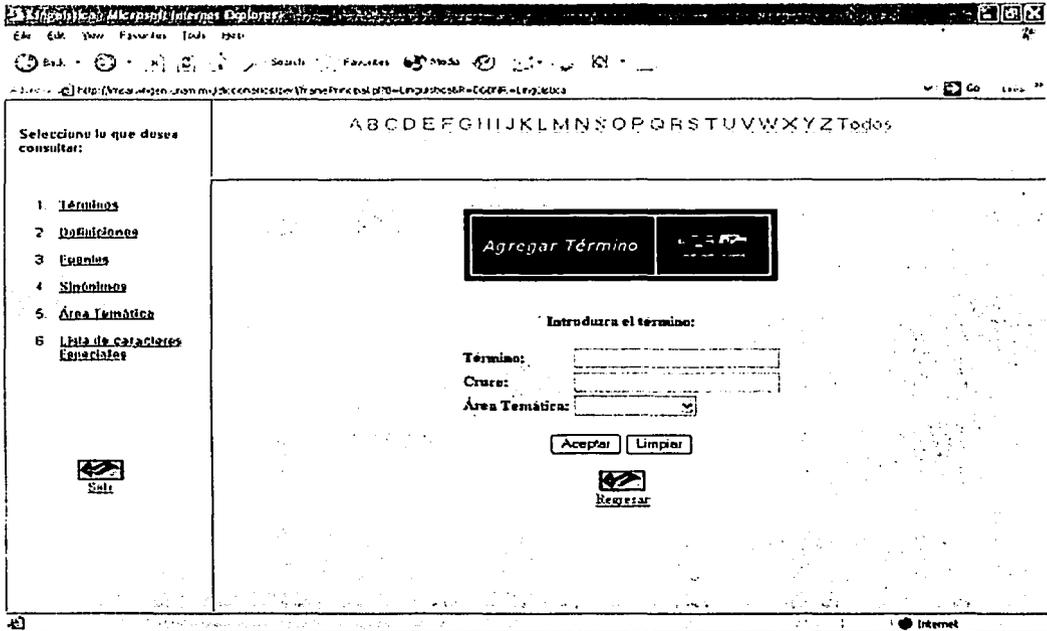
Agregar Eliminar

Término	Área Temática	Cruce	Modificar	Eliminar	Responsable	Fecha	Modificación
pruebaGMC	Lingüística	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	GMunozC	27 Jan 2003 2:04PM	

TESIS CON FALLA DE ORIGEN

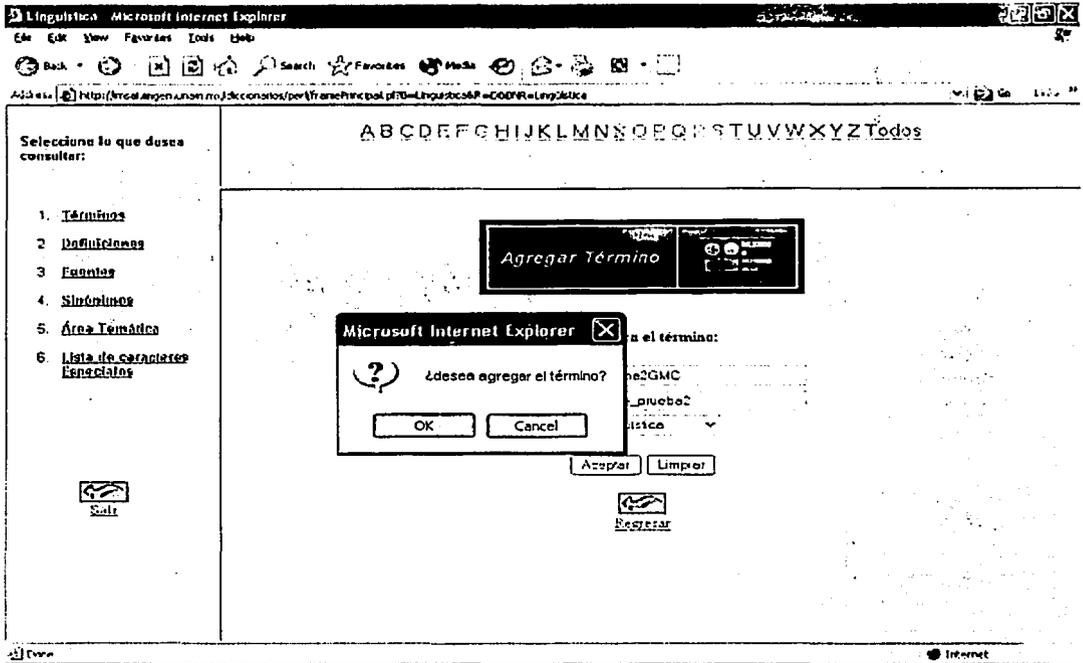
Si escogemos la opción de agregar, nos mostrará la siguiente página que le permitirá al usuario introducir un nuevo término al banco terminológico.

Los campos de término, cruce y área Temática no contienen ningún valor y son obligatorios el término y el área Temática. El término no se agrega hasta que se verifica que el término no está repetido (pueden existir dos términos que se escriban exactamente igual pero con la diferencia que alguno de ellos empiece con una letra mayúscula y el otro con una letra minúscula).



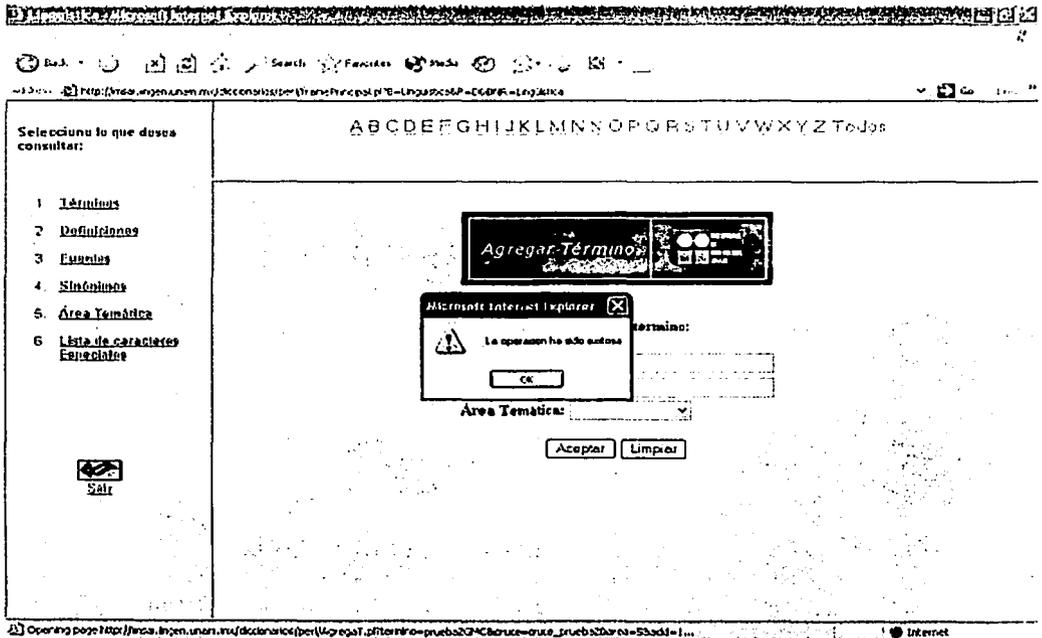
TESIS CON
FALLA DE ORIGEN

Ya que se han llenado los campos, y se presiona el botón de Aceptar, el sistema confirma que se desee agregar ese nuevo término. Esto con el fin de poder modificarlo si es que existe algún error antes de agregarlo.



TESIS CON
FALLA DE ORIGEN

Al confirmar que se desea agregar el término, muestra el resultado de la operación realizada y continúa en la misma página para agregar más términos si así lo requiere el usuario del sistema.



TESIS CON
FALLA DE ORIGEN

Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.argen.unam.mx/3/cenarios/ber/franc/repal/p76-linguistica/DOFR-Ling/2122

ABCDEEFGHIJKLMNOPQRSTUVWXYZ Todos

Seleccione lo que desea consultar:

1. Términos
2. Definiciones
3. Fuentes
4. Sinónimos
5. Área Temática
6. Lista de caracteres Especiales

[Salt](#)

Antepara Término

Introduzca el término:

Término:

Crece:

Área Temática:

[Revertir](#)

Internet

TESIS CON
FALLA DE ORIGEN

Al presionar la liga de Regresar, vuelve a la página de términos, mostrando el término que se acaba de agregar:

Selecciona lo que deseas consultar:

1. Términos
2. Definiciones
3. Eventos
4. Noticias
5. Área Temática
6. Lista de caracteres especiales

ABCDEF GHIJKLMNOPQRSTUVWXYZ Todos

Términos

Selecciona un término:

Término: [prueba2GMC]

[Agregar] [Eliminar]

Término	Área Temática	Cruce	Modificar	Eliminar	Responsable	Fecha	Medición
prueba2GMC	Lingüística	cruce_prueba2	<input type="radio"/>	<input type="checkbox"/>	GMano2C	27 Jan 2003 2:05PM	

Salir

Done Internet

TESIS CON FALLA DE ORIGEN

A continuación, se muestra parte del código del Script que agrega un nuevo término a la base de datos terminológica.

AgregaT.pl

```
use MSSQL::DBlib;      # Se usara el modulo DBlib
use CGI;               # Se usara el modulo CGI

$query=new CGI;

$i = $query->cookie('id');
$p = $query->cookie('pas');
$b = $query->cookie('base');
$r = $query->cookie('roll');
$s="RUBI";

$terminoActual = $query->param('termino');

print $query->header(-cookie=>[$cookie1, $cookie2, $cookie3, $cookie4]);

print "<HTML>\n<HEAD>\n<SCRIPT LANGUAGE=\"JavaScript\">

    function verificaTerm() {
        var termino = document.frmAgrega.termino.value;
        var area = document.frmAgrega.area.value;
        if(termino == "") {
            alert("\Debe introducir un termino\");
            document.frmAgrega.termino.focus()
            document.frmAgrega.termino.select() }
        else {
            document.frmAgrega.submit() } }
</SCRIPT>\n</HEAD>\n";

print <<ET;           # Imprime lo siguiente tal y como se muestra
<BODY TEXT="#000000" BGCOLOR="#FFFFFF" LINK="#FF6666" VLINK="#CC0000" ALINK=
"#FF0000" BACKGROUND="..img/fondo.jpg" NOSAVE>&nbsp;
<CENTER><TABLE BORDER=6 CELLPADDING=10 bordercolor="#FF0000" bordercolorlight=
"#800000" bordercolordark="#CC0000" >
<TR>
<TD VALIGN=CENTER BGCOLOR="#A40000" aling="center"><font size="+2" color="#C0C0C0"
face="Verdana, Arial"><i>Agrega término</i></font></TD>
<TD VALIGN=CENTER BGCOLOR="#000000" aling="center"><img border="0" src=
"..img/logoII.gif" width="89" height="46"></TD>
</TR></TABLE></CENTER>&nbsp; <P><center>
ET

($X = MSSQL::DBlib->dblogin($i, $p, $s)) # Se conecta al servidor $X es la instancia del objeto
or print "<center><b>El login y password proporcionados son incorrectos</B></center>";

($X->dbuse($b) == SUCCEED )
```

```

or "<center><b>No se pudo usar la base de datos</center></b>";
print "<form name=frmAgrega>";
print <<ET;
  <B>Introduzca el término:</B><P>
  <TABLE>
  <tr><td><p><B>Término: </B></td>
  <td><input type="text" name="termino" size="30" MAXLENGTH="100"></td>
ET
($X -> dbcmd("select * from AreaTematica order by AreaTematica asc"))
      # En el buffer de comandos del cliente almacena la sentencia SQL
or print "No puede meter el comando en el buffer";

($X -> dbsqlxec)      # Envía el buffer de comandos al Servidor
or print "Fallo el select";

($X -> dbresults)    # Trae los resultados
or print "Fallo el results";

while(@data = $X->dbnextrow) {
  print "\t<option value=".$data[0].">".$data[1]."</option>\n"; }
print <<ET;
  </select></B></td>
  </TABLE>
  <P><input type="button" value="Aceptar" name="agregar" onClick="verificaTerm()">
  <input type="reset" value="Limpiar">
  <INPUT type="hidden" value=1 name="add"></form>
ET
if($query->param('add')) {
  $termino = $query->param('termino');
  $area = $query->param('area');
  $cruce = $query->param('cruce');
  $letra = $query->param('Op');
  $cruce temp = $cruce;

  if ($cruce eq "") { cruce = "NULL"; }
  else { $cruce = "".$cruce.""; }
  $parametros = "".$termino."".$area.".$cruce.".$l.""; }

$comando = "exec proc_Inserta_Termino $parametros";

($X -> dbcmd("$comando") # En el buffer de comandos del cliente almacena la sentencia SQL
or "no puede meter el comando en el buffer\n";

($X -> dbsqlxec)      # Envía el buffer de comandos al Servidor
or "Fallo el select\n";

($X -> dbresults)    # Trae los resultados
or "Fallo el results\n";
$status = $X->dbretstatus;

```

```
if($status != 0) {
  if($status == -500) {
    print "<SCRIPT LANGUAGE=\"JavaScript\">
    alert(\"No se puede insertar terminos repetidos con la misma área temática\");
    document.frmAgrega.termino.value = \"\$termino\";
    document.frmAgrega.cruce.value = \"\$crucetemp\";
    document.frmAgrega.area.value = \"\$area\"; </SCRIPT>"; }
    else {
    print "<SCRIPT LANGUAGE=\"JavaScript\">
    alert(\"Un error ha ocurrido\")
    document.frmAgrega.termino.value = \"\$termino\";
    </SCRIPT>"; } }
  else {
    $letra = substr($termino,0,1);
    print "<P><input type=\"hidden\" value=\"\$letra\" name=\"Op\">\n";
    print "<SCRIPT LANGUAGE=\"JavaScript\">
    alert(\"La operacion ha sido exitosa\")
    </SCRIPT>";
    $terminoActual = $termino; }
}
else {
  $letra = $query->param('Op');
}

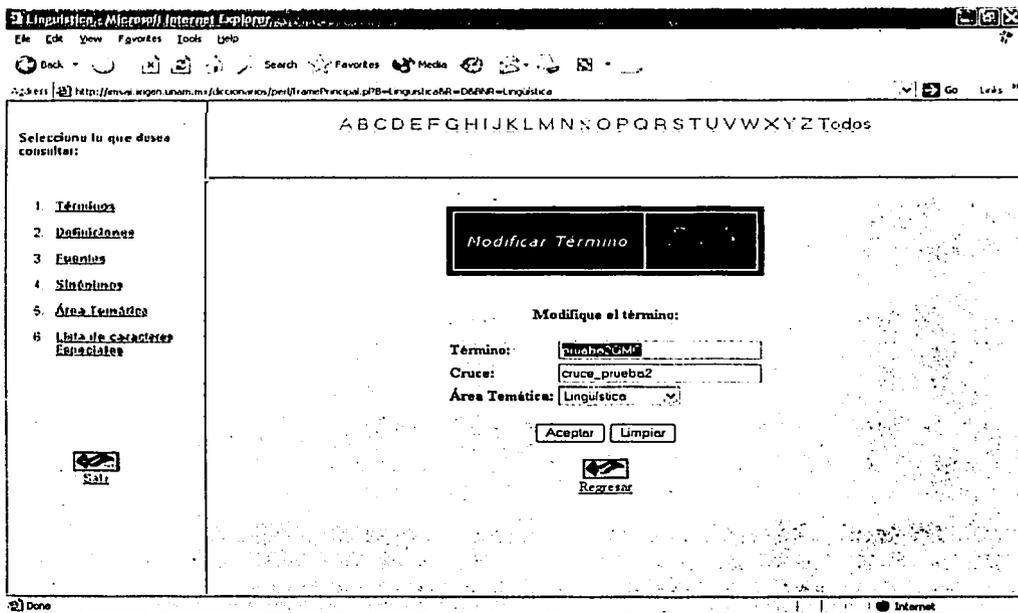
print "<SCRIPT LANGUAGE=\"JavaScript\">
document.frmAgrega.termino.value = \"\$terminoActual\";
document.frmAgrega.termino.focus()
document.frmAgrega.termino.select()
</SCRIPT>\n";

print "<P><center><a href=\"Terminos.pl?Op=$letra&terminoActual=$terminoActual\"><img
SRC=\"../img/logomaspetit2.gif\"><BR>Regresar</a></center>";

print $query->end_html;
```

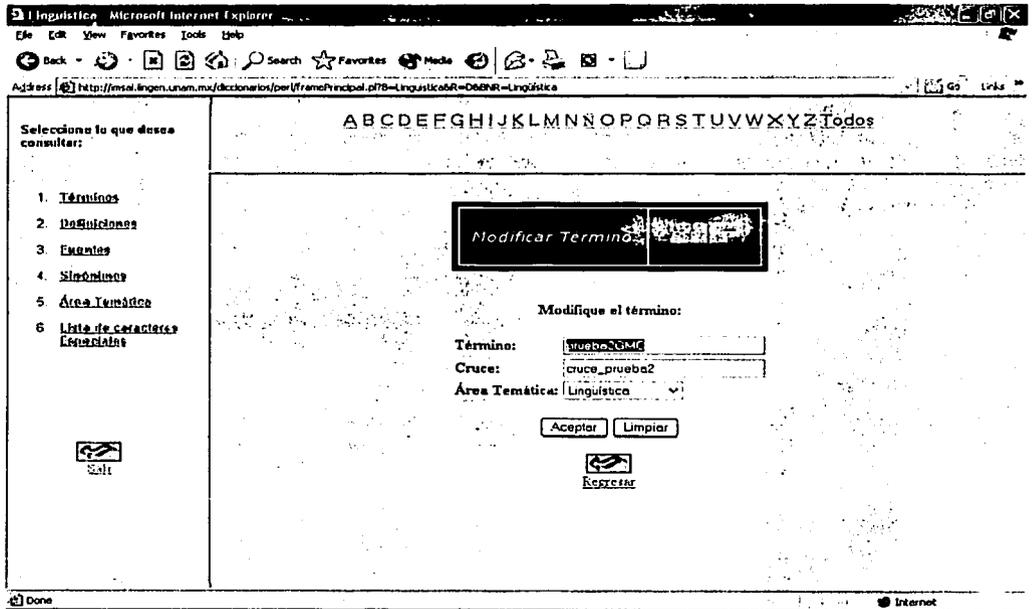
Si estamos en la última página que se mostró, y escogemos la opción de modificar el nuevo término que se acaba de agregar a la base terminológica, el sistema mostrará la siguiente página que le permite al usuario modificar el término seleccionado.

Cabe mencionar que al mostrarnos la página que permite modificar el término, se muestra lo que la base terminológica tiene actualmente, para facilitar la manipulación de la información, aunque también se le da al usuario la opción de limpiar los campos si los cambios van a ser radicales.



TESIS CON
FALLA DE ORIGEN

Una vez que se han hecho las modificaciones necesarias, y se ha presionado el botón de aceptar, el sistema mostrará una alerta para confirmar que se desea realizar la modificación.



TESIS CON
FALLA DE ORIGEN

Una vez que se presiona el botón de OK, el sistema realiza las modificaciones y regresa a la página principal de Términos, mostrando lo que se encuentra en la base de datos terminológicas en ese momento.

Seleccione lo que desea consultar:

- 1 [Términos](#)
- 2 [Definiciones](#)
- 3 [Ejemplos](#)
- 4 [Sinónimos](#)
- 5 [Área Leada](#)
- 6 [Lista de caracteres especiales](#)

Salir

A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z Todos

Términos

Seleccione un término:

Término: prueba2GM modificada

Agregar Eliminar

Término	Área	Cruce	Modificar	Eliminar	Responsable	Fecha	Modificación
prueba2GM modificada	Linguística	cruce prueba2	<input type="checkbox"/>	<input type="checkbox"/>	GMunozC	27 Jan 2003 2:05PM	GMunozC

TESIS CON
FALLA DE ORIGEN

A continuación, se muestra parte del código del Script que permite modificar un término de la base de datos terminológica.

```

function verificaTerm() {
    var termino = document.frmModifica.termino.value;
    var area = document.frmModifica.area.value;
    if(termino == "\\") {
        alert("\Debe introducir un termino")
        document.frmModifica.termino.focus()
        document.frmModifica.termino.select() }
    else {
        var resconf;
        resconf = window.confirm("\¿Desea modificar el término?\");
        if (resconf) {
            document.frmModifica.submit(); } } }

($X = MSSQL::DBlib->dblogin($l, $p, $s)) # Se conecta al servidor $X es la instancia del objeto
or print "<center><b>El login y password proporcionados son incorrectos</B></center>";
($X->dbuse($b) == SUCCEED )
or "<center><b>No se pudo usar la base de datos</center></b>";

print "<form name=frmModifica>";
print <<ET;
    <B>Modifique el término:</B><P><TABLE> <tr>
    <td><p><B>Término:</B></td>
    <td><input type="text" name="termino" size="30" MAXLENGTH="100"></td>
ET    print "<tr>
    <td><p><B>Cruce:</B></td>
    <td><input type="text" name='cruce' size='30' MAXLENGTH='50'></td>";
    print "<tr><td><p><B>Área Temática:</B></td>
    <td><select name='area' size='1'>";

#Conexiones a SQL
($X -> dbcmd("select * from AreaTematica order by AreaTematica asc"))
    # En el buffer de comandos del cliente almacena la sentencia SQL
    or print "No puede meter el comando en el buffer";
($X -> dbsqlxec) # Envía el buffer de comandos al Servidor
    or print "Fallo el select";
($X -> dbresults) # Trae los resultados
    or print "Fallo el results";

while(@data = $X->dbnextrow) {
    print "\t<option value=".$data[0].">".$data[1]."</option>\n";
}
print <<ET;
    </select></B></td>
    </TABLE>
    <P><input type="button" value="Aceptar". name="aceptar" onClick="verificaTerm()">
    <INPUT type="button" value="Limpiar" onClick="window.location.reload()">
    <INPUT type="hidden" value=1 name="mod">

```


3.7 Resumen del capítulo

En este capítulo se describieron algunos de los Scripts más representativos del sistema desarrollado. También se presenta la estructura de algunas páginas estáticas y algunas páginas dinámicas más representativas del sistema.

Después se mencionó cómo se creó la estructura de las bases de datos del sistema, y el espacio reservado para cada una de ellas.

También se muestran algunos de los Scripts de creación de los objetos de la base de datos. Y se muestran algunos Procedimientos almacenados que se realizaron para ejecutar algunas opciones del sistema desarrollado.

Se mencionaron las principales características del Servidor Web utilizado.

Se incluyó a detalle cómo funciona el sistema para la administración en Web del sistema.

Se habló sobre los diferentes tipos de pruebas del sistema que se llevaron a cabo en el sistema.

Y como parte final del capítulo, se mostraron algunas de las páginas ya implantadas con un ejemplo de navegación muy sencillo, junto con el código que generan las páginas Web mostradas.

TESIS CON
FALLA DE ORIGEN

CONCLUSIONES

El proyecto "Desarrollo de un sistema para consultar y alimentar un diccionario terminológico en Web" forma parte de la primera etapa de desarrollo de un proyecto donde se elabora un diccionario integral para búsquedas semasiológicas y onomasiológicas en el área de lingüística.

Este proyecto surgió por la necesidad de tener una base de datos que permitiera recopilar los términos en el área de lingüística y sus definiciones según diversas fuentes.

De forma general, se cumplieron con los requerimientos y las expectativas del sistema, ya que permite a los usuarios introducir la terminología de forma ágil y sencilla desde cualquier computadora con acceso a Internet.

Por medio de encuestas y entrevistas se comprobó que los usuarios pueden navegar a través de las páginas sin ninguna dificultad. Además, señalaron que la forma de presentación de las páginas permite una mayor facilidad de acceso a la información que se desee consultar, así como la ayuda proporcionada para los caracteres especiales permite un mejor manejo del sistema.

Además, las reglas que se solicitaban del sistema (explicadas en el capítulo dos) fueron cubiertas en su totalidad por éste, de forma que el sistema permite un fácil acceso a través del Web de forma segura, sin impedir que personas ajenas no tengan acceso a él. Tampoco se presentaron problemas con los permisos asignados a los diferentes tipos de usuarios, mostrando siempre única y exclusivamente lo que cada uno de los usuarios puede acceder.

El sistema desarrollado cubre con las expectativas planteadas, aunque cabe señalar que se pueden agregar muchas características que podrían hacer aún más interesante al sistema, como por ejemplo, agregar la opción de hacer consultas avanzadas de la información contenida en las bases de datos del sistema, etc. Lo cual podría ser implementado en el sistema en algún trabajo posterior.

En cuanto al sistema de administración, éste sólo es accesible a los desarrolladores del proyecto y realiza las acciones para las cuales fue desarrollado de manera eficaz.

Además, cabe mencionar que este sistema ha comenzado a evolucionar en una segunda versión y existe una tesis alrededor de este tema.