



UNIVERSIDAD MICHOACANA DE
SAN NICOLAS DE HIDALGO

FACULTAD DE INGENIERIA ELECTRICA

*“Herramienta para el manejo de datos de la Secretaría
de Planeación y Desarrollo Estatal (SEPLADE)”*

TESIS

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELECTRICISTA

PRESENTA:
GABRIEL VELAZQUEZ MENDOZA

ASESOR:
M.I. SAMUEL PEREZ AGUILAR

MORELIA MICHOACAN, DICIEMBRE DE 2005





Universidad Michoacana de San Nicolás de Hidalgo
Facultad de Ingeniería Eléctrica

Oficio No.706/2004-2005

C. GABRIEL VELAZQUEZ MENDOZA
PASANTE DE INGENIERIA ELECTRICA

Presente

En atención a su atenta solicitud de fecha 17 de Agosto del año en curso, me permito comunicarle que se aprueba el Tema de Tesis propuesto para presentar Examen Recepcional, en la carrera de Ingeniería Eléctrica.

El Tema de Tesis aprobado **“Herramienta para el manejo de datos de la Secretaria de Planeación y Desarrollo Estatal (SEPLADE)”**, que deberá desarrollarse bajo el siguiente índice:

	Resumen
Capítulo 1.-	Introducción
Capítulo 2.-	Abstracción de datos usando PHP y MySQL.
Capítulo 3.-	Diseño de la base de datos
Capítulo 4.-	Interfaz del sistema
Capítulo 5.-	Conclusiones
	Bibliografía

Para tales efectos fungirá como asesor de su tesis el **M.I. Samuel Pérez Aguilar**.

Atentamente
Morelia, Mich., a 23 de Agosto del 2005

Ing. Gilberto I. López Pedraza
Encargado de la Dirección de la
Facultad de Ingeniería Eléctrica.

GILP/srd

Ciudad Universitaria,
Edificio "A" Primer Piso,
Av. Francisco J. Mújica S/N,
C.P. 58030, Morelia, Mich.
Tel. 01 (443) 3 223500 Ext. 1115,
Tel./Fax 01(443) 3265776,
<http://www.fie.umich.mx/>



Agradecimientos

A mis tíos por su comprensión durante los años de estudio, quienes no solo han sido mis tutores si no que pasaron a ser el motor que me impulso hasta el termino de este trabajo de tesis.

A mis hermanos y primas por confiar en mí.

Al M.I. Samuel Pérez Aguilar por su valioso asesoramiento que sin duda facilito la elaboración y conclusión de este trabajo, más aun por su humanidad solidaria.

A todos mis compañeros.

Por ultimo a todos aquellos que de una manera u otra me alentaron a concluir.

Gracias

Resumen

El presente trabajo tiene como objetivo el de servir como herramienta de captura de datos en el proceso de integrar y publicar cada año información estadística, específicamente el de integrar información en el proceso de análisis estadístico en el ámbito estatal, esto se logrará implementando una aplicación que permite generar formularios con la finalidad de agregar y actualizar los datos de una forma rápida y sencilla.

El desarrollo de esta herramienta de captura de datos implicó la elaboración de una interfaz en la cual se le permite al administrador crear las vistas para los usuarios finales, es decir, desde el navegador crear las tablas que estarán en la base de datos y las vistas finales para cada uno de los usuarios sin que él tenga que conocer los comandos de bases de datos, esto se logra por medio de formularios pertenecientes a la aplicación, elaborados con scripts de PHP y la interrelación de estos con MySQL como gestor de base de datos desde el navegador de Internet, de esta forma se obtienen páginas dinámicas. Es necesario hacerlo de esta forma debido a que la información es proporcionada por distintas fuentes (capturistas de datos), las cuales pueden estar ubicadas en distintos puntos geográficos en el estado.

La información proporcionada por las fuentes será almacenada en la base de datos propia, también se generan los informes de los datos almacenados por cada una de ellas, todo esto sin la necesidad de que los usuarios finales conozcan la parte interna del programa, haciendo uso de la abstracción de datos. Los usuarios podrán acceder a la aplicación desde su propio hogar u oficina a través de una computadora conectada a Internet. Básicamente el administrador debe definir en la herramienta los cuadros de información con la ayuda de formularios, en los cuales se vaciarán mas tarde los datos que serán almacenados y solo podrán ser consultados en su totalidad por el administrador.

La herramienta es capaz de reconocer el tipo de usuario que accede, esto con el objetivo de que solo les sean presentados los formularios que les pertenecen y no puedan borrar o agregar datos donde no les fue asignado.

De esta forma se facilita cumplir con los objetivos que persigue la SEPLADE (Secretaria de Planeación y Desarrollo Estatal) en particular:

- a) ofrecer las estadísticas relevantes más recientes del contexto estatal, bajo normas de presentación que faciliten la consulta.
- b) servir de base para el desarrollo de un banco de datos a nivel de cada estado, en donde se almacene y sistematice la información con mayores niveles de desagregación.
- c) constituir un instrumento de diagnóstico de la información generada en las entidades por los diferentes sectores, que permita la identificación de problemas básicos que requieran atención y solución. ^[9]

Estos objetivos son congruentes con el proceso de modernización nacional^[10], el cual, sin duda, debe sustentarse en más y mejor información estadística, disponible en el momento, el lugar y con las características que se requieran, por lo tanto es muy importante que el tiempo de adquisición de los datos y el análisis sea de manera mas rápida y eficaz, estas razones son el sustento principal para el desarrollo de esta herramienta.

La herramienta de captura de datos residirá en el lado del servidor Web (Apache), y trabajará apoyándose en él, junto con el sistema de gestión de la base de datos MySQL y los scripts de PHP. Todos los paquetes son de distribución libre ya que esta herramienta será un eslabón muy importante dentro del sistema de adquisición de datos, logrando de esta forma reducir costos y tiempo.

Índice:

Capítulo 1. Introducción 1

1.1.	Utilidad de la herramienta	1
1.2.	Aprovechando nuevas tecnologías	10
1.3.	Breve descripción del funcionamiento de la herramienta de captura de datos.....	11
1.4.	Descripción de capítulos.....	12

Capítulo 2. Abstracción de datos usando PHP y MySQL 15

2.1.	Modelo de datos	16
2.2.	Niveles de abstracción.....	17
2.3.	¿Porque utilizar PHP y MySQL?	20
2.3.1.	Algunas de las características de PHP.....	22
2.3.2.	Algunas de las características de MySQL	24
2.4.	Arquitectura de bases de datos en la Web	24
2.4.1.	PHPLIB	26
2.5.	Cifrado con MD5 en PHP.....	29
2.6.	Resumen	31

Capítulo 3 Diseño de la base de datos 32

3.1.	Esquema del nivel externo.....	32
3.2.	Definiendo el nivel o esquema conceptual.....	34
3.3.	Esquema lógico	35
3.3.1.	Tabla principal.....	36
3.3.2.	Tabla a_”id “ del apartado.....	37
3.3.3.	Tabla fuentes	37
3.3.4.	Cuadros.....	38
3.3.5.	Tabla auth_user_md5	41
3.3.6.	Tabla fuentespara.....	42
3.4.	Resumen	45

Capítulo 4. Interfaz del sistema 46

4.1.	Autenticación de usuarios	47
4.2.	Diferentes tipos de usuarios.....	49
4.3.	El administrador de la herramienta.....	50
4.4.	Componentes de la vista del administrador.....	53
4.4.1.	Apartados.....	53
4.4.2.	Administración de usuarios y fuentes.....	54
4.4.3.	Nuevos cuadros	56
4.4.4.	Informes (consultas)	59

4.4.5.	Adición de registros.....	61
4.5.	Manual del usuario	63
4.5.1.	Usuarios finales.....	63
4.5.2.	Agregando los datos	63
4.6.	Resumen	67

Capítulo 5. Conclusiones 68

A-1.	Bibliografía.....	A1
------	-------------------	----

Capítulo 1. Introducción

En este capítulo se presenta un resumen de la importancia que tiene Internet en la actualidad, debido a los variados servicios que presta ya que éstos nos permiten realizar muchas actividades de modo rápido y económico, en la comodidad del hogar o la oficina, ejemplos de tales servicios son el correo electrónico, los servicios financieros por parte de algunas organizaciones Bancarias, y el World Wide Web, entre otros, lo que permite realizar tareas de una forma mas innovadora como la que presento en este trabajo de tesis.

El objetivo de esta tesis es desarrollar una herramienta que será un medio rápido y preciso para el flujo de información, facilitando la recopilación de datos de diferentes fuentes tales como el INEGI, Secretaria de Comunicaciones y Transportes, Comisión Forestal del Estado de Michoacán, H. Ayuntamiento de Morelia, UMSNH, etc., reduciendo de esta forma substancialmente el tiempo que se requiere para repartir los formatos y recuperarlos con los datos que posteriormente son vaciados en alguna base de datos o en una hoja de cálculo del esquema actual, dicha información es utilizada para su publicación, previo análisis estadístico por la SEPLADE (Secretaría de Planeación y Desarrollo Estatal), esta dependencia será el administrador de la herramienta en el nuevo esquema que propongo donde tanto el administrador como los usuarios tendrán una clave de acceso, por otra parte los usuarios serán integrantes de diferentes instituciones de los sectores público, privado y social.

1.1. Utilidad de la herramienta

Dentro del proceso de desarrollo nacional, la información estadística constituye un insumo fundamental para la elaboración de diagnósticos, la formulación, instrumentación y control de planes y programas, así como para evaluar los resultados de la gestión pública; así mismo, es un elemento indispensable en la realización de estudios e investigaciones por parte de usuarios del sector privado, académico y del público en general. Esto es así, en virtud de que con la información estadística es posible caracterizar y conocer los fenómenos económicos y sociales, lo cual permite el análisis y la toma de decisiones para alcanzar los objetivos planeados.

Consecuente con lo anterior, y de acuerdo con la legislación en la materia, el **Instituto Nacional de Estadística, Geografía e Informática (INEGI)** genera

información estadística sobre una amplia variedad de temas económicos y sociodemográficos, a partir de un programa de censos nacionales, otro de estadísticas a corto plazo (encuestas en hogares y en establecimientos) y uno más de estadísticas continuas, sociodemográficas y económicas, cuyos resultados se ponen al alcance de los usuarios a través de libros y medios magnéticos. Simultáneamente, el **INEGI** atiende otros dos programas estadísticos sustentados en parte por los tres ya mencionados: el de Contabilidad Nacional, mediante el que se produce estadística sobre los grandes agregados económicos nacionales, y el de **Integración de Estadísticas Sectoriales, Estatales y Regionales**, por medio del cual, bajo un enfoque de complementariedad, se integran estadísticas seleccionadas producidas por el propio Instituto, y estadísticas generadas por diferentes instituciones de los sectores público, privado y social, lo que permite ofrecer a los usuarios, en un solo producto, la información más relevante que se encuentra disponible sobre un sector o actividad económica, o bien, sobre diversos temas, pero teniendo como circunscripción una determinada entidad federativa o un municipio. Precisamente, dentro de ese programa se ubica el proyecto de Anuarios Estadísticos de los Estados (**AEE**), mismo que se orienta a generar productos, cuyo contenido cubre temas económicos y sociodemográficos con cifras desagregadas al nivel de los municipios que conforman cada estado, es aquí donde entra en juego la herramienta propuesta.

El proyecto **AEE** tiene como objetivo integrar y publicar cada año, información estadística del ámbito estatal, sobre la magnitud, estructura, distribución, comportamiento e interrelaciones de universos y fenómenos de interés general, para el conocimiento básico de cada entidad. Persigue en particular:

- a) ofrecer las estadísticas relevantes más recientes del contexto estatal, bajo normas de presentación que faciliten la consulta;
- b) servir de base para el desarrollo de un banco de datos a nivel de cada estado, en donde se almacene y sistematice la información con mayores niveles de desagregación y
- c) constituir un instrumento de diagnóstico de la información generada en las entidades por los diferentes sectores, que permita la identificación de problemas básicos que requieran atención y solución.

Estos objetivos son congruentes con el proceso de modernización nacional, el cual, sin duda, debe sustentarse en más y mejor información estadística, disponible en el momento, el lugar y con las características que se requieran, por lo tanto es muy importante que el tiempo de adquisición de los datos y el análisis

sea de manera mas rápida y eficaz, estas razones son el sustento principal para el desarrollo de esta herramienta.

La importancia de la herramienta, radica en su contribución al desarrollo de los **Sistemas Estadísticos de los Estados**, ya que su ejecución coadyuva a consolidar aspectos básicos relativos a los procesos de integración y difusión de la información, especialmente el establecimiento de flujos continuos de estadísticas entre los generadores y las fuentes.

Para tener un panorama general de la aplicación de la herramienta para adquirir los datos, en el diagrama de la Fig. 1.1, se muestra el lugar que ocupa dentro del proceso de la información estadística.

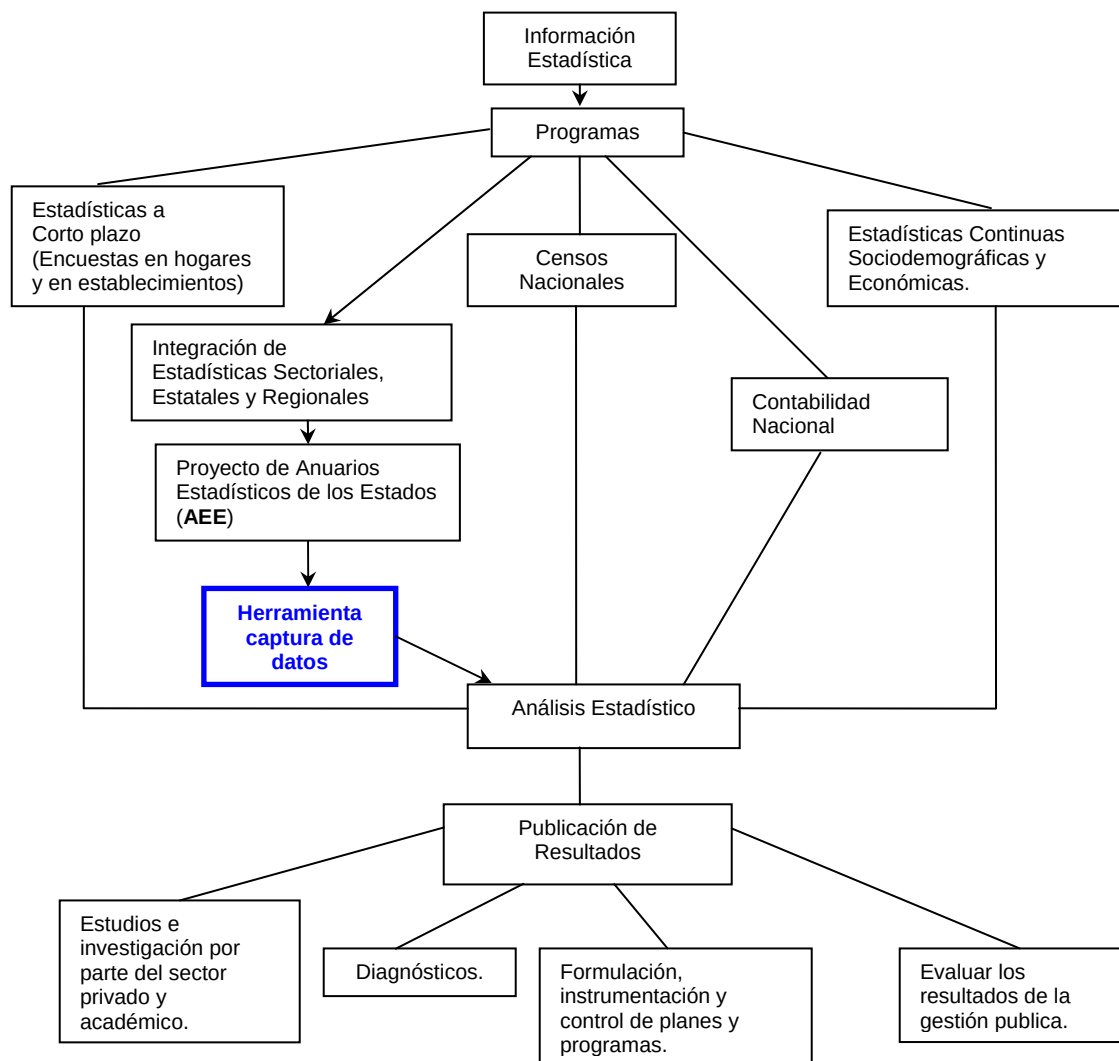


Fig. 1.1 Diagrama de procesamiento de la información estadística.

Prácticamente en el diagrama de la figura 1.1 muestro donde sería implementada esta herramienta de adquisición de datos, pertenecerá al Proyecto de Anuarios Estadísticos de los Estados (AEE)^[9] el cual a su vez forma parte del Programa de Estadísticas Sectoriales, Estatales y Regionales, tienen como objetivo presentar información estadística a los diferentes sectores de la población, para realizar el análisis estadístico es necesario recabar información la cual será adquirida por la herramienta posteriormente.

El proyecto de Anuarios Estadísticos Estatales en el cual está englobada la herramienta, integra actualmente veintidós apartados que pueden ser más en un futuro; el primero de los temas, el de Aspectos Geográficos, proporciona una referencia sobre el aspecto físico al que corresponden las estadísticas; se continúa con el de Medio Ambiente, a través del cual se da una aproximación al conocimiento del estado que guardan algunos de los recursos naturales en la entidad, así como las medidas instrumentadas por parte de la administración pública para la preservación ambiental. Enseguida se presenta lo relativo a Estado y Movimiento de la Población, para dar lugar en secuencia inmediata, a un bloque de temas referidos a los satisfactores socialmente necesarios para el bienestar, desarrollo y prevalencia de la misma, incorporando estadísticas sobre: Vivienda e Infraestructura Básica para los Asentamientos Humanos, Salud, Educación y, Seguridad y Orden Público. También, se ofrece un apartado sobre Empleo y Relaciones Laborales, que sirve de enlace con los de corte económico, mismos que son abordados en los siguientes términos: el de Información Económica Agregada, da una visión sobre la magnitud y principales características del aparato productivo de la entidad, aspecto que se detalla en los apartados subsecuentes sobre los sectores primario (Agricultura, Ganadería, Silvicultura y Pesca), secundario (Minería, Industria Manufacturera, Construcción y Electricidad) y terciario (Comercio, Turismo, Transportes y Comunicaciones, Servicios Financieros y Finanzas Públicas); cada uno de los apartados anteriores contendrá un número variable de cuadros para ser llenados por las diferentes fuentes.

La estructura deseada de la herramienta de capturas de datos es que inicialmente no contenga ningún apartado; después de instalada se podrán ir agregando uno a uno los veintidós apartados actuales, después se procederá a agregarles los cuadros correspondientes, también se podrán agregar nuevos conceptos a los cuadros, aún después de haber sido llenados, el responsable de realizarlo será el administrador, que hará este trabajo por medio de formularios los cuales darán la flexibilidad para generar los nuevos cuadros que se requieran, sin que tenga que modificar nada del código, de esta forma, se tendrá la posibilidad de agregar más de los veintidós apartados y sus cuadros, si así se requiere para la adquisición de nuevos datos. Los apartados por agregar hasta el momento son mostrados en la siguiente tabla:

1. Aspectos Geográficos
2. Medio Ambiente
3. Estado y Movimiento de la Población
4. Vivienda e Infraestructura Básica para los Asentamientos Humanos
5. Salud
6. Educación
7. Seguridad y Orden Público
8. Empleo y Relaciones Laborales
9. Información Económica Agregada
10. Agricultura
11. Ganadería
12. Silvicultura
13. Pesca
14. Minería
15. Industria Manufacturera
16. Construcción
17. Electricidad
18. Comercio
19. Turismo
20. Transportes y Comunicaciones
21. Servicios Financieros
22. Finanzas Públicas

Tabla 1.1 Apartados

Los contenidos de los Anuarios son integrados predominantemente con estadísticas obtenidas a partir de registros administrativos de las fuentes que colaboran a las cuales se les asignará uno o mas nombres de usuarios junto con sus claves o passwords para controlar el acceso al llenado de los cuadros; en todos los casos, las cifras corresponden al último año que esté disponible. Destacando que una fuente puede integrar datos en cuadros de diferentes apartados, algunas de las fuentes mencionadas en la tabla siguiente.

Asociación Mexicana de la Industria Automotriz, Asociación Civil
Banco Nacional de Crédito Rural, Sociedad Nacional de Crédito
Banco de México
Comisión Estatal de Derechos Humanos
Comisión Federal de Electricidad
Comisión Federal de Telecomunicaciones
Comisión Forestal del Estado
Comisión Nacional de Arbitraje Médico
Comisión Nacional de Salarios Mínimos
Comisión Nacional del Agua
Dirección de Seguridad Pública y Tránsito del Estado

Honorable ayuntamiento de Morelia
Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado
Instituto del Fondo Nacional de la Vivienda para los Trabajadores
Instituto Nacional de Educación para los Adultos
Instituto Mexicano del Seguro Social
Instituto Nacional de Antropología e Historia
Instituto Nacional de Estadística, Geografía e Informática
Instituto Nacional de Migración
Leche Industrializada Conasupo, Sociedad Anónima de Capital Variable
Nacional Monte de Piedad
Oficialía Mayor de Gobierno
Procuraduría Federal de Protección al Ambiente
Procuraduría General de Justicia del Estado
Procuraduría General de la República
Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación
Secretaría de Comunicaciones y Transportes
Secretaría de Desarrollo Social
Secretaría de Economía
Secretaría de Educación del Estado
Secretaría de Hacienda y Crédito Público
Secretaría de Medio Ambiente Recursos Naturales
Secretaría de Salud del Estado de Michoacán
Secretaría del Trabajo y Previsión Social
Secretaría Estatal de Turismo
Secretaría General de Gobierno del Estado
Servicio de Administración Tributaria
Servicio Postal Mexicano
Sistema para el Desarrollo Integral de la Familia
Telecomunicaciones de México
Tesorería General del Estado

Tabla 1.2 Fuentes

Estas fuentes serán las encargadas de llenar los diferentes cuadros previa creación y asignación por parte del administrador de la herramienta, pero pudiendo agregar mas fuentes y a su vez mas usuarios para las distintas fuentes.

En la figura 1.2 muestro un ejemplo de como los formatos asignados para la fuente INEGI, se dividen en tres apartados y estos a su vez se subdividen en uno o varios cuadros.

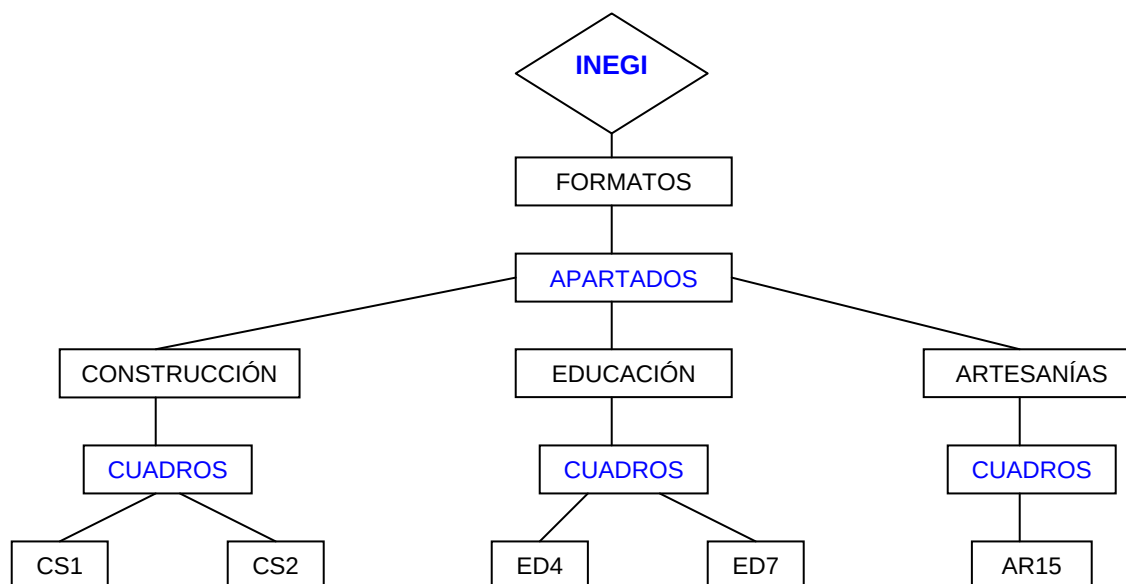


Fig. 1.2 Distribución de formatos

Por la amplia cobertura temática de los anuarios, la desagregación municipal del contenido y la periodicidad de los productos, se les considera documentos de utilidad para todo tipo de usuario que tenga interés en la información estadística de cada estado, tanto los especializados, como aquéllos cuyos requerimientos se ubican en un nivel de consulta básica.

Por lo que se refiere a su difusión, los veintidós anuarios se publican en forma impresa previo análisis estadístico, así como en la página del INEGI y del SEPLADE ^[9] en Internet; asimismo, de acuerdo con los compromisos firmados con autoridades locales, algunos de ellos se ofrecen también en sistemas de consulta en discos compactos.

Con esto y la herramienta se atenderán las exigencias de modernización del servicio, se cubren las posibilidades de acceso más amplias y se mejora su alcance y características.

De acuerdo a la practica de recopilación actual de los datos y que es mostrada en la figura 1.3, es notable que se requiere de un gran esfuerzo, principalmente por que es necesario imprimir grandes volúmenes de hojas para ser distribuidos a las diferentes fuentes, invirtiendo tiempo en la distribución por cuestiones de traslado, cada una de las fuentes tendrá que llenar a mano cada uno de los formatos que le son asignados, dichos formatos se subdividen en

apartados, tal como se menciona anteriormente, una vez que son entregados los formatos, las fuentes llena estos a lápiz, haciendo que el proceso sea mas lento y con un mayor aumentando el numero de errores de captura ya que enseguida son regresados a la Secretaría de Planeación y Desarrollo Estatal par que sean capturados en alguna base de datos, con el objetivo de llevar acabo el análisis estadísticos correspondiente para que finalmente sean publicados.

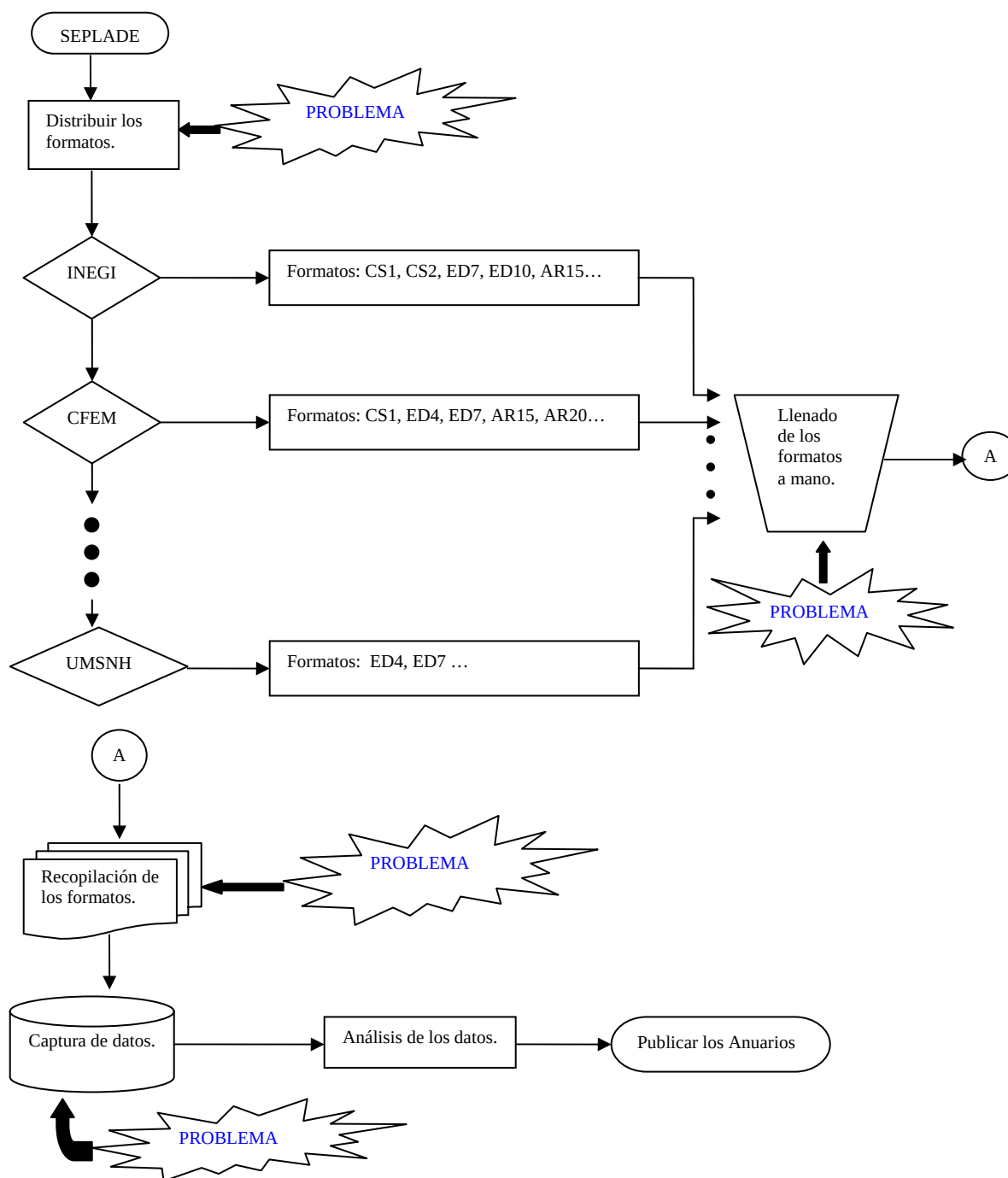


Fig. 1.3 Situación actual

La figura 1.3 muestra el proceso actual para la recopilación de la información donde los principales problemas son por cuestiones económicas y de tiempo, ya que repartir los formatos impresos implica recorrer grandes distancias produciendo gastos, además de que la información no se obtiene de inmediato como para esperar a que sean llenados los formatos y llevárselos en el mismo momento, debido a ello es necesario hacer un doble viaje, lo que implica más gastos y más horas hombre. Estos formatos después de ser llenados por las fuentes es necesario capturarlos, para ser analizados y publicados.

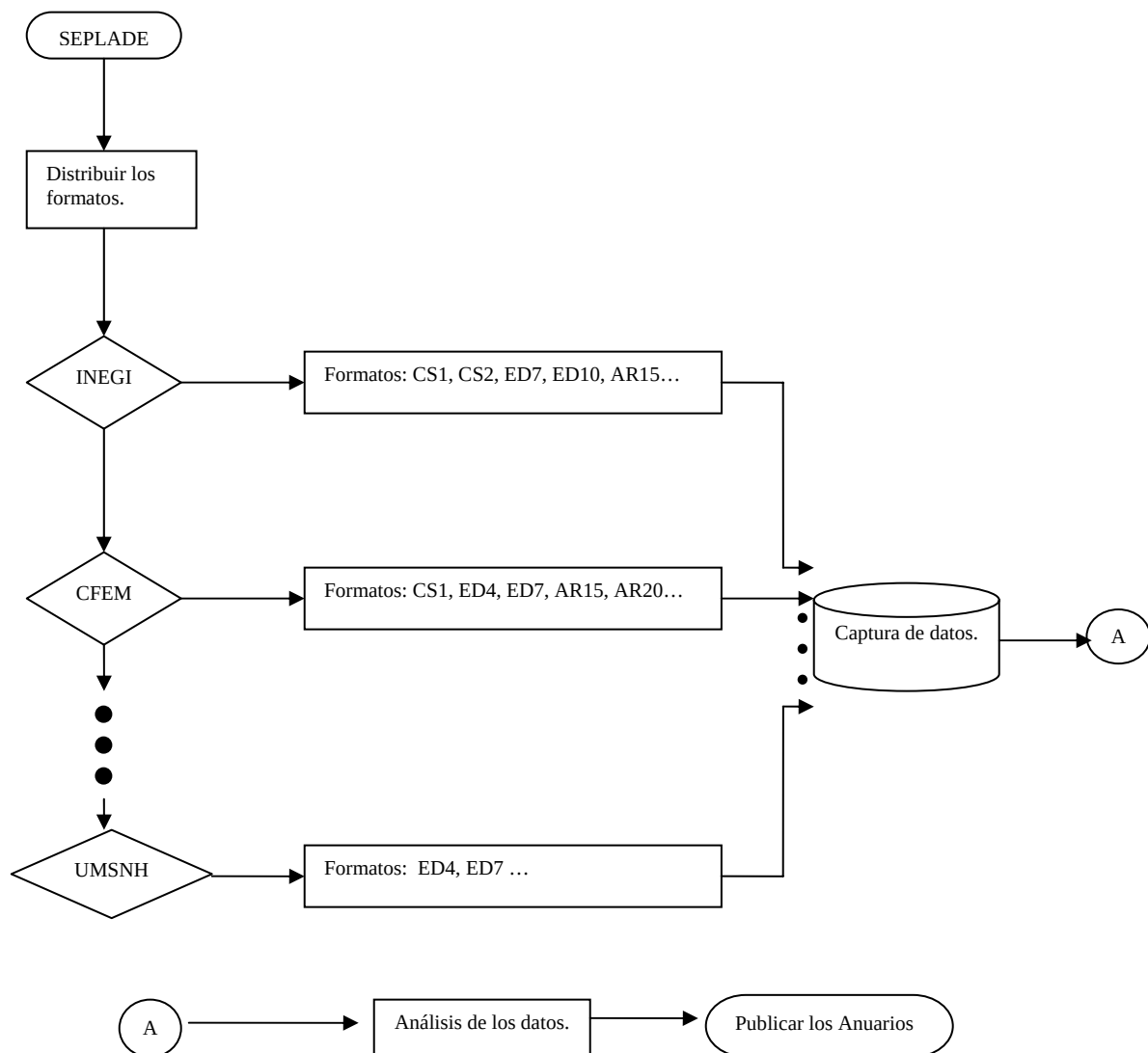


Fig. 1.4 Propuesta de mejora

Con la herramienta de captura de datos se eliminan los gastos por desplazamiento para la distribución de los formatos y al mismo tiempo la recopilación de los mismos, consecuentemente se disminuye el tiempo designado para dicha labor. Con Igual importancia, esta la captura de los datos, a diferencia del esquema actualmente utilizado y que es mostrado en la figura 1.4, estos no se tienen que volver a capturar, ya que serán las mismas fuentes quienes se encargarán de guardarlos en una base de datos propia de la herramienta y que va a estar en el lado del servidor, esto se hará aprovechando las nuevas tecnologías, logrando reducir de esta forma el tiempo y minimizando los errores por transcripción.

1.2. Aprovechando nuevas tecnologías

El gran auge de Internet junto con la aparición del World Wide Web (WWW), logró que Internet goce de la actual popularidad que conocemos. No obstante, Internet proporciona en la actualidad una gran cantidad de servicios, entre los que podemos mencionar:

- Correo electrónico: permite comunicarse con cualquier persona que tenga acceso a Internet, de modo rápido y económico (yahoo, Hotmail, entre otros servicios gratuitos y de pago), junto con esto se tiene acceso a la mensajería instantánea MSN.
- Transferencia de archivos (FTP, SSH): permite acceder a los sistemas de archivos de cualquier computadora de acceso público.
- Comercio electrónico: puedes realizar compras a través de la Web. (por ejemplo: Mercado libre).
- Servicios financieros a través de la Web. (Bancos).
- Búsqueda de información de cualquier tipo. (Google, Altavista, etc.).
- Foros de discusión.
- World Wide Web: proporciona la posibilidad de acceder a documentos de hipertexto.

Paralelamente al auge de la Internet, se ha producido una gran evolución en el desarrollo y mantenimiento de las páginas Web. Inicialmente, la gran mayoría de páginas Web eran de tipo estático, es decir, el usuario sólo podía visualizar la información que contenían. La tendencia actual es la de crear páginas Web

dinámicas (también conocidas como aplicaciones Web), las cuales permiten al usuario una mayor interacción, ofreciendo una información mejor dirigida, escogida y elaborada. La facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

Una mayor interacción por parte del usuario, conlleva la necesidad de reunir y procesar la información proporcionada por él y emitir un resultado. Esto implica que el desarrollo de una aplicación Web lleve asociada una capa de programación. Esto quiere decir, que un usuario que quiera publicar datos almacenados en una base de datos a través de una aplicación Web, por simple que la aplicación sea, necesita una capa de programación. El problema es que no todos los usuarios disponen de los conocimientos suficientes para desarrollar su propia aplicación.

Esta herramienta proporciona una aplicación Web que permite generar formularios con la finalidad de agregar y actualizar los cuadros pertenecientes a las diferentes fuentes de la base de datos propia, también se generan los informes de los datos almacenados por cada una de la fuentes, todo esto sin la necesidad de conocer la parte interna del programa, por parte del usuario final. A esta simplificación se le llama también abstracción de datos. Los usuarios podrán acceder a la aplicación desde su propio hogar u oficina a través de una computadora conectada a Internet. Básicamente, el administrador debe definir en la herramienta los cuadros de información con la ayuda de formularios, en los cuales se vaciarán mas tarde los datos que serán almacenados.

El administrador de la herramienta, no tendrá que crear páginas ni formularios individualizados para los distintos usuarios y no tendrá necesidad de conocer el código de la herramienta.

1.3. Breve descripción del funcionamiento de la herramienta de captura de datos

La herramienta residirá en el lado del servidor Web (Apache), y trabajará apoyándose en él, junto con el sistema de gestión de la base de datos MySQL y los scripts de PHP; MySQL y PHP se describen en el capítulo 2.

Además, serán los propios usuarios los encargados de proporcionar los datos, así como de realizar las pertinentes actualizaciones de los mismos. La herramienta proporciona un conjunto de aplicaciones Web, que permiten que los datos sean accedidos solo por los usuarios con los permisos correspondientes desde su navegador. Se generará una aplicación Web independiente para cada

usuario. Estas aplicaciones Web serán la vía utilizada, por los usuarios propietarios de las aplicaciones, para realizar el mantenimiento de los datos. Por su parte el Administrador de la herramienta, será el encargado de instalar la herramienta, administrar y mantener las aplicaciones Web y la base de datos asociada a la misma.

Un concepto muy importante es la *abstracción* que se puede definir como la capacidad de examinar algo sin preocuparse de los detalles internos. Es suficiente conocer que un procedimiento dado realiza una tarea específica. El cómo se realiza la tarea no es importante; mientras el procedimiento sea fiable se puede utilizar sin tener que conocer cómo funciona su interior.

La *abstracción de datos* permite no preocuparse de los detalles no esenciales.

1.4. Descripción de capítulos

En el presente capítulo se mostró que la información estadística es fundamental para la elaboración de diagnósticos, la formulación, instrumentación y control de planes y programas y para evaluar los resultados de la gestión pública; así mismo, es un elemento indispensable en la realización de estudios e investigaciones por parte de usuarios del sector privado, académico y del público en general. Ya que el proyecto **AEE** tiene como objetivo integrar y publicar cada año, información estadística del ámbito estatal, sobre la magnitud, estructura, distribución, comportamiento e interrelaciones de universos y fenómenos de interés general, es necesario tener la información para el conocimiento básico de cada entidad. De aquí la importancia de la herramienta, ya que aportará agilidad a la recolección de los datos.

Para lograr aportar dicha herramienta se utilizan páginas Web dinámicas pero esto implica que su desarrollo lleve asociada una capa de programación, que sin embargo queda oculta a los usuarios finales.

También se hace una descripción con diagramas de los problemas a los cuales se tienen que enfrentar al momento de recopilar los datos, de igual forma se muestra la solución a la cual se llega para facilitar éstas tareas, soluciones que gracias al avance tecnológico en el ámbito de la informática y al gran desarrollo de la comunicación y transmisión de la información a través de la red.

En el **capítulo dos** se mostrará la importancia de los modelos de datos y los niveles de abstracción, debido a que éstos nos ayudan a ocultar características sobre el almacenamiento físico de los datos que la mayoría de los usuarios no

necesita conocer. Se mencionan los niveles de abstracción ya que nos ofrecen las descripciones de los datos, ayudando a entender la correspondencia o transformación entre ellos. Como parte fundamental del nivel externo tenemos las vistas que nos proporcionan un mecanismo de seguridad, ocultando partes de la base de datos que determinados usuarios no necesitan conocer.

Se mencionan las principales características de MySQL y PHP tales como: portabilidad, facilidad de aprendizaje, costo, número de usuarios, código fuente entre otras, ya que nos da la pauta para utilizarlos para el desarrollo de la herramienta.

Cabe mencionar que la herramienta es de tipo dinámico, es decir, la correspondencia entre los niveles de abstracción no es tan sencilla como los del tipo estático, que consta de solo dos objetos en la cuales uno de ellos hace una petición y el otro ofrece la respuesta, debido a que la arquitectura que describe la herramienta está elaborada con interfaces para el manejo de bases de datos es mas compleja, la correspondencia tiene mas fases implicando a mas de dos objetos: Un Navegador, servidor Web (Apache), PHP y MySQL, por lo cual es mas compleja.

Como parte final del capítulo se muestra una de las librerías (PHPLIB) que es de gran utilidad en este trabajo de tesis, debido a que nos proporciona una gran ayuda con sus clases y métodos para resolver la problemática de autenticación de usuarios, manejo de sesiones, tablas o vistas y lo mas importante que es el control de las contraseñas o password's, ya que utiliza un algoritmo unidireccional para el cifrado de los password's (MD5).

El **Capítulo tres** se enfoca al nivel lógico, dando paso a la descripción principalmente de la estructura de la base de datos, pero no sin antes entender el nivel externo ya que para pasar de un nivel a otro es necesario conocer las entidades y sus atributos, lo cual implica tener conocimiento de las vistas que serán presentadas a cada uno de los usuarios, por lo tanto, es necesario tener presentes las reglas o requisitos que la herramienta debe de satisfacer para el usuario final, teniendo presente que los usuarios finales no tienen conocimiento alguno de lo que es la programación.

Prácticamente en éste capítulo se darán a conocer las tablas que formarán parte de la descripción de la base de datos, utilizando el diccionario de datos de cada una de ellas, para tal descripción. Se describen los campos que contienen cada una de las tablas con el objetivo de tener conocimiento de cómo es la transformación con el nivel externo.

Se muestra en el **Capítulo 4** la parte técnica de las transformaciones entre los niveles de abstracción, mostrando solo algunos fragmentos clave del código, con el objetivo de que posteriormente se le realicen algunas mejoras a la Herramienta de captura de datos. Aclarando que solo se muestra la lógica de programación más no se pretende que sea un manual de PHP. Al mismo tiempo se da a conocer el funcionamiento de la herramienta, pero sin darle tanto énfasis ya que se diseñaron las pantallas o vistas de tal forma que sea intuitivo, simple y secuencial para los usuarios en cada uno de los pasos que son necesarios para su uso.

Finalmente el **Capítulo 5** se refiere a las conclusiones.

Capítulo 2. Abstracción de datos usando PHP y MySQL

Construir correctamente las abstracciones es una tarea difícil, la dificultad tiene que ver con el hecho de que siempre resulta complicado captar la esencia de un problema y articularlo de un modo conciso.

Los modelos de datos son el instrumento principal para ofrecer la abstracción de los datos, los modelos conceptuales, modelos físicos y modelos lógicos se utilizan para representar el *esquema de la base de datos*.

En consecuencia el sistema de gestión de base de datos debe transformar cualquier petición por parte de los usuarios finales expresada en términos de un esquema externo a una petición expresada en términos del esquema conceptual, y luego, a una petición en el esquema interno, que se procesará sobre la base de datos almacenada. Si la petición es de una obtención (consulta) de datos, será preciso modificar el formato de la información extraída de la base de datos almacenada, para que coincida con la vista externa del usuario.

Unos de los problemas al momento de desarrollar una aplicación web es elegir o adaptarse al hardware para el servidor, el sistema operativo, el servidor web, un sistema de gestión de bases de datos y el interprete de los script's. Existen soluciones comerciales y soluciones libres para los aspectos anteriores pero en nuestro caso como no se quiere estar dependiendo del factor económico se seleccionaron MySQL y PHP como gestión de bases de datos y lenguaje PHP con sus script's por su gran portabilidad.^[4]

2.1. Modelo de datos

Una de las características de los sistemas gestión de bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar características sobre el almacenamiento físico que la mayoría de los usuarios no necesita conocer.

Los modelos de datos son el instrumento principal para ofrecer la abstracción. Un modelo de datos es un conjunto de conceptos que sirve para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas (lecturas) y actualizaciones de datos.

Los modelos de datos se pueden clasificar dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos.

- Los modelos de datos de alto nivel, o **modelos conceptuales**, disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios percibe los datos.

Los modelos conceptuales utilizan conceptos como entidades, atributos y relaciones los cuales representan el esquema conceptual.

- Una entidad representa un objeto con existencia física (una cierta persona, una casa, un empleado, un coche,..) o concepto del mundo real (una empresa, un puesto de trabajo, un curso universitario,...).
 - Un atributo representa alguna propiedad específica de interés de una entidad que la describe.
 - Una relación describe una correspondencia, asociación o conexión entre dos o más entidades.^[2]
- Los modelos de bajo nivel, o **modelos físicos**, proporcionan conceptos que describe los detalles de cómo se almacenan los datos en la computadora: el formato de los registros, la estructura de los archivos (desordenados, ordenados, etc.) y los métodos de acceso utilizados (índices, etc.). Los conceptos de los modelos físicos están dirigidos al personal informático, no a los usuarios finales.

- Entre estos dos extremos se encuentran los **modelos lógicos**, cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente, los modelos lógicos ocultan algunos detalles de cómo se almacenan los datos. Cada sistema de gestión de bases de datos soporta un modelo lógico, siendo los más comunes el relacional, el de red y el jerárquico. Estos modelos representan los datos valiéndose de estructuras de registros, porque también se denominan modelos orientados a registros. Hay una nueva familia de modelos lógicos, son los modelos orientados a objetos que están más próximos a los modelos conceptuales.^[2]

A la descripción de una base de datos mediante un modelo de datos se le denomina *esquema de la base de datos*. Este esquema se especifica durante el diseño, y no es de esperar que se modifique a menudo. Sin embargo, los datos que se almacenan en la base de datos pueden cambiar con mucha frecuencia: se insertan datos, se actualizan, etc. Los datos que la base de datos contiene en un determinado momento se denominan *estado de la base de datos* u *ocurrencia de la base de datos*.

La distinción entre el esquema y el estado de la base de datos es muy importante. Cuando definimos una nueva base de datos, sólo especificamos su esquema al sistema de gestión de bases de datos. En ese momento, el estado de la base de datos es el "*estado vacío*", sin datos. Cuando se cargan datos por primera vez, la base de datos pasa al "*estado inicial*". De ahí en adelante, siempre que se realice una operación de actualización de la base de datos, se tendrá un nuevo estado. El sistema de gestión de bases de datos se encarga, en parte, de garantizar que todos los estados de la base de datos serán estados válidos y que satisfaga la estructura y las restricciones especificadas en el esquema. Por lo tanto, es muy importante que el esquema que se especifique al sistema de gestión de base de datos sea correcto y se debe tener muchísimo cuidado al diseñarlo.

2.2. Niveles de abstracción

Hay tres características importantes inherentes a los sistemas de bases de datos: la separación entre los programas de aplicación, los datos y el manejo de múltiples vistas por parte de los usuarios.

El objetivo de la arquitectura de tres niveles es el de separar los programas de aplicación de la base de datos física. En esta arquitectura tal como se muestra en la figura 2.1, el esquema de una base de datos se define en tres niveles de abstracción distintos:

1. En el **nivel interno** se describe la estructura física de la base de datos mediante un esquema interno. Éste esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
2. En el **nivel conceptual** se describe la estructura de toda la base de datos para una comunidad de usuarios mediante un esquema conceptual. Este esquema oculta los detalles de la estructura de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.
3. En el **nivel externo** se describe en varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y ocultando a ése grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

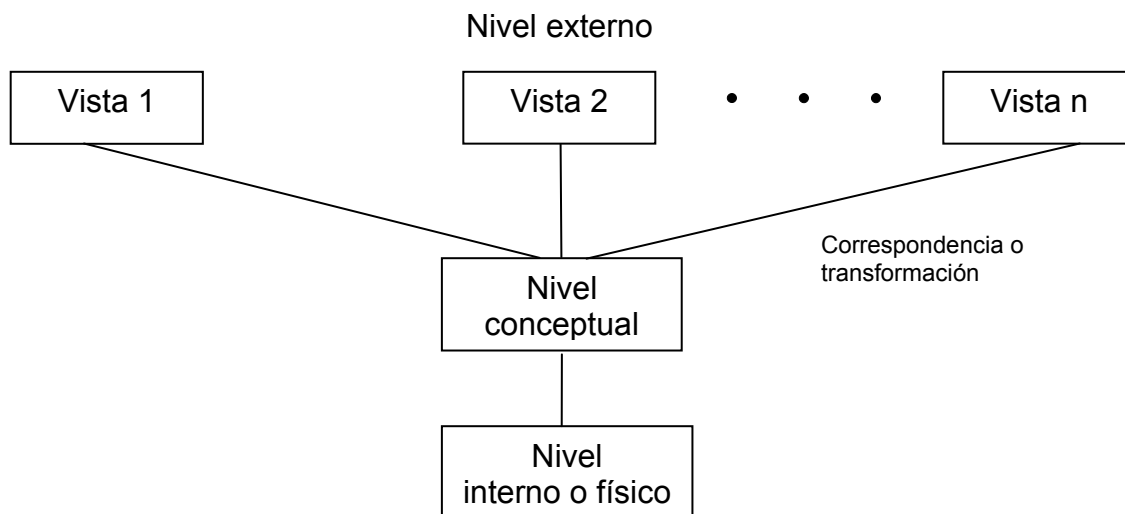


Figura 2.1 Interrelación entre los niveles de abstracción

Hay que destacar que los tres esquemas no son más que descripciones de los mismos datos pero con distintos niveles de abstracción. Los únicos datos que existen realmente están a nivel físico, almacenados en un dispositivo como puede

ser un disco. En un sistema de gestión de base de datos basado en arquitectura de tres niveles, cada grupo de usuarios hace referencia exclusivamente a su propio esquema externo. Por lo tanto, el sistema de gestión de base de datos debe transformar cualquier petición expresada en términos de un esquema externo a una petición expresada en términos del esquema conceptual, y luego, a una petición en el esquema interno, que se procesará sobre la base de datos almacenada. Si la petición es de una obtención (consulta) de datos, será preciso modificar el formato de la información extraída de la base de datos almacenada, para que coincida con la vista externa del usuario. El proceso de transformar peticiones y resultados de un nivel a otro se denomina correspondencia o transformación.

Teniendo en cuenta que las vistas forman parte de los dos tipos de tablas; las *tablas base* y las *vistas*:

- Una tabla base es una tabla "real"; es decir una tabla con existencia física, en el sentido de que existen registros almacenados físicamente, y quizá también índices físicos, en uno o más archivos los cuales representan en forma directa esa tabla almacenada. ^[3]
- Una vista es una tabla "virtual" es decir, una tabla que no está almacenada físicamente aunque desde el punto de vista del usuario parece como si lo estuviera. Las vistas pueden concebirse como formas directas de ver las tablas "reales". ^[3] Las vistas son dinámicas porque los cambios que se realizan sobre las tablas base que afectan a una vista se reflejan inmediatamente sobre ella. Cuando un usuario realice un cambio sobre la vista (no todo tipo de cambios están permitidos), este cambio se realiza sobre la relaciones de las que se deriva.

Las vistas son útiles por varias razones:

- Proporcionan un poderoso mecanismo de seguridad, ocultando partes de la base de datos a ciertos usuarios. El usuario no sabrá que existen aquellos atributos que se han omitido al definir una vista.
 - Permite que los usuarios accedan a los datos en el formato que ellos desean o necesitan, de modo que los mismos datos pueden ser vistos en formatos distintos por distintos usuarios.
 - Se pueden simplificar operaciones sobre las relaciones base que son complejas. Por ejemplo, se puede definir una vista como la concatenación de dos relaciones. El usuario puede hacer restricciones y
-

- proyecciones sobre la vista, el sistema de gestión de bases de datos traducirá en las operaciones equivalentes sobre la concatenación.

Dentro de estas características también está el uso de un catálogo para almacenar el esquema de la base de datos, en el cual se almacenan las descripciones de los datos y no es accesible para todos los usuarios. Este catálogo es lo que se denominan *diccionario de datos* y contiene información que describe los datos de la base de datos (metadatos) ^[2]. Normalmente, un diccionario de datos almacena:

- Nombre, tipo y tamaño de los datos.
- Nombre de las relaciones entre los datos.
- Restricciones de integridad sobre los datos.
- Nombre de los usuarios autorizados a acceder a la base de datos.
- Esquemas externos, conceptuales e interno y correspondencia entre los esquemas.
- Estadística de utilización, tales como la frecuencia de transacciones y el número de accesos realizados a los objetos de la base de datos.

2.3. ¿Porque utilizar PHP y MySQL?

Cuando se comienza el diseño de algún sitio, existen muchos productos que pueden ser utilizados.

Se necesita elegir el hardware para el servidor web, el sistema operativo, el software del servidor web, un sistema de gestión de bases de datos, un lenguaje con el cual comunicarnos con el servidor web y el manejador de bases de datos.

Algunas de estas elecciones van a depender de varios factores. Por ejemplo, no todos los sistemas operativos corren en cualquier hardware, no todos los lenguajes que utilizan script's pueden conectarse a todos los sistemas de gestión de bases de datos.

En este trabajo no se toma gran importancia al hardware, el sistema operativo y el servidor web, pero si en el software de gestión de bases de datos y el script (MySQL y PHP respectivamente). Ya que una de las características de

PHP y MySQL es que se pueden utilizar tanto en Microsoft Windows como en muchas versiones de UNIX y en cualquier versión de servidor web. ^[4]

Otras de las características y ventajas, es el *modelo lógico* en el que esta basado, número de usuarios, número de sitios, costo económico y el ámbito de aplicación. Los modelos lógicos empleados con mayor frecuencia en los sistemas de gestión de base de datos son: el relacional, el de red y el jerárquico. Algunos más modernos se basan en los modelos orientados a objetos. ^[3]

De acuerdo a lo anterior, en la figura 2.2 se resaltan las características que posee el sistema de gestión de bases de datos en cuestión (MySQL).

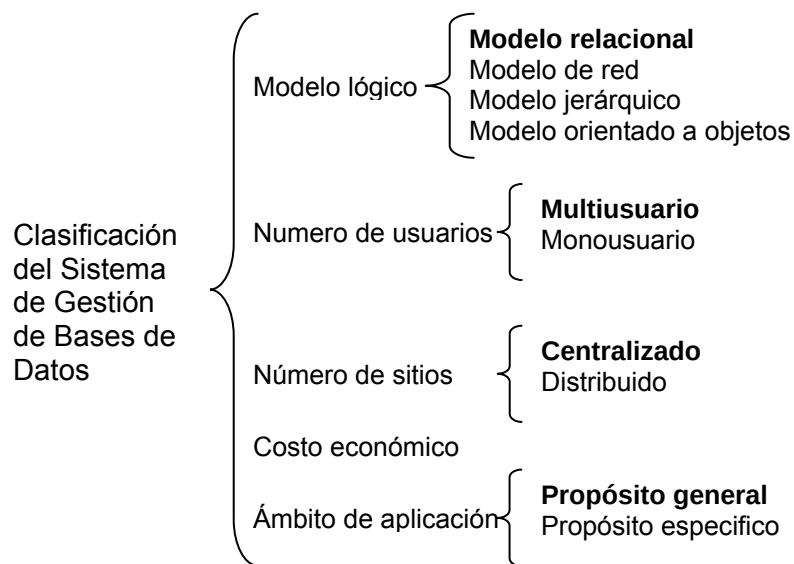


Figura 2.2 Clasificación del Sistema de Gestión de Bases de Datos. ^[2]

2.3.1. Algunas de las características de PHP

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el lado del servidor.

En vez de escribir un programa con muchos comandos para crear una salida en HTML, escribimos el código HTML con cierto código PHP embebido (introducido) en el mismo, que producirá cierta salida. El código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP. Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviéramos un script escrito con código PHP en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido.

Algunos de los principales competidores de PHP son: Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP), y Allaire Cold Fusion.

En comparación con estos productos, PHP tiene muchas fortalezas:

- Rápida respuesta.
- Interfases para muchos sistemas de gestión de bases de datos.
- Bajo costo.
- Fácil de aprender y utilizar.
- Poder desarrollar librerías propias para resolver muchas tareas.
- Portabilidad.
- Es un producto Open Source (OSS).

Dentro del nivel más básico podemos procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de sistemas de gestión de bases de datos. Escribir una interfaz

vía web para una base de datos es una tarea simple con PHP. Los siguientes sistemas de gestión de bases de datos están soportados actualmente:

Adabas D	Ingres	Oracle (OCI7 y OCI8)
dBase	InterBase	PostgreSQL
Empress	FrontBase	Solid
FilePro	mSQL	Sybase
IBM DB2	MySQL	Veloces
Informix	ODBC	Unix dbm

Figura 2.3 Sistemas de gestión de bases de datos soportados.

Características por las que se utiliza PHP: ^[5]

- Respuesta: PHP es muy eficiente. Utilizando un servidor barato, puede servir millones de solicitudes por día.
- Costo: PHP es libre. Se puede descargar la versión más reciente en cualquier momento de <http://www.php.net> sin costo.
- La sintaxis de PHP está basada en otros lenguajes de programación, principalmente C y Perl. Por lo tanto si se tienen conocimientos de C o Perl, C++ o Java, es fácil de aprender este.
- Portabilidad: PHP está disponible para muchos sistemas operativos. Se puede escribir código PHP sobre sistemas operativos libres linux, versiones comerciales de Unix tales como Solaris e IRIX, y sobre diferentes versiones de Microsoft Windows.
- Código fuente: se tienen acceso al código fuente de PHP. Al contrario de los programas comerciales, productos de código cerrado, si se requiere agregar algo se tiene la libertad de hacerlo. No es necesario esperar a que salgan los parches o una nueva versión como sucede con el software comercial.

2.3.2. Algunas de las características de MySQL

Los principales competidores de MySQL son PostgreSQL Serever y Oracle.

MySQL tiene muchas fortalezas, incluyendo rápida respuesta, bajo costo, fácil de configurar, portable y el código está disponible.

Características principales:

- Respuesta: MySQL es indisputablemente rápido.
- Costo: MySQL está disponible sin ningún costo, bajo la licencia de Open Source. www.mysql.com
- Portabilidad: MySQL puede utilizarse en muchos sistemas UNIX, así como también en Microsoft Windows.
- Código fuente: así como PHP, se puede obtener y modificar el código fuente para MySQL.

Tanto PHP y MySQL cumplen con la mayoría de los puntos que se resaltan en la figura 2.2, una característica extra es la de la portabilidad.

2.4. Arquitectura de bases de datos en la Web

La operación básica de un servidor web es mostrada en la figura 2.4

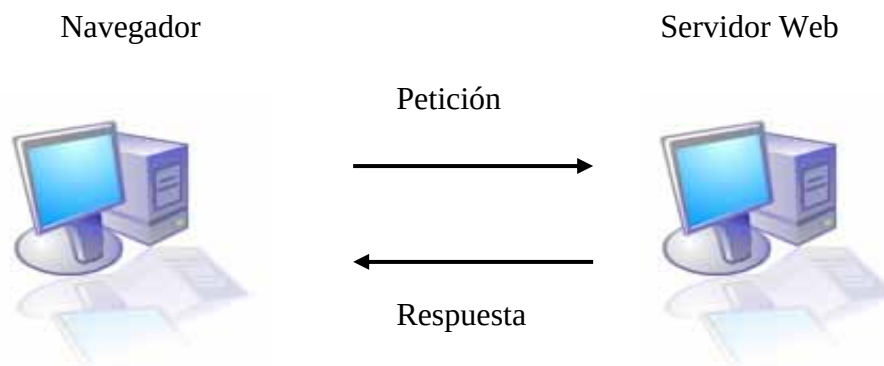


Figura 2.4 Operación básica de un servidor web.

Este sistema consiste en dos objetos: un navegador web y un servidor web. También es requerido un vínculo de comunicación entre ellos. El navegador web hace una petición al servidor. El servidor regresa una respuesta. Esta arquitectura describe páginas estáticas. La arquitectura que describe un sitio web con interfaces para el manejo de bases de datos es más compleja, figura 2.5.

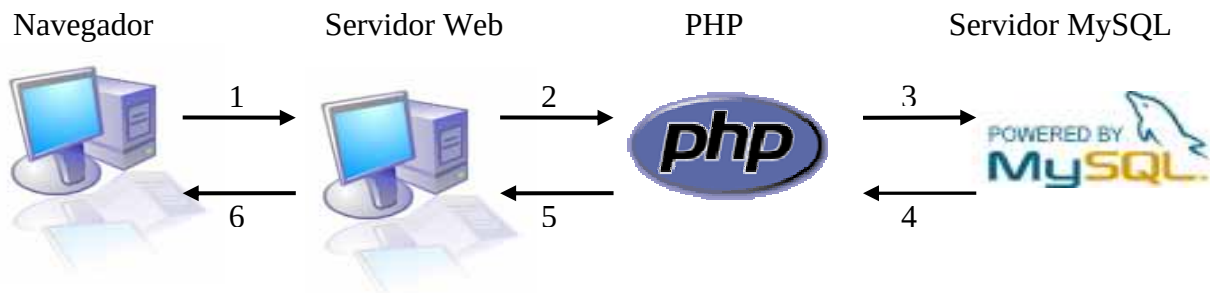


Figura 2.5 Arquitectura para el manejo de bases de datos.

La correspondencia consiste en las siguientes fases:

1. Utilizando el navegador web el usuario solicita una página web en particular, dicha pagina esta en código HTML teniendo incrustado el código php pero sin que el usuario pueda ver dicho código. Por ejemplo, requiere hacer una consulta a una base de datos.
2. El servidor web recibe la petición recuperando el archivo de extensión *.php, una vez recuperado lo pasa a PHP para que sea procesado por este.
3. PHP comienza a ejecutar el script. Dentro del script existe un comando para conectar a la base de datos y ejecutar la consulta (query). PHP abre la conexión con el servidor MySQL (sistema de gestión de bases de datos utilizada) y manda la consulta apropiada.
4. El servidor MySQL recibe la consulta a la base de datos y la procesa, una vez hecho esto manda los resultados de regresó a PHP.

5. PHP termina de correr el script, enviando el resultado envuelto en código HTML al servidor web.
6. El servidor web manda los resultados en código HTML al navegador, donde el usuario puede ver el resultado de su consulta.

2.4.1. PHPLIB

En la fase tres durante la transacción mostrada en la figura 2.4 se hace mención de la conexión con MySQL y la base de datos, dichas tareas son resueltas con PHPLIB, ahorrando un trabajo significativo, ya que la conexión con la base de datos, autenticación de usuarios y por lo tanto administración de sesiones, se realizan con bastante frecuencia, de igual forma se hace uso de tablas para mostrar las vistas a cada uno de los usuarios en la fase seis de la transacción.

Las librerías de PHPLIB están orientadas a objetos, por lo que su contenido es básicamente un conjunto de clases (y algunas funciones separadas). Las únicas funciones sueltas que hay son las que utilizo para manejar el inicio y cierre de sesiones (permisos, usuarios, etc.). Lo que hay que hacer para usar las librerías de PHPLIB es "subclasear" las clases principales para que se ajusten a nuestras necesidades y luego agregarla a cada script que las necesite con un "require". Para simplificar esta tarea, se incluye un archivo (PREPEND.PHP3) que tiene todos los "require" necesarios y algunas otras cosas (como la ubicación de las librerías de PHPLIB) para que solo se tenga que poner un solo "require" (include) en los scripts. De esta manera puedo incluir de manera automática "autoprepended" diferentes archivos a diferentes scripts (ubicados en diferentes subdirectorios).

Se podría decir que PHPLIB constan de lo siguiente^[8]:

- 1. Funcionalidad básica " Core Funtionality "**
 - 2. Funcionalidad extendida " Extended Functionality "**
 - 3. HTML Widgets**
-

1. Funcionalidad básica

Las siguientes clases son parte de las características principales de PHPLIB:

- **DB_Sql**: Esta clase es una abstracción de las bases de datos. Básicamente realiza todas las tareas de acceso a la base de datos que haríamos con las funciones de PHP, con la diferencia que lo hace de forma abstracta. Esto nos permite cambiar de bases de datos sin rescribir el código (excepto algunos "query" que funcionen en una y en otra no). Esta es una clase **independiente**.

```
class DB_SEPLADE extends DB_Sql {  
    var $Host = "localhost";  
    var $Database = "bd";  
    var $User = "nombredeusuario";  
    var $Password = "password";  
}
```

- **Session**: Se encarga del manejo de sesiones. El manejo de sesiones lo hace a través de cookies (aunque si el navegador no lo soporta lo puede hacer a través del metodo GET, pasando el id de sesión a la pagina siguiente). Básicamente tiene métodos para registrar y "desregistrar" variables, verificar si una variable esta registrada, etc. También tiene una función de auto_init que permite ejecutar un script cuando se crea una nueva sesión. (Necesita de la clase **DB_Sql**)
 - **Auth**: Se encarga de la "autenticación" de las sesiones. Esto se hace a través de un usuario/password y puede ponerse un tiempo de expiración. Hay varios modelos de implementación entre los que se encuentra uno que envía toda la información (usuario/password) a través de la red en forma cifrada (ENCRIPTADA) (usando JavaScript). Este último es el que se utiliza en este sistema y se menciona mas adelante. Esta clase también guarda un array con la información del usuario actual (uid, nombre, expiración, permisos) entre otras cosas.
 - **Perm**: Se encarga del manejo de permisos. Se puede proteger una pagina entera (llamando un método al inicio de la pagina con el/los permiso/s necesario/s como argumento/s. De no tenerlos se muestra una pantalla de login) o solamente parte de ella (a través del método
-

que devuelve *true* si se tienen los permisos necesarios). (Necesita de la subclase **Auth**).

- **User**: Es una subclase de Session. Lo que hace es mantener los datos de la sesión para un usuario en particular. De esta manera se puede personalizar una página y al volver al día siguiente, se pueden recuperar las variables de sesión para mantener la personalización de ese usuario. (Necesita **Auth**, subclase de Session).
- Funciones **page_open** y **page_close**: Estas funciones se encargan de crear los objetos necesarios para que podamos usar las características de PHPLIB sin problemas. **page_open** se llama al principio del script (antes que cualquier otra cosa porque pone información en el “Header” de HTML) y recupera los objetos \$sess, \$auth, \$perm y \$user de las bases de datos, mientras que **page_close** se llama cuando no se modifiquen mas variables de sesión (es decir, al final) y se encarga de guardar los datos de los objetos \$sess, \$auth, \$perm y \$user para poder recuperarlos con la próxima llamada a page_open (es decir, en la próxima pagina).

2. Funcionalidad extendida

Este es un conjunto de clases adicionales para complementar la funcionalidad de PHPLib, solo se compone de dos clases Cart y Template.

3. HTML Widgets

Por último están los HTML widgets, básicamente son formas de dibujar Tablas y Formularios de una manera mas cómoda y automatizada. Por ejemplo, a la Tabla, se le puede pasar como argumento un objeto DB_Sql y esta se encarga de hacer automáticamente una tabla con los resultados del query.

Las clases incluidas son:

Table: Crea una tabla HTML desde un array bidimensional o desde el resultado de un query (de un objeto DB_Sql). Se puede elegir que columnas mostrar y se puede mostrar una cabecera con el nombre de cada columna. Soporta StyleSheets para poder darle la apariencia deseada a la tabla (especificando que class usar). También se puede agregar un checkbox a

cada fila para poder realizar operaciones con ellas a través de un formulario. (Clase **independiente**).

CSV_Table: Muestra un desplegado de un array bidimensional o un resultado de un query en formato CSV. (Necesita **Table**, subclase de Table)

Sql_Query: Es una interfaz para permitir a un usuario hacer un query de una forma simple a través de un formulario. Con esos datos nos genera automáticamente una cláusula "WHERE" para incluir en el query. (Necesita **Session**).

2.5. Cifrado con MD5 en PHP

Ya que la información se guarda en la base de datos se tiene que tener algún tipo de protección. Algunos campos se guardan cifrados (encriptados), principalmente cuando una página requiere el nombre de usuario y password, esta última es cifrada y se guarda en la Base de datos.

En PHP se utiliza la función MD5 (Message Digest 5), que es una función hash irreversible (de un sólo sentido), es decir, encripta el password tecleado por el usuario siendo imposible que apartir desde la cadena encriptada se vuelva a la contraseña origen. Por esto mismo no hay problema de que alguien pueda acceder al campo encriptado de la base de datos.

Como en la base de datos se guarda la contraseña encriptada, cuando un usuario quiere acceder, habrá que realizar una comparación entre el password que introduce encriptado en MD5, y lo que tenemos en la base de datos, (que es la contraseña encriptada en MD5), si coincide se le permite el acceso, si no, se rechaza.

MD5 se utiliza también para que cuando un usuario olvida su password, si quiere recuperar la contraseña se le pide que introduzca por ejemplo el correo, y se le envía un mail con una URL tal que si entra en ella genere una nueva contraseña que se le indica al usuario y se reescribe en md5 en la base de datos (borrando la anterior contraseña).

Utilizando la clase **Auth** con el modelo de implementación con el cual el password es enviado de forma plana no es 100% seguro, puesto que el password se encripta en el servidor, entonces al enviar la contraseña desde el cliente al servidor podría ser interceptada.

Para esta herramienta se combina JavaScript MD5 con el modelo de implementación que envía toda la información (usuario/password) a través de la red en forma cifrada, logrando ahora mayor seguridad, ya que JavaScript trabaja del lado del cliente y se realiza la encriptación antes de ser enviado. Lo que se compara es el password cifrado guardado en la base de datos con el password cifrado por el cliente.

El algoritmo MD5 convierte el mensaje en un bloque múltiplo de 512 bits, (si hace falta añadirá bits por el final). Luego coge el primer bloque de 512 bits del mensaje y realiza diversas operaciones lógicas con los 128 bits de cuatro vectores iniciales ABCD de 32 bits cada uno. (Dichos vectores tendrán el valor inicial que nosotros queramos).

Como resultado obtiene una salida de 128 bits que se convierte en el nuevo conjunto de los 4 vectores ABCD. Se repite el algoritmo hasta procesar el último bloque del mensaje. Al terminar, el algoritmo devuelve los últimos 128 bits de estas operaciones.

Un "hash" es un valor (numérico normalmente, aunque podría ser una cadena de texto pequeña, etc) que se genera a partir de la "palabra" (o cadena de texto o lo que sea) que queremos asegurar.

2.6. Resumen

Se mostró como los modelos de datos son útiles para ocultar los detalles del diseño de la herramienta al usuario final, dejando a éste que solo se preocupe por los datos que tiene que proporcionar en cada una de las pantallas o formularios.

Lo anterior se logra gracias a que existe una separación entre los programas de aplicación y los datos, permitiendo el manejo de múltiples vistas por parte de los usuarios, sin que sepan código alguno del sistema de gestión de bases de datos y del lenguaje script.

Ya que se necesita elegir el hardware para el servidor, el sistema operativo, el servidor web, un sistema de gestión de bases de datos y un lenguaje que ejecute los script, se mostraron las principales características por las cuales se decidió utilizar MySQL y PHP, como gestor de la base de datos y lenguaje Script respectivamente.

Tanto MySQL como PHP son los encargados del proceso de transformar peticiones y resultados de un nivel a otro que se denomina correspondencia o transformación, haciendo uso de vistas las cuales son tablas "virtuales".

Podría decirse que los usuarios finales son engañados ya que los únicos datos que existen realmente están a nivel físico, almacenados en un dispositivo como puede ser un disco, él ve solo la parte de los datos que se le permite de una forma sencilla de tal forma que los pueda entender. Estas interfases para el manejo de bases de datos son complejas, tareas como ésta y la autenticación de usuarios son resueltas con PHPLIB, ya que estos no son problemas recientes, ya ha habido alguien mas buscando una solución, algunos han logrado buenos resultados, por lo tanto se opto por utilizar PHPLIB ya que resuelve estos problemas con excelentes resultados.

Capítulo 3 Diseño de la base de datos

El diseño de una base de datos relacional puede seguir dos caminos; por una parte, puede crearse tomando como punto de partida la observación del universo en estudio, dando lugar a un conjunto de esquemas de relaciones, que contengan los atributos y sus restricciones. Por otra parte, puede dividirse en dos fases, la primera de las cuales sería definir el modelo conceptual y su esquema y la segunda transformar el esquema conceptual en un esquema relacional.

3.1. Esquema del nivel externo.

Para poder explicar de manera global la herramienta de captura de datos y por lo tanto el esquema de la base de datos que se va a utilizar, es necesario tener presente las reglas o requisitos que la herramienta debe de satisfacer en el nivel externo (interfaz), en el Capítulo 4 se describirán las vistas finales de dicho nivel.

La herramienta como se mencionó en el Capítulo 1 va a ser un sistema que permita almacenar información del sector público, privado y social, con el objetivo de utilizarla para su análisis estadístico, los resultados van a ser publicados en diferentes medios, como es mostrado en la figura 1.1.

Básicamente almacenar la información o los datos en una base de datos es el principal objetivo a perseguir, sin embargo esto implica algo mas que solo almacenar por almacenar los datos, ya se mencionó en el Capítulo 1 que dicha información no solo es proporcionada por una fuente sino que son muchas las fuentes, ver la tabla 1.2, por lo tanto, implica que estos datos deben de estar organizados e identificados para saber de que fuente fueron proporcionados; ahora, cada una de estas fuentes no solo proporciona datos de un solo tema, mejor dicho aun, de un solo apartado, ver tabla 1.1.

Otras de las características de la herramienta de captura de datos es que facilitará al administrador la elaboración de los formatos o cuadros, ver fig. 1.2, los cuales se elaboran desde el navegador sin conocer comando alguno de PHP o MySQL conforme a la abstracción de datos mencionada en el capítulo 2.

Tanto apartados como cuadros y fuentes son descritos a continuación:

Fuentes: Son quienes ingresarán los datos a cada uno de los cuadros que les sean asignados por parte del administrador. Cada una de las fuentes puede tener uno o más usuarios sin distinguir niveles de usuario entre ellos. Tabla 1.2

Apartados: Se refieren a los diferentes temas económicos y sociodemográficos, pero teniendo como circunscripción una determinada entidad federativa o municipio. Tabla 1.1

Cuadros: Serán generados dentro de cada uno de los apartados para subdividir o desagregar los conceptos en cada apartado, para cada una de las fuentes.

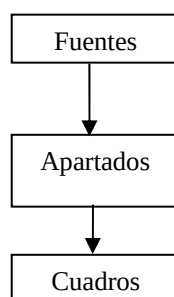


Figura 3.1 Formatos

Los conceptos antes mencionados se muestran en la figura 3.1, la cual está basada principalmente en lo expuesto en el Capítulo 1 específicamente en la figura 1.2 Distribución de formatos, en la cual se muestra que las fuentes son las que llenarán los cuadros y dichas fuentes son enlistadas en la tabla 1.2, los cuadros pertenecen a los diferentes apartados los cuales son mostrados en la tabla 1.1, cada cuadro pertenecerá única y exclusivamente a un apartado.

Básicamente se puede observar que para esta herramienta de captura de datos solo son necesarios dos grupos de usuarios, administradores y los usuarios pertenecientes a las fuentes.

Clasificación de usuarios:

Son quienes utilizan la herramienta y se clasifican en los siguientes grupos:

Administrador: Es el encargado de generar los apartados, cuadros, agregar fuentes y usuarios para cada uno de ellas, modificar los cuadros y hacer consultas a los cuadros.

Usuarios: Son asignados uno o mas para cada una de las fuentes, tanto su nombre como el password son asignados por el administrador, posteriormente el usuario puede cambiar su password. Un usuario solo puede pertenecer a una sola fuente.

3.2. Definiendo el nivel o esquema conceptual.

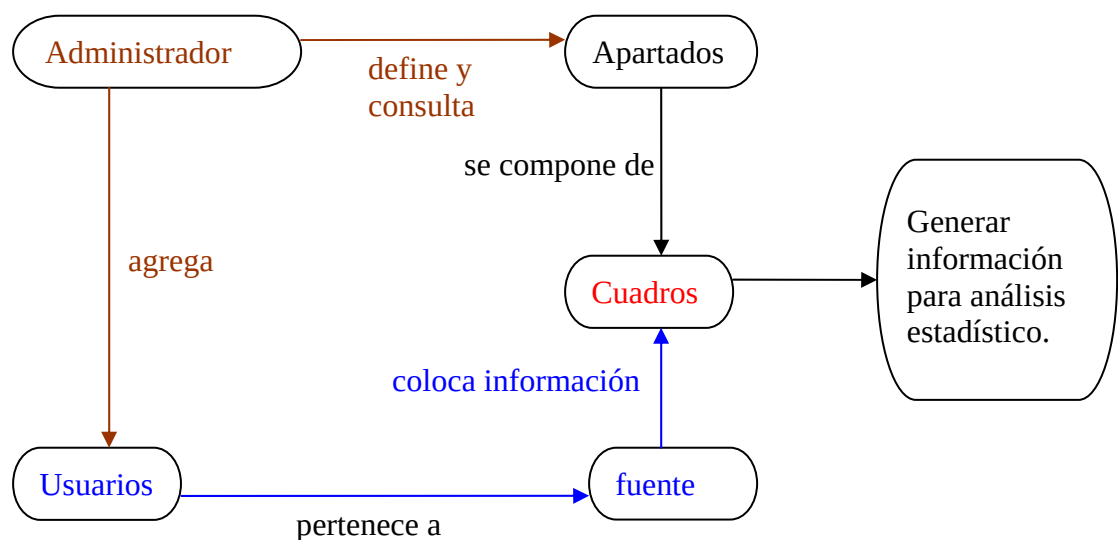


Figura 3.2 Mapa Conceptual

El resultado al que se llega del análisis del capítulo 1, en el cual se menciona la problemática actual para la adquisición de los datos, los requisitos mínimos son mostrados en el mapa conceptual de la figura 3.2 logrando así describir el contenido de la información de la base de datos y las estructuras de

almacenamiento que se necesitará para manejar esta información. Aunado a esto en la figura 3.2 se muestra además:

- La perspectiva que cada usuario tiene de los datos.
- La naturaleza de los datos, independientemente de su representación física.
- El uso de los datos a través de las áreas de aplicación.

Dentro del nivel conceptual se distinguen dos partes fundamentales en la descripción de una base de datos:

- la descripción de datos la cual se logra por medio del mapa conceptual y de forma descriptiva figura 3.2.
- manipulación de datos. Esta parte es manejada en el capítulo 4 donde se realizan las transacciones correspondientes entre los niveles de abstracción.

A su vez, cada una se organiza en torno a tres niveles: externo, conceptual e interno.

3.3. Esquema lógico

Como resultado del estudio del nivel conceptual se obtiene el siguiente esquema lógico donde se describe la estructura de la base de datos.

Para traducir el esquema lógico en un esquema que se pueda implementar se consideraran las siguientes convenciones con el fin de identificar los elementos descritos en el resto del trabajo:

- Nombre de tabla: fuente en negritas mas cursiva (***principal***).
 - Campos: fuente con cursiva (*descripción*).
 - Las tablas de las cuales su nombre se originen de los datos introducidos por los usuarios estarán compuestas por dos partes, una parte ya definida en negritas mas cursiva y la otra parte entre comillas la cual será reemplazada al momento de crear la tabla en la base de datos (***meta***"tabla" → ***metacs1***). Donde ***meta***+tabla se refiere a una tabla contenedor de los datos de las tablas que se creen a partir de esta, es decir, esta tendrá los datos sobre los datos que contendrán las que se originen a partir de esta.
-

Prácticamente la base de datos no comenzará en el estado vacío ^[8], tendrá definida parte de su estructura, la cual se define al momento de crear la base de datos, pero el numero de tablas se irá incrementando conforme se agreguen apartados y cuadros, por lo tanto a nivel lógico estará cambiando constantemente, pero el código de PHP no tendrá que ser modificado, es decir, el nivel externo no se vera afectado facilitando de esta forma el trabajo del administrador. Las tablas que definen la estructura inicial de la base de datos serán:

principal (vacía)

fuentes (vacía)

fuentespara (vacía)

auth_user_md5 (cuando menos tendrá los datos del administrador).

Como ya mencione anteriormente no basta con solo agregar los datos a una base de datos, si no que es necesario conocer quien los proporciona y de esta forma facilitar su consulta, este problema lo resuelvo creando y/o modificando las tablas enlistadas anteriormente. Todas las tablas son consultadas por cada uno de los usuarios pero gracias a los niveles de abstracción solo será mostrada la parte que le interesa a cada uno de los usuarios.

La tabla que almacenará los usuarios será la única que al principio haga que la base de datos cambie del estado vacío al estado inicial, ***auth_user_md5***, las demás tablas estarán vacías incluyendo la tabla que se genera para almacenar el nombre de los apartados ***principal***.

3.3.1. Tabla ***principal***

En esta tabla solo se guardan los nombres de los apartados y su identificador. Esta es la primera tabla que se muestra a todos los usuarios al momento de ingresar a la herramienta pero con una vista distinta para cada uno. La tabla contiene los siguientes campos:

principal

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI	NULL	auto_increment
apartado	varchar(100)	YES		NULL	

Figura 3.3 Nombre de apartados

3.3.2. Tabla **a_ "id"** del apartado

Al momento de agregar un apartado en la tabla **principal** se genera otra tabla con un nombre que se genera en función del identificador del apartado.

a_ "id"

Field	Type	Null	Key	Default	Extra
clave_c	int(11)		PRI	NULL	auto_increment
nombre_c	varchar(5)				
descripcion	varchar(200)				

Figura 3.4 Relación de cuadros con apartados

nombre_c: se guardarán los nombres de todos los cuadros que correspondan al apartado en cuestión.

descripcion: se almacenará el encabezado de cada uno de los cuadros. Estos pueden ser modificados solo por el administrador.

Esta tabla es utilizada para almacenar la relación de los cuadros con los apartados.

3.3.3. Tabla **fuentes**

Para el caso de las fuentes (orígenes de datos) tenemos la siguiente tabla:

fuentes

Field	Type	Null	Key	Default	Extra
clave	int(11)		PRI	NULL	auto_increment
fuelle	varchar(200)				

Figura 3.5 Origen de los datos

Al igual que la tabla de los apartados tenemos solo el nombre del apartado y su identificador.

3.3.4. Cuadros

Los cuadros de acuerdo a la figura 3.2 son la parte medular de este trabajo, esto se debe a que se requiere que:

1. cada uno de los cuadros generados debe de tener asignada una fuente cuando menos.
2. cuando se le asigne más de una fuente se debe de generar un cuadro para cada fuente. Esto con el fin de tener por separado los datos de cada una de las fuentes.
3. poder agregar nuevos conceptos después de haber sido generados, aun cuando hayan pasado al estado inicial y que además debe de estar en un estado valido, es decir, cuando ya contienen datos, esto es aplicable para los dos casos anteriores, pero para el segundo caso es de notar que no es necesario modificar todos los cuadros manualmente ya que se modifica un **meta**"cuadro" y los demás se actualizarán automáticamente. Esto forma parte de la independencia de datos.
4. al igual que el punto anterior es importante que se puedan agregar columnas en cada uno de los cuadros sin la necesidad de modificar el código sin tener que preocuparse por la cantidad de columnas ya que, como es sabido, la mayoría de los manejadores de bases de datos y hojas de calculo tiene limites muy inferiores comparado con el de las filas, esta herramienta podrá tener tantas columnas como filas pueda manejar MySQL (5,000 millones de rows o filas). Y como no se tiene un dato exacto de cuantas columnas serán necesarias, con este enfoque se elimina la preocupación del limite de columnas.

Para lograr que el punto 2 sea satisfecho, fue necesario hacer el diseño del esquema de tal forma que al momento de insertar los cuadros de cada una de las fuentes, la transformación hacia el nivel lógico, corresponda a dos tablas por cada cuadro, una de ellas con la finalidad de tener una **meta**"tabla", es decir, una tabla base con el catalogo o diccionario de las tablas que se deriven de ella, la **meta**"tabla" no contendrá la información o los datos que serán suministrados por las fuentes, solo contendrá la descripción de las filas de todas las tablas que hagan referencia a ésta.

La segunda tabla que se genera, será muy similar a la anterior con la diferencia de que a partir de esta tabla se generarán las tablas necesarias para cada una de las fuentes, en las cuales sí se almacenará la información.

La estructura de las dos tablas es la siguiente:

***meta*"tabla"**

Field	Type	Null	Key	Default	Extra
campo	char(3)		PRI		
descripcion	varchar(200)	YES		NULL	

Figura 3.6 Cuadros

Los campos que hacen posible la relación son: *campo* de la ***meta*"tabla"** y *clave* de "tabla", por cada *descripción* que se agregue a la ***meta*"tabla"** en "tabla" se agrega una columna usando como nombre el identificador de descripción de esta con el nombre del identificador de *descripción*.

"tabla"

Field	Type	Null	Key	Default	Extra
clave	varchar(100)	YES			
c1	varchar(6)			NULL	

Figura 3.7 Tablas de las fuentes

Pero si se agrega una nueva clave en la tabla: "tabla" solo se verán afectadas las tablas derivadas de esta, no se modificara ***meta*"tabla"**, por que, como se menciono anteriormente solo contiene las descripciones.

Todo lo anterior se verá reflejado en el nivel exterior, en las vistas de cada una de las fuentes. Ver figura 3.8.

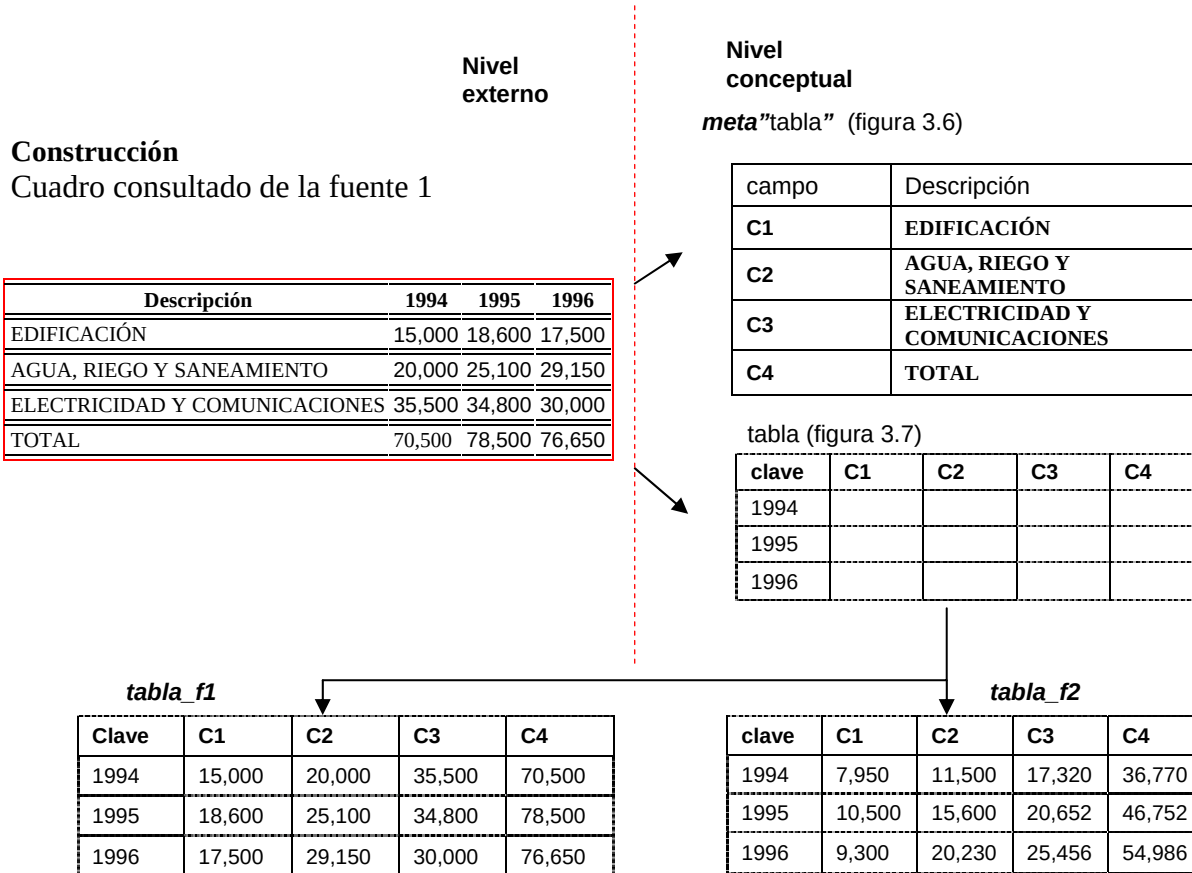


Figura 3.8 Nivel externo e interno

Para la figura 3.8 podemos observar como en el nivel externo las vistas son formadas por tablas virtuales originadas a partir de tablas almacenadas en la base de datos ya que estas no se podrían interpretarse ya que no corresponde al modelo final que utilizan los usuarios, la vista en el nivel externo se puede observar como esta compuesta por la combinación de **meta**"tabla" y tabla esta ultima corresponde a la fuente 1 **tabla_f1** según la figura 3.8.

3.3.5. Tabla *auth_user_md5*

Esta tabla base es la ideal para tener el control de cada uno de los usuarios y las fuentes a las que pertenecen.

auth_user_md5

Field	Type	Null	Key	Default	Extra
user_id	varchar(32)		PRI		
username	varchar(32)		UNI		
password	varchar(32)				
perms	varchar(255)	YES		NULL	
cfuente	int(11)			0	

Figura 3.9 Seguridad

Los campos *perms* y *cfuente*, serán de gran utilidad al momento de ingresar a la herramienta, ya que es uno de los determinantes para decidir que vista mostrar en el nivel exterior.

También los campos *username* y *password* son de gran importancia ya que con ellos se logra un mejor control y seguridad para tener acceso a la herramienta. El password se almacena en forma cifrada en la base de datos, en el lado servidor, y en el navegador o en el nivel exterior se cifra usando JavaScript con un algoritmo hash. Si es interceptado el password no se puede saber que cadena de caracteres lo produjo, de la misma forma si por algún motivo sin explicación se tiene acceso a esta tabla no verán los passwords solo verán los datos codificados.

username	password	perms	cfuente
kris	098f6bcd4621d373cade4e832627b4f6	user	1
gabriel	647431b5ca55b04fdf3c2fce31ef1915	admin	0

Figura 3.10 Almacenamiento en forma cifrada

Con la tabla **auth_user_md5** junto con la tabla **fuentespara** se logra tener el control de la herramienta, en **auth_user_md5** se sabe a que fuente pertenece cada usuario, siendo la tabla **fuentespara** la que permite hacer la relación con cada uno de los cuadros a los que tiene acceso cada uno de los usuarios.

3.3.6. Tabla **fuentespara**

Cuando se agregan los cuadros a los apartados es un buen momento para asignar las fuentes que van a llenarlos. Es así como se crea la relación apartado, cuadro, fuente; esto se ve en la siguiente tabla en la cual se guardan los identificadores de cada uno de los datos para la relación.

fuentespara

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI	NULL	auto_increment
nombre_cuadro	varchar(5)	YES		NULL	
clave_f	int(5)	YES		NULL	
id_a	int(11)			0	

Figura 3.11 Relación usuario-fuente

Una vez que se introduce el password y el nombre del usuario la herramienta en el nivel lógico hace una consulta a la base de datos para saber a que fuente pertenece dicho usuario, enseguida se buscan los apartados en los cuales se le ha asignado cuadros. Esto permite ver la relación entre las tablas **auth_user_md5** y **fuentespara**, los campos **cfuente** y **clave_f** son las claves de la relacion entre las tablas.

El procedimiento al momento de la transformación y para entender mejor la relación entre las tablas tenemos la siguiente figura 3.12.

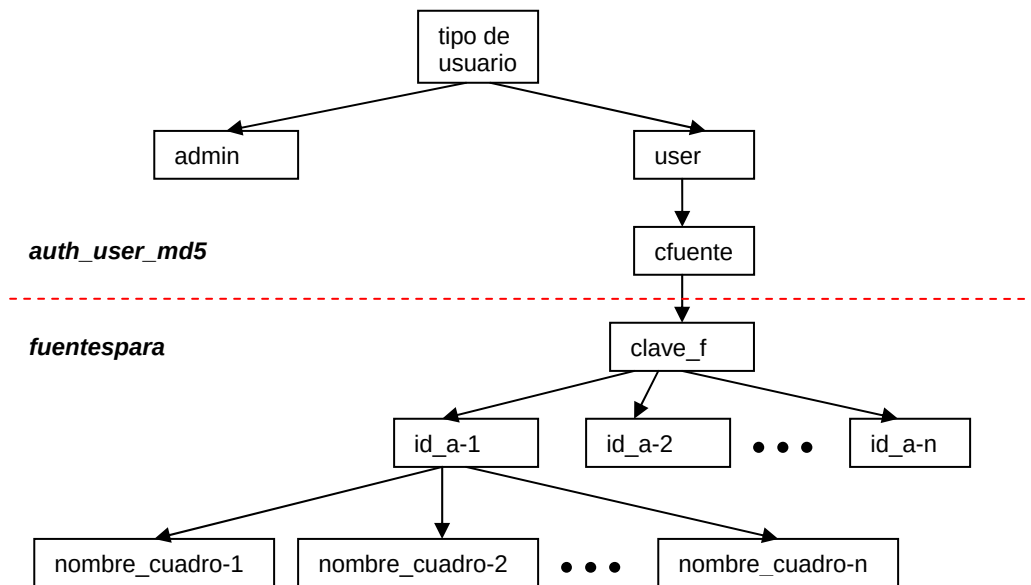


Figura 3.12 Transformaciones grupo usuarios

El procedimiento a grandes rasgos para la adquisición de los datos es el siguiente: según la figura 3.12 al momento de acceder a la herramienta de adquisición de datos es necesario identificar a que grupo de usuarios pertenece quien accede a esta, esto se logra consultado principalmente a la tabla ***auth_user_md5***, una vez identificado a que grupo pertenece para el caso de la figura 3.12 supondremos que pertenece al grupo “usuarios”, al mismo tiempo será identificado a que fuente pertenece, los apartados que le fueron asignados se identifica consultando la tabla ***fuentespara*** una vez hecho lo anterior se procederá a identificar que cuadros de cada uno de estos apartados le pertenecen a este usuario para que pueda proceder a almacenar la información.

Otra de las características del grupo de usuarios es que pueden modificar su password de la tabla ***auth_user_md5***.

Sin embargo la secuencia que se sigue al momento de acceder como administrador es otra ya que se presenta de tal forma que tenga el mayor parecido al formato en papel.

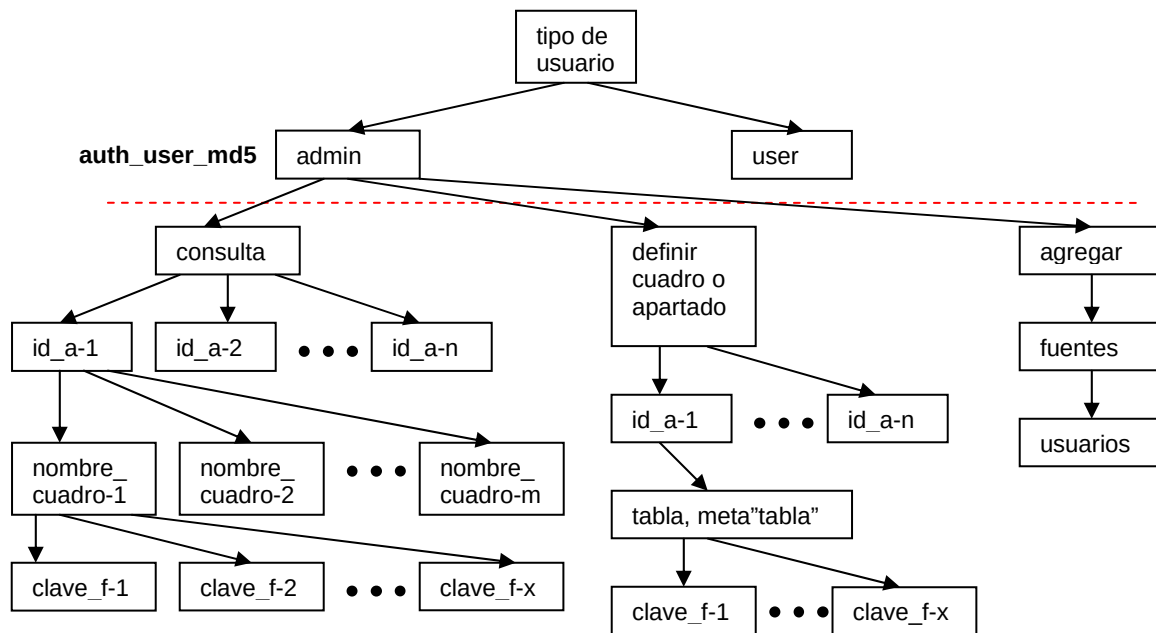


Figura 3.13 Transformaciones grupo administrador

Prácticamente en la figura 3.13 se muestra que el administrador consulta y modifica la estructura de la base de datos en su totalidad, hace una consulta al momento de buscar los datos que almacenaron las fuentes, modifica la estructura cuando agrega un apartado o un cuadro, hace cambiar de estado a la base de datos por ingresar una nueva fuente y los usuarios.

3.4. Resumen

Principalmente para desarrollar el esquema de la base de datos es necesario tener presente los requisitos mínimos que la herramienta debe de satisfacer en el nivel externo teniendo en cuenta cada una de las partes en las que se divide la herramienta.

De acuerdo a lo anterior y la descripción de la problemática descrita en el capítulo 1 se llega a la definición del esquema conceptual, originando cada una de las tablas que se presentaron.

Por lo tanto para el diseño de esta base de datos se siguió el camino donde primero definimos el modelo conceptual y su esquema, consecuentemente de transforma el esquema conceptual en un esquema relacional.

Entender todos los esquemas de los diferentes niveles es primordial para lograr que las correspondencias entre ellos sean correctas.

Se trata de comprobar que el esquema obtenido puede acomodar los futuros cambios en los requisitos con un impacto mínimo. Si el esquema lógico se puede extender fácilmente, cualquiera de los cambios previstos se podrá incorporar al mismo con un efecto mínimo sobre los usuarios existentes.

Capítulo 4. Interfaz del sistema

Una interfaz se diseña específicamente para la gente que la usará.

La idea fundamental en el concepto de interfaz es el de mediación, entre hombre y máquina. La interfaz es lo que "media", lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como la computadora. Esto implica, además, que se trata de un sistema de traducción, ya que los dos "hablan" lenguajes diferentes.

Para desarrollar la interfaz fue necesario estudiar los siguientes puntos:

- Analizar el tipo de usuario que va a usar la herramienta,
- Especificar las acciones que la interfaz debe soportar con el fin de cumplir con los objetivos del usuario,
- Especificar los tipos de datos (objetos) sobre los que se ejecutarán las acciones,
- Definir el *plan o estrategia* por el cual un usuario potencial alcanzará sus metas o requerimientos.

La principal dificultad es la asociada a comprender las tareas que el usuario desea realizar sobre la herramienta y las características de los diferentes usuarios que lo manejarán.

A diferencia del administrador, se definió un modelo de usuario genérico, representando a un grupo de usuarios de similares características. La interfaz es confiable, predecible y sencilla para el usuario, quien de otra forma no se sentiría a gusto trabajando con ella. El proceso no fue sencillo: no existe el "usuario promedio", el conocimiento del usuario no es estático y aun somos incapaces de crear modelos precisos. Pero podemos desarrollar un modelo razonablemente valedero.

Al tratar de incorporar la herramienta, es importante darse cuenta que no solo se esta cambiando un proceso, esta intentando cambiar la cultura de la organización y las creencias más apreciadas por sus miembros. Es posible lograrlo, pero requiere tener claro el objetivo al que se quiere arribar y una comunicación excelente sobre los beneficios del cambio.

Con el objeto de que alguien mas pueda mejorar las cosas se presenta en este capítulo información técnica, también para tener posibilidad de que se presenten nuevas propuestas, que mejoren el servicio.

4.1. Autenticación de usuarios

El acceso a base de datos es una de las tareas más comunes, en cualquier aplicación sobre Internet. Y una de las primeras que cualquier programador trata de automatizar. PHPLIB incorpora librerías para la gestión de sesiones, autenticación, etc. tal como se menciona en el Capítulo 2.

Básicamente una sesión es la secuencia de páginas que un usuario visita. Para esto se crea un identificador único que asignamos a cada una de estas sesiones de navegación, a este identificador de sesión se le denomina, comúnmente, como **la sesión**. En PHP tenemos la suerte de que toda la gestión de sesiones la hace el mismo PHP.

Retomando la parte de PHPLib en la cual se menciona la clase **Session** ya que mantiene una lista de propiedades (variables de una clase) y provee de un conjunto de funciones. Para reconocer los tipos de sesiones.

Estas variables se configuran para todas las clases y sesiones en el archivo local.inc o local4.inc ^[8]

La ventaja de esto es que solo tenemos que autenticar (identificar al usuario que se encuentra al lado cliente de la sesión) y restringirles las sesiones.

La autenticación que ofrece PHPLib es realizada con formularios HTML. La cual tiene varias ventajas:

- Se puede deshacer. Una sesión puede ser des-autenticada, el usuario puede hacer un "log out".
 - Puede expirar. Una sesión se podría des-autenticar automáticamente después de cierto tiempo.
 - Puede ser configurada. Ya que, se tiene un control completo de la pantalla de login, por que es una página HTML normal.
 - Es basada en base de datos. La autenticación es realizada contra una base de datos de diseño propio.
 - Es por página. Uno decide, según la página, cuales serán autenticadas y cuales no.
 - Se puede integrar con un esquema de verificación de permisos.
-

La autenticación de PHPLib hace uso de la clase Auth. Utilizando la sub-clase `Example_Challenge_Cryp_auth`.^[8]

Esta sub-clase tiene varios parámetros los utilizados pero los que nos interesan por ahora en particular son:

- nombre de archivo html que mostrará el formulario (`crcloginform.ihtml`)
- la tabla con la información de usuario (`auth_user_md5`)

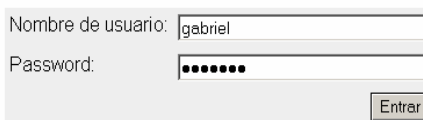
La principal función por la que es utilizada es para saber si la combinación de usuario/password existe en la tabla de usuarios de la base de datos.

Una vez configuradas correctamente nuestras clases de sesión y autenticación, se crean páginas restringidas donde sólo a usuarios validos podran ver el contenido; específicamente los que se encuentran en nuestra tabla **`auth_user_md5`**.

El archivo `crclonginform.ihtml` generara la pantalla de la figura 4.1 al momento de tratar de entrar a una página restringida.

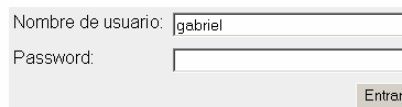
Introduce tu password

Escribe tu nombre de usuario y un password :



Introduce tu password

Escribe tu nombre de usuario y un password :



El nombre de usuario o tu password no son validos.
Intenta de nuevo!

Figura 4.1. Autenticación usuario/password

Para lograr lo anterior basta que usemos las funciones `page_open()` y `page_close()`, al inicio y final de la página respectivamente para que todo lo que se

encuentre entre estas funciones sólo pueda ser ejecutado por usuarios autenticados.

```
<?php

page_open(array("sess" => "Example_Session",
                "auth" => "Example_Challenge_Crypt_Auth",
                "perm" => "Example_Perm"));

    // código restringido a usuarios no válidos

    page_close();

?>
```

La función `page_open` recibe como parámetros las clases `sess`, `auth` y `perm` previamente configuradas, validando al usuario/password ingresado vía el formulario que se muestra en la figura.

En cada página restringida se llama a las funciones `page_open()` y `page_close()`, y cada vez que se acceda a las páginas se verificará si existe una sesión autenticada, si no es el caso, se mostrara el formulario de usuario/password de la figura.

El manejo de privilegios depende de una sesión autenticada. Se asocia un conjunto de permisos requeridos con una página. El contenido de la página es sólo visible para usuarios con TODOS los privilegios correspondientes; el resto de los usuarios verán una página personalizada con un mensaje de error.

4.2. Diferentes tipos de usuarios

Los usuarios validos, específicamente los que se encuentran en nuestra tabla **`auth_user_md5`** y que se irán agregando conforme sea requerido.

El objetivo de tener diferentes tipos de usuarios esta vinculada con la función de proteger sólo cierta funcionalidad de una página, es decir, cada usuario tiene distintos privilegios, esto se logra utilizando el método `have_perm()` de la siguiente manera.

```
<?php

page_open(array("sess" => "Example_Session", "auth" => "
Example_Challenge_Crypt_Auth ", "perm" => " Example_Perm "));

if ($perm->have_perm("admin"))
{
    # código sólo para usuarios con perfil "admin"
}

# código para cualquier usuario autenticado

page_close();

?>
```

Los principales usuarios de esta herramienta están identificados con los permisos y/o privilegios que se les asignan y están almacenados como ya se menciono en la tabla ***auth_user_md5*** de la base de datos propia, para el caso del administrador sus privilegios son de **admin**, siendo para los demás usuarios el de **user**. De esta forma se protege la información que debe de visualizar cada uno. Cabe mencionar que esto solo se hace para identificar los privilegios, pero esto no es todo en el proceso de generar las vistas para cada uno de los usuarios, ya que cada usuario pertenece a fuentes distintas y por lo tanto tienen en el nivel externo diferentes vistas.

Por tal motivo se debe de tener un control de las vistas para los usuarios con el mismo tipo de privilegio (**user**).

4.3. El administrador de la herramienta

En el Capítulo 3 se mencionaron las principales funciones del administrador, ahora son mostradas en la figura 4.2 ya que esta es la vista principal que el administrador tendrá al acceder a la herramienta. Cada una de las partes de esta vista o pantalla son descritas a continuación.

Tu autentificación es valida hasta 30 May 2005, 00:17:23

APARTADO CONSULTAR

Artesanías [CONSULTAR](#)

Construcción [CONSULTAR](#)

Educación [CONSULTAR](#)

Nuevo Apartado

[Administración de usuarios](#) Agrega o modifica los datos de los usuarios.

[Fuentes](#) Agregar las fuentes.

[Salir](#) Borra tu información de autentificación.

Figura 4.2. Vista principal del administrador

- **Apartados:** Consultas a los cuadros de cada uno de los apartados y/o modificarlos.
 - Nuevos cuadros: Dentro de cada uno de los apartados se puede agregar la cantidad de cuadros necesarios hasta el momento teniendo la oportunidad de agregar más cuando sea necesario.
 - Informes (consultas): En esta parte se pueden hacer solo consultas de los datos previamente aportados por las fuentes.
 - Adición de registros: Para cada uno de los cuadros es posible agregar más registros por medio de un formulario.
- **Nuevos apartados.** Agregar la cantidad necesaria de apartados.
- **Administración de usuarios y fuentes:** Agregar o modificar los datos de los usuarios así como asignarlos a las fuentes a las que pertenecen.

Estas vistas son generadas en la página index.php en la cual se hace la autentificación y validación de la siguiente forma:

```
<html>

<body>

<?php
.
.
.
if ($perm->have_perm("admin")) {

    //Código que genera las vistas del administrador.

}elseif($perm->have_perm("user")){

    // Código que genera las vistas para cada uno de los usuarios

}else{

    my_error("<h1>No tienes permisos para ver esta pagina</h1>");

}

?>

</body>
</html>

<?php

    page_close()

?>
```

Haciendo una comparación, si se utilizara solo código HTML se tendría que escribir el código para cada uno de los usuarios con el fin de tener su vista final, es decir una pagina HTML para el administrador y una para cada uno de los usuarios sumando a esto las paginas de las siguientes ligas de cada usuario.

Por lo tanto la ventaja de usar código PHP incrustado en el de HTML es de que nos ayuda substancialmente a reducir tanto la cantidad de código por escribir cada vez que se agregue un usuario o algún otro dato, así como la problemática de escribir una página para cada usuario. Logrando de esta forma que el administrador no tenga que conocer el detalladamente el código.

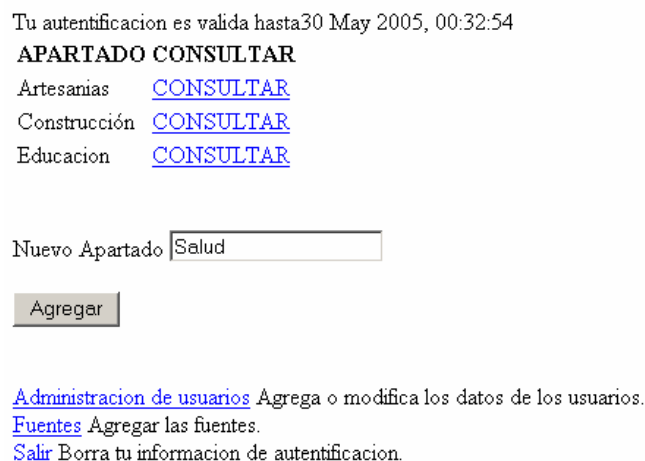
4.4. Componentes de la vista del administrador.

4.4.1. Apartados

Una vez autenticado, validado e identificado al tipo de usuario, en este caso, siendo el Administrador de la herramienta, aun estando en la pagina index.php, una de las tareas por desarrollar es la siguiente:

Como ya se menciona en el Capítulo 2 la herramienta requiere de una división bien marcada en cada uno de los conceptos que se manejan, por tal motivo es necesario comenzar de la raíz la cual se localiza en los apartados.

Se tiene la opción de agregar cada uno de los apartados, al momento de ingresar un nuevo apartado se modifican los estados de la tabla **principal** y se crea una nueva tabla.



Tu autentificacion es valida hasta 30 May 2005, 00:32:54

APARTADO CONSULTAR

Artesanias [CONSULTAR](#)

Construcción [CONSULTAR](#)

Educacion [CONSULTAR](#)

Nuevo Apartado

[Administracion de usuarios](#) Agrega o modifica los datos de los usuarios.

[Fuentes](#) Agregar las fuentes.

[Salir](#) Borra tu informacion de autentificacion.

Figura 4.3. Nuevo apartado

En la tabla principal solo es donde se guarda el nombre del apartado que es ingresado en el campo **Nuevo apartado** de la figura 4.3.

```
$q="insert into principal (apartado) values('$apartado')";  
$db->query($q);
```

Estas instrucciones son usa clases de phplib. [8].

Para el caso de la tabla, que se agrega a la base de datos consecuente de haber agregado un nuevo apartado, tiene la finalidad principal de almacenar los nombres de cada uno de los cuadros y sus descripciones al mismo tiempo que se agrega un identificador. El nombre de la tabla que se genera esta compuesto por los caracteres "a_1" en el cual el número se el de identificación propio de cada uno de los apartados. Por lo tanto existirán tantas tablas de este tipo como apartados sean agregados.

Tu autentificacion es valida hasta 30 May 2005, 00:34:20

APARTADO CONSULTAR

Artesanias [CONSULTAR](#)

Construcción [CONSULTAR](#)

Educacion [CONSULTAR](#)

Salud [CONSULTAR](#)

Nuevo Apartado

[Administracion de usuarios](#) Agrega o modifica los datos de los usuarios.

[Fuentes](#) Agregar las fuentes.

[Salir](#) Borra tu informacion de autentificacion.

Figura 4.4. Actualización

Es importante mencionar que ya no hay necesidad de editar el código de la pagina php, ya que automáticamente es actualizado el cambio que se realizó, esto se ve en la figura 4.4, es un ejemplo del funcionamiento de las páginas dinámicas.

4.4.2. Administración de usuarios y fuentes

Al momento de crear la base de de datos para la herramienta se debe de cambiar a su estado inicial, básicamente la tabla que hará cambiar de estado a la base de datos será `auth_user_md5`; podría considerarse un punto crítico para la herramienta ya que sin estos datos no seria posible iniciar sesión alguna.

La instrucción para insertar en la tabla `auth_user`, los datos tales como: `user_id`, `username`, `password`, `perms` y `cfuente` pero el password se guarda de forma plana siendo esto muy riesgoso.

table **'auth_user'**

```
INSERT INTO auth_user VALUES
('c14cbf141ab1b7cd009356f555b607dc','gabriel','gabriel','admin','0');
```

Para darle una solución a este problema se utiliza la tabla `auth_user_md5` dentro de la misma base de datos en esta tabla se guarda el password de forma cifrada utilizando el algoritmo de MD5 para lograr que si la información es interceptada no sea posible saber el password ya que dicho algoritmo es unidireccional.

table **'auth_user_md5'**

```
INSERT INTO auth_user_md5 VALUES
('da99f15745fe93e6eb7dabf40403d58a','gabriel','647431b5ca55b04fdf3c2fce31ef1915','admin','0');
```

Estos datos son cargados a la base de datos después de haber sido creada, el archivo que contiene la estructura de la base de de datos se llama `create_database.mysql`.

Administración de usuarios

Estas registrado como **gabriel** con nivel de **admin**.
 Tu autentificación es valida hasta 27 Nov 2005, 19:06:29.

Nombre de usuario	Password	Nivel	Fuentes	Accion
		user admin	Anuario Estadístico del Estado de Michoacán	Agregar usuario
gabriel	*****	user admin	SEPLADE Inventario de Obra Pública	Eliminar Modificar
samuel	*****	user admin	INEGI	Eliminar Modificar
yo	*****	user admin	Anuario Estadístico del Estado de Michoacán	Eliminar Modificar

[Salir](#) Borra tu informacion de autentificación.

Figura 4.5 Administración de usuarios.

Se ha visto hasta el momento donde se almacena el nombre de usuario y el password así como a la fuente a la que pertenece cada uno de los usuarios. Estos datos se ingresan en la vista del administrador que se muestra en la figura 4.5

esta es generada con el archivo `new_user_md5.php` que forma parte de la librería `phplib`, se modificó parte del código para lograr el objetivo ya que tal como esta en la librería no es del todo funcional. Esta vista es mostrada solo al administrador.

El nivel o permisos que se les puede otorgar a los usuarios esta relacionando en base a la configuración de los archivos `local.inc` y `auth.inc`.

Por otro lado las fuentes son agregadas a la tabla **fuentes** de la base de datos por medio de la vista que proporciona el archivo **fuentes.php** a la cual solo el administrador tiene acceso, figura 4.6.

Tu autentificacion es valida hasta 30 May 2005, 00:47:32

Advertencia: Si vas a agregar una fuente no olvides asignarle un usuario

Agregar fuente

FUENTES	borrar
Anuario Estadístico del Estado de Michoacán	borrar
Comision Forestal del Estado de Michoacán	borrar
INEGI	borrar
Procuraduria Federal de Proteccion al Ambiente	borrar
H. Ayuntamiento de Morelia, Dirección de Aseo Público	borrar
Sistema de Cuentas Nacionales de México	borrar
Registros Estadísticos del Centro	borrar
Secretaría de Comunicaciones y Transportes	borrar
SEPLADE Inventario de Obra Pública	borrar
Regresa al inicio Terminar	

[Administracion de usuarios](#) Agrega o modifica los datos de los usuarios.

Figura 4.6 Fuentes

4.4.3. Nuevos cuadros

Cada uno de los apartados tendrá cuadros con la información que será requerida a cada una de las fuentes y a sus respectivos usuarios. El proceso para agregar estos cuadros a cada uno de los apartados es muy sencillo para el administrador, ya que solo tendrá que ir llenando algunos formularios con los datos requeridos, esto se logra al momento de que el administrador sigue la liga [CONSULTAR](#) del apartado al que sea necesario agregar los cuadros, la pagina a la cual nos manda es `consultApartado.php` conjuntamente manda como parámetro

el identificador del apartado en este caso un numero, proporcionándonos una nueva vista. Observando que no es necesario que conozca los comandos de MySQL, la herramienta lo realiza de manera transparente para el administrador.

Construcción

CUADRO	CONCEPTO	consultar modificar
cs3	Valor de la producción en el sector formal de la industria de la construcción según tipo de obra 1994-2003	consultar modificar
cs4	Valor de la producción en el sector formal de la industria de la construcción según sector institucional 1994-2001	consultar modificar

Agregar cuadro

Concepto:

Licencias de construcción expedidas en la ciudad de Morelia por destino de la obra 2002

Nombre del cuadro

No. de campos

Figura 4.7 Nuevos cuadros

En la figura 4.7 se ve el formulario que se tiene que llenar para agregar los cuadros. Al momento de generarlos, se crean simultáneamente dos tablas base más N tablas, las N tablas dependen de la cantidad de fuentes que llenarán este cuadro, tal como se muestra en la figura 4.11, una de las tablas base corresponderá a una **meta** "tabla", es decir, es el diccionario de datos del resto de las tablas, la segunda tabla será copiada para crear una tabla por cada fuente que llenará este cuadro.

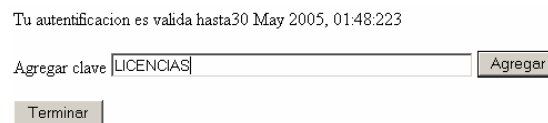
Tu autentificacion es valida hasta 30 May 2005, 01:46:14
 El nombre de la tabla es cs5
 El numero de campos es 4
 El numero de a_3
 Descripción Licencias de construcción expedidas en la ciudad de Morelia por destino de la obra 2002

Descripción	Tipo	Longitud
TOTAL	INT	9
HABITACIONAL	INT	9
COMERCIAL	INT	9
CONDOMINIO	INT	9

Figura 4.8 Formulario para la **meta** "tabla"

El formulario de la figura 4.8 es generado en la pagina definEstruC.php. los datos proporcionados formaran parte de la **meta**"tabla" los cuales seran los renglones de los cuadros para cada una de las vistas de las fuentes, solo se creará una **meta**"tabla" por cada cuadro.

Para definir la estructura de la tabla se utiliza la pagina columnasNuev.php, dicha estructura será copiada para cada una de las fuentes, la vista se muestra en las figuras 4.9 y 4.10 los siguientes formularios.

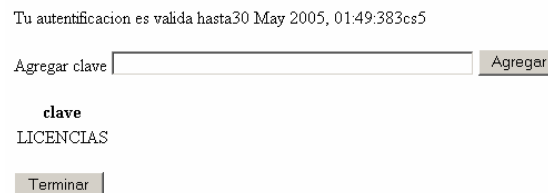


Tu autentificacion es valida hasta 30 May 2005, 01:48:223

Agregar clave

Figura 4.9 Nueva columna del cuadro

Segundo formulario



Tu autentificacion es valida hasta 30 May 2005, 01:49:383cs5

Agregar clave

clave
LICENCIAS

Figura 4.10 Columna agregada

Cada uno de los datos proporcionados será mostrado como las columnas de las vistas de cada uno de los usuarios. Cabe señalar que el administrador debe de tener una visión de cómo es que debe de quedar el cuadro.

Al final de llenar todos los formularios para el cuadro en cuestión es necesario que se asigne la fuente que será la encargada de mantener actualizados los datos.

Tu autentificacion es valida hasta 30 May 2005, 02:03:01
cs5
3

Advertencia: Seleccionar la fuente para la tabla :cs5 del apartado: Construcción

FUENTES	borrar
☒ Anuario Estadístico del Estado de Michoacán	borrar
☒ Comision Forestal del Estado de Michoacán	borrar
☒ INEGI	borrar
☐ Procuraduria Federal de Proteccion al Ambiente	borrar
☐ H. Ayuntamiento de Morelia, Dirección de Aseo Público	borrar
☐ Sistema de Cuentas Nacionales de México	borrar
☐ Registros Estadísticos del Centro	borrar
☐ Secretaria de Comunicaciones y Transportes	borrar
☐ SEPLADE Inventario de Obra Pública	borrar
Crear cuadros Cuadros	

Figura 4.11. Fuentes asignadas al cuadro

La vista mostrada en la figura 4.11 se origina del archivo fuentes.php dentro del proceso para crear un nuevo cuadro, modificará la tabla **fuentespara** en la cual se almacenan datos tales como el nombre del cuadro, clave de la fuente y la clave del apartado para facilitar su consulta posteriormente.

Una vez más se muestra la funcionalidad de las páginas dinámicas como se observa en la figura 4.12 que al finalizar el llenado de los formularios automáticamente aparecen los datos proporcionados al igual que las opciones del nuevo cuadro.

Cabe señalar que el administrador debe de tener un conocimiento global de la estructura que irá adquiriendo la herramienta.

4.4.4. Informes (consultas)

Es el proceso mediante el cual la herramienta toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa en la línea de comandos por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas como php, que forman parte de la abstracción de datos.

Las consultas solo las realiza el administrador todos los apartados y por lo tanto todos los cuadros pueden ser consultados por el administrador, las vistas proporcionadas para tal objetivo se muestran en la figura 4.12

Tu autentificación es valida hasta 30 May 2005, 02:03:55

Construcción

CUADRO	CONCEPTO	consultar modificar
cs3	Valor de la producción en el sector formal de la industria de la construcción según tipo de obra 1994-2003	consultar modificar
cs4	Valor de la producción en el sector formal de la industria de la construcción según sector institucional 1994-2001	consultar modificar
cs5	Licencias de construcción expedidas en la ciudad de Morelia por destino de la obra 2002	consultar modificar

Agregar cuadro

Concepto:

Nombre del cuadro

No. de campos

Figura 4.12. Cuadros del apartado construcción

Cuando se selecciona un cuadro para hacer una consulta solo aparecen los nombres de las fuentes a las cuales les fue asignado dicho cuadro hasta el momento, con su respectiva liga par acceder a la consulta.

Construcción

FUENTES	consultar
Anuario Estadístico del Estado de Michoacán	consultar
INEGI	consultar

Figura 4.13 Fuentes existentes

Cuando se elige alguna de las fuentes de la figura 4.13 para consultar, se desplegará un cuadro o mejor dicho una tabla virtual del contenido de la base de datos mostrado en la figura 4.14.

Construcción

Valor de la producción en el sector formal de la industria de la construcción según tipo de obra 1994-2003

cs3

Descripcion	1994o	1995o	1996o	1997o	1998o	1999o	2000o	2001o	2002o	2003o
TOTAL										
EDIFICACION										
AGUA, RIEGO Y SANEAMIENTO										
ELECTRICIDAD Y COMUNICACIONES										
TRANSPORTE										
PETROLEO Y PETROQUIMICA										
OTRAS CONSTRUCCIONES										

FUENTE: INEGI

Figura 4.14 Tabla virtual

En esta parte se utiliza la clase *table* de **phpLib** para generar esta vista ya que simplifica el trabajo de la realización de la tablas en código html. ^[8].

4.4.5. Adición de registros

Utilizando los elementos de la vista de la figura 4.12 al seleccionar la liga modificar se ingresa a una parte de la herramienta donde se pueden agregar columnas y/o renglones a determinado cuadro esta es mostrada en la figura 4.15.

- En el campo **Concepto** es específicamente para modificar el texto que se mostrará como título del cuadro al momento de ser consultado por los usuarios.
- En el campo **Agregar columna**, se escribe el nombre de la columna para el cuadro actual que desee agregar a la tabla base.
- En el campo **descripción**, escriba el nombre del nuevo renglón o fila para el cuadro actual.
- Los campos **Tipo** y **Longitud** son básicamente para identificar los datos que serán almacenados en ese renglón.

Construcción

CUADRO A MODIFICAR:

cs3

Concepto:

Valor de la producción en el sector
formal de la industria de la
construcción según tipo de obra 1994-

Agregar columna

Modificar

columnas

1994
1995
1996
1997
1998
1999
2000
2001
2002
2003

descripcion

TOTAL
EDIFICACION
AGUA, RIEGO Y SANEAMIENTO
ELECTRICIDAD Y COMUNICACIONES
TRANSPORTE
PETROLEO Y PETROQUIMICA
OTRAS CONSTRUCCIONES

Nueva Descripción

Descripción	Tipo	Longitud
	VARCHAR	

Agregar

Terminar

Figura 4.15 Adición de registros.

Es de importancia notar que las tablas de la base de datos que se modifican son la meta tabla, la tabla base y todas aquellas que fueron copiadas de la tabla base, es decir, todas las tablas creadas donde se almacenan los datos para cada una de las fuentes asignadas a dicho cuadro.

4.5. Manual del usuario

4.5.1. Usuarios finales

Frente a cualquier tipo de tecnología, muy poca gente se toma el tiempo de leer instrucciones. En su lugar, avanzamos experimentando. Por tal motivo se pretende crear una interfaz intuitiva en la cual no sea necesario ser un experto en el uso del Internet.

Permitiendo a los usuarios que tienen diferentes niveles de habilidad se adapten a la herramienta fácilmente.

La interfaz tiene pocas operaciones para que se entienda con facilidad, haciendo la más efectiva para los usuarios menos experimentados, evitando un conjunto de operaciones amplias y difíciles de comprender.

Ya que por medio de un conjunto simplificado de formularios y opciones, es fácil de aprender y recordar, porque puede concentrarse más en la aplicación en lugar de hacerlo en los detalles de la interfaz. Por lo tanto, la interfaz ofrece un medio para simplificar la complejidad de la herramienta, de manera que los usuarios principiantes puedan utilizar el sistema sin confundirse por tantos detalles.

Permitiendo que los usuarios interactúen en su lenguaje con la computadora. No se requieren habilidades especiales del usuario. Los problemas de implementación son mínimos.

Tomando en consideración los siguientes puntos para la interfaz escogida, el período de capacitación para los usuarios debe ser corto, el tiempo para corrección de los errores del sistema y de los usuarios debe ser corto, los usuarios poco frecuentes deben ser capaces de aprender rápidamente y recordar sin problemas la dinámica de llenado de los cuadros.

4.5.2. Agregando los datos

Para poder hacer el llenado de los cuadros es necesario que el usuario sea autenticado, una vez que logra acceder le mostrará la vista de la figura 4.15 en donde muestra el nombre del usuario y la fuente a la que pertenece y de igual forma los apartados en los cuales tiene que llenar los cuadros que le fueron asignados.

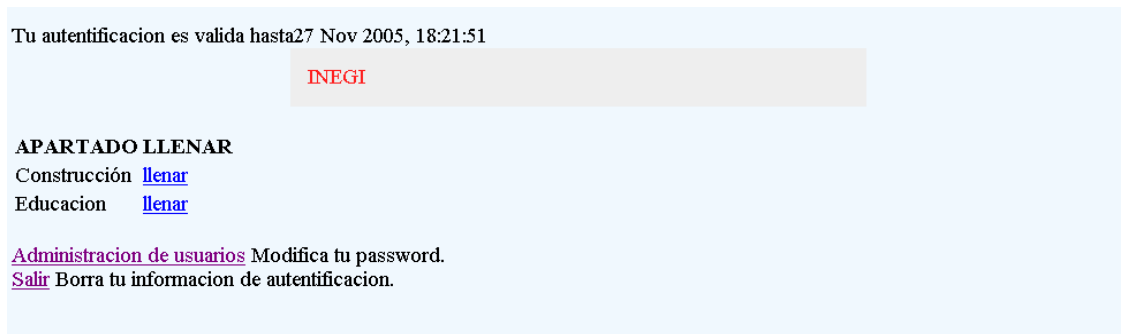


Figura 4.16 Usuario y apartados por llenar

La herramienta debe ajustarse a la mano de quien la usa para hacer su tarea más fácil de realizar. Por tal motivo se presenta de tal forma que le sea lo mas familiar posible de acuerdo a como lo ha estado realizando con lápiz y papel.

Tu autentificación es valida hasta 27 Nov 2005, 18:31:44

INEGI

Construcción

CUADRO	CONCEPTO	llenar
cs3	Valor de la producción en el sector formal de la industria de la construcción según tipo de obra 1994-2003	llenar
cs4	Valor de la producción en el sector formal de la industria de la construcción según sector institucional 1994-2001	llenar
cs5	Licencias de construcción expedidas en la ciudad de Morelia por destino de la obra 2002	llenar

Figura 4.17 Cuadros por llenar por los usuarios.

El aumento de la velocidad en la captura de datos y la reducción de errores es una de las ventajas que presenta la herramienta ya que anteriormente se tenía que pasar por varias manos para ser analizados los datos suministrados por las diferentes fuentes. Como se muestra en la figura 1.2 del Capítulo 1.

En esta etapa es importante estar consciente de la resistencia al cambio del personal y usuarios que están involucrados, ya que estos sistemas son importantes en el ahorro de mano de obra.

La figura 4.18 sugiere uno de los cuadros que generalmente son llenadas a mano, pretendiendo hacer uso de una idea u objeto ya existente en el dominio del conocimiento familiar del usuario hacia otro menos familiar, una base de datos.

Con el objetivo de que le sea fácil utilizar, recordar y cumplir con el objetivo de almacenar los datos de cada una de las fuentes.

Tus permisos son: user

INEGI

Construcción

Valor de la producción en el sector formal de la industria de la construcción según tipo de obra 1994-2003

Descripcion	1994o	1995o	1996o	1997o	1998o	1999o	2000o	2001o	2002o	2003o
TOTAL	15200									
EDIFICACION										
AGUA, RIEGO Y SANEAMIENTO										
ELECTRICIDAD Y COMUNICACIONES										
TRANSPORTE										
PETROLEO Y PETROQUIMICA										
OTRAS CONSTRUCCIONES										

Grabar datos

Figura 4.18 Formulario de almacenamiento de datos

Cambio de password

Para el usuario no es relevante saber como funcionan las cosas, en tanto que pueda usarla. Si logran hacer que algo funcione, no importa que tan mal lo haga, no buscarán otra manera de hacerlo. Si dieran con una mejor opción, seguro la usarían pero no la buscaran por ellos mismos. De ahí la utilización de la abstracción de datos.

Lo que en realidad es importante es que sepan como usarla.

Una de la opciones que tiene cada uno de los usuario es el cambiar su password, este es tratado como ya se menciono en el Capítulo 2 con el cifrado de el password tanto en el lado cliente como en el lado servidor en la base de datos para mantener la seguridad e integridad de los datos.

En la figura 4.19 se muestra activo solo el campo del password correspondiente al usuario actual. Tanto el nombre de usuario, nivel y fuente a la que pertenece el usuario solo lo puede cambiar el administrador.

Administracion de usuarios

Estas registrado como **angie** con nivel de **user**.
Tu autentificacion es valida hasta 27 Nov 2005, 18:19:22.

Nombre de usuario	Password	Nivel	Fuentes	Accion
angie	*****	user		Modificar

[Salir](#) Borra tu informacion de autentificacion.

Figura 4.19 Modificar el password

4.6. Resumen

Se utiliza el concepto de abstracción por sus propiedades esenciales, ya que nos ayuda a identificar los aspectos importantes, o examinar selectivamente ciertos aspectos de un problema, posponiendo o ignorando los detalles menos sustanciales, distractivos o irrelevantes, para los usuarios finales.

La principal dificultad para lograr dicha abstracción fue la asociada a comprender las tareas que el usuario desea realizar sobre la herramienta y las características de los diferentes usuarios que la utilizaran, aunado a esto tenemos que frente a cualquier tipo de tecnología, muy poca gente se toma el tiempo de leer instrucciones. En su lugar, avanzamos experimentando. Consecuentemente se ve en la necesidad de crear una interfaz intuitiva en la cual no sea necesario ser un experto en el uso del Internet.

Por lo tanto se utilizó la idea de un objeto ya existente en el dominio del conocimiento familiar del usuario hacia otro menos familiar, es decir, la herramienta se ajustó de tal forma que le sea lo mas parecido posible a cada uno de los usuarios de acuerdo a como lo ha estado realizando a lápiz y papel, para que al momento de ingresar los datos sea más fácil de realizar la tarea.

Al tratar de incorporar la herramienta, es importante darse cuenta que no solo se esta cambiando un proceso, esta intentando cambiar la cultura de la organización y las creencias más apreciadas por sus miembros. Es posible lograrlo, pero requiere tener claro el objetivo al que se quiere arribar y una comunicación excelente sobre los beneficios del cambio.

De igual forma tenemos una parte del nivel conceptual utilizado para generar cada una de las vistas que se le presenta a los diferentes tipos de usuarios, con el objeto de que se tenga una visión general de las partes esenciales de este nivel de la herramienta, mostrando nombre de tablas y de las páginas que se tendrían que modificar al momento de querer hacerle alguna mejora.

Capítulo 5. Conclusiones

Uno de los problemas a los que me enfrenté fue el de construir correctamente las abstracciones debido a que es una tarea difícil, la dificultad tiene que ver con el hecho de que siempre resulta complicado captar la esencia de un problema y articularlo de un modo conciso, de tal forma que a los usuarios finales les sea lo mas familiar posible, por ejemplo, hacer que la interfaz de la herramienta coincida con lo que conocen en la vida real, ya que las tablas de la base de datos no podrían mostrarse tal cual fueron diseñadas para la solución del problema de la captura de datos; aunado a esto fue necesario realizar la tarea de autenticar a los usuarios con el objetivo de restringir el acceso y al mismo tiempo proteger los datos por medio del manejo de sesiones, logrando a su vez que cada usuario solo vea la información que le corresponde o que se le permite ver ya sea para consultar y/o agregar datos, además de esta forma se tendrán mas organizados los datos almacenados para las consultas consecuentes.

El administrador puede realizar de una manera sencilla los cuadros para cada uno de los usuarios y el control de sesiones, todo esto lo podrá hacer sin que él tenga que conocer el código de la herramienta y mucho menos los comandos de bases de datos, esto incluye desde la creación de tablas que formaran parte de la estructura de la base de datos de esta herramienta, realizara las consultas de las estas sin preocuparse que comando utilizar para hacer una consulta acertada.

Por lo tanto hay que estar auditando y mejorando el código ya que no todos los programas son estables y siempre existirá una mejora, logrando que se fortalezca para llevarlo a un punto óptimo de funcionamiento.

Dentro de las mejoras esta contemplada la interfase gráfica o vistas finales ya que le faltaría hacerla un poco mas amigable, pero sin olvidar que el usuario promedio tiene poco conocimiento del uso de una computadora. Dentro de la autenticación del usuario faltaría establecer la generación y envío del password a una cuenta de correo electrónico.

Las propuestas para que la herramienta se fortalezca, es decir, llevarla prácticamente mas allá del propósito que se persiguió en este trabajo, sería hacer el Análisis Estadístico con la ayuda de la misma herramienta, facilitar la tarea de publicar los resultados, otra sería que una vez puesto en marcha y lograda una estabilidad en el estado, promover para que sea utilizada en el resto de los estados de la Republica Mexicana.

Concluyendo con la tesis, esta herramienta será un eslabón muy importante dentro del sistema de adquisición de datos reduciendo costos y tiempo en el proceso de recolección de datos, aportando un poco en el proceso de modernización nacional ya que su ejecución nos lleva a consolidar aspectos básicos relativos a los procesos de integración y difusión de la información.

Uno de los puntos finales y de gran importancia nos lleva a dirigir las miradas al gran auge del código abierto ya que gracias a este se nos facilita el desarrollo de nuevos sistemas, dejando a un lado el factor económico pero sin olvidarlo por completo, de igual forma y como un punto mas a favor de la herramienta reside en la facilidad de acceder a esta, ya que los usuarios finales excluyendo al administrador no tendrá el problema de tener que instalar un software, ya que con la experiencia adquirida hasta el momento es uno de los principales problemas de aceptabilidad de los nuevos sistemas, de igual forma esta la gran variedad de los sistemas operativos y la compatibilidad de los programas con estos.

Bibliografía

- [1] T. Connolly, C. Begg, A. Strachan (1996)
Database Systems. A Practical Approach to Design,
Implementation and Management
Addisson-Wesley
Segunda Edición 1998
 - [2] R. Elmasri, S. B. Navathe (1997)
Sistemas de Bases de datos conceptos fundamentales
Segunda Edición
Addisson-Wesley Iberoamericana
Tercera Edición 1999 (en ingles, por Addison-Wesley)
 - [3] C.J. Date (1993)
Introducción a los sistemas de Bases de Datos
Volumen I, Quinta Edición
Addison-Wesley Iberoamericana
Sexta Edición en 1995 (en ingles, por Addison-Wesley)
 - [4] PHP and MySQL Web Developmet
Luke Welling and Laura Thomson (2001)
Sams Publishing
 - [5] PHP (Personal Home Page, Hypertext Preprocessor)

<http://www.php.net>
<http://www.zend.com>
<http://www.phpwizard.net>
<http://www.phpbuilder.com>
 - [6] MySQL

<http://www.mysql.com>
 - [7] Apache Resources

<http://www.apache.org>
 - [8] Web Programación en castellano

http://www.programacion.net/tutorial/php4/4/#php4_phplib1
 - [9] SEPLADE (Secretaria de Planeación y Desarrollo Estatal)

<http://seplade.michoacan.gob.mx/anuario/index.php>
 - [10] INEGI

<http://www.inegi.gob.mx/>
<http://pnd.presidencia.gob.mx/>
-