



Benemérita Universidad

---

Autónoma de Puebla

Facultad de Ciencias de la Computación.

“Sistema para el manejo de información  
electrónica de una biblioteca”

TESIS PROFESIONAL QUE PARA OBTENER  
EL TÍTULO DE LICENCIADA EN CIENCIAS  
DE LA COMPUTACIÓN.

PRESENTA:  
Maritza Alejandrez Olivo.

ASESOR:  
Dr. Manuel Martín Ortiz.

Puebla, Pue.

DICIEMBRE 2009



## CONTENIDO

<b>CONTENIDO .....</b>	<b>2</b>
<b>CONTENIDO DE FIGURAS.....</b>	<b>5</b>
<b>RESUMEN.....</b>	<b>6</b>
<b>AGRADECIMIENTOS.....</b>	<b>7</b>
<b>INTRODUCCIÓN .....</b>	<b>8</b>
<b>DISTRIBUCIÓN DE LA TESIS.....</b>	<b>9</b>
<b>CAPÍTULO 1.....</b>	<b>11</b>
<b>CONCEPTOS Y TÉRMINOS BÁSICOS DE LA INGENIERÍA DE SOFTWARE Y BASES DE DATOS RELACIONALES .....</b>	<b>11</b>
<b>INTRODUCCIÓN .....</b>	<b>11</b>
<b>1.1 ¿QUÉ ES LA INGENIERÍA DE SOFTWARE? .....</b>	<b>11</b>
<b>1.2 ETAPAS GENERALES EN EL DESARROLLO DE SOFTWARE. ....</b>	<b>12</b>
<b>1.3 ANÁLISIS DE REQUERIMIENTOS.....</b>	<b>16</b>
<b>1.4 DIAGRAMA DE ENTIDAD-RELACIÓN. ....</b>	<b>18</b>
<b>1.5 EL MODELO RELACIONAL. ....</b>	<b>19</b>
<b>1.6 BASES DE DATOS RELACIONALES. ....</b>	<b>21</b>
<b>1.7 MODELO CLIENTE SERVIDOR. ....</b>	<b>23</b>
<b>CAPÍTULO 2.....</b>	<b>27</b>
<b>ASPECTOS DE ESPECIFICACIÓN DEL SISTEMA .....</b>	<b>27</b>
<b>INTRODUCCIÓN.....</b>	<b>27</b>
<b>2.1 DEFINICIONES FORMALES.....</b>	<b>27</b>
<b>2.2 ANTECEDENTES. ....</b>	<b>28</b>
<b>2.3 MOTIVACIONES. ....</b>	<b>29</b>
<b>2.4 OBJETIVOS.....</b>	<b>30</b>
<b>2.5 REQUERIMIENTOS DEL SISTEMA (RECURSOS DEL PROYECTO). ....</b>	<b>30</b>
<b>2.6 RESULTADOS ESPERADOS. ....</b>	<b>31</b>
<b>2.7 DESCRIPCIÓN DE INTERFACES CON OTROS SISTEMAS.....</b>	<b>32</b>
<b>CAPÍTULO 3.....</b>	<b>34</b>
<b>DESCRIPCIÓN DE LA INFORMACIÓN .....</b>	<b>34</b>



---

INTRODUCCIÓN.....	34
3.1 DISEÑO DE LA BASE DE DATOS.....	34
3.2 ANALISIS ESTRUCTURADO DEL SISTEMA ITZA. ....	35
3.2.1 DIAGRAMA DE FLUJO DE DATOS .....	35
3.2.2 DICCIONARIO DE DATOS.....	42
3.2.3 MINI-ESPECIFICACIONES.....	44
3.3 DISEÑO CONCEPTUAL DE LA BASE DE DATOS DEL SISTEMA ITZA. ....	46
3.3.1 MODELO ENTIDAD-RELACIÓN. ....	46
3.4 DISEÑO LÓGICO. ....	47
3.4.1 MODELO RELACIONAL.....	47
3.4.2 VALIDACIÓN DE LAS REGLAS DE NORMALIZACIÓN DE LA BASE DE DATOS DEL SISTEMA ITZA. ....	47
3.4.3 DISEÑO RELACIONAL DE LA BASE DE DATOS DEL SISTEMA ITZA.....	51
3.5 DISEÑO DE LA INTERFAZ DEL USUARIO DEL SISTEMA ITZA. ....	51
<b><u>CAPÍTULO 4.....</u></b>	<b><u>54</u></b>
<b><u>ASPECTOS DE IMPLEMENTACIÓN DEL SISTEMA .....</u></b>	<b><u>54</u></b>
4.1 PROGRAMAS DE LIBRE DISTRIBUCIÓN EN PHP.....	55
4.2 CREACIÓN DE LA BASE DE DATOS, USUARIO Y PRIVILEGIOS. ....	56
4.3 CREACIÓN DE LAS TABLAS DEL SISTEMA ITZA. ....	57
4.4 PROGRAMACIÓN DEL SISTEMA.....	59
4.5.1 PÁGINA PRINCIPAL DEL SISTEMA ITZA. ....	59
4.5.2 INTERFAZ DEL BUSCADOR DE DOCUMENTOS ELECTRÓNICOS POR TÍTULO.....	60
4.5.3 INTERFAZ DE INICIO DE SESIÓN.....	62
4.5.4 INTERFAZ DE PANTALLA CON MENU DE ADMINISTRADORES.....	62
4.5.5 INTERFAZ DEL ADMINISTRADOR PARA REGISTRO DE PROFESORES, CURSOS Y FACULTADES. ....	63
4.5.6 INTERFAZ DEL ADMINISTRADOR PARA CONSULTA DE PROFESORES, DOCUMENTOS, CURSOS Y FACULTADES. ....	64
4.5.7 INTERFAZ DEL ADMINISTRADOR PARA MODIFICACIÓN DE PROFESORES, DOCUMENTOS, CURSOS Y FACULTADES. ....	66
4.5.8 INTERFAZ DEL ADMINISTRADOR PARA ELIMINACIÓN DE PROFESORES, CURSOS Y FACULTADES. ....	67
4.5.9 INTERFAZ DEL ADMINISTRADOR PARA RESPALDO Y RESTAURACIÓN DE LA BASE DE DATOS.....	68
4.5.10 INTERFAZ DEL MENU DEL EDITOR. ....	69
4.5.11 SALIR DEL SISTEMA.....	69
CONCLUSIONES.....	70
LIMITACIONES.....	70
PERSPECTIVAS.....	70
<b><u>BIBLIOGRAFÍA.....</u></b>	<b><u>71</u></b>
<b><u>APÉNDICE 1.....</u></b>	<b><u>73</u></b>
<b><u>MANUAL DE INSTALACIÓN.....</u></b>	<b><u>73</u></b>

---



---

<b>APÉNDICE 2.....</b>	<b>75</b>
<b>MANUAL DE USUARIO.....</b>	<b>75</b>
¿CÓMO ACCEDER AL SISTEMA SIENDO ADMINISTRADOR?.....	75
¿CÓMO ACCEDER AL SISTEMA SIENDO EDITOR?.....	80
¿CÓMO ACCEDER AL SISTEMA SIENDO INVITADO?.....	82



## CONTENIDO DE FIGURAS

FIGURA 1.1 FASE DE DEFINICIÓN Y PLANEACIÓN EN EL DESARROLLO DE SOFTWARE. ....	13
FIGURA 1.2 FASE DE DESARROLLO DE SOFTWARE.....	14
FIGURA 1.3 FASE DE MANTENIMIENTO EN EL DESARROLLO DE SOFTWARE.....	16
FIGURA 1.4 DIAGRAMA ENTIDAD-RELACIÓN EXTENDIDO. ....	19
FIG. 3.1 DIAGRAMA DE CONTEXTO DEL SISTEMA DE BASE DE DATOS ITZA. ....	36
FIG. 3.2 DFD NIVEL 1 DEL SISTEMA DE BASE DE DATOS ITZA.....	37
FIG. 3.3 PROCESA LOGIN, PASSWORD. ....	38
FIG. 3.4 ACTUALIZACIÓN DE DOCUMENTOS.....	39
FIG. 3.5 ACTUALIZACIÓN DE ADMINISTRACIÓN.....	40
FIG. 3.6 MOSTRAR TIPOS DE BÚSQUEDA. ....	41
FIG. 3.7 ELECCIÓN DE RESULTADO.....	42
FIG. 3.8 DIAGRAMA E-R DEL SISTEMA DE BASE DE DATOS ITZA.....	46
FIG. 3.9 PRIMERA VISIÓN DE LA INTERFAZ DEL SISTEMA. ....	52
FIG. 4.1 INTERFAZ WEB DE LA APLICACIÓN PHPMYADMIN. ....	56
FIG. 4.2 INSERCIÓN DE LA NUEVA BASE DE DATOS ITZA. ....	57
FIG. 4.3 DISEÑO RELACIONAL DE LA BASE DE DATOS ITZA.....	58
FIG. 4.4 CREACIÓN DE LAS TABLAS EN PHPMYADMIN.....	58
FIG. 4.5 PÁGINA PRINCIPAL DEL SISTEMA.....	60
FIG. 4.6 BÚSQUEDA POR TÍTULO DE DOCUMENTO. ....	60
FIG. 4.7 VISTA DETALLADA DEL DOCUMENTO. ....	61
FIG. 4.8 DESCARGA DEL DOCUMENTO ESPECÍFICO. ....	61
FIG. 4.9 INICIO DE SESIÓN.....	62
FIG. 4.10 PANTALLA DE INICIO DE ADMINISTRADOR DEL SISTEMA.....	63
FIG. 4.11 REGISTRO DE DATOS DE UN NUEVO PROFESOR AL SISTEMA.....	64
FIG. 4.12 REGISTRO DE UN NUEVO CURSO AL SISTEMA. ....	64
FIG. 4.13 CONSULTA DE PROFESORES.....	65
FIG. 4.14 CONSULTA DE DOCUMENTOS.....	65
FIG. 4.15 PANTALLA PARA ELECCIÓN DE REGISTRO A MODIFICAR. ....	66
FIG. 4.16 PANTALLA DE MODIFICACIÓN DEL REGISTRO. ....	67
FIG. 4.17 PANTALLA DE ELIMINACIÓN DEL REGISTRO.....	67
FIG. 4.18 RESPALDO DE BASE DE DATOS. ....	68
FIG. 4.19 RESTAURACIÓN DE BASE DE DATOS. ....	68
FIG. 4.20 REGISTRO DE UN DOCUMENTO Y SU ARCHIVO.....	69



## RESUMEN

El siguiente proyecto de tesis es una herramienta y un apoyo tanto para la comunidad estudiantil como para el personal académico, la realización de este proyecto se basa en la importancia y la percepción de que en los últimos años los profesores de cualquier nivel, en especial el nivel superior, tienen investigaciones en formato electrónico generados por ellos mismos o documentos útiles para la comunidad estudiantil, esta documentación electrónica por lo regular se encuentra en páginas de Internet propias y de su existencia sólo la saben algunos. El sistema de base de datos para la consulta de los medios electrónicos de una Biblioteca se aplicará para la consulta de este material. El sistema tendrá una cantidad importante de documentos, se ordena la documentación en una base de datos para que toda persona pueda ingresar a ésta información de forma rápida, también se mantiene el control y la seguridad del sistema mediante un administrador que será la única persona que dará de alta a los profesores en la base de datos y los profesores serán los responsables de administrar su propia documentación.

**A continuación se describe de una forma breve las características de este proyecto:**

Se utiliza una base de datos y una interfaz amigable, la base de datos guardará los datos característicos de cada documento como son: un id, el nombre del profesor que lo creo, la fecha de ingreso en la base de datos, el número de páginas, título, resumen, formato, la ubicación del archivo en el servidor etc., también guardará todo lo referente a los cursos y datos generales de los profesores.

La interfaz del sistema se basa en los estándares Web y es por medio de ella que los usuarios pueden comunicarse e interactuar. Las principales funciones de la interfaz Web son, el control de acceso, la manipulación de información del sistema y de los archivos, la actualización del sistema y la búsqueda, recuperación de información y descarga de documentos.

Desde el inicio de este documento se trata de llevar de la mano al lector para no dejarle duda de lo que se está haciendo o por lo menos dejarle las menos posibles, también se indican las referencias necesarias para la investigación aún más profunda de cada tema.

Además es importante mencionar que la información contenida en las tesis es actual y fidedigna, además, por estar en formato digital proporcionan ventajas como son: la ubicación de obras de manera rápida, el acceso ilimitado a los diversos ejemplares, la preservación de la estructura original, entre otras.



## ***AGRADECIMIENTOS***

*Primero a mi papá Chuchito, por estar presente en cada momento de mi vida, por no dejarme vencer y agradecerle infinitamente la familia que me regalo, que es lo más grande e importante que tengo.*

*A mis papás Chucho y Mary, porque nunca me han abandonado y por todo el AMOR que me han brindado, los quiero mucho, los AMO.*

*Finalmente a todos los angelitos que estuvieron conmigo en su momento y me apoyaron, entre ellos el Dr. Manuel Martín Ortiz por las enseñanzas y el apoyo que me brindo durante todo este tiempo.*



# INTRODUCCIÓN

Sabemos que en la actualidad la sistematización en las empresas modernas juega un papel muy importante, ya que además de disminuir el papeleo agiliza muchas operaciones, esto ha modificado substancialmente el papel que juega la informática en las empresas y en las instituciones educativas, además de ser un elemento de apoyo a las operaciones básicas, se ha convertido en un medio para obtener ventajas tecnológicas y competitivas, además que también se ha modificado la forma de operar de las organizaciones, y esto, incluye varias responsabilidades a las ciencias de la computación.

- Las aplicaciones deben ser desarrolladas en un corto tiempo, esto quiere decir que se deberán desarrollar más rápidamente, pues los requerimientos de las empresas e instituciones cambian continuamente.
- La importancia de contar con una buena información, lo que destaca lo importante que son los sistemas de información ejecutivos y los sistemas de soporte a las decisiones.
- Cada vez es más importante el hacer que la información esté disponible en donde se necesita. Para lograr esto, tanto la información como los sistemas para procesarla deben ser distribuidos.
- Las nuevas aplicaciones deben basarse en tecnologías que disminuyan los costos de desarrollo y mantenimiento, en aspectos relacionados con el hardware, el software, la operación, el entrenamiento, el personal y el mantenimiento. Además, se requiere que las nuevas aplicaciones se puedan comunicar con las existentes.
- Con el fin de aumentar la productividad y de facilitar el uso de las aplicaciones por parte de los usuarios, se requieren interfaces simples e intuitivas.
- Las tecnologías computacionales modernas buscan responder a las necesidades de las empresas y para ello plantean nuevas formas de hacer las cosas.

Mucha gente hoy en día utiliza la tecnología computacional como el Internet para dar a conocer sus ideas, así como sus publicaciones. Lo hacen por este medio ya que muchas veces es más rápido, accesible, seguro y confiable.

Personas dedicadas a la investigación, entre ellos profesores de nivel medio o superior tienen investigaciones o información que les interesa compartir con la comunidad estudiantil, esta documentación en formato electrónico la distribuyen en algún medio extraíble como puede ser disco compacto, memoria USB u otros medios. También es muy común que estas investigaciones generadas por ellos mismos o documentos útiles como referencias para sus cursos los almacenen en servidores Web para ser accedidos a través de Internet sin importar la cantidad y la ubicación de las personas que soliciten este tipo de información.

Muchas veces es mejor hacerlo mediante este medio, ya que anteriormente se hacía sacando copias al documento o a muchas páginas de un libro. Esto es costoso en dinero, como el desgaste físico del documento y hasta la confusión o pérdida de información.

Es necesario la creación de un sistema de base de datos en Internet que mantenga almacenada, ordenada y disponible toda esta información para mejorar el proceso de consulta; considerando la importancia de la difusión de los documentos generados en formato electrónico, como pueden ser: PDF, documentos creados en algún editor de texto, hojas de cálculo,



documentos escaneados, etc. y la necesidad de que no se pierdan los datos al llegar a su destinatario, teniendo la certeza que la información es confiable para su consulta y/o descarga por medio de un sistema administrativo.

## ***DISTRIBUCIÓN DE LA TESIS***

### **CAPÍTULO 1**

En este capítulo se presentan los conceptos y términos básicos de la Ingeniería de Software y de las bases de datos relacionales, son definiciones formales, los métodos y procedimientos para el desarrollo del sistema. Con este capítulo entenderemos el procedimiento que se utilizó en la realización de este proyecto de tesis.

### **CAPÍTULO 2**

En el capítulo 2 se presentan los aspectos de especificación del sistema, los objetivos, servicios (arquitectura desde el punto de vista del usuario) y la plataforma de instalación. Tiene como propósito especificar los requerimientos que utiliza el sistema, es decir, describir de manera general lo que realiza el sistema; se propone que de manera sencilla se pueda comprender el ámbito del trabajo a realizar, los riesgos en los que se pueda incurrir, los recursos requeridos, las tareas a llevar a cabo y los pasos que hay que recorrer.

También se hace una explicación breve de lo que es el mundo de las bibliotecas, sus definiciones y la problemática actual.

### **CAPÍTULO 3**

El capítulo 3 muestra los aspectos del diseño del sistema, el diseño de los procedimientos (arquitectura del sistema desde el punto de vista del programador), el diseño de la base de datos (diagrama del flujo de datos), diseño conceptual (modelo E-R), diseño lógico (tablas), validación de las reglas de normalización (tomar cada tabla y validar el proceso de normalización) y el diseño de la interfaz de usuario (pseudocódigo).

### **CAPÍTULO 4**

En el último capítulo se describen los aspectos de implementación del sistema, las herramientas de programación (lenguajes), la plataforma de instalación; es decir, se describe cómo implementar en MySQL la base de datos diseñada; se menciona porque el uso del lenguaje PHP en la creación de las páginas Web y se muestran algunas pantallas que muestran el funcionamiento del sistema. En este capítulo también se presentan las conclusiones y perspectivas del proyecto, la bibliografía utilizada y finalmente se concluye la tesis con dos apéndices, el primero es un manual de instalación y el segundo un manual de usuario.



# *CAPÍTULO*

# *1*

## *CONCEPTOS Y TÉRMINOS BÁSICOS DE LA INGENIERÍA DE SOFTWARE Y BASE DE DATOS RELACIONALES.*

### *INTRODUCCIÓN*

- 1.1. ¿QUÉ ES LA INGENIERÍA DE SOFTWARE?*
- 1.2. ETAPAS GENERALES EN EL DESARROLLO DE SOFTWARE.*
- 1.3. ANÁLISIS DE REQUERIMIENTOS.*
- 1.4. DIAGRAMA ENTIDAD-RELACIÓN.*
- 1.5. EL MODELO RELACIONAL.*
- 1.6. BASES DE DATOS RELACIONALES.*
- 1.7. MODELO CLIENTE-SERVIDOR.*



# CAPÍTULO 1

## CONCEPTOS Y TÉRMINOS BÁSICOS DE LA INGENIERÍA DE SOFTWARE Y BASES DE DATOS RELACIONALES.

### *INTRODUCCIÓN*

Actualmente los sistemas computacionales se caracterizan por una rápida evolución de los componentes hardware y el software, puesto que es el mecanismo que nos permite utilizar y explotar este potencial.

Entonces, a la hora de plantearnos la adquisición de un sistema informático completo para gestionar una empresa o controlar un proceso industrial, el software es lo que marca la diferencia. El desarrollo del software no es una tarea fácil, es necesario analizar que es lo que tenemos que hacer, como lo vamos a hacer y como se va a controlar el desarrollo del mismo de forma que al final obtengamos los resultados esperados.

Existen buenos métodos de desarrollo de software, pero quizá el problema esté en que no están suficientemente difundidos o valorados. Recientemente, estas técnicas están logrando una amplia aceptación [1].

### *1.1 ¿QUÉ ES LA INGENIERÍA DE SOFTWARE?*

Es una disciplina que integra métodos (“*como*” construir técnicamente el software), herramientas (suministran un soporte automático para los métodos) y procedimientos (junta los métodos y las herramientas, facilita un desarrollo racional y oportuno) para el desarrollo de software de computadora [1]. Se han propuesto varios paradigmas diferentes, cada uno exhibe ventajas y desventajas, pero todos tienen una serie de fases genéricas en común.

La Ingeniería de Software está compuesta por una serie de pasos que abarcan los métodos, las herramientas y los procedimientos. Estos pasos se denominan frecuentemente *paradigmas de la Ingeniería de Software*. Entre los paradigmas se tiene los siguientes [1]:

- **Clásico o modelo Secuencial:** Esta aproximación ve al proceso de software como un conjunto de etapas sucesivas. Después de que se define una, se aprueba y se procede a desarrollar la siguiente etapa. Este modelo se emplea como un medio para hacer el proceso de desarrollo más visible en las actividades de planeación y generación de reportes.
- **Diseño por prototipos:** Es un proceso que facilita al programador la creación de un modelo de software, el cual tomará una de estas tres formas:
  1. Un prototipo de papel o un modelo basado en computadora que describa la interacción humano-computadora



2. Un prototipo que implementa algunos subconjuntos de la función requerida del programa deseado
  3. Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.
- **Modelo Espiral:** Se ha desarrollado para cubrir las mejores características de los dos modelos anteriores, añadiendo un nuevo elemento: el análisis de riesgo. El modelo define cuatro actividades principales:
    1. Planificación. Determinación de objetivos, alternativas y restricciones.
    2. Análisis de riesgo. Análisis de alternativas e identificación y/o resolución de riesgos.
    3. Ingeniería. Desarrollo del producto del “siguiente nivel”.
    4. Evaluación del cliente. Valoración de los resultados de la ingeniería.
  - **De desarrollo evolutivo.**
  - **Desarrollo basado en componentes.**
  - **Técnicas de 4ª. Generación:** Este término abarca un amplio espectro de herramientas de software, que facilitan la especificación de algunas características del software a alto nivel. Es evidente que cuando mayor sea el nivel de la especificación del software, más rápido se podrá construir el programa. Este paradigma se orienta hacia la posibilidad de especificar el software a un nivel más próximo al lenguaje natural.

## ***1.2 ETAPAS GENERALES EN EL DESARROLLO DE SOFTWARE.***

El proceso de desarrollo contiene tres fases genéricas independientemente del paradigma elegido. Las tres fases: *planeación y definición, desarrollo y mantenimiento*, se encuentran en todos los desarrollos de software [1].

**Planeación y Definición:** Se centra sobre el ¿qué se quiere? (Fig. 1.1) Se intenta identificar que información ha de ser procesada, que función y rendimiento se desea, qué interfaces han de establecerse, qué restricciones de diseño existen y qué criterios de validación se necesitan para definir un sistema correcto y se producirán tres pasos específicos:

1. Análisis del sistema: El análisis define el papel de cada elemento de un sistema.
2. Planificación del proyecto de software: Una vez establecido el ámbito del software, se analizan los riesgos, se asignan los recursos, se definen las tareas y se planifica el trabajo.
3. Análisis de requerimientos: El ámbito proporciona la dirección a seguir, pero antes de iniciar el trabajo, es necesario disponer de la información de manera más detallada del ámbito de la información y de la función del software.

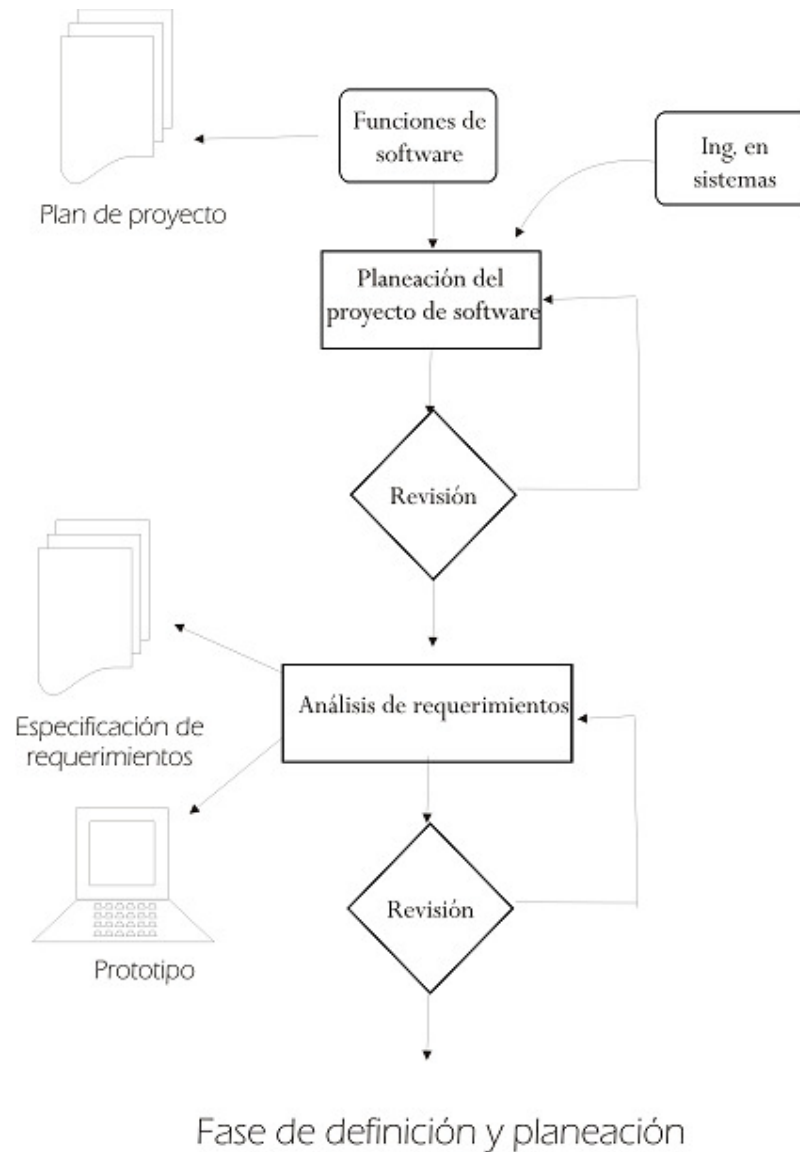


Figura 1.1 Fase de definición y planeación en el desarrollo de software.

**Desarrollo:** Se centra en el ¿cómo se logrará? (Fig. 1.2) El desarrollador intenta descubrir como han de diseñarse las estructuras de datos y la arquitectura del software, como han de implementarse los detalles procedurales, como ha de traducirse el diseño a un lenguaje de programación, y como se deben realizar las pruebas. Se producirán tres pasos concretos:

1. Diseño del software. Traduce los requerimientos del software a un conjunto de representaciones que describen la estructura de los datos, la arquitectura, el procedimiento algorítmico y las características de la interfaz.
2. Codificación: Las representaciones del diseño deben ser traducidas a un lenguaje de programación, dando como resultado unas instrucciones ejecutables por computadora.



3. Prueba de software: Una vez que se ha implementado el software en una forma ejecutable por la máquina, debe ser probado para descubrir los defectos que pueden existir en la función, en la lógica y en la implementación.

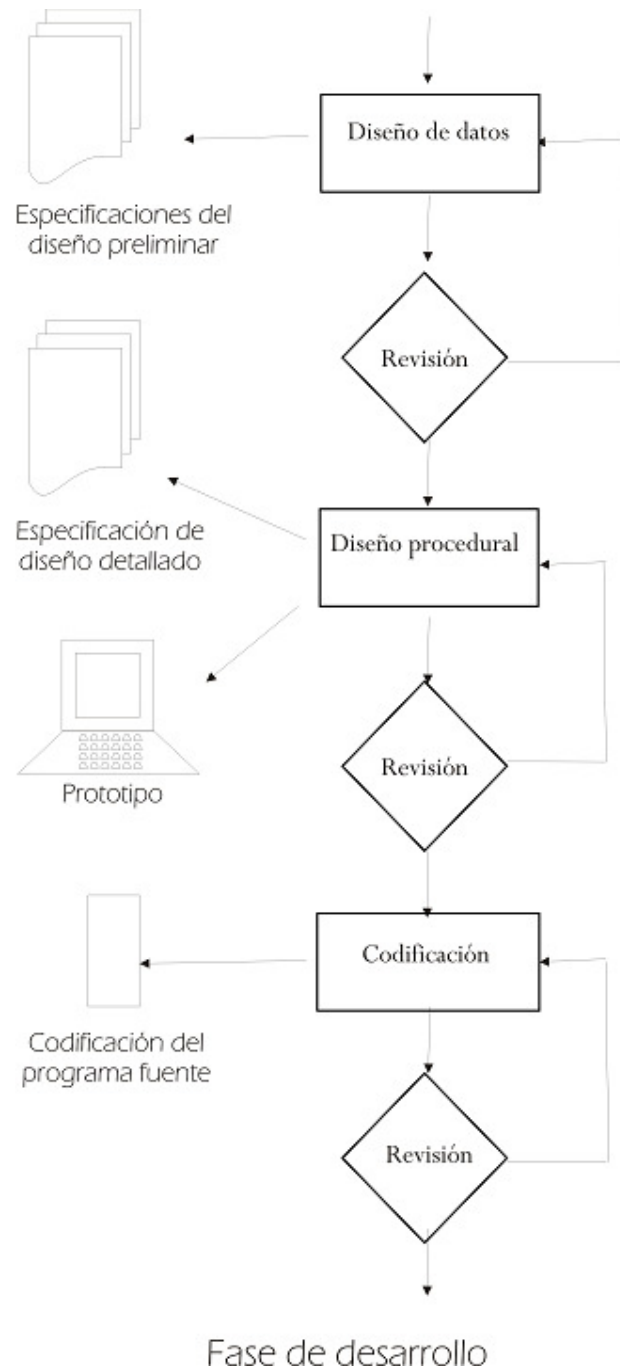
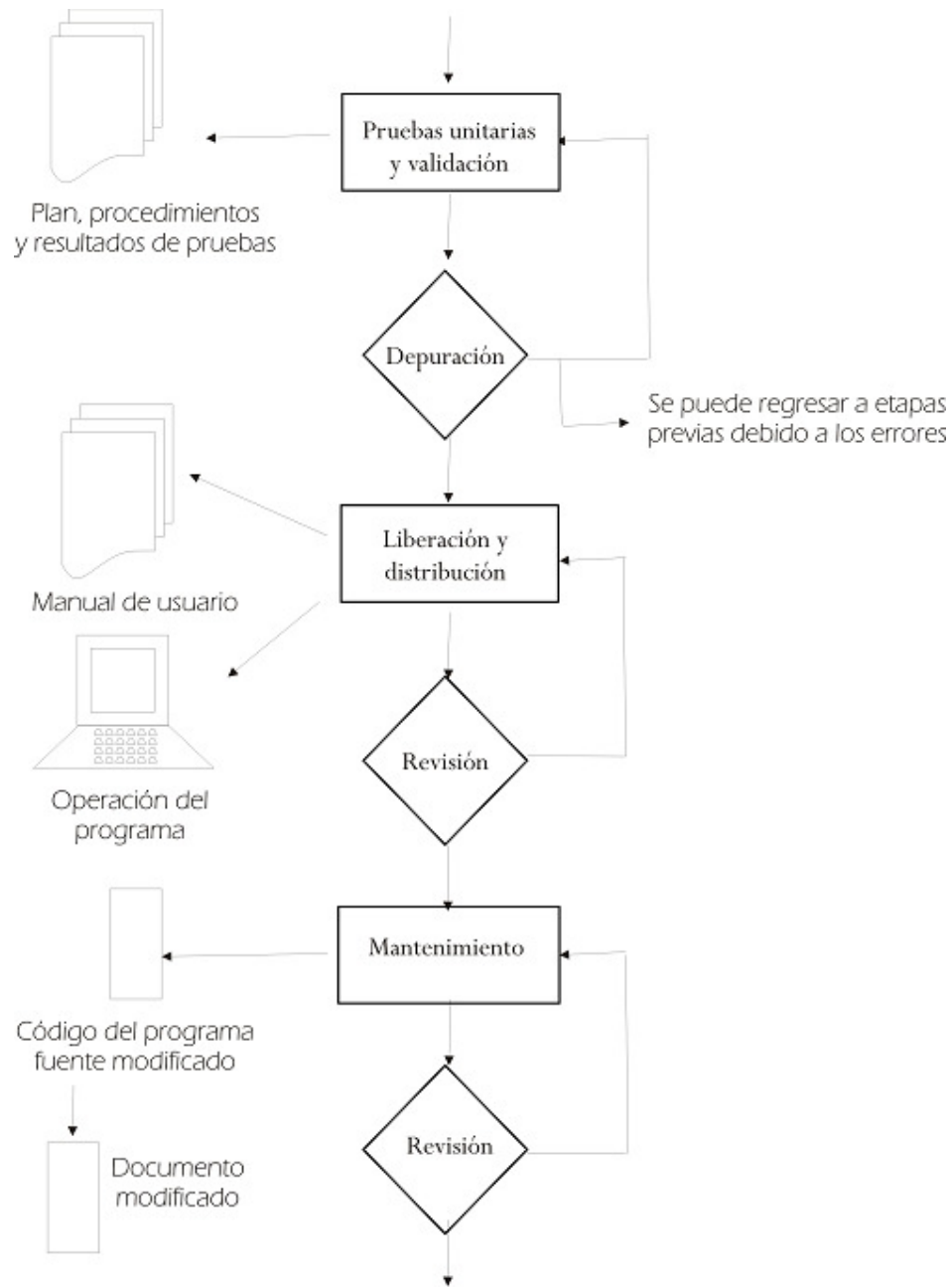


Figura 1.2 Fase de desarrollo de software.



**Mantenimiento:** Se centra en el ¿cómo se controlará el cambio? (Fig. 1.3). Esta fase vuelve a aplicar los pasos de definición, planeación y desarrollo, pero en el contexto del software ya existente. Durante esta fase se encuentran tres tipos de cambios:

1. **Corrección:** El mantenimiento correctivo cambia el software para corregir los defectos.
2. **Adaptación:** Consiste en modificar el software para acomodarlos a los cambios de su entorno externo con el paso del tiempo.
3. **Mejora:** Este mantenimiento amplía el software más allá de sus requerimientos funcionales originales.



### Fase de mantenimiento

Figura 1.3 Fase de mantenimiento en el desarrollo de software.

## 1.3 ANÁLISIS DE REQUERIMIENTOS.

El análisis de requerimientos es un proceso de descubrimiento, refinamiento, modelado y especificación. En este análisis se pueden identificar cinco áreas de esfuerzo: 1) reconocimiento del problema, 2) evaluación y síntesis, 3) modelización, 4) especificación, 5) revisión.



Como resultado del análisis se desarrolla la *especificación de requerimientos de software*. La revisión esencial para asegurar que el desarrollador del software y el cliente tengan la misma percepción del sistema [1].

## **ANÁLISIS ESTRUCTURADO.**

El análisis estructurado, es una actividad de construcción de modelos. Mediante una notación que es única del método, creamos modelos que reflejan el flujo y contenido de la información (datos y control).

### **Diagrama de flujo de datos (DFD).**

Permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por *conductos* y *tanques de almacenamiento de datos*. También se utilizan otros sinónimos para el diagrama de flujo de datos como son: carta de burbujas, DFD, diagrama de burbujas, modelo de proceso, diagrama de flujo de trabajo o modelo de función [1].

El DFD es sólo una de las herramientas de modelado disponibles y que únicamente proporciona un punto de vista de un sistema, el orientado a las funciones. Los componentes de un DFD son:

**El proceso.** Es el primer componente del DFD, algunos sinónimos son: burbuja, función o transformación. Muestra una parte del sistema que transforma entradas en salidas. Se representa gráficamente como un círculo, un óvalo o un rectángulo con esquinas redondeadas.

**El flujo.** Se representa gráficamente por medio de una flecha que entra o sale de un proceso, es para describir el movimiento de bloques o paquetes de información de una parte del sistema a otra. En la mayor parte de los sistemas, los flujos realmente representan datos, es decir: bits, caracteres, mensajes, etc.

**El almacén.** Se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por dos líneas paralelas. Algunos desarrolladores se refieren a los almacenes como archivos o bases de datos.

El último componente de un DFD es el **terminador** que se representa por un rectángulo y representan entidades externas con las cuales el sistema se comunica.

### **El diccionario de datos.**

Es otra herramienta muy importante del modelado [1]. Es un listado organizado de todos los datos pertinentes del sistema, con definiciones precisas y rigurosas para que tanto el usuario como el desarrollador tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios. Este diccionario se define haciendo lo siguiente:

- Describe el *significado* de los flujos y almacenes que se muestran en los DFD.
- Describe la *composición* de agregados de paquetes de datos que se mueven a lo largo de flujos, es decir, paquetes complejos que pueden descomponerse en unidades más elementales.
- Describen la composición de los paquetes de datos en los almacenes.



- Especifica los *valores* y unidades relevantes de piezas elementales de información en los flujos de datos y en los almacenes de datos.

### **Especificaciones de proceso.**

Es la descripción de que es lo que sucede en cada burbuja primitiva de nivel más bajo de un DFD [1]. Existe una variedad de herramientas que podemos utilizar para producir la especificación de proceso:

*Tablas de decisión.* Se utilizan si el proceso debe producir alguna salida o tomar alguna acción basada en decisiones complejas. Se crea listando todas las variables relevantes (condiciones o entradas) y todas las acciones relevantes en su lado izquierdo.

*Lenguaje estructurado.* Es el lenguaje español con importantes restricciones sobre el tipo de frases que pueden utilizarse y la manera en que puedan utilizarse dichas frases. Su propósito es hacer un balance razonable entre la precisión del lenguaje formal de programación, la informalidad y legibilidad del lenguaje cotidiano.

*Pre/post condiciones.*

*Diagramas de flujo.* Se da en el caso de la descripción lógica detallada.

*Diagramas de Nass/Shneiderman.* Usualmente son más organizados, más estructurados y más comprensibles que un diagrama de flujo típico, por esto son preferidos como herramientas para crear especificaciones de proceso.

Estas herramientas deben satisfacer dos requerimientos cruciales:

1. La especificación del proceso debe expresarse de una manera que puedan verificar tanto el usuario como el desarrollador.
2. El proceso debe especificarse en una forma que pueda ser comunicada efectivamente al público amplio que esté involucrado.

## ***1.4 DIAGRAMA DE ENTIDAD-RELACIÓN.***

Es una notación gráfica para modelar datos el cual describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema, y representa un gran beneficio ya que enfatiza las relaciones entre almacenes de datos en el DFD (Fig. 1.4) [1].

Existen cuatro componentes principales en un diagrama de entidad-relación:

1. Tipos de objetos.
2. Relaciones.
3. Indicadores asociativos de tipo de objeto.
4. Indicadores de supertipo/subtipo.



El tipo de objeto se representa por medio de una caja rectangular y representa un conjunto de objetos (cosas) del mundo real cuyos miembros individuales tienen las siguientes características:

- Cada uno puede identificarse de manera única por algún medio.
- Cada uno juega un papel necesario en el sistema que se construye. Es decir, para que el tipo de objeto sea legítimo, debe poder decirse que el sistema no puede operar sin tener acceso a esos miembros.
- Cada uno puede describirse por uno o más datos.

Los objetos se conectan entre sí mediante relaciones. Una relación representa una conexión de objetos y se representa por medio de un rombo. Una situación común es ver múltiples relaciones entre múltiples objetos.

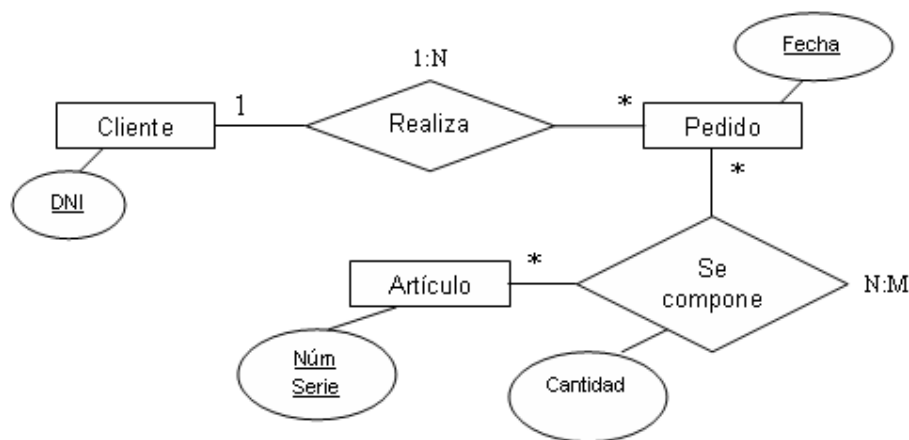


Figura 1.4 Diagrama entidad-relación extendido.

Una notación especial en el diagrama E-R es el indicador asociativo de tipo de objeto; representa algo que funciona como objeto y como relación.

Los tipos de objeto de subtipo/súpertipo consisten en tipos de objeto de una o más subcategorías conectados por una relación.

### 1.5 EL MODELO RELACIONAL

El modelo relacional fue propuesto originariamente por E.F. Codd en un ya famoso artículo de 1970. Gracias a su coherencia y facilidad de uso, el modelo se convirtió en los años 80 en el más usado para la producción de DBMS (*data base management system*).

La estructura fundamental del modelo relacional es precisamente esa: "relación" es decir, una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos). Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad [10].



En este modelo todos los datos son almacenados en relaciones y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene mayor relevancia (a diferencia de otros modelos como el jerárquico y el de red). La información puede ser recuperada o almacenada por medio de «consultas» que ofrecen una amplia flexibilidad y poder para administrar la información [11].

Para distinguir una tupla de otra se recurre al concepto de "llave primaria", o sea a un conjunto de atributos que permiten identificar unívocamente una tupla en una relación. Naturalmente, en una relación puede haber más combinaciones de atributos que permitan identificar unívocamente una tupla ("llaves candidatas"), pero entre éstas se elegirá una sola para utilizar como llave primaria. Los atributos de la llave primaria no pueden asumir el valor nulo (que significa un valor no determinado), en tanto que ya no permitirían identificar una tupla concreta en una relación. Esta propiedad de las relaciones y de sus llaves primarias está bajo el nombre de integridad de las entidades (*entity integrity*) [10].

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales: el álgebra relacional y el cálculo relacional. El álgebra relacional permite describir la forma de realizar una consulta, en cambio, el cálculo relacional sólo indica lo que se desea devolver [11].

Todas las manipulaciones posibles sobre las relaciones se obtienen gracias a la combinación de tan sólo cinco operadores del álgebra relacional: **RESTRICT**, **PROJECT**, **TIMES**, **UNION** y **MINUS**.

Por comodidad, se han definido también tres operadores adicionales que de todos modos se pueden obtener aplicando los cinco fundamentales: **JOIN**, **INTERSECT** y **DIVIDE**. Los operadores relacionales reciben como argumento una relación o un conjunto de relaciones y restituyen una única relación como resultado.

Veamos brevemente estos operadores:

- **RESTRICT:** Restituye una relación que contiene un subconjunto de las tuplas de la relación a la que se aplica. Los atributos se quedan como estaban.
- **PROJECT:** Restituye una relación con un subconjunto de los atributos de la relación a la que viene aplicado. Las tuplas de la relación resultado se componen de las tuplas de la relación original, de manera que siguen siendo un conjunto en sentido matemático.
- **TIME:** Se aplica a dos relaciones y efectúa el producto cartesiano de las tuplas. Cada tupla de la primera relación está concatenada con cada tupla de la segunda.
- **JOIN:** Se concatenan las tuplas de dos relaciones de acuerdo con el valor de un conjunto de sus atributos.
- **UNION:** Aplicando este operador a dos relaciones compatibles, se obtiene una que contiene las tuplas de ambas relaciones. Dos relaciones son compatibles si tienen el mismo número de atributos y los atributos correspondientes en las dos relaciones tienen el mismo dominio.
- **MINUS:** Aplicado a dos relaciones compatibles restituye una tercera que contiene las tuplas que se encuentran sólo en la primera relación.
- **INTERSECT:** Aplicado a dos relaciones compatibles restituye una relación que contiene las tuplas que existen en ambas.



- **DIVIDE:** Aplicado a dos relaciones que tengan atributos comunes, restituye una tercera que contiene todas las tuplas de la primera relación que se puede hacer que correspondan con todos los valores de la segunda relación.

El usuario interactúa con la base de datos a través de una interfaz diferente. El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, el cual es un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales [10].

### **Esquema**

Un esquema es la definición de una estructura (generalmente relaciones o tablas de una base de datos) es decir, determina la identidad de la relación y que tipo de información podrá ser almacenada dentro de ella; en otras palabras son los metadatos de la relación. Todo esquema constará de:

- Nombre de la relación (su identificador).
- Nombre de los atributos (o campos) de la relación y sus dominios; el dominio de un atributo o campo define los valores permitidos para el mismo, es equivalente al tipo de dato por ejemplo *character*, *integer*, *date*, *string*, etc [11].

### **Instancias**

Se puede definir como el contenido de una tabla en un momento dado, pero también es válido referirnos a una instancia cuando trabajamos o mostramos únicamente un subconjunto de la información contenida en una relación o tabla, como por ejemplo:

- Ciertos caracteres y números (una sola columna de una sola fila).
- Algunas o todas las filas con todas o algunas columnas
- Cada fila es una tupla. El número de filas es llamado *cardinalidad*.
- El número de columnas es llamado *aridad o grado* [11].

## ***1.6 BASES DE DATOS RELACIONALES.***

Los sistemas de bases de datos surgieron con el objetivo de resolver los problemas que planteaban los sistemas de archivos. Una base de datos es un conjunto de datos relacionados que recogen las necesidades de información de una empresa u organización. Estos datos se comparten por todos los usuarios y son el tipo de bases de datos actualmente más difundido.

Los motivos de este éxito son fundamentalmente dos [10]:

1. Ofrecen sistemas simples y eficaces para representar y manipular los datos.
2. Se basan en un modelo, el relacional con sólidas bases teóricas.

Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina **ID**, identificador o clave. A esta manera de construir bases de datos se le denomina **modelo relacional**.



Estrictamente hablando el término se refiere a una colección específica de datos pero a menudo se le usa, en forma errónea como sinónimo del software usado para gestionar esa colección de datos. Ese software se conoce como **SGBD** (sistema gestor de base de datos) o el relacional **RDBMS** (del inglés *relational database management system*) [11].

El SGBD es un conjunto de programas que permiten a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a dicha base de datos. La base de datos contiene tanto los datos como su definición. Todos los accesos a la base de datos se realizan a través del SGBD y éste proporciona un lenguaje de definición de datos que permite a los usuarios definir la base de datos, y un lenguaje de manejo de datos que permite a los usuarios la inducción, actualización, eliminación y consulta de datos de la base de datos. El SGBD es el que proporciona un acceso controlado, la seguridad, la integridad, concurrencia entre datos y controla la recuperación ante fallos. Además, proporciona un mecanismo de vistas que permite mostrar a los usuarios sólo aquellos datos que les interesan.

Las personas involucradas en el entorno de una base de datos son: el administrador de la base de datos, los diseñadores de la base de datos, los programadores de aplicaciones y usuarios finales.

Los sistemas de bases de datos representan una serie de ventajas tanto por el hecho de compartir los datos, como por la existencia de SGBD. Algunas de estas ventajas son el control de la redundancia, la consistencia de datos, la mejora en los aspectos de seguridad y la integridad. Algunos de sus inconvenientes son su elevada complejidad, su coste y su vulnerabilidad ante fallos.

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización de una base de datos, el cual es entendido como el proceso necesario para que una base de datos sea utilizada de manera óptima.

### **Formas normales**

Las relaciones que describe Codd [18] al definir su modelo relacional, son transformadas en tablas al momento de trabajar con una base de datos. Las Formas Normales buscan optimizar estas estructuras eliminando básicamente la redundancia utilizando como medio principal las dependencias funcionales. Las Formas Normales son un pequeño número de reglas que de cumplirse hacen que las estructuras posean la menor cantidad de redundancia posible y son:

- **Primera Forma Normal (1NF):** No hay campos múltiples (todos los campos son atómicos). Todas las filas deben tener el mismo número de columnas.
- **Segunda Forma Normal (2NF):** Todo campo que no sea clave debe depender por completo de toda la clave.
- **Tercera Forma Normal (3NF):** No hay dependencias transitivas. Un campo debe depender de la clave y no de otro campo.
- **Forma Normal de Boyce-Codd (BCNF):** Todos los determinantes de la tabla son clave candidata.
- **Cuarta Forma Normal (4NF):** Una fila no debe contener dos o más campos multivalorados (aquellos que pueden contener más de un valor simultáneamente) sobre una entidad.



- **Quinta Forma Normal (5NF):** Una tabla puede almacenar atributos dependientes a la clave **sólo** por unión.
- **Reglas de Codd.**

### **1.7 MODELO CLIENTE SERVIDOR.**

Esta arquitectura consiste básicamente en que un programa –el cliente- realiza peticiones a otro programa -el servidor- que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debido a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La **arquitectura cliente-servidor** sustituye a la arquitectura **monolítica** en la que no hay distribución, tanto a nivel físico como a nivel lógico [14].

#### **Características del modelo Cliente-Servidor (C/S).**

En la arquitectura C/S el **remite de una solicitud** es conocido como cliente y el **receptor de la solicitud** enviada por cliente se conoce como servidor.

Características de un cliente:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo **maestro o amo**).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Características de un servidor:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo **esclavo**).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.



- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

TCP es un protocolo orientado a conexión [15]. No hay relaciones maestro/esclavo. Las aplicaciones sin embargo, utilizan un modelo C/S en las comunicaciones. Algunos servidores esperan las solicitudes en puertos bien conocidos de modo que sus clientes saben a qué zócalo IP deben dirigir sus peticiones. El cliente emplea un puerto arbitrario para comunicarse. Los clientes que se quieren comunicar con un servidor que no usa un puerto bien conocido tienen otro mecanismo para saber a qué puerto dirigirse. Este mecanismo podría usar un servicio de registro como Portmap, que utiliza un puerto bien conocido.

### **Comparación de la arquitectura C/S con otras arquitecturas de red**

Las **redes de pares**, también conocidas como redes **par-a-par** o **peer-to-peer** (abreviado con las siglas **P2P**) son otro tipo de arquitectura de red; en ellas cada nodo o elemento del sistema puede actuar al mismo tiempo como cliente y como servidor; cada nodo tiene, por tanto, las responsabilidades y estados de ambos elementos.

Si bien la clásica arquitectura C/S requiere uno de los puntos terminales de comunicación para actuar como un servidor que puede ser algo más difícil de aplicar, la arquitectura Cliente-Cola-Cliente habilita a todos los nodos para actuar como clientes simples, mientras que el servidor actúa como una cola que va capturando las peticiones de los clientes (un proceso que debe pasar sus peticiones a otro, lo hace a través de una cola, por ejemplo, una consulta a una base de datos, entonces, el segundo proceso conecta con la base de datos, elabora la petición, la pasa a la base de datos, etc.). Esta arquitectura permite simplificar en gran medida la implementación de software. La arquitectura **P2P** originalmente se basó en el concepto **Cliente-Cola-Cliente** [14].

### **Ventajas**

- **Centralización del control:** Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- **Escalabilidad:** Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento:** Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- **Existen tecnologías suficientemente desarrolladas y diseñadas para el paradigma de C/S** que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.
- Una ventaja adicional del uso del esquema Cliente/Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones pues se pueden emplear las herramien-



tas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets o el RPC).

### **Desventajas**

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuantos más nodos hay, mejor es el ancho de banda se tiene.
- El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está *caído*, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal no puede servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor para satisfacer el trabajo. Por supuesto, esto aumentará el coste.
- La seguridad de un esquema Cliente/Servidor es otra preocupación importante. En este caso los mecanismos son distintos que en el caso de los sistemas centralizados. Por ejemplo, se deben hacer verificaciones en el cliente y en el servidor. También se puede recurrir a otras técnicas como el encriptamiento.

### **Tipos de Cliente/Servidor**

Los sistemas Cliente/Servidor pueden ser de muchos tipos, dependiendo de las aplicaciones que el servidor pone a disposición de los clientes [17]. Existen como por ejemplo:

- Servidores de Impresión, mediante el cual los usuarios comparten impresoras.
- Servidores de Archivos, con el cual los clientes comparten discos duros.
- Servidores de Bases de Datos, donde existe una única base de datos.
- Servidores de Lotus Notes, que permite el trabajo simultáneo de distintos clientes con los mismos datos, documentos o modelos.
- Los Servidores Web también utilizan la tecnología Cliente-Servidor aunque añaden aspectos nuevos y propios a la misma.



# *CAPÍTULO*

# 2

## *ASPECTOS DE ESPECIFICACIÓN DEL SISTEMA.*

### *INTRODUCCIÓN*

- 2.1 DEFINICIONES FORMALES.*
- 2.2 ANTECEDENTES.*
- 2.3 MOTIVACIONES.*
- 2.4 OBJETIVOS.*
- 2.5 REQUERIMIENTOS DEL SISTEMA (RECURSOS DEL PROYECTO).*
- 2.6 RESULTADOS ESPERADOS.*
- 2.7 DESCRIPCIÓN DE INTERFACES CON OTROS SISTEMAS.*



# CAPÍTULO 2

## ASPECTOS DE ESPECIFICACIÓN DEL SISTEMA

### *INTRODUCCIÓN.*

El siguiente capítulo tiene como propósito especificar los requerimientos que utiliza el sistema, es decir, describir de manera general lo que realiza el sistema; se propone que de manera sencilla se pueda comprender el ámbito del trabajo a realizar, los riesgos en los que se pueda incurrir, los recursos requeridos, las tareas a llevar a cabo y los pasos que hay que recorrer.

También se hace una explicación breve de lo que es el mundo de las bibliotecas, sus definiciones y la problemática actual.

### *2.1 DEFINICIONES FORMALES.*

#### **Biblioteca**

La voz **biblioteca** puede traducirse desde un punto de vista estrictamente etimológico como el lugar donde se guardan los libros. En la actualidad esta concepción se ha visto hace tiempo superada para pasar a referirse tanto a las colecciones bibliográficas como a las instituciones que las crean y las ponen en servicio para satisfacer las demandas de los usuarios [4].

Según la norma UNE 50113-1:1992<sup>1</sup> sobre conceptos básicos de información y documentación, el término biblioteca puede definirse en dos sentidos [4]:

1. Cualquier colección organizada de libros y publicaciones en serie impresos u otros tipos de documentos gráficos o audiovisuales disponibles para el préstamo o consulta.
2. Organismo o parte de él cuya principal función consiste en construir bibliotecas, mantenerlas, actualizarlas y facilitar el uso de los documentos que precisen los usuarios para satisfacer sus necesidades de información, investigación, educativas o de esparcimiento, contando para ello con un personal especializado [4].

#### **Biblioteca digital**

Una **biblioteca digital** es una biblioteca en que una proporción significativa de los recursos de información se encuentran disponibles en el formato digital (pdf, doc, etc. o micro forma) accesible por medio de las computadoras. El volumen digital se sostiene localmente; si es posible acceder a la biblioteca de forma remota vía Internet, se trata de una biblioteca virtual. En estas bibliotecas, el proceso de digitalización empezó con el catálogo, continuó con los índi-

---

<sup>1</sup> Vocabulario

UNE 50113-1:1992.- Documentación e información. Vocabulario. Parte 1: Conceptos fundamentales



ces de la revista y los servicios de resumen y finalmente la publicación de los ebook o libros electrónicos [5]. Algunas de las bibliotecas digitales en español más grandes y más exitosas son Biblioteca Virtual Miguel de Cervantes, la Biblioteca Virtual de la UNMSM, la Biblioteca Digital de la UCM o la de la Fundación Sancho El Sabio.

Una biblioteca digital es un sistema de tratamiento técnico de acceso y transferencia de información digital estructurado alrededor del ciclo de vida de una colección de documentos digitales sobre los cuales se ofrecen servicios interactivos de valor añadido para el usuario final

La digitalización de libros, bases de datos y publicaciones es una labor titánica que está siendo emprendida por las grandes ventajas que ofrece entre las que podemos mencionar:

- Ahorro de papel.
- Disminución de la necesidad de espacios en las bibliotecas.
- Crecimiento y mejor organización de los acervos.
- Optimización de los mecanismos de búsqueda de textos, imágenes, videos y audio.
- Facultad de acceder a información desde cualquier parte del mundo e igualmente compartirla.
- La información en algunos casos se puede imprimir, grabar, enviar por correo electrónico e incluso manipular.

Estas características las convierten en un gran recurso para científicos, profesores y alumnos principalmente [6].

### **Desventajas de las bibliotecas digitales**

- Al tener las obras en formatos digitales y ser tan sencilla su distribución se corre el riesgo de que la piratería se haga presente y con ello los autores carezcan de los beneficios que, por derecho les corresponden.

Para limitar el duplicado indebido de la información se han definido algunas restricciones como:

- Se implementan páginas que no permiten la copia o impresión de las páginas de los sitios y así el usuario sólo tiene permisos de lectura.
- Los discos compactos que distribuyen las editoriales incluyen llaves que no permiten la copia de la información a disco duro o tienen integrada una clave de registro que autoriza instalar una sola vez el producto.

## ***2.2 ANTECEDENTES.***

### **Situación actual de las bibliotecas**

Es de vital importancia para el desarrollo de un país acercar las fuentes de conocimiento con quienes generan avances tecnológicos, científicos, económicos y políticos.



A pesar de la ruptura de las barreras geográficas y de la globalización que se tiene de la información con el uso de Internet, en México aún se dispone de muy poca información electrónica de fuentes confiables que se ajuste a las necesidades de la educación superior. Y aunque existen muchas bibliotecas con catálogos electrónicos, las bibliotecas digitales con acceso a textos completos se encuentran en una fase inicial.

Las bibliotecas de consulta general son el pilar de la educación de toda la población y las bibliotecas especializadas son apoyo de conocimiento para los estudiantes de nivel superior, docentes e investigadores. Sin embargo las distancias geográficas y las restricciones en el uso de sus servicios impiden un eficiente aprovechamiento de los recursos que ponen a disposición de ellos. Además, en estas épocas cambiantes, con renovaciones políticas, sociales y tecnológicas; las bibliotecas deben ampliar sus responsabilidades y su horizonte laboral: redes, bases de datos, análisis estadístico, evaluación, disseminación y comercialización de la información, nuevos tipos de usuarios, nuevas disciplinas, nuevos formatos de información, imágenes, hipertextos, educación a distancia, telecomunicación, protección de la información, derechos de autor, criptografía y correo electrónico.

Ante estas necesidades, surge la iniciativa de desarrollar servicios, sistemas y redes de cómputo que fomenten el uso de la tecnología de vanguardia, la expansión de la educación y la cultura en beneficio de la sociedad.

El desarrollo de bibliotecas digitales especializadas será un gran paso en este ámbito al abrir una nueva puerta para la adquisición de conocimientos mediante las tecnologías de la información y con los conceptos que han perdurado por siglos en el área bibliotecológica.

### ***2.3 MOTIVACIONES.***

Las motivaciones para la realización de este proyecto son que en los últimos años los maestros que imparten cursos en las universidades o en otro nivel escolar tienen investigaciones en formato electrónico generados por ellos mismos o documentos que son útiles para el curso. Esta documentación electrónica casi siempre la tienen en su página de Internet y de su existencia solo la saben los alumnos que están tomando dicho curso.

El sistema de base de datos para la consulta de los medios electrónicos de una Biblioteca se aplicará para la consulta de este material. El sistema tendrá una cantidad importante de documentos, se propone ordenar esta documentación en la base de datos para que toda persona pueda ingresar a ésta. Cada profesor tendrá su clave de acceso al sistema e ingresará un nuevo documento o actualizará uno ya existente. El administrador de la base de datos (BDA) tendrá la facultad de insertar, modificar y eliminar del sistema a otros administradores, profesores y cursos.

Los usuarios podrán ingresar a la consulta, teclean el nombre o algunas características del documento que solicitan, si se encuentra la información con características similares, entonces aparecerá un reporte en pantalla sobre lo que se buscó, en caso contrario el usuario podrá hacer más peticiones de búsqueda del material.



## **2.4 OBJETIVOS.**

A continuación se presentan los principales objetivos del sistema de la base de datos para la búsqueda de información electrónica de una biblioteca:

- Proponer que aplicaciones de software se deben desarrollar, especificar sus características formalmente, así como la relación entre ellas.
- Construir un modelo formal y claro que permita crear el sistema propuesto, así como sus requerimientos de desarrollo e implementación.
- Analizar, diseñar e implementar la BD asociada.
- Desarrollar una solución local en una LAN y a nivel Web.

## **2.5 REQUERIMIENTOS DEL SISTEMA (RECURSOS DEL PROYECTO).**

### **Requerimientos del sistema**

A continuación se describe el análisis de requerimientos del sistema:

1. El sistema será manejado por usuarios permitidos entre los que se encuentran: el administrador y los profesores que contarán con una clave de acceso, así podrán manejar todos o algunos de los módulos del sistema claro está dependiendo de los privilegios que cada uno tenga y por último los alumnos que podrán manejarlo sin necesidad de estar dados de alta.
2. Al ingresar al sistema como administrador, tendrá la facultad de hacer uso de la información del personal académico; los profesores tendrán acceso a modificar la información de sus documentos o insertar nuevos documentos a la base de datos. De lo anterior surgen dos entidades que serán PROFESOR Y DOCUMENTO.
3. Para la creación de passwords, privilegios y estados de actividad de los profesores se creará una tabla ligada a la tabla profesor llamada ID.
4. Una vez que tenemos la tabla donde insertaremos la información de cada documento debemos hacer mención del curso o área a la que pertenecen, por esto se guardarán en una tabla llamada CURSO; a su vez este curso pertenece a una facultad que por el momento sólo será la de Ciencias de la Computación que es donde será instalada la aplicación, por lo tanto crearemos la tabla FACULTAD.
5. Los alumnos o usuarios que no estén registrados en el sistema pueden hacer uso de este y podrán realizar búsquedas específicas de documentos mediante varios filtros de búsqueda.
6. Hacer la descarga o visualización de los documentos específicos encontrados en su formato original.
7. Hacer reportes acerca de las diferentes fuentes de información que le convenga al área administrativa y demás áreas. Se harán consultas a la base de datos que podrán ser visualizadas en pantalla.
8. Es necesario mencionar que el sistema será una aplicación bajo el modelo Cliente-Servidor, ya que por un lado se encontrarán los administradores, profesores y visitas que realizarán peticiones y por el otro lado se encontrará el servi-



dor que proporcionará los servicios al cliente y enviará sus resultados. En el servidor alojará la base de datos que contendrá los documentos disponibles para descarga.

Para un mejor desempeño del sistema se requerirá del siguiente software y hardware tanto para el servidor como para los clientes:

#### **Requerimientos del Software:**

- Se tiene pensado utilizar interprete PHP<sup>2</sup>, un servidor común tal como Apache<sup>3</sup> y un Sistema Administrador de base de datos como MySQL<sup>4</sup> para trabajar sobre una plataforma Windows o Linux y si hubiese necesidad de transportar a otra plataforma este es el que se adaptaría de manera más sencilla, ya que es útil para cualquier plataforma y cualquier terminal.

#### **Requerimientos del hardware:**

- La computadora para el desarrollo del software será cualquiera a partir de una Pentium I con un disco duro de hasta 30 GB, memoria en RAM con 256 MB que cuente con una tarjeta de red para la conexión a una LAN y sobre el Web. El servidor que se encuentra en la FCC con nombre bd1.cs.buap.mx del laboratorio de investigación.
- Para el cliente se necesitará una computadora personal con procesador Pentium III o superior, 128 Mb (Mega Bytes) de memoria RAM como mínimo, tarjeta de red u otro dispositivo (modem, cable módem, tarjeta inalámbrica) que permita una conexión a Internet y un monitor VGA configurado con una resolución mínima de 800x600 píxeles.
- Para el servidor se solicita una computadora con un disco duro de 60 Gb (Giga Bytes) como mínimo, procesador Pentium IV o superior, 256 Mb (Mega Bytes) de memoria RAM y una tarjeta de red ethernet o inalámbrica configurada apropiadamente.

## ***2.6 RESULTADOS ESPERADOS.***

Se espera que al final del desarrollo del software se tenga todo concentrado en un solo lugar para que así las búsquedas se realicen con rapidez, eficacia y no se pierda mucho tiempo en buscar la información en lugares dispersos. También se desea que este sistema sea integrado a la base de datos de la biblioteca de la FCC y así se pueda utilizar en la red de área local de la Facultad de Ciencias de la Computación (FCC). Por medio de Internet para que desde cualquier computadora se pueda ingresar al servidor y buscar información específica de algún curso impartido por los profesores sin necesidad de estar en la misma facultad.

Que el DBA sea el indicado para dar el mantenimiento, hacer los respaldos de la base de datos, crear o modificar cursos impartidos en la facultad y administrar profesores.

---

<sup>2</sup> Acrónimo recursivo de PHP: **H**ypertext **P**reprocessor, sitio oficial <http://www.php.net/>

<sup>3</sup> <http://www.apache.org/>

<sup>4</sup> <http://www.mysql.com/>



Además, los profesores puedan agregar, modificar o eliminar sus propios documentos electrónicos.

## ***2.7 DESCRIPCIÓN DE INTERFACES CON OTROS SISTEMAS.***

Se hará uso de la base de datos que se tiene para la consulta del acervo electrónico de la biblioteca de la FCC. La siguiente parte de este proyecto es utilizar lo que ya viene funcionando, como por ejemplo: la búsqueda de algún libro electrónico; sólo hay que adicionar la parte de búsqueda del material generado por los profesores.



# *CAPÍTULO*

# *3*

## *DESCRIPCIÓN DE LA INFORMACIÓN.*

### *INTRODUCCIÓN*

*3.1 DISEÑO DE LA BASE DE DATOS.*

*3.2 ANÁLISIS ESTRUCTURADO DEL SISTEMA.*

*3.3 DISEÑO CONCEPTUAL DE LA BASE DE DATOS DEL SISTEMA ITZA.*

*3.4 DISEÑO LÓGICO.*

*3.5 DISEÑO DE LA INTERFAZ DE USUARIO DEL SISTEMA ITZA.*



# CAPÍTULO 3

## DESCRIPCIÓN DE LA INFORMACIÓN

### *INTRODUCCIÓN.*

La razón para preocuparse por el diseño de las bases de datos es que es crucial para la consistencia, integridad y precisión de los datos. Si una base de datos está mal diseñada, los usuarios tendrán dificultades a la hora de acceder a ciertos tipos de información y existe el riesgo añadido de que ciertas búsquedas puedan producir información errónea.

Se pueden hacer malos diseños, pero una base de datos bien diseñada contendrá información correcta, almacenará los datos eficientemente, será más fácil de gestionar y mantener.

En la primera etapa del proceso de creación de la base de datos se especifica el ámbito y los límites de la aplicación de la base de datos, así como con qué otros sistemas interactúan. También hay que determinar quiénes son los usuarios y las áreas de aplicación.

### *3.1 DISEÑO DE LA BASE DE DATOS.*

Esta etapa consta de tres fases: diseño conceptual, diseño lógico y diseño físico. La primera consiste en la producción de un esquema conceptual, que es independiente de todas las consideraciones físicas. Este modelo se refina después de un esquema lógico eliminando las construcciones que no se pueden representar en el modelo de la base de datos escogido (relacional, orientado a objetos, etc.) En la tercera fase, el esquema lógico se traduce en un esquema físico para el SGBD elegido. La fase del diseño físico considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente a la base de datos en memoria secundaria.

#### **Diseño Conceptual**

En esta etapa se debe construir un esquema de la información que se usa en la empresa, independientemente de cualquier consideración física. A este esquema se le denomina *esquema conceptual*.

El esquema conceptual se puede utilizar para que el diseñador transmita a la empresa lo que ha entendido sobre la información que esta maneja. Para ello, ambas partes deben estar familiarizadas con la notación utilizada en el esquema. La más popular es la notación del modelo **entidad-relación**.

El esquema conceptual se construye utilizando la información que se encuentra en la especificación de los requisitos del usuario. El diseño conceptual es completamente independiente de los aspectos de implementación, como puede ser el SGBD que se vaya a usar, los



programas de aplicación, los lenguajes de programación, hardware disponible o cualquier otra consideración física. Durante todo el proceso de desarrollo del esquema conceptual, éste se prueba y se valida con los requisitos de los usuarios. El esquema conceptual es una fuente de información para el diseño lógico de la base de datos.

### **Diseño Lógico**

Es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo de base de datos específico, independiente del SGBD concreto que se vaya a utilizar y de cualquier otra consideración física.

En esta etapa se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar, como puede ser el modelo relacional, el modelo de red, el modelo jerárquico o el modelo orientado a objetos.

La **normalización** es una técnica que se utiliza para comprobar la validez de los esquemas lógicos basados en el modelo relacional, ya que asegura que las relaciones (tablas) obtenidas no tienen datos redundantes.

El esquema lógico es una fuente de información para el diseño físico. Además juega un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos se representen correctamente en la base de datos.

### **Diseño físico**

Es el proceso de producir la descripción de la implementación de la base de datos en memoria secundaria: estructuras de almacenamiento y métodos de acceso que garanticen un acceso eficiente a los datos.

Para llevar a cabo esta etapa, se debe haber decidido cuál es el SGBD que se va a utilizar, y que el esquema físico se adapte a él. Entre el diseño físico y el lógico hay una realimentación, ya que algunas de las decisiones que se tomen durante el diseño físico para mejorar las prestaciones pueden afectar la estructura del esquema lógico [2].

## ***3.2 ANALISIS ESTRUCTURADO DEL SISTEMA ITZA.***

### ***3.2.1 Diagrama de flujo de datos (DFD).***

Los DFD son un modelo lógico-gráfico para representar el funcionamiento de un sistema en un proyecto de software. Sus elementos son: proceso (círculos), flujo de datos (flechas) y almacenes (rectángulos abiertos) [21].

A continuación se describen los DFD:

### Diagrama de Contexto o Nivel 0.

Representa el nivel más alto del sistema, muestra solo entradas, salidas y comunicación con el mundo externo.

En el siguiente diagrama de flujo de datos (Fig. 3.1) se muestra de manera general el funcionamiento del sistema para el manejo de información electrónica de una biblioteca el cual cuenta con tres tipos de entradas que indican las entidades externas que interactúan con el sistema y siempre una salida que será el resultado después de que la información de entrada haya sido procesada.



Fig. 3.1 Diagrama de contexto del sistema de base de datos ITZA.

**ADMINISTRADOR.** Persona que tendrá la capacidad de dar mantenimiento a la base de datos, ingresará, modificará y dará de baja a los profesores o cursos. Cuenta con un nombre de usuario (*login*) y contraseña (*password*) que lo autenticarán dentro de la base de datos para permitirle realizar las tareas antes mencionadas.

**PROFESOR.** Es la persona registrada que cuenta con un *login* y *password*, los cuales el administrador dio de alta previamente en la base de datos. En su nivel de usuario podrá ingresar documentos y realizar las modificaciones convenientes sólo a los documentos que él mismo haya ingresado.

**USUARIO (INVITADO).** Es la persona externa al sistema, requiere de una petición de búsqueda para ingresar SOLO A LA CONSULTA de documentos en la base de datos.

Del otro lado se encuentra la salida que es el resultado de las peticiones que hiciera cualquier tipo de entrada (administrador, profesor o usuario), se generará un reporte que será el documento electrónico que el profesor haya ingresado o alguna salida visual de cualquier cambio generado en la base de datos (consulta y/o modificación de usuarios o mantenimiento de documentos electrónicos a la base de datos, etc.).



## Nivel 0.

En este (3.2) y los siguientes niveles se muestra en forma detallada el funcionamiento del sistema para el manejo de información electrónica de una biblioteca:

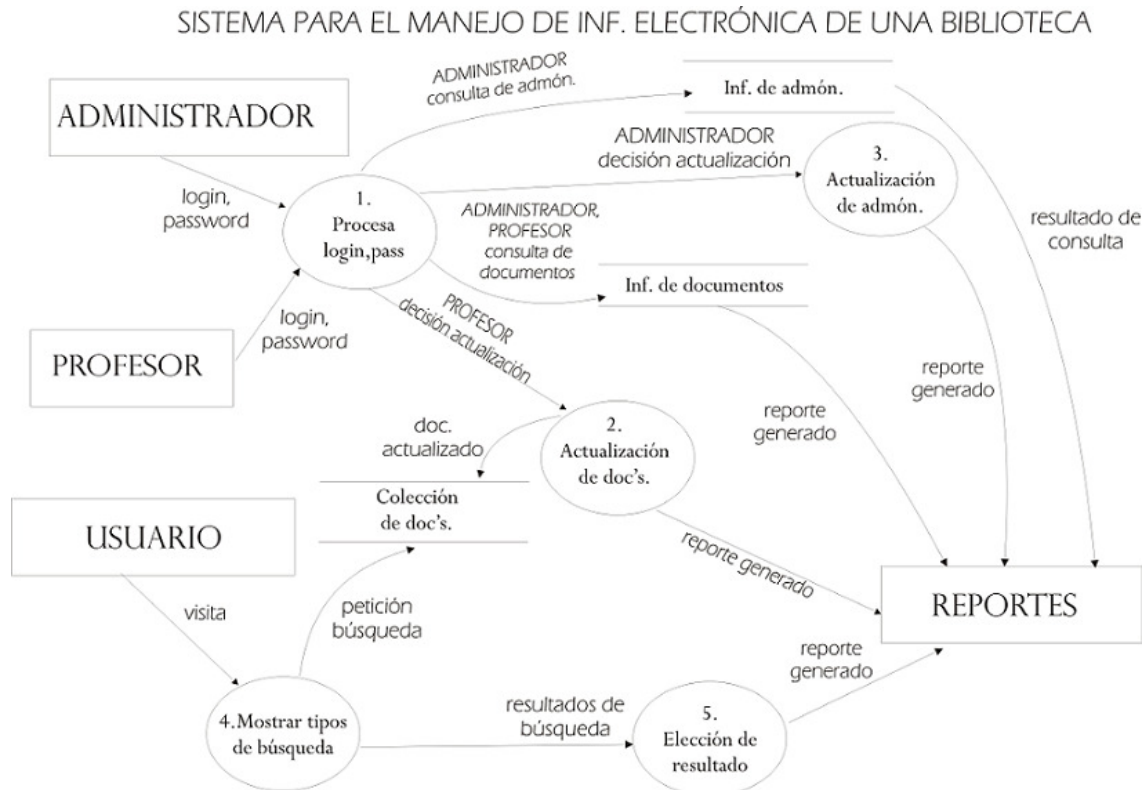


Fig. 3.2 DFD nivel 0 del sistema de base de datos ITzA.

El administrador y/o profesor podrán ingresar al sistema con un *login* y *password* que el administrador previamente dio de alta en el sistema, de ahí el administrador podrá buscar en la base de datos mediante una *consulta de administración* o modificar la base de datos en cuanto a usuarios o cursos, mediante la *decisión actualización* que lo llevará hacia el proceso **ACTUALIZACIÓN DE ADMINISTRACIÓN**. En su defecto, si opta por ir a una *consulta de documentos*, sólo podrá hacer la revisión de los documentos ingresados en la base de datos. Si verificando el *login* y *password* de quien ingresó al sistema es un profesor, entonces éste podrá seguir hacia la decisión **ACTUALIZACIÓN DE DOCUMENTOS**. Ahí podrá actualizar los registros de sus propios documentos o podrá realizar la consulta de cualquier documento.

En caso de que quien visita el sistema es un “usuario externo o usuario visita”, éste solo podrá realizar consultas mediante una *petición de búsqueda* después de que el sistema haya mostrado los **TIPOS DE BÚSQUEDA** existentes. Posteriormente llegará al proceso de **ELECCIÓN DE RESULTADO** donde se mostrarán todos los resultados de búsquedas de los cuales el “usuario visita” haya hecho petición.

En cualquiera de los tres casos de entrada al sistema todos llegarán a un **REPORTE** que es una vista de usuario, ya sea de actualización de usuarios, cursos, documentos, peticiones



de búsquedas; mostrándose los resultados de los cambios o consultas que haya recibido la base de datos, por otra parte se podrán visualizar los archivos creados por los profesores en su formato electrónico original.

### Descripción de la primitiva 1. Procesa login, password.

En el siguiente diagrama (Fig. 3.3) se verifica si la persona que desea ingresar al sistema está registrada en la base de datos y las operaciones que puede realizar.

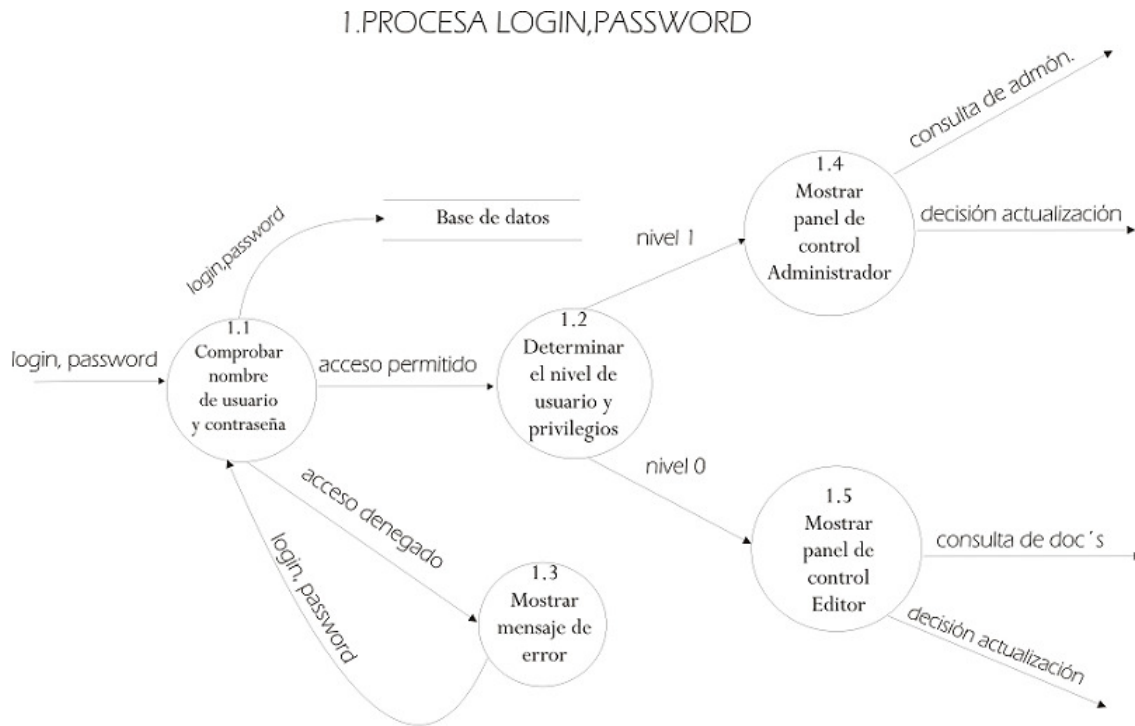


Fig. 3.3 Procesa login, password.

Primero se deberá comprobar que el nombre de usuario y contraseña se encuentren registrados en la base de datos con el proceso **COMPROBAR NOMBRE DE USUARIO Y CONTRASEÑA**, si existen entonces se permitirá el acceso y se determinará el nivel de usuario con el proceso **DETERMINAR EL NIVEL DE USUARIO Y PRIVILEGIOS**. Dependiendo de la función que cada usuario tenga: si es administrador su nivel será 1 y se mostrará el **PANEL DE CONTROL** que le corresponde como Administrador, ahí podrá realizar las tareas correspondientes, como son: consultas de administración, actualización de los registros de la base de datos (insertar, modificar o desactivar registros; si el usuario que ingresó su *login* y *password* está registrado como profesor, su nivel de usuario será 0 y se mostrará el **PANEL DE CONTROL DE EDITOR** en el que contendrá las funciones para consultar documentos de la base de datos, insertar nuevos documentos, modificar o eliminar sus propios documentos.

Si al comprobar el nombre de *login* y *password* en el sistema no se encuentran registrados en la base de datos inmediatamente se envía un mensaje de error negándole el acceso, por



lo que deberá ingresar los datos nuevamente para verificar su existencia y así poder pasar al siguiente nivel del sistema.

### Descripción de la primitiva 2. Actualización de documentos.

En este diagrama (Fig. 3.4) se muestran los subprocesos del proceso ACTUALIZACIÓN DE DOCUMENTOS.

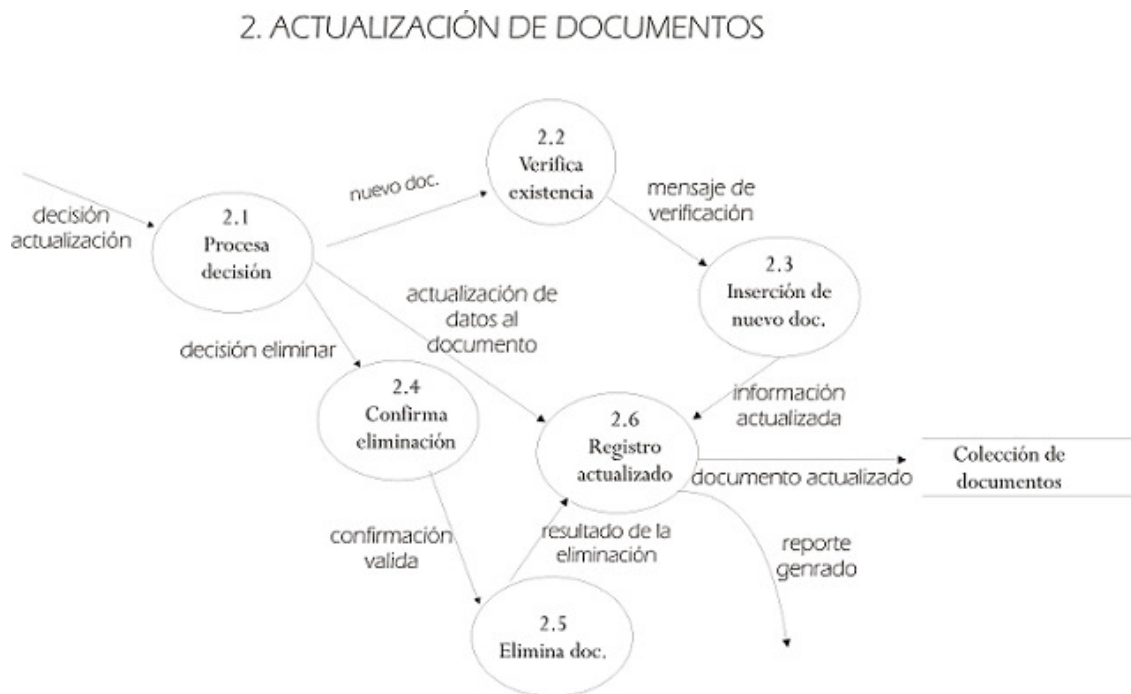


Fig. 3.4 Actualización de documentos.

Si el profesor entra a este proceso de actualización, lo primero que se verificará es la decisión de actualizar PROCESA DECISIÓN que puede ser: *ingresar un nuevo documento* en el que se VERIFICA EXISTENCIA del documento, enviándose un mensaje de validación. Si este mensaje muestra que el documento no se encuentra en la base de datos por lo tanto entrará a INSERCIÓN DEL NUEVO DOCUMENTO. *Actualización de datos al documento* en el cual solamente se hará la modificación o actualización y la *decisión de eliminar* que realizará el proceso CONFIRMA ELIMINACIÓN y una vez confirmada se procederá a eliminar el documento de la base de datos. Todos estos subprocesos nos llevan al último REGISTRO ACTUALIZADO en que el registro se actualiza y finalmente nos da el resultado de la base de datos actualizada.

### Descripción de la primitiva 3. Actualización administración.

En este diagrama (Fig. 3.5) se muestran los subprocesos del proceso ACTUALIZACIÓN ADMINISTRACIÓN el cual actualiza los datos de administración como pueden ser alta, baja, modificación de administradores, profesores, cursos, etc.

### 3. ACTUALIZACIÓN DE ADMINISTRACIÓN (Alta, baja, modificación de profesores, cursos, etc.)

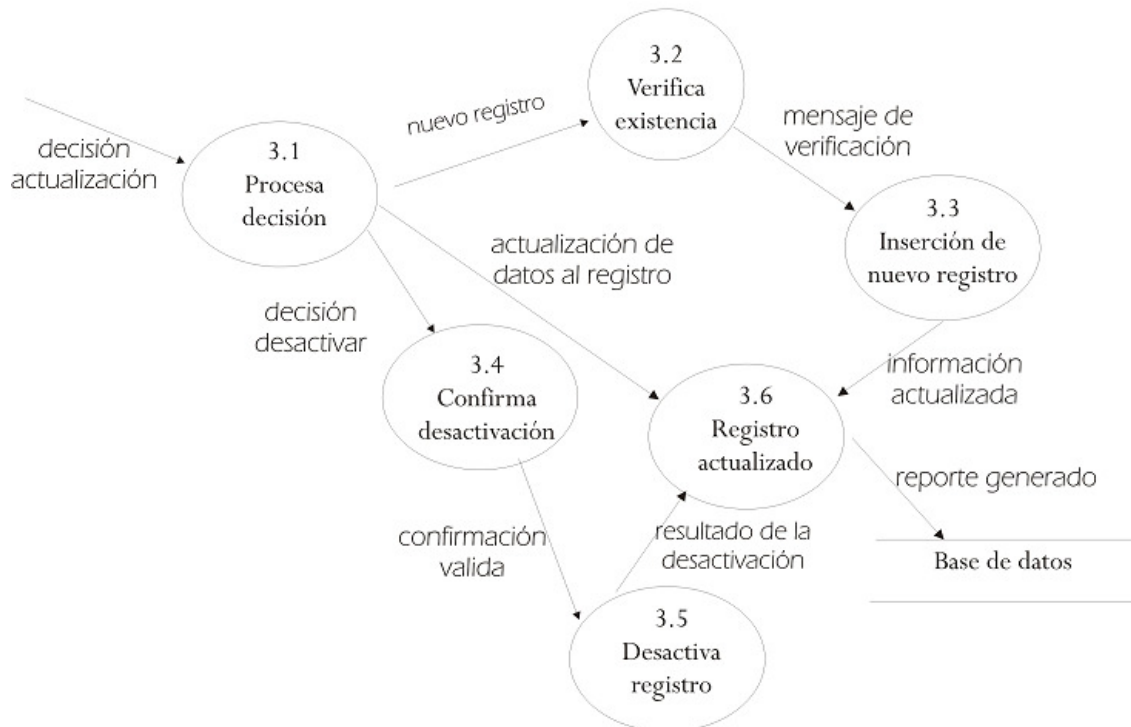


Fig. 3.5 Actualización de administración.

Dependiendo de la decisión actualizar entra al subproceso: **PROCESA DECISIÓN**. El resultado de este procesamiento podría ser: *nuevo registro* en el que **VERIFICA EXISTENCIA** del registro, enviándose un mensaje de verificación, si éste muestra que el nuevo registro no se encontró en la base de datos, entonces pasará a **INSERCIÓN DEL NUEVO REGISTRO**; *actualización de datos al registro* en el que solamente se hará la modificación o actualización; y la *decisión de desactivar* que entrará a **CONFIRMA DESACTIVACIÓN** y una vez confirmada se procederá a desactivar el registro de la base de datos (*nota: ningún registro no se elimina de la base datos por las consultas que se necesiten realizar posteriormente*). Todos estos subprocesos nos llevan al último **REGISTRO ACTUALIZADO** en el que el registro se actualizará y finalmente base de datos guardará los cambios.

#### Descripción de la primitiva 4. Mostrar tipos de búsqueda.

En el siguiente diagrama de flujo de datos (Fig. 3.6) se muestra la descripción detallada de los subprocesos: **MOSTRAR TIPOS DE BÚSQUEDA**.

Aquí la entrada será un **USUARIO VISITA** que deseará consultar algún documento(s) de la base de datos. El sistema mostrará distintos filtros de búsqueda que dependiendo de la información a consultar el **USUARIO VISITA** obtendrá el proceso resultante correspondiente.



Fig. 3.6 Mostrar tipos de búsqueda.

Si el USUARIO VISITA tiene el título del documento que está buscando, se efectuará el subproceso BÚSQUEDA POR TITULO que le mostrará los resultados esperados.

Si el USUARIO VISITA ingresa la información del profesor, entonces se realizará el subproceso: BÚSQUEDA POR PROFESOR que mostrará un reporte con la información relacionada con dicho parámetro.

Se cuenta con el subproceso BÚSQUEDA POR FECHA O PERIODO, el cual contendrá todos los documentos que coincidan con la fecha o rango de tiempo deseado.

Finalmente existe el subproceso BÚSQUEDA POR CÓDIGO O MATERIA en el que el USUARIO VISITA realiza la búsqueda por código o materia del que pertenece el documento buscado.

#### Descripción de la primitiva 5. Elección del resultado.

En este diagrama (Fig. 3.7) se muestra la ELECCIÓN DEL RESULTADO *del usuario invitado*.



## 5.ELECCIÓN DE RESULTADO

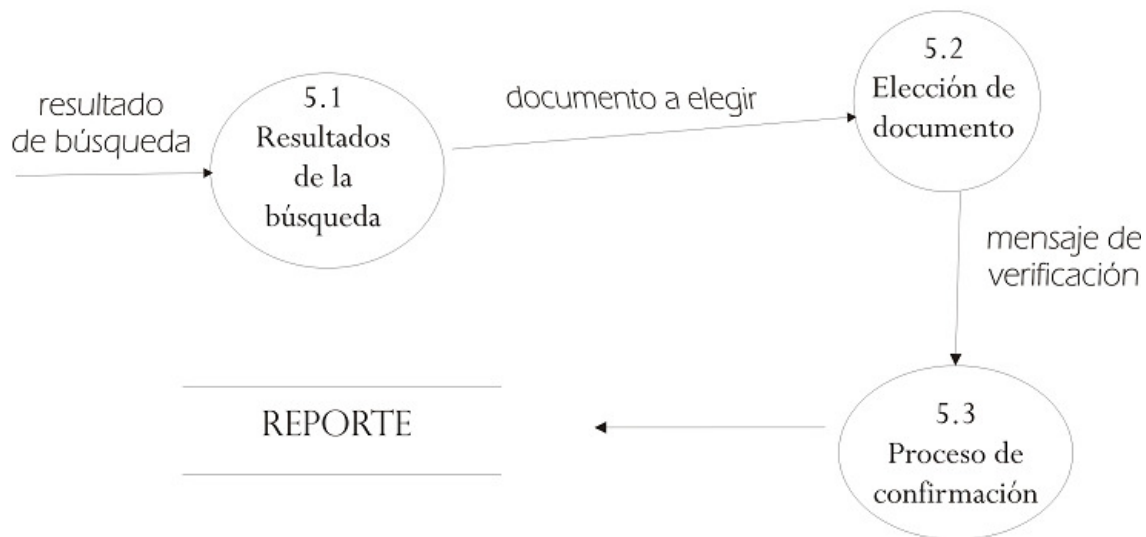


Fig. 3.7 Elección de resultado.

Se mostrará en pantalla la relación de documentos con algún dato en común, que dará como resultado de la búsqueda basada en la petición solicitada. Se podrá elegir el documento preciso que se buscaba realizando la pregunta de confirmación y finalmente se visualizará el archivo deseado.

### 3.2.2 Diccionario de datos.

Un **diccionario de datos** es un conjunto de metadatos que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización [23].

A continuación se presenta el diccionario de datos del sistema el manejo de información electrónica de una biblioteca:

NOMBRE	DESCRIPCIÓN
Inf. admón.	{Inf_profesor + Inf_doc + Cantidad_doc + Cantidad_prof + Código/Materia}
Colección_doc	{Documentos}
Login	[Parte_alfanumérica Parte_numérica]
Password	*Parte_alfanumérica o parte _numérica cifradas*
Consulta de admón.	*Petición de búsqueda de Inf. admón*
Consulta de documentos	{Petición de búsqueda}
Resultados de búsqueda	*Información de documentos con características similares entre sí*
Resultado de consulta	*Información de los registros de la base de datos con características similares entre sí*
Visita	*Persona externa al sistema que realiza búsquedas de documen-



	tos*
<b>Petición _ búsqueda</b>	[Título   Inf_profesor   Fecha/Periodo   Código/Materia]
<b>Reporte generado</b>	*Resultados de los cambios o consultas que haya recibido la base de datos, también se podrán visualizar los archivos creados por los profesores en su formato original*
<b>Doc actualizado</b>	Información actualizada que fue generada por un profesor*
<b>Decisión _ actualización</b>	[Nuevo_doc   Actualización de datos al doc   Decisión eliminar   Nuevo registro   Actualización de datos al registro   Decisión desactivar]
<b>Nuevo_doc</b>	*Nueva información generada por un profesor*
<b>Mensaje de verificación</b>	[Verdadero   Falso]
<b>Inf actualizada</b>	*Actualización de Inf. admón.*
<b>Actualización de datos al doc.</b>	* Modificación de información a un documento existente *
<b>Decisión eliminar</b>	[Aceptación   Cancelación]
<b>Confirmación válida</b>	[Aceptación]
<b>Resultado de la eliminación</b>	*Documento eliminado de la base de datos*
<b>Documento actualizado</b>	*Documento guardado en la base de datos con información nueva o actualizada, o en su defecto un documento eliminado*
<b>Nuevo registro</b>	*Alta de un nuevo profesor o curso*
<b>Actualización de datos al registro</b>	*Modificación de un registro existente*
<b>Decisión desactivar</b>	[Aceptación   Cancelación]
<b>Resultado de la desactivación</b>	*Registro desactivado, dado de baja temporal o definitivamente en el sistema*
<b>Título</b>	*Nombre de documento generado por un profesor*
<b>Inf_profesor</b>	Nombre_profesor + Primer_apellido + Segundo_apellido + Grado _ académico + (Dirección) + (Teléfono) + (Email)
<b>Inf_doc</b>	Título + Inf_profesor + Fecha + Código + Formato
<b>Fecha/Periodo</b>	*Fecha o periodo en que fue guardado el documento a la base de datos*
<b>Curso/Código/Materia</b>	*Clave o nombre del curso al que pertenece el documento generado*
<b>Formato</b>	*Especificación para almacenar datos digitales*
<b>Doc. a elegir</b>	*Documento buscado y encontrado en la base de datos*
<b>Acceso permitido</b>	*Permiso del sistema para ingresar al sistema con determinados privilegios*
<b>Acceso denegado</b>	*Negación del permiso para ingresar al sistema*
<b>Nivel 1</b>	*Definición de los privilegios del Administrador*
<b>Nivel 0</b>	*Definición de los privilegios del Editor*
<b>Doc</b>	* Información generada por el profesor *
<b>Datos_doc/Inf_doc</b>	Título + Datos_profesor + Curso + Fecha
<b>Cantidad_doc</b>	*Número de documentos generados por profesores y almacenados en la base de datos*
<b>Cantidad_prof</b>	*Número de profesores dados de alta en la base de datos*
<b>Datos_curso/Inf_curso</b>	*Nombre del curso y fecha*
<b>Nombre_profesor</b>	1{letra}20
<b>Primer_apellido</b>	1{letra}20
<b>Segundo_apellido</b>	1{letra}20
<b>Grado _ académico</b>	[Lic   Mto.   Dr.]
<b>Letra</b>	[a   ...   z   A   ...   Z]
<b>Parte numérica</b>	{0 ... 9}
<b>Parte alfanumérica</b>	{Letra   Parte _ numérica}



### **3.2.3 Mini-especificaciones.**

Debe haber mini-especificaciones por cada primitiva funcional.

#### **Comprobar nombre de usuario y contraseña.**

Escribe nombre y contraseña para ingresar al sistema.

Si es correcta.

Ingresa a la base de datos para iniciar tareas predeterminadas.

#### **Mostrar mensaje de error.**

Si el nombre de usuario y contraseña son incorrectos.

Envía mensaje de error de contraseña.

#### **Determinar el nivel de usuario y privilegios.**

Determina el nivel del usuario que ingreso al sistema.

Si es 1.

Muestra el panel de control del administrador.

Si es 0.

Muestra el panel de control del editor.

#### **Mostrar panel de control del administrador.**

Muestra en un menú las tareas que puede realizar como administrador.

#### **Mostrar panel de control del editor.**

Muestra en un menú las tareas que puede realizar como editor.

#### **Procesa Decisión.**

Elige una de las decisiones a tomar, ya sea:

Nuevo documento.

Modifica/Actualiza documento.

Elimina documento.

Nuevo registro.

Modifica/Actualiza registro.

Desactiva registro.

#### **Verifica Existencia.**

Si el nuevo documento o registro existe.

Envía mensaje de error.

Si no.

Va a la siguiente burbuja.

#### **Inserción de nuevo documento /nuevo registro.**

Inserta un documento o registro a la base de datos.

#### **Confirma Eliminación.**

Si el documento elegido se quiere eliminar.

Verifica la orden para eliminación.



**Confirma Desactivación.**

Si el registro elegido se quiere desactivar.  
Verifica la orden para desactivación.

**Elimina Doc.**

Elimina el documento elegido y verificado de la base de datos.

**Desactiva Registro.**

Desactiva registro elegido y verificado de la base de datos.

**Registro Actualizado.**

Actualiza la base de datos, ya sea con un nuevo documento o registro, un documento eliminado, un registro desactivado, o un documento o registro modificado.

**Resultados Búsqueda.**

Muestra una lista con las opciones buscadas.

**Elección de documento.**

Elige el documento o registro apropiado.

**Proceso de Confirmación.**

Si doc\_a\_elegir o registro es útil.  
Confirma la acción de elegir el documento o registro para generar el reporte.

**Tipos de Búsqueda.**

Elige un tipo de búsqueda según la información que el usuario proporcione.

**Búsqueda por Título.**

Si el usuario proporciona el título del documento, el sistema realizará búsquedas por título y mostrará las coincidencias.

**Búsqueda por Profesor.**

Si el usuario proporciona la información del profesor que realizó el documento, el sistema realizará búsquedas por profesor y mostrará las coincidencias.

**Búsqueda por Fecha/Periodo.**

Si el usuario proporciona la fecha o periodo de cuando fue creado el documento, el sistema realizará búsquedas por fecha o periodo y mostrará las coincidencias.

**Búsqueda por Código/Materia.**

Si el usuario proporciona el código o materia al que pertenece el documento, el sistema realizará búsquedas por código o materia y mostrará un reporte con resultados.

### 3.3 DISEÑO CONCEPTUAL DE LA BASE DE DATOS DEL SISTEMA ITZA.

#### 3.3.1 MODELO ENTIDAD-RELACIÓN.

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño de bases de datos, está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación solo incluían los conceptos de entidad, relación y atributo. Más tarde se añadieron otros conceptos como: los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido [2].

A continuación se presenta el modelo entidad-relación del sistema de base de datos ITZA (Fig. 3.8).

#### DIAGRAMA ENTIDAD-RELACION EXTENDIDO

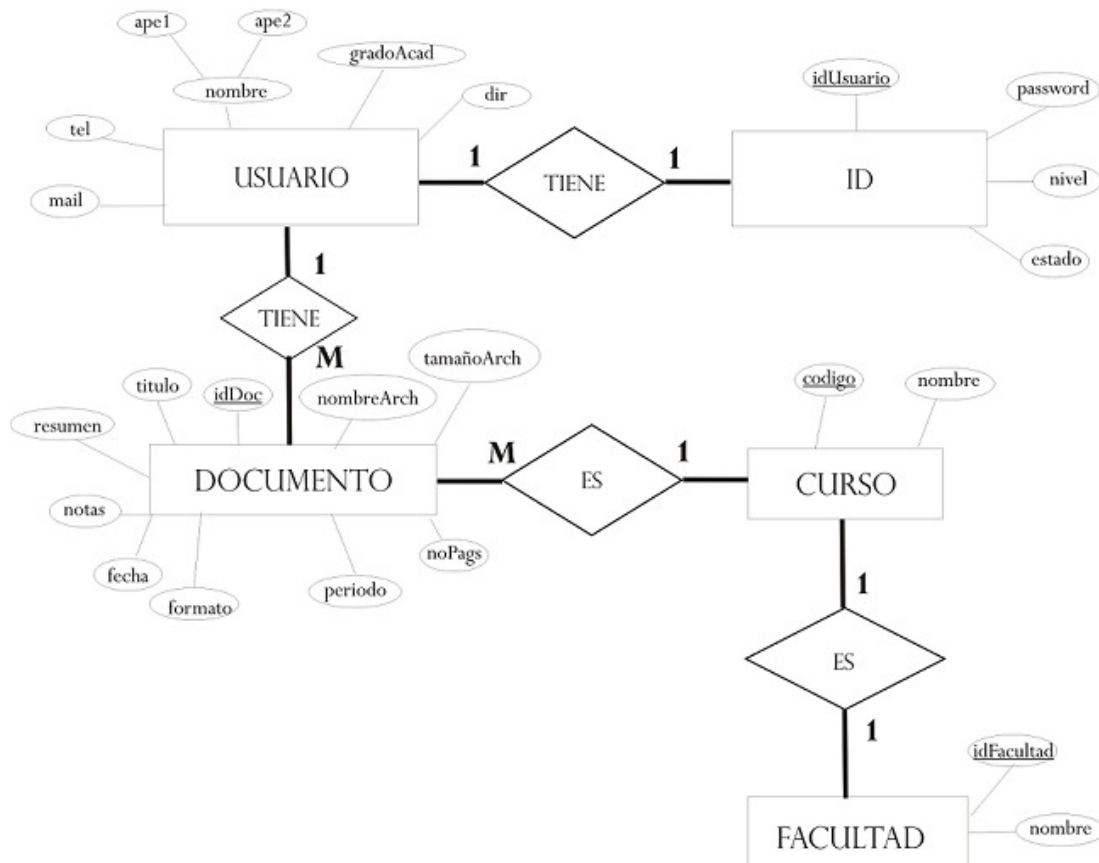


Fig. 3.8 Diagrama E-R del sistema de base de datos ITZA.



### 3.4 DISEÑO LÓGICO.

#### 3.4.1 MODELO RELACIONAL.

Es el modelo en el que se basan la mayoría de los DBMS. Los principios básicos de este modelo son: *la estructura de datos relacional y las reglas de integridad [2]*.

Tres aspectos de los datos:

- Estructura de datos en donde existen las tablas y las relaciones.
- La integridad.
- El manejo de datos.

Existen varios tipos de relaciones que son:

- Base.
- Vistas (virtuales).
- Instantáneas (reales).
- Resultados de consultas.
- Resultados intermedios.
- Resultados temporales.

#### 3.4.2 VALIDACIÓN DE LAS REGLAS DE NORMALIZACIÓN DE LA BASE DE DATOS DEL SISTEMA ITZA.

A continuación se presenta el diseño lógico del sistema, en que se aplicarán: 1ª forma normal donde no se permiten atributos compuestos, multivaluados y relaciones anidadas; la 2ª forma normal donde una tabla se encuentra en 2a NF, si está en 1a 1ª NF y cada atributo que NO es llave es "completamente" dependiente de la llave.

##### Tabla del profesor en su primera forma normal.

En la tabla *profesor* se almacenarán los datos personales de los profesores que realicen documentos para ingresarlos a la base de datos, existen atributos opcionales como son: apellido materno, dirección y número telefónico.

<i>Profesor</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
nomProf		varchar (30)	No	-
Ap1Prof		varchar (30)	No	-
Ap2Prof		varchar (30)	-	-
gradAcad		varchar (20)	No	-
dirProf		varchar (255)	-	-
eMail		varchar (50)	No	-
tel		varchar (30)	-	-



**Tabla de id en su primera forma normal.**

En la tabla *id* se almacenarán los datos de los profesores para la validación al ingreso del sistema, su llave primaria será el *idProf* que será el número de trabajador de la institución. El password será una variable de tipo *varchar* de 32 caracteres ya que se desea que este encriptada para mayor seguridad. Más adelante se explicará detalladamente la idea de la encriptación.

<i>Id</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
<i>idProf</i>	primaria	integer	No	-
<i>pass</i>		varchar (32)	No	-
<i>nivel</i>		integer	No	-
<i>estado</i>		varchar(8)	No	-

**Tabla de los documentos en su primera forma normal.**

En la siguiente tabla se muestran las características que van a tener los documentos en su primera forma normal, la llave será el *idDoc* que auto incrementará cada vez que se ingrese al sistema un nuevo documento; el resumen será una explicación breve de lo que trata el documento y las notas serán algunas instrucciones sobre el documento, estas podrán ser nulas, los demás atributos serán obligatorios.

<i>Documento</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
<i>idDoc</i>	Primaria	integer	no	Si
<i>título</i>		varchar(255)	no	-
<i>resumen</i>		text	-	-
<i>notas</i>		text	-	-
<i>fechaDoc</i>		date	no	-
<i>periodo</i>		varchar(20)	no	-
<i>noPags</i>		integer	no	-
<i>formato</i>		varchar(50)	no	-
<i>nomArch</i>		varchar(100)	no	-
<i>tamañoArch</i>		varchar(20)	no	-

**Tabla del curso en su primera forma normal.**

Esta tabla contendrá el nombre y el código de cada curso, la llave primaria será el código ya que cada curso tiene una clave distinta, y servirá como catálogo que tendrá el nombre de todos los cursos del mapa curricular.

<i>Curso</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
<i>codigo</i>	primaria	varchar(20)	no	-
<i>nomCurso</i>		varchar(50)	no	-



### Tabla de la facultad en su primera forma normal.

La última tabla muestra el nombre de la facultad y su id, por el momento solo será la FCC, posteriormente podrán agregarse algunas otras como el sistema vaya creciendo.

<i>Facultad</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
idFac	primaria	varchar(20)	no	-
nomFac		varchar(50)	no	-

### Tabla del profesor en su segunda forma normal.

La tabla *profesor* tiene una relación **uno a uno** con la tabla *id*, esta tabla contiene todos los datos personales del profesor, no cuenta con una llave primaria, la llave será foránea ya que estas dos tablas tendrán la misma llave que será el número de trabajador.

<i>Profesor</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
idProf	foránea	Integer	no	-
nomProf		varchar (30)	no	-
ap1prof		varchar (30)	no	-
ap2Prof		varchar (30)	-	-
gradAcad		varchar (20)	no	-
dirProf		varchar (255)	-	-
eMail		varchar (50)	no	-
tel		varchar (30)	-	-

### Tabla de id en su segunda forma normal.

En la tabla *id* se almacenarán los datos de los profesores para la validación al ingreso del sistema, su llave primaria será el idProf que será el número de trabajador de la institución. También tiene una relación **uno a muchos** con la tabla *documento* ya que cada profesor puede hacer más de un documento para alimentar la base de datos.

<i>Id</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
idProf	primaria	integer	no	-
pass		varchar (32)	no	-
nivel		varchar (32)	no	-
estado		varchar(8)	no	-

### Tabla de los documentos en su segunda forma normal.

La tabla *documento* contiene las siguientes relaciones con otras tablas como a continuación se describe:

- Relación **uno a muchos** de la tabla *id* a la tabla *documento*.
- Relación **uno a muchos** de la tabla *curso* a la tabla *documento*.



Después de haber sido normalizada en su segunda forma normal tendremos la siguiente tabla, con las llaves foráneas correspondientes.

<i>Documento</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
idDoc	primaria	integer	no	Si
idProf	foránea	integer	no	-
código	foránea	varchar(20)	no	-
título		varchar(255)	no	-
resumen		text	-	-
notas		text	-	-
fecha		date	no	-
periodo		varchar(20)	no	-
noPags		integer	no	-
formato		varchar(50)	no	-
nomArch		varchar(100)	no	-
tamañoArch		varchar(20)	no	-

#### Tabla del curso en su segunda forma normal.

Esta tabla tiene una relación *uno a uno* con la tabla *facultad* por lo que una vez normalizada quedará de la siguiente manera:

<i>Curso</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
Código	primaria	varchar(20)	no	-
idFac	foránea	varchar(20)	no	-
nomCurso		varchar(50)	no	-

#### Tabla de la facultad en su segunda forma normal.

Esta última tabla tiene una relación *uno a uno* con la tabla de *curso*, se paso la llave id-Fac a la tabla *curso*, por lo tanto su estructura queda igual a la propuesta en su primera forma normal.

<i>Facultad</i>				
<i>Atributo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Auto_incremento</i>
idFac	primaria	varchar(20)	no	-
nomFac		varchar(50)	no	-

#### Tablas de la base de datos ITzA en su tercera forma normal.

Una base de datos está en 3FN si está en 2FN y además todas las columnas que no sean claves dependen de la clave completa de forma no transitiva.

No existen dependencias transitivas, de modo que podemos afirmar que nuestra base de datos está en 3FN.



### 3.4.3 DISEÑO RELACIONAL DE LA BASE DE DATOS DEL SISTEMA ITzA.

A continuación se presenta el modelo relacional que contiene las tablas, atributos, llaves primarias y foráneas.

#### ID

<u>id</u>	pass	nivel	estado
-----------	------	-------	--------

#### PROFESOR

nomProf	ap1Prof	ap2Prof	gradAcad	dirProf	eMail	tel	ID_id
---------	---------	---------	----------	---------	-------	-----	-------

FK

#### DOCUMENTO

<u>idDoc</u>	titulo	resumen	notas	fecha-Doc	perio-do	no-Pags	forma-to	nomArch	tamArch	ID_id	CURSO_codigo
--------------	--------	---------	-------	-----------	----------	---------	----------	---------	---------	-------	--------------

FK

FK

#### CURSO

<u>codigo</u>	nomCurso	FACULTAD_idFac
---------------	----------	----------------

FK

#### FACULTAD

<u>idFac</u>	nomFac
--------------	--------

### 3.5 DISEÑO DE LA INTERFAZ DEL USUARIO DEL SISTEMA ITzA.

Para finalizar este capítulo se hace mención de cómo se desea que sea la interfaz del sistema ITzA. Será una aplicación Web para ser visualizada mediante un navegador de Internet y que cumpla con los estándares de diseño Web XHTML, CSS y JavaScript; por lo que se requiere que sea una interfaz sencilla y de fácil manejo para los usuarios.

En la Fig. 3.9 se muestra una de las páginas Web conformada de tres marcos:

- El encabezado donde se mostrará el título del sistema, la fecha actual, menú para los usuarios no registrados y el menú que tendrá el acceso al sistema para los usuarios registrados.



Fig. 3.9 Primera visión de la interfaz del sistema.

- La parte derecha mostrará un menú para los usuarios registrados.
- En el cuadro central se mostrarán los resultados y reportes de consultas o procedimientos que se realicen el usuario final.

La página Web estará diseñada con una resolución de 640x480 píxeles y cumplirá con los estándares de programación Web para su mejor visualización desde cualquier navegador indistintamente de sistema operativo o la plataforma que posea el cliente (exceptuando dispositivos móviles).



# *CAPÍTULO*

# *4*

## *ASPECTOS DE IMPLEMENTACIÓN DEL SISTEMA.*

- 4.1 PROGRAMAS DE LIBRE DISTRIBUCIÓN EN PHP.*
- 4.2 CREACIÓN DE LA BASE DE DATOS, USUARIO Y PRIVILEGIOS.*
- 4.3 CREACIÓN DE LAS TABLAS DEL SISTEMA ITzA.*
- 4.4 PROGRAMACIÓN DEL SISTEMA.*
- 4.5 INTERFAZ DEL SISTEMA.*



## CAPÍTULO 4

### ASPECTOS DE IMPLEMENTACIÓN DEL SISTEMA

En este capítulo vamos a desarrollar las técnicas que se utilizaron para la realización del sistema ITzA. Mencionaremos las herramientas de programación y las plataformas de instalación.

Este proyecto se visualizará desde Internet para ello será necesario programar una interfaz visible desde la Web para permitir a los usuarios registrados, no registrados y administradores del sitio interactuar remotamente por medio de un navegador de Internet o *browser*.

Como se indicó en el capítulo II se va a utilizar el interprete PHP, el servidor común Apache y el Sistema Administrador de base de datos MySQL para trabajar sobre una plataforma Windows o Linux y si hubiese la necesidad de transportarlo a otra plataforma.

En general podemos decir que la implementación del diseño físico es crear las estructuras de la base de datos dentro del SGBD que se vaya a utilizar, para nuestro caso se empleará **MySQL**, el cual nos proporciona las siguientes características:

- Es un sistema manejador de bases de datos relacionales.
- Es un software abierto.
- El servidor de base de datos es rápido, confiable y de fácil uso.
- Capaz de manejar grandes base de datos con millones de datos.
- Permite conexiones entre diferentes máquinas con distintos sistemas operativos.

Para crear la interfaz del sistema, se ha optado por elegir **PHP** como lenguaje de programación, sus características son las siguientes:

- Es un lenguaje de programación de páginas Web del lado del servidor (se ejecuta en el servidor Web justo antes de que se envíe la página a través de Internet al cliente) cuyas características principales son la independencia de plataforma y su gratuidad.
- Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red y otras tareas para crear la página final que verá el cliente.
- El cliente solamente recibe una página con el código HTML resultante de la ejecución de PHP.
- PHP es uno de los lenguajes que sirven para la programación de scripts del lado del servidor y junto con otros lenguajes como ASP o JSP que tienen características similares.
- Para poder programar en PHP se requiere de un servidor preparado para ello. Como el lenguaje de programación es multiplataforma, cualquiera de los principales servidores Web nos servirán para ello.

Para la creación de las estructuras y para el fácil manejo del SGBD se utilizaron algunas herramientas tales como las que se mencionan a continuación:



#### **4.1 PROGRAMAS DE LIBRE DISTRIBUCIÓN EN PHP.**

El lenguaje de programación de páginas dinámicas del servidor multiplataforma y totalmente gratuito, PHP, tiene varias ventajas. Una de estas ventajas es que existen en Internet multitud de programas gratuitos creados en PHP para realizar una buena cantidad de tareas habituales. Estos programas los podemos instalar en nuestro sistema libremente, por ser de libre distribución y disfrutar de sus funcionalidades en nuestros servidores.

Hay muchos programas de libre distribución en PHP. Ejemplos típicos son servidores de banners, servidores de sistemas de afiliación, servidores orientados al control de bases de datos, sistemas de agenda, sistemas de chat, comercios electrónicos, etc., algunos ejemplos son los siguientes:

##### **PhpAdsNew**

Un servidor de banners que tiene las funcionalidades de los mejores productos comerciales. Posibilidad de crear clientes, campañas, banners, y administrar las impresiones de banners de nuestro sitio de la manera que mejor queramos. Mantiene unas excelentes estadísticas de impresiones, reportes, etc.

##### **PHP-Nuke**

Un sistema completo para la administración de un portal Web. Con este sistema podemos administrar un buscador de enlaces, noticias y artículos en general, usuarios registrados, encuestas, etc. Se trata de un software muy popular, ya que existen muchos desarrolladores que han adoptado esta solución para crear su propio portal sin necesidad de muchos esfuerzos en el tiempo de desarrollo.

##### **PhpMyAdmin**

Es un gestor de base de datos MySQL, a través del cual podemos administrar todos los procesos de nuestra base de datos. Muy útil para tenerlo instalado en nuestro alojamiento PHP, ya que, la base de datos MySQL se tiene que administrar por consola y por línea de comandos y a veces el proveedor no nos ofrece esa posibilidad de conexión. Además, porque nos hace más fácil administrar los datos que con sentencias SQL, como se hace por línea de comandos.

Se decidió hacer uso de esta herramienta ya que está completa y permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz Web muy intuitiva (Fig. 4.1).

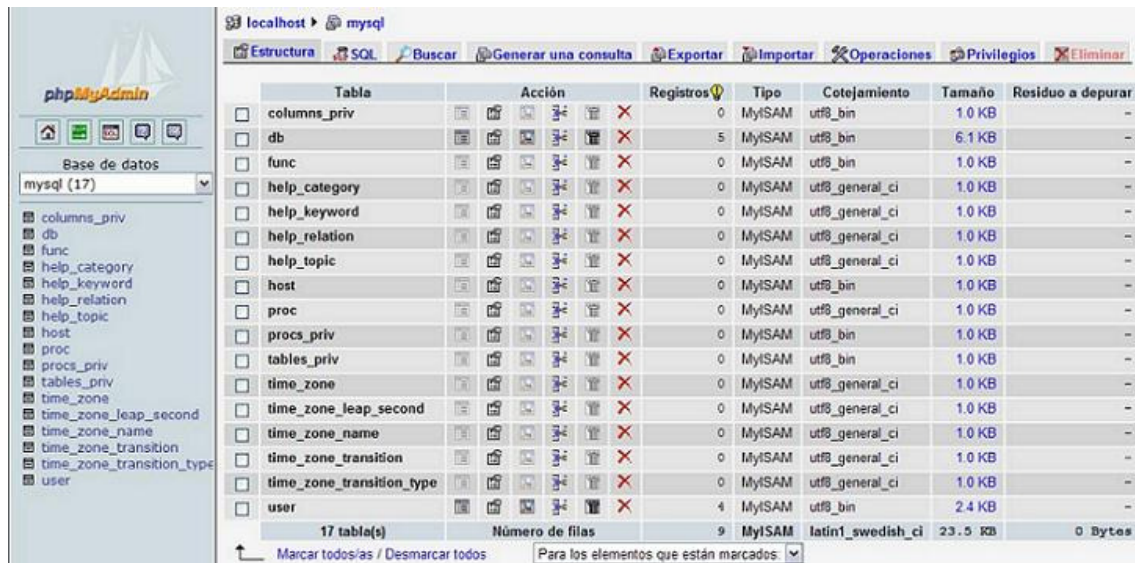


Fig. 4.1 Interfaz Web de la aplicación PhpMyAdmin.

La aplicación en si no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor Web de modo que, cuando accedemos a esos archivos nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas. La herramienta nos permite crear tablas, insertar datos tablas existentes, navegar, editar y borrar registros, borrar tablas, etc., incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

## 4.2 CREACIÓN DE LA BASE DE DATOS, USUARIO Y PRIVILEGIOS.

Como ya se mencionó, se utilizó la herramienta PhpMyAdmin para la creación de la base de datos. La base de datos *mysql* cuenta con 17 tablas, la tabla para la inserción de nuestra base de datos **ITZA** fue la denominada *db* donde se agrego el Host: **localhost**, la base de datos: **itza**, igual que el usuario: **itza**; con todos los privilegios permitidos, como se aprecia en la figura (Fig. 4.2).

Para el caso de la creación del o los usuarios que acceden a la base de datos ITZA se hizo de manera similar a la inserción del nombre de la base de datos. Este está en la base de datos *mysql* en la tabla denominada *user* donde se agregó:

Host: **localhost**,  
Usuario: **itza**  
Password: \*93F94DE2AC4A2A79D0471F0805C84D07F9FB16DF  
de tipo password,  
Privilegios permitidos: Todos.

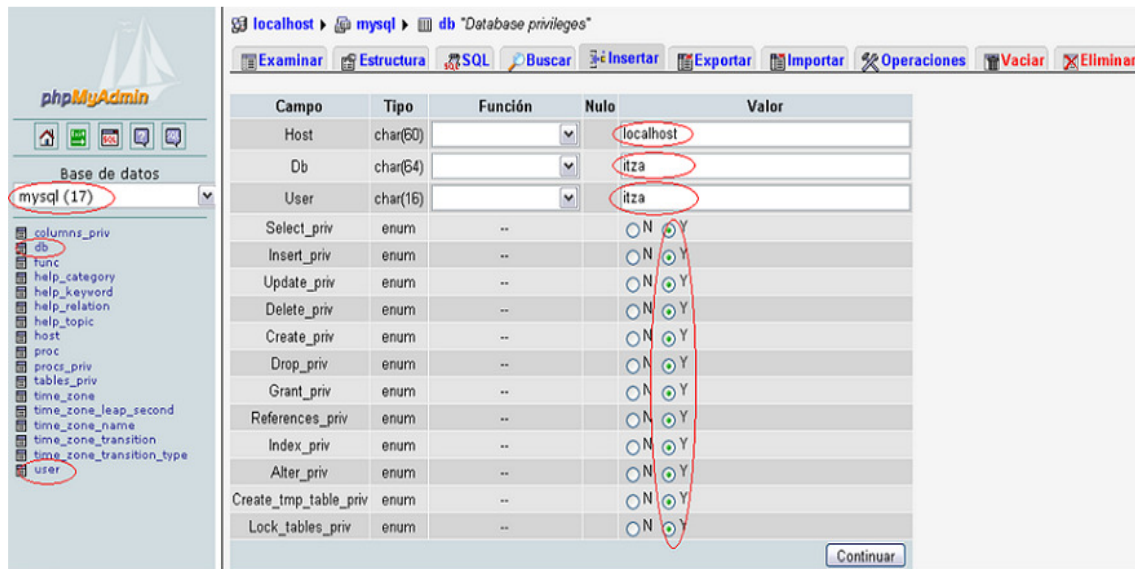


Fig. 4.2 Inserción de la nueva base de datos ITZA.

### 4.3 CREACIÓN DE LAS TABLAS DEL SISTEMA ITZA.

Para la creación de las tablas del sistema ITZA se utilizó la herramienta DBDesigner 4<sup>5</sup> la cual se describe brevemente:

- DBDesigner es un sistema gratuito para el diseño de bases de datos, integra el modelado, creación y mantenimiento en uno solo.
- Trabaja en plataformas tanto para Windows como Linux.
- Es desarrollado y optimizado para MySQL.

En DBDesigner siempre se trabaja con un modelo. Un modelo es una visualización de la meta-información almacenada en una base de datos (ejemplo: tablas e índices, relaciones, etc.) Aunque esto es posible para almacenar datos iniciales para cada tabla directamente en el modelo, esto sólo representa la meta-información, no los datos mismos.

A continuación se presenta el diagrama relacional creado con *DBDesigner*, (Fig. 4.3) que contiene las tablas, atributos, llaves primarias y foráneas, así como las relaciones de cada tabla.

<sup>5</sup> <http://www.fabforce.net/dbdesigner4/>

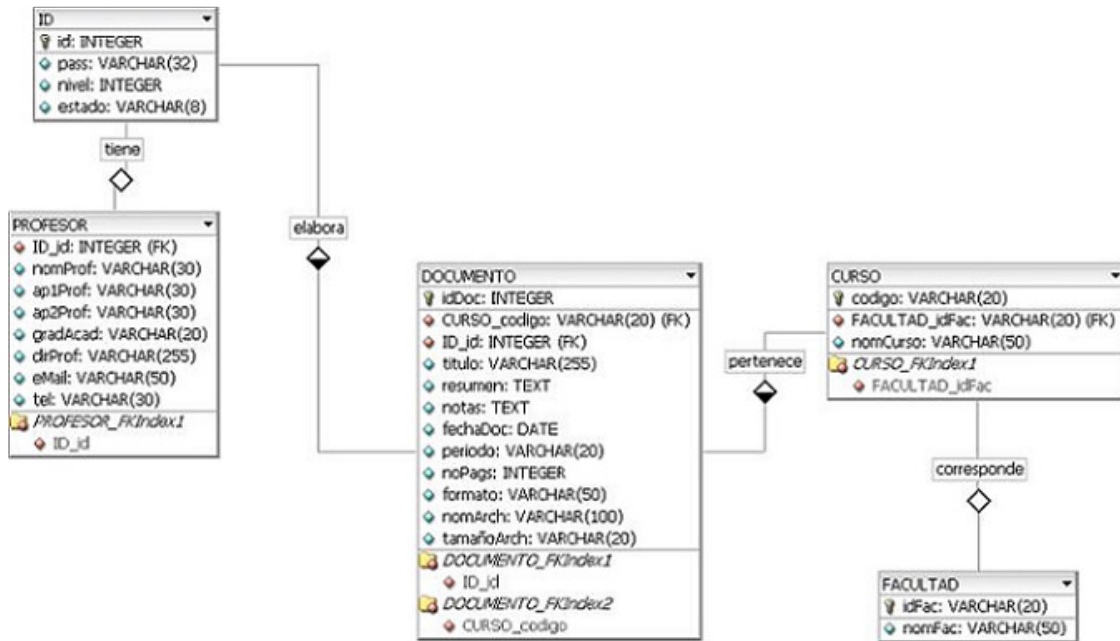


Fig. 4.3 Diseño relacional de la base de datos ITZA.

Este software cuenta con una herramienta que hace la exportación del script SQL para la creación de las tablas seleccionadas y todas sus llaves dependiendo de las relaciones que existan entre las mismas tablas. Esta exportación se copia en un editor de textos o en su defecto, nosotros la copiamos hacia la aplicación *phpMyAdmin* en la pestaña de *SQL* (fig. 4.4) para poder crear las tablas correspondientes.

The screenshot shows the phpMyAdmin interface with the 'SQL' tab selected. The SQL query displayed is:

```
CREATE TABLE DOCUMENTO (
  idDoc INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  CURSO_codigo VARCHAR(20) NOT NULL,
  ID_jd INTEGER UNSIGNED NOT NULL,
  titulo VARCHAR(255) NOT NULL,
  resumen TEXT NULL,
  notas TEXT NULL,
  fechaDoc DATE NOT NULL,
  periodo VARCHAR(20) NOT NULL,
  noPag INTEGER UNSIGNED NOT NULL,
  formato VARCHAR(10) NOT NULL,
  nomArch VARCHAR(100) NOT NULL,
  tamañoArch VARCHAR(20) NOT NULL,
  PRIMARY KEY (idDoc),
  INDEX DOCUMENTO_FKIndex1 (ID_jd),
  INDEX DOCUMENTO_FKIndex2 (CURSO_codigo)
) TYPE = INNODB;
```

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
curso	[Icons]	0	MyISAM	latin1_swedish_ci	1.0 KB	-
documento	[Icons]	0	MyISAM	latin1_swedish_ci	1.0 KB	-
facultad	[Icons]	0	MyISAM	latin1_swedish_ci	1.0 KB	-
id	[Icons]	23	MyISAM	latin1_swedish_ci	3.4 KB	-

Fig. 4.4 Creación de las tablas en phpMyAdmin.



PhpMyAdmin también cuenta con la facilidad de poder crear las tablas con sus atributos de una forma manual, donde paso a paso se van definiendo el nombre de la tabla, cada uno de sus campos con sus características propias y sus llaves correspondientes.

#### ***4.4 PROGRAMACIÓN DEL SISTEMA.***

La implementación del sistema se refiere a la programación de la versión final del sistema. En esta etapa se muestran los módulos del sistema, las herramientas para la captura y validación de la información, los métodos de seguridad implementada y la interacción con la base de datos.

El contenido de las páginas Web del sistema ITzA estará escrito en lenguaje HTML con el objetivo de avanzar en el proyecto de la W3C de lograr una Web semántica, en donde la información y la forma de presentarla estén claramente separadas. En este sentido XHTML servirá únicamente para transmitir la información que contiene el documento, dejando para las hojas de estilo en cascada (CSS) y a JavaScript su aspecto y diseño.

Por tal motivo que en este sistema se emplearán hojas de estilo y scripts en JavaScript, además; toda la parte interactiva del sistema se realizará por medio de scripts en lenguaje PHP. A continuación se dan breves detalles sobre este lenguaje de programación de código abierto y gratuito.

#### ***4.5 INTERFAZ DEL SISTEMA.***

La interfaz del sistema está basada en los estándares Web 2.0 y es por medio de ella que los usuarios pueden comunicarse e interactuar. Las principales funciones de la interfaz Web son las siguientes:

- Control de acceso.
- Manipulación de información del sistema.
- Manipulación de archivos.
- Actualización del sistema.
- Búsqueda, recuperación de información y descarga de documentos.

##### ***4.5.1 PÁGINA PRINCIPAL DEL SISTEMA ITzA.***

La página principal del sistema cuenta con dos marcos (frames) como se aprecia en la imagen (Fig. 4.5):

El encabezado: siempre estará presente en el sistema, independientemente de que la estén utilizando usuarios registrados o usuarios visitas para realizar búsquedas. Cuenta con el nombre del sistema, la fecha actual y los escudos de la Universidad y de la Facultad de Ciencias de la Computación. También tiene el menú horizontal donde cualquier usuario puede realizar sus búsquedas, ver el mapa del sitio, algunos recursos extras y la parte para poder ingresar al sistema con un nombre y contraseña.



Fig. 4.5 Página principal del sistema.

#### 4.5.2 INTERFAZ DEL BUSCADOR DE DOCUMENTOS ELECTRÓNICOS POR TÍTULO.

En esta parte del sistema como se mencionó anteriormente se encuentra la barra del buscador, donde la búsqueda se puede realizar de cuatro formas: por título del documento, nombre del profesor, nombre de la materia o código y por periodo. En la figura siguiente se muestra la búsqueda por título del documento (Fig. 4.6)



Fig. 4.6 Búsqueda por título de documento.



Después de que el sistema muestra una lista con documentos con el nombre similar o exacto en la barra de búsqueda; al dar clic al título del documento el sistema nos muestra una pantalla con: id del documento, clave del curso, nombre del profesor que lo ingresó al sistema, título del documento, resumen del documento, notas acerca del documento, el periodo, fecha en el que se ingresó y el nombre del archivo donde ahí se podrá descargar (Fig. 4.7).

Id Documento:	2
Clave Curso	CCO113
Profesor	ARMANDO HERNANDEZ
Título	programación y logica computacional
Resumen	consiste en la aplicación del corpus de conocimiento sobre lógica para el diseño de lenguajes de programación; no debe confundirse con la disciplina de la lógica computacional
Notas	Realizar un resumen de los dos paradigmas de programación: la programación declarativa y la programación funcional
Periodo	otoño 2009
Fecha	2009-09-15
Archivo	<a href="#">CARTA_AUTORIZACION.pdf</a>

Regresar

Fig. 4.7 Vista detallada del documento.

Después de haber mostrado una vista detallada del documento la siguiente figura 4.8 muestra la pantalla de descarga de archivo seleccionado, en esta parte del sistema se podrá abrir directamente el documento, guardar en una ubicación específica o cancelar la descarga.

¿Desea abrir o guardar este archivo?

Nombre: CARTA\_AUTORIZACION.pdf  
Tipo: Adobe Acrobat Document  
De: localhost

Abrir Guardar Cancelar

Aunque los archivos procedentes de Internet pueden ser útiles, algunos archivos pueden llegar a dañar el equipo. Si no confía en el origen, no abra ni guarde este archivo. ¿Cuál es el riesgo?

Id Docu	
Clave Cu	
Profesor	
Título	
Resumen	
Notas	Realizar un resumen de los dos paradigmas de programación: la programación declarativa y la programación funcional
Periodo	otoño 2009
Fecha	2009-09-15
Archivo	<a href="#">CARTA_AUTORIZACION.pdf</a>

Regresar

Fig. 4.8 Descarga del documento específico.

La interfaz de búsqueda de documentos por nombre del profesor, materia y periodo son parecidas a la búsqueda de documento por título, simplemente se diferencian en el tipo de búsqueda en el que se ingresan datos distintos y en las que se presentan todas las coincidencias donde al menos, aparece una vez el término deseado. Finalmente la vista detallada es exactamente igual al de la Fig. 4.7.

#### 4.5.3 INTERFAZ DE INICIO DE SESIÓN.

En la siguiente imagen se muestra la interfaz cuando un usuario, ya sea administrador o editor quiere ingresar al sistema mediante una *login* y un *password*, que previamente proporcionó el administrador del sistema. Si el *login* o *password* son erróneos se presenta una pantalla de intento fallido; si el usuario está temporalmente desactivado también muestra un mensaje de información en el cual el usuario no podrá ingresar al sistema hasta que el administrador cambie su estatus (Fig. 4.9).



The screenshot shows the login page of the Benemérita Universidad Autónoma de Puebla. At the top, there is a navigation bar with the university logo on the left, the text "Benemérita Universidad Autónoma de Puebla" in the center, and "Facultad de CIENCIAS de la COMPUTACIÓN" on the right. Below this is a dark blue bar with menu items: "Inicio", "Búsquedas", "Recursos", "Mapa del Sitio", "Entrar al Sistema", and the date "Martes 15 de Septiembre de 2009". The main content area is white and contains a central logo for the "Facultad de la Computación". Below the logo, the text "Ingreso al Sistema de Información Electrónica" is displayed. There are two input fields: "Usuario" and "Password". At the bottom of the form are two buttons: "Ingresar" and "Regresar".

Fig. 4.9 Inicio de sesión.

#### 4.5.4 INTERFAZ DE PANTALLA CON MENÚ DE ADMINISTRADORES.

En la siguiente figura 4.10 se puede apreciar el menú del administrador o administradores del sistema en el cual pueden realizar diferentes tareas como son:

- Administración de usuarios. Tareas como las de insertar, buscar, modificar o eliminar datos de usuarios.
- Administración de cursos. Insertar, buscar, modificar o eliminar cursos.
- Administración de facultades. Insertar, buscar, modificar o eliminar facultades.
- Consulta de documentos.
- Respaldo de la base de datos.



- Restauración de la base de datos.
- Término de sesión.



Fig. 4.10 Pantalla de inicio de administrador del sistema.

#### ***4.5.5 INTERFAZ DEL ADMINISTRADOR PARA REGISTRO DE PROFESORES, CURSOS Y FACULTADES.***

A continuación se muestra la pantalla con privilegios de administrador para registrar los datos generales de un nuevo profesor a la base de datos (Fig. 4.11).

Aparecen diversos campos del formulario que se deben llenar apropiadamente para agregar el usuario al sistema; al final, en la parte inferior se muestran tres botones, al pulsar el botón *Insertar* se verifica que ningún campo quede vacío y si eso se cumple, se almacena el registro en la base de datos; al pulsar el botón *Limpiar formulario* se elimina todo lo escrito hasta el momento y el botón *Cancelar* abortará la adición del nuevo usuario y retornará al inicio.



Benemérita Universidad Autónoma de Puebla  
BÚSQUEDA ELECTRÓNICA  
Facultad de CIENCIAS de la COMPUTACIÓN  
Miércoles 16 de Septiembre de 2009

Insertar

Profesores Cursos Facultades

Datos del Profesor

Id_Prof:	<input type="text"/>
Clave Prof	<input type="text"/>
Nivel de Usuario	EDITOR
Nombre(s)	<input type="text"/>
Apellido Paterno	<input type="text"/>
Apellido Materno	<input type="text"/>
Grado Académico	LIC.
Dirección	<input type="text"/>

Administrador: MARITZA

Insertar  
Buscar  
Modificar  
Eliminar  
Respalda BD  
Restaurar BD  
Salir

Fig. 4.11 Registro de datos de un nuevo profesor al sistema.

La siguiente figura 4.12 muestra el registro de un nuevo curso con su clave correspondiente y la facultad a la que pertenece.

Benemérita Universidad Autónoma de Puebla  
BÚSQUEDA ELECTRÓNICA  
Facultad de CIENCIAS de la COMPUTACIÓN  
Miércoles 16 de Septiembre de 2009

Insertar

Profesores Cursos Facultades

Datos del Curso

Código:	00203
Facultad	CIENCIAS DE LA COMPUTACION
Nombre	algoritmos y estructura de datos

Insertar Limpia formulario Cancelar

Administrador: MARITZA

Insertar  
Buscar  
Modificar  
Eliminar  
Respalda BD  
Restaurar BD  
Salir

Fig. 4.12 Registro de un nuevo curso al sistema.

El registro de una nueva facultad es similar a las dos pantallas anteriores.

#### 4.5.6 INTERFAZ DEL ADMINISTRADOR PARA CONSULTA DE PROFESORES, DOCUMENTOS, CURSOS Y FACULTADES.

La Fig. 4.13 se muestra el resultado de la consulta de profesores con los datos correspondientes, el cual contiene en la parte superior un criterio de búsqueda rápida, también con-



tiene un sistema de paginación que muestra en pantalla resultados en grupos de diez y un sistema de ordenación por id, nivel, estado, etc.

ID	Nivel	Estado	Nombre	ApellidoPat	ApellidoMat	Grado	Dirección	Email	Telefono
980010896	0	INACTIVO	ERNESTO	CORONEL	ALEJANDREZ	DOC	GUADALAJARA, JALISCO	netito@yahoo.com.mx	-----

Fig. 4.13 Consulta de profesores.

La consulta de documentos desde el menú del administrador del sistema se muestra en la siguiente figura 4.14, la cual contiene un criterio de búsqueda rápido, un método de ordenación con paginación de diez elementos. También desde esta parte se puede descargar el documento deseado.

ID	Curso	Profesor	Titulo	Resumen	Notas	Fecha	Periodo	No. Pags	Formato	Archivo	Tamaño
2	CCO113	ARMANDO HERNANDEZ ORTIZ	programación y logica computacional	consiste en la aplicación del corpus de conocimiento sobre logica para el diseño de lenguajes de programación; no debe confundirse con la disciplina de la logica computacional	Realizar un resumen de los dos paradigmas de programación: la programación declarativa y la programación funcional	2009-09-15	otoño 2009	0	application/pdf	CARTA_AUTORIZACION.pdf	21296

Fig. 4.14 Consulta de documentos.

La consulta de cursos y facultades es similar a las expuestas anteriormente.

#### 4.5.7 INTERFAZ DEL ADMINISTRADOR PARA MODIFICIACIÓN DE PROFESORES, DOCUMENTOS, CURSOS Y FACULTADES.

La interfaz de modificación de profesores, cursos y facultades en el menú del administrador es muy parecida a excepción por los datos contenidos en cada uno, nótese que en este menú el administrador no puede insertar ni modificar ni eliminar documentos, solo puede consultar. A continuación se mostrará en las dos figuras siguientes esta interfaz.

La pantalla muestra una lista de los profesores existentes dentro de la base de datos, contiene un método de búsqueda rápida para localizar un registro específico en menor tiempo, dentro de esta pantalla se elige el registro el cual se desea modificar y posteriormente se presiona el botón *Modifica* (Fig. 4.15)

Benemérita Universidad Autónoma de Puebla  
BÚSQUEDA ELECTRÓNICA  
Facultad de CIENCIAS de la COMPUTACIÓN  
Miércoles 16 de Septiembre de 2009

Modificar

Profesores Cursos Facultades

<input checked="" type="radio"/>	980010	1	ACTIVO	CESAR	CORDERO	SÁNCHEZ	DOC	U.H. LA MARGARITA	ccorderos@gmail.com	2339531
<input type="radio"/>	980010895	1	ACTIVO	MARITZA	ALEJANDREZ	OLIVO	LIC	CDA. "A" JOSEFA ORTIZ DE DOMINGUEZ 12309 COL. GUADALUPE HIDALGO	itza_ale@hotmail.com	(222)241091

Número de profesores encontrados N = 11  
Ordenados por: ID\_id  
Criterio de búsqueda:  
1 2 siguiente  
Modifica

Inicio Inserta Búsqueda Modifica Elimina

Administrador: MARITZA  
Insertar  
Buscar  
Modificar  
Eliminar  
Respalda BD  
Restaurar BD  
Salir

Fig. 4.15 Pantalla para elección de registro a modificar.

Después de haber elegido el registro a modificar se muestra la pantalla de modificación el cual contiene los datos actuales y la opción de modificarlos (Fig. 4.16). Aquí se puede cambiar el *password* del profesor, dejar intacto o en blanco el registro *Clave*. Cuando se presione el botón *Corregir* los datos que hayan sido modificados quedarán registrados en la base de datos.



Benemérita Universidad Autónoma de Puebla  
BÚSQUEDA ELECTRÓNICA

Facultad de CIENCIAS de la COMPUTACIÓN  
Miércoles 16 de Septiembre de 2009

Id_Prof	980010	(20 char max)
Nombre(s)	CESAR	(50 char max)
Apellido Paterno	CORDERO	(50 char max)
Apellido Materno	SÁNCHEZ	(50 char max)
Grado Académico	DOC. ▾	(Lic, Mto., Dr.)
Dirección	U.H. LA MARGARITA	(80 char max)
E-Mail	ccorderos@gmail.com	(50 char max)
Telefono	2339531	(50 char max)
Nivel Usuario	ADMINISTRADOR ▾	(Administrador, Editor)
Estado Usuario	ACTIVO ▾	(Activo, Inactivo)
Clave		(32 char max)

Corregir Cancelar

Administrador: MARITZA

Inserir  
Buscar  
Modificar  
Eliminar

Respaldar BD  
Restaurar BD

Salir

Fig. 4.16 Pantalla de modificación del registro.

#### 4.5.8 INTERFAZ DEL ADMINISTRADOR PARA ELIMINACIÓN DE PROFESORES, CURSOS Y FACULTADES.

Para la eliminación de registros se utilizó la interfaz parecida a la de modificación. A continuación solo se mostrará la pantalla para eliminación de uno o varios cursos; en este caso cabe mencionar que se pueden eliminar uno o varios elementos a la vez y que si algún elemento esta siendo utilizado en otra tabla no se podrá eliminar (Fig.4.17).

Benemérita Universidad Autónoma de Puebla  
BÚSQUEDA ELECTRÓNICA

Facultad de CIENCIAS de la COMPUTACIÓN  
Miércoles 16 de Septiembre de 2009

Eliminar

Profesores Cursos Facultades

<input checked="" type="checkbox"/>	MAT220	CIENCIAS DE LA COMPUTACION	MATEMATICAS
<input checked="" type="checkbox"/>	TCU 100	DERECHO	LENGUA EXTRANJERA
<input checked="" type="checkbox"/>	TCU 101	CIENCIAS DE LA COMPUTACION	LENGUA EXTRANJERA II
<input checked="" type="checkbox"/>	TCU 221	CIENCIAS DE LA COMPUTACION	DERECHOS HUMANOS

Número de cursos encontradas N = 8

Ordenados por: codigo

Criterio de búsqueda:

1

Eliminar Cancelar

Inicio Inserta Curso Busqueda Curso Modifica Curso Elimina Curso

Administrador: MARITZA

Inserir  
Buscar  
Modificar  
Eliminar

Respaldar BD  
Restaurar BD

Salir

Fig. 4.17 Pantalla de eliminación del registro.



#### 4.5.9 INTERFAZ DEL ADMINISTRADOR PARA RESPALDO Y RESTAURACIÓN DE LA BASE DE DATOS.

A continuación se presentan las pantallas para respaldar la base de datos ITzA (Fig.4.18) y restauración de la base de datos ITzA (Fig. 4.19).

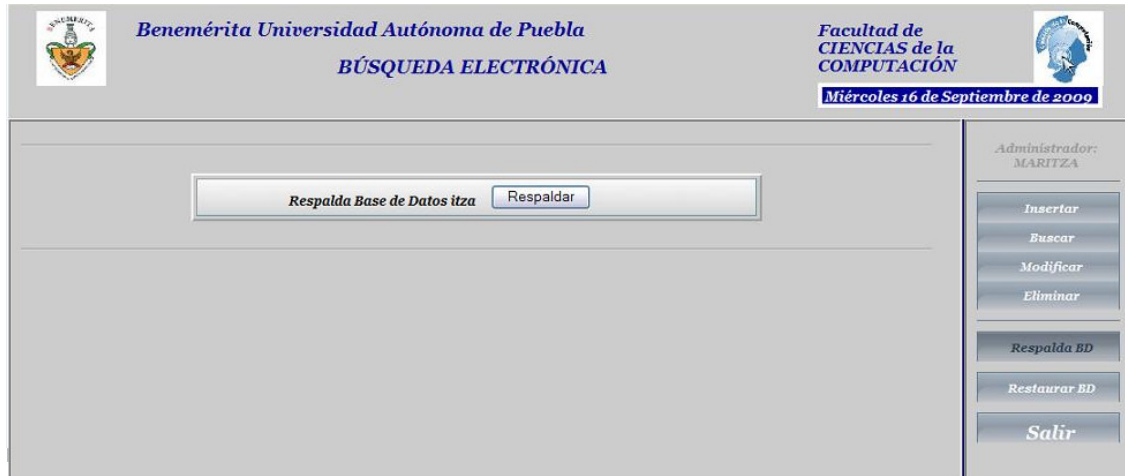


Fig. 4.18 Respaldo de base de datos.

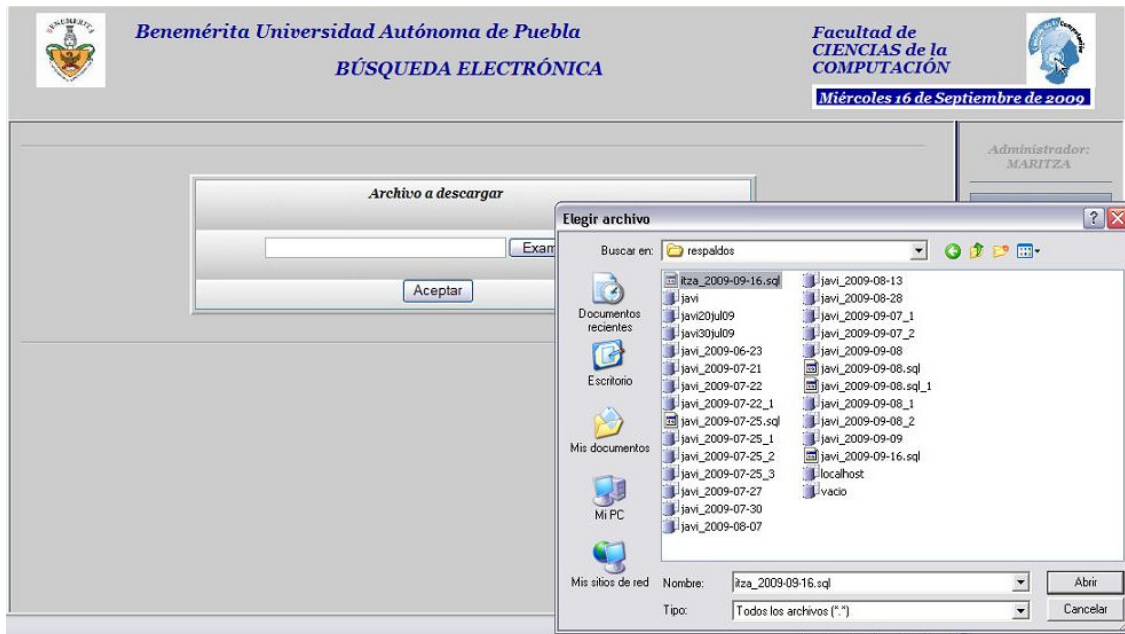


Fig. 4.19 Restauración de base de datos.

#### 4.5.10 INTERFAZ DEL MENÚ DEL EDITOR.

El menú del editor consta de tareas de administración de sus propios documentos, solo puede insertar, modificar y eliminar sus documentos; también puede consultar documentos de cualquier profesor como lo hace un usuario que solo es invitado y consultar profesores, cursos y facultades.

La figura 4.20 muestra la inserción de un documento, se muestra un formulario que ya contiene el nombre del profesor por default, la elección de un curso que ya fue dado de alta por el administrador, y los espacios para insertar el título del documento, un resumen, algunas notas, el número de páginas y la opción de buscar el archivo que se guardará para la consulta posterior del documento.

Cuando se haya completado el registro de los datos del documento y se haya elegido el archivo digital, presionando el botón *Insertar* se guardarán los datos y el archivo correspondiente.

The screenshot displays the 'Benemérita Universidad Autónoma de Puebla' interface for 'BÚSQUEDA ELECTRÓNICA'. The page header includes the university logo, the faculty name 'Facultad de CIENCIAS de la COMPUTACIÓN', and the date 'Miércoles 16 de Septiembre de 2009'. The main content area is titled 'Información del Documento' and contains a form with the following fields:

- Profesor:** LUIS GAEL OLIVERA
- Clave Materia (CURSO):** CC0203 -->ALGORITMOS Y ESTRUCTURA DE DATOS
- Título:** ALGORITMOS
- Resumen:** ALGORITMO: conjunto finito de reglas que dan una secuencia de operaciones para resolver todos los problemas de un tipo dado. De forma más sencilla, podemos decir que un algoritmo es un conjunto de pasos que nos permita obtener un dato. Además
- Notas:** Leer tema para exámen
- No. pags.:** 55
- Nombre Archivo:** C:\Documents and Settings\liza\Mis doc

At the bottom of the form are buttons for 'Insertar', 'Limpiar formulario', and 'Cancelar'. On the right side, there is a sidebar with buttons for 'Insertar Doc', 'Buscar Registro', 'Modificar Doc', 'Eliminar Doc', and 'Salir'. The editor's name 'Editor: LUIS GAEL' is also visible in the sidebar.

Fig. 4.20 Registro de un documento y su archivo.

Las tareas correspondientes a la consulta, modificación y eliminación de documentos en el menú del editor son similares a las que se describieron anteriormente del menú de administrador por lo que resulta comprensible que se omitan en este texto.

#### 4.5.11 SALIR DEL SISTEMA.

Finalmente la opción *Salir* que se presenta en la interfaz tanto del editor como del administrador es para terminar la sesión dentro del sistema, con ello se le da a entender al servidor que no se realizarán más tareas administrativas dentro del sistema.



## ***CONCLUSIONES.***

En el desarrollo tanto de la tesis como del sistema se esperaba que al final se obtuviera una base de datos y una interfaz Web dinámica real, perfectamente amoldable para administrar, ordenar y publicar documentos digitales en Internet, todo concentrado en un solo lugar para que así las búsquedas y descargas de los archivos se realicen con rapidez, eficacia y no se pierda mucho tiempo en buscar la información en lugares dispersos.

Dado que el análisis fue hecho con cautela al igual que el modelado conceptual, tomando en cuenta todo lo que requería el usuario, se tiene un modelo conceptual capaz de ser implementado en cualquier Base de Datos, el sistema permite de una manera muy sencilla y confiable la inserción, consulta, modificación, eliminación de los documentos y la administración de profesores, cursos y facultades, además de que permite para mayor seguridad respaldar y restaurar la base de datos.

Existe la posibilidad de expandir aún más la funcionalidad de este sistema, sin embargo eso queda fuera del marco de desarrollo propuesto en esta tesis.

Finalmente se puede concluir que el sistema cumple con las expectativas que se tuvieron en un principio.

## ***LIMITACIONES.***

Una de las limitaciones que considero tiene el sistema, es la dependencia del intérprete PHP ya que sin éste no funcionaría, también dependencia de MySQL que debe estar instalado y funcionando adecuadamente en la máquina que sea servidor, pues sin este gestor de base de datos, será prácticamente imposible almacenar los datos.

Es posible que después se detecten algunos errores o casos inesperados que no fueron tomados en cuenta, aún cuando en la última fase de desarrollo, el sistema fue sometido a una serie de pruebas con la finalidad de evaluar su funcionalidad y usabilidad, así como para detectar y eliminar errores de programación y/o lógicos.

## ***PERSPECTIVAS.***

Sería muy favorable que una vez que el sistema haya sido probado y funcione correctamente sea implementado para otras facultades, el cual tiene la posibilidad de crecer pues el análisis que se llevó a cabo se realizó consciente y cuidadosamente con la finalidad de que este pudiera crecer en un futuro.



# BIBLIOGRAFÍA

- [1] Sánchez López Abraham. Ingeniería de Software. Diplomado en Bases de Datos. Agosto 2005. Facultad de Ciencias de la Computación. BUAP.
- [2] Somodevilla García Ma. Josefa. Análisis y Diseño de Bases de Datos. Otoño 2005. Facultad de Ciencias de la Computación. BUAP.
- [3] Martín Ortiz Manuel. Aplicaciones de Bases de Datos Cliente Servidor. Febrero 2006. Facultad de Ciencias de la Computación. BUAP.
- [4] Biblioteca. Página oficial de Wikipledia.  
<http://es.wikipedia.org/wiki/Biblioteca>
- [5] Biblioteca digital. Página oficial de Wikipledia.  
[http://es.wikipedia.org/wiki/Biblioteca\\_digital](http://es.wikipedia.org/wiki/Biblioteca_digital)
- [6] Bibliotecas digitales. CiberHábitat, ciudad de la informática.  
<http://ciberhabitat.gob.mx/universidad/bibliotecas/bibliotecasb.htm>
- [7] Ingeniería de software. E-clases  
<http://eclases.tripod.com/id119.html>
- [8] Diseño lógico de bases de datos. Mail x mail.  
<http://www.mailxmail.com/curso-diseno-base-datos-relacionales/diseno-logico-bases-datos>
- [9] Diseño de bases de datos relacionales. Lycos.  
<http://usuarios.lycos.es/cursosgbd/UD4.htm>
- [10] Modelo de datos relacional. Lobocom sistemas.  
<http://personal.lobocom.es/claudio/modrel001.htm>
- [11] Modelo relacional. Página Oficial de Wikipledia.  
[http://es.wikipedia.org/wiki/Modelo\\_relacional](http://es.wikipedia.org/wiki/Modelo_relacional)
- [12] Modelo relacional, Formas normales. UDLAP.  
<http://ict.udlap.mx/people/carlos/is341/bases03.html>
- [13] Modelo entidad-relación. Página Oficial de Wikipledia.  
[http://es.wikipedia.org/wiki/Diagrama\\_e-r](http://es.wikipedia.org/wiki/Diagrama_e-r)
- [14] Cliente-servidor. Página Oficial de Wikipledia  
<http://es.wikipedia.org/wiki/Cliente-servidor>
- [15] El modelo cliente-servidor. NEO Networking and Emerging Optimization.  
<http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>
- [16] El modelo cliente-servidor. Universidad de los Andes.  
<http://agamenon.uniandes.edu.co/~revista/articulos/cliser.html>



- [17] Modelo cliente servidor. INICTEL.  
<http://sipan.inictel.gob.pe/users/hherrera/mcse.htm>
- [18] Formas normales, modelo relacional. MiTecnologico.  
<http://www.mitecnologico.com/Main/ModeloRelacional>
- [19] Tutorial HTML. Universidad de Murcia.  
<http://www.um.es/psibm/tutorial/#t01>
- [20] Diagrama de flujo de datos. Página Oficial de Wikipledia  
[http://es.wikipedia.org/wiki/Diagrama\\_de\\_flujo\\_de\\_datos](http://es.wikipedia.org/wiki/Diagrama_de_flujo_de_datos)
- [21] Diagrama de flujo de datos. MiTecnologico.  
<http://www.mitecnologico.com/Main/DiagramasDeFlujosDeDatos>
- [22] Diagrama de transición de estados. Facultad de ciencias exactas y naturales y agrimensura, Universidad Nacional del Nordeste.  
[http://exa.unne.edu.ar/depar/areas/informatica/anasistem2/public\\_html/apuntes/maf/anexos/transicion.htm](http://exa.unne.edu.ar/depar/areas/informatica/anasistem2/public_html/apuntes/maf/anexos/transicion.htm)
- [23] Diccionario de datos. Página Oficial de Wikipledia  
[http://es.wikipedia.org/wiki/Diccionario\\_de\\_datos](http://es.wikipedia.org/wiki/Diccionario_de_datos)
- [24] Álvarez Miguel Ángel. MySQL Reference Manual. Copyright c ° 1997-2004 MySQL AB Taller de PHP.
- [25] Hernández Sampieri Roberto, Fernández Colado Carlos, Baptista Lucia Pilar. Metodología de la investigación. Mc Graw Hill, 1ra. Edición.
- [26] Pressman Roger. Ingeniería de software, un enfoque práctico. Mc Graw Hill. 4ta. edición, 1998.
- [27] Date C.J. Introducción a los sistemas de base de datos. Prentice Hall. 7ª edición, 2001.



# APÉNDICE 1.

## MANUAL DE INSTALACIÓN.

Para poder utilizar el sistema ITzA, a continuación se presenta un pequeño manual para instalarlo.

### Requerimientos Básicos

**Hardware:** Computadora con un disco duro de 60 Gb (Giga Bytes) al menos, procesador Pentium IV o superior, 256 Mb (Mega Bytes) de memoria RAM y una tarjeta de red Ethernet o inalámbrica configurada apropiadamente.

**Software:** Se utilizó el intérprete PHP, el servidor Apache y el Sistema Administrador de base de datos MySQL para trabajar sobre la plataforma Windows.

Tener instalado un navegador de internet, como por ejemplo: Internet Explorer versión 4 ó superior.

### Instalación rápida

**Apache2Triad** es una distribución de software del tipo WAMP<sup>6</sup> que integra algunos de los servidores y de los intérpretes de código libre más populares para desarrollar en un entorno Web y proporcionar un webhosting<sup>7</sup>, entre su contenido más notable está: el servido Apache, el interprete PHP y una interfaz gráfica de usuario phpMyAdmin.

Apache2Triad se puede llamar un sistema de infraestructura de Internet, porque contiene todos los servidores, intérpretes, e interfaces usuarios ya configurados y listos para ser utilizados.

Para una instalación rápida y sencilla solo hay que descargar este paquete e instalar siguiendo las instrucciones que el mismo sistema nos presente, e ingresando un nombre y una contraseña de usuario. Una vez instalado, se crea una carpeta en raíz C:\apache2Triad.

En el directorio C:\apache2triad\htdocs se guardan todos los archivos que se deseen correr en el servidor local, aquí se deberá copiar la carpeta DisB que contiene todos los scripts y archivos y carpetas que permiten la funcionalidad para la implementación del sistema itZa.

---

<sup>6</sup> WAMP es el acrónimo usado para describir un sistema de infraestructura de Internet que usa las herramientas: windows, como sistema operativo; Apache, como servidor web; MySQL, como gestor de BD y PHP como lenguaje de programación.

<sup>7</sup> Servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía Web.



Para dar de alta el usuario y privilegios como gestor de la base de datos (DBM), ejecutar el explorador de Internet, ingresar **http://localhost/phpmyadmin**, capturar el usuario y contraseña que fueron registrados al momento de instalación del servidor local Apache.

En la tabla *db* de la base de datos *mysql* se crea nuestra base de datos “bd = itza” y como anteriormente se mencionó, también se da de alta el usuario y contraseña “usuario = itza contraseña = maritza”, con todos los permisos requeridos en la tabla *user* dentro de la misma base *mysql*.

Finalmente, ya instalado el paquete apache2Triad, configurado el nombre de usuario, contraseña, los permisos necesarios y copiada la carpeta DisB, es necesario abrir una página del navegador Web e ingresar **http://localhost/DisB/**, en la pestaña de *Entrar al Sistema* teclear como login ‘1’ y password ‘1’ para empezar a utilizar el sistema como administrador.



## APÉNDICE 2.

### MANUAL DE USUARIO.

ITzA es una herramienta de apoyo a la administración, organización y consulta de personal académico, datos de los cursos y facultades y documentos digitales. Puede guardar, organizar y descargar documentos digitales, también puede respaldar y restaurar la base de datos con la que trabaja.

Este manual le permitirá aprender a utilizar todas las funcionalidades básicas del sistema.

### ¿CÓMO ACCEDER AL SISTEMA SIENDO ADMINISTRADOR?

Acceda a <http://localhost/DisB/>, y de click en la pestaña de *Entrar al sistema*, introduzca su “login” y password.

Benemérita Universidad Autónoma de Puebla

BÚSQUEDA ELECTRÓNICA

Facultad de CIENCIAS de la COMPUTACIÓN

Martes 15 de Septiembre de 2009

Inicio Búsquedas Recursos Mapa del Sitio Entrar al Sistema

Ingreso al Sistema de Información Electrónica

Usuario

Password

Ingresar Regresar

Inicio de sesión.

Una vez entrando al sistema usted podrá realizar cualquiera de las funciones que estén permitidas como administrador del sistema:

- Insertar: Profesores, cursos o facultades.
- Buscar: Profesores, documentos, cursos o facultades.
- Modificar: Profesores, cursos o facultades.
- Eliminar: Profesores, cursos o facultades.
- Respalda la base de datos.
- Restaurar la base de datos.



Pantalla de inicio del Administrador del sistema.

### ***Insertar registros***

Para poder insertar un registro de cualquiera de los tres grupos que se mencionaron anteriormente solo habrá que llenar debidamente el formulario que se presenta y dar click en el botón *Insertar*.



Registro de datos de un nuevo profesor al sistema.

Después de presionar el botón *Insertar* se mostrará una pantalla de confirmación.



### Buscar registros

Para la búsqueda de cualquiera de los grupos mencionados solo habrá que ingresar la palabra exacta o similar al registro que se este buscando.

La consulta de documentos se muestra en la siguiente figura, la cual contiene un criterio de búsqueda rápido, un método de ordenación con paginación de diez en diez elementos; también desde esta parte se puede descargar el documento deseado.

ID	Curso	Profesor	Título	Resumen	Notas	Fecha	Período	No. Págs	Formato	Archivo	Tamaño
2	CCO113	ARMANDO HERNANDEZ ORTIZ	programación y lógica computacional	consiste en la aplicación del corpus de conocimiento sobre lógica para el diseño de lenguajes de programación; no debe confundirse con la disciplina de la lógica computacional	Realizar un resumen de los dos paradigmas de programación: la programación declarativa y la programación funcional	2009-09-15	otoño 2009	0	application/pdf	CARTA_AUTORIZACION.pdf	21296

Consulta de documentos.

### Modificar registros

La interfaz de modificación de profesores, cursos y facultades en el menú del administrador son muy parecidas a excepción por los datos contenidos en cada uno, nótese que en este menú el administrador no puede insertar, modificar ni eliminar documentos, solo los puede consultar. A continuación se mostrará en las dos figuras siguientes un ejemplo de esta interfaz.

La pantalla muestra una lista de los profesores existentes dentro de la base de datos, contiene un método de búsqueda rápida para localizar un registro específico en menor tiempo, dentro de esta pantalla se elige el registro el cual se desea modificar y se presiona el botón *Modifica*.



Benemérita Universidad Autónoma de Puebla  
**BÚSQUEDA ELECTRÓNICA**  
Facultad de CIENCIAS de la COMPUTACIÓN  
Miércoles 16 de Septiembre de 2009

Modificar

Profesores Cursos Facultades

<input checked="" type="radio"/>	980010	1	ACTIVO	CESAR	CORDERO	SÁNCHEZ	DOC	U.H. LA MARGARITA	ccorderos@gmail.com	2339531
<input type="radio"/>	980010895	1	ACTIVO	MARITZA	ALEJANDREZ	OLIVO	LIC	CDA. "A" JOSEFA ORTIZ DE DOMINGUEZ 12309 COL. GUADALUPE HIDALGO	itza_ale@hotmail.com	(222)241091

Número de profesores encontrados N = 11  
Ordenados por: ID\_id  
Criterio de búsqueda:  
1 2 siguiente  
Modifica

Inicio Inserta Búsqueda Modifica Elimina

Administrador: MARITZA  
Insertar  
Buscar  
Modificar  
Eliminar  
Respalda BD  
Restaurar BD  
Salir

Pantalla para elección de registro a modificar.

Después de haber elegido el registro a modificar se muestra la pantalla de modificación del registro el cual contiene los datos actuales y la opción de modificarlos. Aquí se puede cambiar el password del profesor o dejar el actual escribiendo o dejando en blanco el registro *Clave*. Cuando se presione el botón *Corregir* los datos que hayan sido modificados quedarán registrados en la base de datos.

Benemérita Universidad Autónoma de Puebla  
**BÚSQUEDA ELECTRÓNICA**  
Facultad de CIENCIAS de la COMPUTACIÓN  
Miércoles 16 de Septiembre de 2009

<b>Id_Prof</b>	980010	(50 char max)
<b>Nombre(s)</b>	CESAR	(50 char max)
<b>Apellido Paterno</b>	CORDERO	(50 char max)
<b>Apellido Materno</b>	SÁNCHEZ	(50 char max)
<b>Grado Académico</b>	DOC. ▾	(Lic, Mto., Dr.)
<b>Dirección</b>	U.H. LA MARGARITA	(80 char max)
<b>E-Mail</b>	ccorderos@gmail.com	(50 char max)
<b>Telefono</b>	2339531	(50 char max)
<b>Nivel Usuario</b>	ADMINISTRADOR ▾	(Administrador, Editor)
<b>Estado Usuario</b>	ACTIVO ▾	(Activo, Inactivo)
<b>Clave</b>		(32 char max)

Corregir Cancelar

Administrador: MARITZA  
Insertar  
Buscar  
Modificar  
Eliminar  
Respalda BD  
Restaurar BD  
Salir

Pantalla de modificación del registro.



### Eliminar registros

Para la eliminación de registros se utilizó la interfaz parecida a la de modificación, en este caso cabe mencionar que se pueden eliminar uno o varios elementos a la vez y que si algún elemento está siendo utilizado en otra tabla no se podrá eliminar. Una vez eliminados los registros no se podrán recuperar al menos que se haya tenido un respaldo previo de la base de datos y así restaurarlo.

**Benemérita Universidad Autónoma de Puebla**  
**BÚSQUEDA ELECTRÓNICA**

Facultad de CIENCIAS de la COMPUTACIÓN  
Miércoles 16 de Septiembre de 2009

Eliminar

Profesores **Cursos** Facultades

<input checked="" type="checkbox"/>	MAT220	CIENCIAS DE LA COMPUTACION	MATEMATICAS
<input checked="" type="checkbox"/>	TCU 100	DERECHO	LENGUA EXTRANJERA
<input checked="" type="checkbox"/>	TCU 101	CIENCIAS DE LA COMPUTACION	LENGUA EXTRANJERA II
<input checked="" type="checkbox"/>	TCU 221	CIENCIAS DE LA COMPUTACION	DERECHOS HUMANOS

Número de cursos encontradas N = 8

Ordenados por: código

Criterio de búsqueda:

1

Eliminar Cancelar

Inicio Inserta Curso Busqueda Curso Modifica Curso Elimina Curso

Administrador: MARITZA

Insertar  
Buscar  
Modificar  
Eliminar

Respalda BD  
Restaurar BD

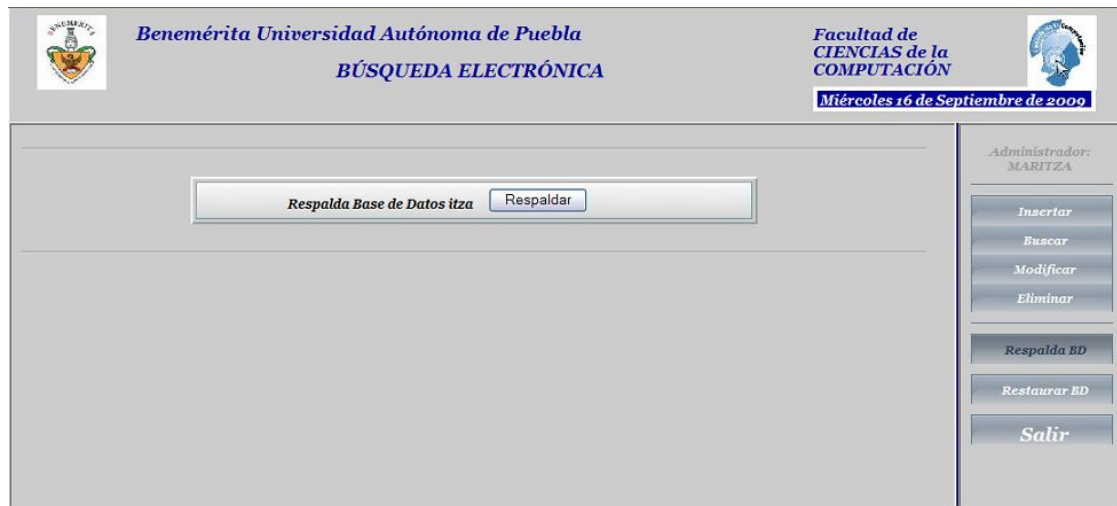
Salir

Pantalla de eliminación del registro.

### Respaldar base de datos

A continuación se presentan las pantallas para respaldar la base de datos itZa. Solo es necesario dar clic en el botón *Respaldar* y automáticamente se guardará el archivo correspondiente a la base de datos y a la fecha actual por ejemplo: itza\_2009-09-16.sql

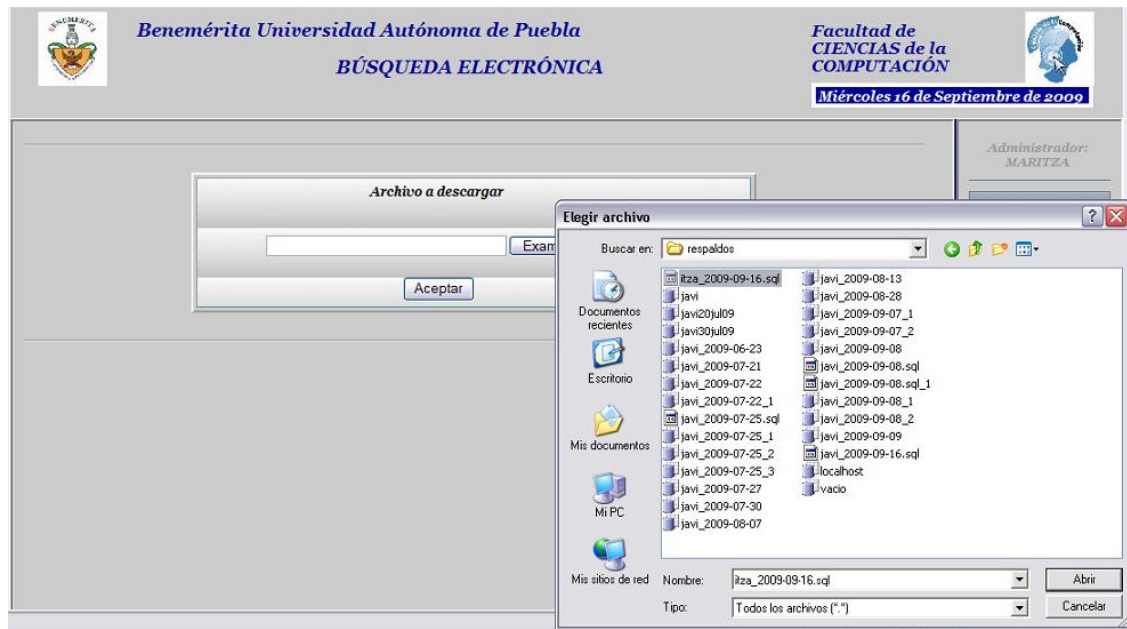
La carpeta actual donde se encuentran los respaldos es: C:\apache2triad\mysql\respaldos



Respaldo de base de datos.

### Restauración de base de datos

En esta parte solo se busca el respaldo que se desea restaurar en la dirección C:\apache2triad\mysql\respaldos y se presiona *Aceptar* con lo que se restaurará el archivo elegido.



Pantalla de restauración de la base de datos.

### ¿CÓMO ACCEDER AL SISTEMA SIENDO EDITOR?

Un editor es la persona que alimenta la base de datos con los documentos (profesores). La dinámica es similar a la de un Administrador solo que con algunas restricciones.



Como se muestra en la siguiente figura, el menú del editor consta de tareas de administración de sus propios documentos, solo puede insertar, modificar y eliminar sus documentos; también puede consultar documentos de cualquier profesor como lo hace un usuario que solo es invitado y consultar profesores, cursos y facultades.

La figura muestra la inserción de un documento, se muestra un formulario que ya contiene el nombre del profesor por default, la elección de un curso que ya fue dado de alta por el administrador, y los espacios para insertar el título del documento, un resumen, algunas notas, el número de páginas y la opción de buscar el archivo que se guardará para la consulta posterior del documento.

Cuando se haya completado el registro de los datos del documento y se haya elegido el archivo digital, presionando el botón *Insertar* se guardarán los datos y el archivo correspondiente.

Información del Documento	
Profesor	LUIS GAEL OLIVERA
Clave Materia (CURSO)	CC0203 --> ALGORITMOS Y ESTRUCTURA DE DATOS
Titulo	ALGORITMOS
Resumen	ALGORITMO: conjunto finito de reglas que usan una secuencia de operaciones para resolver todos los problemas de un tipo dado. De forma más sencilla, podemos decir que un algoritmo es un conjunto de pasos que nos permite obtener un dato. Además
Notas	Leer tema para examen
No. pags.	55
Nombre Archivo	C:\Documents and Settings\liza\Mis do... Examinar...

Registro de un documento y su archivo.

Las tareas correspondientes a la consulta, modificación y eliminación de documentos en el menú del editor son similares a las que se describieron anteriormente del menú de administrador por lo que resulta comprensible que se omitan en este texto.

### *Salir del sistema*

Finalmente la opción *Salir* que se presenta en la interfaz tanto del editor como del administrador es para terminar la sesión dentro del sistema, con ello se le da a entender al servidor que no se realizarán más tareas administrativas dentro del sistema.



## ¿CÓMO ACCEDER AL SISTEMA SIENDO INVITADO?

Como invitado se identifica a toda persona que puede visitar el sistema sin necesidad de un login ni password. El usuario que es invitado tiene la facultad solo de consultar los documentos organizados en la base de datos.

### *Interfaz del buscador de documentos electrónicos por título*

En esta parte del sistema se encuentra la barra del buscador, donde la búsqueda se puede realizar de cuatro formas: por título del documento, nombre del profesor, nombre de la materia o código y por periodo. En la figura siguiente se muestra la búsqueda por título del documento.

Benemérita Universidad Autónoma de Puebla  
BÚSQUEDA ELECTRÓNICA  
Facultad de CIENCIAS de la COMPUTACIÓN

Inicio Búsquedas Recursos Mapa del Sitio Entrar al Sistema Lunes 14 de Septiembre de 2009

Título Documento Profesor Materia Periodo

Búsqueda de Documentos

Nombre del Documento

Buscar Limpiar

Documentos encontrados: 2

1. programación y logica computacional  
Profesor: ARMANDO HERNANDEZ ORTIZ  
ID Documento: 2  
Clave: CCO113  
Fecha: 2009-08-27

Búsqueda por título de documento.

Después de que el sistema nos muestra una lista de los documentos encontrados con el nombre parecido o exactamente escrito en la barra de búsqueda, al dar clic al título del documento, el sistema nos muestra una pantalla de especificaciones del documento elegido como son: id del documento, clave del curso, nombre del profesor que lo ingreso al sistema, el título del documento, un resumen del documento, notas sobre el documento, el periodo y la fecha en el que se ingreso, y el nombre del archivo donde ahí se podrá descargar.



Id Documento:	2
Clave Curso	CCO113
Profesor	ARMANDO HERNANDEZ
Título	programación y lógica computacional
Resumen	consiste en la aplicación del corpus de conocimiento sobre lógica para el diseño de lenguajes de programación; no debe confundirse con la disciplina de la lógica computacional
Notas	Realizar un resumen de los dos paradigmas de programación: la programación declarativa y la programación funcional
Periodo	otoño 2009
Fecha	2009-09-15
Archivo	<a href="#">CARTA_AUTORIZACION.pdf</a>

Vista detallada del documento.

Después de haber mostrado una vista detallada del documento, la siguiente figura muestra la pantalla de descarga del archivo seleccionado, en esta parte del sistema se podrá abrir directamente el documento, guardar en una ubicación específica o cancelar la descarga.

¿Desea abrir o guardar este archivo?

Nombre: CARTA\_AUTORIZACION.pdf  
Tipo: Adobe Acrobat Document  
De: localhost

Aunque los archivos procedentes de Internet pueden ser útiles, algunos archivos pueden llegar a dañar el equipo. Si no confía en el origen, no abra ni guarde este archivo. ¿Cuál es el riesgo?

Descarga del documento específico.

La interfaz de búsqueda de documentos por nombre del profesor, materia y periodo son parecidas a la búsqueda de documento por título, simplemente se diferencian en el tipo de búsqueda en el que se ingresan datos distintos y en las que se presentan todas las coincidencias donde al menos, aparece una vez el término deseado, finalmente la vista detallada es exactamente igual a la figura anterior.