



**Benemérita**

**Universidad Autónoma De Puebla**

Facultad de Ciencias de la Computación

“CONTROL DE UNA BIBLIOTECA  
ESCOLAR “

**TESIS PROFESIONAL**

**Para obtener el título de:**

LICENCIADO EN CIENCIAS DE LA  
COMPUTACIÓN

**Para obtener el título de:**

LICENCIADO EN CIENCIAS DE LA  
COMPUTACIÓN

**Presenta:**

FRANCISCO FLORES CAMARILLO

**Asesor:**

DRA. DARNES VILARIÑO AYALA

Puebla de Zaragoza a 19 de Noviembre de 2008

## INTRODUCCIÓN

Poco a poco el uso de Internet se ha convertido en un medio de comunicación que empieza a tomar un lugar preponderante frente a otros medios como la televisión, o el teléfono. Si bien no ha logrado desplazarlos por completo, sí ha ido colocándose, poco a poco, por encima de ellos. No es coincidencia la preferencia sobre otros medios, porque Internet ha resultado ser una herramienta útil para acortar distancias y disminuir el tiempo de la comunicación. Podríamos afirmar que Internet es el medio de comunicación para y de una época, en la que sólo existe el presente por la simultaneidad con que se transmiten las cosas.

En los últimos cincuenta años, las computadoras han tenido un gran desarrollo. En la actualidad las computadoras son una herramienta muy útil en gran diversidad de disciplinas y actividades. Una de ellas es la educación, en la cual las bibliotecas son de gran importancia. Es en este punto en donde nace la relación Biblioteca - Computadora. Desde hace algunos años se inicia el uso de computadoras en las bibliotecas para buscar en un catálogo los títulos disponibles, representando sólo una herramienta útil dentro de una biblioteca convencional.

Con los avances en las ciencias computacionales, entre los cuales están el almacenamiento, búsqueda, recuperación y procesamiento de grandes volúmenes de información, el surgimiento de hipermedios y de grandes redes de computadoras como Internet, ha sido posible llevar mucho más allá esta relación. De hecho, algunos autores como I. Asimov, H.G Wells y B. Fuller, plantearon la posibilidad de que en el futuro existiera una fuente global de información que contuviese toda la información generada en la historia de la humanidad, la cual fuese accesible en cualquier lugar del planeta para cualquier persona que quisiera hacerlo. Ahora, con el desarrollo tecnológico, nos encontramos ante la posibilidad de hacer esto realidad.

En los campos académico y de investigación, el uso del ciberespacio para el intercambio de ideas ha resultado de gran eficacia, debido a que el conocimiento se ha ampliado y diversificado, no sólo por la facilidad para obtener opiniones diferentes y distantes, sino porque se tiene un mayor acceso a la información y a fuentes documentales, visuales y auditivas, en forma mucho más fácil y eficaz, siendo favorecidos, en gran medida, las investigaciones y los enfoques de la vida académica.

Con la digitalización de los acervos documentales y fotográficos antiguos, se ha aumentado la durabilidad de las fuentes, al tiempo que, como se mencionó, se hacen más accesibles al público. Con el proceso de digitalización es más factible la conservación de archivos, porque los documentos se exponen menos a la intemperie y al constante manejo de los usuarios, lo que poco a poco va deteriorando el papel (o la piel en que se elabora la fuente, si de documentos escritos se trata) y el contenido, la mayoría de las veces plasmado con tintas de origen vegetal. Además, la digitalización permite tener un respaldo que puede ser usado, bajo una correcta asesoría sobre el proceso, para hacer consultas especializadas con una calidad similar a la de un documento original.

En le presente trabajo se pretende automatizar el proceso en una biblioteca, permitiendo controlar los préstamos, devoluciones, usuarios y sanciones.

Este documento está estructurado por cinco capítulos, el cual inicia con los inicios de las aplicaciones Web y con la descripción de la base de datos, posteriormente se describe la estructura y el funcionamiento del sistema.

# Capítulo 1 MARCO TEÓRICO

## 1.1. HISTORIA DE LAS APLICACIONES WEB

En un principio la Web era sencillamente una colección de páginas estáticas, documentos, etc., para su consulta o descarga. El paso inmediatamente posterior en su evolución fue la inclusión de un método para elaborar páginas dinámicas que permitieran que lo mostrado tuviese carácter dinámico (es decir, generado a partir de los datos de la petición). Este método fue conocido como CGI ("Common Gateway Interface") y definía un mecanismo mediante el cual se pasaba información entre el servidor y ciertos programas externos. Los CGIs siguen utilizándose ampliamente; la mayoría de los servidores Web permiten su uso debido a su sencillez. Además, dan total libertad para elegir el lenguaje de programación que se desea emplear.

El funcionamiento de los CGIs tenía un punto débil: cada vez que se recibía una petición, el servidor debía lanzar un proceso para ejecutar el programa CGI. Como la mayoría de CGIs estaban escritos en lenguajes interpretados, como Perl o Python, o en lenguajes que requerían "run-time environment", como Java o Visual Basic, el servidor se veía sometido a una gran carga. La concurrencia de múltiples accesos al CGI podía ocasionar problemas graves.

Por eso se empiezan a desarrollar alternativas a los CGIs que solucionarán el problema del rendimiento. Las soluciones llegan básicamente por 2 vías: **1)** se diseñan sistemas de ejecución de módulos mejor integrados con el servidor, que evitan la instanciación y ejecución de varios programas, y **2)** se dota a los servidores un intérprete de algún lenguaje de programación que permita incluir el código en las páginas de forma que lo ejecute el servidor, reduciendo el intervalo de respuesta.

Entonces se experimenta un aumento del número de arquitecturas y lenguajes que permiten desarrollar aplicaciones Web. Todas siguen alguna de estas vías. Las más útiles y las más utilizadas son las que permiten mezclar los 2 sistemas: un lenguaje integrado que permita al servidor interpretar comandos "incrustados" en las páginas HTML y, además, un sistema de ejecución de programas mejor enlazado con el servidor, que no implique los problemas de rendimiento propios de los CGIs. [1]

Con los avances obtenidos, cada Empresa, Escuela, o Institución se ve en la necesidad de desarrollar Aplicaciones Web que permitan al usuario acceder desde cualquier parte, a la información por él requerida. El desarrollo de cualquier Aplicación Web impone seguir un proceso de ingeniería de software.

## *1.2. INGENIERÍA DE SOFTWARE*

La Ingeniería de Software aparece a finales de los años sesenta y principios de los setenta empezando con las Técnicas de Programación Estructurada, incorporándolas a las fases del ciclo de vida del software. La programación estructurada fue seguida por otros métodos estructurados de análisis y también métodos estructurados de diseño. Al mismo tiempo, comienzan a usarse tecnologías orientadas a objetos.

En un nivel más Técnico, la Ingeniería de Software empieza con una serie de tareas que hacen modelos y que resultan en una especificación completa de requisitos y una representación del diseño del software que será construido.

En ciertos sistemas de información críticos, el uso de métodos formales es utilizado en el ciclo de vida del software, con el objetivo de producir sistemas con la integridad más alta. Los métodos formales confían en las técnicas matemáticas que expresan y modelan los requisitos de cualquier producto en el ciclo de vida del software.

La Ingeniería de Software se define como una disciplina de la ingeniería en donde se resuelven problemas relacionados con el desarrollo de sistemas de software el cual permite obtener software confiable, rentable que trabaje con máquinas reales y que tenga un funcionamiento eficiente que satisfaga las necesidades del usuario. Además se preocupa de la fiabilidad y el rendimiento, tratando de reducir costos y complejidad. Los modelos que utilizan están basados en teorías matemáticas sólidas. [2]

También, se define como la disciplina tecnológica preocupada de la producción sistemática y del mantenimiento de los productos de software que son desarrollados y modificados en tiempo y dentro de un presupuesto definido.

Las actividades de la ingeniería de Software son: analizar, diseñar, construir e implementar sistemas de información. [3]

Para comprender lo que es el software (y consecuentemente la ingeniería del software), es importante examinar las características del software que lo diferencian del hardware, las cuales son:

El software se desarrolla, no se fabrica en un sentido clásico.

El software no se estropea.

La mayoría del software se construye a medida, en vez de ensamblar componentes existentes.

### 1.3. CICLOS DE VIDA DE SOFTWARE

El ciclo de vida del software también llamado proceso de desarrollo del software requiere un conjunto de conceptos, una metodología, un lenguaje propio y comprende cuatro grandes fases: concepción, elaboración, construcción y transición.

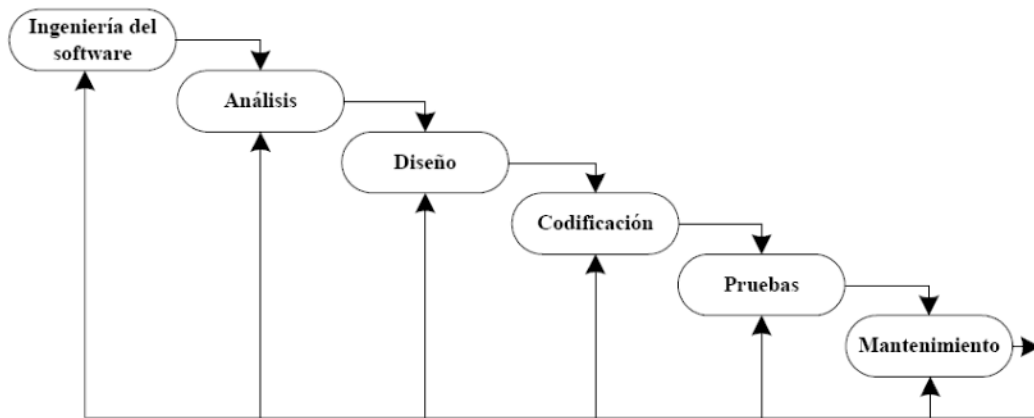
La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

Existen diversos enfoques que dan lugar a distintos paradigmas de la ingeniería de software entre los que se encuentran el ciclo de vida clásico (o en cascada), ciclo de vida en espiral, ciclo de vida con prototipos, etc. En este caso, sólo nos enfocaremos en el ciclo de vida clásico, ya que es el que utilizaremos durante el desarrollo del sistema de base de datos para el control administrativo del centro de servicio autorizado.

### 1.4. MODELO DE CASCADA

El modelo en cascada (o ciclo de vida clásico) abarca un enfoque sistemático, secuencial del desarrollo de software que comienza en un nivel de identificación del problema y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

El ciclo de vida clásico es sencillo y fácil de implantar y gestionar, además es muy frecuente que se utilice, ya que se está muy acostumbrado a este paradigma. En la figura 1.1 se muestra el modelo en cascada.



**Figura 1.1 Modelo en cascada**

Las flechas muestran el flujo de información entre las fases. La flecha de avance muestra el flujo normal. Las flechas hacia atrás representan la retroalimentación. Uno de los arbitrios más importantes del modelo cascada es para los administradores, ya que facilita avanzar en el desarrollo del sistema.

A continuación tenemos una descripción de cada una de las fases del ciclo de vida clásico:

- **Ingeniería del software:** comienza estableciendo requisitos de todos los elementos del sistema.
- **Análisis de requisitos del software:** permite especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software. En esta etapa se construyen los modelos del software que se traducen en diseño de datos, arquitectónico, de interfaz y procedimental.
- **Diseño:** la fase de diseño produce un diseño de datos, un diseño arquitectónico, un diseño de interfaz y un diseño procedimental.
- **Codificación:** durante el desarrollo, un ingeniero de software intenta definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función como una arquitectura del software, cómo han de implementarse detalles procedimentales, cómo han de caracterizarse las interfaces, cómo ha de traducirse el diseño en un lenguaje de programación y cómo han de realizarse la pruebas.

- **Pruebas:** el proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, éstas pruebas se realizan para la detección de errores y para asegurar que se producen resultados reales de acuerdo con los resultados requeridos.
- **Mantenimiento:** El software indudablemente sufrirá cambios después de ser entregado al cliente. Se producirán cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios de su entorno externo, o porque el cliente requiere mejoras funcionales o de rendimiento. Es decir, los cambios pueden ser por corrección, adaptación, mejora y prevención.

## 1.5. DIAGRAMAS DE UML

La explicación se basará en los diagramas, en lugar de en vistas o anotación, ya que son estos la esencia de UML. Cada diagrama usa la anotación pertinente y la suma de estos diagramas crean las diferentes vistas. Las vistas existentes en UML son:

- Vista casos de uso: Se forma con los diagramas de casos de uso, colaboración, estados y actividades.
- Vista de diseño: Se forma con los diagramas de clases, objetos, colaboración, estados y actividades.
- Vista de procesos: Se forma con los diagramas de la vista de diseño. Recalcando las clases y objetos referentes a procesos.
- Vista de implementación: Se forma con los diagramas de componentes, colaboración, estados y actividades.
- Vista de despliegue: Se forma con los diagramas de despliegue, interacción, estados y actividades.

Se dispone de dos tipos diferentes de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica. Los diagramas estáticos son:

- Diagrama de clases: muestra las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.
- Diagrama de objetos: Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas. Se da una visión de casos reales.
- Diagrama de componentes: Muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.
- Diagrama de despliegue: Muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice.
- Diagrama de casos de uso: Muestran los casos de uso, actores y sus relaciones. Muestra quien puede hacer que y relaciones existen entre acciones (casos de

uso). Son muy importantes para modelar y organizar el comportamiento del sistema.

Lo diagramas dinámicos son:

- Diagrama de secuencia, Diagrama de colaboración: Muestran a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que nos dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.
- Diagrama de estados: muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.

Diagrama de actividades: Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

## 1.6. BASES DE DATOS

### 1.6.1. DEFINICIÓN DE BASE DE DATOS

Una base de Datos es un conjunto de datos almacenados entre los que existen relaciones lógicas, están pensadas para resolver los problemas asociados al manejo de grandes volúmenes de información.

En un sistema de Base de Datos se distinguen 4 componentes: Datos o información, Software o programas, hardware o equipo y usuarios.

1- Datos: Los datos almacenados se dividen en una o más bases de datos, por lo tanto, una Base de Datos es un recipiente de datos almacenados y en general es tanto integrada como compartida. [5]

- **Integrada:** Se entiende que la base de datos puede considerarse como una unificación de varios archivos de datos independientes, de donde se elimina parcial o totalmente cualquier redundancia entre los mismos.
- **Compartida:** Los elementos individuales de información de las Bases de Datos pueden compartirse entre varios usuarios distintos, en el sentido de que cada uno de ellos puede tener acceso a la misma parte de la Base de datos y usarla con propósitos distintos.

2- Software: Entre las bases de datos físicas en si, es decir, el almacenamiento real de datos y los usuarios existe un nivel de software que recibe el nombre de DBMS (Sistema de Administración de Bases de Datos). Su objetivo es manejar todas las solicitudes formuladas por los usuarios y/o programas de acuerdo a una función general del DBMS, otro objetivo es proteger a los usuarios contra los detalles a nivel de

hardware, es decir, el DBMS ofrece una visión a los usuarios que está por encima del hardware y apoya las operaciones de éste.

3- Hardware: Está compuesto por los medios de almacenamiento secundarios, memoria principal, etc., donde permanece residente la base de datos. Junto con los dispositivos de E/S asociados, controladores de dispositivos, canales de E/S, etc., ya que se asume que la Base de Datos es demasiado grande para localizarla en memoria.

4- Usuarios: Existen 3 tipos de usuarios:

- Usuario Final: Interactúa con la Base de Datos desde una Terminal en línea a través de un lenguaje de consulta, o a través de un programa (interfaz). Mediante el lenguaje de consulta, el usuario queda libre para realizar cualquier operación y mediante un programa el usuario queda restringido a lo que en éste, se ha establecido.
- Programador de Aplicaciones: Desarrolla los sistemas necesarios para permitir la posibilidad de comunicación o extensión de información desde la Base de Datos.
- Administrador de Base de Datos: Dentro de sus funciones se puede mencionar:
  - 1- Mantener en forma óptima y eficiente la Base de Datos, controlando procedimientos, instalaciones, procesos, etc.
  - 2- Realizar funciones de auditoría, manejando la seguridad de la base de Datos, además de crear usuarios y accesos permitidos.

Un sistema de gestión de base de datos (SGBD) consiste en una colección de base de Datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada Base de Datos contiene información acerca de una empresa particular. El primer objetivo de un SGBD es proporcionar un entorno que sea tanto práctico como eficiente de usar, en la recuperación y almacenamiento de la información. [6]

## 1.6.2. MODELO ENTIDAD - RELACIÓN (E-R)

El Modelo entidad-relación es el modelo conceptual más utilizado para el diseño de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación 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 incluía 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*.

Los elementos usados para el modelo E-R son los siguientes:

### **Entidad**

Cualquier objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuerte y débil. Una *entidad débil* es una entidad cuya existencia depende de la existencia de otra entidad. Una *entidad fuerte* es una entidad que no es débil

### **Relación**

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior. [4]

### **Atributo**

Es una característica de interés o un hecho sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda información extensiva es portada por los atributos. Gráficamente, se representa mediante círculos conectados a las entidades o relaciones a las que pertenecen [7].

### 1.6.3. NORMALIZACIÓN

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. En una estrategia de diseño de abajo a arriba: se parte de los atributos y estos se van agrupando en relaciones (tablas) según su afinidad. Aquí no se utiliza la normalización como una técnica de diseño de bases de datos, sino como una etapa posterior a la correspondencia entre atributos no deseados. Las ventajas de la normalización son las siguientes:

- Evita anomalías en Inserciones, Modificaciones y Borrados
- Mejora la independencia de los datos
- No establece restricciones artificiales en la estructura de los datos

La normalización se lleva a cabo en una serie de pasos. Cada paso corresponde a una forma normal que tiene sus propiedades

#### **Primera forma normal (1FN)**

Una relación está en primera forma normal si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Si se ven la relación gráficamente como una tabla, estará en primera forma normal si tiene un sólo valor en la intersección de cada fila con cada columna.

#### **Segunda Forma Normal**

Una relación está en segunda forma normal si, y sólo si, está en 1FN y además, cada atributo no primo (que no está en la clave primaria) es completamente dependiente de la clave primaria.

#### **Tercera Forma Normal**

Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo no primo no depende transitivamente de la clave primaria. La dependencia  $x \rightarrow z$  es transitiva si existe las dependencias  $x \rightarrow y$ ,  $y \rightarrow z$ , siendo  $x, y$ , atributos o conjunto de atributos de una misma relación [8].

#### 1.6.4. ARQUITECTURA CLIENTE - SERVIDOR

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.
- Por su parte los servidores realizan, entre otras, las siguientes funciones:
- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

#### Uso de Wamp Server 2.0

WampServer es un paquete para el desarrollo de sitios web dinámicos para Windows que combina las últimas versiones del servidor Apache, de la base de datos MySQL y

del lenguaje de programación PHP. WampServer también incluye PHPMyAdmin, PECL y SQLiteManager para una sencilla administración de bases de datos.

El programa se instala y configura automáticamente de modo que se ejecuta el servidor sin necesidad de seleccionar ningún archivo de configuración.

Una característica interesante de WampServer es que permite instalar varias versiones de PHP, MySQL y Apache, de modo que se utilizara cualquiera de ellas instalándola y luego seleccionándola en el menú del icono de la barra de sistema.

La última versión (2.0) incluye:

- Apache 2.2.8
- PHP 5.2.6 + PECL
- SQLiteManager
- MySQL 5.0.51b
- Phpmyadmin

## MySQL

La elección de MySQL como gestor de base de datos radica en que es gratuito tanto para usos privados como comerciales (sólo hay que pagar en el caso de que se desarrolle un producto comercial que este basado en MySQL), y en su disponibilidad para distintos sistemas operativos (Unix, Windows 9x/NT/2000/XP, OS/2, etc.), en que es capaz de trabajar con millones de registros y porque, además, es muy rápido y no necesita grandes recursos de máquina.

MySQL permite la creación de aplicaciones *cliente/servidor* y ofrece una solución integral operando bajo el modelo de código abierto. En este caso es necesario arrancar el programa servidor (Servidor Apache) y el programa cliente (MySQL) y realizar una serie de funciones implementadas en PHP para crear la conexión, puesto que los comandos que podemos introducir en la sesión con MySQL tienen su equivalente como función de PHP.

## LENGUAJE PHP.

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año de 1994 por Rasmus Lerdorf como un CGI escrito

en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas 2 herramientas, sería la primera versión compacta del lenguaje PHP/FI.

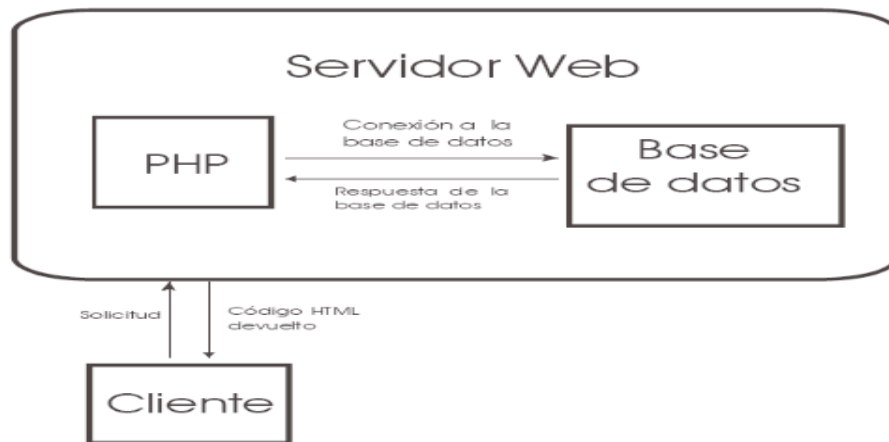
La siguiente gran contribución al lenguaje se realizó a mediados del 97 cuando se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las Bases de Datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3. Actualmente PHP se encuentra en su versión 4. Algunas mejoras de esta nueva versión es su rapidez gracias a que primero se compila y luego se ejecuta, mientras que antes se ejecutaba mientras se interpretaba el código.

## PHP.

PHP corresponde a las iniciales de Personal Home Page, Procesador de Hipertexto. Se trata de un lenguaje de programación con una sintaxis similar a los lenguajes C y Perl, que se interpreta por un servidor web Apache y genera código HTML dinámico. Es decir, nos permite crear un programa que se ejecute en el servidor desde un programa visualizador de páginas web y dar respuesta en función de los datos que introduzca el usuario.

El cliente nunca verá el código del programa PHP, sólo le llegarán las páginas HTML que genere el programa. A diferencia de JavaScript, que se ejecuta en las máquinas clientes, un programa PHP se ejecuta en el servidor Web.

En la figura 1.1 siguiente se puede apreciar el esquema de funcionamiento de un programa en PHP: El cliente realiza una petición de un programa a un servidor web como si se tratara de cualquier otra página; el cliente no sabrá distinguirlo. El servidor web, por la extensión de la página (.php), se la envía al intérprete PHP y este, a su vez ejecutado el programa, le devuelve los resultados al navegador cliente.



**Figura 1.2 Diagrama de funcionamiento de PHP**

## Ventajas de PHP

PHP presenta múltiples ventajas frente a otros lenguajes de programación que necesariamente harán que este lenguaje se imponga como una alternativa para el desarrollo de todo tipo de aplicaciones.

- Es muy fácil de aprender.
- Esta ampliamente probado como herramienta. Un millón y medio de servidores por el momento, en todo el mundo lo avalan como una plataforma para desarrollar aplicaciones de portales, comercios electrónicos, aplicaciones en intranets, etc.
- Es multiplataforma.
- Aunque no dispone de soporte comercial, existen numerosas listas de correo en las que se pueden obtener soluciones a los problemas que se puedan presentar.
- Existen recursos en la web que pueden facilitar el desarrollo de las aplicaciones.
- No requiere demasiados recursos. Es posible tener PHP funcionando en un portátil 486 con 16m de RAM para desarrollo.

## Capítulo 2 ANÁLISIS DEL SISTEMA

### 2.1. PLANTEAMIENTO

El objetivo del presente trabajo es automatizar el préstamo de libros y de material didáctico de una biblioteca escolar, la cual presenta pérdidas de material. Además presenta deficiencia en cuanto a los servicios de préstamos y devoluciones, ya que dichos registros se llevan en bitácoras.

Para la biblioteca es necesario llevar un control eficiente con respecto al orden de los libros y al préstamo que se hace, es necesario saber identificar cuantos existen de cada título, también conocer cuántos faltantes hay, saber qué libros son consultados de manera frecuente, así como también saber categorizarlos y controlar los usuarios que no realizan las devoluciones oportunamente.

La base de datos debe almacenar información acerca de los libros, se debe almacenar el título, la editorial, el año de edición, los autores, la clave del libro, la clasificación, así como su estado (prestado o disponible). Se debe almacenar otra tabla similar acerca de revistas de la cual se debe almacenar también el título, el año de edición, el autor, el idioma, el género y su respectivo estado (prestado o disponible). Dicha base también contendrá documentos en formato PDF, los cuales posteriormente podrán ser consultados.

Debe almacenarse además la información tanto de los usuarios que solicitan el servicio de la biblioteca. De los cuales se debe registrar el nombre y la clave de usuario, el password, el e-mail, así como los libros o revistas prestados.

Esta base de datos tendrá un módulo de préstamos, donde se debe almacenar la clave del usuario, la clave del material prestado ya sea libro o revista, la fecha de préstamo, y el sistema calculará diariamente la sanción monetaria correspondiente a cada usuario.

Para trabajar en la base de datos se tienen que realizar algunas operaciones que permitan hacer cambios como lo son: las altas, bajas, modificaciones y consultas. La operación de Altas permitirá agregar algún Libro, Revista, PDF, usuarios; como a su vez también se podrá hacer modificación de algún dato de la tabla que se requiera actualizar. También se podrá eliminar de una tabla un libro, revista, algún usuario o PDF.

El sistema permitirá que se realicen consultas y se puedan extraer las tuplas de la tabla elegida y que se puedan imprimir.

La especificación de requerimientos permite entender y conocer específicamente que es lo que se requiere del sistema; ya que a través de la especificación podemos saber cuál es el problema, cómo se va a solucionar, qué datos manejará y con qué funciones debe contar el sistema.

### 2.2. ESPECIFICACIÓN DE REQUERIMIENTOS

Se necesita de un sistema óptimo y eficiente que sea capaz de realizar las tareas encomendadas por el cliente en un tiempo de espera mínimo. El sistema deberá ejecutar las tareas de registro y control por parte del responsable del sistema, y deberá brindar los servicios de consulta y reservación por parte de los usuarios. Además se estima que el sistema debe estar basado sobre alguna plataforma Web, para que desde cualquier navegador con acceso a Internet los usuarios puedan hacer uso de los servicios ofrecidos.

Las operaciones realizadas por el sistema serán:

- 3- Autenticación
- 4- Altas
- 5- Bajas
- 6- Consultas
- 7- Modificaciones
- 8- Reportes
- 9- Préstamos
- 10- Devoluciones

El sistema de la biblioteca requiere que se cumpla con ciertos requisitos para lograr un buen funcionamiento del mismo, donde todo sea manejable, accesible y fácil de ocupar. También se requiere que se presenten algunos reportes y actividades como lo son:

- Agilizar el préstamo.
- Agilizar la colocación del libro
- Clasificar los libros.
- Libros en existencia
- Devolución de libros
- Registrar las sanciones monetarias

## Diagrama ER (ENTIDAD - RELACION)

### 2.2.1 CASOS DE USO

El análisis anterior nos proporcionó con detalle los requerimientos del sistema, identificando que se identifican dos actores principales que son:

**Usuarios.** Puede ser cualquier estudiante o profesor de la escuela que puede acceder a la Biblioteca.

**Administrador.** Su función es la de gestión de la Biblioteca, así como el mantenimiento de la misma.

#### Diagrama de Actores

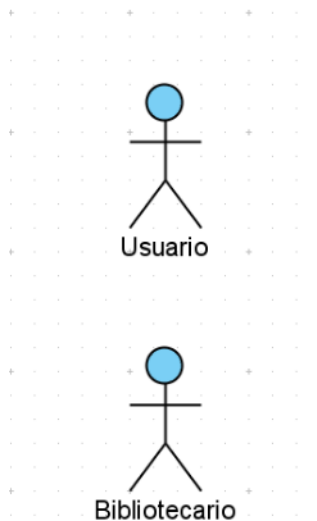


Figura 2.1

Aquí se muestran los actores principales de nuestro sistema, los cuales se encuentran sumamente relacionados en las acciones que determina el sistema de control de la Biblioteca, de los actores vistos se especificarán los diferentes privilegios con los que cuenta cada actor.

## Diagrama de Casos de Uso General

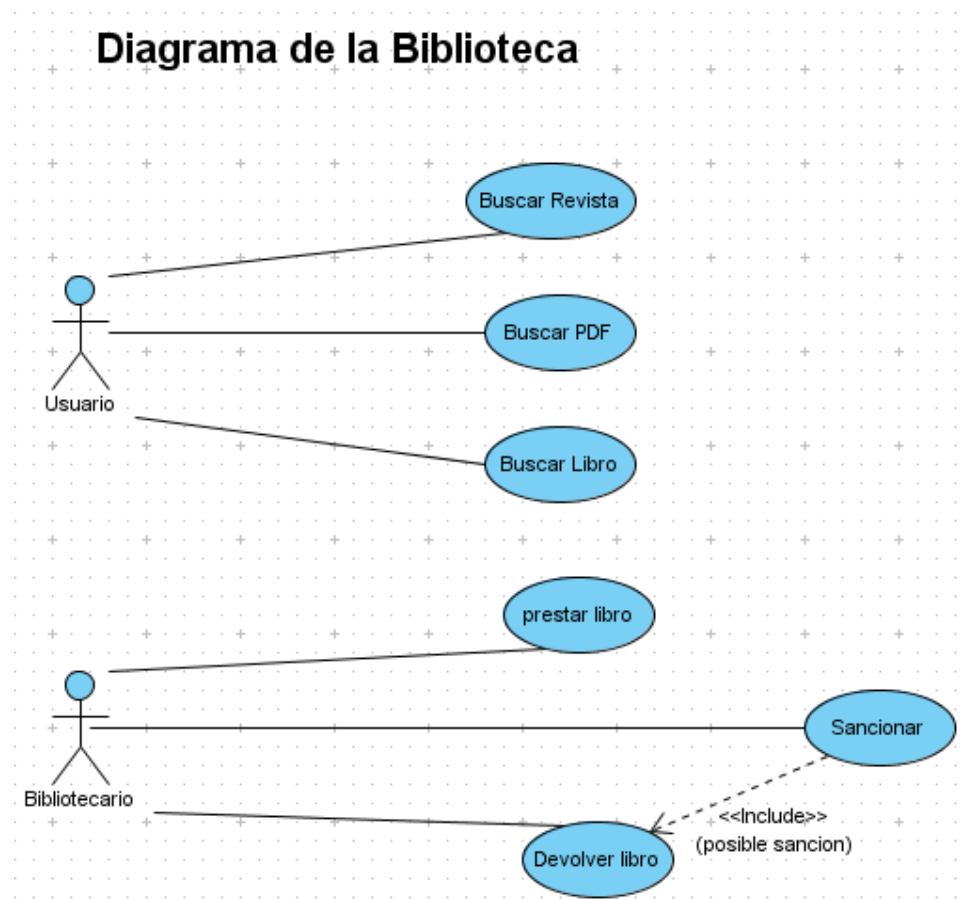


Figura 2.2

En el caso de la Biblioteca se pueden apreciar el contacto que se tiene entre el usuario y el administrador, ya que el usuario realizará primero una consulta y después acudirá con el administrador para solicitar el material anteriormente consultado, y el administrador (Bibliotecario) evaluará la condición de cada usuario según el sistema y determinará si se realiza la acción solicitada.

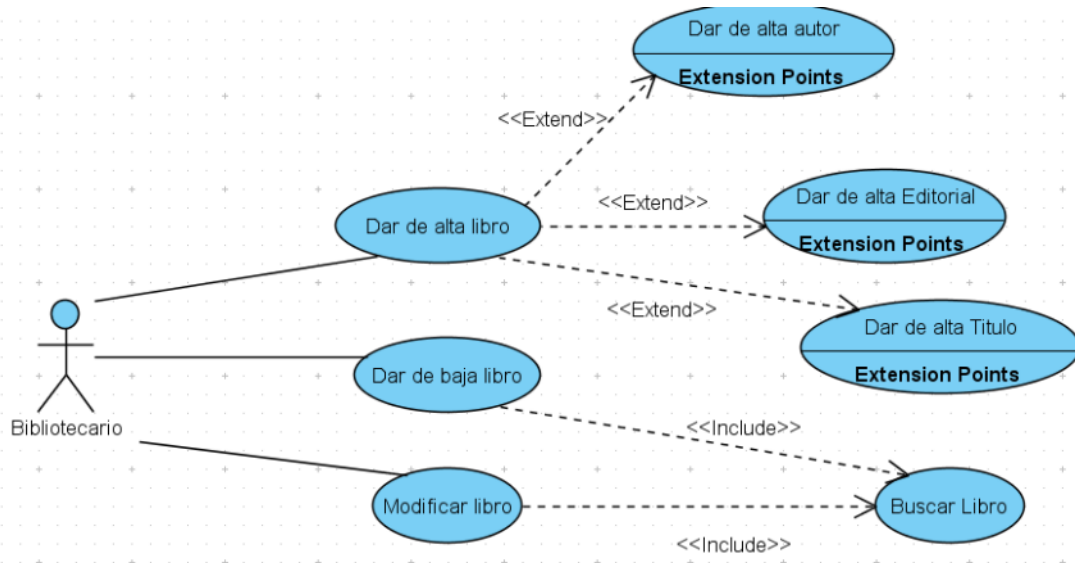


Figura 2.3

En el siguiente caso se describen las acciones del bibliotecario como son dar de alta un nuevo libro, lo cual implica dar de alta varios campos como son el de autor, editorial, título por mencionar algunos. Las consultas de acción son operaciones básicas en la vista del administrador que nos permitirán llevar un control del sistema.

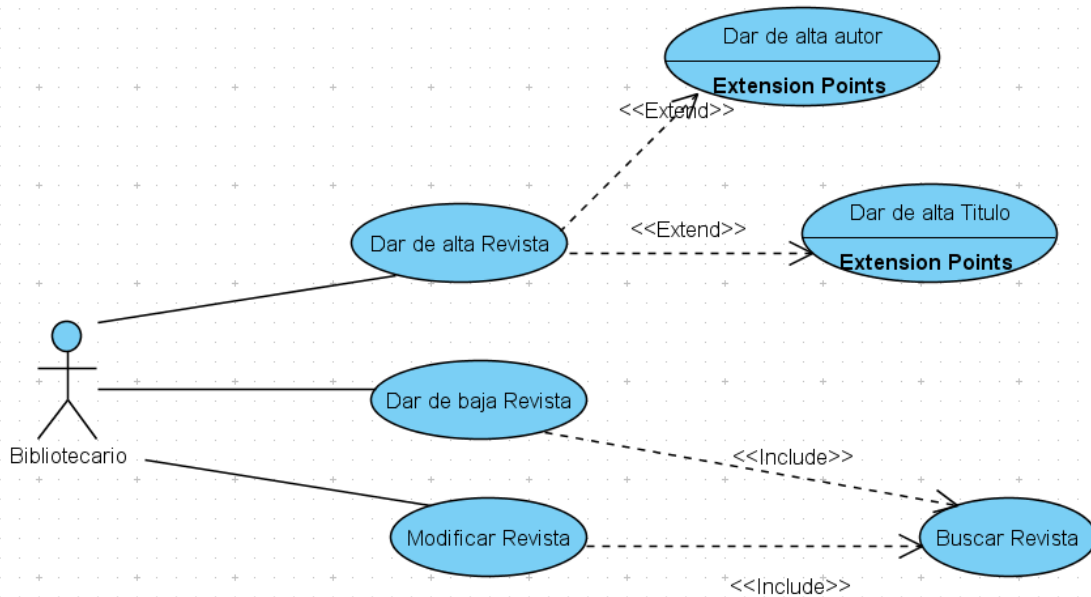


Figura 2.4

En el siguiente caso se describen las acciones del bibliotecario con respecto a la entidad revistas, de igual forma se presentan las consultas de acción alta, baja y modificar, ahora respectivamente hacia la entidad revista.

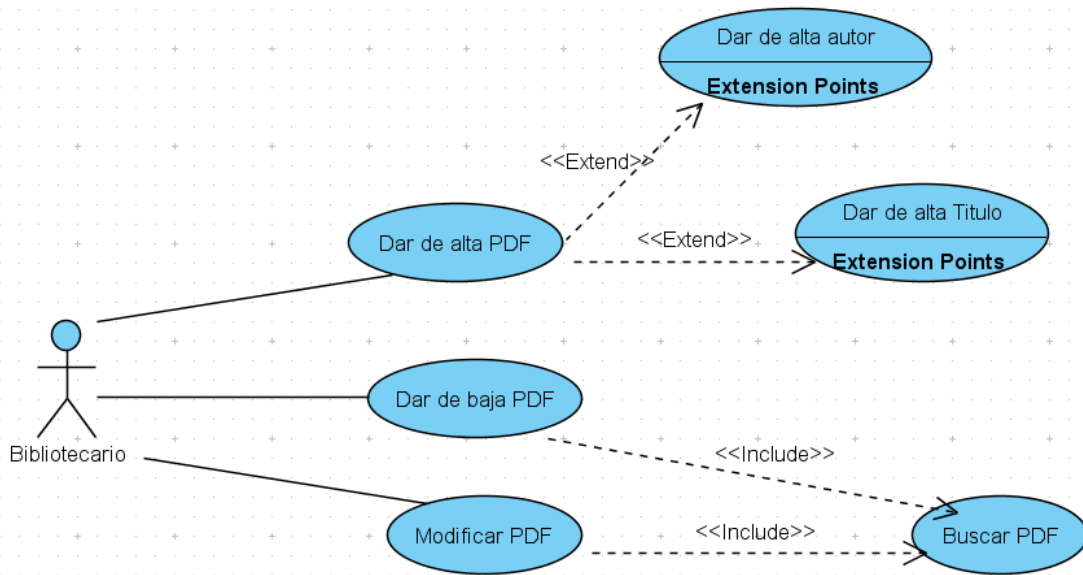


Figura 2.5

En el caso de Administrador de Libros (Bibliotecario) se puede agregar, eliminar y modificar los campos de los diferentes elementos como son: libros, revistas o pdf, tales como un género, un idioma, una editorial, etc., dependiendo del elemento.

## Gestion de Usuarios

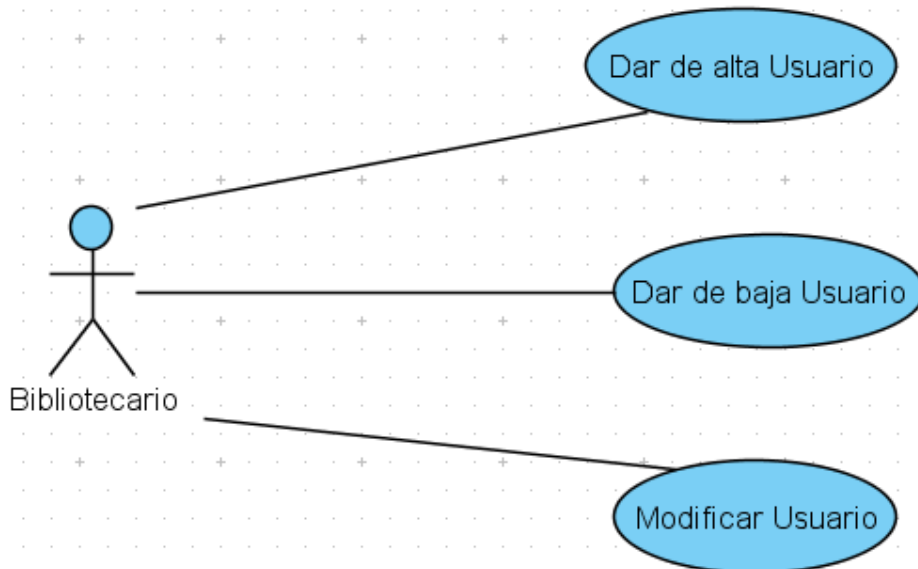


Figura 2.6

En el caso de Administrador de Usuarios (Bibliotecario) se puede agregar, eliminar y modificar los usuarios permitidos el acceso a la Biblioteca Digital.

Existen diferentes diagramas con los que cuenta UML, en la mayoría de los casos no se requieren de todos, y UML permite definir solo los necesarios, por lo tanto para el desarrollo este sistema se ocuparan los siguientes diagramas:

### Diagrama de Secuencia del sistema

El diagrama nos permite ver como se relacionan los objetos con el paso del tiempo, en cual se puede observar las operaciones que se realizan cuando el usuario requiere de una consulta y como el sistema responde a la petición.

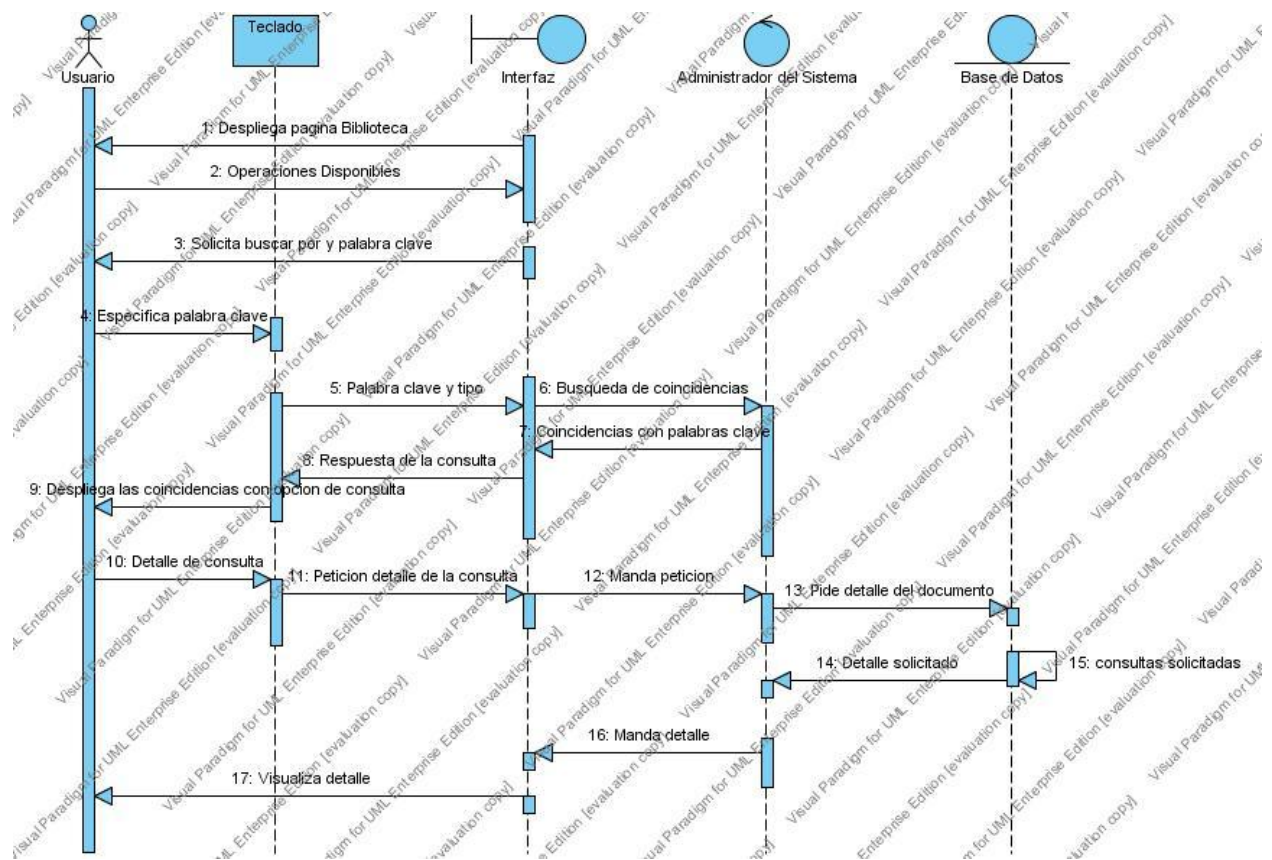


Figura 2.7 Diagramas de Secuencia “Búsquedas “

Este diagrama de secuencias nos permite ver las acciones que realiza el Administrador de Usuarios cuando se requiere de la operación agregar un nuevo usuario, de igual forma nos permite visualizar las acciones que realiza cuando se requiere de la operación eliminar un usuario y finalmente podemos observar las acciones que se realizan en la operación de modificar un usuario.

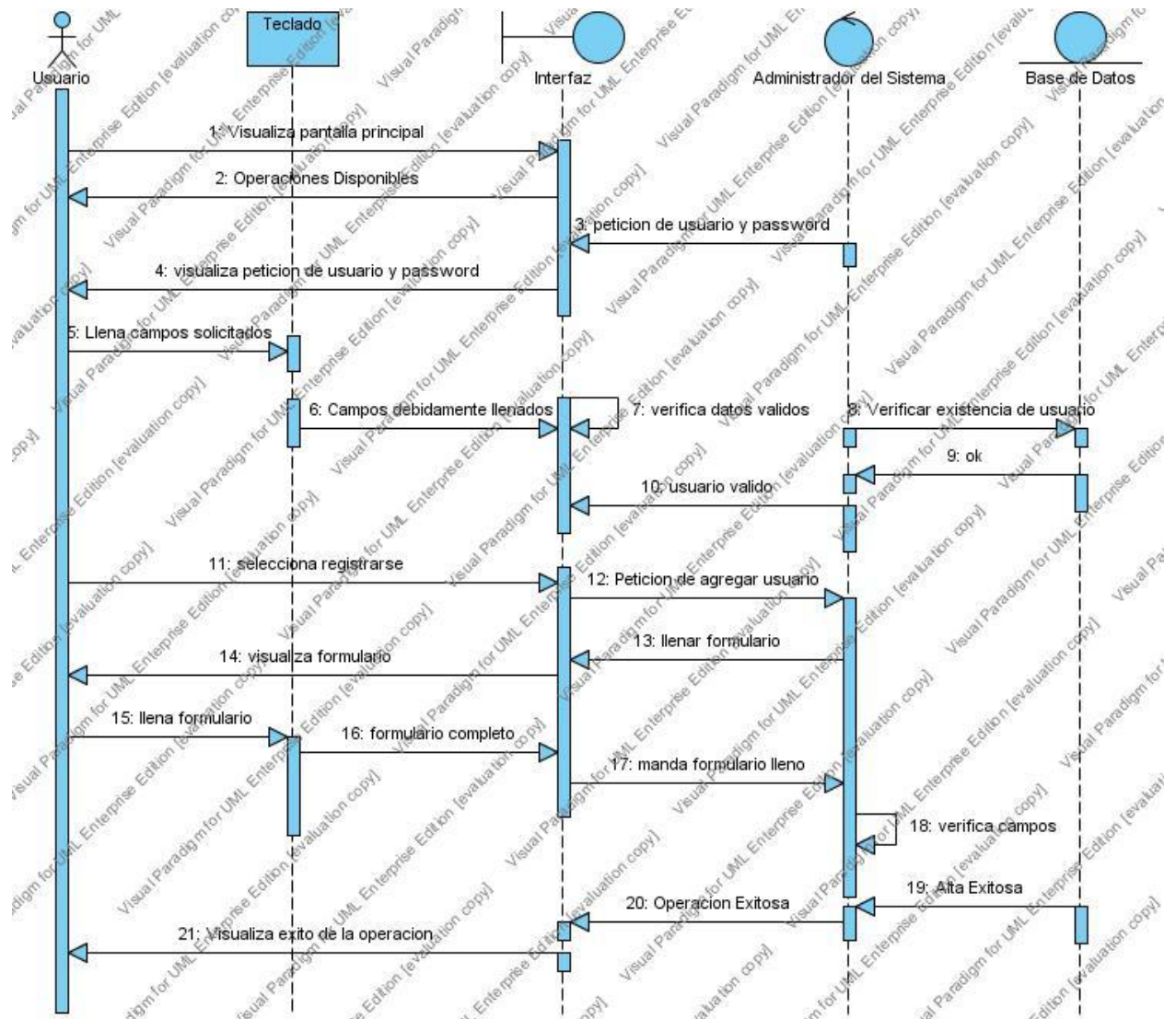


Figura 2.8 Diagrama de Secuencia “Administrador de Usuarios (Agregar usuario)”

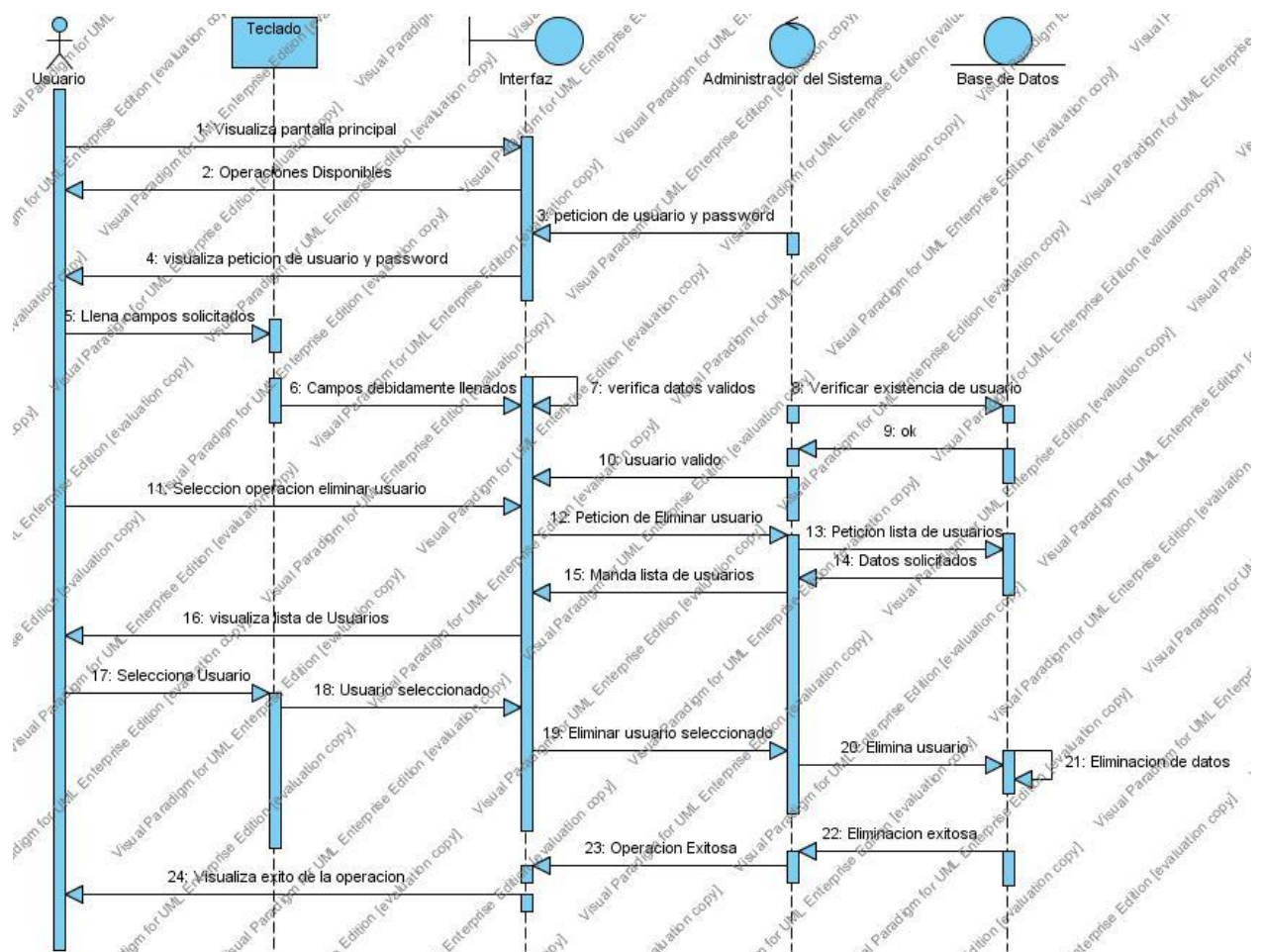


Figura 2.9 Diagrama de Secuencia “Administrador de Usuarios (Eliminar usuario)”

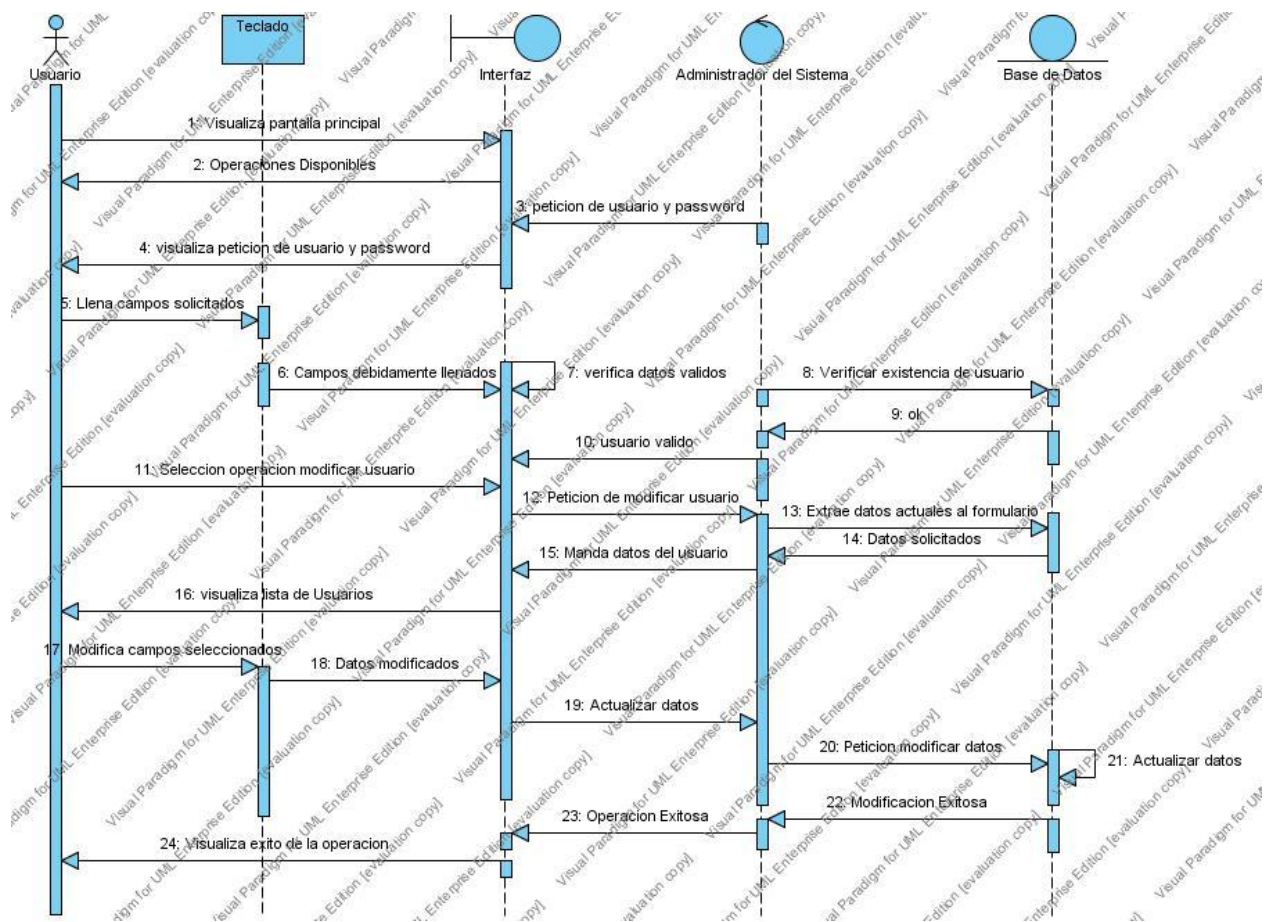


Figura 2.10 Diagrama de Secuencia “Administrador de Usuarios (Modificar usuario)”

## Diagramas de Maquina de Estados

El diagrama de la figura muestra el comportamiento del conjunto de estados por los cuales pasa un objeto durante su vida, junto con los cambios que permiten pasar de un estado a otro en alguno de los Administradores que conforman al sistema.

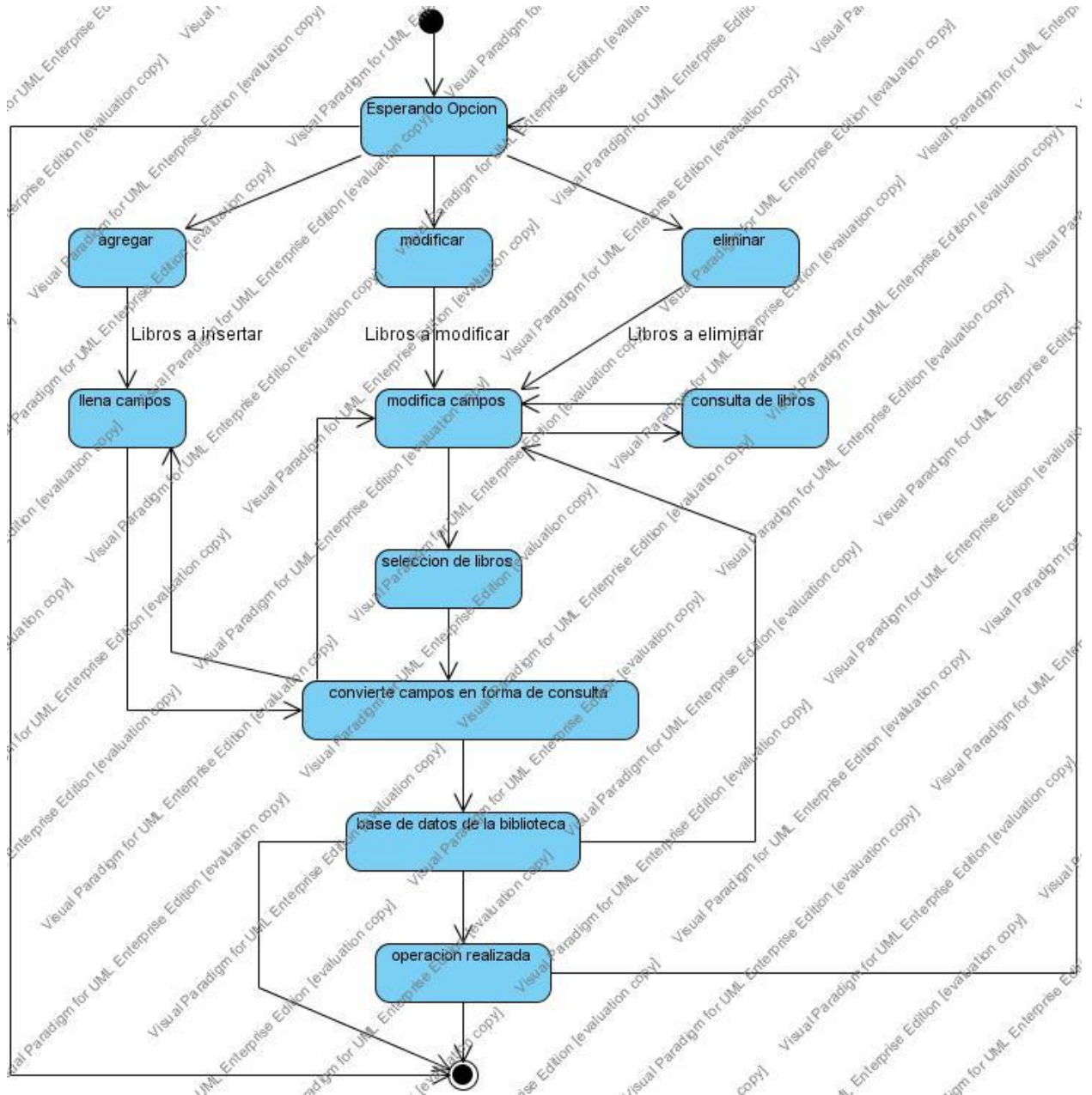


Figura 2.11

En el diagrama se presentan los estados en que puede encontrarse la Búsqueda de los diferentes documentos (Libros, Revistas, PDF) del sistema, también se pueden ver las transiciones entre los estados, mostrándose los puntos inicial y final de una secuencia de cambios de estado.

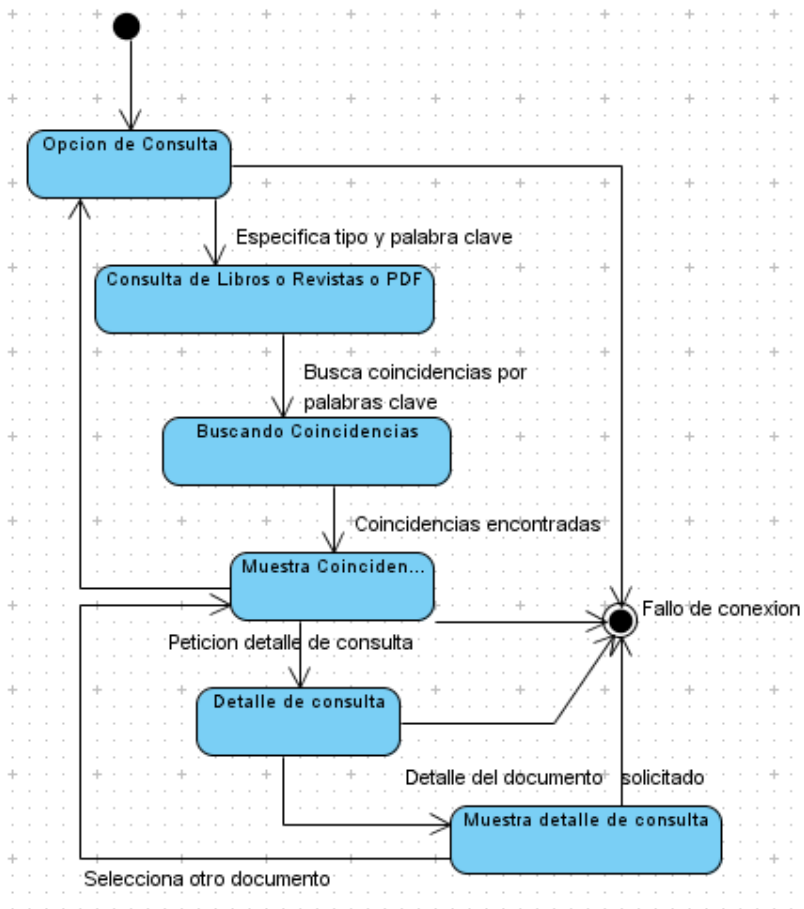


Figura 2.12

## Capítulo 3 DISEÑO DEL SISTEMA

### Diseño conceptual

En esta etapa debe construirse un esquema de la información que va a ser representada en la Base de Datos. A este esquema se le llama esquema conceptual, aquí se descubre la semántica de los datos de la empresa u organización. En particular se encuentran entidades, atributos y relaciones.

Esta es la etapa más importante en el desarrollo de cualquier aplicación de software, así se puede considerar al diseño conceptual como la base para las siguientes fases. Aquí es donde se pone en práctica toda la teoría de bases de datos, sus reglas y normalizaciones cabe mencionar que existen tres reglas básicas de normalización, lo cual se mencionaron en el Capítulo II.

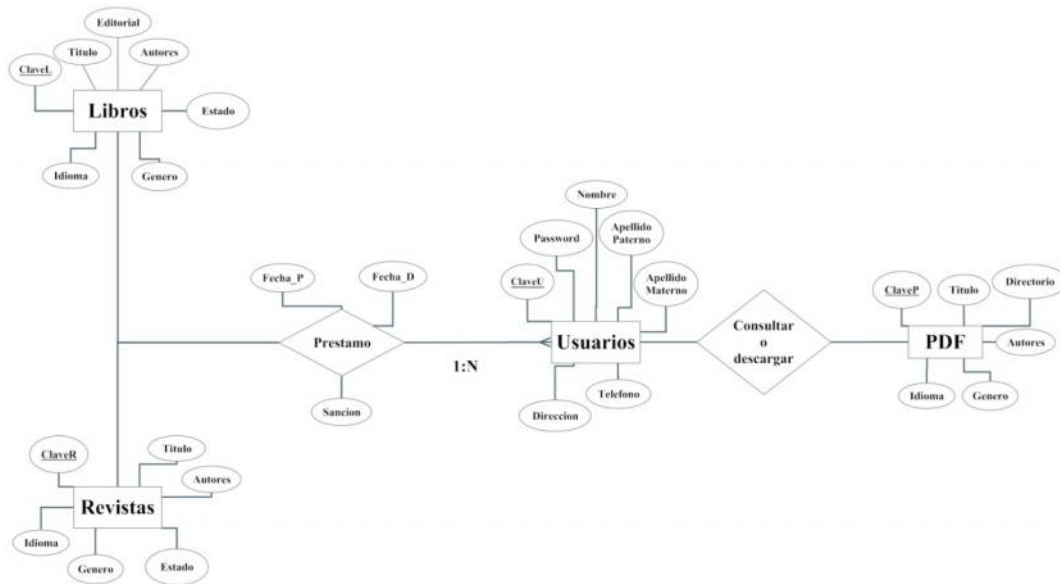
El modelo conceptual de datos con el que se va a trabajar es el modelo Entidad-Relación (E-R) que se presentan; pues es uno de los modelos más expresivos y con conceptos únicos, representado cada uno con un símbolo, lo cual conlleva a una fácil lectura y comprensión del diagrama, tanto por el analista de sistema, así como por el cliente o usuario.

El modelo de datos debe ser tal que nos permita describir la realidad de forma esquemática. Un buen modelo conceptual debe cubrir las siguientes cualidades:

- Expresividad
- Simplicidad
- Minimalidad
- Formalidad

**El diagrama entidad relación de la base de datos de muestra a continuación:**

## Diagrama de Entidad Relación



La entidad Usuarios contará con la siguiente información:

- Clave única de usuario y password de al menos 6 caracteres validos.
- Nombre de los usuarios que acceden al sistema.
- Dirección y teléfono de los usuarios, estos datos son opcionales.

En la entidad PDF nos dará la información de los documentos en formato PDF los cuales contienen título, genero, idioma y autor.

Si el documento es un libro o revista el atributo Editorial nos dirá a que editorial pertenece el documento.

El atributo Clasificación nos dirá en que clasificación está el documento.

Dependiendo del documento que se proporcione se tendrá una relación con las demás entidades, por ejemplo si se quiere almacenar un Libro, las entidades involucradas serian ClaveL(título del libro, nombre del autor del libro, editorial), la entidad Revista ClaveR( Título, autor), en cambio si el documento es un PDF las entidades involucradas serian, Título, Idioma, Género y Clasificación.

## Diccionario de datos

En el Diccionario de Datos se definen los elementos que conforman la base de datos de nuestro sistema.

### Biblioteca

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>	<b>Comentarios</b>
<u>RFC</u>	varchar(13)	No		
Nombre	varchar(50)	No		

### Libros

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>	<b>Comentarios</b>
<u>ClaveL</u>	varchar(10)	No		
Titulo	varchar(250)	No		
Editorial	varchar(250)	No		
Ano_Edicion	date	No		
Autor	varchar(250)	No		
Idioma	enum('Ingles', 'Español')	No		
Genero	enum('Ciencias Sociales', 'Computacion', 'Español', 'Geografia', 'Ingles', 'Matematicas', 'Quimica')	No		
Estado	enum('1', '0')	No		

### Pdf

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>	<b>Comentarios</b>
<u>ClaveP</u>	int(10)	No		
Titulo	varchar(250)	No		
Autor	varchar(250)	No		

Genero	enum('Ciencias Sociales', 'Computacion', 'Español', 'Geografia', 'Ingles', 'Matematicas', 'Quimica')	No
--------	--	----

Idioma	enum('Ingles', 'Español')	No
Directorio	varchar(250)	No

**prestamos\_libros**

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>	<b>Comentarios</b>
<u>ClaveL</u>	varchar(10)	No		
<u>ClaveU</u>	varchar(9)	No		
Fecha_P	date	No		
Fecha_D	date	Sí	<i>NULL</i>	
DiasP	varchar(2)	No		7
Sancion	double	Sí		0

**préstamos\_revistas**

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>	<b>Comentarios</b>
<u>ClaveR</u>	varchar(10)	No		
<u>ClaveU</u>	varchar(9)	No		
Fecha_P	date	No		
Fecha_D	date	Sí	<i>NULL</i>	
Sancion	double	Sí	<i>NULL</i>	

**Revistas**

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>	<b>Comentarios</b>
<u>ClaveR</u>	varchar(10)	No		
Titulo	varchar(250)	No		
Autor	varchar(250)	No		
Ano_Edicion	date	No		
Idioma	enum('Ingles', 'Español')	No		

Genero	enum('Ciencias Sociales', 'Computacion', 'Español', 'Geografia', 'Ingles', 'Matematicas', 'Quimica')	No
Estado	enum('1', '0')	No

### Usuarios

Campo	Tipo	Nulo	Predeterminado	Comentarios
ClaveU	varchar(9)	No		
Password	varchar(10)	No	1111111111	
Nombre	varchar(20)	No		Nombre del Alumno
ApellidoP	varchar(10)	No		Apellidos del Alumno
ApellidoM	varchar(10)	No		
Direccion	varchar(30)	No		Direccion del Alumno
Telefono	varchar(10)	No		Telefono del Alumno
Email	varchar(20)	No		
LibrosP	int(1)	No	0	

### Normalización

El proceso de normalización consiste en verificar el cumplimiento de ciertas reglas que aseguran la eliminación de redundancias e inconsistencias, que a continuación se realiza a las tablas que se mapearon del diseño conceptual.

#### Usuarios

ClaveU	Password	Nombre	ApellidoP	ApellidoM	Direccion	Telefono	Email	LibrosP
<b>PK</b>							<b>FK</b>	<b>FK</b>

#### Libros

ClaveL	Titulo	Editorial	Ano_Edicion	Autor	Idioma	Genero	Estado
<b>PK</b>							<b>FK</b>

#### Revistas

ClaveR	Titulo	Autor	Ano_Edicion	Idioma	Genero	Estado
<b>PK</b>						<b>FK</b>

### PDF

ClaveP	Titulo	Autor	Genero	Idioma	Directorio
--------	--------	-------	--------	--------	------------

**PK**

### Prestamos\_Libros

ClaveU	ClaveL	Fecha_P	Fecha_D	DiasP	Sancion
--------	--------	---------	---------	-------	---------

**PK**

### Prestamos\_Libros

ClaveU	ClaveR	Fecha_P	Fecha_D	DiasP	Sancion
--------	--------	---------	---------	-------	---------

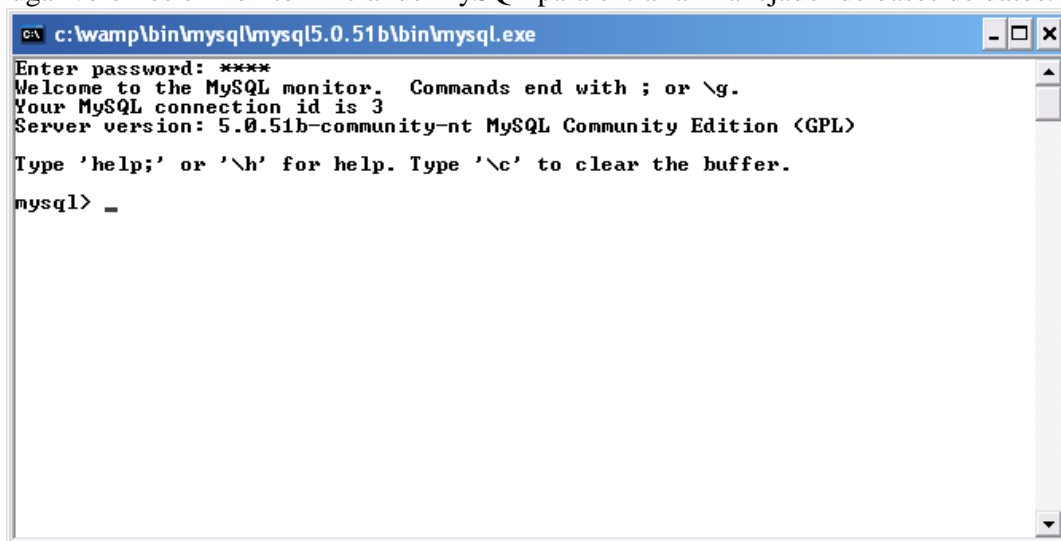
**PK**

Las tablas están en primera forma normal ya que los atributos son atómicos. También están en segunda forma normal porque están en primera forma normal y la llave primaria no tiene múltiples atributos y los atributos no llave dependen solamente de la llave primaria, y por último las Tablas están en tercera forma normal ya que está en segunda forma normal y no existen dependencias funcionales entre los atributos no llaves.

## Diseño Físico

En lo que respecta al diseño físico hay que llevar a cabo una serie de pasos que describan la implementación de la base de datos, lo que incluye estructuras de almacenamiento y los métodos de acceso que se utilizarán para conseguir un acceso eficiente a los datos.

En el estudio de este sistema y concretamente el SGBD se puede ver el soporte de la creación de una base de datos, tablas, llaves primarias y foráneas, las cuales llevan a cabo las uniones entre las tablas para poder llevar a cabo las consultas. También el sistema soporta la definición de datos requeridos y la definición de dominios. En primer lugar veremos el monitor inicial de MySQL para entrar al manejador de bases de datos.



```
CA c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.51b-community-nt MySQL Community Edition <GPL>

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
```

## Capítulo 4 IMPLEMENTACION DEL SISTEMA

Una vez terminada la etapa de análisis y diseño del Sistema, se realiza la implementación del mismo a través de un sistema administrador de bases de datos, lenguajes de programación y herramientas de diseño para su funcionamiento.

### 4.1 Manejador de Bases de Datos

El administrador de bases de datos que se utilizó fue MySQL, ya que es sencillo de usar y es muy rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Algunas de las características principales de MySQL son:

- Es un gestor de base de datos.
- El código fuente se puede descargar y está accesible a cualquier usuario.
- Es un programa interactivo que permite conectarnos a un servidor, ejecutar consultas y ver los resultados.
- Integración con cualquier lenguaje de programación.
- Estandarización.
- Fácil y rápido aprendizaje.

### 4.2 Lenguaje de Programación

El lenguaje de programación que se utilizó para desarrollar el Sistema fue PHP, ya que es un lenguaje de programación usado frecuentemente para la creación de sitios Web, se pueden programar las páginas html y los códigos fuente y es usado para la creación de aplicaciones para servidores.

Debido al diseño de PHP, también es posible crear aplicaciones con una interfaz gráfica para el usuario, su interpretación y ejecución se da en el servidor web, en el cual se encuentra almacenado el script y el cliente sólo recibe el resultado de la ejecución.

## *Ventajas de PHP*

- Muy sencillo de aprender.
- Similar en sintaxis a C y a PERL
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática, librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son validos se hace en el servidor y no en el cliente (como se hace con javascript ) de forma que se puede evitar que chequear que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía HTTP.

## *Por que utilizar PHP y no otras opciones:*

- Se pueden hacer grandes cosas con pocas líneas de código, lo que hace que merezca la pena aprenderlo.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor ( acceso a base de datos, encriptación, envío de correo, gestión de un e-commerce, xml, creación de PDF ... ).
- Al poderse encapsular dentro de código html se puede recoger el trabajo del diseñador gráfico e incrustar el código php posteriormente.
- Está siendo utilizado con éxito en varios millones de sitios web.
- Hay multitud de aplicaciones php para resolver problemas concretos ( weblogs, tiendas virtuales , periódicos , ... ) , listas para usar.
- Es multiplataforma, funciona en todas las plataformas que soporten apache.
- Es software libre. Se puede obtener en la web y su código está disponible bajo la licencia GPL.

## **4.3 Herramientas de Diseño**

Para realizar el diseño la herramienta utilizada fueron:

Macromedia Dreamweaver CS3:

- Es un editor de HTML visual
- Hace muy fácil el crear páginas
- Genera HTML dinámico y el código resultante es compatible con las últimas versiones de los navegadores actuales.

Macromedia Flash CS3:

Macromedia Flash es una herramienta de creación de páginas Web que permite crear sitios ricos en gráficos y animaciones con más control sobre estos gráficos. Los gráficos y animaciones creados con Flash son mucho más pequeños que los formatos

tradicionales (gifs animados, avi, etc) porque utilizan vectores en vez de mapas de bits. (Los mapas de bits contienen instrucciones que dividen el gráfico en puntos e indican al ordenador qué se debe hacer con cada uno de estos puntos, en cambio los vectores contienen las instrucciones que son una especie de fórmulas para dibujar líneas y sus coordenadas).

#### 4.4 Funciones importantes del sistema

##### *Función de conexión de MySQL con PHP*

Para establecer una conexión con MySQL desde PHP se realizó la función “conectarse”, dicha función devuelve un link que será el vínculo que permite la comunicación con la base de datos, que ayudará a realizar diversas actividades como hacer consultas, agregar, eliminar y actualizar la base de datos.

```
<?php

function Conectarse()
{
    if (!($link=mysql_connect("localhost","root","root")))
    {
        echo "Error conectando a la base de datos.";
        exit();
    }
    if (!mysql_select_db("biblioteca",$link))
    {
        echo "Error seleccionando la base de datos.";
        exit();
    }
    return $link;
}

$link=Conectarse();

mysql_close($link); //cierra la conexion
?>
```

##### *Función de “Buscar”*

La función buscar nos permite realizar la búsqueda de alguna de las tres entidades disponibles libros, revistas y PDF, de ahí que nuestra búsqueda se debe ir filtrando hasta encontrar los parámetros por los cuales se necesita que se despliegue dicha búsqueda.

```
<?php
include("conexion.php");
$link=Conectarse();
if($_POST['Buscar']){
    $busca=$_POST['busca'];
    $por=$_POST['por'];
    $palabra=$_POST['palabra'];
```

```

$result=mysql_query("select * from $busca where $por like '%$palabra%'", $link) or
die (mysql_error());
}
?>

```

## 4.5 Interfaz Sistema

En la figura 4.1 se muestra la pantalla inicial del Sistema de nombre Biblioteca Pública, en esta pantalla se muestra las opciones que tiene disponible el usuario, con una opción para el administrador.



Figura 4.1

En la figura 4.2 se muestra la pantalla donde el usuario puede hacer búsquedas de libros, revistas y documentos electrónicos, con sólo introducir lo que se requiere buscar y seleccionar que tipo de búsqueda e, y el sistema le mostrará las coincidencias encontradas.

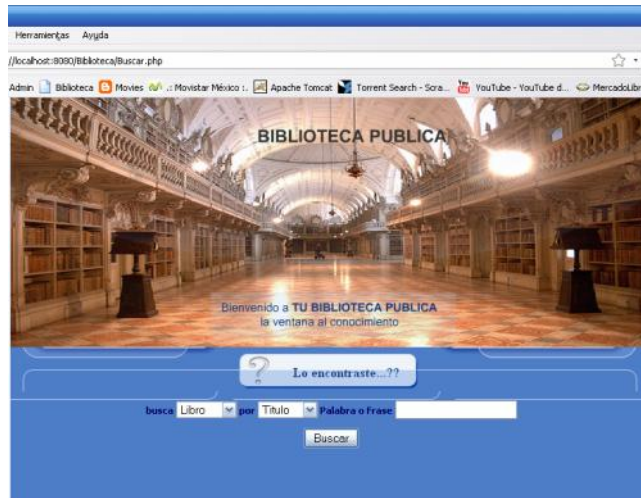


Figura 4.2

En la figura 4.3 se muestra la pantalla de acceso al sistema referente a los usuarios inscritos en el portal Web Biblioteca Pública.



Figura 4.3

En la figura 4.4 se muestra la pantalla de acceso al sistema referente al administrador del mismo, se muestra que se requiere de una cuenta y contraseña para entrar a la Biblioteca Pública en modo administrador.



Figura 4.4

En la figura 4.5 se muestra la pantalla de acceso al sistema referente al administrador del sistema, se muestra que se requiere de una cuenta y contraseña para entrar a la Biblioteca Publica en modo administrador.

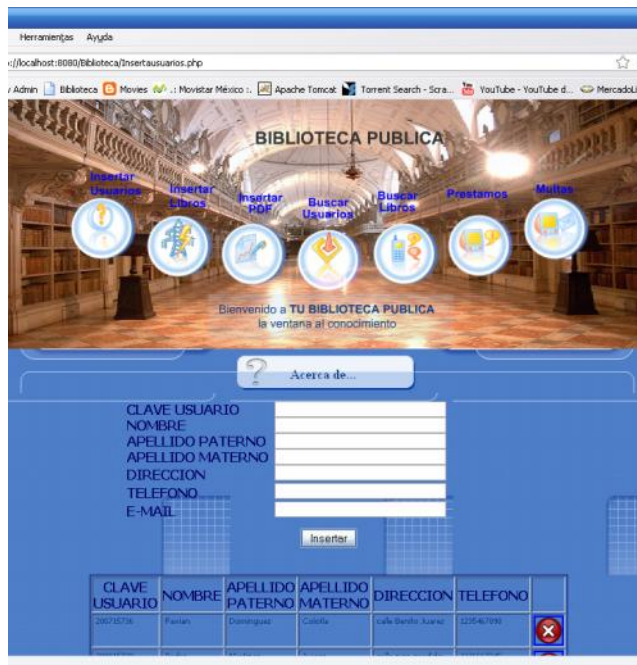


Figura 4.5

## Capítulo 5 PRUEBAS

En el capítulo IV se presentó la implementación del sistema, y las pantallas (interfaz) que lo conforman, en este capítulo se presentan algunas pruebas para ver el comportamiento del mismo y por último el manual del usuario.

### 5.1 Pruebas

#### Búsqueda de Documentos

Se presenta en la figura 5.1 un ejemplo de una consulta dentro del sistema y los resultados que se proporcionan (figura 5.2), en este caso se hace una consulta por título.

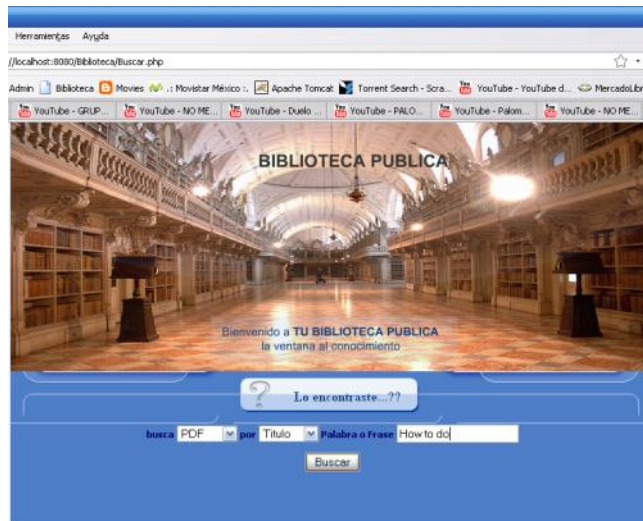


Figura 5.1 Usuario introduce palabras claves para su búsqueda



Figura 5.2 Resultados de la búsqueda

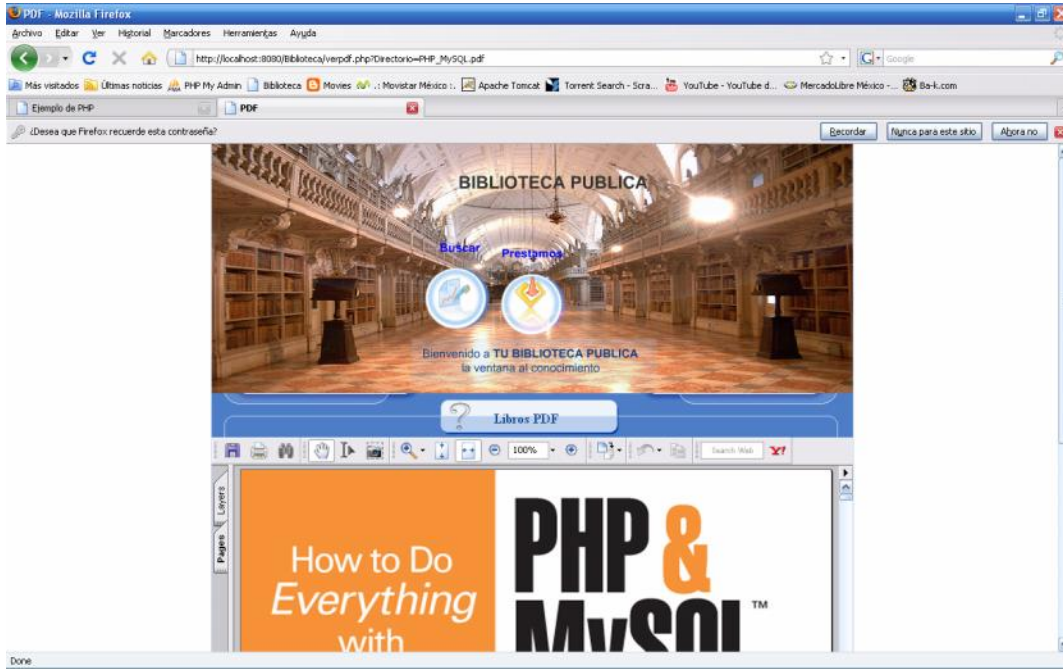


Figura 5.3 Documento PDF desplegado en la pagina

## Conclusión

El sistema está listo para usarse cumpliendo con el objetivo general y los objetivos particulares y se apega a los requisitos planteados:

- Todos los documentos tienen un género, título, autor y idioma.
- El sistema cuenta con consultas para acceder a los documentos, libros y revistas.
- El sistema puede calcular diariamente las sanciones correspondientes a cada usuario, de acuerdo a las correspondientes fechas.

El sistema cuenta con una interfaz amigable, con menús de selección para agregar, modificar y eliminar los elementos o PDF rápidamente.

El sistema sería una alternativa más para el estudiante ya que pone a su disposición información para consulta e investigación.

## Perspectivas

- El sistema tiene acceso al público en general, sólo la consulta y descarga de PDF, pero si se requiere de algún libro o revista deberá acudir con el administrador de la biblioteca.

## ANEXO

### Manual del Sistema

En este punto se describen cada tipo de usuario del sistema y las operaciones que tiene disponible el administrador.

### Administrador

Como se ha mencionado anteriormente este se encarga de la administración de los documentos PDF, así como de los libros y las revistas.

### Agregar Libro

En agregar libro (Figura 5.4) se describen los datos del libro y en el caso de registrar un nuevo ejemplar, se procederá a llenar un formulario, mediante el cual se recopila información que facilite las búsquedas de los usuarios y se almacena en la base de datos del sistema, en este formulario todos los campos son requeridos ya que no pueden existir datos nulos, además se enlistan abajo los registros insertados para poder corroborar que el registro fue exitosamente insertado.

CLAVE LIBRO	TITULO	EDITORIAL	AÑO_EDICION	AUTOR	IDIOMA	GENERO
200410002	Geografía de México	Edigna	2002-09-11	Ana María Peralta	Español	Geografía
200411000	Matemática	Paray	2003-09-10	P.	Español	Matemáticas

Figura 5.4 Formulario de insertar libro

### Eliminar un Usuario

Cuando requerimos de eliminar algún usuario por alguna razón se utiliza esta opción, se deberá primeramente hacer la búsqueda del registro deseado, proporcionando la clave del usuario, dicha búsqueda nos mostrará una tupla con toda la información correspondiente al usuario seleccionado, y bastará simplemente dar un clic sobre el botón de eliminar y automáticamente se eliminará la tupla que contiene a nuestro usuario consultado.

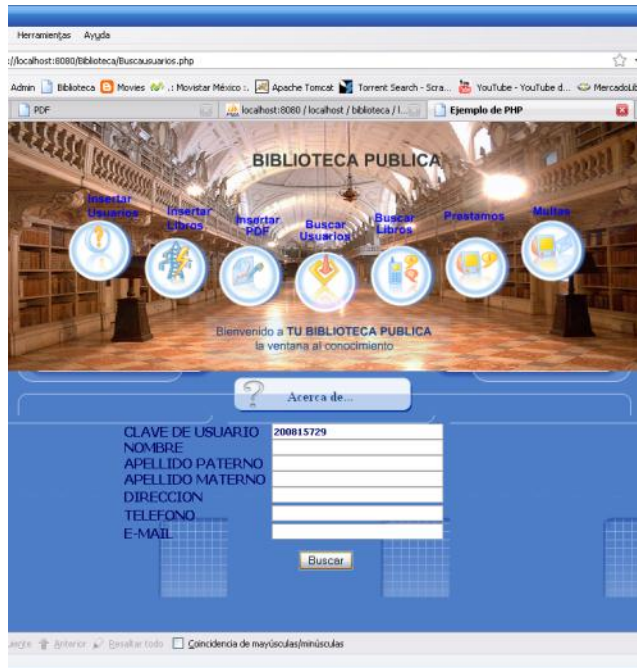


Figura 5.5 Búsqueda de usuarios

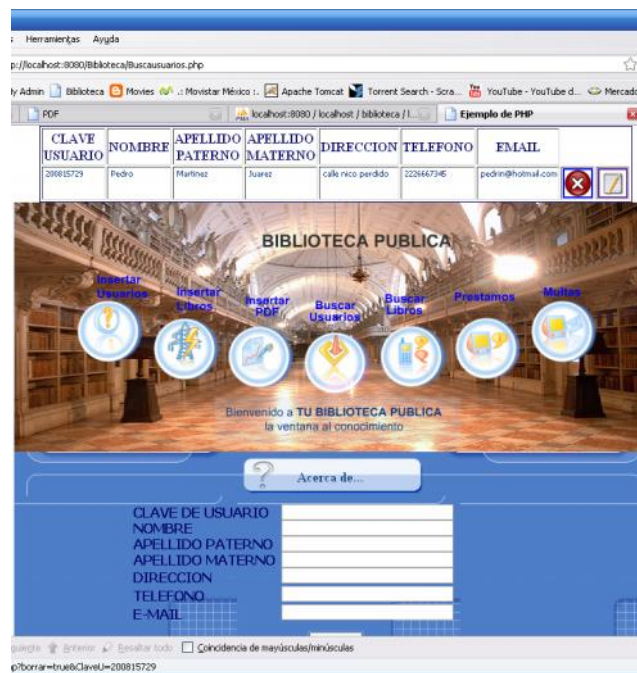


Figura 5.6 Pantalla eliminar usuario seleccionado

## Bibliografía

[1] <http://www.uoc.edu/masters/esp/img/873.pdf>

[2] **Elmasri, R.; Navathe, S.B.:** “Sistemas de bases de datos, Conceptos Fundamentales”, Ed.: Addison-Wesley Iberoamericana, 3a ed. Madrid, pp. 25-27, 411-422, 2000.

[3] **Miguel, A. de, Piattini, M. y Marcos, E.:** “Diseño de Bases de Datos

[4] **Jesús Bobadilla S. Alonso.:** “HTML Dinámico a través de ejemplos”, Ed. Alfaomega-rama, 2000.

[5]. Miguel, A. de y Piattini, M. (1997): Fundamentos y Modelos de Bases de Datos. Ed. RAMA, Madrid.

[6]. Introducción a los Sistemas de Bases de Datos, C.J. Date, Addison Wesley, Volumen 1 Quinta Edición

[7]. Diseño de bases de datos: El Proceso de Normalización. Carlota Tamayo Islas, María del Rocío Boone rojas.

[8]. [http://www.salnet.com.ar/inv\\_mysql/pag01\\_intro.htm](http://www.salnet.com.ar/inv_mysql/pag01_intro.htm)

Páginas Visitadas.

[9]. <http://www.php.net/>

[10]. <http://www.mysql.com/>

## ÍNDICE

### Tabla de contenido

<b>INTRODUCCIÓN</b> .....	2
Capítulo 1 MARCO TEÓRICO .....	4
1.1. HISTORIA DE LAS APLICACIONES WEB.....	4
Capítulo 2 ANÁLISIS DEL SISTEMA.....	17
Capítulo 3 DISEÑO DEL SISTEMA .....	29
<b>Diagrama de Entidad Relación</b> .....	30
<b>Diccionario de datos</b> .....	31
<b>Normalización</b> .....	33
Capítulo 4 IMPLEMENTACION DEL SISTEMA.....	35
Capítulo 5 PRUEBAS.....	41
<b>Bibliografía</b> .....	46