Capítulo 4

Implementación

Una parte importante en la construcción de una base de datos es la provisión de las herramientas que permitan tanto el almacenamiento, como la recuperación de la información sin olvidar el mantenimiento de la misma [Silberschatz et al. 1999].

De ahí la existencia en este proyecto de los módulos de administración de bases de datos, así como el módulo de consulta y navegación, último que permitiría precisamente la interacción de la base de datos con el usuario final.

Para facilitar la integracióndel sistema propuesto con los componentes de U-DL-A, y considerando la posibilidad de que el sistema sea utilizado vía web, se empleó el lenguaje de programación Java, además de que dicho lenguaje permitía una fácil integración con el manejador de bases de datos MySQL.

En las siguientes secciones, explicaremos a detalle aquellos módulos que involucraron aspectos importantes en cuanto a su implementación, la construcción de la base de datos y la implementación de los algoritmos propuestos.

4.1 Construcción de la base de datos

El manejador de bases de datos utilizado fue MySQL, debido a la disponibilidad, facilidad de uso y característicascomo mayor rapidez en operaciones básicas de lectura e inserción (Tabla 4.1), conectividad y seguridad para acceder bases de datos en Internet entre otra gran variedad de funciones que lo hacen una de las mejores opciones actualmente.

En la Tabla 4.1 se muestra la comparación de algunas funciones y los tiempos de acceso correspondientes a esas funciones de tresmanejadores de bases de datos incluyendo MySQL.

Tabla 4.1 Funciones y tiempos de acceso a bases de datos [MySQL].

Manejador de Base de Datos	Tiempo de lectura(seg.)	Tiempo de inserción(seg.)
MySQL	367	381
Informix_odbc	121126	2692
Oracle_odbc	20800	11291

Se utilizó a su vez el estándar de conexión a bases de datos JUDBC (Java Universal DataBase Connectivity) (http://ict.pue.udlap.mx/private/index.html) del laboratorio de ICT yel lenguaje estructurado de consultas SQL.

En base al diseño propuesto en el capítulo 3, se construyó la base de datos que contendría la información referente al acervo franciscano siguiendo criterios que ayudarían a agilizar el proceso y que se detallarán en las siguiente sección.

4.2 Módulos del sistema

El sistema conformado por los módulos de administraciónyconsulta- navegación fue implementado en su totalidad utilizando el lenguaje de programación Java. Cada módulo representóun grado de dificultad de acuerdo a la funcionalidad a la que estaban orientados.

La implementación del módulo de consulta y navegaciónse basó principalmente en la interacción con la base de datos construida.

Sin embargo, para que el módulo anterior tuviera funcionalidad, era indispensableproveer de un módulo que permitiera almacenar información en la base de datos. De ahí que se generara lo que nosotros denominamos módulo de administración.

Para ambos módulos del sistema, se utilizó unainterfaz gráfica, la cual se caracteriza por la utilización conjunta de los estilos mencionados en el apartado 3.5, a excepción de la entrada en línea de comandos debido a sus características intrínsecas, que no facilitarían en mucho la labor de un usuario. Se emplearon a su vezotros componentesgráficos que forman parte del leguaje Java y que son a grandes rasgos los siguientes:

- Contenedores primarios: como applets, diálogos y frames.
- Contenedores secundarios: paneles con y sin lengüetas, paneles con barras de desplazamiento.
- Componentes básicos: botones, etiquetas, casillas de selección, listas, campos y áreas de texto, entre otros.

4.2.1 Módulo de administración

El módulo de administración permite al administrador de la base de datos almacenar, actualizar y dar mantenimiento a la información contenida en esta.

Este módulo está conformado por cuatro componentes que son:

- Componente de altas: que permite almacenar información en la base de datos.
- Componente de bajas: empleado para eliminar información de la base de datos.
- Componente de actualización: permite modificar cierta información de la base de datos.
- Componente de consultas: utilizado para ver la información existente en la base de

datos.

Según el diagrama entidad-relaciónde la Figura 3.2 mostrada en el capítulo anterior, se tienen5 entidades, las cuales determinarían los encabezados de los menús del módulo de administración con 4 acciones asociadas a estosy que son las operaciones básicas que se realizan sobre una base de datos: altas, bajas, consultas y actualizaciones de información (Figura 4.1).

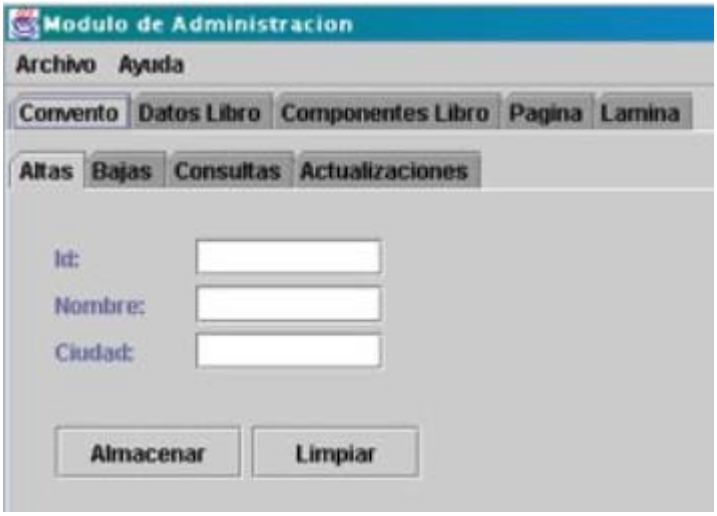

Figura 4.1 Interfaz de administración de la base de datos.

En la misma figura podemos observar que la interfaz maneja formas de entrada, y botones que lanzarán acciones según la pestaña donde el usuario esté posicionado.

La interfaz muestra una estructuración sencilla , que permite al usuario entender fácilmente las acciones a realizar. El sistema en cierta manera va guiando al usuario mediante la utilización de diálogos que le indican cuando un proceso ha concluido, el estado de dicho proceso o si tuvo algún error al momento de ejecutar alguna acción.

A su vez, se utilizó un lenguaje sencillo en cuanto a las etiquetas de botones que indicaban de forma clara y precisa la acción a realizar, evitando así la confusión del usuario.

4.2.1.1 Componente de altas

Antes de almacenar la información en la base de datos fue necesario organizarla siguiendo la estructura que se muestra en la Figura 4.2.

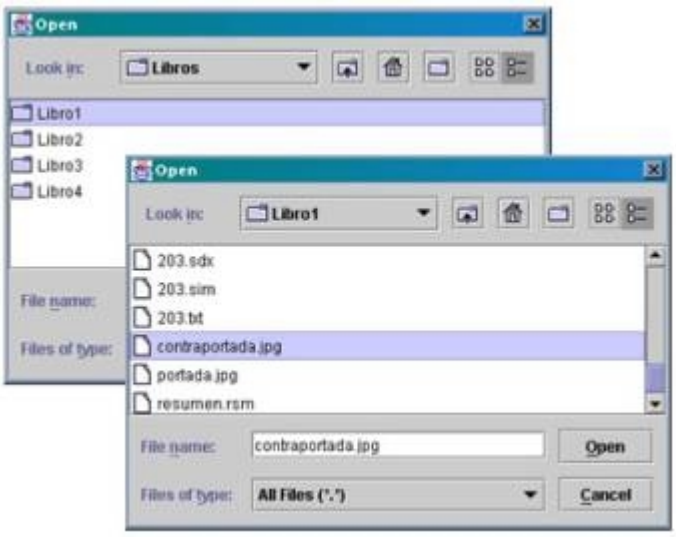

Figura 4.2 Organización de archivos utilizados.

Esto con la finalidad de facilitarle al usuario almacenar información a menor o mayor escala, es decir, por archivos o por fólder respectivamente. En esta figura se puede observar por ejemplo que la carpeta seleccionada etiquetada como Libro1, contiene archivos de tipo imagen correspondientes a portada y contraportada de ese libro así como otro tipo de archivos que pertenecen al mismo. Así, según la selección del usuario, siendo en este ejemplo el archivo etiquetado como "contraportada.jpg", únicamente este será almacenado en la base de datos.

También, como se puede observar en la Figura 4.2, la información correspondiente acada libro de la colección ,se almacenó dentro de una carpeta etiquetada con su identificador correspondiente, pudiendo ser este cualquier nombre que permitiera distinguir entre un libro y otro.

Esta información a su vez fue etiquetada considerando dos aspectos:

- tipo de archivo (de texto o imagen)
- nombre

Para los archivos de tipo texto a su vez se hizo una subdivisión de acuerdo al tipo de información que contendría, de tal manera que existiera una correspondencia con los datos almacenados en la base de datos. Esta subdivisión se puede observar en la Tabla 4.2 con respecto al tipo de extensión asignado.

Tabla 4.2 Tipos de archivo y extensión

Tipo de información	Extensión
Codificación “Soundex”	.sdx
Codificación “Clases de Caracteres”	.chc
Codificación “Similarex”	.sim
Pie de lámina	.dsc
Texto OCR	.txt
Resumen	.rsm
Imagen a color	.jpeg o .jpg
Imagen blanco y negro	.tiff

A continuación describiremos brevemente el contenido de cada tipo de archivo:

-Archivos de extensión .sdx, .chc, .sim: correspondientes a la codificación de los algoritmos propuestos en el capítulo 2 y sobre los cuales se realizarán las búsquedas de información. Estos archivos son generados dinámicamente por el sistema.

-Archivo .dsc: se refiere al pie de lámina de las figuras contenidas en algunas

páginas de los libros, sobre las cuales también se realizarán búsquedas.

-Archivo .txt: contiene la información original generada previamente con OCR.

-Archivo .jpg o jpeg: formato de imagen a color correspondiente a cada página de un libro.

-Archivo .rsm: contiene la síntesis general de un libro, la cual puede o no existir de acuerdo al libro.

-Archivo .tiff: formato de imagen en blanco y negro correspondiente a cada página de un libro,y necesaria parael reconocimiento de caracteres.

Con respecto al nombre de los archivos, este se definió a partir del número de la página a la que se hacia referencia, valor asignado al momento de guardarel archivo que contenía el texto reconocido por OCR y que correspondía a cada página de un libro. Esto con la finalidad de facilitar el proceso de almacenamiento, y la creación de los archivos codificados,los cuales se generaban dinámicamente al momento de que el sistema detectaba aquellos que contenían el texto reconocido y cuya extensión era “.txt”.

Enla Figura 4.3 por ejemplo, se puede observar que los nombres de los archivos efectivamente corresponden a un número de página específico, y que se distinguen de otros archivos por su extensión.

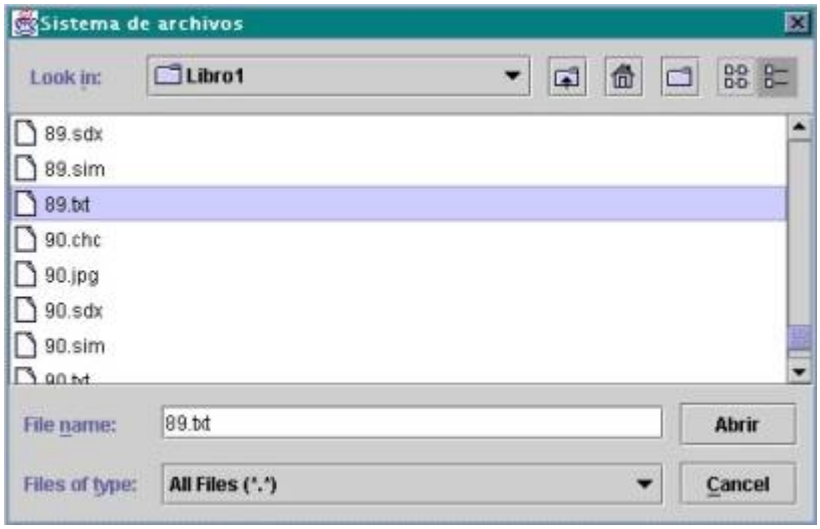

Figura 4.3 Ejemplo de nombres de archivo

En cuanto a la información de un libro que nosotros denominamos "información básica" en el capítulo 2 como lo es portada, contraportada, fue etiquetada precisamente con esos títulos para poder hacer la distinción, debido a que son archivos de tipo imagen cuya extensión es jpeg o jpg. Con respecto al índice de un libro, se tuvo que etiquetar con la leyenda “i” más un número secuencial que permitiría distinguir cuando existiera más de una página de índice y para el archivo correspondiente al resumen de un libro, este fue etiquetado con la leyenda “resumen”. En la Figura 4.4 se pueden observar algunos ejemplos.

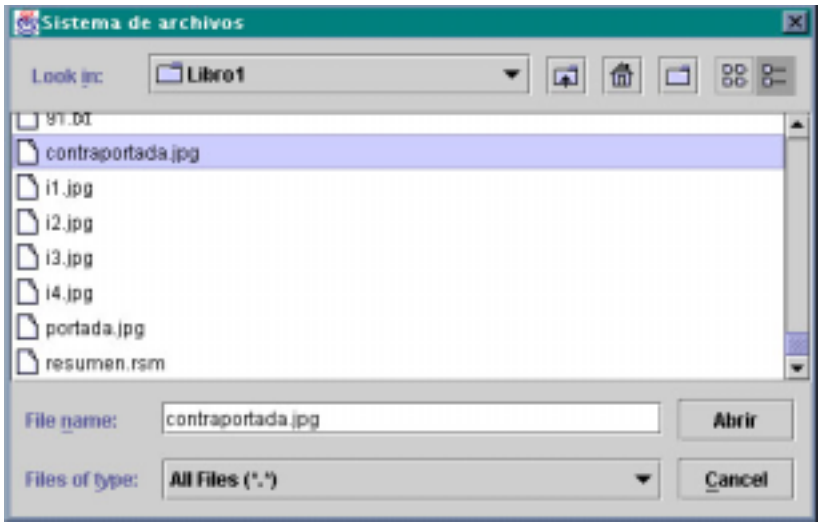

Figura 4.4 Ejemplonombres de archivo (información básica)

Finalmente, se definió este tipo de estructuración, para facilitarle al usuario la organización y localización de la información referente a cada libro, evitarle la asignación de nombres a cada tipo de archivo cada vez que tuviera que agregar información a la base de datos, y que pudiera en cierta forma generar conflictos o pérdida de tiempo al producir datos repetitivos**.**

4.2.1.2 Componentes de bajas, consulta y actualización

Algunas veces es necesario que el objeto que se esta manipulando en un sistema sea visible al usuario [Preece et al. 1990], en nuestro caso por ejemplo, al estar trabajando con páginas de un libro (imágenes), era necesario visualizarlas en la pantalla por diversas razones, entre las cuales tenemos: la comprobación de que la imagen que se esta dando debaja, consultando o actualizando sea la correcta (Figura 4.5); al incluir pies de lámina o descripciones de ilustraciones de una página era indispensable comprobar que la información del archivo generado por el usuario y la correspondiente al pie de lámina de una figura fueran idénticas.

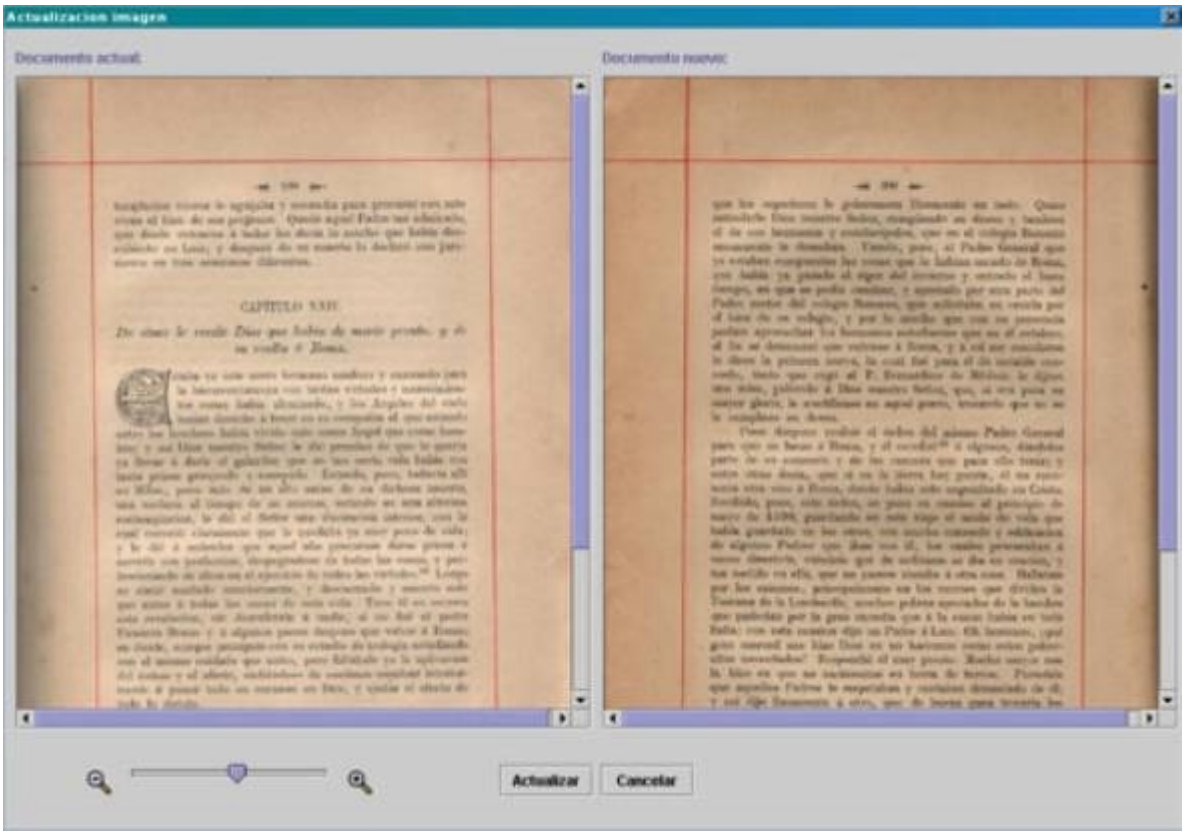

Figura 4.5 Interfaz de actualización de información

Finalmente, la implementación de cada componente del módulo de administración, involucró principalmente la utilización de SQL y el JUDBC para interactuar con la base dedatos y poder efectuar cada una las operaciones correspondientes a dichos componentes.

4.2.2 Módulo de consulta y navegación

Este módulo implicó en su implementación dos tipos de consulta de información : básica y de contenido.Para poder llevar a cabo la consulta en contenido fue necesario implementar controles que permitieran al usuario un seguimiento de la información a través de cada página de un libro, es decir, la navegación del mismo. Las operaciones en dicha navegación incluyen:

- página siguiente
- página anterior

- acceso rápido a una página específica.

Sin embargo, la consulta de la información no se limitó únicamente a la navegación de un libro, a su vez, se implementó un sistema de búsqueda para consultar su contenido, que finalmente facilitaría al usuario localizar yubicarde manera directadatos relevantes para el mismo.Para poder recuperar información a este nivel y tomando en cuenta que se utilizarían los algoritmospropuestos en el capítulo anterior, fue necesario codificar lasolicitud del usuario basándose en los mismos,para posteriormente buscar el código generado enlos textos correspondientes a cada uno de ellos, existentes en la base de datos.

Al igual que el módulo de administración, la interfaz del módulo de consulta está constituida por diversos componentes de una interfaz gráfica, siendo aún más importante la visualización de ciertos objetos en la pantalla, ya que en este módulo el usuario busca, consulta y navega la información (páginas)para su análisis.

Se respetaron las mismas consideraciones echas en el módulo de administración en cuanto al lenguaje utilizado, empleo de diálogosde tipo informativo y de validación, además de incluir iconos en botones que indicaban o reforzaban el entendimiento de los procesos y acciones que el usuario podía ejecutar.

4.3 Integración con UVA

Como se mencionó en el capítulo anterior, se pretende que el sistema CIText interactúe con otros componentes de U-DL-A como UVA y SIR. En este caso se logró la integración con UVA, interfaz que permite la consulta y navegación de colecciones organizadas jerárquicamente [Proal et al. 2000].

Dicha integraciónhace posible no solamente la consulta de material bibliográficoa nivel general sino de contenido a través de CIText. El proceso de integración consistió

únicamente en la inicialización por parte de UVA de un objeto del sistema CIText con los parámetros necesarios para la visualización y consulta en contenido.

La interacción de ambos componentes consiste básicamente en lo siguiente: el usuario busca un libro a través de UVA, ubica el nodo que contiene los datosde su interés y finalmente selecciona dicho nodo, acción que lo remitirá automáticamente al sistema CIText para consultar el libro digitalmente (Apéndice F).

4.4 Algoritmos

Para poder aplicar los algoritmos propuestos en el capítulo 2, fue necesario generar archivos que tuviesen la información de cada página de un libro previamente recuperada con un reconocedor óptico. Esto permitiría posteriormente codificarlos y procesarlos para la búsqueda de información.

Antes del proceso de codificación fue básico depurar cada archivo, es decir, eliminar caracteres que no representaban información útil, siendo estos todos aquellos que no correspondían a letras del alfabeto.

Un aspecto importante en esta parte del proceso fue el hecho de contemplar palabras truncadas al final de cada línea del texto que pudiesen perderse o distorsionarsepor continuar en la siguiente línea. Para ello, al momento de la depuración si el sistema detectabauna palabra al final de cada línea que tuviera un guión, dicha palabra era concatenada con el resto de la palabra ubicada en el siguiente renglón..

Finalmente, cada archivo se codificó de acuerdo a las características de cada algoritmo, y cuyo proceso consistió en lo siguiente:

- Recuperación de cada una de las cadenas del archivo de texto reconocido.
- Separación de cada una de las palabras contenida enlas cadenas recuperadas

- Codificación de cada palabra según tipo de algoritmo.
- Escritura de códigos a un archivo generado con la extensión correspondiente a cada algoritmo.

El proceso en general, se basóprincipalmente en el manejo de archivos, cadenas yoperaciones características de cada uno de ellos.

En el siguiente capítulo, podremos apreciar algunos aspectos y funcionalidades descritos en las sección 4.2 del sistema CIText.