



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
UNIDAD PROFESIONAL "ADOLFO LÓPEZ MATEOS"

"CREACIÓN DEL ARCHIVO MULTIMEDIA PARA EL LABORATORIO DE ACÚSTICA MUSICAL"

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMUNICACIONES Y ELECTRONICA

PRESENTA:
ALFARO AMAYA MIGUEL ÁNGEL

ASESORES:

DRA. ITZALA RABADÁN MALDA
ING. ROSA MERCADO MORENO
ING. JOAQUIN CONTRERAS ESTRADA



MÉXICO, D.F. 29 DE NOVIEMBRE DE 2011

INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
UNIDAD PROFESIONAL "ADOLFO LOPEZ MATEOS"

TEMA DE TESIS

**QUE PARA OBTENER EL TITULO DE
POR LA OPCION DE TITULACION
DEBERA(N) DESARROLLAR**

**INGENIERO EN COMUNICACIONES Y ELECTRONICA
TESIS Y EXAMEN ORAL INDIVIDUAL
C. MIGUEL ANGEL ALFARO AMAYA**

"CREACIÓN DEL ARCHIVO MULTIMEDIA PARA EL LABORATORIO DE ACÚSTICA MUSICAL"

DISEÑAR UNA APLICACIÓN PARA WEB LOCAL QUE PRESERVE LA INFORMACIÓN DE LA ACADEMIA DE ACÚSTICA DE LA CARRERA DE INGENIERÍA EN COMUNICACIONES Y ELECTRÓNICA EN LA ESCUELA SUPERIOR DE INGENIERÍA EN COMUNICACIONES Y ELECTRÓNICA DE LA UNIDAD ZACATENCO, REUNIDA A TRAVÉS DE LOS AÑOS EN LOS QUE SE RECAUDO INFORMACIÓN DE LAS TESIS DE EX ALUMNOS, LIBROS ANTIGUOS , IMÁGENES DE REVISTAS CIENTÍFICAS Y MEDIOS MULTIMEDIA DIRIGIDOS POR LOS PROFESORES Y ESTUDIANTES DE DICHA ESPECIALIDAD.

ANTECEDENTES
MARCO TEÓRICO
DISEÑO DEL ARCHIVO MULTIMEDIA

MÉXICO D.F. A 22 DE NOVIEMBRE DE 2012.

ASESORES


DRA. ITZALA RABADÁN MALDA


ING. ROSA MERCADO MORENO


ING. JOAQUÍN CONTRERAS ESTRADA


M. EN C. DAVID VÁZQUEZ ÁLVAREZ
JEFE DEL DEPARTAMENTO ACADÉMICO DE
INGENIERÍA EN COMUNICACIONES Y ELECTRÓNICA



Agradecimientos

Este trabajo es dedicado a mi familia, mis padres que ellos son mi inspiración para todo, me apoyaron y brindaron lo necesario para llegar a este momento.

A mi madre que insistió siempre en mi educación, nunca se rindió y hacia lo posible para que yo saliera adelante

A mi padre que siempre estuvo ahí en las buenas y en las malas, que siempre puso buena cara a mis decisiones y siempre me ha dado su apoyo incondicional en todo.

Mi s hermanos que han crecido conmigo y que eventualmente se han convertido en grandes amigos que me ayudaron a darme cuenta cuando estaba en un error y que me han aguantado por bastante tiempo

A mi hermana, que es una enorme persona que no se guarda nada y que siempre dice lo que piensa, me ha ayudado mucho a crecer como persona.

A mi hermano que con su ejemplo me ha ayudado a pensar mejor las cosas, a saber que nunca estaré lo suficientemente preparado para lo que la vida me ofrece y que es necesario pensar bien decisiones para no errar en la vida.

En general a todos mis maestros, los cuales con sus enseñanzas y experiencias de vida me dieron la inspiración suficiente para continuar estudiando y tomarle cariño a la carrera que con tanto esfuerzo termino.

Quiero agradecer a la maestra Itzalá Rabandan Malda la cual es una profesora ejemplar sin su ayuda no abría podido entusiasarme con el trabajo y creer en que las cosas que hacemos en este nivel valen mucho la pena.

A la maestra Rosa Mercado Moreno y a su Esposo por su ayuda y asesoramiento, fue inexistente este proyecto sin usted de verdad muchas gracias.

A mis maestros asesores y a todo aquel que en verdad por falta de espacio colaboró a la realización de este proyecto.

Además, muchas gracias a ti por leerlo.

Índice

Objetivo

Justificación

Introducción

Capitulo 1 Antecedentes

Capitulo 2 Marco teórico

Digitalización

Base de datos

HTML

MyEclipse

SQL

MySQL

JAVA

Apache Tomcat

Servidor

Capitulo 3 Diseño del archivo multimedia

Diseño de la presentación del archivo multimedia

Estructura del archivo multimedia

Instructivo para el administrador (Simulación del servidor local)

Instructivo para el usuario.

Presupuesto Estimado

Recomendaciones a futuro

Administradores de contenidos

AJAX

Anexos

A Comandos HTML

B Comandos SQL

C Servidores

D Normalización de bases de datos

Glosario

Bibliografía y referencias

Objetivo

Diseñar una aplicación para web local que preserve la información de la academia de acústica de la carrera de ingeniería en comunicaciones y electrónica en la escuela superior de ingeniería mecánica y eléctrica unidad Zacatenco reunida a través de los años en los que se ha recaudado información vía tesis de ex alumnos, libros de texto, imágenes de revistas científicas y medios multimedia digitalizados por los profesores y estudiantes de la especialidad.

Justificación

La creación de esta aplicación web, surgió por la necesidad de tener un recurso didáctico en la enseñanza de los diversos campos de la acústica. También para tener un catálogo musical y socio-cultural diverso. Debido a que no se cuenta con un mecanismo de almacenaje para tales catálogos dentro de la academia ni tampoco por la institución se ha ido perdiendo a través de los años.

También es necesario realizar esta aplicación porque en un futuro no muy lejano será muy difícil acceder a determinada información por medios analógicos como libros o videos de cinta. Con este archivo electrónico se estará en la vanguardia de las redes de información y se mantendrá agrupada la información generada o reunida por la academia de acústica de la ESIME Zacatenco.

Introducción

Existen pequeñas confusiones entre los archivos web y los archivos multimedia. Un archivo web consta de todo tipo de archivos que son compatibles con la web y visible desde un navegado web. Estos archivos pueden tener una extensión de archivo web gráficos como: .gif, .jpg, .bmp, .png, o una extensión de documentos como: .html, .txt, etc. Esto nos lleva a deducir que el resto de los archivos que no son compatibles con la web, serian tomados como archivos de descargas y no para visualizar contenido desde un navegador; pero esto no se cumple siempre, hay casos que cuando se instalan ciertas aplicaciones se hacen posible la visualización desde el navegador, como los conocidos archivos PDF si tenemos instalado un programa de visualización de archivos PDF, de no ser así siempre encontraríamos una opción de descarga de dicho archivo.

El término Multimedia se utiliza para referirse a cualquier objeto o sistema que utiliza múltiples medios de expresión (físicos o digitales) para presentar o comunicar información. De allí la expresión "multi-medios". Los medios pueden ser variados, desde texto e imágenes, hasta animación, sonido o video. Las presentaciones multimedia pueden verse en un escenario, proyectarse, transmitirse, o reproducirse localmente en un dispositivo por medio de un reproductor multimedia. En este trabajo se plantea la creación de un archivo multimedia pero no de reproductores multimedia, se usara para almacenar archivos de audio videos e imágenes de cualquier formato con un límite de su tamaño en el disco duro que sean pertenecientes a la academia de acústica de la Carrera de ICE en ESIME Zacatenco, se plantea en específico para el laboratorio de musical pero no se cierra en ello, se pueden almacenar graficas, hojas de datos, o manuales completos que sean de utilidad en cualquier otro laboratorio de la academia de acústica. Se crea un archivo multimedia de fácil acceso y utilización para los maestros de la academia además de poder almacenar archivos de texto imagen, audio y de video, utilizando lenguajes de programación como HTML y SQL. Entre todos ellos se ha elegido utilizar los lenguajes HTML, SQL y JAVA, junto con su respectivo software: My Eclipse Blue Edition, MySQL y Apache Tomcat, siendo todos ellos "open source". Se expresan las condiciones para que se pueda crear un administrador de contenidos que lleva al siguiente nivel a este proyecto, dejando una base sólida para los cambios futuros.

Para esto se considera el equipo de cómputo que se utilizara para conservar la aplicación web, el cual se encuentra en el laboratorio de acústica musical y cuentan con las siguientes características:

Maquina Hewlett-Packard Company

HP Compaq dc7900 convertible

Procesador Intel Celeron

2.94 GB de RAM

Extensión de dirección Física

EL sistema operativo es el Microsoft Windows XP Profesional Versión 2002 Service Pack2

Capítulo 1 Antecedentes

En la actualidad existen miles de documentos y archivos acústicos que por su importancia informativa o histórica se tiene la necesidad de preservarse para una mayor conservación y almacenamiento. Dentro de la academia de Acústica de la ESIME Zacatenco se ha intentado buscar un lugar donde libros, imágenes y sonidos valiosos tengan un lugar para su consulta y valoración de parte del alumnado y de los profesores de la especialidad perdiéndose o destruyéndose muchos en el intento. Este trabajo está pensado para que en un futuro la mayoría de los archivos sean digitalizados y puestos a la vista de toda la comunidad estudiantil dentro de una base de datos, además, que no se limite solamente a las personas pertenecientes a la especialidad si no también a la comunidad politécnica ideándose así una aplicación web para archivos multimedia.

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

Una base de datos es un almacén que permite guardar grandes cantidades de información de forma organizada para que luego se pueda encontrar y utilizar fácilmente. Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Con la evolución de los formatos digitales y diferentes tipos de archivos, surgieron nuevos tipos de bases de datos, denominadas multimedia.

El nacimiento de la multimedia ha renovado por completo la idea de base de datos. Con la llegada de la Multimedia el abanico de posibilidades complementa la representación de los objetos que se desean almacenar. El ambiente de un sistema de gestión de bases de datos multimedia (MM-DBMS¹) integra texto, datos, video, imagen (estática y dinámica) y sonido.

Para poder crear una base de datos multimedia, se necesitan datos digitalizados, para esto, debemos de llevar a cabo el proceso de digitalización. La digitalización de documentos es el mecanismo más sencillo, eficiente y rentable para almacenar, administrar y consultar grandes volúmenes de documentos, en forma de imágenes digitales, o archivos sonoros en formatos digitales.

A través de la digitalización de archivos, se han podido preservar obras y documentos de importancia histórica, como tratados o pergaminos para que futuras generaciones puedan consultarlos sin maltratar los originales de esta misma manera podemos preservar audio y video.[1]

¹ Es un sistema de gestión de base de datos multimedia

Capítulo 2 Marco Teórico

2.1 Digitalización

Digitalizar es traducir información tal como pueden ser textos, imágenes, sonidos o videos en un formato que pueda ser entendido por un microprocesador. Cada uno de estos tipos de archivos, tienen un proceso diferente para llevar a cabo su transformación a un formato digital.

Un proceso de digitalización consta de los mismos procedimientos para lograr convertir una señal analógica en una digital, estos procedimientos son los siguientes:

- Muestreo
- Cuantización
- Codificación

Aunque la manera de realizar cada uno de estos procedimientos, para diferentes señales analógicas puede no ser el mismo, sus principios no cambian. Es decir, el concepto es el mismo, sin embargo la manera en que se muestrea, cuantifica y codifica una señal de audio, imagen o video, es diferente.

A grandes rasgos, el muestreo de una señal analógica es literalmente la toma de muestras o valores de esta señal cada cierto tiempo o en diferentes locaciones, dependiendo si es un archivo de audio, video o imagen.

La cuantización se refiere a la asignación de valores discretos a cada una de estas muestras. Y en la codificación a cada muestra obtenida se le asigna un código en sistema binario, es aquí donde se realiza la conversión a señal digital.

Estos procedimientos se llevan a cabo mediante la ayuda de instrumentos que se han creado para facilitar la digitalización de archivos.

2.2 Base de datos

Como se menciona en los antecedentes una base de datos es un almacén que permite guardar grandes cantidades de información teniendo un fácil y rápido acceso a ellos, para hacer esto posible existen muchos métodos.

Entre todos ellos se ha elegido utilizar los lenguajes HTML, SQL y JAVA, junto con su respectivo software: My Eclipse Blue Edition, MySQL y Apache Tomcat.

a)2.2.1 Lenguaje HTML.

HTML es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Etiquetas de Hipertexto. Es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto. Este lenguaje de programación utilizado para la

creación de páginas de internet. Fue creado como proyecto por el científico Tim Berners-Lee allá por 1990, su principal objetivo fue el de facilitar el acceso a la información de distintas universidades. Nunca se imaginó que su proyecto le daría paso a crear la web como tal. En un principio solo contaba con texto y algunas imágenes solo como medio divulgativo.

Para poder crear una página HTML se requiere un simple editor de texto y un navegador de internet. El lenguaje está constituido de elementos que el navegador interpreta y los despliega en la pantalla de acuerdo a su objetivo. Estos elementos son llamados etiquetas (tags), que tienen esta forma o <P>. Cada etiqueta tiene su propio significado, por ejemplo significa que se escriba en negrita (bold) o <P> significa un párrafo, <A> es un enlace, etc. Casi todas las etiquetas tienen su correspondiente etiqueta de cierre, que indica que a partir de ese punto no debe de afectar la etiqueta. Por ejemplo se utiliza para indicar que se deje de escribir en negrita. Así que el HTML no es más que una serie de etiquetas que se utilizan para definir el contenido del documento y algún estilo básico: Esto está en negrita. Para futuras referencias indicamos la utilización y significado de las etiquetas más importantes en el anexo A. [2]

b)2.2.3 Lenguaje SQL

SQL, Structure Query Language (Lenguaje de Consulta Estructurado) es un lenguaje de programación para trabajar con base de datos relacionales como MySQL, Oracle. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella.

Los laboratorios de IBM definen el lenguaje SEQUEL (Structured English Query Language) que más tarde sería ampliamente implementado por el sistema de gestión de bases de datos (SGBD) experimental System R, desarrollado en 1977 también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en 1979 en un programa comercial.

El SEQUEL terminaría siendo el predecesor de SQL, siendo éste una versión evolucionada del primero. El SQL pasa a ser el lenguaje por excelencia de los diversos sistemas de gestión de bases de datos relacionales surgidos en los años siguientes y es por fin estandarizado en 1986 por el ANSI, dando lugar a la primera versión estándar de este lenguaje, el "SQL-86" o "SQL1". Al año siguiente este estándar es también adoptado por la ISO. [4]

En este caso se utilizara el gestor MySQL que es un interpretador de SQL, es un servidor de base de datos.

c)2.2.5 Language JAVA

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

El diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática, como por ejemplo en dispositivos móviles y *sistemas empotrados*.

- En el navegador web.- Desde la primera versión de java existe la posibilidad de desarrollar pequeñas aplicaciones (Applets) en Java que luego pueden ser incrustadas en una página HTML para que sean descargadas y ejecutadas por el navegador web.
- En sistemas de servidor.- Es sencillo implementar aplicaciones para interactuar con servidores en una red, estas servirían como un contenedor y java haría el papel de interfaz...
- En aplicaciones de escritorio.- Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los PC de usuario de los sistemas operativos más usados en el mundo. Además, muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación de modo que se ejecuten en cualquier PC.
- Plataformas soportadas.- Microsoft y Mac pueden soportar en sus sistemas operativos entornos de ejecución java, sin embargo, solo Mac puede hacerlo por defecto. [16]

2.2.2 Programa MyEclipse

MyEclipse es un producto comercial Java EE y Ajax IDE creado y mantenido por la empresa Genuitec, miembro fundador de la Fundación Eclipse.

MyEclipse tiene dos versiones principales (aparte del "Blue Edition" y "MyEclipse Edición Primavera", en adelante), una profesional y una edición estándar. La edición estándar incluye herramientas de base de datos, un diseñador de páginas web visuales, herramientas de persistencia y un número de otras características para el perfil de desarrollo básico de Eclipse Java. Compite con el proyecto Web Tools, que es una parte de Eclipse en sí.

MyEclipse es un proyecto separado por completo, y ofrece un conjunto de características diferentes.

Con este programa se levanta el servicio de la base de datos, usando códigos java manda llamar al Tomcat para que cree el servidor local, este a su vez jala los códigos HTML que son la parte visual del proyecto.[3]

2.2.4 Programa MySQL

Permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones, etc., resumiendo: administrar bases de datos.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

En el desarrollo del archivo multimedia se implementan varios comandos explicando su uso, el qué significa y porque se usa, en el anexo B se detallan varios comandos muy usados en el lenguaje SQL. [5]

2.2.6 Programa Apache Tomcat

Es un servidor de internet (especializado en aplicaciones Java). Es un programa que hace a una computadora capaz de almacenar los sitios Web que creas (en simple HTML o en Java) y enviarlos a las personas que se conecten a él como una página que puedan ver en sus navegadores. Es bastante popular por ser gratuito y soportar Java entre los estudiantes y empresas que recién comienzan.

Es relevante en este proyecto por su función de simular un servidor local en la computadora sin necesidad de conectar todo el proyecto a la red de internet. [7]

2.2.7 Servidor

Un servidor, para informática, es una computadora que forma parte de una red de computadoras y que a la vez proporciona ciertos servicios a otras computadoras que forman parte de la misma red llamadas clientes. Existen distintos tipos de servidores, muchos de ellos no son relevantes para este proyecto es por eso que se especifican en el Anexo C.

El servidor que compete es el que dice su definición informática, se necesita que una maquina brinde el servicio de guardar los datos y que permita verlos, para eso se utiliza el programa Tomcat, hace que el local host (computadora local) se convierta en un servidor local capaz de brindar estos servicios. [8]

Capítulo III Diseño de la base de datos

3.1 Diseño de la presentación de la Base de datos

La primera parte fue hecha a partir del lenguaje de programación HTML, fue elegido para ser la coraza del archivo multimedia porque a futuro se planea enviar este archivo a la red y se pueda consultar a través de internet.

Se comienza diseñando la parte que los usuarios verán al adentrarse en el archivo multimedia, en primer plano se muestra la división de 2 ventanas, cada ventana representa un programa HTML diferente. La primera ventana de izquierda a derecha despliega el menú índice, el cual muestra los links o ligas correspondientes a la portada y el vínculo para agregar o revisar los archivos en la base de datos. Para la construcción del programa se utilizan etiquetas las cuales en su mayoría indican el inicio o final de una acción específica. Explicando comando a comando, el programa principia con la etiqueta <HTML>.

Nombre del comando	Sintaxis	Descripción
HTML	<HTML>"Contenido" </HTML>	Indica el inicio y final del código HTML.

Luego la etiqueta <head>

Nombre del comando	Sintaxis	Descripción
Head o cabecera	<head> "contenido" </head>	Representa la cabecera, donde se puede ingresar datos de información sobre el documento, todo opcional.

En este caso se agrega un título de trabajo como información.

Después el comando <title>

Nombre del comando	Sintaxis	Descripción
Title o título	<title> "contenido" </title>	Introduce un título a la página, aparecerá como nombre en el navegador de internet, se cierra esta acción con la etiqueta </title>

Hecho esto se procede con el cuerpo usando la etiqueta <body>

Nombre del comando	Sintaxis	Descripción
Body o cuerpo	<body> "contenido" </body>	Crea un cuerpo a la página aquí se introduce todo lo que se quiere mostrar.

En este caso se muestran 2 ventanas una para el índice o menú y otra que muestre la página actual que se seleccione. Para esto se utiliza la etiqueta `<frameset rows=* cols= "251,776">`

Nombre del comando	Sintaxis	Descripción
Frameset, localizador de ventana	<code><frameset rows=* cols="251,776"></code>	Permite dividir la pantalla de la pagina en columnas y filas, (rows y columns en ingles), se indica en rows un asterisco (*) para obtener en las fila todo el espacio de altura de la ventana disponible en pixeles, y en cols se colocan 2 números separados por una coma y encerrados en comillas, así se indica la anchura de cada ventana en pixeles.

Luego de eso se asigna un nombre a cada una de las ventanas para poder referenciarlas con la pagina individual que van a mostrar. Con la etiqueta: `<frame name="menu" src="menuprincipal.html" title="menu"/>`,

Nombre del comando	Sintaxis	Descripción
Frame name and title o nombre de ventana y tirulo	<code><frame name="menu" src="menuprincipal.html" title="menu"/></code> ,	Se asigna nombre y se asigna la pagina a mostrar indicando su dirección "url" o dirección de ubicación en la maquina.

Hecho eso se coloca la etiqueta de terminación para división de ventana `</frameset>`.

Nombre del comando	Sintaxis	Descripción
Terminación de división de ventana	<code></frameset> <noframes></code>	Por seguridad Se agrega la etiqueta <code><noframes></code> , por si el navegador no puede gestionar tantas ventanas utilizaría una página alternativa.

Cada vez que se seleccione el vínculo de archivos se elige si se quiere agregar datos al archivo o si se quiere revisarlos,

Para esto se usan vínculos, uno que lleve a los formularios correspondientes para la captura de información y otro que muestre a través de un "query" o petición, los datos almacenados.

Los formularios para HTML permiten ver como los usuarios pueden pedir u ofrecer información. La etiqueta que define el inicio de un formulario es “<form>” y el final del mismo “</form>”.

Nombre del comando	Sintaxis	Descripción
Inicio y final de un formulario	<form>”Contenido“ </form>	Define el inicio y fin de un formulario, puede contener los atributos ACTION, METHOD y ENCTYPE.

A continuación se listan los atributos que puede contener el formulario dentro de su declaración:

- ACTION. Atributo que determina la acción que debe realizarse al pulsar el botón de ejecución
- METHOD: Indica el método de transferencia de información introducida en el formulario. Se pueden emplear los métodos GET y POST.
- ENCTYPE: Atributo que determina la forma de codificación empleada para el transporte del contenido del formulario.

Así entonces, después de definir los atributos del formulario se procede a definir todos los elementos que va a contener, botones, cajas de texto, casillas de verificación. Para este caso solo se usan cajas de texto y botones. Para poder ingresar el nombre del archivo y su descripción solo se necesitan cajas de texto, es declarada por defecto por el atributo “TYPE= “text” ” así que es posible prescindir de él.

Nombre del comando	Sintaxis	Descripción
Caja de texto	<input name="archivo" type="text" >	Se usa para crear una caja de texto donde se pueda escribir el nombre del archivo.

Además de eso se usa otro botón para enviar la información a la base de datos. Para enviar la información se usa otro botón este se especifica en el atributo “value” dentro del comando “input”. Para restaurar o borrar el contenido de las casillas se usa otro botón llamado “reset” también se especifica en el atributo valor (“value”) del comando” input” dentro del formulario.

Nombre del comando	Sintaxis	Descripción
Declaración de un botón	<code><p align="center"><input name="boton" type="reset" id="boton" value="Limpiar"></p></code>	“<p” indica que es el texto del botón, “aling” indica la posición donde se alineara el texto del botón, luego se indica el nombre del botón “name="boton" “ y después su tipo que es “submit” en este caso, el identificador del botón que vuelve a ser “botón”, después el valor que mostrara al usuario “value="Limpiar” ” y por último el comando de cierre “</p>”.

Ahora para poder subir algún archivo se utiliza dentro del comando input el botón file, que se encargara de mostrar una ventana de selección de archivo que se quiera guardar dentro de la base de datos.

“<input name="archivo" type="file" id="archivo">

`<p align="center"><input name="boton" type="submit" id="boton" value="Enviar"></p>`

Nombre del comando	Sintaxis	Descripción
Introducción de un archivo	<code><input name="archivo" type="file" id="archivo"></code>	Esta instrucción crea una casilla en la cual se pone la dirección url o dirección de destino, dentro de la computadora, del archivo que se quiera introducir en el archivo multimedia.

Nombre del comando	Sintaxis	Descripción
Enviar a la computadora el archivo seleccionado	<code><p align="center"><input name="boton" type="submit" id="boton" value="Enviar"></p></code>	Indica un botón para subir la información ya cargada en la casilla. Este a su vez interactúa con los comandos SQL y de JAVA que envían a su sitio los archivos seleccionados.

Así quedaría la página al abrirse en un explorador de internet. (Figura 3.1)



Figura 3.1 Visualización de la caratula para el archivo multimedia.

3.2 Estructura de la Base de datos

Anteriormente se especificó que se utilizaría el programa My SQL para crear la estructura, es sencillo de instalar, el único inconveniente es especificar una contraseña de root o de administrador. La cuenta “root” es usada para instalar paquetes, actualizar RPMs y realizar la mayoría de las tareas de mantenimiento del sistema. Conectándose como root le da control completo sobre el sistema. Es por eso importante recordar esta cuenta, pasando esto lo demás es solo dar clic en siguiente.

Lo primero sería crear la estructura del archivo multimedia, para esto se utiliza el lenguaje SQL, este lenguaje utiliza el concepto de campos y registros muy similar a columnas y filas del programa Excel de Windows. En este caso las columnas sería similar a los campos y las filas serian similares a los registros. Cada registro que se agregue a la tabla será en realidad una fila de datos relacionados entre sí. Cada campo de la base de datos será equivalente al tipo de dato que se guardara en el, por ejemplo si se pretende guardar un nombre con apellidos se propone que exista un campo para nombre un campo para apellido materno y otro para apellido paterno, sin embargo es posible poner un campo de nombre completo. En estos casos el registro seria Miguel Ángel en el campo nombre Alfaro en apellido paterno y Amaya en apellido materno. Para agregar un campo es necesario especificar el tipo de dato que contendrá a la hora de declarar la creación de la tabla, en programación existen varios tipos de datos, para este caso se utilizan los char para caracteres, los int para números y el long blob para convertir a bytes el dato deseado.

Para hacer las pruebas se crean solo 3 campos:

- ⦿ id autoincrement, se genera automáticamente un numero por cada registro ingresado
- ⦿ Nombre del archivo con su extensión (tipo de archivo)
- ⦿ Archivo deseado, en bytes tipo blob.

Como prueba solo se ponen estos campos es más fácil agregar un tipo de archivo que agregar una tabla a cada archivo nuevo por lo tanto se crea un campo adicional para identificar si el registro es de video, de audio o de imagen. Para hacer la conexión entre la estructura creada en MySQL, la pagina WEB y el servidor local, se usan comandos con el lenguaje JAVA, Si recordamos, JAVA es un lenguaje orientado a objetos, esto lo hace trabajar en base a clases para hacer sus aplicaciones. Haciendo la analogía con C++ se puede confirmar que en ambas existen clases que están por defecto, eso quiere decir que hay pequeños programas que ya hacen alguna acción en especifico solo con mandarlas llamar poniendo su nombre en la cabecera del nuevo programa que se haría en base a ellas.

Estas clases en C++ son llamadas librerías con extensión “.h”, en Java existe algo muy similar solo que estas “librerías” se identifican con la extensión “.jar”.

Para la programación del archivo multimedia, la librería conector.jar ya está por defecto, almacena un conector el cual hace interactuar a la página web con el servidor local y a su vez con el código creado en MySQL.

las vistas, si no está, seleccionamos otros (other) y abrimos el modo “debug” para identificarlo tiene un icono en forma de un bicho.

Seleccionada la vista se puede observar que están visibles todos los servidores disponibles, todos están en un estado detenido, se debe verificar que esté el servidor tomcat 6.x con un símbolo de + en su costado izquierdo, al seleccionar ese símbolo se convierte en un símbolo de menos y se visualiza la carpeta con el nombre del proyecto que se creó, en el apartado de ubicación (location) está la dirección url de la aplicación que se creó para simular un servidor dentro de la misma computadora (Fig. 3.2).

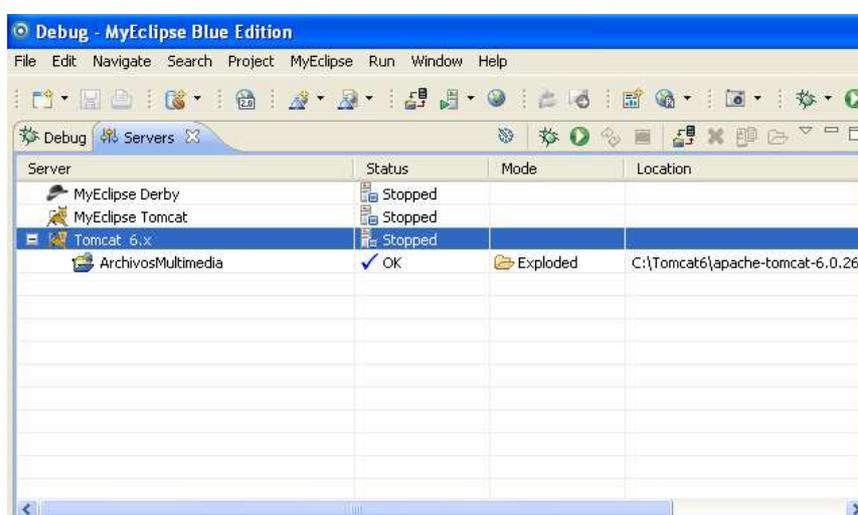


Figura 3.2 Vista del servidor Tomcat 6.X en la ventana de My Eclipse

Verificada la existencia de la aplicación Tomcat dentro del servidor, se procede a correr el programa, seleccionando el botón “debug server” el cual tiene un icono verde con una flecha blanca en medio, eclipse comienza a avanzar paso por paso por la aplicación creada, en la parte inferior donde dice consola (console) se observan las acciones que toma eclipse para generar el *servidor local*, es en esta parte donde nos indicaría si existe o no algún error (Figura 3.3).



```
tomcat6Server [Remote Java Application] C:\Archivos de programa\Java\jdk1.5.0_15\bin\javaw.exe (10/11/2011 12:35:01)
INFO: sessionListener: contextinitialized()
10/11/2011 12:35:17 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Despliegue del directorio ROOT de la aplicación web
10/11/2011 12:35:17 PM org.apache.coyote.http11.Http11AprProtocol start
INFO: Arrancando Coyote HTTP/1.1 en puerto http-8080
10/11/2011 12:35:18 PM org.apache.coyote.ajp.AjpAprProtocol start
INFO: Arrancando Coyote AJP/1.3 en ajp-8009
10/11/2011 12:35:18 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 7836 ms
```

Figura 3.3 Consola de acciones, muestra el procedimiento para simular el servidor.

La consola indica cuando el servidor ya comienza a operar, como se observa en la Figura 3.3, la consola muestra como información que el servidor ha sido inicializado en 7836 milisegundos, así entonces se verifica que el servidor fue subido sin ningún contratiempo y ya está listo para recibir la información por medio de la pagina HTML creada para la base de datos.

Para acceder a la pagina usando el servidor local se abre cualquier explorador de internet que esté al alcance, en la parte que es para escribir la dirección web a la que queremos acceder se escribe la siguiente dirección: “http://localhost:8080/ArchivosMultimedia/”, así, automáticamente abre la pagina antes creada. (Figura 3.4)



Figura 3.4 Página cargada con la dirección: http://localhost:8080/ArchivosMultimedia./

Así pues se procede a subir algún archivo, en el apartado de instrumentos se aprecian 2 vínculos, uno para agregar instrumento y otro para consultar instrumento.

Para agregar un instrumento se selecciona ese vínculo, se puede observar que están los recuadros creados en la sección del diseño de la pagina web, ahora se selecciona el botón “Examinar” el cual abrirá una ventana para seleccionar el archivo deseado, después se oprime el botón subir para que sea cargado el archivo a la base de datos. Aparecerá un mensaje de confirmación “Archivo XXX:XXX cargado “. (Figura 3.5)

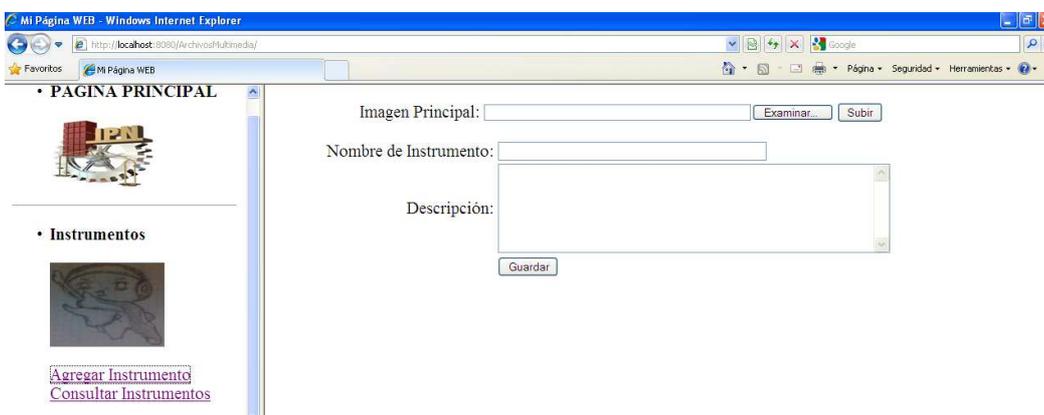


Figura 3.5 a)

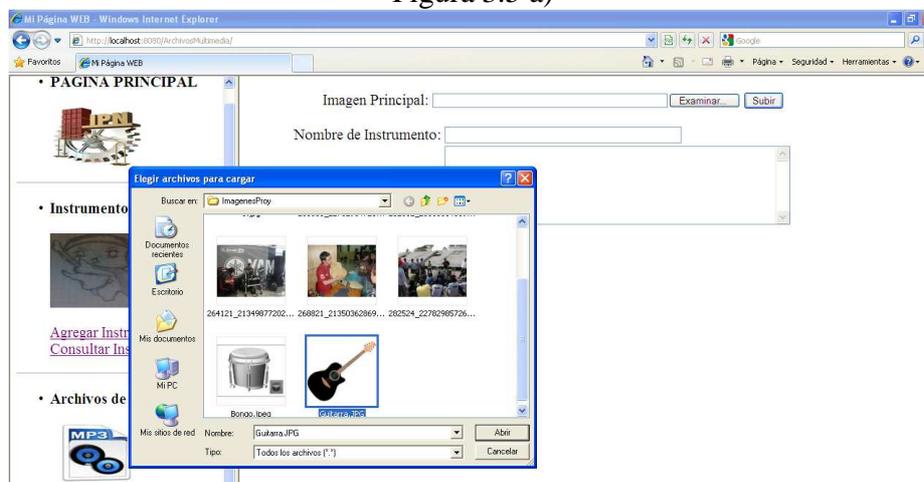


Figura 3.5 b)

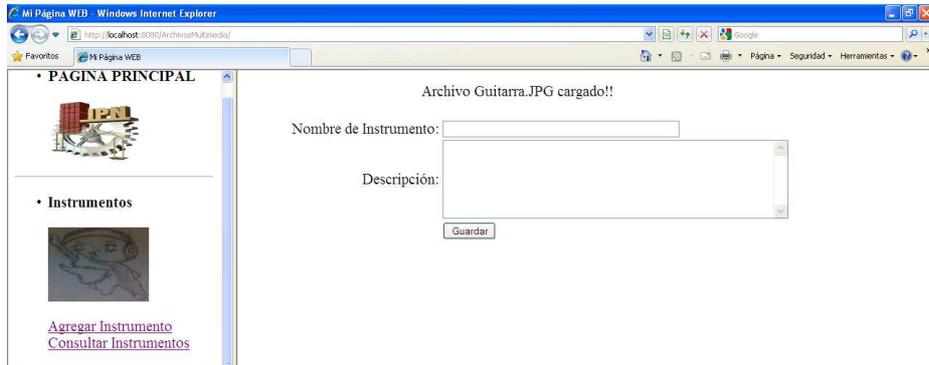


Figura 3.5 c)

Figura 3.5 a) Vinculo “Agregar un archivo”, b) Oprimir botón examinar
c) Confirmación de archivo cargado

Luego se agrega un nombre al archivo subido y una descripción relacionada con este último, luego se selecciona el botón guardar, si todo salió bien aparece otro mensaje de confirmación el cual menciona que el instrumento se ha guardado correctamente. Es importante oprimir el botón subir antes de agregar nombre y descripción al archivo, si no, este no será guardado en el archivo multimedia y tendrá que repetirse el procedimiento. (Figura 3.6)

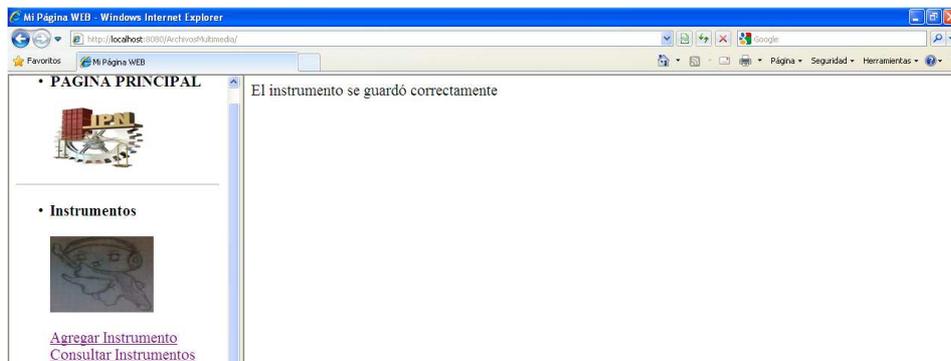


Figura 3.6 El instrumento se guardó correctamente.

Para poder consultar los datos en la base se oprime el vínculo de consultar instrumentos, este lleva a la vista donde se encuentran todos los archivos subidos hasta el momento con su respectivo nombre y descripción. Fig. (3.7)

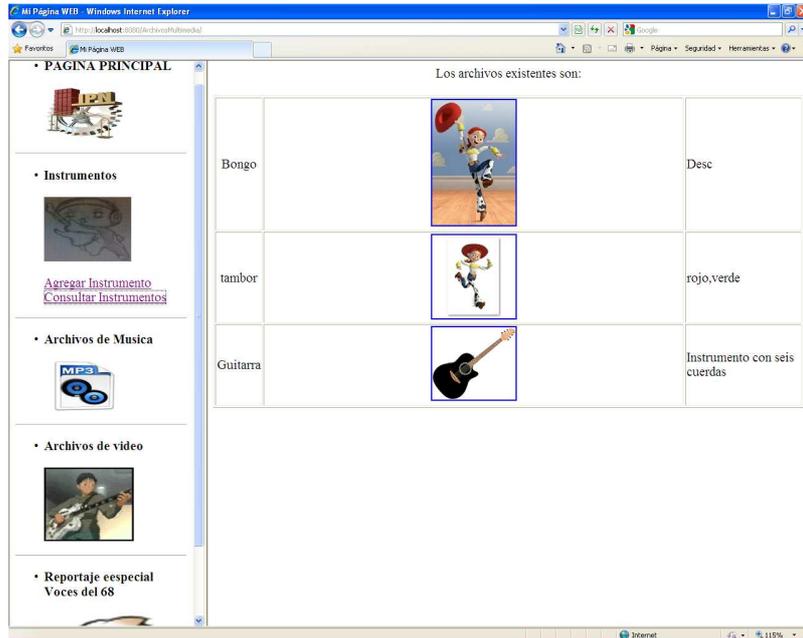


Figura 3.7 Consulta a los instrumentos.

Finalmente para dar de baja al servidor local se vuelve a la ventana de MyEclipse y se oprime el botón de alto a la aplicación (stop) que tiene un icono en forma de recuadro rojo, el cual para cualquier acción que el servidor pudiese estar haciendo y lo pone en estado detenido.(figura 3.7)

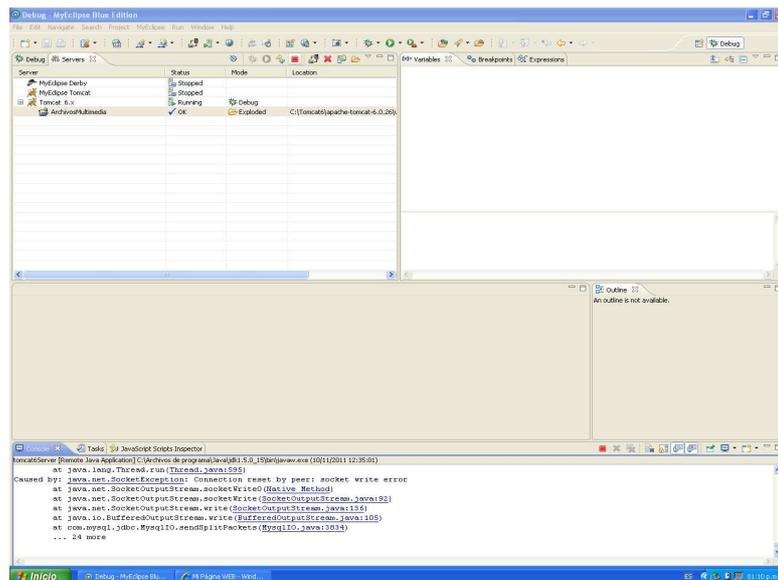


Figura 3.7 Regreso a estado bajo del servidor local.

Recomendaciones para el administrador

- ⦿ 1.- Cerrar la ventana de servicios después de activado el servicio MySQL
- ⦿ 2.- No cambie la contraseña de ROOT, en dado caso de hacerlo avisarle al encargado del laboratorio de musical acerca de este cambio.
- ⦿ 3.- Revisar la veracidad de la información subida al Archivo Multimedia.
- ⦿ 4.- Hacer lo posible por mejorar el desempeño del Archivo Multimedia haciendo y recibiendo aclaraciones y sugerencias de los usuarios de este trabajo.

3.4 Instructivo para el usuario

- ⦿ 1.- Asegúrese de que exista corriente eléctrica.
- ⦿ 2.- Verifique que la computadora este bien conectada
- ⦿ 3.- Encienda la computadora
- ⦿ 4.- Abra cualquier explorador de internet que este instalado en la computadora (Internet Explorer, Firefox, Google Chrome Figura 3,8).



Figura 3.8 Ejemplos de exploradores de internet

- ⦿ 5.- Escriba en la barra de direcciones la siguiente dirección de internet (Figura 3.9)
`http://localhost:8080/ArchivosMultimedia/`



Figura 3.9 Pagina para acceder al servidor local

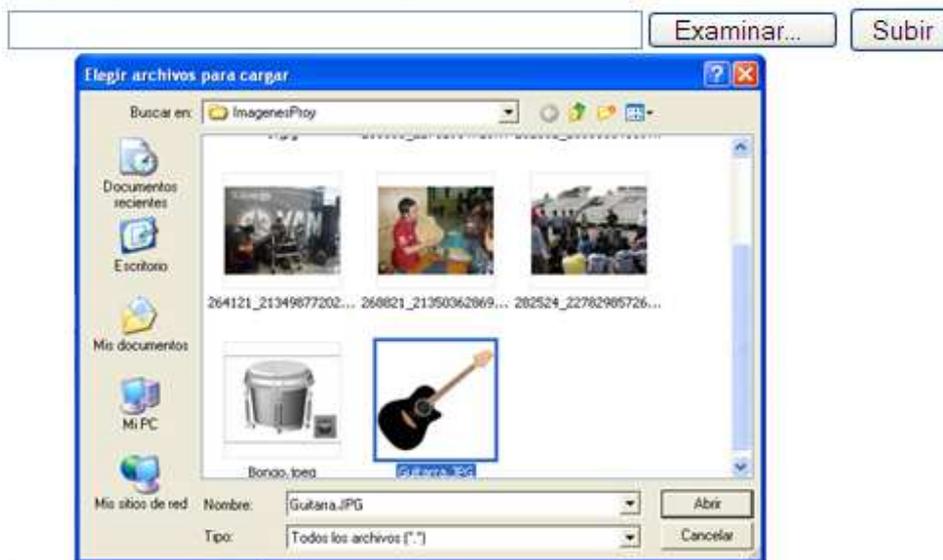
- ⦿ 6.- Se abrirá la página de internet creada para el archivo multimedia, proceda a elegir alguna opción “Agregar” o “Consultar” archivos (Figura 3.10)



[Agregar Instrumento](#)
[Consultar Instrumentos](#)

Figura 3.10 agregar o consultar archivos

- 7.- Si se oprime la opción cargar un archivo se especifica en primera instancia el archivo a cargar, se oprime el botón examinar, se selecciona el archivo, seleccione abrir y luego seleccione subir. (Figura 3.11)



Archivo Guitarra.JPG cargado!!

Figura 3.11 Cargar el archivo

- 8.- Agregue un nombre al archivo, no debe de sobrepasar los 45 caracteres, junto con una descripción que no debe de sobrepasar los 800 caracteres. (Figura 3.12)

Archivo Guitarra.JPG cargado!!

Nombre de Instrumento:

Descripción:

Figura 3.12 Agregar nombre y descripción al instrumento.

- 9.- Después de llenar ambos campos seleccione el botón guardar, espere hasta ver el mensaje de confirmación (Figura 3.13)

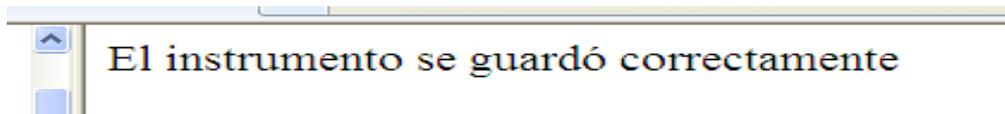


Figura 3.13 Mensaje que confirma que el archivo se cargo correctamente.

- 10.- Si se selecciona el *vínculo* de consultar se verán todos los archivos subidos al archivo multimedia hasta el momento, se ordenan conforme se agregan los archivos. (Figura 3.14)



Figura 3.14 Vista de los archivos subidos al archivo multimedia hasta el momento.

Recomendaciones para los usuarios

- 1.- Se recomienda traer sus archivos ya digitales en un disco compacto o una memoria tipo USB sin virus.
- 2.- El Archivo Multimedia recibe archivos en cualquier tipo de formato, sin embargo, para propósitos de la misma academia, se recomienda agregar archivos con el formato que más calidad les brinde, sin sobrepasar los 20 Mb por archivo.
- 3.- Al agregar un archivo, asegúrese de seguir las instrucciones, si intenta guardar todo el registro sin antes subir el elemento digital, el servidor marcará un error y tendrá que repetir todo el procedimiento.

Presupuesto estimado

Si se pretende ganar 15,000 pesos mensuales y se tarda 10 meses en diseñar y armar este proyecto es posible que se cobre 150,000 pesos, pero si se cuentan los gastos adicionales como tornamesa, escáner, capturadora de video y videocasetera el costo se elevaría aun mas.

Observaciones y conclusiones

Conclusiones:

- ⦿ Se construyó una aplicación web capaz de guardar hacia una base de datos cualquier tipo de archivo en cualquier tipo de formato. Además de ser una aplicación la cual se tiene acceso atreves de una red local y posteriormente desde internet.

Observaciones:

- ⦿ Es viable a actualizaciones futuras ya que su simple construcción lo hace viable a cualquier tipo de cambio visual o total
- ⦿ Termina siendo de fácil acceso y de un simple uso para cualquier persona perteneciente a la Academia de Acústica.
- ⦿ Es posible subirle a la red de internet con ayuda del servidor del poli gracias a que está realizado bajo un ambiente de página web.

Recomendaciones a futuro

El beneficio de la academia de acústica se necesita digitalizar el material análogo aun disponible y hacer un programa para el manejo del más antiguo para que su forma digital sea de la mejor calidad posible.

La aplicación actualmente guarda cualquier tipo de archivo (ima, pdf Excel, etc) es observable la dirección de almacenamiento aquí solamente las imágenes son accesibles mediante la aplicación.

Desarrollar la fase de implementación para que se tenga acceso a todos los tipos de datos antes mencionados ya que por el momento solo se accesa a imágenes de la base de datos. Creación de reproductores y visualizadores multimedia.

Dar seguimiento en esta fase con “open source” utilizar la misma tecnología o los mismos ambientes de desarrollo para su buen funcionamiento.

Actualizar y mantener la base de datos para un mejor desempeño del manejo de la información, gestionar los datos en cuanto a veracidad y optimización del espacio de memoria en la computadora servidor.

Crear las condiciones necesarias para crear una conexión en red local que permita utilizar la aplicación en los demás equipos de laboratorio de acústica musical de la academia de Acústica de la ESIME Zacatenco.

Crear un dominio dedicado, Gestionar para subir la aplicación web a Internet y que se utilice en cualquier momento y lugar.

Anexos

Anexo A Comandos del lenguaje HTML

COMANDO	DESCRIPCION	EJEMPLO
<code><html></code> <code></html></code>	Para indicarle al browser que es código HTML el que viene descrito a continuación, se limita todo el código por estos dos parámetros. Dentro de ellos se escribe el código de la página	
<code><title></code> <code></title></code>	Entre estos dos comandos va escrito el título de la página, este va en la parte superior de la ventana que despliega el browser.	
<code><body bgcolor=#</code> <code>text=# link=#</code> <code>vlink=# alink=#</code> <code>background=""></code> <code></body></code>	Entre estos parámetros se escribe todo el texto de la página, el cuerpo del documento (BODY). Con estos comandos se define el color general de:	bgcolor=# color del fondo de la página, este color puede ser cambiado por un fondo, (background) text=# es el color del texto general de la página link=# color del texto con link vlink=# color del texto de link visitado alink=# color del texto de link activo background="" en este comando se puede detallar un fondo específico para la página, un archivo de imagen. Los colores se definen en formato RGB, se puede ver ejemplos de los colores con sus códigos en al siguiente página (http://www2.ing.puc.cl/~iic10621/clases/HTML/colores/index.html). Estos colores consisten

		de seis dígitos, o letras desde la A hasta la F.
<code>
</code>	Este comando sirve para insertar un salto de línea dentro de la página	
<code><p></code>	Sirve para ingresar un salto de párrafo dentro del contenido	
<code><h!> </h!></code>	Especifica el tipo de título con un número del 1 al 6 en vez de !!, especificando la importancia del título.	
<code> </code>	Sirve para especificar las fuente del texto que va entre estos dos parámetros	face="Arial,Helvetica" Especifica el tipo de letra a utilizar. color=# Especifica el color del texto a utilizar, descrito en anterioridad Los colores se definen en formato RGB,. Estos colores consisten de seis dígitos, o letras desde la A hasta la F. size=%% Detalla el tamaño de la letra elegido, número entre el -6 y el +6, en vez de %%.
<code> </code>	El texto que va entre estos dos parámetros se encuentra en BOLD, o Negritas.	Ejemplo de Bold
<code><i> </i></code>	El texto que va entre estos dos comandos se encuentra en Itálicas.	<i>Ejemplo de Itálicas</i>
<code><u> </u></code>	El texto que va entre estos dos parámetros está subrayado.	<u>Ejemplo de subrayado</u>

<code></code>	Este comando inserta imágenes dentro de las páginas web, tiene que estar especificado el directorio en el caso que no se encuentre la imagen en el mismo directorio que la página.	
<code></code>	Detalla el alineamiento que tendría la imagen (derecho=right, izquierdo=left y centrado=center)	
<code> </code>	Sirve para realizar direccionamiento a nuevas páginas webs, LINKS. Puede ser un texto o una imagen. En vez de \$\$, se puede colocar la dirección a donde va a apuntar,	http://www.ing.puc.cl o puede ser colores.htm
<code>&ntilde</code>	Con este comando se escriben la ñ.	
<code>&nbsp;</code>	Se utiliza para ingresar espacios dentro del texto, para ingresar más de un espacio en la página	
<code> </code>	Necesarios para tabular alguna	

	sección de la página, desde textos, imágenes y tablas.	
<code> </code>	Ayuda a ingresar viñetas dentro del texto.	
<code> ...</code>	Este comando inserta viñetas con números, estos son contiguos.	
<code><center> </center></code>	Centra un objeto, texto, imagen, tabla, etc.	
<code><p align="left">izquierda </p></code>	Deja el texto a la izquierda de la página	
<code><p align="right">derecha </p></code>	Deja el texto a la derecha de la página	
<code><table> <tr colspan="N "> <td rowspan="N "> </table></code>	Comandos para definir una tabla,	<code><tr></tr></code> Se utiliza para definir una fila. <code><td> </td></code> Define una columna de la tabla

[8]

Anexo B Comandos MySQL

Miguel Ángel Sanz Santos

Comandos básicos MYSQL

Comandos básicos de MYSQL

Miguel Ángel Sanz Santos
Fac. CC. Geológicas – UCM
2002-2003

Comandos principales de MYSQL.•Arranque de los servidores.

Servidor Linux:

**safe_mysqld -user=mysqladm &
root &**

Servidor Windows:

Mysqld

Arranque como usuario:

mysql -u usuario -p clave

En el caso en el que el servidor no se encuentre en el ordenador que estamos trabajando la orden de uso será:

mysql -h nombre_host -u usuario -p clave•Consultas generales sobre las bases de datos

Mostrar las bases de datos:

show databases;

Mostrar las bases de datos seleccionadas:

select databases();

Mostrar las tablas que contiene una base de datos:

show tables;

Seleccionar una base de datos:

use nombre_base;

Describir la estructura de campos de una tabla:

describe nombre_tabla;•Creación de bases de datos, tablas y registros*Crear una base de datos:***create database nombre_base;***Creación de una tabla:*

```
create [temporary] table [if no exists] nombre_tabla (
  nombre_campo tipo1 opciones2 cláusulas3,
  " " " '
  [último campo] );
```

[temporary] --> la tabla existirá mientras exista la conexión con el cliente actual o hasta que se emita la instrucción **drop table**.

[if no exist] --> si existe la tabla no se crea una nueva.

(1) Las posibles opciones de tipo de campo son:

tinyint --> 1 byte
 smallint --> 2 byte
 mediumint --> 3 byte
 int --> 4 byte
 bigint --> 8 byte
 float --> 4 byte
 double --> 8 byte
 decimal --> variable
 char(n) --> cadena de caracteres de longitud fija
 varchar(n) --> cadena de caracteres de longitud variables
 tinyblob --> objeto binario largo (muy pequeño)
 blob --> objeto binario largo (pequeño)
 mediumblob --> objeto binario largo (medio)
 longblob --> objeto binario largo (grande)
 tinytext --> cadena de texto muy pequeña
 text --> cadena de texto pequeña
 mediumtext --> cadena de texto media
 longtext --> cadena de texto larga
 enum --> una enumeración
 set --> un conjunto
 date --> valor fecha (aaaa-mm-dd)
 time --> valor de hora (hh-mm-ss)
 datetime --> valor de fecha y hora
 timestamp --> valor de lapso de tiempo (aaaammddhhmmss)
 year --> valor de año

(2) Las posibilidades del apartado opciones son:

Generales:

null --> admite valores nulos
 no null --> rechaza dejar el campo en blanco
 default --> permite establecer un valor por defecto

Columnas numéricas:

auto_increment --> para generar identificadores únicos o valores en serie.
 unsigned --> rechazo de valores negativos

Cadena:

binary --> trata los valores como cadenas binarias (campos char y varchar)

(3) Cláusulas:

primary key --> columna indexada para búsqueda rápida. Clave principal, solo puede haber una
 unique --> crea un índice con valores únicos
 index, key --> son sinónimos y crean índices que pueden contener valores repetidos.

Inserción de registros en una base de datos:

insert [low_priority | delayed][ignore][into] nombre_tabla forma_introducción¹

I. LOW_PRIORITY | DELAYED: la primera de las opciones hace que la inserción sea postergada hasta que ningún cliente utilice la tabla. La segunda genera una cola de introducción de registros, que se añadirán a la tabla cuando esta quede libre de usuarios.

II. IGNORE: se desechan las filas que duplican valores para claves únicas.

III. INTO: se requiere en versiones 3.22.5.

(1) Las formas de introducción son:

- ⊗(lista_columnas) VALUE (lista_valores_o_expresión)
- ⊗(lista_columnas) SELECT (opción_de_recuperación)
- ⊗SET (nombre_columna=expresion,)

Uso de una base de datos

Use database;

•Consulta de una base de datos:

La forma general del comando de consulta es:

select opciones¹ lista² cláusulas³;

(1) Tipos de opciones permitidas:

all --> provoca el regreso de todas las filas

distinct, distinctrow --> especifican que las líneas duplicadas deben ser eliminadas

high_priority --> da mayor prioridad a la instrucción en el caso de espera.

sql_byg_result, sql_small_result --> especifican que el conjunto de resultados será mas largo o más corto

straight_join --> fuerza a las tablas a unirse en el orden citado en la cláusula from

(2) Lista --> especifica las columnas que deben devolverse, el símbolo "*" especifica que deben extraerse todas las columnas. La separación entre los diferentes nombres de columnas se hacen con el símbolo ",".

(3) Cláusulas (en caso de haber varias cláusulas deben seguir el orden expuesto en la lista):

Intro outfile '*nombre_fichero*' opciones --> el resultado de la consulta es transferido a un fichero, pero lo cual debemos tener privilegio de crear el fichero y además este no debe existir. Las opciones de importación son las siguientes:

[fields [terminate by '*cadena*']^a [optionally] [enclosed by '*carácter*']^b [escape by '*carácter*']^c [lines terminated by '*carácter*']^d

a) especifica el carácter o caracteres que delimitan valores dentro de la línea

b) especifica un carácter entrecomillado que se quita del final del campo de valores. OPTIONALLY establece que los valores solamente se entrecomillan para las columnas char y varchar

c) Se utiliza para especificar el carácter de escape especial

d) Especifica un carácter o caracteres que determinan el final de la línea.

Los caracteres especiales son los siguientes:

- \0 ascii 0
- \b retroceso
- \n línea nueva
- \r retorno de carro
- \s espacio
- \t tabulación
- \' comilla simple

-\' comilla doble
-\\ barra inversa

from *tabla-tablas* --> especifica una o varias tablas desde las que se obtendrán las filas. Existen varias posibilidades de unión de tablas (mirar libro de mysql).

where *expresión* --> establece una expresión que se aplica a las filas seleccionadas

group by *lista_columnas* --> agrupa filas del conjunto de resultados de acuerdo con las columnas citadas.

having *expresión* --> especifica una expresión secundaria para limitar filas después de satisfacer las condiciones expuestas en la cláusula where.

order by *entero_unsigned|nombrecolumna|fórmula* --> indica como se va a clasificar el conjunto de resultados. Las posibilidades son:

asc – ascendente
desc – descendente
rand() – orden aleatorio

procedure *nombre_procedimiento* --> indica un nombre de procedimiento al que serán enviados los datos antes de su salida.

- Alteración de una tabla de datos y registros:

Modificación de la estructura de las tablas: permite renombrar la tabla o variar la estructura de la tabla de datos. La sintaxis base es:

alter [ignore] table nombre¹ lista_de_acción²

La cláusula [ignore] se utiliza si existen datos duplicados en los valores de las claves de los índices, de esta forma no se eliminarán una vez introducido el primero.

(1) Nombre: establece el nombre de la tabla sobre la que deseamos actuar.

(2) Lista_de_acción: establece una o varias acciones a realizar, las posibilidades son:

a) ADD INDEX [nombre _ índice][columna _ índice]: añade un índice a la tabla basado en la columna especificada. Si se especifican varias columnas deben ir separadas por comas.

b) ADD PRIMARY KEY [columna _ índice]: añade una clave principal siguiendo la columna establecida.

c) ADD UNIQUE [nombre _ índice][columna _ índice]: añade un índice de valor único a la tabla en función de la columna especificada.

d) ALTER [column] nombre _ columna (set default valor | drop default): modifica el valor de una columna o reduce su valor predeterminado actual.

e) CHANGE [column] nombre _ columna declaración _ columna: cambia el nombre y la definición de una columna; nombre _ columna -> nombre de la columna a modificar, declaración _ columna -> es el valor de definición por el cual se debe regir la columna, sus opciones son las mismas que las señaladas para la creación de los campos de las tablas.

f) DROP [column] nombre _ columna: elimina la columna especificada.

- g) DROP INDEX nombre_index: elimina el índice de la tabla.
- h) DROP PRIMARY KEY: elimina la clave principal de la tabla. Si no existe clave principal y si índices con especificación UNIQUE se elimina el primero existente.
- i) MODIFY [column] declaración _ columna: cambia la declaración de una columna.
- j) RENAME [AS] nombre_tabla_nueva: cambia el nombre de la tabla.

Borrado de registros: para borrar filas de una tabla se utiliza la expresión:

delete [low_priory] from nombre_tabla [where expresión][limit n]

- ⊗LOW_PRIORY: posterga la petición hasta que ningún usuario este utilizando la tabla.
- ⊗WHERE expresión: especifica una condición que deben cumplir los registros a borrar, si se omite en la petición se eliminan todos los registros de la tabla.
- ⊗LIMIT n: estable un número máximo de registros a borrar.

Sustitución de registro: permite modificar registros existentes.

replace (sigue la misma sintaxis que INSERT).

La salvedad se refiere a los valores correspondientes a índices unique, si el valor existe en la tabla previamente a la inserción de la modificación se eliminara.

Mitificación de registros en bloque:

Updates [low_priority] nombre_tabla set nombre_col=expresion, ...[where expresión_where][limit n]

Borrado de tablas: elimina una o varias tablas:

DROP TABLE [if exists] (lista_tablas_separadas_po_comas)

Borrado de una base de datos. Elimina la base de datos y todas las tablas que contiene.

DROP DATABASE [if exist] nombre_database

Optimización de tablas: optimiza el espacio asignado a la tabla.

OPTIMIZE TABLE nombre_tabla

- Índices, creación, modificación y eliminación

Un índice corresponde a un método de ordenación de una o varias columnas que permite a los motores de búsqueda de bases de datos trabajar de forma eficiente y rápida.

Los índices se pueden crear al crear las tablas con la instrucción *create table* o bien con posterioridad a través del comando *alter table* o el comando:

create [UNIQUE] index nombre _ índice ON nombre_tabla (columna _ índice)

La eliminación de un índice se realiza por el comando:

drop index nombre_index on nombre_tabla

- Instrucciones para mostrar información

Para mostrar las bases de datos disponibles:

show databases [like patrón _ búsqueda]

Para mostrar las tablas de una base de datos:

show tables [from nombre_base_datos][like patrón _ búsqueda]

Para mostrar información descriptiva de una tabla:

show table status [from nombre_base-datos][like patrón _ búsqueda]

El resultado obtenido muestra:

- ⓄName: nombre de la tabla.
- ⓄType: tipo de tablas
- ⓄRow format: formato de almacenamiento de la fila
- ⓄRows: número de filas
- ⓄAvg_row_length: promedio de bytes usados por las filas de la tabla
- ⓄData_length: tamaño actual en bytes de fichero de tabla
- ⓄMax_data_length: tamaño máximo en bytes que puede alcanzar el archivo.
- ⓄIndex_length: tamaño actual en bytes del archivo de índice.
- ⓄData_free: número de bytes en el archivo de datos que no se utilizan.
- ⓄAuto_increment: el siguiente valor generado por una columna con la propiedad.
- ⓄCreate_time: hora que se creó la tabla.
- ⓄUpdate_time: hora de la última modificación
- ⓄCheck_time: última vez que la tabla fue reparada o revisada por myisamchk
- ⓄCreate_options: opciones adicionales especificadas.
- ⓄComment: comentarios aparecidos en la creación de la tabla.

Para mostrar las diferentes columnas de una tabla:

show columns from nombre_tabla [from nombre_base-datos][like patrón-búsqueda]

El resultado de la petición muestra:

- ⓄField: nombre de la columna.
- ⓄType: el tipo de columna.
- ⓄNull: si la columna puede contener valores nulos.
- ⓄKey: si la columna está indexada.
- ⓄDefault: el valor por defecto.
- ⓄExtra: información extra de la columna:
- ⓄPrivileges: los privilegios de la columna.

Para mostrar los índices que contiene una tabla se utiliza las instrucciones:

show index from nombre_tabla
show keys from nombre_tabla

El resultado de la petición se muestra en:

- ⓄTable: el nombre de la tabla que contiene el índice.
- ⓄNom_unique: 1 si puede contener valores duplicados.
- ⓄKey_name: El nombre del índice.
- ⓄSeq_in_index: el número de columnas en el índice.
- ⓄColumn_name: el nombre de la columna.
- ⓄCollation: Orden de clasificación de la columna dentro del índice.
- ⓄCardinality: El número de valores únicos en el índice.
- ⓄSub_part: La longitud del prefijo.
- ⓄPacked: si toda la clave está empaquetada.
- ⓄComment: Reservado para comentarios internos sobre los índices.

Las siguientes variantes de la orden show están relacionadas con la administración del entorno:

Información sobre usuarios:

show grants for nombre_usuario

Información sobre los procesos del servidor:

show processlist

La salida de la información se estructura en los campos:

- ⓄId: el nº id del hilo para el cliente:
- ⓄUser: el nombre del cliente asociado al hilo.
- ⓄHost: el host desde el que se conecta.
- ⓄDb: la base de datos predeterminada para el hilo.
- ⓄCommand: el comando que se ejecuta en el hilo
- ⓄTime: el tiempo en segundos que utiliza el comando.
- ⓄState: Información de lo que hace Mysql mientras se ejecuta una instrucción SQL
- ⓄInfo: ejecuta la consulta.

Información sobre las variables de estado del servidor:

show status

La información dada por el servidor es muy alta:

- ⓄAborted_clients: número de conexiones de un cliente canceladas
- ⓄAborted_connects: número de intentos fallidos de conexión al servidores.
- ⓄConnectios: número de intentos realizados para conectarse al servidor.
- ⓄCreated_tmp_tables: número de tablas temporales creadas mientras se procesan las consultas.
- ⓄDelayed_errors: nº de errores ocurridos mientras se procesan filas *insert delayed*.
- ⓄDelayed_insert_threads: nº de errores actuales del mensaje *insert delayed*.
- ⓄDelayed_writes: nº de filas *insert delayed* introducidas.
- ⓄFlush_commands: nº de columnas flush que se han ejecutado.
- ⓄHendler_delete: nº de solicitudes necesarias para suprimir una fila de una tabla.
- ⓄHendler_read_first: nº de solicitudes para leer la primera fila de la tabla.
- ⓄHandler_read_key: nº de solicitudes para leer una fila basada en un valor de índice.
- ⓄHandler_read_next: nº de solicitudes para leer la siguiente fila basada en un valor de índice.
- ⓄHandler_read_md: nº de solicitudes para leer una fila basada en su posición.
- ⓄHandler_update: nº de solicitudes para actualizar una fila.
- ⓄHandler_write: nº de solicitudes para insertar una fila.
- ⓄKey_blocks_use: nº de bloques en uso en la cache de índice.

- ⊗Key_read_request: nº de solicitudes para leer un bloque desde la cache.
- ⊗Key_reads: nº de lecturas físicas de los bloques de índice desde el disco.
- ⊗Key_write_requests: nº de requisitos para escribir un bloque en la cache de índice.
- ⊗Key_writes: nº de escrituras físicas de los bloques de índices de disco.
- ⊗Max_used_connections: nº máximo de conexiones que se han abierto simultáneamente.
- ⊗Not_flushed_delayed_rows: nº de filas a la espera de ser escritas por las consultas *insert delayed*.
- ⊗Not_flushed_key_blocks: nº de bloques en la cache de clave que han sido modificados pero aun no han sido vaciados del disco.
- ⊗Open_files: nº de archivos abiertos.
- ⊗Open_streams: nº de flujos abiertos.
- ⊗Open_tables: nº de tablas abiertas.
- ⊗Opened_tables: total de tablas que han sido abiertas.
- ⊗Questions: nº de consultas que ha recibido el servidor.
- ⊗Slow_queries: nº de consultas que tardan más segundos en ejecutarse que *long_query_time*.
- ⊗Threads_connected: nº de conexiones abiertas en ese momento.
- ⊗Threads_running: nº de hilos que están latentes.
- ⊗Uptime: Segundos desde que el servidor comenzó a ejecutarse.

Para ver las variables en el servidor se utiliza la forma:

show variables

Administración:•Mysgladmin:*** mysqladmin [opciones] comando**

opciones generales:

--compres, -C -> activa la compresión para el protocolo utilizado en el protocolo cliente-servidor.
 --debug=opcion_depuración, -# opciones _ depurador -> activa la salida del depurador.
 --host =nombre_depurador, -h nombre_host -> especifica a que host debemos conectarnos.
 --port=número_puerto -P número _ puerto -> para los programas clientes este el número de puerto del servidor al que deben conectarse.
 --user=nombre_usuario, -u nombre _ usuario -> nombre del usuario que conecta con el servidor.
 --password , -p ->clave de usuario para acceder al servidor.
 --silent, -s -> especifica modo silencioso, es decir el programa produce menos mensajes de lo habitual.
 --version, -V -> muestra la versión del programa.
 --help, -¿ -> nuestra mensajes de ayuda.
 --pipe, -W ->utiliza una tubería con nombre para conectarse al servidor (solamente se utiliza por programas clientes funcionando bajo Windows).
 --socket=nombre_ruta, -S nombre_ruta -> para los programas clientes indica el fichero de socket que deben usar cuando se conectan al servidor.

opciones específicas:

--force, -f -> hace que no se pida confirmación cuando se ejecuta drop db_nombre y cuando se ejecutan múltiples comando intenta ejecutar todos los comandos aunque se produzcan errores.
 --relative ->muestra la diferencia de los valores anterior y posterior cuando se ejecuta --sleep.
 ----sleep=n, -i n -> ejecuta los comandos dados en la línea de comandos repetidamente cada n segundos.
 --wait[=n], -w [n] -> establece el número de veces a esperar y reintentar si no puede establecer conexión con el servidor.

Comandos:

create db_name -> crea una base de datos con el nombre dado.
 drop db_name -> borra la base de datos con el nombre dado y cualquier tabla que exista en ella.
 flush_host -> vacía el fichero log del servidor.
 flush_status -> limpia las variables de estado.
 flush_tables ->vacía la cache de tablas.
 kill id, id .. -> mata los procesos del servidor.
 password new -> cambia la contraseña de usuario.
 ping -> comprueba si el servidor se está ejecutando.
 process list -> muestra una lista de datos.
 refresk -> vacía la cache de las tablas.
 reload -> recarga la tabla de permisos.
 shutdown -> desconecta el servidor.
 status -> muestra el estado del servidor.
 variables -> muestra los valores y nombres de la variable.
 version -> version del servidor.

•Creación de privilegios de usuarios:

*** grant [privilegio] [columna] on [nivel] to [usuario] identified by "contraseña" with grant option**

privilegios -> si hay varios privilegios se separan por comas.

alter -> alterar tablas e índices.

create -> crear bases de datos y tablas.

delete -> borrar de las tablas registros.

drop -> eliminar bases de datos y tablas.

index -> crear o eliminar índices.

insert -> insertar nuevos registros en las tablas.

references -> no se utiliza.

select -> recuperar registros de tablas.

update -> modificar registros de las tablas.

file -> leer o escribir archivos del servidor.

proces -> ver información sobre los hilos en ejecución dentro del servidor y poder matarlos.

reload -> recargar tablas de concesión.

shutdown -> cerrar el servidor.

all -> cualquier cosa.

usage -> un privilegio especial "sin privilegios".

Columna: indica las columnas a las que se aplicarán los privilegios y es opcional.

Nivel: especifica a que se otorgan los privilegios, que pueden ser globales, de bases de datos o de tablas. Si se especifica *.* indica todas las tablas de todas las bases de datos; nombre_base.* indica todas las tablas de la base de datos; nombre_base.nombre_tabla especifica una tabla de una determinada base de datos.

Usuario: el usuario al que se le otorgan los privilegios y consiste en un usuario y un host ([usuario@host](#)); si no se especifica host se entiende que puede ser desde cualquiera y si no se especifica usuario puede ser cualquiera (anónimo).

Contraseña: la clave de acceso del usuario.

With grant option: se le permite al usuario dar privilegios a otros usuarios. Es una cláusula opcional.

Revocar privilegios:

revoke [privilegios] [columna] on [nivel] from [usuario]

Eliminar usuarios:

delete from user where user=[usuario] and host="nombre_host"

Después de eliminar un usuario es necesario recargar la tabla de usuarios

flush privileges;

- Copias de seguridad

mysqldump [nombre de la base] > [fichero de salida]

Es una orden externa del programa que genera una copia en texto plano de las tablas de la base de datos con todo el conjunto de instrucciones para volver a crearlas e incluir los registros existentes.

La recuperación de la base de datos se realiza de forma simple:

mysql < [nombre fichero de copia de seguridad]

Para obtener una recuperación rápida sin necesidad de operaciones previas debemos introducir en la cabecera del fichero las líneas:

CREATE DATABASE [nombre de la base de datos];
USE [nombre base de datos];

Ⓞ Operaciones diversas:

EXPLAIN

explain nombre_tabla¹
explain sentencia_select²

(1) equivale a la orden que informa sobre las columnas de una tabla.

(2) Da información de como se va a ejecutar la sentencia select.

FLUSH vacía varias caches usadas por el servidor.

flush opcion_flush, ..

opciones_flush pueden ser:

- Ⓞ Hosts: vaciar la cache del servidor.
- Ⓞ Logs: vacia las caches de registros cerrándolos y volviéndolos a abrir
- Ⓞ Privileges: recarga las tablas cedidas.
- Ⓞ Status: vuelve a iniciar los estatus variables.
- Ⓞ Tables: cierra cualquier tabla abierta en la tabla cache.

KILL elimina hilos del servidor.

Kill id_hilo

LOCK TABLES bloqueo de tablas:

lock tables lista_bloqueo

Las opciones de lista _ bloqueo son los nombre de las tablas separadas por comas; las tablas deben tener el formato.

nombre_tabla [as nombre_alias](read | [low_priority] write)

- Ⓞ Read: bloqueo de solo lectura.
- Ⓞ Write: bloqueo de escritura, bloquea la tabla completamente.
- Ⓞ Low_priority: pide permiso para leer la tabla.

UNLOCK TABLES: desbloqueo de las tablas.

Unlock tables

SET: se utiliza para especificar un conjunto de caracteres usando por el cliente.

Set (option) opción_configuración

Las opciones _ configuración pueden ser:

- Ⓞ Carácter set (conjcar_nombre | default) especifica el conjunto de caracteres usado por el cliente.
- Ⓞ Insert_id=n: especifica el valor que se utilizara en la columna auto_incrementen la próxima instrucción insert.

- ⓉLast_insert Id=n: especifica el valor que será devuelto por last_insert_id() -> actualización de procedimiento de registro.
- ⓉPassword [for usuario] = password("contraseña"):sin la cláusula for se establece contraseña para el usuario actual, si se indica se establece para el usuario dado. Hay que tener privilegios de establecimiento de contraseñas.
- ⓉSql_auto_is_null=[0|1]: si se establece en 1, la última fila insert que contiene el valor auto_increment puede seleccionarse usando la cláusula *where auto_inc_coll is null*.
- ⓉSql_big_tables [1|0]: si se establece a 1; se admiten las instrucciones select que aparezcan más que la filas max_join_size.
- ⓉSql_log_off=[0|1]: Si esta opción se establece en 1 las ordenes sql del cliente actual no aparecerán en el registro del archivo general.
- ⓉSql_log_update=[1|0]: es igual que la anterior pero afecta al registro de actualización y no al general.
- ⓉSql_low_priority_updates=[0|1]: si se establece en 1, las instrucciones que modifican la tabla de contenidos esperaran hasta que las instrucciones select terminen.
- ⓉSql_select_limit=(n|default): especifica el nº máximo de registros que se podrán devolver desde una instrucción select.
- ⓉSql_warnings=(1|0): si se establece en 1 mysql informa de los errores tipo 'aviso'.
- ⓉTimestamp=(timestamp_value|default): especifica un valor timestamp de actualización del registro actual.

Anexo C Servidores

¿Que es un servidor?

En Internet, un servidor es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras.

En redes locales se entiende como el software que configura un PC como servidor para facilitar el acceso a la red y sus recursos. Los Servidores almacenan información en forma de páginas web y a través del protocolo HTTP lo entregan a petición de los clientes (navegadores web) en formato HTML.

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen hosting o hospedaje. Alternativamente, el servidor web podría referirse al software, como el servidor de http de Apache, que funciona en la máquina y maneja la entrega de los componentes de los páginas web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor. En la web, un servidor web es un ordenador que usa el protocolo http para enviar páginas web al ordenador de un usuario cuando el usuario las solicita.

Los servidores web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet.

Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que un mismo ordenador puede tener diferentes programas de servidor funcionando al mismo tiempo.

Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía línea telefónica o digital.

Distintos tipos de servidores del mercado actual:

Plataformas de Servidor (*Server Platforms*): Un término usado a menudo como sinónimo de sistema operativo, la plataforma es el hardware o software subyacentes para un sistema, es decir, el motor que dirige el servidor.

Servidores de Aplicaciones (*Application Servers*): Designados a veces como un tipo de *middleware* (software que conecta dos aplicaciones), los servidores de aplicaciones ocupan una gran parte del territorio entre los servidores de bases de datos y el usuario, y a menudo los conectan.

Servidores de Audio/Video (*Audio/Video Servers*): Los servidores de Audio/Video añaden capacidades multimedia a los sitios web permitiéndoles mostrar contenido multimedia en forma de flujo continuo (*streaming*) desde el servidor.

Servidores de Chat (*Chat Servers*): Los servidores de chat permiten intercambiar información a una gran cantidad de usuarios ofreciendo la posibilidad de llevar a cabo discusiones en tiempo real.

Servidores de Fax (*Fax Servers*): Un servidor de fax es una solución ideal para organizaciones que tratan de reducir el uso del teléfono pero necesitan enviar documentos por fax.

Servidores FTP (*FTP Servers*): Uno de los servicios más antiguos de Internet, (File Transfer Protocol) permite mover uno o más archivos con seguridad entre distintos ordenadores proporcionando seguridad y organización de los archivos así como control de la transferencia. La seguridad se ha convertido en un tema candente. Durante años, los servidores ftp comunicaban con los clientes "en abierto," es decir, que la información de la conexión y de la contraseña eran vulnerables a la interceptación. Ahora, los servidores ftp, tales como BulletProof FTP, SecureFTP, SurgeFTP, TitanFTP, y WS_FTP, soportan SSL/TLS y utilizan el mismo tipo de cifrado presente en los sitios web seguros. Con SSL/TLS, los servidores ftp pueden cifrar los comandos de control entre los clientes del ftp y el servidor, así como los datos del archivo. Con la ayuda del PGP, como en WS_FTP pro, los datos del archivo se aseguran todavía más con el cifrado público.

Servidores Groupware (*Groupware Servers*): Un servidor groupware es un software diseñado para permitir colaborar a los usuarios, sin importar la localización, vía Internet o vía Intranet corporativo y trabajar juntos en una atmósfera virtual.

Servidores IRC (*IRC Servers*): Otra opción para usuarios que buscan la discusión en tiempo real, Internet Relay Chat consiste en varias redes de servidores separadas que permiten que los usuarios conecten el uno al otro vía una red IRC.

Servidores de Listas (*List Servers*): Los servidores de listas ofrecen una manera mejor de manejar listas de correo electrónico, bien sean discusiones interactivas abiertas al público o listas unidireccionales de anuncios, boletines de noticias o publicidad.

Servidores de Correo (*Mail Servers*): Casi tan ubicuos y cruciales como los servidores web, los servidores de correo mueven y almacenan el correo electrónico a través de las redes corporativas (vía LANs y WANs) y a través de Internet.

Servidores de Noticias (*News Servers*): Los servidores de noticias actúan como fuente de distribución y entrega para los millares de grupos de noticias públicos actualmente accesibles a través de la red de noticias USENET.

Servidores Proxy (*Proxy Servers*): Los servidores proxy se sitúan entre un programa del cliente (típicamente un navegador) y un servidor externo (típicamente otro servidor web) para filtrar peticiones, mejorar el funcionamiento y compartir conexiones.

Servidores Telnet (*Telnet Servers*): Un servidor telnet permite a los usuarios entrar en un ordenador huésped y realizar tareas como si estuviera trabajando directamente en ese ordenador.

Servidores Web (*Web Servers*): Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI, seguridad SSL y páginas activas del servidor (ASP). [9]

Normalización de bases de datos

Se explican los conceptos de la normalización de bases de datos, mismos que son necesarios para un buen diseño de una base de datos.

Qué es la normalización

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataban de manipular los datos.

La normalización también hace las cosas fáciles de entender. Los seres humanos tenemos la tendencia de simplificar las cosas al máximo. Lo hacemos con casi todo, desde los animales hasta con los automóviles. Vemos una imagen de gran tamaño y la hacemos más simple agrupando cosas similares juntas. Las guías que la normalización provee crean el marco de referencia para simplificar una estructura de datos compleja.

Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco.

El proceso de normalización tiene un nombre y una serie de reglas para cada fase. Esto puede parecer un poco confuso al principio, pero poco a poco se va entendiendo el proceso, así como las razones para hacerlo de esta manera.

Grados de normalización

Existen básicamente tres niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización. No siempre es una buena idea tener una base de datos conformada en el nivel más alto de normalización, puede llevar a un nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

En la tabla siguiente se describe brevemente en que consiste cada una de las reglas, y posteriormente se explican con más detalle.

Regla	Descripción
Primera Forma Normal (1FN)	Incluye la eliminación de todos los grupos repetidos.
Segunda Forma Normal (2FN)	Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK).
Tercera Forma Normal (3FN)	Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

Primera Forma Normal

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Poner la base de datos en la Primera Forma Normal resuelve el problema de los encabezados de columna múltiples. Muy a menudo, los diseñadores de bases de datos inexpertos harán algo similar a la tabla no normalizada. Una y otra vez, crearán columnas que representen los mismos datos. La normalización ayuda a clarificar la base de datos y a organizarla en partes más pequeñas y más fáciles de entender. En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos diferentes aspectos, sólo tenemos que entender los objetos pequeños y más tangibles, así como las relaciones que guardan con otros objetos también pequeños.

Segunda Forma Normal

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos.

Una vez alcanzado el nivel de la Segunda Forma Normal, se controlan la mayoría de los problemas de lógica. Podemos insertar un registro sin un exceso de datos en la mayoría de las tablas.

Tercera Forma Normal

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Comentamos anteriormente que una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen de otras columnas que tampoco son llave.

Cuando las tablas están en la Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria, y no deben haber datos repetidos. Esto provee un esquema limpio y elegante, que es fácil de trabajar y expandir.

Un dato sin normalizar no cumple con ninguna regla de normalización. Para explicar con un ejemplo en que consiste cada una de las reglas, vamos a considerar los datos de la siguiente tabla.

ID_ORDEN	FECHA	ID_CLIENTE	NOM_CLIENTE	ESTADO	NUM_ITEM	DESC_ITEM	CANT	PRECIO
2301	2/23/03	101	MARTI	CA	3786	RED	3	35
2301	2/23/03	101	MARTI	CA	4011	RAQUETA	6	65
2301	2/23/03	101	MARTI	CA	9132	PAQ-3	8	4.75
2302	2/25/03	107	HERMAN	WI	5794	PAQ-6	4	5.0
2303	2/27/03	110	WE-SPORTS	MI	4011	RAQUETA	2	65
2303	2/27/03	110	WE-SPORTS	MI	3141	FUNDA	2	10

Al examinar estos registros, podemos darnos cuenta que contienen un grupo repetido para NUM_ITEM, DESC_ITEM, CANT y PRECIO. La 1FN prohíbe los grupos repetidos, por lo tanto tenemos que convertir a la primera forma normal. Los pasos a seguir son:

- » Tenemos que eliminar los grupos repetidos.
- » Tenemos que crear una nueva tabla con la PK de la tabla base y el grupo repetido.

Normalización de bases de datos

Los registros quedan ahora conformados en dos tablas que llamamos ORDENES y ARTICULOS_ORDENES

- ORDENES

ID_ORDEN	FECHA	ID_CLIENTE	NOM_CLIENTE	ESTADO
2301	2/23/03	101	MARTI	CA
2302	2/25/03	107	HERMAN	WI
2303	2/27/03	110	WE-SPORTS	MI

- ARTICULOS_ORDENES

ID_ORDEN	NUM_ITEM	DESC_ITEM	CANT	PRECIO
2301	3786	RED	3	35
2301	4011	RAQUETA	6	65
2301	9132	PAQ-3	8	4.75
2302	5794	PAQ-6	4	5.0
2303	4011	RAQUETA	2	65
2303	3141	FUNDA	2	10

Ahora procederemos a aplicar la segunda forma normal, es decir, tenemos que eliminar cualquier columna no llave que no dependa de la llave primaria de la tabla. Los pasos a seguir son:

- » Determinar cuáles columnas que no son llave no dependen de la llave primaria de la tabla.
- » Eliminar esas columnas de la tabla base.
- » Crear una segunda tabla con esas columnas y la(s) columna(s) de la PK de la cual dependen.

La tabla ORDENES está en 2FN. Cualquier valor único de ID_ORDEN determina un sólo valor para cada columna. Por lo tanto, todas las columnas son dependientes de la llave primaria ID_ORDEN.

Por su parte, la tabla ARTICULOS_ORDENES no se encuentra en 2FN ya que las columnas PRECIO y DESC_ITEM son dependientes de NUM_ITEM, pero no son dependientes de ID_ORDEN. Lo que haremos a continuación es eliminar estas columnas de la tabla ARTICULOS_ORDENES y crear una tabla ARTICULOS con dichas columnas y la llave primaria de la que dependen.

Las tablas quedan ahora de la siguiente manera.

- ARTICULOS_ORDENES

ID_ORDEN	NUM_ITEM	CANT
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

- ARTICULOS

NUM_ITEM	DESC_ITEM	PRECIO
3786	RED	35
4011	RAQUETA	65
9132	PAQ-3	4.75
5794	PAQ-6	5.0
4011	RAQUETA	65
3141	FUNDA	10

La tercera forma normal nos dice que tenemos que eliminar cualquier columna no llave que sea dependiente de otra columna no llave. Los pasos a seguir son:

- » Determinar las columnas que son dependientes de otra columna no llave.
- » Eliminar esas columnas de la tabla base.
- » Crear una segunda tabla con esas columnas y con la columna no llave de la cual son dependientes.

Al observar las tablas que hemos creado, nos damos cuenta que tanto la tabla ARTICULOS, como la tabla ARTICULOS_ORDENES se encuentran en 3FN. Sin embargo la tabla ORDENES no lo está, ya que NOM_CLIENTE y ESTADO son dependientes de ID_CLIENTE, y esta columna no es la llave primaria.

Para normalizar esta tabla, moveremos las columnas no llave y la columna llave de la cual dependen dentro de una nueva tabla CLIENTES. Las nuevas tablas CLIENTES y ORDENES se muestran a continuación.

- ORDENES

ID_ORDEN	FECHA	ID_CLIENTE
2301	2/23/03	101
2302	2/25/03	107
2303	2/27/03	110

- CLIENTES

ID_CLIENTE	NOM_CLIENTE	ESTADO
101	MARTI	CA
107	HERMAN	WI
110	WE-SPORTS	MI

¿Qué tan lejos debe llevar la normalización?

La siguiente decisión es ¿qué tan lejos debe llevar la normalización? La normalización es una ciencia subjetiva. Determinar las necesidades de simplificación depende de nosotros. Si nuestra base de datos va a proveer información a un solo usuario para un propósito simple y existen pocas posibilidades de expansión, normalizar los datos hasta la 3FN quizá sea algo exagerado. Las reglas de normalización existen como guías para crear tablas que sean fáciles de manejar, así como flexibles y eficientes. A veces puede ocurrir que normalizar los datos hasta el nivel más alto no tenga sentido.

¿Se están dividiendo tablas sólo para seguir las reglas o estas divisiones son en verdad prácticas?. Éstas son el tipo de cosas que nosotros como diseñadores de la base de datos, necesitamos decidir, y la experiencia y el sentido común nos pueden auxiliar para tomar la decisión correcta. La normalización no es una ciencia exacta, más bien subjetiva.

Existen seis niveles más de normalización que no se han discutido aquí. Ellos son Forma Normal Boyce-Codd, Cuarta Forma Normal (4NF), Quinta Forma Normal (5NF) o Forma Normal de Proyección-Unión, Forma Normal de Proyección-Unión Fuerte, Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio. Estas formas de normalización pueden llevar las cosas más allá de lo que necesitamos. Éstas existen para hacer una base de datos realmente relacional. Tienen que ver principalmente con dependencias múltiples y claves relacionales.

En resumen

La normalización es una técnica que se utiliza para crear relaciones lógicas apropiadas entre tablas de una base de datos. Ayuda a prevenir errores lógicos en la manipulación de datos. La normalización facilita también agregar nuevas columnas sin romper el esquema actual ni las relaciones.

Existen varios niveles de normalización: Primera Forma Normal, Segunda Forma Normal, Tercera Forma Normal, Forma Normal Boyce-Codd, Cuarta Forma Normal, Quinta Forma Normal o Forma Normal de Proyección-Unión, Forma Normal de Proyección-Unión Fuerte, Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio. Cada nuevo nivel o forma nos acerca más a hacer una base de datos verdaderamente relacional.

Se discutieron las primeras tres formas. Éstas proveen suficiente nivel de normalización para cumplir con las necesidades de la mayoría de las bases de datos. Normalizar demasiado puede conducir a tener una base de datos ineficiente y hacer a su esquema demasiado complejo para trabajar. Un balance apropiado de sentido común y práctico puede ayudarnos a decidir cuándo normalizar.

Glosario

Open source: Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. La idea del código abierto se centra en la premisa de que al compartir el código, el programa resultante tiende a ser de calidad superior al software propietario, es una visión técnica.

Sitio web: Es una colección de páginas web relacionadas y comunes a un dominio de Internet o subdominio en la World Wide Web en Internet. Una página web es un documento HTML/XHTML que es accesible generalmente mediante el protocolo HTTP de Internet.

Sistema embebido o empotrado es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas, los sistemas embebidos se diseñan para cubrir necesidades específicas

RPM Package Manager (o RPM, originalmente llamado Red Hat Package Manager) es una herramienta de administración de paquetes pensada básicamente para GNU/Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas.

URL localizador de recursos uniforme,(sigla en inglés de uniform resource locator): es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones digitales, etc.

Navegador o navegador web (del inglés, web browser) es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que estos puedan ser leídos (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local).

Vínculo: es una conexión desde un recurso web a otro. Un vínculo tiene dos extremos (llamados en inglés anchors, anclas), y una dirección. El vínculo comienza en el "ancla de origen" (origen del vínculo) y apunta al "ancla destino" (destino del vínculo), que puede ser cualquier recurso de la Web por ejemplo una imagen, un videoclip, un archivo de sonido, un programa, un documento HTML, un elemento dentro de un documento HTML, etc.

Bibliografía y Referencias

- 1.-<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>
- 2.-<http://www.htmla.com.ar/>
- 3.-<http://en.wikipedia.org/wiki/MyEclipse>
- 4.-<http://es.wikipedia.org/wiki/SQL>
- 5.-<http://www.sinemed.com/recursos/docs/MySQL.pdf>
- 6.-<http://www.javaya.com.ar/>
- 7.-<http://es.answers.yahoo.com/question/index?qid=20110525141740AAwO6Vp>
- 8.-<http://es.wikipedia.org/wiki/Servidor>
- 9.- <http://www.masadelante.com/faqs/tipos-de-servidores>
- 10.- <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node15.html>