



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN**

**“AGENCIA DE CONSULTA Y CONTRATACIÓN  
PARA GRUPOS MUSICALES”**

**TRABAJO ESCRITO  
EN LA MODALIDAD DE SEMINARIOS  
Y CURSOS DE ACTUALIZACIÓN Y  
CAPACITACIÓN PROFESIONAL  
QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN**

**P R E S E N T A:  
JOSÉ LUIS BUSTAMANTE BUSTAMANTE**

**ASESOR:  
M. EN C. JESÚS HERNÁNDEZ CABRERA**



**México, D.F. 2009**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## Dedicatorias

Es un poco difícil realizar una dedicatoria, pues es mucha gente la que ha ayudado para que pueda llegar a este punto de mi trayectoria universitaria y profesional, sin embargo este trabajo va con dedicación en especial:

A **Guadalupe Tiburcia y Luis Bustamante**, mis padres. Por tomar la decisión de que yo existiera, por su temple de ayudarme y soportarme sin ninguna restricción, así mismo por enseñarme a ser un hombre de bien en todos los aspectos y darme el ejemplo de cómo se debe caminar por la vida y por transmitirme todos los valores que rigen y dan sentido a mi existencia.

A **Banco Azteca**, mi empresa. Por haberme dado la oportunidad de poner en práctica todos mis conocimientos adquiridos durante el transcurso de la Carrera y el Diplomado y así poderme desenvolver en el ámbito profesional.

A la **UNAM**, mi universidad. Por ser la mejor Universidad de Latino América, realmente me siento muy orgulloso de haber forjado mis estudios académicos en esta gran institución, en ella aprendí a ser un profesionista con capacidad de análisis, pensamiento individual y creatividad.

A la **FES ARAGÓN**, mi facultad. Por ser la escuela en donde conocí a verdaderos profesores con ganas de impartir todos sus conocimientos a sus alumnos, además de ser grandes personas en el ámbito profesional y personal, gracias a esto pude adquirir los conocimientos necesarios para poder salir adelante en el transcurso de la carrera. Así mismo, por ser la institución en donde conocí a verdaderos compañeros y amigos universitarios.

Finalmente, a mi **Padre Dios**. Por darme la salud necesaria para salir adelante y no desvanecer en ningún momento, por poner a este hijo en este espacio de tiempo para realizar las tareas que me ha encomendado, y por el soporte espiritual mayor.

## **Agradecimientos**

Agradezco infinitamente a todos quienes han colaborado en esta tarea. Especialmente a mi asesor, el profesor Jesús Hernández Cabrera, quien con sus conocimientos, actitud y paciencia, me ayudó bastante a poder finalizar este trabajo de titulación.

Al jefe de la Carrera de Ingeniería en Computación, el Maestro Marcelo Pérez Medel y a todos mis revisores de Tesis, por su gran ayuda para poder hacer realidad un sueño más en mi vida.

A Alma, por ser una gran persona y entender en muchas ocasiones mis múltiples actividades, comprenderme y apoyarme en mi desarrollo personal, universitario y profesional.

A mis amigos, que siempre me han acompañado para llevar a buen término mi carrera universitaria.

Y a mis compañeros de carrera que siempre he podido contar con ellos en todo momento y nunca me han defraudado.

**“AGENCIA DE CONSULTA Y CONTRATACIÓN PARA  
GRUPOS MUSICALES”**

## Agencia de Consulta y Contratación para Grupos Musicales

Índice	Página
Introducción	7
Antecedentes y Justificación	8
Objetivo	9
<b>1. Marco Teórico</b>	
1.1 Sistemas de Información	11
1.2 Arquitectura de los Sistemas de Información	11
1.2.1 Arquitectura de Dos Capas	11
1.2.2 Arquitectura de Tres Capas	12
1.3 Instalación y Configuración de servicios para WWW	14
1.3.1 Servidor Web Apache	14
1.3.2 Módulos de Apache	14
1.3.3 Tomcat	15
1.3.4 Mod Jk	15
1.4 UML (Unfied Modeling Language – Lenguaje De Modelado Unificado)	16
1.5 Lenguajes de Programación Script e Intérpretes	16
1.6 Introducción a Php	17
1.6.1 Características de Php	18
1.7 Bases de Datos	19
1.7.1 Modelo Entidad-Relación (E/R)	19
1.7.2 Manejador de Bases de Datos	21
1.8 Aplicaciones de Bases de Datos en Web	23
<b>2. Análisis del Sistema</b>	
2.2 Definición y planteamiento del problema	25
2.2 Entrevista con el usuario	25
2.3 Requerimientos Funcionales	26
2.4 Requerimientos No Funcionales	28
2.4.1 Recursos de Hardware	28
2.4.2 Recursos de Software	30
2.5 Utilizando UML para el Análisis del Sistema	31
2.5.1 Diagrama de Casos de Uso General	31
2.5.2 Narrativas de Casos de Uso	32

### **3. Diseño del Sistema**

3.1	Introducción	47
3.2	Modelo Conceptual	47
3.3	Diagrama de Clases	49
3.3.1	Definición de Clase	49
3.3.2	Clases Boundary	49
3.3.3	Clases Entity	50
3.3.4	Clases de Control	50
3.4	Diagrama de Estados	51
3.4.1	Diagrama para la Navegación del Usuario	52
3.4.2	Diagrama para la Navegación del Administrador	53
3.4.3	Diagrama para la Navegación del Gerente	54
3.4.4	Diagrama para la Navegación del Representante	55
3.5	Diagrama de Actividades para el Usuario Final	56
3.6	Diagrama de Secuencias	57

### **4. Implementación del Sistema**

4.1	Introducción	64
4.2	Diseño de la Base de Datos	64
4.3	Modelo Entidad-Relación	65
4.4	Diccionario de Datos	68
4.5	Creación de la Base de Datos	70
4.5.1	Creación de Tablas, Llaves Primarias y Foráneas	71
4.5.2	Inserción de Datos	75
4.6	Diseño de Interfaces	80
4.7	Codificación de la Capa de Negocio	81

Conclusiones	109
--------------	-----

Bibliografía	110
--------------	-----

## Introducción

En la actualidad, en el mundo del Internet las agencias de cualquier tipo se han vuelto de gran importancia para todo el público en general, además de ser éstas gran negocio para sus dueños y socios. Hoy en día, los trámites en Internet que se realizan en agencias tienen muchas ventajas para todos los usuarios interesados en obtener informes y detalles sobre los servicios que ofrecen estas agencias, además de agilizar considerablemente estos trámites que por otras vías son muy tardados de realizar.

Los distintos servicios que ofrecen las agencias son de gran utilidad para aquellas personas interesadas en la contratación de alguno de estos. El hecho de que las agencias ofrezcan sus portales por Internet, hacen más fácil al usuario el poder obtener información del servicio a contratar, además del trámite que deben seguir para la contratación de dicho servicio.

Hoy en día, cualquier agencia ofrece su portal por Internet, aunque debemos saber que detrás de todo esto, existe un gran análisis y desarrollo por parte de expertos en lo que al tema se refiere, es decir, existen diseñadores gráficos, analistas, desarrolladores en distintos lenguajes de programación involucrados para realizar un buen trabajo y así poderlo ofrecer a los usuarios y cumplir las expectativas al 100%.

Para realizar un excelente trabajo en la creación de un sistema Web, en la actualidad existe una gran diversidad de Herramientas y Lenguajes de Programación que ayudan y son de gran utilidad para los Analistas, Líderes de Proyecto, Desarrolladores y todos los involucrados en el Análisis y Diseño de un Sistema Web.

Las Bases de Datos representan la esencia para lograr la organización, automatización y control de la información, las cuales se han visto sorprendidas por la revolución que se ha dado, más sin embargo, los logros han sido más que satisfactorios.

Existe una gran gama de Sistemas Web basados en portales para diferentes tipos de agencias, las cuales ofrecen servicios y ventajas para los usuarios. En éste trabajo el tema será enfocado al análisis y diseño de una Agencia basada en la consulta y contratación para Grupos Musicales.

## Antecedentes y Justificación

Desde los orígenes de la civilización, el ser humano siempre ha estado involucrado con diferentes sonidos. Como actividades musicales comenzó a utilizar la danza y las ceremonias religiosas. A la vez fue elaborando las técnicas de los sonidos, el hombre comenzó a cantar acompañándose con palmas y golpes con los pies.

Conforme avanzó la historia de la música fueron surgiendo infinidad de elementos que hicieron que ésta ciencia fuera haciéndose más perfecta en todo el mundo. Surgieron elementos como el pentagrama, el diapasón, el metrónomo, infinidad de instrumentos musicales, diferentes ritmos musicales, entre otras muchas cosas más.

En la actualidad la música viva se ha convertido en un talismán para los diversos eventos sociales que se presentan hoy en día. Existen ritmos musicales para todo gusto, podemos citar algunos como: el rock, pop, jazz, salsa, electrónico, mariachi, banda, norteño, etc. Así mismo existen infinidad de grupos tocando estos diferentes ritmos. Aquí debemos señalar que existen dos tipos de agrupaciones: una que son los grupos versátiles que son los que tocan todos los ritmos musicales y la otra son los grupos que solo tocan un ritmo musical ya sea rock, salsa o cualquier otro.

Hoy en día la sociedad contrata grupos musicales para sus diversos eventos sociales. Por lo regular es raro no ver en alguna fiesta o evento social música viva. Cada agrupación musical tiene su forma de trabajar en lo que a costos se refiere, de igual manera lleva una agenda para su organización de eventos. Las agrupaciones musicales por lo regular están afiliadas a Sindicatos o a secciones de éstos. La máxima organización es el Sindicato Nacional de Trabajadores de la Música de la República Mexicana (SNTMRM). México es un país que se destaca por sus fiestas y como ya mencioné anteriormente, en donde hay fiesta hay música viva.

En ocasiones las personas no tienen ni idea de cómo es la organización de un grupo musical, haciendo referencia al número de integrantes, instrumentos musicales, equipo de sonido, repertorio, etc., y mucho menos tienen idea del costo de algún grupo de estos. Suele suceder que en ocasiones las personas para sus eventos quieren contratar un grupo de salsa, pero tal vez para otro evento ahora quieran un grupo norteño y no tengan idea de cuánto cobra; así como éstas hay muchas inquietudes que el público en general tiene a la hora de enfocarse en la parte musical de sus eventos.

Consciente de todas esas exigencias e inquietudes surge la necesidad de desarrollar un sistema que sirva de apoyo a todo el público en general que esté interesado en la contratación de algún grupo musical para cualquier evento social, el sistema contará con toda la información necesaria de cada grupo de distinto género dado de alta en el sistema.

### **Objetivo General**

Desarrollar un sistema que permita al público en general, consultar toda la información necesaria de los diferentes grupos musicales dados de alta en el sistema, así como poder contratarlos para diferentes eventos sociales.

# **CAPÍTULO 1**

## **MARCO TEÓRICO**

## 1.1 Sistemas de información

Un sistema de información se diseña a fin de satisfacer las necesidades de información de una organización y esta inmerso en ella. El sistema de información ha de tomar los datos del entorno y sus resultados han de ser la información que dicha organización necesita para su gestión y toma de decisiones.

## 1.2 Arquitectura de los sistemas de información

La arquitectura de un sistema de información es la forma en que se proyecta y construye el mismo, esta forma de elaboración nos dice las partes que constituyen al sistema y el modo en que éstas se relacionan e interactúan entre si.

Todo sistema de información tiene código de presentación, de reglas y de manejo de datos. La arquitectura de un sistema dependerá de cómo esté distribuido este código.<sup>1</sup>

Existen dos tipos de arquitectura de los sistemas de información: la arquitectura de dos capas, y la arquitectura de tres capas; las cuales analizaremos a continuación para poder elegir la mejor opción que será la base de nuestro sistema.

### 1.2.1 Arquitectura de Dos Capas

La arquitectura de 2 capas está constituida por lo siguiente:

**Capa 1:** Es donde básicamente se encuentran los Datos del sistema y en la práctica se refiere a la base de datos.

**Capa 2:** En esta capa se encuentran los procesos, algoritmos, reglas, procedimientos y restricciones aplicables al usuario del sistema, lo anterior se conoce como la lógica de negocios.

También se encuentra en ésta la interfaz del usuario que básicamente está formada por ventanas, menús, cuadros de diálogo, botones, reportes y todos los controles necesarios para que el usuario interactúe con el sistema.

La tecnología en dos capas tiene los siguientes problemas:

---

<sup>1</sup> Entendemos por código a una serie de instrucciones que nos permiten darle órdenes a la computadora para que ejecute ciertas operaciones y procesos.

- Cliente pesado ya que realiza una gran parte del procesamiento de la aplicación localmente.
- El mantenimiento es difícil y costoso en cada cliente, debido a que la revisión de los archivos que contienen la interfaz y la lógica de negocios se vuelve complicada por el extenso número de líneas de código y se complica más aún si dicho código no se encuentra documentado.
- Existe un difícil manejo de la interfaz debido a que para cambiar la interfaz es necesario también cambiar la lógica de programa.

Esto trae como consecuencia de que sería un sistema cerrado, ya que en caso de aparecer nuevas tecnologías de información es complicado diseñar nuevamente la interfaz para utilizarse.

En términos generales esta arquitectura es muy costosa para su mantenimiento, difícil de escalar y pesada de depurar. Muchas soluciones actualmente utilizan esta arquitectura y sobre todo aquellas que son cliente/servidor no Internet.

### **1.2.2 Arquitectura de Tres Capas**

Derivado de los problemas antes mencionados aparece la tecnología en 3 capas, la cual se compone por:

**Capa de interfaz de usuario.** Esta capa contiene la interfaz con la cual el usuario interactúa con el sistema. Dicha interfaz contiene propiamente las ventanas, menús, gráficos, cuadros de diálogo, botones, reportes, elementos multimedia en general, modelo de navegación del sistema y los adaptadores para cada modo de acceso. Su función principal es la de captura y visualización de datos, que se realiza mediante peticiones que atiende la capa lógica de negocios.

En un sistema basado en Web ésta capa esta compuesta por distintas páginas Web que presentan los datos en un navegador (Internet Explorer, Netscape Navigator, Opera, etc.) tanto en una PC como en los distintos dispositivos móviles (celulares y Palm). Puede estar en el servidor o en el cliente.

**Capa de Datos.** En ésta encontramos todo lo relacionado a la administración, acceso, almacenamiento, actualización, consulta y recuperación de datos contenidos en el sistema. Típicamente la capa esta conformada por la Base de Datos compuesta por sus respectivas tablas y registros que son administrados por un manejador de base de datos<sup>2</sup> y siempre se localizará en el servidor.

---

<sup>2</sup> DBMS por sus siglas en ingles Doto Base Management System

Esta capa solo se comunica con la capa lógica de negocios para el manejo de los datos que puede incluir el ingreso y consulta de información.

La capa de datos puede estar en el mismo servidor que las de lógica de negocio y presentación, ó en un servidor independiente.

**Capa lógica de Negocios.** En esta capa se encuentran todos los procesos, algoritmos, procedimientos, reglas, lógica, validaciones, cálculos y restricciones aplicables que requiere el sistema de información para tener una funcionalidad específica.

La capa entra en funcionamiento al recibir peticiones del usuario a través de la capa de interfaz de usuario, una vez procesada dicha información ésta capa se comunicará con la Capa de datos para insertar, extraer y/o actualizar información, y por último enviará a la capa de interfaz de usuario el resultado de la petición.

Puede estar en el servidor, dependerá de las necesidades y naturaleza del sistema; así mismo nos permite una integración más sencilla y eficaz con sistemas externos.

Las principales ventajas de esta forma de trabajo aplicada a los sistemas de información son:

- Tiene un mantenimiento fácil y barato.
- Los datos son muy aparte de la lógica de negocios.
- Existe un manejo más fácil de la interfaz, por lo que podemos decir que un sistema en esta arquitectura es flexible.
- Permite la implementación de un cliente ligero, ya que solo realiza las tareas de presentación, esto permite que la computadora del cliente sea menos potente.
- Permite una gran variedad de entornos distintos de sistemas operativos en el cliente.
- Además reduce la complejidad de administración, instalación, configuración y mantenimiento del software en el cliente.

Las aplicaciones en tres capas típicamente tienen mayor capacidad de crecimiento y son más sencillas de mantener, dada su naturaleza altamente modular, aunque cabe aclarar que este tipo de arquitectura es la más compleja.

### **1.3 Instalación y Configuración de servicios para WWW**

En la actualidad para los Servicios Web y para las diversas Aplicaciones empresariales se necesita trabajar sobre algún Servidor, para este trabajo se utilizará el Servidor Web Apache, debido a que es un Software totalmente libre y que además tiene las características indispensables para que pueda acoplarse al Sistema Web, entre sus características principales están en que corre sobre una multitud de plataformas y Sistemas Operativos, además ofrece tecnología libre y de código abierto, otorgándole una transparencia y dando la posibilidad de conocer que es lo que realmente estamos instalando, es configurable y de diseño modular, capaz de ampliar su funcionalidad y calidad de servicios.

Por otro lado tenemos el Servidor Web Tomcat, el cuál también se va a utilizar para el Sistema Web, debido a que es capaz de soportar servlets y JSP's, los cuales se van a utilizar para el Sistema.

A continuación se hablará un poco más de éstos servicios los cuales se utilizarán para que el Sistema Web pueda implementarse de una manera exitosa y de igual manera tenga un funcionamiento correcto.

#### **1.3.1 Servidor Web Apache**

El servidor Web Apache es un software (libre), servidor HTTP de código abierto para plataformas Unix, Linux, Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular National Center for Supercomputing Applications (NCSA) HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache tiene amplia aceptación en la red: desde el 2006, Apache fue el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado. Hoy en día Apache es el Servidor más popular y utilizado en el mundo entero.

#### **1.3.2 Módulos de Apache**

La arquitectura del servidor Apache es modular, consta de un núcleo y mucha de la funcionalidad que podría considerarse básica para un servidor Web es provista por módulos.

Algunos de estos son:

- mod\_ssl - Comunicaciones Seguras vía TLS.
- mod\_rewrite - Reescritura de direcciones servidas (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a

los navegantes o a los motores de búsqueda en cuanto a como fueron desarrolladas estas páginas).

- mod\_dav - Soporte del protocolo WebDAV (RFC 2518).
- mod\_deflate - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- mod\_auth\_ldap - Permite autenticar usuarios contra un servidor LDAP.
- mod\_proxy\_ajp - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).

El servidor base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- mod\_perl - Páginas dinámicas en Perl.
- mod\_php - Páginas dinámicas en PHP.
- mod\_python - Páginas dinámicas en Python.
- mod\_rexx - Páginas dinámicas en REXX y Object REXX.
- mod\_ruby - Páginas dinámicas en Ruby.
- mod\_aspdotnet - Páginas dinámicas en .NET de Microsoft (Módulo retirado).
- mod\_security - Filtrado a nivel de aplicación, para seguridad.

### **1.3.3 Tomcat**

Tomcat es un servidor Web con soporte de servlets y JSP's. Incluye el compilador Jasper, que compila JSP's convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

### **1.3.4 Mod Jk**

Es un conector que permite al contenedor de Java Server Pages (JSP) Jakarta Tomcat interactuar con servidores Web como Apache, Netscape, iPlanet, SunOne e incluso IIS usando el protocolo AJP.

La principal funcionalidad de este módulo es permitir a servidores de aplicaciones o al servidor Jakarta Tomcat enlazarse con un servidor Web. Este servidor Web, típicamente el servidor HTTP Apache, introduce una mayor gestión en las conexiones de los clientes y mayor la seguridad en las transacciones del sistema. Así mismo se puede enlazar varias instancias al servidor Web permitiendo así una mayor tolerancia a errores y aligerar la carga en los servidores Java.

## 1.4 UML (Unified Modeling Language – Lenguaje De Modelado Unificado)

Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software a partir de un conjunto de diagramas; no define un proceso de desarrollo específico, tan solo se trata de una notación que permite crear diseños que capturan las ideas de los analistas de sistemas en forma convencional y fácil de comprender para comunicarlas a otras personas.

## 1.5 Lenguajes de Programación Script e Intérpretes

Los lenguajes de programación script son un conjunto de expresiones y sentencias que son analizadas por un interprete para realizar diferentes funciones que van desde mostrar texto en un navegador Web hasta la consulta e inserción de información en una base de datos.

Entre los más populares para consultar e insertar información en una base de datos están: ASP, PHP, Perl y ColdFusión. La característica fundamental de todos ellos es la capacidad de embeberse en las páginas HTML.<sup>3</sup>

Los *interpretes* de los lenguajes script son un módulo que funciona en conjunto con el servidor Web, es decir, se trata de un programa encargado de interpretar las peticiones realizadas al servidor por un cliente mediante una página HTML que tienen incrustado el lenguaje script, una vez que interpreta el script donde le solicitan consultas e inserciones de información de la base de datos, lo que hace es traducir éstas consultas para inserción y recuperación de datos que envía al manejador de la base, quien le devolverá la información solicitada misma que será convertida por el interprete en una página HTML que enviará al cliente a través del servidor como respuesta a su petición.

## 1.6 Introducción a PHP

PHP es un lenguaje de programación el cual se ejecuta en los servidores web y que te permite crear contenido dinámico en tus páginas HTML. Es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como C o C++, es rápido a pesar de ser interpretado, multiplataforma y dispone de una gran cantidad de librerías que facilitan muchísimo el desarrollo de las aplicaciones. El significado de sus siglas es *HyperText Preprocessor*. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos.

Al ser un lenguaje de script, se programa mediante simples sentencias o frases (no compiladas), que son interpretadas por el motor PHP instalado en el servidor, devolviendo al visitante de la página el texto resultante. Esto quiere decir que en el código fuente de la página final solo verás html, nada de php.

---

<sup>3</sup> HTML, acrónimo de HyperText Markup Language, lenguaje de marcas de hipertexto. En informática, formato estándar de los documentos que circulan en lo World Wide Web WWW

Es multiplataforma, en el sentido de que PHP puede ejecutarse en entorno Unix y Windows, y es independiente del navegador que utilices, en el sentido de que el código php se ejecuta en el servidor, que devuelve al navegador solo texto

El funcionamiento de las páginas en **PHP** alojadas en un servidor es el siguiente:

- El navegador del cliente solicita el documento PHP.
- Llega la solicitud del servidor y el servidor localiza el documento, lanza el intérprete de PHP y ejecuta todo su código.
- Una vez ejecutado el código se genera el resultado en HTML y lo devuelve al servidor para que lo transfiera al cliente.
- El servidor transfiere el resultado en HTML y es mostrado en el navegador del cliente.

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML (Wap).

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP. A continuación en la siguiente figura se muestra el funcionamiento de PHP:



PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI, puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto.

PHP ha sido diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI, Perl o C y con la correcta selección de las opciones de configuración de tiempo de compilación y ejecución se consigue la exacta combinación de libertad y seguridad que se necesita. Ya que existen diferentes modos de utilizar PHP, existe también una multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes aplicaciones, pero también significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.

### **1.6.1 Características de PHP**

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

## 1.7 Bases de Datos

Las bases de Datos son cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar.

Los elementos principales que conforman una Base de Datos son las entidades y sus relaciones entre sí que pueden ser de distinta forma. Las entidades están expresadas en tablas las cuales están compuestas en una serie de campos que forman registros.

Para ejemplificar lo anterior podemos mencionar una tabla (entidad) llamada colonia, la cual está compuesta de los campos: nombre de la colonia y código postal, un registro será un renglón que contenga los datos de los 2 campos anteriores.

Un ejemplo de las relaciones sería que podemos tener una tabla llamada calle, la cual tendrá los campos: nombre de la calle y nombre de la colonia a la que pertenece, la relación será de uno a muchos, es decir, una colonia puede tener muchas calles.

### 1.7.1 Modelo Entidad-Relación (E/R)

El Modelo Entidad-Relación es un concepto de modelado para bases de datos, mediante el cual se pretende visualizar los objetos que pertenecen a la Base de Datos como entidades las cuales tienen unos atributos y se vinculan mediante relaciones.

Es una representación conceptual de la información. Mediante una serie de procedimientos se puede pasar del modelo E-R a otros, como por ejemplo el modelo relacional.

El modelado entidad-relación es una *técnica* para el modelado de datos utilizando diagramas entidad relación. No es la única técnica pero sí la más utilizada. Brevemente consiste en los siguientes pasos:

1. Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
2. Se hace una lista de los sustantivos y verbos que aparecen.
3. Los sustantivos son posibles entidades o atributos.
4. Los verbos son posibles relaciones.
5. Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
6. Se elabora el diagrama (o diagramas) entidad-relación.
7. Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

El modelado de datos no acaba con el uso de esta técnica. Son necesarias otras técnicas para lograr un modelo directamente implementable en una base de datos. Como por ejemplo:

- Transformación de relaciones múltiples en binarias.

- Normalización de una base de datos de relaciones (algunas relaciones pueden transformarse en atributos y viceversa).
- Conversión en tablas (en caso de utilizar una base de datos relacional).

### **Entidades**

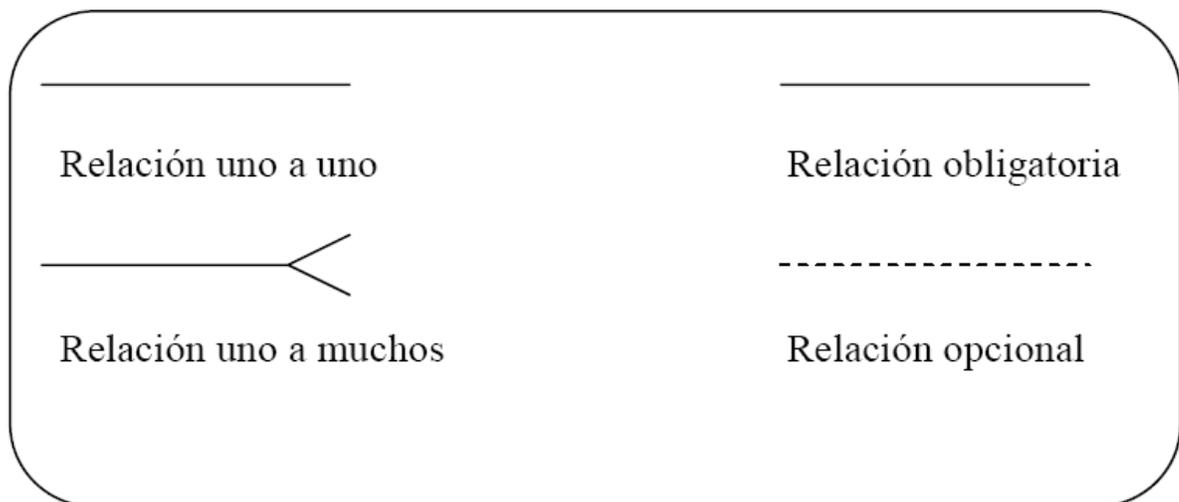
El término entidad, cuando se le refiere en base de datos, es cualquier objeto sobre el que se tiene información. Una entidad está descrita por sus características. Por ejemplo, la entidad **Persona** lleva consigo las características de: Nombre, Apellido, Género, Estatura, Peso, Fecha de nacimiento, etc. Se representa mediante un rectángulo o "caja" etiquetada en su interior mediante un **identificador**. Ejemplos de entidades habituales en los sistemas de información son: factura, persona, empleado, etc.

### **Relaciones**

Una relación describe cierta dependencia entre entidades. Se representa mediante un rombo etiquetado en su interior con un **verbo**. Este rombo se debe unir mediante líneas con las entidades (rectángulos) que relaciona.

Una relación tiene sentido al expresar las entidades que relaciona. Por ejemplo: una persona (entidad) trabaja para (relación) un departamento (entidad).

Tipos de Relaciones:



## **Atributos**

Los atributos son propiedades relevantes propias de una entidad y/o relación. Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta.

Por motivos de legibilidad, los atributos no suelen representarse en un diagrama entidad-relación, sino que se describen textualmente en otros documentos adjuntos.

Los atributos describen información útil sobre las entidades. En particular, los **atributos identificativos** son aquellos que permiten diferenciar a una instancia de la entidad de otra distinta. Por ejemplo, el atributo identificativo que distingue a un empleado de otro es su número de la Seguridad Social.

## **Herencia**

La herencia es un intento de adaptación de estos diagramas al paradigma orientado a objetos. La herencia es un tipo de relación entre una entidad "padre" y una entidad "hijo". La entidad "hijo" hereda todos los atributos y relaciones de la entidad "padre". Por tanto, no necesitan ser representadas dos veces en el diagrama. La relación de herencia se representa mediante un triángulo interconectado por líneas a las entidades. La entidad conectada por el vértice superior del triángulo es la entidad "padre". Solamente puede existir una entidad "padre" (herencia simple). Las entidades "hijo" se conectan por la base del triángulo.

### **1.7.2 Manejador de Bases de Datos**

Un manejador de Base de Datos es un programa que sirve para controlar el almacenamiento, recuperación y manejo de la información, dicha información está contenida en la Base de Datos.

Entre los más populares manejadores de Bases de Datos tenemos: MS-SQL Server, Oracle, Informix, DB2, PostgreSQL y MySQL, es importante, mencionar que la mayoría de ellos tienen una arquitectura típica como la que se enlista abajo

*Interfaz de Aplicación.* Son librerías para comunicarse con la base de datos. Muchos de los Manejadores de Bases de Datos tienen una simple línea de comandos.

*Interprete de SQL.* Revisa la sintaxis de las consultas realizadas y traslada a un lenguaje interno para su ejecución dentro del Manejador.

*Evaluador de las consultas.* Genera diferentes evaluaciones de las consultas y considerando las características y propiedades de la base de datos traslada las consultas a acciones de bajo nivel para su ejecución.

*Acceso a Datos.* Es un módulo que maneja los accesos para el almacenamiento de la información en el disco de la computadora, incluye los manejadores de transacciones, recuperación, seguridad y archivos.

*Base de Datos.* Son los datos físicos que se almacenan en archivos y que contienen archivos de ordenamiento para un acceso rápido a la información.

Con la aparición de los primeros manejadores de bases de datos surgió la necesidad de almacenar, organizar y consultar la información de cada una de las bases de datos creadas por distintos productos, para ello se utilizaban lenguajes propios de cada manejador (lenguaje de gestión con la base de Datos), pero esto representaba un problema, ya que para gestionar cada una de las bases de datos se requerían especialistas en cada producto.

En Junio de 1970 el Dr. E. F. Cood, investigador de IBM publico un artículo titulado “Un Modelo Relacional de datos para grandes bancos de datos compartidos” que perfilaba una teoría matemática de cómo almacenar y manipular datos utilizando una estructura tabular. Las bases de datos y SQL tienen sus orígenes en este artículo.

SQL es un lenguaje para organizar, gestionar y recuperar datos almacenados en una base de Datos. SQL es una abreviación de Structured Query Lenguaje (Lenguaje Estructurado de Consultas). SQL funciona con un tipo específico de bases de datos, llamado base de datos relacional.

La importancia de SQL radica básicamente en que después de su aparición en los años 80 se crearon 3 estándares publicados por la American National Standards Institute (ANSI) y la International Standards Organization (ISO); por ello hoy en día la mayoría de los Manejadores de Bases de Datos utilizan SQL como lenguaje de gestión con la base de datos, entre ellos: SQL Server, Oracle, SyBase, Informix, DB2, PostgreSQL y MySQL.

Básicamente SQL esta compuesto de 2 partes:

#### *Data Definition Lenguaje (Lenguaje de Definición de Datos)*

Es un conjunto de comandos SQL para definir y administrar objetos de bases de datos tales como de datos, tablas, vistas, índices, etc.

- CREATE
- ALTER
- DROP

#### *Data Manipulation Lenguaje (Lenguaje de Manipulación de Datos)*

Es un conjunto de comandos SQL para manipular los datos contenidos en los objetos base de datos.

- SELECT
- INSERT
- UPDATE
- DELETE

## 1.8 Aplicaciones de Bases de Datos en Web

En el nivel más básico, las aplicaciones Web trabajan en una arquitectura Cliente/Servidor, simplemente se necesita de un servidor central y una aplicación cliente, a diferencia de las aplicaciones tradicionales que no requieren de un servidor para su funcionamiento como sería el caso de las primeras bases de datos como FoxPro, Dbase, Clipper o incluso Access.

La arquitectura de las aplicaciones Web funciona de la forma que se observa en la figura de arriba, un cliente que realiza peticiones al servidor quien las recibe y procesa para posteriormente enviar las respuestas al cliente.

Ahora bien esta arquitectura esta formada por 3 capas: la capa de cliente, la capa de aplicación y la capa de datos.

Una de las ventajas que ofrece esta arquitectura, es el hecho de que los clientes son ligeros y no dependen del sistema operativo, solo basta tener un navegador de Internet el cual se comunica al servidor Web mediante el protocolo http, enviando al servidor una petición a través de una pagina HTML y recibiendo como respuesta un pagina igual.

## **CAPÍTULO 2**

# **ANÁLISIS DEL SISTEMA**

## 2.1 Definición y planteamiento del problema

Hoy en día la gente contrata grupos musicales para sus diversos eventos sociales. Por lo regular es raro no ver en alguna fiesta o evento social música viva. Suele suceder que en ocasiones las personas interesadas en contratar algún grupo musical para sus diversos eventos sociales, no tienen idea de como es la organización de un grupo musical, haciendo referencia al número de integrantes, instrumentos musicales, equipo de sonido, repertorio, etc., y mucho menos tienen idea del costo de algún grupo de estos. Cabe mencionar que cada agrupación musical tiene su forma de trabajar en lo que a costos se refiere, de igual manera lleva una agenda para su organización de eventos, todo esto es indispensable que la persona interesada en contratar el servicio lo sepa, ya que de toda esta información va depender si se realiza la contratación.

Consciente de todas esas exigencias e inquietudes surge la necesidad de desarrollar un sistema que sirva de apoyo a todo el público en general que esté interesado en la contratación de algún grupo musical para cualquier evento social, el sistema contará con toda la información necesaria de cada grupo de distinto género dado de alta en el sistema.

## 2.2 Entrevista con el usuario

Es indispensable y de vital importancia que ya después de haberse definido y planteado el problema, realizar una entrevista con el que será el dueño del sistema, el usuario, esto con el fin de establecer una comunicación entre los involucrados (usuario, analista, diseñador, arquitecto, desarrollador, líder de proyecto, etc.). Esta entrevista es la base para el análisis y diseño del sistema, ya que aquí se tocarán varios puntos, entre los más importantes tenemos los siguientes:

- Requerimientos Funcionales
- Requerimientos No Funcionales

Es muy importante crear un ambiente de confianza entre todos los involucrados, ya que es indispensable para poder dar diferentes puntos de vista y determinar que es lo mejor para que el sistema a desarrollar tenga tanto una apariencia como una eficiencia correcta. En ocasiones una entrevista no es suficiente para determinar y aclarar los puntos principales y secundarios por lo que será necesario citar a otras entrevistas posteriores.

Al final de la entrevista o entrevistas, todos los involucrados estarán conscientes del sistema que se necesitará analizar y desarrollar, de acuerdo a los puntos que se aclararon en las entrevistas.

## 2.3 Requerimientos Funcionales

El sistema que se desea realizar es para apoyar a todo el público en general interesados en la contratación de algún grupo musical de distinto género musical para sus eventos sociales.

Al sistema podrán acceder usuarios (todo el público en general) interesados en el tema sin ninguna clave personal. El usuario podrá consultar información de todos los grupos que le interesen para así poder tomar la mejor decisión para su contratación de acuerdo a sus gustos musicales y necesidades.

El sistema contará con los siguientes géneros musicales: Norteño, Rock, Versátil, Duranguense, Pop, Banda, Mariachi y Salsa.

El administrador del sistema se hará cargo de gestionar a los grupos musicales, es decir, darlos de alta, de baja y actualizar su información si alguno lo solicita, para esto el Administrador deberá contar con una clave de acceso al Sistema.

El gerente del sistema se hará responsable de todas las contrataciones que los usuarios vayan solicitando.

El sistema apoyará al representante de cada grupo musical para administrar su agenda de eventos. El representante podrá acceder al sistema como cualquier otro usuario para poder consultar y corroborar que los datos de su grupo estén correctos, consultar dudas o comentarios sobre el mismo y por supuesto consultar las fechas de eventos de su grupo musical.

Después de que el usuario haya consultado toda la información necesaria y los detalles del grupo a contratar de acuerdo a sus necesidades y gustos musicales, ésta se llevará a cabo de la siguiente manera:

El Sistema contará con varias formas para que el usuario pueda tener contacto con el Gerente de la Agencia, y así llevar a cabo dicha contratación. El Sistema contará en la opción de Contrataciones con un formulario, en el cual se deberán llenar todos los datos necesarios y ser enviado, posteriormente el Gerente o personal de la Agencia se pondrán en contacto con la persona interesada. Por otro lado, existirían los números telefónicos a donde los usuarios se podrán comunicar para conocer el proceso de la contratación. Así mismo, el sistema contará con el domicilio de la Agencia, por si el usuario desea realizar la Contratación personalmente. En todas estas opciones el usuario deberá proporcionar todos los datos necesarios para ser capturados, dar seguimiento a la contratación y llevar un control sobre los detalles de las contrataciones de los grupos.

En caso de que el usuario decida llenar el formulario, los campos serían los siguientes:

- Nombre
- Apellido Paterno
- Apellido Materno
- Domicilio
- Teléfono particular
- Celular
- E-mail
- Fecha del evento
- Lugar del evento
- Hora del evento
- Localidad/provincia del evento
- No. de asistentes (aprox.)
- Grupo a contratar
- Notas

Vía telefónica. El usuario hablará directamente con el Gerente o personal de la Agencia, para cotizar costos y todos los detalles del grupo interesado a contratar, el gerente pedirá todos los datos que se solicitan en el formulario, los cuales serán capturados en forma de solicitud de contratación.

La otra vía de contratación es que el usuario vaya personalmente a la Agencia a realizar su contratación, aquí se realizará una entrevista con el Gerente en dónde se aplicará el mismo procedimiento que en el medio anterior.

Para todos los posibles medios de contratación se deberá otorgar por parte del usuario el 50% de anticipo del costo total del evento. Este proceso se realizará ya sea en forma personal o por medio de un depósito a un número de cuenta que se proporcionará por vía telefónica o personalmente según haya sido la forma de contratar. El otro 50% del costo total se deberá liquidar al inicio o término del evento según se haya acordado.

Una vez que dicha contratación ya se llevó a cabo con el usuario, el Gerente llamará por teléfono al Representante del grupo contratado para comentarle todo lo sucedido. El Representante si lo desea podrá pedir un anticipo al Gerente sobre dicho evento o simplemente una vez cumplido dicho evento podrá cobrar su sueldo al Gerente.

El Gerente proporcionará al administrador del Sistema las nuevas fechas de eventos para los grupos contratados para que se encargue de actualizarlas en el sistema.

El sistema tendrá como primera pantalla una Bienvenida con una música de fondo.

El color del fondo será azul con algunos toques de color amarillo y negro. Las letras serán de color blanco y negro.

Se contará con una opción de "Saltar Introducción" para el caso de los usuarios que deseen pasar inmediatamente a la siguiente ventana. Para los usuarios que deseen escuchar

y observar toda la Introducción se contará con la opción de “Entrar” para acceder a la siguiente ventana.

En la segunda ventana se contarán con varias secciones, una será de Presentación, otra será un Buscador para que el usuario pueda consultar la información de su importancia. Se contará con una sección llamada Categorías en donde aparecerán todos los géneros musicales con los que cuenta la agencia, e información necesaria para el usuario. Así mismo se contará con la sección de Contrataciones, Comentarios, etc.

En el sistema existirán vínculos, los cuáles los llevarán a otras páginas de interés personal.

Los Grupos Musicales deben estar actualizados en el sistema.

El desarrollo se realizará sobre una plataforma PHP y SQL Server, utilizando algunas imágenes diseñadas con Flash.

## **2.4 Requerimientos No Funcionales**

Es muy importante para el análisis y desarrollo de Sistemas Web los recursos que se van a utilizar, tanto el hardware como el software, además de otros elementos como lo son seguridad, velocidad de acceso y servicios. A continuación se mencionarán los Requerimientos No Funcionales más importantes que se necesitan determinar para la elaboración del sistema Web.

### **2.4.1 Recursos de Hardware**

#### **Servidores**

Son las máquinas donde será depositada la información que deseamos publicar en Internet. Esta información puede estar distribuida en varios ordenadores o en un ordenador más grande.

#### **Memoria RAM**

La mayoría de páginas que hay en Internet son HTML, por lo que una vez que se haya publicado una de ellas, esta pasará a la RAM de nuestro servidor, por lo que si un usuario viene inmediatamente después, se encontrará con la página de una manera mucho más rápida que el usuario anterior. Una buena cantidad de RAM siempre hará que nuestro servidor no tenga que hacer tantos accesos a disco duro, por lo que la velocidad de servicio aumentará. El mínimo que se considera de memoria RAM para nuestro Servidor es de 4 Gb.

Hay que tener en cuenta que si se trabaja con páginas que se generan de forma aleatoria, que se generan con una consulta a la base de datos o simplemente a gusto del

usuario, diremos que la RAM no juega un factor tan importante ya que es muy difícil que una misma sea cargada de una forma continuada.

## **Disco Duro**

Debido a toda la información que reside en disco duro, se debe tener una velocidad de acceso a éste lo más elevada posible. Con un par de imágenes, un fondo que no sea liso y un texto más o menos grande nos vamos a un tamaño de una página de unos 25-35Kb. Si tenemos en cuenta que las imágenes normalmente se encuentran en directorios distintos a los documentos HTML, y que por lo general los discos duros se encuentran llenos, se recomiendan el uso de discos duros SCSI o SCSI II ya que tienen una velocidad de respuesta mayor a la de los discos duro normales (IDE).

Debido a que en todas las plataformas los navegadores Web requieren mucho espacio de disco libre el mínimo requerido está entre 50 y 100Mb libres.

## **Procesador**

Con las páginas HTML, no se gana mucho con la velocidad del procesador, aunque también es un factor importante. Pero si la página Web que nosotros generamos se hace a partir de una consulta a una base de datos o, a parte de la página Web, se generan archivos de consulta para guardar o imprimir, esto se realizará básicamente con el procesador, y cuanto más potencia se tenga (velocidad de proceso) estas se generarán de una manera más rápida.

Para las PC, se desea al menos un Pentium 100MHz o un Pentium 133MHz con 1 Gb de memoria.

A parte de los aspectos físicos del servidor, hay que tener en cuenta quien los gestiona. Esto es el sistema operativo.

## **Tipos de Línea**

Actualmente la forma más extendida de acceder a un servidor es por medio de una Frame Relay, que es una técnica de conmutación por paquetes que nos servirá para enviar información. Esta técnica básicamente lo que hace es transferir a los terminales las funciones de flujo de control, corrección de errores y otras que anteriormente hacían los protocolos de comunicación como X.25. Frame Relay esta orientado a paquetes como X.25 y también transparente a los protocolos como el modo circuitos.

## **Router**

Estos dispositivos operan en el nivel de red y permiten la interoperatividad entre redes diferentes eligiendo siempre el mejor camino para enviar un paquete.

## **2.4.2 Recursos de Software**

### **Servidores (Software)**

El software para los Servidores se refiere a los programas que se necesitan para que el usuario se beneficie de los servicios que se ofrecen. Como por ejemplo:

#### **DNS (Domain Name Server)**

Conocido como servidor de dominio. Este servidor lo que hace es traducir la dirección que se envía con formato texto, en una dirección IP. Si reconoce esa dirección, dentro del dominio, la sirve; si no, reenvía la dirección a un servidor inmediato superior que si tampoco la reconoce, actuará de la misma manera.

EL DNS se puede entender como un servicio de directorios de Internet que proporciona información acerca de la asociación entre el nombre simbólico de un nodo con su dirección IP. Otra de sus particularidades es que divide la red en dominios y subdominios a partir de una estructura arbórea que identifica por caminos, por lo que una zona será una región compuesta de uno o más dominios y subdominios gestionada por un único administrador. Cada nodo se identifica por una etiqueta o alias, de longitud inferior a 63 caracteres, y cada nombre de dominio de un nodo del servicio DNS está formado por una concatenación de etiquetas, cada una de ellas separada por un punto.

#### **FTP (File Transfer Protocol)**

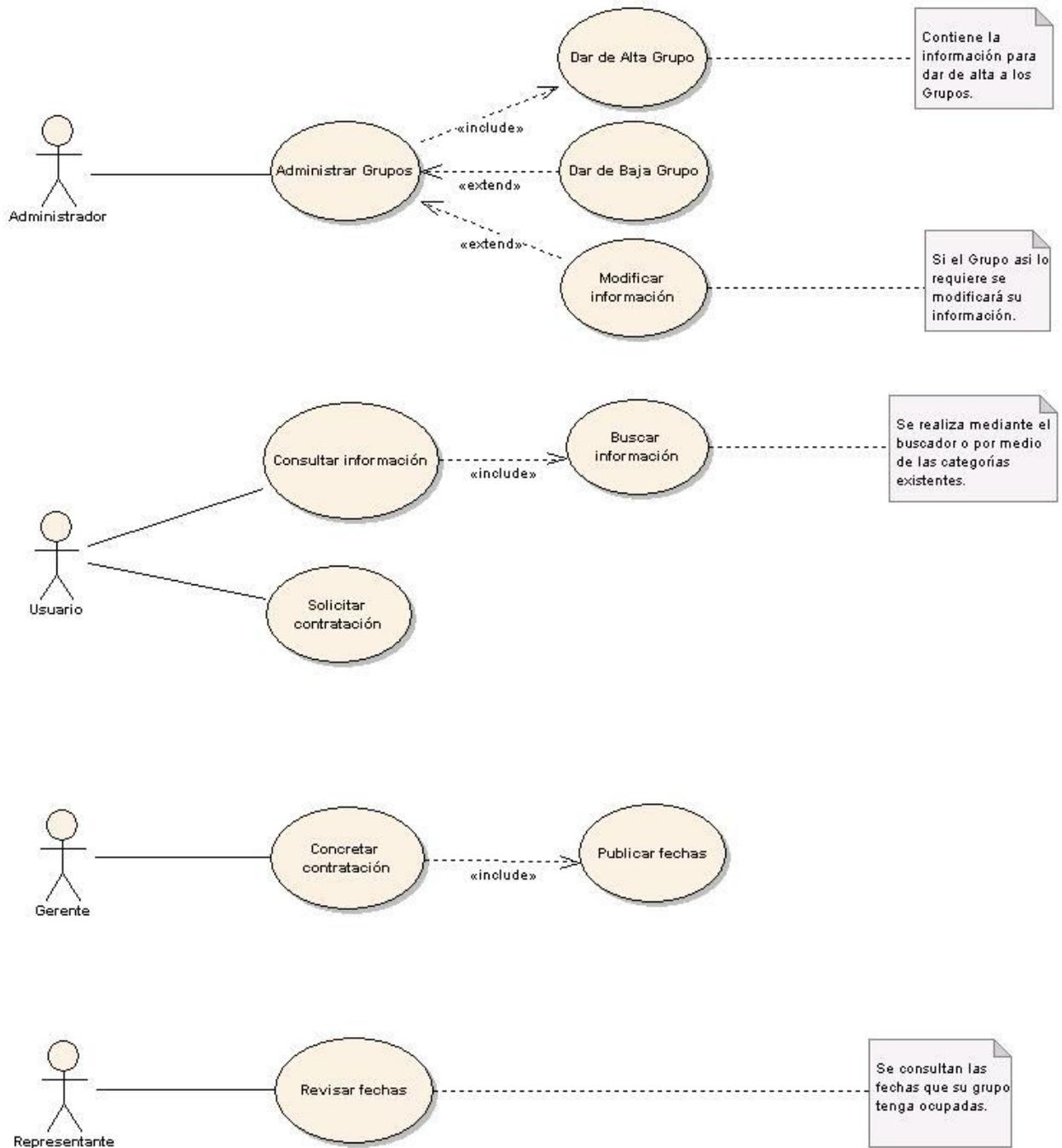
Se utiliza para poder intercambiar información en forma de archivos entre dos ordenadores. Generalmente, los usuarios utilizan el FTP para poder poner sus páginas Web en su servidor de una manera más clara, rápida y segura, ya que cada usuario normalmente tiene acceso a su directorio particular.

#### **Hospedaje Web o Hosting**

Se refiere al alojamiento de los archivos que se desea publicar en Internet, los cuales deberán estar almacenados necesariamente en un servidor con la configuración e implementación necesaria para su salida a la red con alta velocidad, además de contar con la instalación del software necesario para el correcto funcionamiento de las aplicaciones.

## 2.5 Utilizando UML para el Análisis del Sistema

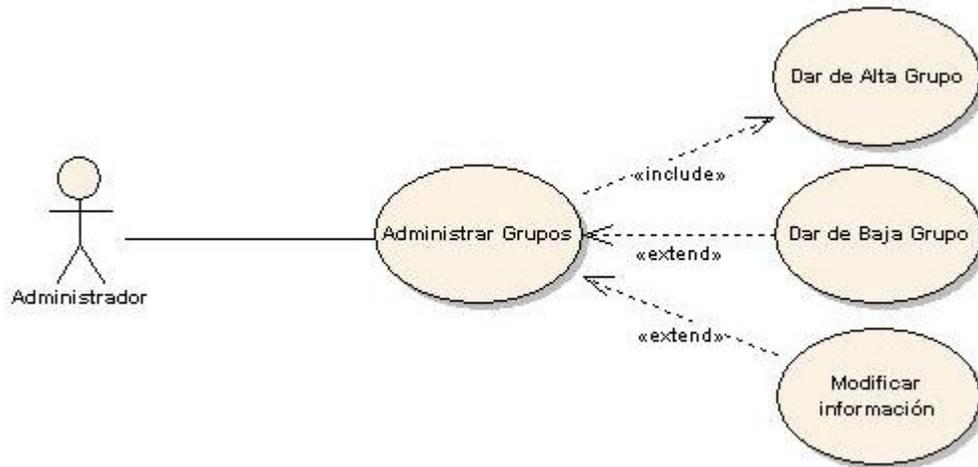
### 2.5.1 Diagrama de Casos de Uso General



### 2.5.2 Narrativas de Casos de Uso

**Caso de Uso:** 1. Administrar Grupos

**Actor:** Administrador



**Descripción:** El Administrador llevará el control de todos los Grupos Musicales, por medio de formatos que se tendrán que llenar.

**Precondiciones:**

- El Administrador debe haber ingresado al sistema.
- El Administrador deberá contar con toda la información necesaria del Grupo Musical.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El Administrador desea Administrar Grupos Musicales.	2	Muestra las opciones para el Administrador.	
3	El Administrador ingresa a la opción de Administrar Grupos.	4	Despliega una pantalla con las siguientes opciones: "Dar de Alta Grupo", "Dar de Baja Grupo" y "Modificar información".	E1

**Excepciones:**

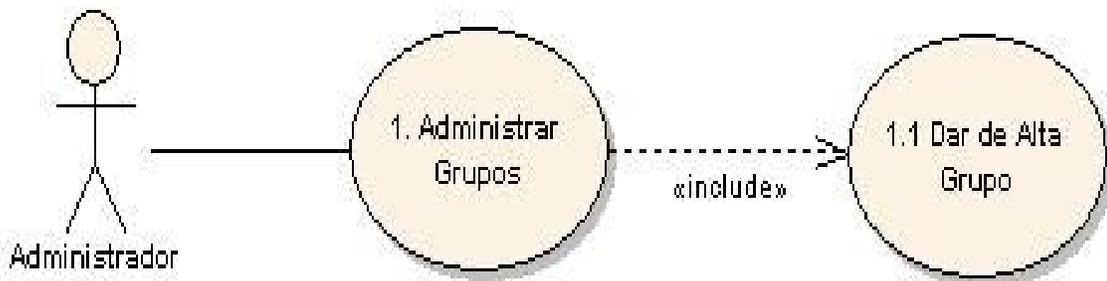
Id	Nombre	Acción
E1	Si la opción seleccionada es incorrecta.	El sistema permite nuevamente su captura.

**Post-condiciones:**

- El Administrador puede dar de alta a un grupo, darlo de baja y modificar la información de algún grupo.

**Caso de Uso:** 1.1 Dar de Alta Grupo

**Actor:** Administrador



**Descripción:** El Administrador puede dar de alta a todos los Grupos Musicales que lo soliciten, llenando el respectivo formato.

**Precondiciones:**

- El Administrador debe haber ingresado al sistema.
- El Administrador deberá contar con toda la información necesaria del Grupo Musical que va a dar de Alta.

**Flujo:**

<b>ACTOR</b>		<b>SISTEMA</b>		
<b>Paso</b>	<b>Acción</b>	<b>Paso</b>	<b>Acción</b>	<b>Excepción</b>
1	El Administrador desea Dar de Alta un Grupo Musical.	2	Muestra las opciones con las que cuenta el sistema para Administrar a los Grupos Musicales.	
3	El Administrador selecciona la opción de Dar de Alta Grupo.	4	Muestra el formato de Dar Alta Grupo para que sean llenados los campos requeridos.	E1
5	El Administrador ingresa toda la información del Grupo y la guarda.	6	Informa que los datos han sido guardados y que hay un nuevo grupo dado de alta en el sistema.	E2, E3

**Excepciones:**

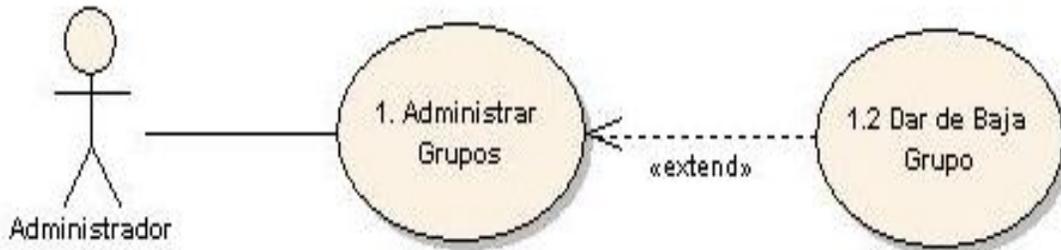
<b>Id</b>	<b>Nombre</b>	<b>Acción</b>
E1	Si la opción seleccionada es incorrecta.	El sistema permite nuevamente su captura.
E2	Si el nombre del Grupo ya existe.	El sistema informará de la situación y permitirá nuevamente la captura de otro nombre hasta que éste no esté dado de Alta en el sistema.
E3	Si falta algún dato en el formato.	El sistema informa de la situación y permite nuevamente su captura.

**Post-condiciones:**

- El Administrador ha dado de alta a un nuevo grupo musical.
- La agencia ahora cuenta con un grupo más para ofrecerle al público.

**Caso de Uso:** 1.2 Dar de Baja Grupo

**Actor:** Administrador



**Descripción:** El Administrador puede dar de baja a algún Grupo Musical que no esté cumpliendo con los estatutos de la Agencia.

**Precondiciones:**

- El Administrador debe haber ingresado al sistema.
- El Administrador conoce la situación del Grupo Musical que va a dar de Baja.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El Administrador desea Dar de Baja un Grupo Musical.	2	Muestra las opciones con las que cuenta el sistema para Administrar a los Grupos Musicales.	
3	El Administrador selecciona la opción de Dar de Baja Grupo.	4	Despliega la pantalla para Dar de Baja a un Grupo con el formato correspondiente.	E1
5	El Administrador ingresa la información para dar de baja al grupo y la guarda.	6	Pregunta, si ¿está seguro de querer dar de baja al grupo correspondiente?	E2, E3
7	El Administrador confirma su opción.	8	Informa que el grupo ha sido dado de baja y ya no está más en el sistema.	E4

**Excepciones:**

Id	Nombre	Acción
E1	Si la opción seleccionada es incorrecta.	El sistema permite nuevamente su captura.
E2	Si algún dato es incorrecto.	El sistema informará de la situación y permitirá nuevamente la captura de los datos.
E3	Si falta algún dato en el formato para darlo de baja.	El sistema informa de la situación y permite nuevamente su captura.
E4	Si el Administrador decide no quererlo dar de baja.	El sistema sale de la pantalla y regresa a la pantalla principal.

**Post-condiciones:**

- El Administrador ha dado de baja a un grupo musical.
- La agencia ahora cuenta con un grupo menos.

**Caso de Uso:** 1.3 Modificar información

**Actor:** Administrador



**Descripción:** El Administrador podrá modificar la información de algún Grupo Musical, si así lo requiere su Representante, para estar actualizado y el público pueda tomarlo en cuenta para posibles contrataciones.

**Precondiciones:**

- El Administrador debe haber ingresado al sistema.
- El Administrador debe conocer la información nueva del Grupo Musical que va a modificar.

**Flujo:**

<b>ACTOR</b>		<b>SISTEMA</b>		
<b>Paso</b>	<b>Acción</b>	<b>Paso</b>	<b>Acción</b>	<b>Excepción</b>
1	El Administrador desea Modificar información de un Grupo Musical.	2	Muestra las opciones con las que cuenta el sistema para Administrar a los Grupos Musicales.	
3	El Administrador selecciona la opción de Modificar información.	4	Muestra una pantalla en donde se deberá escribir el nombre del Grupo Musical a modificar.	E1
5	El Administrador ingresa el nombre del Grupo Musical que va a modificar.	6	Despliega toda la información del grupo por medio de campos, los cuáles podrá modificar el administrador.	E2, E3
7	El Administrador realiza todas las modificaciones con la nueva información y la guarda.	8	Pregunta, si ¿está seguro de querer modificar la información del grupo correspondiente?	E4
9	El Administrador confirma su opción.	10	Informa que la información del grupo ha sido modificada.	E5

**Excepciones:**

<b>Id</b>	<b>Nombre</b>	<b>Acción</b>
E1	Si la opción seleccionada es incorrecta.	El sistema permite nuevamente su captura.
E2	Si el nombre del Grupo no existe.	El sistema informará de la situación y permitirá nuevamente la captura.
E3	Si el nombre del Grupo es incorrecto.	El sistema informa y pide que se vuelva a capturar el nombre

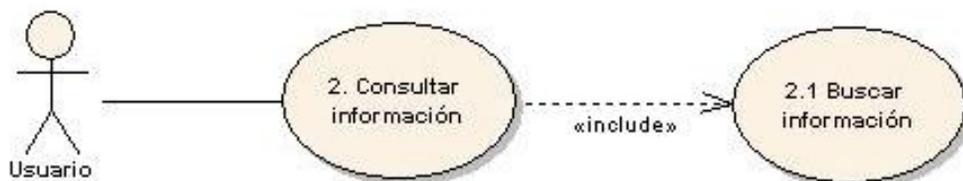
E4	Si algún dato es incorrecto.	El sistema informa de la situación y permite nuevamente su captura.
E5	Si el Administrador decide no modificar la información.	El sistema sale de la pantalla y regresa a la pantalla principal.

**Post-condiciones:**

- El Administrador ha modificado la información de un Grupo Musical.

**Caso de Uso: 2. Consultar información**

**Actor:** Usuario



**Descripción:** El usuario podrá estar consultando toda la información de todos los Grupos Musicales de distintos géneros que el requiera para estar seguro de su posible contratación.

**Precondiciones:**

- El usuario debe haber ingresado al sistema.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El usuario desea Consultar información de algún Grupo Musical.	2	Muestra la información del Grupo Musical correspondiente.	
3	El usuario realiza su consulta, analizando todas las características del Grupo Musical.			

**Post-condiciones:**

- El usuario ha consultado la información de uno o varios Grupos Musicales.

**Caso de Uso: 2.1 Buscar información**

**Actor:** Usuario



**Descripción:** El usuario puede realizar su búsqueda de información de los Grupos Musicales ya sea por medio del buscador o ingresando directamente a las distintas categorías musicales con las que contara el sistema.

**Precondiciones:**

- El usuario debe haber ingresado al sistema.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El usuario desea realizar una búsqueda de información de Grupos Musicales.	2	Muestra las opciones para realizar la búsqueda: "Buscador" o Ingresando directamente a las categorías musicales.	
3	El usuario selecciona alguna opción.	4	Despliega una pantalla con el formato de la opción seleccionada.	E1
5	El usuario realiza su búsqueda ingresando algún dato como: nombre de un Grupo, Género musical.	6	Despliega la información del dato que ingresó el usuario.	E2, E3

**Excepciones:**

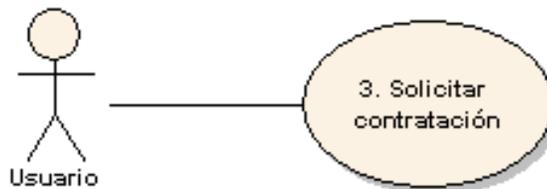
Id	Nombre	Acción
E1	Si la opción seleccionada es incorrecta.	El sistema permite nuevamente su captura.
E2	Si el dato ingresado es incorrecto.	El sistema informa de la situación y permite nuevamente su captura.
E3	Si el dato ingresado no está en la Base de Datos.	El sistema informa de la situación y permite nuevamente su captura.

**Post-condiciones:**

- El usuario ha realizado su búsqueda de información de uno o varios Grupos Musicales.
- El usuario podrá realizar su consulta, analizando las características del Grupo seleccionado.

**Caso de Uso: 3. Solicitar contratación**

**Actor:** Usuario



**Descripción:** El usuario puede solicitar su contratación, escribiéndole un e-mail al Gerente de la Agencia, para que este le explique el procedimiento de dicha contratación.

**Precondiciones:**

- El usuario debe haber ingresado al sistema.
- El usuario deba haber realizado una búsqueda de información.
- El usuario debe haber realizado una consulta de información de algún Grupo Musical para llevar a cabo su solicitud de contratación.

**Flujo:**

<b>ACTOR</b>		<b>SISTEMA</b>		
<b>Paso</b>	<b>Acción</b>	<b>Paso</b>	<b>Acción</b>	<b>Excepción</b>
1	El usuario desea solicitar una contratación de algún Grupo Musical.	2	Muestra la opción de "Solicitar contratación".	
3	El usuario selecciona la opción.	4	Despliega una pantalla con las instrucciones y el formato para escribir el mail al Gerente de la Agencia.	E1
5	El usuario lee las instrucciones y escribe el mail al Gerente.	6	Despliega una pantalla en donde dice que su mail ha sido enviado correctamente.	E2, E3

**Excepciones:**

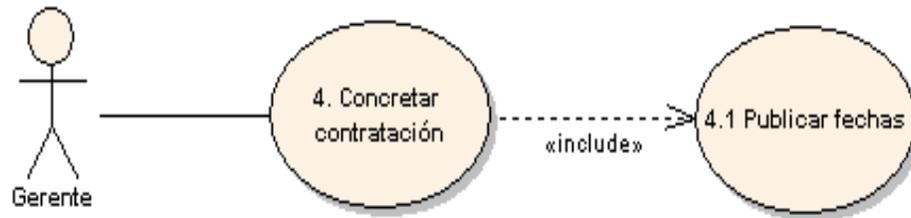
<b>Id</b>	<b>Nombre</b>	<b>Acción</b>
E1	Si la opción seleccionada es incorrecta.	El sistema permite nuevamente su captura.
E2	Si algún dato ingresado es incorrecto.	El sistema informa de la situación y permite nuevamente su captura.
E3	Si ocurre un problema a la hora de enviarse el mail.	El sistema informa de la situación y permite nuevamente que se envíe.

**Post-condiciones:**

- El usuario ha solicitado una contratación de algún posible Grupo Musical dado de alta en el sistema.
- El usuario espera la respuesta del Gerente.

**Caso de Uso:** 4. Concretar contratación

**Actor:** Gerente



**Descripción:** El Gerente de la Agencia, responderá al mail enviado por el usuario para que le explique como es que se lleva a cabo la contratación y concretarla.

**Precondiciones:**

- El Gerente debe haber ingresado al sistema.
- El Gerente debe haber recibido el e-mail del usuario.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El Gerente desea Concretar una contratación con el usuario.	2	Muestra el e-mail recibido del usuario, con todas las opciones del formato del e-mail.	
3	El Gerente lee y selecciona la opción de Responder.	4	Despliega una pantalla para escribir la respuesta al usuario.	E1
5	El Gerente redacta los 2 medios (Vía telefónica o entrevista), por los cuales se puede llevar a cabo la contratación y la envía.	6	Despliega una pantalla en donde dice que su mail ha sido enviado correctamente.	E2, E3

**Excepciones:**

Id	Nombre	Acción
E1	Si la opción seleccionada es incorrecta.	El sistema permite nuevamente su captura.

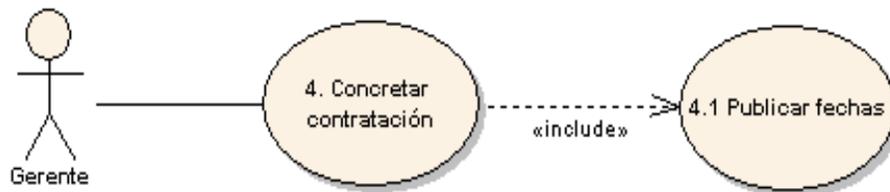
E2	Si algún dato ingresado es incorrecto.	El sistema informa de la situación y permite nuevamente su captura.
E3	Si ocurre un problema a la hora de enviarse el mail.	El sistema informa de la situación y permite nuevamente que se envíe.

**Post-condiciones:**

- El Gerente ha concretado una contratación con el usuario.
- El usuario decidirá algún medio por la cual se realice la contratación y se lleve a cabo todo el respectivo procedimiento.
- El Gerente informa de lo sucedido al Representante del Grupo Musical a contratar.

**Caso de uso: 4.1 Publicar fechas**

**Actor:** Gerente



**Descripción:** El Gerente publicará en el sistema la fecha en la que el Grupo haya sido contratado, para que los demás usuarios sepan que fechas ya tienen ocupadas los diferentes grupos dados de alta en el sistema y así puedan consultar otro Grupo.

**Precondiciones:**

- El Gerente debe haber realizado la contratación con el usuario.
- El Gerente debe haber ingresado al sistema.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El Gerente desea publicar una fecha que ya esté	2	Muestra las opciones con las que cuenta el sistema	

	ocupada de ese Grupo.		para Administrar a los Grupos Musicales.	
3	El Gerente selecciona la opción de Publicar fechas.	4	Muestra una pantalla en donde se deberá escribir el nombre del Grupo Musical.	E1
5	El Gerente ingresa el nombre del Grupo Musical.	6	Despliega las fechas que el Grupo ya tiene ocupadas y aparece la opción de "Publicar nueva fecha"	E2, E3
7	El Gerente elige la opción de Publicar nueva fecha.	8	Muestra una pantalla para que se agregue la nueva fecha a publicar, con el nombre del lugar del evento.	
9	El Gerente captura la nueva fecha del Grupo y el lugar del evento y la guarda.	10	Informa que la nueva fecha del grupo ha sido guardada y publicada.	E4

**Excepciones:**

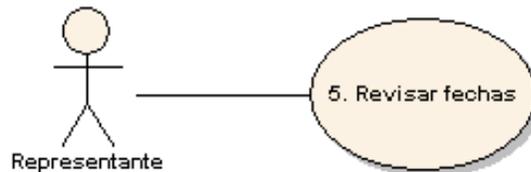
<b>Id</b>	<b>Nombre</b>	<b>Acción</b>
E1	Si la opción seleccionada es incorrecta.	El sistema permite nuevamente su captura.
E2	Si el nombre del Grupo no existe.	El sistema informará de la situación y permitirá nuevamente la captura.
E3	Si el nombre del Grupo es incorrecto.	El sistema informa y pide que se vuelva a capturar el nombre
E4	Si algún dato es incorrecto.	El sistema informa de la situación y permite nuevamente su captura.

**Post-condiciones:**

- El Gerente ha publicado una nueva fecha de un Grupo Musical.
- El Grupo ahora tiene un nuevo evento.

**Caso de uso:** 5. Revisar fechas

**Actor:** Representante



**Descripción:** El Representante puede revisar las fechas que su Grupo ya tiene ocupadas para corroborarlas.

**Precondiciones:**

- El Representante debe haber ingresado al sistema.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El Representante desea revisar las fechas que su Grupo ya tenga ocupadas.	2	Muestra la información del Grupo Musical con todas sus opciones.	
3	El Representante selecciona la opción de “ver fechas”.	4	Despliega una pantalla con todas las fechas que el Grupo ya tiene publicadas en el sistema.	
5	El Representante realiza su consulta y revisa que todas las fechas estén correctas.			

**Post-condiciones:**

- El Representante ha revisado las fechas que su Grupo ya tiene ocupadas.

## **CAPÍTULO 3**

# **DISEÑO DEL SISTEMA**

### 3.1 Introducción

En éste capítulo se tocarán puntos muy importantes en lo que se refiere al Diseño del Sistema Web. Este capítulo estará basado por Diagramas que serán de gran utilidad para poder entender el diseño y desarrollo del Sistema. Los diagramas son de distinto tipo y en primera instancia se deben de crear de acuerdo al análisis que ya se realizó en el capítulo anterior ya que estos son muy importantes para los desarrolladores, cabe mencionar que en futuro tal vez estos diagramas puedan cambiar de acuerdo a los nuevos requerimientos que vayan surgiendo para el Sistema.

### 3.2 Modelo Conceptual

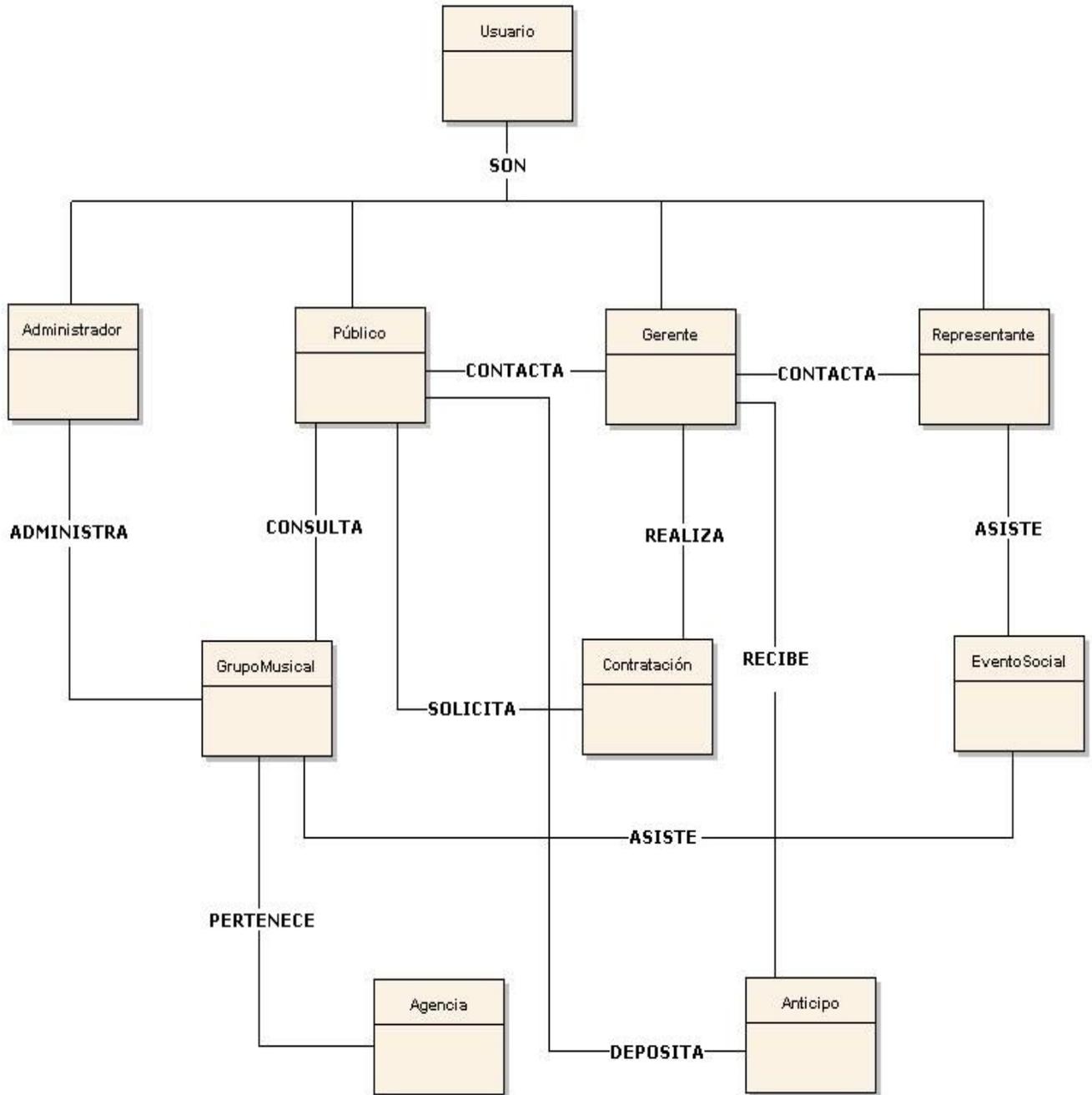
El diseño conceptual parte de las especificaciones de los requerimientos del usuario y su resultado es el esquema conceptual de la base de datos. Un *esquema conceptual* es una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD que se vaya a utilizar para manipularla. Un *modelo conceptual* es un lenguaje que se utiliza para describir esquemas conceptuales.

El objetivo del diseño conceptual es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información.

El diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. Un *esquema lógico* es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD. Un *modelo lógico* es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red, etc.). El diseño lógico depende del tipo de SGBD que se vaya a utilizar, no depende del producto concreto.

El diseño físico parte del esquema lógico y da como resultado un esquema físico. Un *esquema físico* es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos.

El siguiente Diagrama mostrará cuales son los elementos principales del Sistema Web y como se relacionan entre sí. La finalidad, es facilitar el conocimiento del vocabulario del dominio y de los conceptos que se incluyen en los requerimientos.



### 3.3 Diagrama de clases

#### 3.3.1 Definición de Clase

**Clase.** Es la unidad básica que encapsula información.

Un Diagrama de Clases realiza lo siguiente:

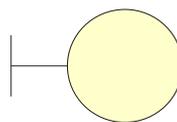
- Muestran el vocabulario del sistema.
- Generados durante el flujo de análisis y refinados durante el flujo de diseño.
- Propuestos de manera heurística con base en el modelo de casos de uso.

El modelo de análisis sugiere la siguiente agrupación de clases:

- Clases Boundary
- Clases Entity
- Clases de Control

#### 3.3.2 Clases Boundary

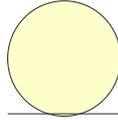
- Clases de interacción con el usuario (boundary). En un desarrollo de tipo WEB, cada HTML y JSP, representará una clase “**boundary**”, en donde los métodos estarán determinados por las ligas o botones incluidos.



Boundary

#### 3.3.2 Clases Entity

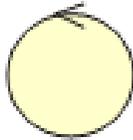
- Clases de persistencia (entity). Las clases de tipo “**entity**” son los candidatos a tener una larga vida durante la ejecución del sistema; es decir, serán clases cuyo estado debe ser almacenado en disco. Generalmente, son clases que son identificadas de manera inmediata, durante el flujo de análisis del desarrollo.



Entidad

### **3.3.3 Clases de Control**

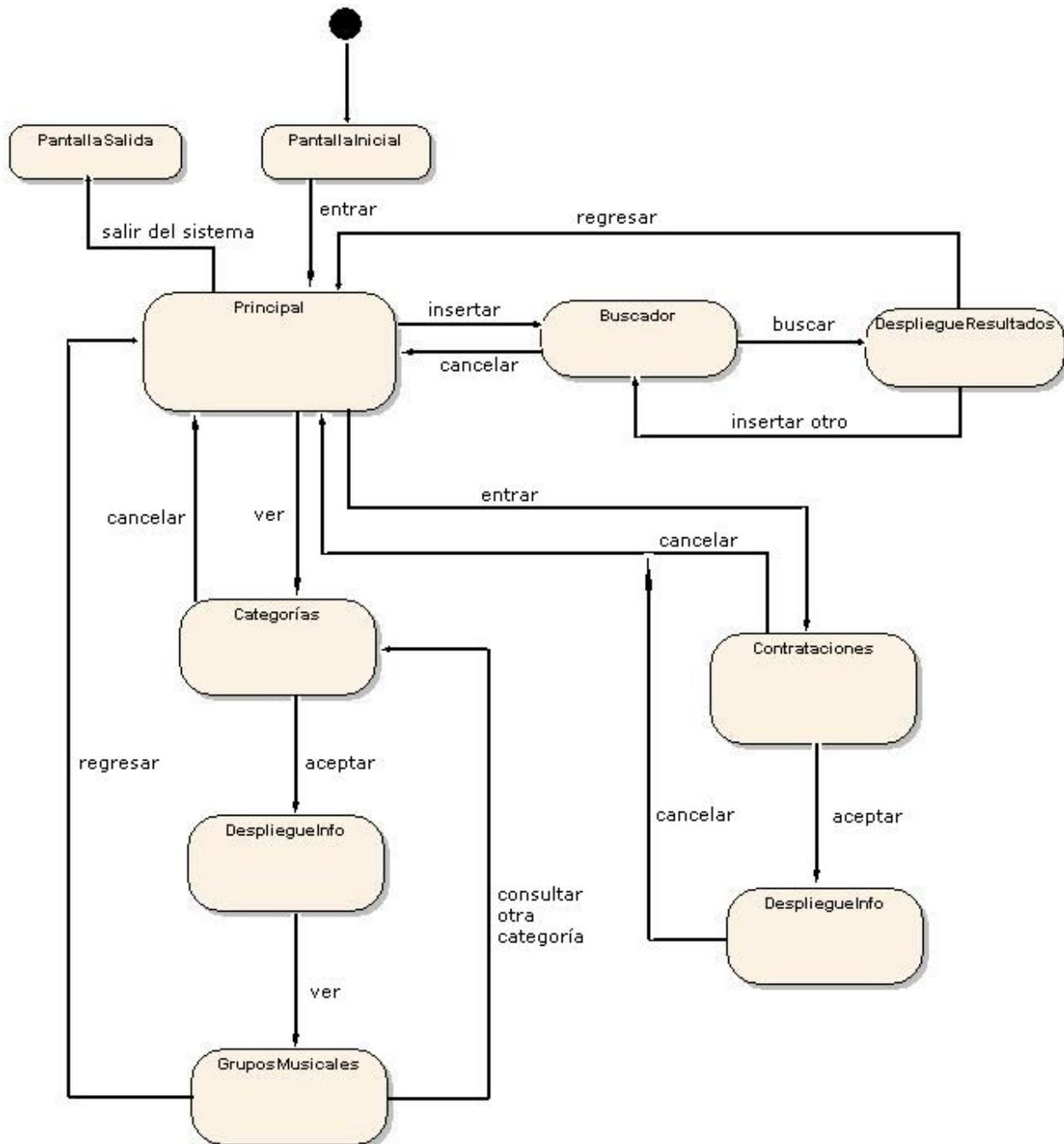
- Clases de regla de negocio (control). Las clases de tipo “**Regla de negocio**” o “**de control**” son clases que involucran la operación del resto del sistema. Están incluidas aquellas entidades que definen la lógica y aquellas que realizan cálculos u operaciones que involucran a las clases “Entity”.



Control

### 3.4 Diagramas de Estados

Diagrama de estados para la navegación del Usuario



### Diagrama de estados para la navegación del Administrador

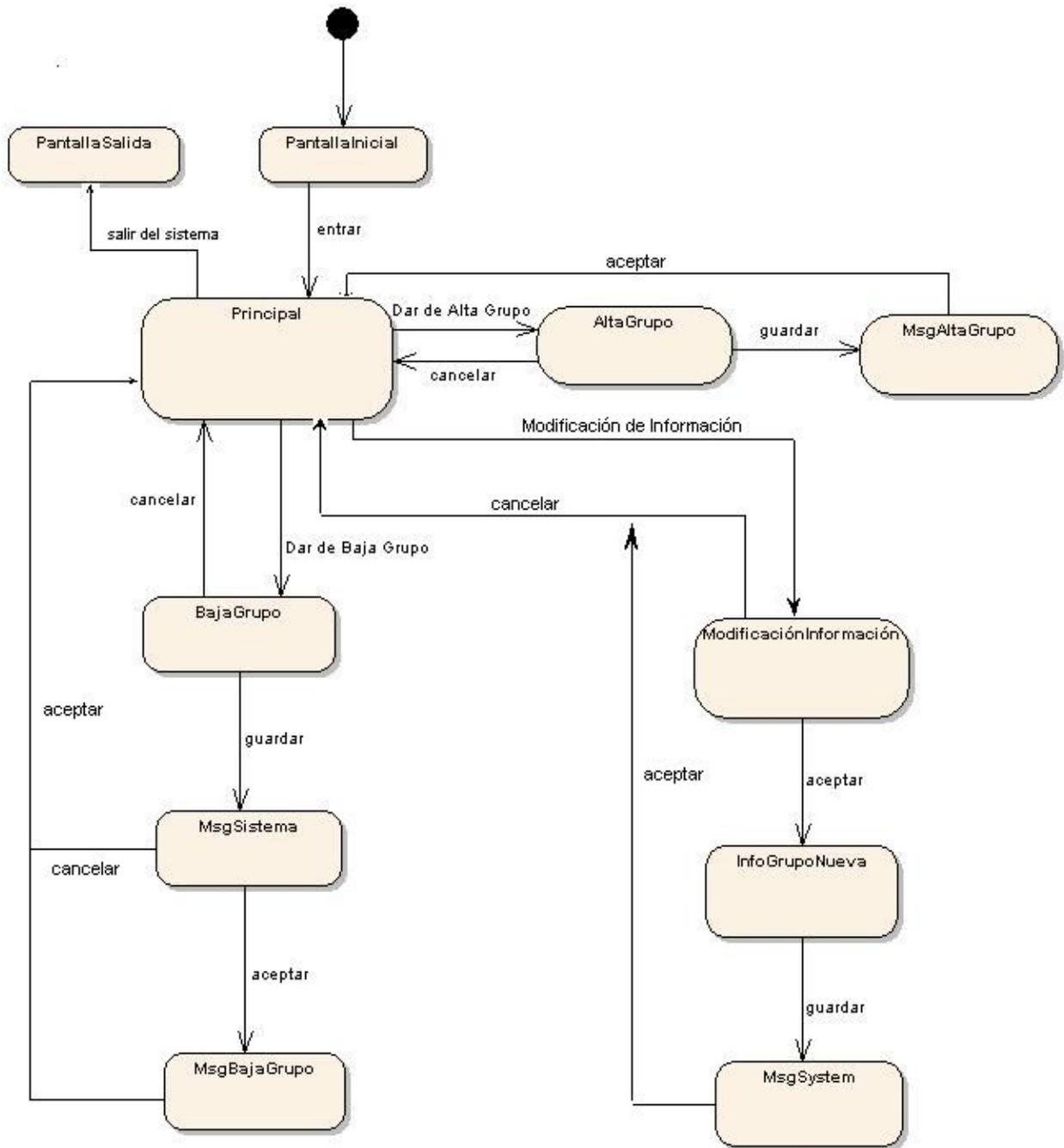


Diagrama de estados para la navegación del Gerente

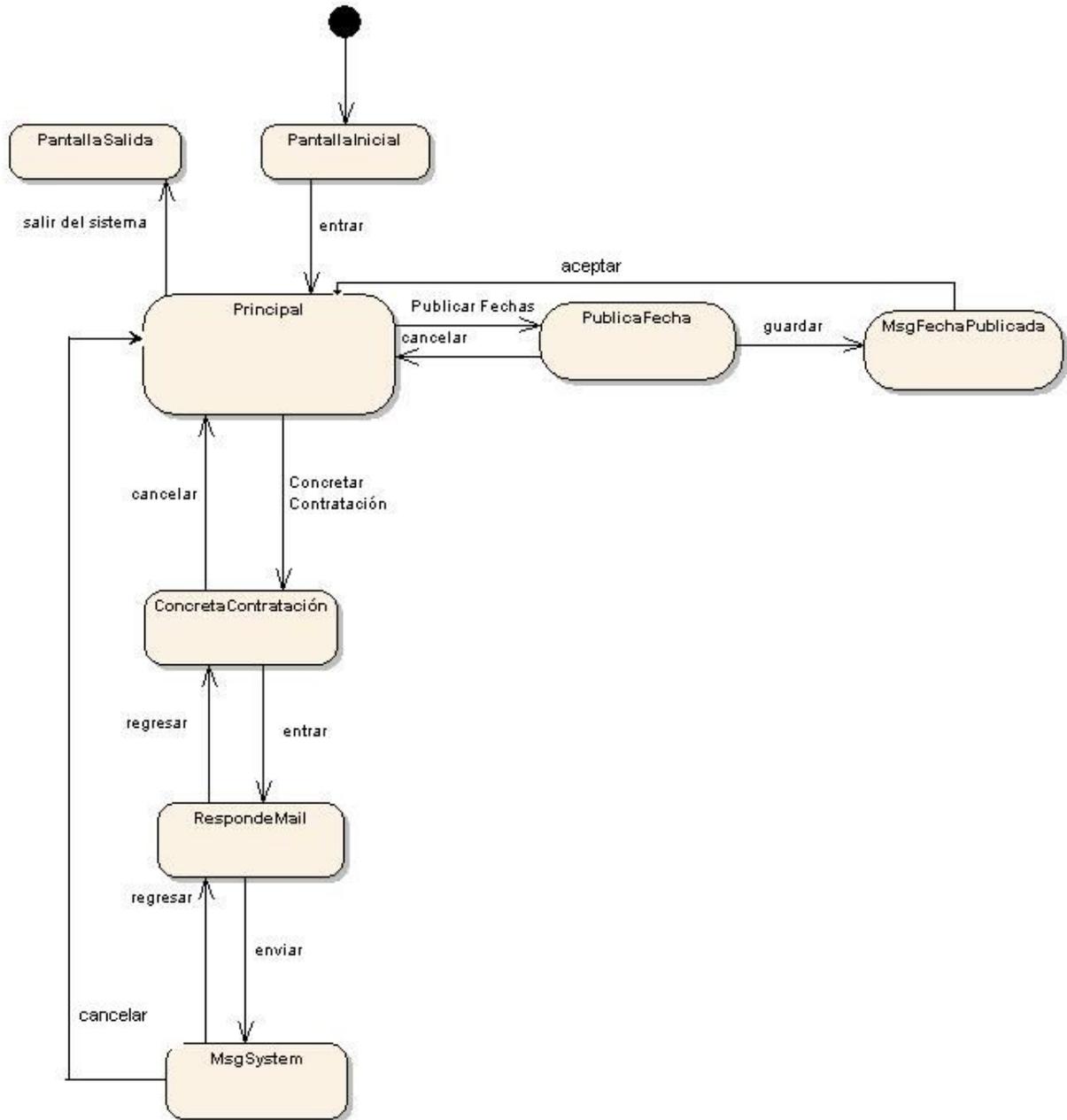
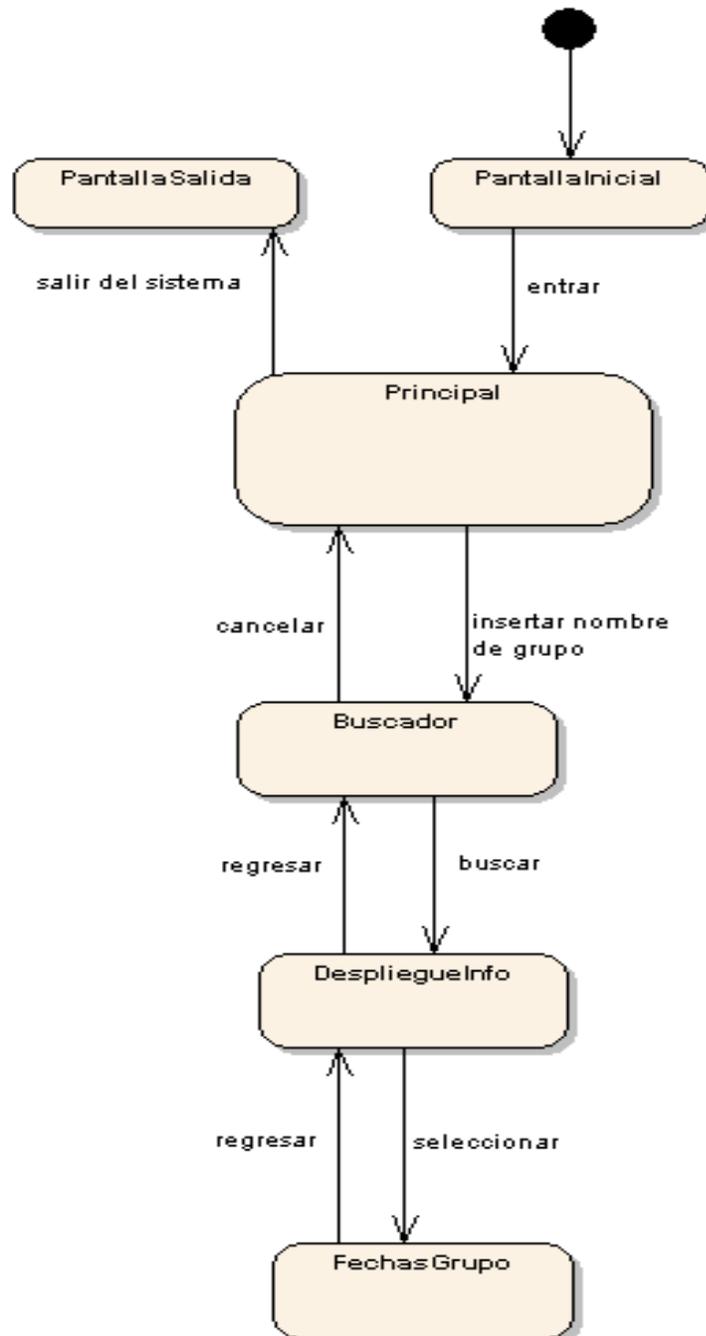
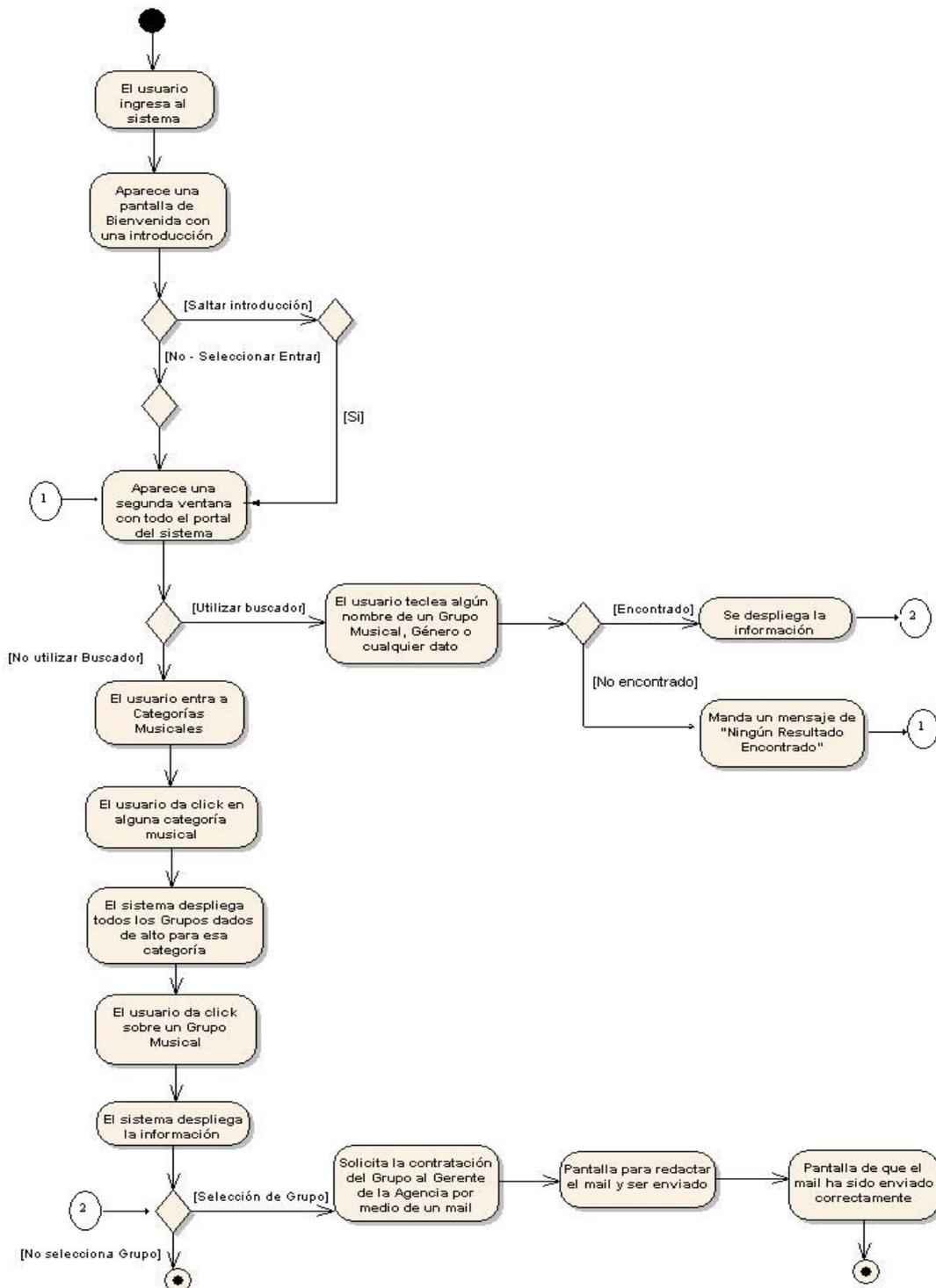


Diagrama de estados para la navegación del Representante



### 3.5 Diagrama de Actividades

Diagrama de actividades para el Usuario Final



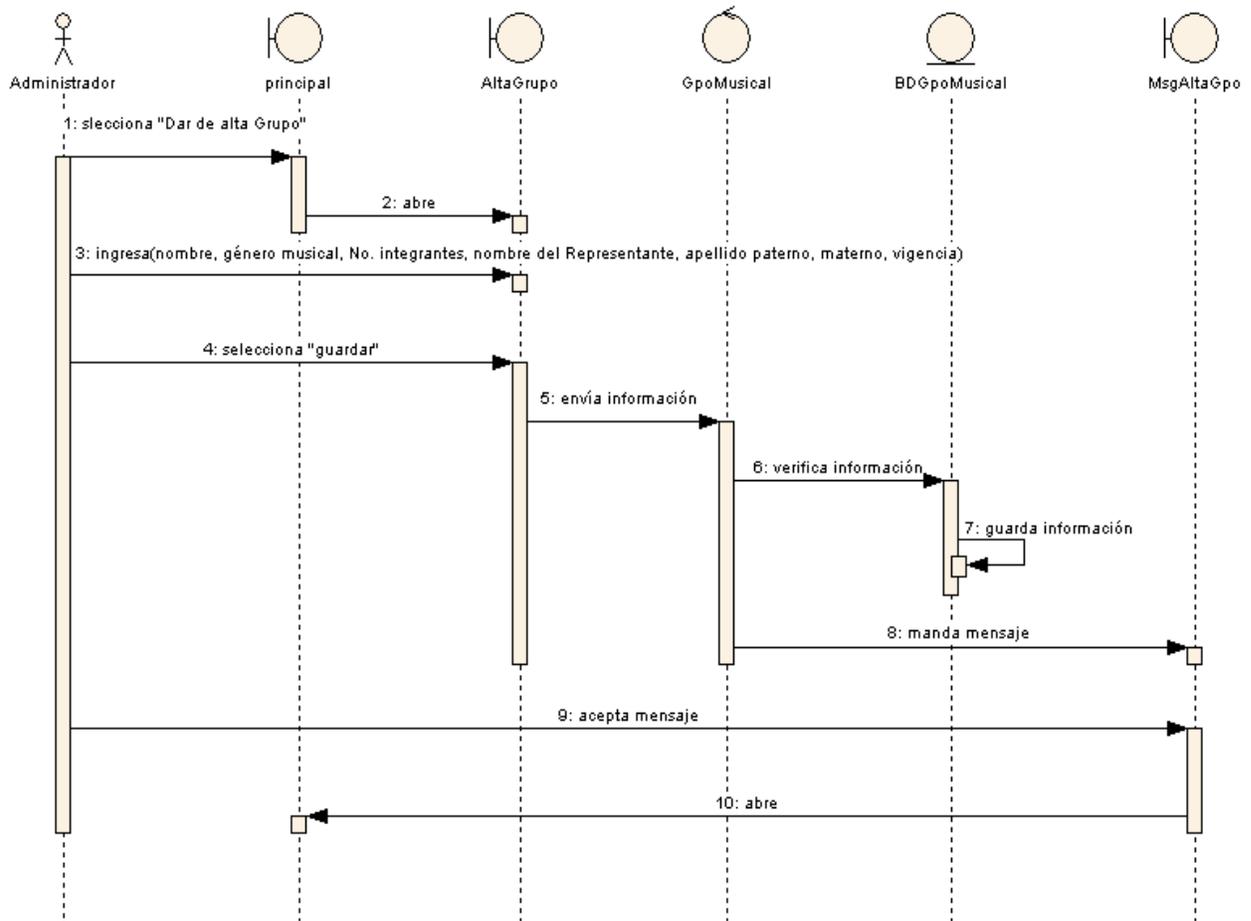
### 3.6 Diagrama de Secuencias

Muestra gráficamente los eventos que fluyen de los actores al sistema, además representa la forma en como un actor o cliente y las clases, se comunican entre sí en petición a un evento.

A continuación se muestran los Diagramas de Secuencia para la Agencia:

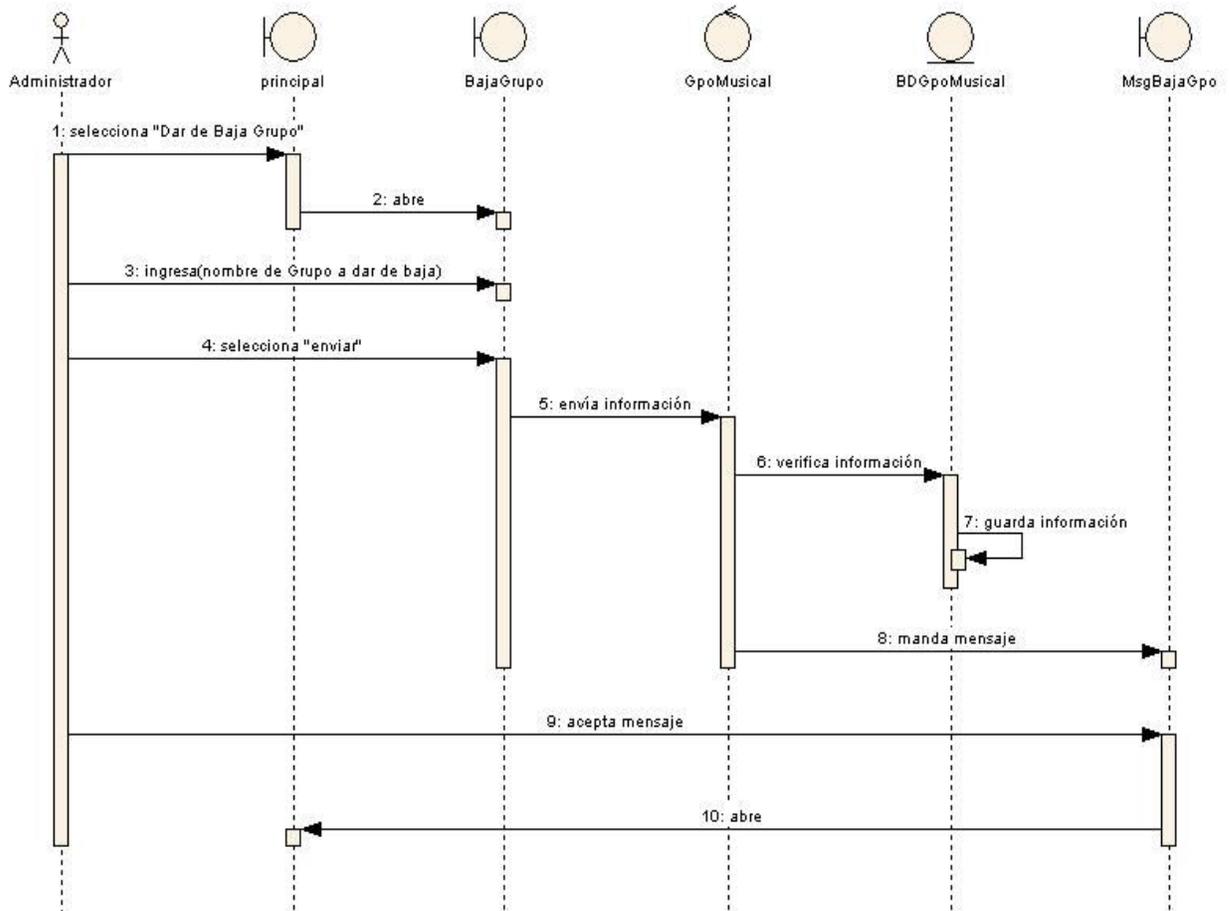
#### Administrador

#### Dar de Alta Grupo



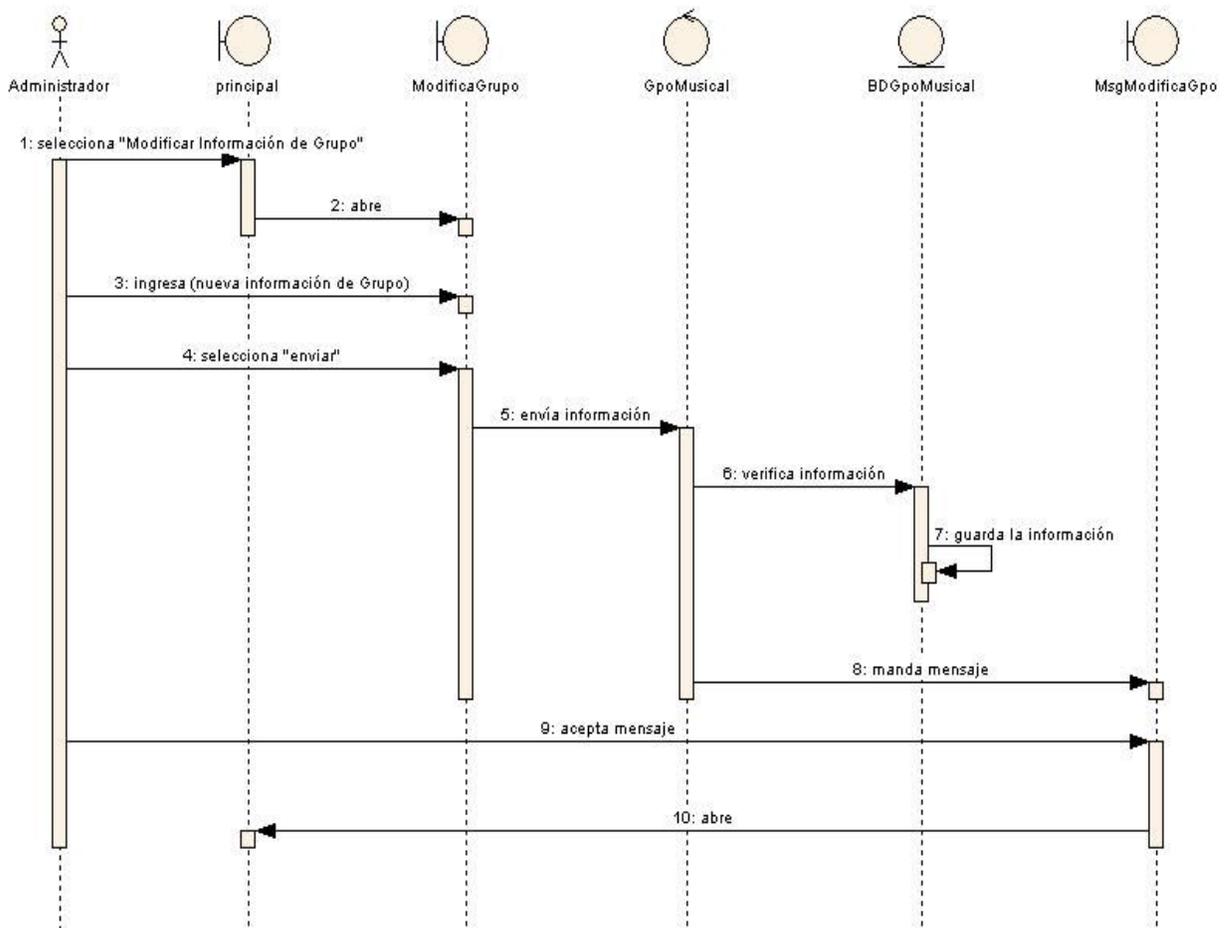
## Administrador

### Dar de Baja Grupo



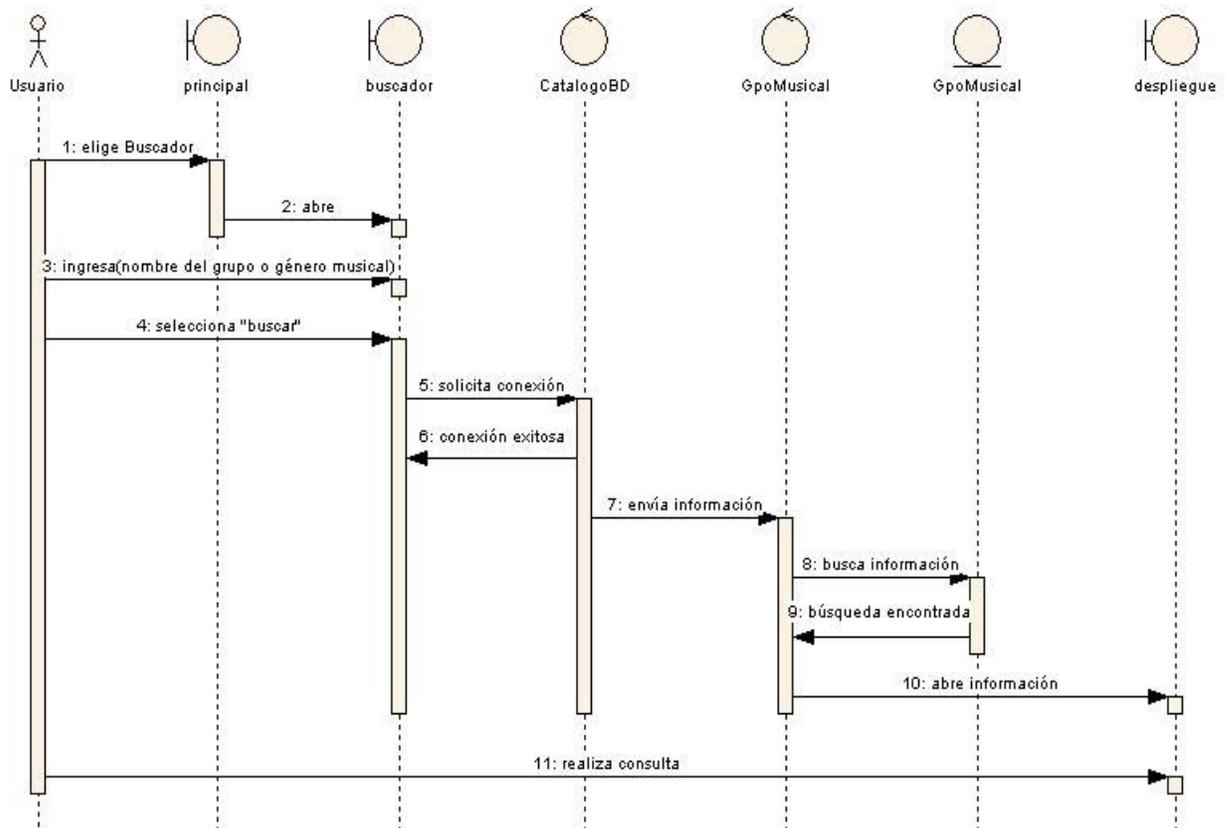
## Administrador

### Modificar información de Grupo



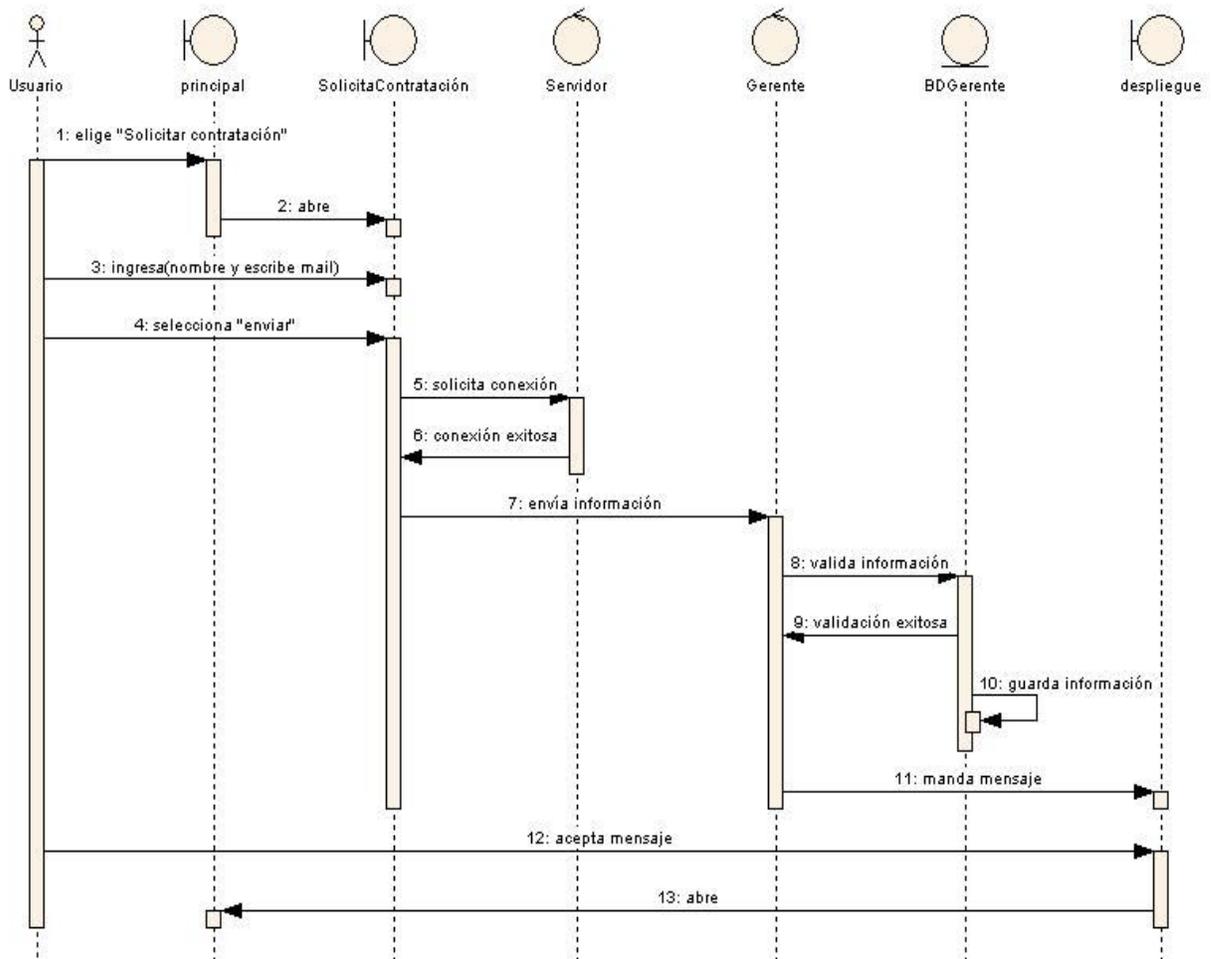
## Usuario

### Consultar información de Grupos



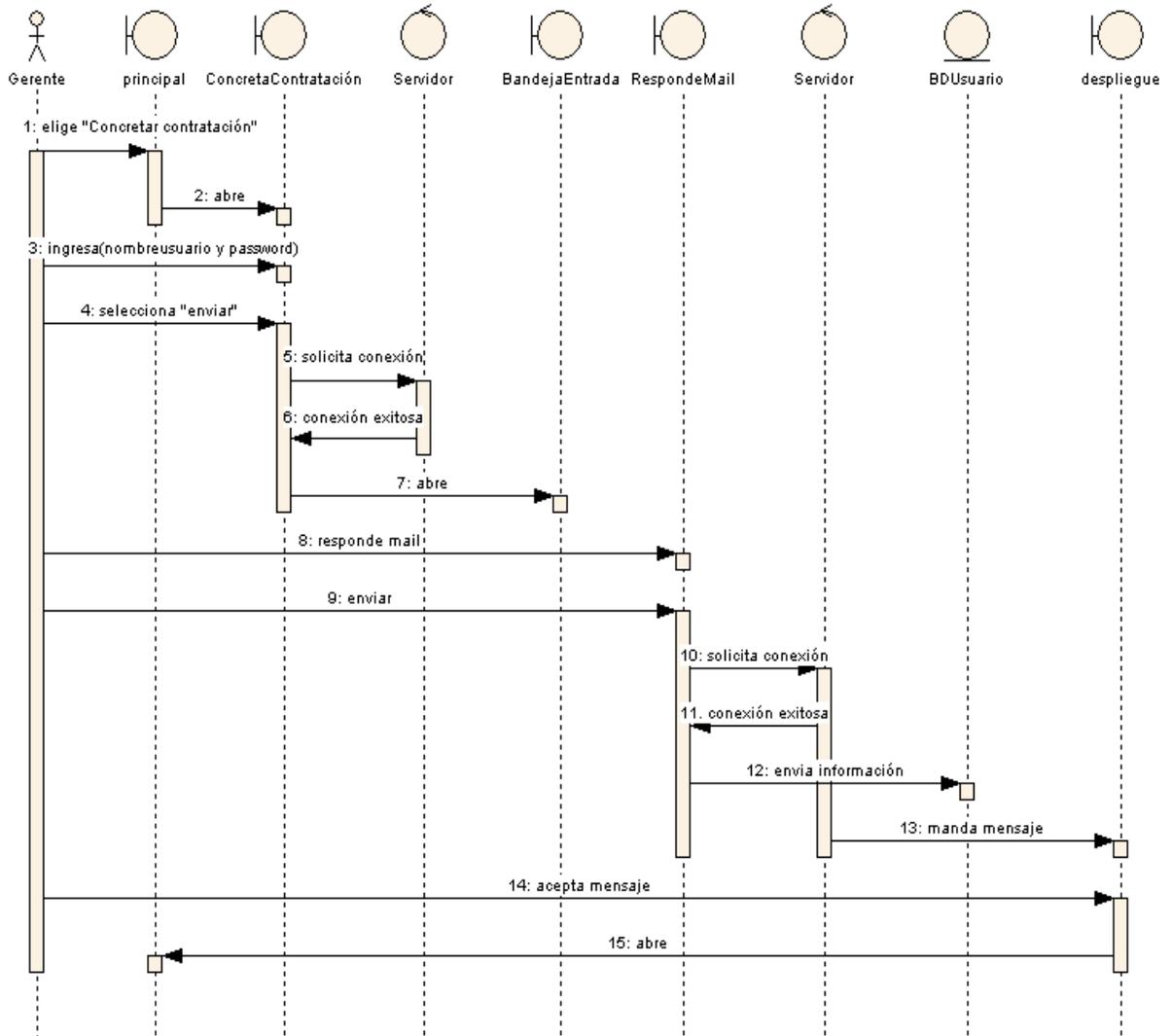
## Usuario

### Solicitar contratación



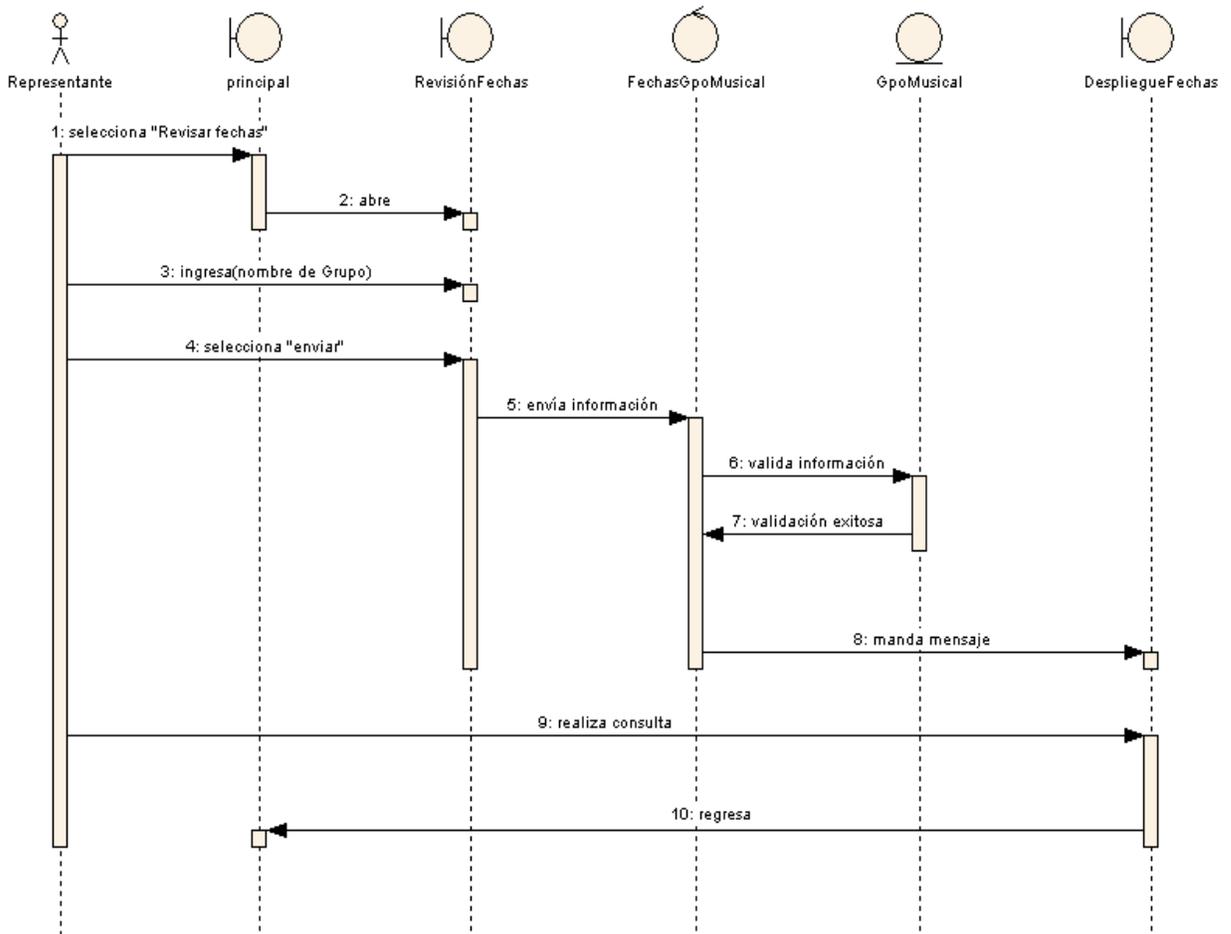
## Gerente

### Concretar contratación



## Representante

### Revisar fechas



# **CAPÍTULO 4**

## **IMPLEMENTACIÓN DEL SISTEMA**

## 4.1 Introducción

En este último Capítulo se trabajará en la fase de Implementación del Sistema y el Diseño de la Base de Datos, a fin de cubrir los Requerimientos que el Sistema exige.

Los objetivos de este capítulo son:

- La creación de la Base de Datos y las Tablas que se van a utilizar en el Sistema.
- Implementar el Sistema con las funciones descritas en los Requerimientos.
- Especificar las herramientas utilizadas en el Sistema.

## 4.2 Diseño de la Base de Datos

El diseño de una Base de Datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

La elección del motor de Bases de Datos a utilizar es una de las primeras decisiones más importantes que se toman. Todos los manejadores tienen sus ventajas y sus desventajas; por ello se debe hallar un punto medio entre las posibilidades que nos brindan y su facilidad de implementación. En este caso utilizaremos Sql Server para crear la Base de Datos y las Tablas.

Se pensó utilizar este manejador ya que sus características principales satisfacen de manera apropiada los requerimientos y características del Sistema. Este manejador tiene características como las que a continuación se mencionan:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.
- Algunos costos y beneficios de utilizar Sql Server serían los siguientes:

Reduce el costo de hardware y el mantenimiento al proveer una solución de consolidación de servidores flexible que permite a las organizaciones centralizar los servicios de datos en menos servidores.

SQL Server soporta servidores de bases de datos altamente escalables gracias a la habilidad de tomar ventaja de sistemas de múltiples procesadores. Esto reduce el tiempo fuera de servicio, lo que permite a los administradores agregar recursos para aumentar un servidor sin detener los servicios de base de datos.

SQL Server provee virtualización y soporte para múltiples instancias de servicio en un solo servidor físico, además de las herramientas y funcionalidades que son necesarias para migrar datos distribuidos a un centro de datos consolidado. Al proveer una interfase de servicio de administración de datos centralizada y administración de configuración basada en políticas declaradas, SQL Server ayuda a las organizaciones con múltiples fuentes de datos a reducir el trabajo administrativo.

Finalmente, SQL Server, al liderar la industria en desempeño y con la habilidad para controlar explícitamente la forma de utilizar los recursos de su servidor, permite consolidar sus servicios de datos mientras entrega escalabilidad y desempeño para las necesidades de sus aplicaciones.

Todas estas características se acoplan correctamente a las necesidades que exige el Sistema, por ello de que se eligió trabajar con este manejador.

Para realizar el Modelado de la Base de Datos para el Sistema se utilizó la Herramienta Erwin Data Modeler. A continuación se explicarán algunos conceptos básicos para entender lo que es un Modelo y un Modelado de Base de Datos.

### ***Modelo***

Un modelo es una abstracción de algo con el propósito de comprenderlo antes de construirlo. Aísla los detalles importantes de aquellos que no lo son, con base en un propósito específico.

### **4.3 Modelo Entidad-Relación**

El Modelo Entidad-Relación es un concepto de modelado para bases de datos, mediante el cual se pretende visualizar los objetos que pertenecen a la Base de Datos como entidades las cuales tienen unos atributos y se vinculan mediante relaciones.

El modelado entidad-relación es una *técnica* para el modelado de datos utilizando diagramas entidad relación.

### **Entidades**

Una Entidad es una persona, lugar, cosa, evento o concepto acerca del cual el negocio conserva datos. Cada Entidad es una colección de objetos similares llamados *instancias*. Cada instancia debe de ser identificable en forma única y distinta de todas las otras instancias.

A continuación se mencionan las Entidades sobre el Sistema Web:

- Administrador (Persona que llevará la Administración del Sistema Web).
- Grupo Musical (Entidad dada de alta en el Sistema).
- Representante (Persona relacionada a un Grupo Musical).
- Género Musical (Entidad dada de alta en el Sistema).
- Evento Social (Entidad dada de alta en el Sistema).
- Gerente (Persona encargada del Sistema Web).

### **Atributos**

Un Atributo es una característica distintiva de una Entidad para la cual se mantienen datos. Existen dos tipos de Atributos: Llave y No-Llave.

- Administrador (Id\_Admin, Password, Nombre, Ap\_Paterno, Ap\_Materno, Sexo, Fecha\_Contratacion).
- Grupo Musical (Id\_Grupo, Nombre, Genero\_Musical, Numero\_Integrantes, Fecha\_Alta, Fecha\_Baja, Precio\_Contrato, Telefono, Id\_Genero).
- Representante (Id\_Representante, Nombre, Ap\_Paterno, Ap\_Materno, Id\_Grupo).
- Género Musical (Id\_Genero, Tipo\_Genero).
- Evento Social (Id\_Evento, Tipo\_Evento, Duración).
- Gerente (Id\_Gerente, Nombre, Ap\_Paterno, Ap\_Materno, Sexo).

### **Relaciones**

Una Relación es una liga lógica entre dos Entidades que representa una regla de negocio o restricción.

ADMINISTRADOR **CONTROLA** GRUPO MUSICAL

GRUPO MUSICAL **TIENE UN** REPRESENTANTE

GRUPO MUSICAL **ASISTE A UN** EVENTO SOCIAL

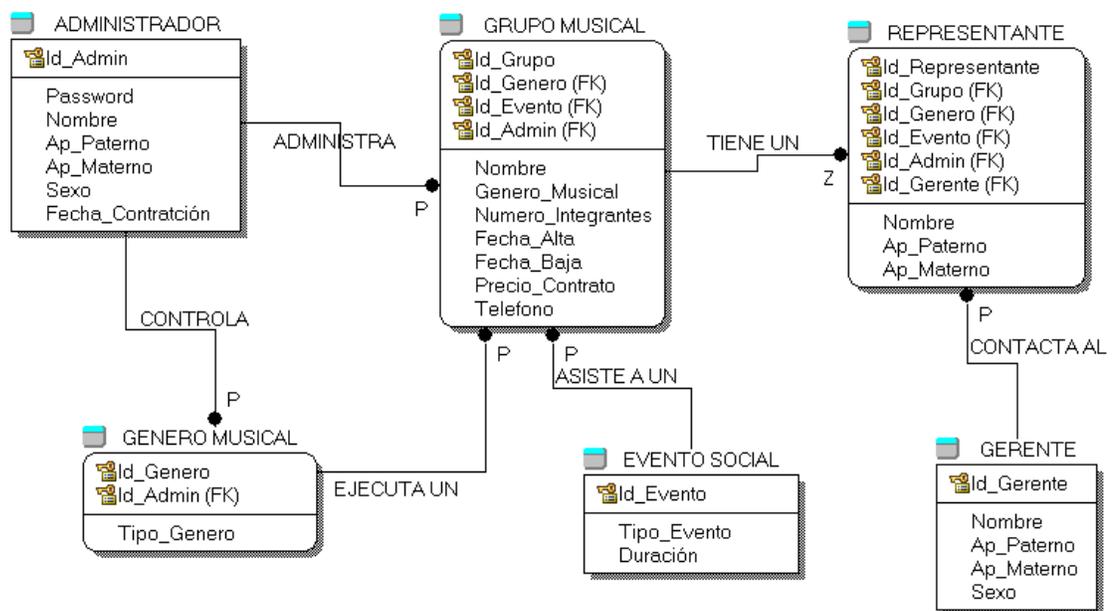
ADMINISTRADOR **CONTROLA** GENERO MUSICAL

GRUPO MUSICAL **TIENE UN** GENERO MUSICAL

GERENTE **CONTACTA AL** REPRESENTANTE

A continuación de acuerdo con los datos descritos anteriormente se muestra el diagrama general entidad-relación para el Sistema Web, donde se pueden ver las entidades, sus atributos y las relaciones que existen entre las mismas.

### Diagrama General Entidad-Relación



### Modelo de Datos

Es una especificación de las estructuras de datos y reglas de negocio para representar los requerimientos del negocio.

### Modelado de Datos

Es un análisis estructurado utilizado para identificar los principales componentes de de un sistema.

#### 4.4 Diccionario de Datos

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

El diccionario de datos define los datos de la siguiente manera:

- Describe el significado de los flujos y almacenes que se muestran en los Diagramas.
- Describe la composición de agregados de paquetes de datos que se mueven a lo largo de los flujos, por ejemplo, como domicilio de un usuario, puede descomponerse en unidades como ciudad, estado, código postal, etc.
- Describen la composición de los paquetes de datos en los almacenes.
- Especifica los valores y unidades relevantes de piezas elementales de información en los flujos de datos y en los almacenes de datos.
- Describe los detalles de las relaciones entre almacenes que se enfatizan en un diagrama de entidad-relación.

Este diccionario se desarrolla durante el análisis de flujo de datos y ayuda a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño del proyecto.

#### Diccionario de Datos

Nombre de la Entidad	Nombre del Atributo	Descripción	Tamaño
Administrador	Id_Admin	*Dígitos* [0 1 2 3 ...9]	numeric (2,0)
	Password	*Alfanumerico* [A-Z  a-z 0 1 2 3 ...9]	varchar (20)
	Nombre	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Ap_Paterno	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Ap_Materno	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Sexo	*Cadena de Caracteres* [A-Z  a-z]	char (1)
	Fecha_Contratacion	*Alfanumerico* [A-Z  a-z 0 1 2 3 ...9]	datetime
Representante	Id_Representante	*Dígitos*	numeric (2,0)

		[0 1 2 3 ...9]	
	Nombre	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Ap_Paterno	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Ap_Materno	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Id_Grupo	*Dígitos* [0 1 2 3 ...9]	numeric (4,0)
Genero_Musical	Id_Genero	*Dígitos* [0 1 2 3 ...9]	numeric (4,0)
	Tipo_Genero	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
Grupo_Musical	Id_Grupo	*Dígitos* [0 1 2 3 ...9]	numeric (4,0)
	Nombre	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Genero_Musical	*Cadena de Caracteres* [A-Z  a-z]	varchar (15)
	Numero_Integrantes	*Dígitos* [0 1 2 3 ...9]	numeric (2,0)
	Fecha_Alta	*Alfanumerico* [A-Z  a-z 0 1 2 3 ...9]	datetime
	Fecha_Baja	*Alfanumerico* [A-Z  a-z 0 1 2 3 ...9]	datetime
	Precio_Contrato	*Dígitos* [0 1 2 3 ...9]	numeric (2,0)
	Telefono	*Alfanumerico* [A-Z  a-z 0 1 2 3 ...9]	varchar (15)
	Id_Genero	*Dígitos* [0 1 2 3 ...9]	numeric (4,0)
Evento_Social	Id_Evento	*Dígitos* [0 1 2 3 ...9]	numeric (4,0)
	Tipo_Evento	*Cadena de Caracteres* [A-Z  a-z]	varchar (30)
	Duracion	*Dígitos* [0 1 2 3 ...9]	numeric (2,0)
Gerente	Id_Gerente	*Dígitos* [0 1 2 3 ...9]	numeric (4,0)
	Nombre	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Ap_Paterno	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Ap_Materno	*Cadena de Caracteres* [A-Z  a-z]	varchar (50)
	Sexo	*Cadena de Caracteres* [A-Z  a-z]	char (1)

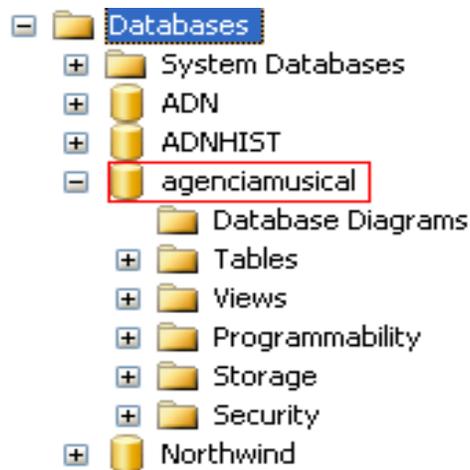
## 4.5 Creación de la Base de Datos

A continuación con los antecedentes ya mencionados se procederá a crear la Base de Datos en Microsoft SQL Server Management Studio 2005.

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de la Base de Datos  
-----  
  
SET NOCOUNT ON  
GO  
  
USE master  
GO  
if exists (select * from sysdatabases where name='agenciamusical')  
    drop database agenciamusical  
GO  
if not exists (select * from sysdatabases where name='agenciamusical')  
    create database agenciamusical  
GO
```

Results

The CREATE DATABASE process is allocating 0.63 MB on disk 'agenciamusical'.  
The CREATE DATABASE process is allocating 0.49 MB on disk 'agenciamusical\_log'.

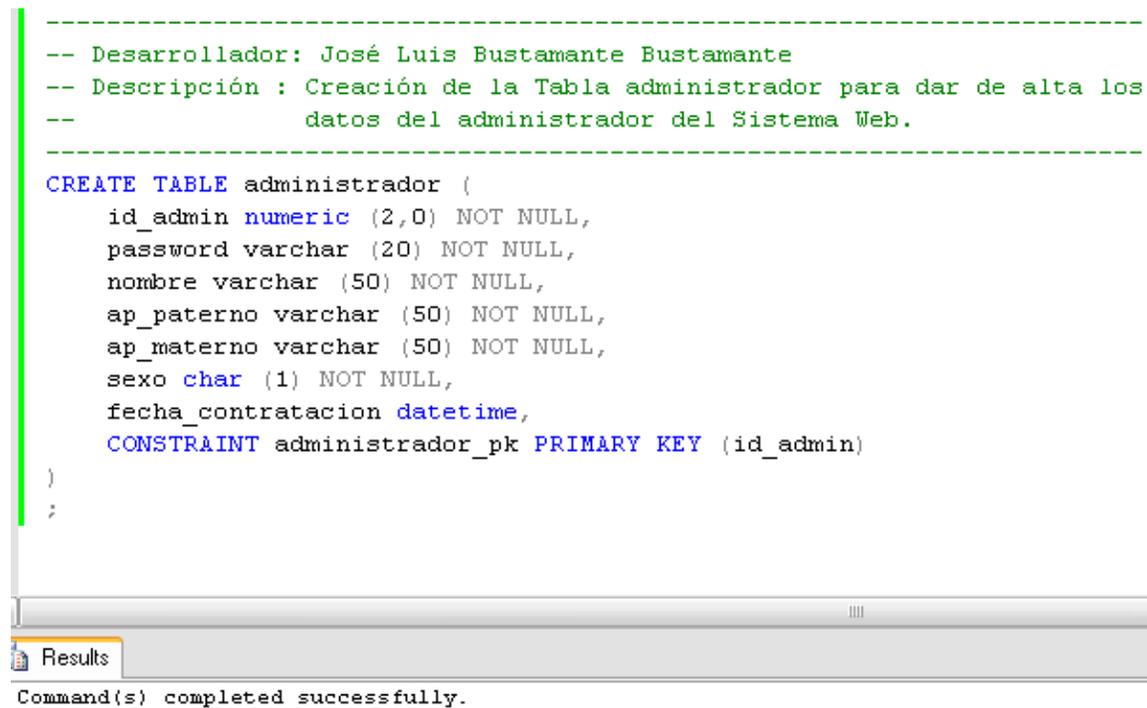


#### 4.5.1 Creación de Tablas, Llaves Primarias y Foráneas

A continuación se procederá a crear las Tablas dentro de la Base de Datos ya creada.

##### Creación de la Tabla administrador

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de la Tabla administrador para dar de alta los  
--               datos del administrador del Sistema Web.  
-----  
  
CREATE TABLE administrador (  
    id_admin numeric (2,0) NOT NULL,  
    password varchar (20) NOT NULL,  
    nombre varchar (50) NOT NULL,  
    ap_paterno varchar (50) NOT NULL,  
    ap_materno varchar (50) NOT NULL,  
    sexo char (1) NOT NULL,  
    fecha_contratacion datetime,  
    CONSTRAINT administrador_pk PRIMARY KEY (id_admin)  
)  
;
```



Results

Command(s) completed successfully.

### Creación de la Tabla grupo \_ musical

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de la Tabla grupo_musical para dar de alta todos  
--               los grupos musicales que estarán en el Sistema Web.  
-----  
  
CREATE TABLE grupo_musical (  
    id_grupo numeric (4,0) NOT NULL,  
    nombre varchar (50) NOT NULL,  
    genero_musical varchar (15) NOT NULL,  
    numero_integrantes numeric (2,0) NOT NULL,  
    fecha_alta datetime NOT NULL,  
    fecha_baja datetime,  
    precio_contrato numeric (8,2) NOT NULL,  
    telefono varchar (15),  
    id_genero numeric (4,0) NOT NULL,  
    CONSTRAINT grupo_musical_pk PRIMARY KEY (id_grupo)  
)  
;
```

Results

Command(s) completed successfully.

### Creación de la Tabla evento \_ social

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de la Tabla evento_social para dar de alta todos  
--               los diferentes tipos de eventos en donde se podrán presentar  
--               los grupos musicales dados de alta en el Sistema Web. |  
-----  
  
CREATE TABLE evento_social (  
    id_evento numeric (4,0) NOT NULL,  
    tipo_evento varchar (30) NOT NULL,  
    duracion numeric (2,0) NOT NULL,  
    CONSTRAINT evento_social_pk PRIMARY KEY (id_evento)  
)  
;
```

Results

Command(s) completed successfully.

### Creación de la Tabla representante

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de la Tabla representante para dar de alta los  
--               datos de los Representantes de cada Grupo Musical. |  
-----  
  
CREATE TABLE representante (  
    id_representante numeric (2,0) NOT NULL,  
    nombre varchar (50) NOT NULL,  
    ap_paterno varchar (50) NOT NULL,  
    ap_materno varchar (50) NOT NULL,  
    id_grupo numeric (4,0) NOT NULL,  
    CONSTRAINT representante_pk PRIMARY KEY (id_representante)  
)  
;
```

Results

Command(s) completed successfully.

### Creación de la Tabla genero \_ musical

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de la Tabla genero_musical para dar de alta  
--               todos los generos musicales que ejecutarán los Grupos  
--               Musicales dados de alta en el Sistema Web. |  
-----  
  
CREATE TABLE genero_musical (  
    id_genero numeric (4,0) NOT NULL,  
    tipo_genero varchar (30) NOT NULL,  
    CONSTRAINT genero_musical_pk PRIMARY KEY (id_genero)  
)  
;
```

Results

Command(s) completed successfully.

### Creación de la Tabla gerente

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de la Tabla gerente para dar de alta los datos  
--               de la persona encargada del Sistema Web.  
-----  
  
CREATE TABLE gerente (  
    id_gerente numeric (4,0) NOT NULL,  
    nombre varchar (50) NOT NULL,  
    ap_paterno varchar (50) NOT NULL,  
    ap_materno varchar (50) NOT NULL,  
    sexo char (1) NOT NULL,  
    CONSTRAINT gerente_pk PRIMARY KEY (id_gerente)  
)  
;  
  
Results  
Command(s) completed successfully.
```

### Creación de Llaves Foráneas

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de Llave Foránea en la Tabla grupo_musical  
-----  
  
ALTER TABLE grupo_musical  
ADD  
    CONSTRAINT fk_grupo_genero  
    FOREIGN KEY (id_genero)  
    REFERENCES genero_musical (id_genero);  
  
Results  
Command(s) completed successfully.
```

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Creación de Llave Foránea en la Tabla representante  
-----  
  
ALTER TABLE representante  
ADD  
    CONSTRAINT fk_repre_grupo  
    FOREIGN KEY (id_grupo)  
    REFERENCES grupo_musical (id_grupo);  
  
Results  
Command(s) completed successfully.
```

#### 4.5.2 Inserción de Datos

##### *Inserción de datos en la Tabla administrador*

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Insertar datos en la Tabla administrador  
-----  
  
INSERT INTO administrador  
VALUES ('admin', 'administrator', 'DAVID', 'CISNEROS', 'GUTIERREZ', 'M', '01-12-2008')  
  
Results  
  
(1 row(s) affected)
```

### Inserción de datos en la Tabla grupo \_ musical

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Insertar datos en la Tabla grupo_musical  
-----  
  
INSERT INTO grupo_musical  
VALUES (1, 'BANDA ALAKRAN', 'BANDA', 20, '08-12-2008', null, 30.000, '57484846', 1)  
INSERT INTO grupo_musical  
VALUES (2, 'BANDA RODEO', 'BANDA', 18, '08-12-2008', null, 25.000, '51122356', 1)  
INSERT INTO grupo_musical  
VALUES (3, 'MARIACHI SANTA CECILIA', 'MARIACHI', 16, '08-12-2008', null, 20.000, '85828745', 7)  
INSERT INTO grupo_musical  
VALUES (4, 'MARIACHI AMANECER RANCHERO', 'MARIACHI', 15, '08-12-2008', null, 20.000, '74859652', 7)  
INSERT INTO grupo_musical  
VALUES (5, 'OXIDO', 'ROCK', 5, '08-12-2008', null, 10.000, '85121000', 2)  
INSERT INTO grupo_musical  
VALUES (6, 'GASOLINA', 'ROCK', 6, '08-12-2008', null, 12.000, '74858695', 2)  
INSERT INTO grupo_musical  
VALUES (7, 'LOS CANTANTES', 'SALSA', 10, '08-12-2008', null, 15.000, '57357241', 8)  
INSERT INTO grupo_musical  
VALUES (8, 'SALSA RIKA', 'SALSA', 10, '08-12-2008', null, 15.000, '45424100', 8)  
INSERT INTO grupo_musical  
VALUES (9, 'VALIOZZO NORTE', 'NORTEÑO', 5, '08-12-2008', null, 15.000, '57849652', 3)  
INSERT INTO grupo_musical  
VALUES (10, 'PESADO', 'NORTEÑO', 5, '08-12-2008', null, 20.000, '57415655', 3)  
INSERT INTO grupo_musical  
VALUES (11, 'GRUPO TRIPULACION', 'VERSATIL', 7, '08-12-2008', null, 20.000, '57893230', 6)  
INSERT INTO grupo_musical  
VALUES (12, 'GRUPO ILUSION', 'VERSATIL', 8, '08-12-2008', null, 25.000, '56428910', 6)  
INSERT INTO grupo_musical  
VALUES (13, 'TE DE TILA', 'POP', 5, '08-12-2008', null, 15.000, '56525458', 4)  
INSERT INTO grupo_musical  
VALUES (14, 'POPSTARS', 'POP', 5, '08-12-2008', null, 15.000, '14151612', 4)  
INSERT INTO grupo_musical  
VALUES (15, 'VENGANZA DE MÉXICO', 'DURANGUENSE', 7, '08-12-2008', null, 18.000, '57898495', 5)  
INSERT INTO grupo_musical  
VALUES (16, 'AK7', 'DURANGUENSE', 7, '08-12-2008', null, 18.000, '41454200', 5)
```

Results

```
(1 row(s) affected)  
  
(1 row(s) affected)  
  
(1 row(s) affected)  
  
(1 row(s) affected)  
  
(1 row(s) affected)
```

### ***Inserción de datos en la Tabla genero \_ musical***

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Insertar datos en la Tabla genero_musical  
-----
```

```
INSERT INTO genero_musical  
VALUES (1, 'BANDA')  
INSERT INTO genero_musical  
VALUES (2, 'ROCK')  
INSERT INTO genero_musical  
VALUES (3, 'NORTEÑO')  
INSERT INTO genero_musical  
VALUES (4, 'POP')  
INSERT INTO genero_musical  
VALUES (5, 'DURANGUENSE')  
INSERT INTO genero_musical  
VALUES (6, 'VERSATIL')  
INSERT INTO genero_musical  
VALUES (7, 'MARIACHI')  
INSERT INTO genero_musical  
VALUES (8, 'SALSA')
```

Results

```
(1 row(s) affected)  
  
(1 row(s) affected)
```

---

### Inserción de datos en la Tabla evento\_social

```
-----  
-- Desarrollador: José Luis Bustamante Bustamante  
-- Descripción : Insertar datos en la Tabla evento_social  
-----
```

```
INSERT INTO evento_social  
VALUES (1, 'BODA', 5)  
INSERT INTO evento_social  
VALUES (2, 'XV AÑOS', 5)  
INSERT INTO evento_social  
VALUES (3, 'BAILES', 2)  
INSERT INTO evento_social  
VALUES (4, 'RODEOS', 2)  
INSERT INTO evento_social  
VALUES (5, 'LUGARES CERRADOS', 2)  
INSERT INTO evento_social  
VALUES (6, 'FIESTAS POPULARES', 5)  
INSERT INTO evento_social  
VALUES (7, 'PRESENTACIONES', 5)  
INSERT INTO evento_social  
VALUES (8, 'OTROS', 5)
```

Results

```
(1 row(s) affected)  
  
(1 row(s) affected)
```

### Inserción de datos en la Tabla representante

```
-- Desarrollador: José Luis Bustamante Bustamante
-- Descripción : Insertar datos en la Tabla representante
-----
INSERT INTO representante
VALUES (1, 'ROGELIO', 'GUERRA', 'MARTINEZ', 16)
INSERT INTO representante
VALUES (2, 'ALBERTO', 'TAMAYO', 'ACOSTA', 15)
INSERT INTO representante
VALUES (3, 'LAURA', 'ESPINOZA', 'GUTIERREZ', 14)
INSERT INTO representante
VALUES (4, 'HUGO', 'PINEDA', 'CERVANTES', 13)
INSERT INTO representante
VALUES (5, 'CESAR', 'CATAÑO', 'AGUILAR', 12)
INSERT INTO representante
VALUES (6, 'LUIS', 'BUSTOS', 'MORALES', 11)
INSERT INTO representante
VALUES (7, 'ERNESTO', 'GOMEZ', 'PEREZ', 10)
INSERT INTO representante
VALUES (8, 'HECTOR', 'BUSTAMANTE', 'BUSTAMANTE', 9)
INSERT INTO representante
VALUES (9, 'ALEJANDRO', 'AMARO', 'AMBROSIO', 8)
INSERT INTO representante
VALUES (10, 'ULISES', 'QUINTERO', 'MENDOZA', 7)
INSERT INTO representante
VALUES (11, 'ANA', 'QUINTANA', 'MORENO', 6)
INSERT INTO representante
VALUES (12, 'DAVID', 'OLIVAREZ', 'SOLORZANO', 5)
INSERT INTO representante
VALUES (13, 'LUIS ALFONSO', 'MIYAR', 'GONZALEZ', 4)
INSERT INTO representante
VALUES (14, 'JULIO', 'MARTINEZ', 'CARRASCO', 3)
INSERT INTO representante
VALUES (15, 'CARLOS', 'LUNA', 'LANDEROS', 2)
INSERT INTO representante
VALUES (16, 'RICHARD', 'CHAVIRA', 'VIDAL', 1)
```

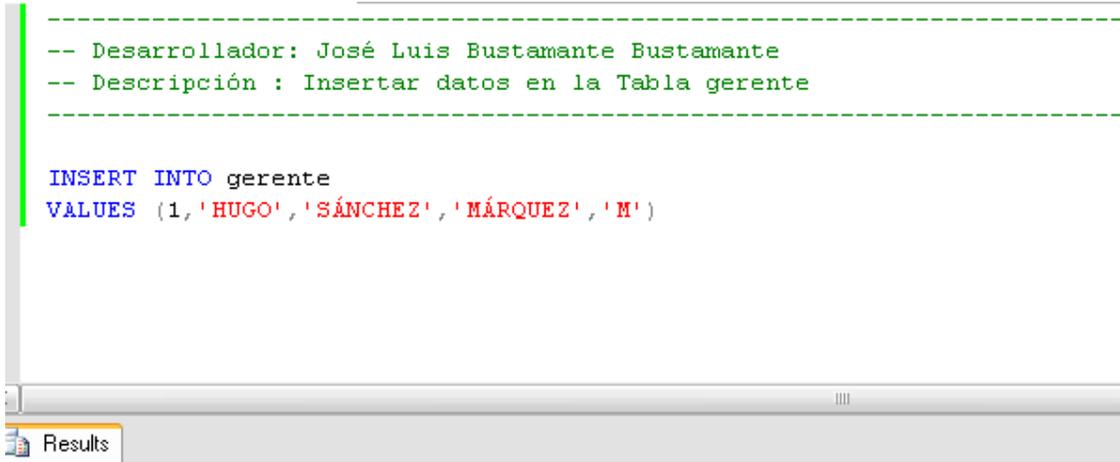
Results

(1 row(s) affected)

### ***Inserción de datos en la Tabla gerente***

```
-- Desarrollador: José Luis Bustamante Bustamante
-- Descripción : Insertar datos en la Tabla gerente

INSERT INTO gerente
VALUES (1, 'HUGO', 'SÁNCHEZ', 'MÁRQUEZ', 'M')
```



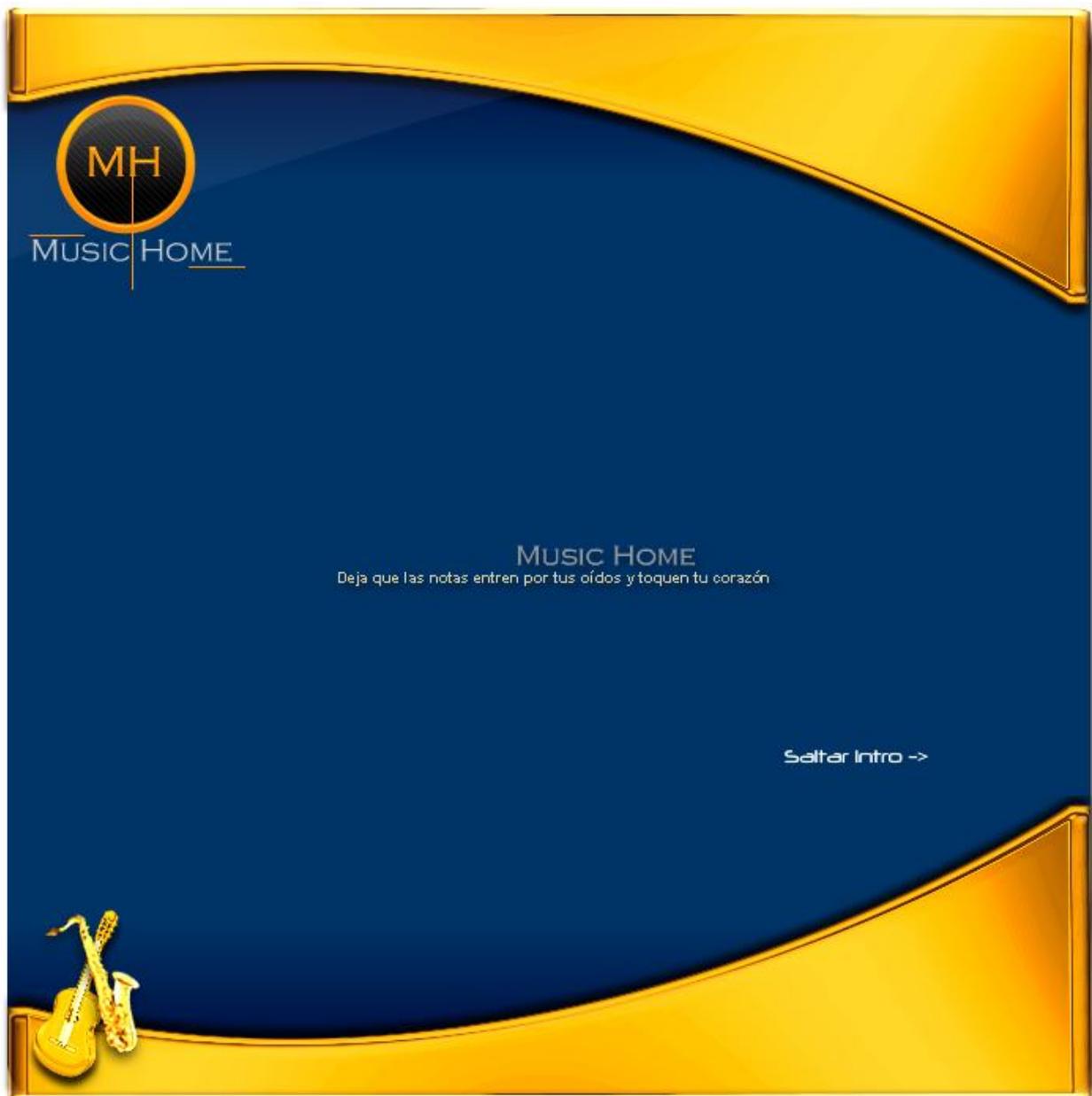
(1 row(s) affected)

## **4.6 Diseño de Interfaces**

En esta sección del proyecto solo se presentaran los códigos y pantallas más importantes del Sistema Web.

#### 4.7 Codificación de la Capa de Negocio

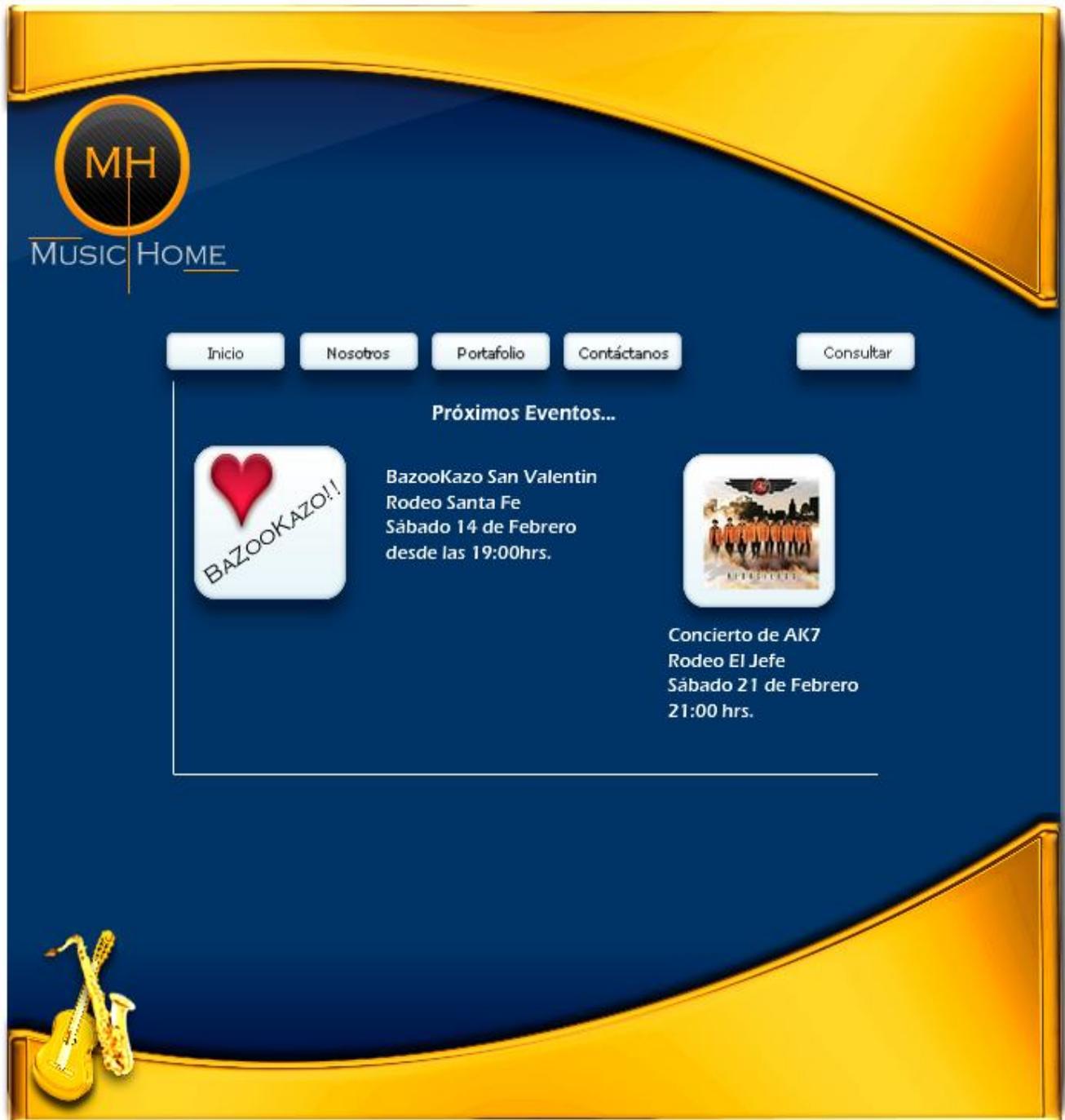
Pantalla principal del Sistema Web **Music Home** (Pantalla Web para todos los usuarios Visitantes).



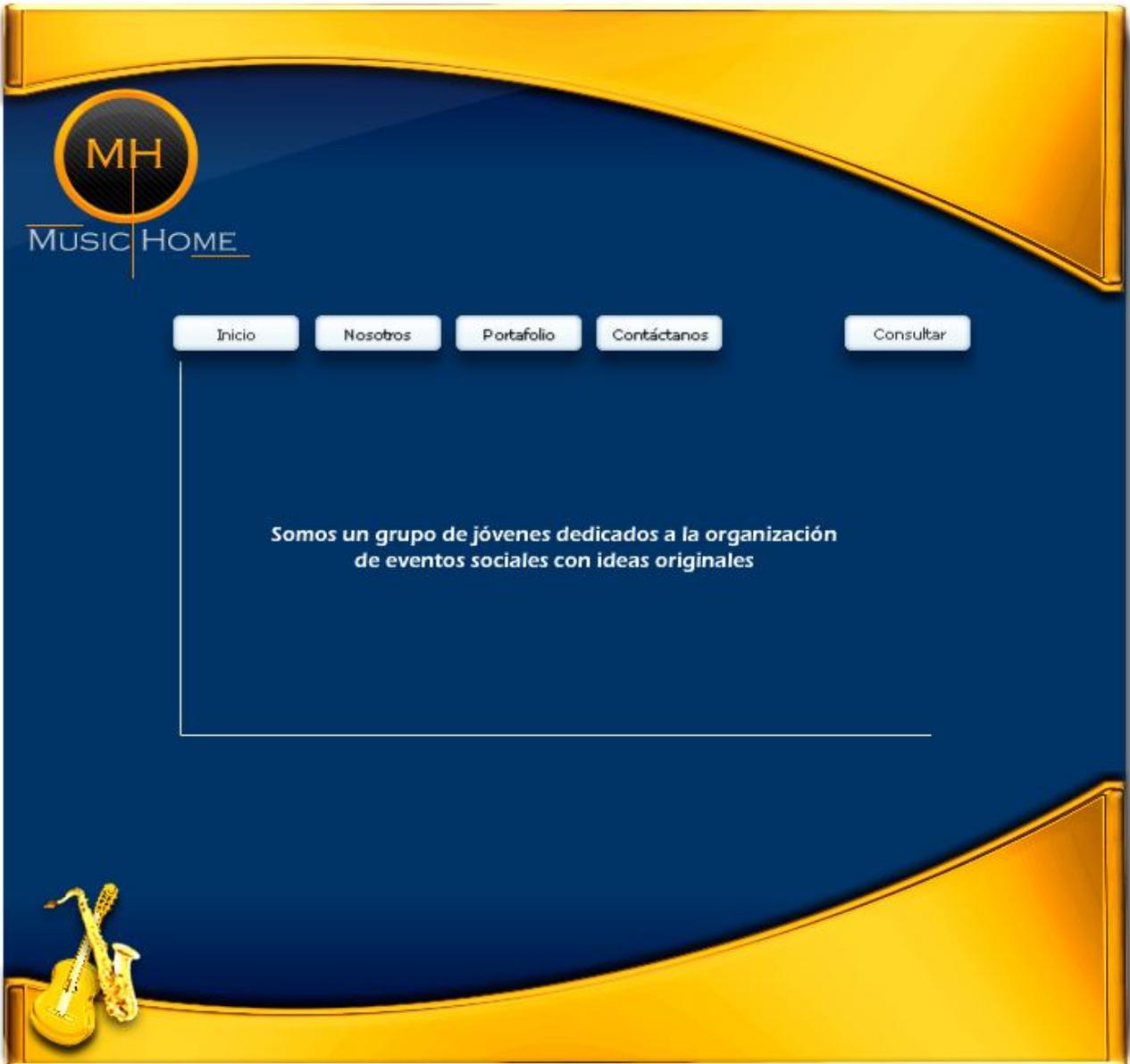
El código para la pantalla principal del sistema es el siguiente:

```
<?php
Echo " <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
Echo " 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>";
Echo " <html xmlns='http://www.w3.org/1999/xhtml'>";
Echo " <head>";
Echo " <meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1' />";
Echo " <title>Music Home</title>";
Echo " <link rel = 'stylesheet' type = 'text/css' href = 'Estilos/Estilos.css' />";
Echo " </head>";
Echo " <body bgcolor = '#FFFFFF'>";
Echo " <center>
    <div class='divTopComplete'>
    </div>
    <div class='divCenterComplete'>
        <embed src='Flash/Intro.swf' width='500px' height='300px'></embed>
    </div>
    <div class='divBottomComplete'>
    </div>
</center>
</body>
</html>";
```

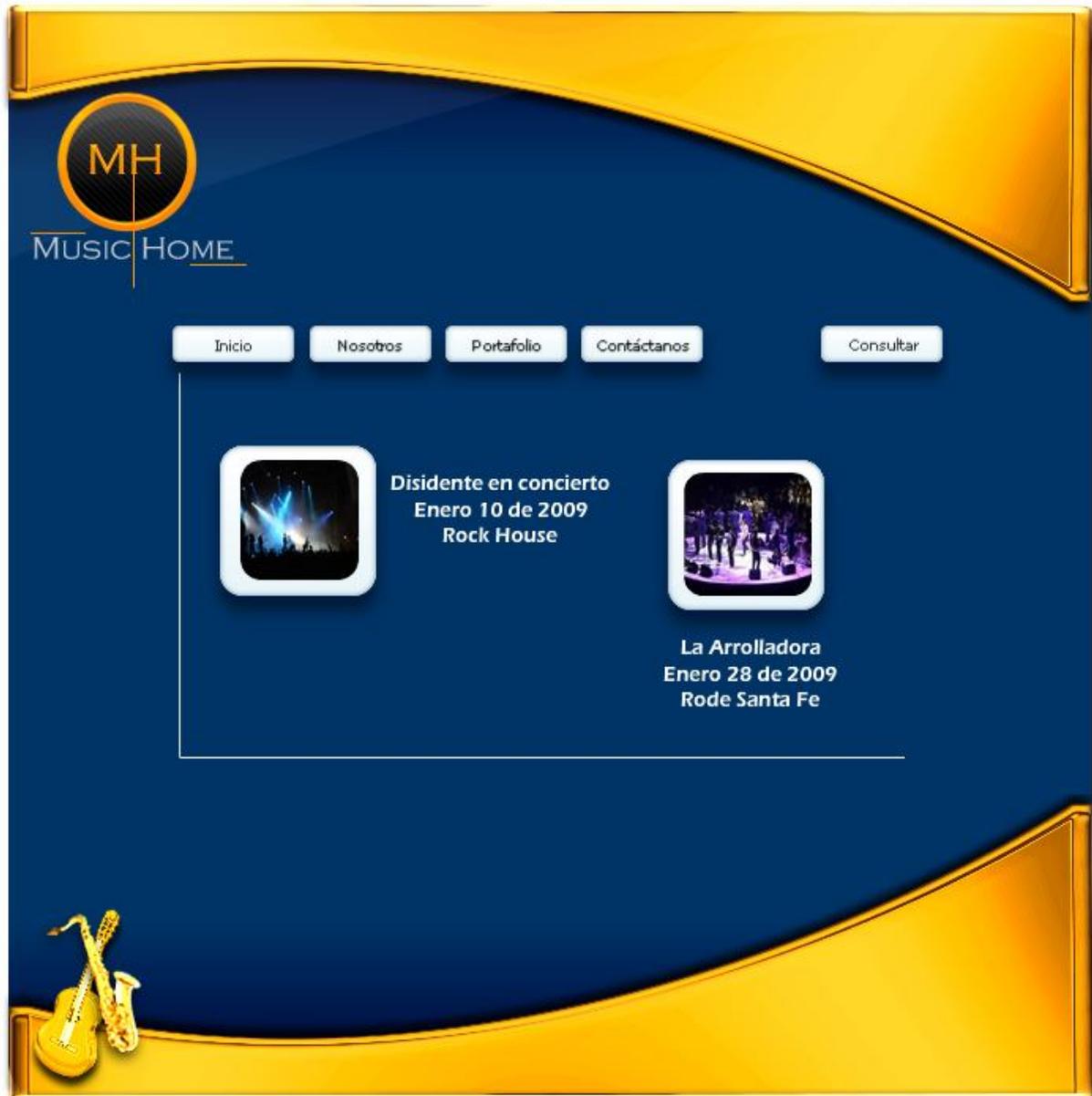
Pantalla del Menú **Inicio** (Usuarios Visitantes).



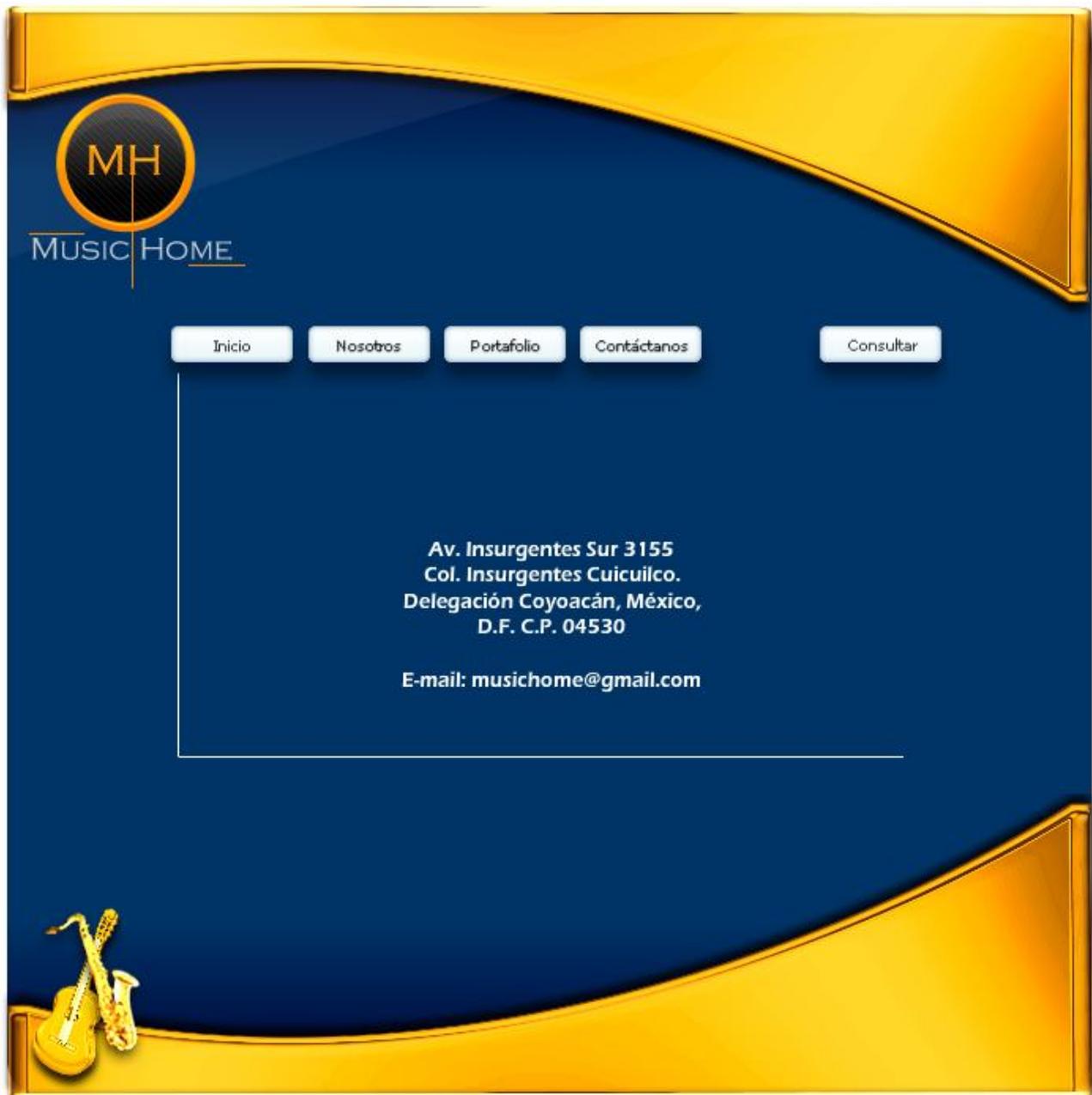
Pantalla de la opción **Nosotros** (Usuarios Visitantes).



Pantalla de la opción **Portafolio** (Usuarios Visitantes).



Pantalla de la opción **Contáctanos** (Usuarios Visitantes).

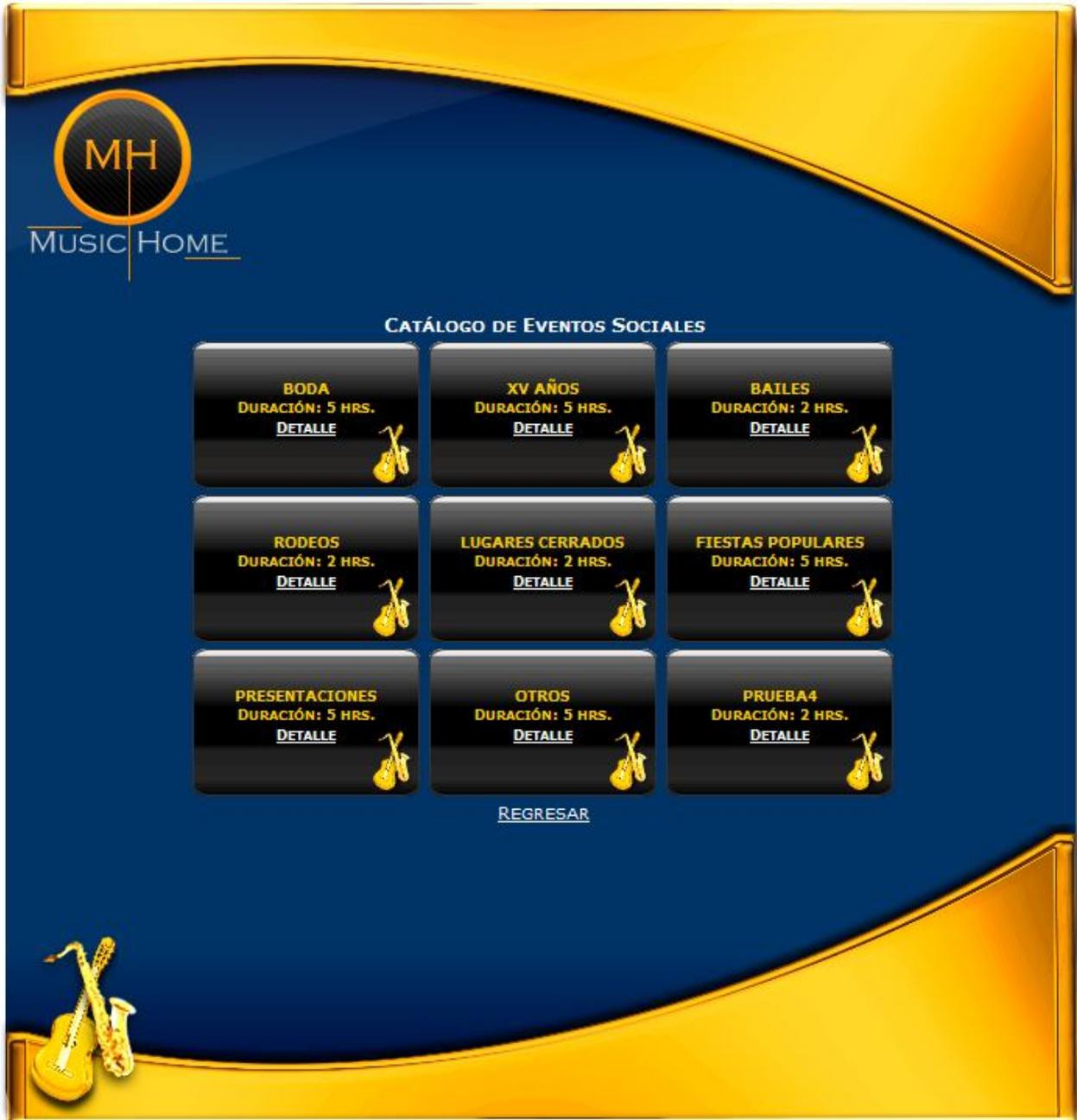


Pantalla de la opción **Consultar (Submenús)** (Usuarios Visitantes).





Pantalla del **Catálogo de Eventos Sociales** (Usuarios Visitantes).

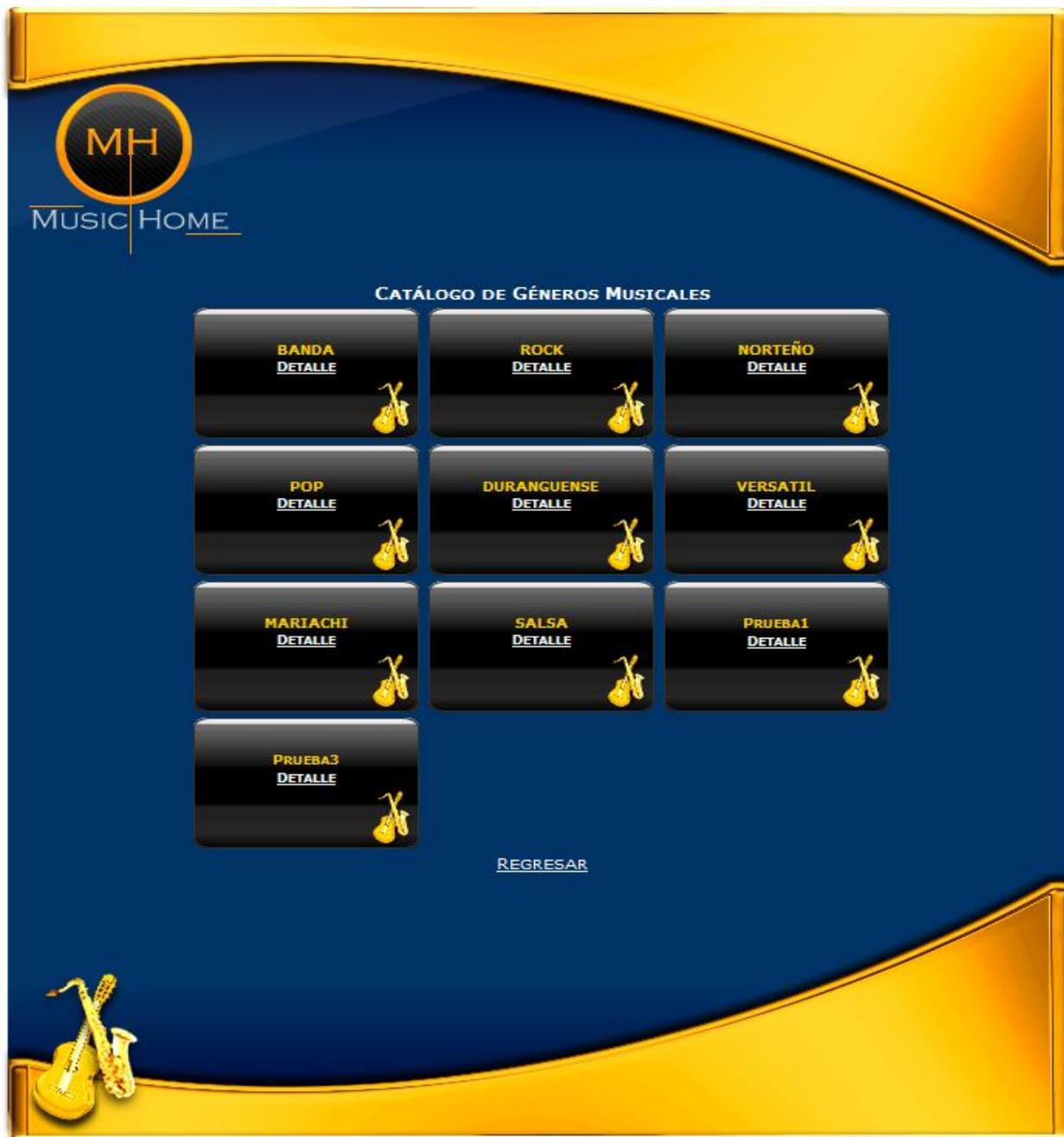


El código para la Pantalla de **Catálogo de Eventos Sociales** es el siguiente:

!<?php

```
$Cnx = mysql_connect("localhost", "devuser", "devuser");
$Bd = mysql_select_db("agenciamusical", $Cnx);
If ($Cnx)
{
    If ($Bd)
    {
        Echo "<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'";
        Echo "'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>";
        Echo "<html xmlns='http://www.w3.org/1999/xhtml'>";
        Echo "<head>";
        Echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1' />";
        Echo "<title>Music Home: (Eventos Sociales)</title>";
        Echo "<link rel = 'stylesheet' type = 'text/css' href = 'Estilos/Estilos.css' />";
        Echo "</head>";
        Echo "<body bgcolor = '#FFFFFF' alink = '#FFFFFF' vlink = '#FFFFFF'>";
        Echo "<center>
            <div class='divTopComplete'>
            </div>
            <div class='divCenterComplete' style='font-weight:bold'>Catálogo de Eventos Sociales
            <table>
            <tr>";
        $Ren = 1;
        $QrySql = "SELECT Id_Evento, Tipo_Evento, Duracion FROM Evento_Social ORDER BY Id_Evento ASC";
        $Consulta = mysql_query($QrySql, $Cnx);
        While ($Rst = mysql_fetch_array($Consulta))
        {
            Echo "            <td><div class='divOpcion2' style='color:#ffcc00;'><br><br>" . $Rst["Tipo_Evento"] . "<br>Duración: " .
            $Rst["Duracion"] . " hrs.<br><a style='color:#ffffff;' href='EventoSocialDetalle.php?IdEvento=" . $Rst["Id_Evento"] . "'>Detalle</a></div>
            </td></div>";
            If ($Ren % 3 == 0)
            {
                Echo "            </tr><tr>";
            }
            $Ren = $Ren + 1;
        }
        Echo "            </tr>";
        Echo "            </table>
            <a style='color:#ffffff; text-align:right; font-size:8pt; font-weight:normal;
            text-variant:small-caps;' href = 'Main.php'>Regresar</a>
            </div>
            <div class='divBottomComplete'>
            </div>
            </center>
            </body>
            </html>";
    }
}
```

Pantalla del **Catálogo de Géneros Musicales** ((Usuarios Visitantes).



El código para la Pantalla del **Catálogo de Géneros Musicales** es el siguiente:

```
<?php
$Cnx      = mysql_connect("localhost", "devuser", "devuser");
$Bd       = mysql_select_db("agenciamusical", $Cnx);
If ($Cnx)
{
    If ($Bd)
    {
        Echo "<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN";
        Echo "'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'";
        Echo "<html xmlns='http://www.w3.org/1999/xhtml'";
        Echo "<head>";
        Echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1' />";
        Echo "<title>Music Home: (Géneros Musicales)</title>";
        Echo "<link rel = 'stylesheet' type = 'text/css' href = 'Estilos/Estilos.css' />";
        Echo "</head>";
        Echo "<body bgcolor = '#FFFFFF' alink = '#FFFFFF' vlink = '#FFFFFF'>";
        Echo "<center>
                <div class='divTopComplete'>
                </div>
                <div class='divCenterComplete' style='font-weight:bold'>Catálogo de Géneros Musicales
                <table>
                <tr>";

        $Ren = 1;
        $QrySql = "SELECT Id_Genero, Tipo_Genero FROM Genero_Musical ORDER BY Id_Genero ASC";
        $Consulta = mysql_query($QrySql, $Cnx);
        While ($Rst = mysql_fetch_array($Consulta))
        {
            Echo "                <td><div class='divOpcion2' style='color:#ffcc00;'><br><br>" . $Rst["Tipo_Genero"] . "<br><a
style='color:#ffffff;' href='GeneroMusicalDetalle.php?IdGenero=" . $Rst["Id_Genero"] . "'>Detalle</a></div></td></div>";
            If ($Ren % 3 == 0)
            {
                Echo "                </tr><tr>";
            }
            $Ren = $Ren + 1;
        }
        Echo "                </tr>";
        Echo "            </table>
                <a style='color:#ffffff; text-align:right; font-size:8pt; font-weight:normal; text-variant:small-caps;' href =
'Main.php'>Regresar</a>
                </div>
                <div class='divBottomComplete'>
                </div>
            </center>
        </body>
    </html>";
    }
}
Else
{
    Echo "Se ha producido un error al intentar conectar con la Base de datos";
}
```

Pantalla del **Catálogo de Grupos Musicales** (Usuarios Visitantes)

**MH**  
MUSIC HOME

**CATÁLOGO DE GRUPOS MUSICALES**

<b>BANDA ALAKRAN</b> GENERO: BANDA <a href="#">DETALLE</a>	<b>BANDA RODEO</b> GENERO: BANDA <a href="#">DETALLE</a>	<b>MARIACHI SANTA CECILIA</b> GENERO: MARIACHI <a href="#">DETALLE</a>
<b>MARIACHI AMANECER RANCHERO</b> GENERO: MARIACHI <a href="#">DETALLE</a>	<b>OXIDO</b> GENERO: ROCK <a href="#">DETALLE</a>	<b>GASOLINA</b> GENERO: ROCK <a href="#">DETALLE</a>
<b>LOS CANTANTES</b> GENERO: SALSA <a href="#">DETALLE</a>	<b>SALSA RIKA</b> GENERO: SALSA <a href="#">DETALLE</a>	<b>VALIOZZO NORTE</b> GENERO: NORTEÑO <a href="#">DETALLE</a>
<b>PESADO</b> GENERO: NORTEÑO <a href="#">DETALLE</a>	<b>GRUPO TRIPULACION</b> GENERO: VERSATIL <a href="#">DETALLE</a>	<b>GRUPO ILUSION</b> GENERO: VERSATIL <a href="#">DETALLE</a>
<b>TE DE TILA</b> GENERO: POP <a href="#">DETALLE</a>	<b>POPSTARS</b> GENERO: POP <a href="#">DETALLE</a>	<b>VENGANZA DE MÉXICO</b> GENERO: DURANGUENSE <a href="#">DETALLE</a>
<b>AK7</b> GENERO: DURANGUENSE <a href="#">DETALLE</a>	<b>PRUEBA1</b> GENERO: DURANGUENSE <a href="#">DETALLE</a>	<b>PRUEBA3</b> GENERO: DURANGUENSE <a href="#">DETALLE</a>

[REGRESAR](#)

El código para la Pantalla del **Catálogo de Grupos Musicales** es el siguiente:

```
<?php
$cnx      = mysql_connect("localhost", "devuser", "devuser");
$db       = mysql_select_db("agenciamusical", $cnx);
if ($cnx)
{
    if ($db)
    {
        echo "<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'";
        echo "'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'";
        echo "<html xmlns='http://www.w3.org/1999/xhtml'";
        echo "<head";
        echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1' />";
        echo "<title>Music Home: (Grupos Musicales)</title>";
        echo "<link rel = 'stylesheet' type = 'text/css' href = 'Estilos/Estilos.css' />";
        echo "</head>";
        echo "<body bgcolor = '#FFFFFF' alink = '#FFFFFF' vlink = '#FFFFFF'>";
        echo "<center>
        <div class='divTopComplete'>
        </div>
        <div class='divCenterComplete' style='font-weight:bold'>Catálogo de Grupos Musicales
        <table>
        <tr>";

        $ren = 1;
        $qrySql = "SELECT Id_Grupo, Nombre, Genero_Musical FROM Grupo_Musical ORDER BY Id_Grupo ASC";
        $consulta = mysql_query($qrySql, $cnx);
        while ($rst = mysql_fetch_array($consulta))
        {
            echo "        <td><div class='divOpcion2' style='color:#ffcc00;'><br><br>" . $rst["Nombre"] . " <br>Genero: " . $rst[
"Genero_Musical"] . " <br><a style='color:#ffffff;' href='GrupoMusicalDetalle.php?IdGrupo=" . $rst["Id_Grupo"] . "'>Detalle</a></div></td>
</div>";
            if ($ren % 3 == 0)
            {
                echo "        </tr><tr>";
            }
            $ren = $ren + 1;
        }
        echo "        </tr>";
        echo "    </table>
    <a style='color:#ffffff; text-align:right; font-size:8pt; font-weight:normal; text-variant:small-caps;' href =
'Main.php'>Regresar</a>
    </div>
    <div class='divBottomComplete'>
    </div>
    </center>
    </body>
    </html>";
    }
}
else
{
```

Pantalla de **Apartar Evento** (Usuarios Visitantes)

**MH**  
MUSIC HOME

**APARTAR EVENTO**

NOMBRE:	<input type="text"/>	APELLIDO PATERNO:	<input type="text"/>
APELLIDO MATERNO:	<input type="text"/>	DOMICILIO:	<input type="text"/>
TELÉFONO PARTICULAR:	<input type="text"/>	CELULAR:	<input type="text"/>
E-MAIL:	<input type="text"/>	FECHA DE EVENTO:	<input type="text"/>
LUGAR DE EVENTO:	<input type="text"/>	HORA DE EVENTO:	<input type="text"/>
LOCALIDAD/PROVINCIA DE EVENTO:	<input type="text"/>	ESTIMADO DE ASISTENTES:	<input type="text"/>
GRUPO A CONTRATAR:	<input type="text"/>	NOTAS:	<input type="text"/>

[REGRESAR](#)



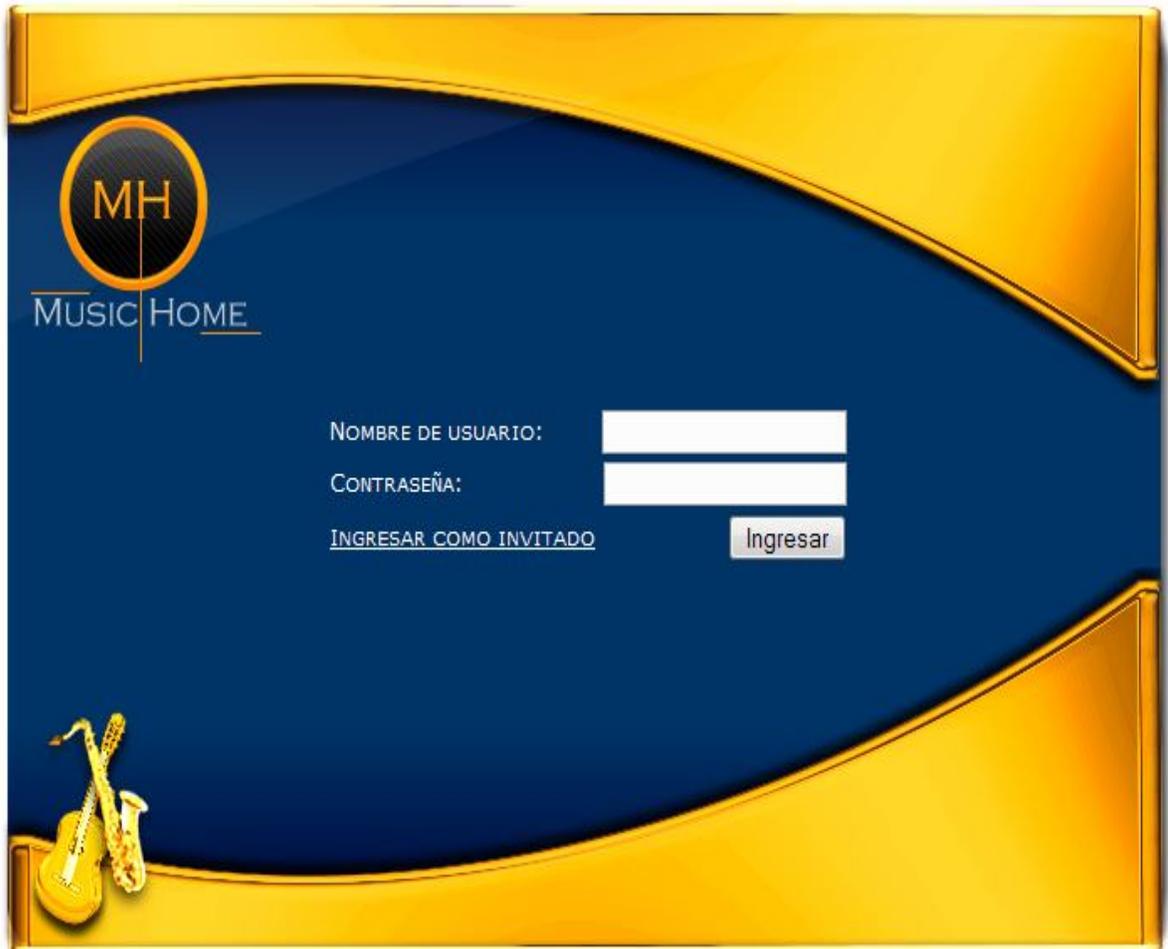
El código para la Pantalla de **Apartar Evento** es el siguiente:

**<?php**

```
$Cnx      = mysql_connect("localhost", "devuser", "devuser");
$Bd      = mysql_select_db("agenciamusical", $Cnx);

Echo "<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN";
Echo "'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'";
Echo "<html xmlns='http://www.w3.org/1999/xhtml'>";
Echo "<head>";
Echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1' />";
Echo "<title>Music Home: " . $Rst["Nombre"] . " (Apartar Evento)</title>";
Echo "<link rel = 'stylesheet' type = 'text/css' href = 'Estilos/Estilos.css' />";
Echo "</head>";
Echo "<body bgcolor = '#FFFFFF'>";
Echo "<center>
    <div class='divTopComplete'>
    </div>
    <div class='divCenterComplete' style='font-weight:bold'>Apartar Evento<br>
    <form name = 'frmDatos' action = 'ApartaEvento.php' method = 'POST'>
    <table>
    <tr>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:left'>
        Nombre:
    </td>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:right'>
        <input type = 'Text' name = 'txtNombre' style = 'width:100px;'>
    </td>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:left'>
        Apellido Paterno:
    </td>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:right'>
        <input type = 'Text' name = 'txtApPaterno' style = 'width:100px;'>
    </td>
    </tr>
    <tr>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:left'>
        Apellido Materno:
    </td>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:right'>
        <input type = 'Text' name = 'txtApMaterno' style = 'width:100px;'>
    </td>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:left'>
        Domicilio:
    </td>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:right'>
        <input type = 'Text' name = 'txtDomicilio' style = 'width:100px;'>
    </td>
    </tr>
    <tr>
    <td style = 'font-size:7pt; font-weight:bold; color:#ffffff; text-variant:small-caps; text-align:left'>
```

Pantalla de **Login** (Sólo para usuarios Administradores)



MH  
MUSIC HOME

NOMBRE DE USUARIO:

CONTRASEÑA:

[INGRESAR COMO INVITADO](#)





Pantalla del **Menú Principal** (Sólo para usuarios Administradores)



El código para la Pantalla del **Menú Principal** (Sólo para usuarios Administradores) es el siguiente:

```
<?php
If ((isset($_POST["txtUsuario"]) && $_POST["txtUsuario"] != "") && (isset($_POST["txtPassword"]) && $_POST["txtPassword"] != ""))
{
    $Usr      = $_POST["txtUsuario"];
    $Pwd      = $_POST["txtPassword"];
    $TipoAcceso = "A";
}
Else
{
    $Usr      = $_COOKIE['CookieSession'];
    $Pwd      = $_COOKIE['CookieSession2'];
    $TipoAcceso = $_COOKIE['CookieSession3'];
}

$Cnx      = mysql_connect("localhost", "devuser", "devuser");
$Bd       = mysql_select_db("agenciamusical", $Cnx);
$QrySql   = "SELECT Nombre, Password, TipoAcceso FROM Administrador WHERE Nombre = '";
$QrySql   = $QrySql . $Usr . "' AND Password = ' . $Pwd . "' AND TipoAcceso = ' . $TipoAcceso . "'";
$Consulta = mysql_query($QrySql, $Cnx);
$Rst      = mysql_fetch_array($Consulta);

If ($Cnx)
{
    If ($Bd)
    {
        If ($Rst["Nombre"])
        {
            setcookie("CookieSession", $Usr, time()+3600, "/", "");
            setcookie("CookieSession2", $Pwd, time()+3600, "/", "");
            setcookie("CookieSession3", $TipoAcceso, time()+3600, "/", "");

            Echo "<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'";
            Echo "'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>";
            Echo "<html xmlns='http://www.w3.org/1999/xhtml'>";
            Echo "<head>";
            Echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1' />";
            Echo "<title>Music Home: " . $Rst["Nombre"] . "</title>";
            Echo "<link rel = 'stylesheet' type = 'text/css' href = 'Estilos/Estilos.css' />";
            Echo "</head>";
            Echo "<body bgcolor = '#FFFFFF'>";
            Echo "<center>
                <div class='divTopComplete'>";
            If ($Rst["TipoAcceso"] == 'A')
            {
                Echo "      <br>
                <br>
                <br>
                <br>";
            }
        }
    }
}
}
```

Pantalla del **Catálogo de Eventos Sociales** (Sólo para usuarios Administradores)

**MH**  
MUSIC HOME

Usuario: JOSE LUIS [Cerrar Sesión](#)

**CATÁLOGO DE EVENTOS SOCIALES**

<b>BODA</b> DURACIÓN: 5 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	<b>XV AÑOS</b> DURACIÓN: 5 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	<b>BAILES</b> DURACIÓN: 2 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
<b>RODEOS</b> DURACIÓN: 2 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	<b>LUGARES CERRADOS</b> DURACIÓN: 2 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	<b>FIESTAS POPULARES</b> DURACIÓN: 5 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>
<b>PRESENTACIONES</b> DURACIÓN: 5 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	<b>OTROS</b> DURACIÓN: 5 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>	<b>PRUEBA4</b> DURACIÓN: 2 HRS. <a href="#">EDITAR</a> <a href="#">ELIMINAR</a>

[REGRESAR](#)



Pantalla sobre el **Catálogo de Géneros Musicales** (Sólo para usuarios Administradores)





Pantalla sobre el **Catálogo de Grupos Musicales** (Sólo para usuarios Administradores)

The screenshot displays the 'CATÁLOGO DE GRUPOS MUSICALES' interface. At the top left is the 'MH MUSIC HOME' logo. At the top right, it shows the user 'Usuario: JOSE LUIS' and a 'Cerrar Sesión' link. The main content is a grid of 18 group cards, each containing the group name, genre, and 'EDITAR' and 'ELIMINAR' buttons. A 'REGRESAR' link is located at the bottom center.

Group Name	Genre	Buttons
BANDA ALAKRAN	BANDA	EDITAR, ELIMINAR
BANDA RODEO	BANDA	EDITAR, ELIMINAR
MARIACHI SANTA CECILIA	MARIACHI	EDITAR, ELIMINAR
MARIACHI AMANECER RANCHERO	MARIACHI	EDITAR, ELIMINAR
OXIDO	ROCK	EDITAR, ELIMINAR
GASOLINA	ROCK	EDITAR, ELIMINAR
LOS CANTANTES	SALSA	EDITAR, ELIMINAR
SALSA RIKA	SALSA	EDITAR, ELIMINAR
VALIOZZO NORTE	NORTEÑO	EDITAR, ELIMINAR
PESADO	NORTEÑO	EDITAR, ELIMINAR
GRUPO TRIPULACION	VERSATIL	EDITAR, ELIMINAR
GRUPO ILUSION	VERSATIL	EDITAR, ELIMINAR
TE DE TILA	POP	EDITAR, ELIMINAR
POPSTARS	POP	EDITAR, ELIMINAR
VENGANZA DE MÉXICO	DURANGUENSE	EDITAR, ELIMINAR
AK7	DURANGUENSE	EDITAR, ELIMINAR
PRUEBA1	DURANGUENSE	EDITAR, ELIMINAR
PRUEBA3	DURANGUENSE	EDITAR, ELIMINAR

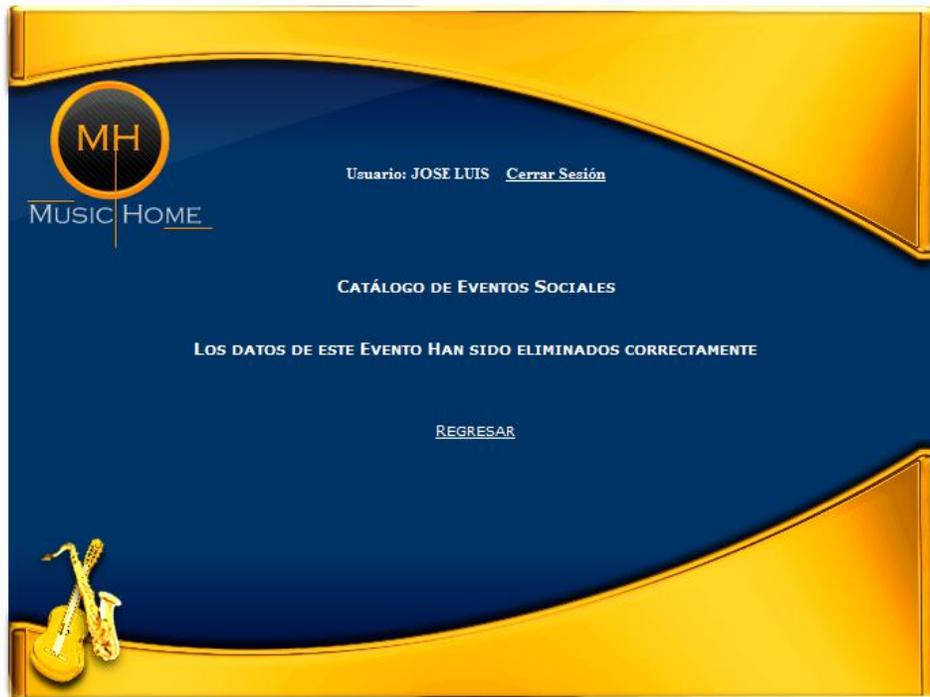
[REGRESAR](#)

El código para el **Catálogo de Grupos Musicales** (Sólo para usuarios Administradores) es el siguiente:

```
<?php
$Usuario = $_COOKIE['CookieSession'];
$Password = $_COOKIE['CookieSession2'];
$Acceso = $_COOKIE['CookieSession3'];
$Cnx = mysql_connect("localhost", "devuser", "devuser");
$Bd = mysql_select_db("agenciamusical", $Cnx);
$QrySql = "SELECT Nombre, Password, TipoAcceso FROM Administrador WHERE Nombre = ";
$QrySql = $QrySql . $Usuario . "' AND Password = '" . $Password . "' AND TipoAcceso = '" . $Acceso . "'";
$Consulta = mysql_query($QrySql, $Cnx);
$Rst = mysql_fetch_array($Consulta);

If ($Cnx)
{
    If ($Bd)
    {
        If ($Rst["Nombre"])
        {
            Echo "<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'";
            Echo "'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>";
            Echo "<html xmlns='http://www.w3.org/1999/xhtml'>";
            Echo "<head>";
            Echo "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1' />";
            Echo "<title>Music Home: (Grupos Musicales)</title>";
            Echo "<link rel = 'stylesheet' type = 'text/css' href = 'Estilos/Estilos.css' />";
            Echo "</head>";
            Echo "<body bgcolor = '#FFFFFF' alink = '#FFFFFF' vlink = '#FFFFFF'>";
            Echo "<center>
                <div class='divTopComplete'>";
            If ($Rst["TipoAcceso"] == 'A')
            {
                Echo "
                    <br>
                    <span style = 'color:#ffffff; font-weight:bold;'>Usuario: " . $Rst["Nombre"];
                Echo "
                    &nbsp; &nbsp; &nbsp; <a style = 'color:#ffffff;' href = 'Admin.php'>Cerrar Sesión</a></span>
                </div>
                <div class='divCenterComplete' style='font-weight:bold;'>Catálogo de Grupos Musicales";
                Echo "
                    <table>";
                Echo "
                    <tr>";
                $Ren = 1;
                $QrySql = "SELECT Id_Grupo, Nombre, Genero_Musical FROM Grupo_Musical ORDER BY Id_Grupo ASC";
                $Consulta = mysql_query($QrySql, $Cnx);
                While ($Rst = mysql_fetch_array($Consulta))
```

Pantalla para **Eliminar Eventos Sociales** (Sólo para usuarios Administradores)





## Conclusiones

Dado a los conocimientos previos que se adquirieron en el periodo que comprendió la carrera de Ingeniería en Computación y gracias al 6º. Diplomado de Desarrollo de Sistemas en Web, se pudieron alcanzar varios objetivos sobre este proyecto, así mismo gracias a este seminario se logró desarrollarse más profesionalmente en el ámbito laboral.

Fue de gran ayuda conocer nuevas tecnologías que existen en nuestro mundo moderno y lo mejor de todo esto, aprender a manejarlas y realizar con ellas buenos sistemas que en la actualidad es lo principal que una empresa exige para poder llegar a estar entre las primeras del mercado, un sistema con una buena tecnología y un buen funcionamiento siempre estará en competencia con muchas otras y esto es lo que hace que la tecnología vaya evolucionando., así mismo provocan que las personas involucradas en todo el desarrollo de un sistema cada día se vaya preparando más para así cubrir todos los requerimientos que un sistema exige.

En este Sistema titulado “Agencia de Consulta y Contratación para Grupos Musicales”, involucró muchas cosas, como realizar diversas juntas con el dueño del sistema, recopilar la información necesaria de Ingeniería de Software y Bases de Datos para poder diseñar y desarrollar la base de datos para que cumpliera con los requerimientos, así mismo encontrar las herramientas apropiadas para realizar un sistema completamente funcional, que en este caso fueron Sql Server y Php y posteriormente desarrollar el sistema web, esto realmente implica mucho tiempo tanto para el análisis como para el desarrollo, pero lo más importante es que el Sistema está listo para operar y cumplir con sus objetivos.

Quiero aprovechar este espacio para agradecer al apoyo brindado por mi asesor Jesús Hernández Cabrera, el cual en todo momento me hizo las observaciones necesarias para que el Sistema Web no se desviara de sus objetivos y estos puedan lograrse en un 100%.

El Sistema está abierto a todo el público esperando pueda ser de gran ayuda para a los que está dirigido y hecho este Sistema Web.

Debemos de tomar algo muy en cuenta muchas veces tendemos a pensar que las herramientas solucionarán todos nuestros problemas y no es así. Las herramientas son solo el soporte y debemos basarnos en el hábito del trabajo. Sin método y disciplina, aunque tengamos las mejores herramientas no se cumplirán con los objetivos.

## Bibliografía

- Programación en Microsoft Sql Server 2005. Guerrero, Fernando G. Rojas, Carlos Eduardo ED. Prentice Hall.
- PHP and MySQL Web Development. By Luke Welling and Laura Thomson.
- Fundamentos y modelos de Bases de Datos. Adoración de Miguel Castaño, Mario G. Piattini Velthuis, Ed. Ra – Ma 2007.
- Manual sobre la Herramienta ERwin Data Modeler r7. Autor. Pedro Alfonso Perisquía González. 2008
- Aprendiendo UML en 24 horas, Joseph Schmuller, ED. Prentice Hall.
- Biblioteca de Consulta Microsoft ® Encarta 2008. © Microsoft Corporation. Derechos Reservados.
- Fundamentos de Programación, Ernesto Peñaloza Romero, D. R. © Universidad Nacional Autónoma de México. FES Aragón.
- MySQL/PHP DataBase Aplicattions, Jay Greenpan & Brad Bulguer, ED. M&T Books 2002, Pp. 54 – 59
- Principios de Programación en 24 horas, Greg Perry, ED. Prentice Hall.
- Web DataBase Aplicattions with PHP & SQL, Hugo E. William & David Lane, ED. O'Rilley 2002.