



**BENEMERITA UNIVERSIDAD  
AUTONOMA DE PUEBLA**

---

---

**Facultad de Ciencias de la Computación**

**SISTEMA AGENDA PARA LA ADMINISTRACION  
DE EVENTOS**

**TESIS PROFESIONAL**

QUE PARA OBTENER EL TITULO DE:

**LICENCIADO EN CIENCIAS DE LA  
COMPUTACION**

PRESENTA:

**MARI CARMEN ESPINOZA CASTELLANOS**

ASESOR:

**M.C. PEDRO BELLO LOPEZ**

Puebla, Pue.

Mayo de 2008

# **AGRADECIMIENTOS**

A Dios en primer lugar, por la oportunidad de esta maravillosa experiencia que llamamos vida y que me ha traído por este camino.

A mis Padres que han hecho posible esta oportunidad, todos los años que compartimos, gracias por el apoyo y las experiencias con las que me han formado.

A mis hermanos y hermanas por estar ahí y demostrarme que no importa que tan diferentes podemos ser, siempre trataremos de apoyarnos y darnos ánimos.

A mis amigas, sin cuya amistad mi vida no sería la misma.

A los Profesores que he tenido a lo largo de mi formación, que han contribuido a que me encuentre en este punto de mi vida.

Y a todos los que he conocido hasta ahora, porque siempre he aprendido algo de ellos.

# **INDICE**

<b>INTRODUCCION</b>	<b>4</b>
<b>CAPITULO I: MARCO TEORICO</b>	<b>5</b>
I.1 Ingenieria de Software	5
I.1.2 Modelos Genéricos del Desarrollo del Software	6
I.2 Sistema de Bases de datos	11
I.2.1 Sistema Gestor de Bases de Datos	12
I.2.2. Sistema de Bases de Datos	12
I.2.3 Modelos de datos conceptuales	13
I.2.4 Herramientas para un Sistema de Bases de datos Web	16
<b>CAPITULO II: ANALISIS DEL SISTEMA</b>	<b>18</b>
II.1 Planteamiento del problema	18
II.1.1 Especificación de Requerimientos	18
II.1.2 Antecedentes	19
II.1.3 Alcances	19
II.1.4 Metodología	20
II.1.5 Definición de Requerimientos Funcionales	20
II.1.6 Definición de Requerimientos no Funcionales	20
II.1.7 Diagrama de Casos de Uso	20
II.1.71 Casos de uso y sus descripciones	22
II.1.7.2 Subcasos de Uso	29

<b>CAPITULO III: DISEÑO DEL SISTEMA</b>	<b>40</b>
III.1 Modelo Conceptual	40
III.2. Diseño inicial de la Base de Datos	41
III.2.1 Normalización	42
III.2.2 Diseño de la Base de Datos	44
III.2.3 Diccionario de Datos	46
<b>CAPITULO IV: IMPLEMENTACION Y PRUEBAS DEL SISTEMA.</b>	<b>48</b>
IV.1 Herramientas para el Desarrollo del Sistema	48
IV.1.2 Implementación de la Base de Datos en MySQL	49
IV.2 Conexión de MySQL con PHP	51
IV.3 Interfaces del Sistema Administrador de Eventos	53
IV.3.1 Interfaces de Usuario Administrador	56
IV.3.2 Interfaces de Usuario Registrado	59
<b>CONCLUSIONES</b>	<b>62</b>
<b>PERSPECTIVAS</b>	<b>63</b>
<b>BIBLIOGRAFIA</b>	<b>64</b>

# **INTRODUCCIÓN**

Anteriormente al automatizar un sistema para llevar un mejor control de la información, se implementaba en un solo equipo con una aplicación manejada por una sola persona a la vez. Con el tiempo se llegó a desarrollar aplicaciones que trabajaran en red, pero con la limitante de tener que hacer la instalación de la aplicación específica en los equipos a trabajar.

Con el auge de Internet, cada vez mas sistemas son desarrollados para trabajar en un entorno web debido al fácil acceso que desde cualquier parte del mundo se puede hacer a un sitio en Internet tan solo contando con un navegador web.

El Objetivo General de este trabajo es desarrollar un Sistema de Bases de Datos basado en Web que permita la administración de una Agenda de Eventos y facilite la reservación de los mismos para estudiantes así como permitir la generación de constancias de asistencias principalmente a eventos académicos.

El proyecto siguió la metodología de diseño en cascada de Ingeniería de software; recabando información sobre lo necesario del sistema, realizando un diseño de base de datos relacional e implementando el sistema utilizando el Manejador de Base de datos MySQL haciendo uso de el lenguaje PHP para realizar el enlace con la base de datos y lograr una interfaz de entorno web para el manejo de la información que administra el Sistema. Finalmente se llevó a término el Sistema de Agenda de Eventos capaz de hacer una administración de eventos acontecidos o a realizarse dentro de la Facultad de Ciencias de la Computación en un entorno web que puede ser consultado desde cualquier sitio con acceso a Internet,

El contenido de la tesis se presenta de la siguiente manera:

EL CAPITULO I, señala y presenta conceptos importantes a tomar en cuenta de Ingeniería de Software así como de Bases de Datos; Manejadores de Bases de datos, en particular MySQL y del lenguaje de programación PHP.

EL CAPITULO II, presenta el análisis hecho al sistema a desarrollar, los requerimientos, alcances y Descripción de la información en Casos de Uso y Flujo de Eventos.

EL CAPITULO III, se establece el diseño del sistema; diseño conceptual de la Base de Datos, el modelo Entidad Relación, diseño lógico, normalización de la Base de Datos y su implementación en tablas en MySQL

EL CAPITULO IV, contiene la implementación del sistema utilizando PHP conectado a la Base de Datos diseñada en MySQL e interfaces con las cuales interactuarán los usuarios.

Finalmente se establecen las conclusiones y las perspectivas del presente proyecto.

# **CAPITULO I**

## **MARCO TEORICO**

Los conceptos básicos que se utilizaron en el desarrollo del sistema se encuentran contenidos en este capítulo. Es necesario hacer una recapitulación y breve resumen de esta información que es la base sobre la que se encuentran sustentados todos los procesos que se llevan a cabo y que lleva a una mejor comprensión de lo que es el Sistema.

### **I.1 INGENIERIA DE SOFTWARE**

El Software es un elemento básico en la sociedad actual. El proceso para realizarlo conlleva toda una planeación y esfuerzo que son necesarios para, al término de todo el desarrollo, conseguir obtener un producto que satisfaga las necesidades y expectativas así como la funcionalidad esperada que cubrirá el objetivo específico para el que fue desarrollado y creado.

Para lograr el desarrollo de software se hace uso de la Ingeniería de Software que no es más que una serie de pasos a seguir para lograr realizar un software que funcione con eficiencia, seguridad y confiabilidad unos de los muchos requisitos que se espera cumpla el software desarrollado.

La Ingeniería de Software se puede definir como la aplicación práctica y sistemática de conocimientos científicos para la producción de programas correctos que se desarrollan a tiempo y dentro de las estimaciones del presupuesto y la correspondiente documentación para desarrollarlos, usarlos y mantenerlos.[1]

La ingeniería de software comprende teorías y métodos para el desarrollo profesional del software, así como el diseño, construcción y mantenimiento de sistemas grandes.

La Ingeniería de Software es el establecimiento y uso de principios de ingeniería para obtener software que sea confiable y que funcione eficientemente en máquinas reales.

La Ingeniería de Software es relativamente nueva ya que aparece a finales de los años sesenta y principios de los setenta, comenzando con las Técnicas de Programación Estructurada, incorporándolas a las fases del ciclo vital de software. La Programación Estructurada fue seguida por otros métodos estructurados de análisis y también métodos estructurados de diseño. Además, comenzaron a usarse tecnologías orientadas a objetos. En un principio la programación era lo principal dentro de la Ingeniería de Software pero ahora la ingeniería y el diseño de requisitos, y en general el proceso de software, son más importantes.

En los años noventa la gerencia de proyecto ganó interés y llegó a ser un componente importante en la Ingeniería de Software. En la década pasada, los estándares de la

Ingeniería de Software y la madurez de proceso han caracterizado la industria del software como una disciplina madura .

Al hablar de un sistema o software se habla también de un Producto de Software que no es más que un conjunto de programas que realizan procesos establecidos y la documentación asociada a dicho conjunto de software.[4]

Se distingue a los productos de software de la siguiente forma:

- **Productos genéricos:** Productos que son producidos por una organización para ser vendidos en el mercado.
- **Productos hechos a medida:** Sistemas que son desarrollados bajo pedido a un desarrollador específico.

Actualmente interesa que el producto de software que se obtenga cumpla con siguientes atributos de calidad:

- **Mantenibilidad.-** Que el software pueda cambiar con el tiempo es decir evolucione, pero sin que pierda sus especificaciones iniciales.
- **Confiabilidad.-** Que el software no cause problemas a futuro por algún falló, ya sea económicos o de otro tipo.
- **Eficiencia.-** Que el software no debe desperdiciar los recursos del sistema.
- **Usabilidad.-** El software debe contar con una interfaz de usuario adecuada y su documentación.

### I.1.2 Modelos Genéricos del Desarrollo del Software

Para el desarrollo de software y con independencia del área de aplicación, tamaño o complejidad del proyecto, cualquier sistema se encontrará al menos una de las siguientes fases genéricas:

- **Definición.** Análisis (qué debe hacer el sistema)
- **Desarrollo.** Diseño, codificación, prueba (cómo construir el sistema)
- **Mantenimiento.** (una vez construido el sistema el software es sometido a reparaciones y modificaciones cada vez que se detecte un fallo).

## MODELOS DE DESARROLLO DE SOFTWARE

### ➤ CASCADA

El modelo en cascada refleja la necesidad impuesta por la realidad de retornar con frecuencia desde una fase hacia las anteriores con la información generada al avanzar el desarrollo.

Las representaciones más habituales de este modelo son las representadas en las figuras 1.1 y 1.2.

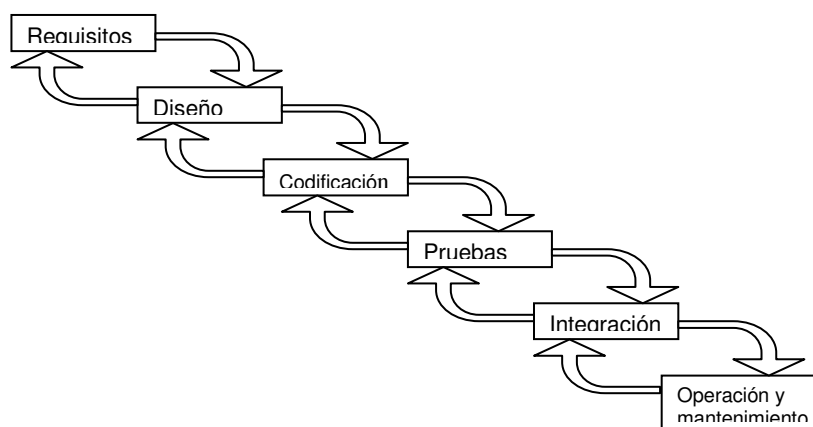


Figura 1.1. Modelo en Cascada, también llamado ciclo de vida clásico

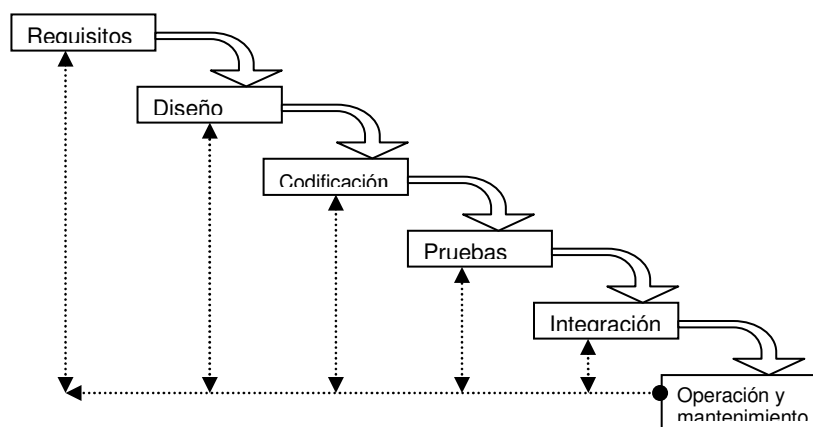


Figura 1.2 Representación del Modelo en Cascada o Modelo Clásico

La figura 1.1. parece indicar que el retorno posible se da solamente entre una fase y la anterior, mientras que en la figura 1.2. se refleja mejor el hecho de que en cualquier fase puede surgir un retorno para modificar cualquiera de las anteriores.

Este modelo, reconoce la importancia de disponer de unos requisitos y un diseño previo antes de comenzar con la codificación del sistema, pero al mismo tiempo se enfrenta al hecho de que en la realidad la dificultad que supone disponer de documentación elaborada de requisitos y diseño antes de empezar a codificar puede actuar como una barrera que bloquee el comienzo de la siguiente fase.

Por estas razones el modelo no se ha hecho muy popular, y los equipos que lo aplican pueden caer en la tentación de comenzar con el diseño o incluso con la codificación, sin tener un conocimiento suficiente de los requisitos.

Resulta apropiado para:

- Desarrollar nuevas versiones de sistemas ya veteranos en los que el desconocimiento de las necesidades de los usuarios, o del entorno de operación no plantean riesgos.
- Sistemas pequeños, sin previsión de evolución a corto plazo.

### **Fases del Modelo en Cascada**

- Análisis de requerimientos y definición.
- Diseño del sistema y del software.
- Implementación y prueba de unidades
- Integración y prueba del sistema.
- Operación y mantenimiento.
- La dificultad en esta modelo reside, en la dificultad de hacer cambios entre etapas.

### **➤ DESARROLLO EVOLUTIVO**

Dentro del Modelo de Desarrollo Evolutivo se tiene las características:

- Se adaptan más fácilmente a los cambios introducidos a lo largo del desarrollo.
- Iterativos
- En cada iteración se obtienen versiones más completas del software.

Este modelo construye una serie de grandes versiones sucesivas de un producto. El modelo evolutivo asume que los requerimientos no son completamente conocidos al inicio del proyecto. En este modelo, los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. Los desarrolladores construyen una implementación parcial del sistema que recibe sólo estos requerimientos.

El sistema es entonces desarrollado, los usuarios lo usan y proveen retroalimentación a los desarrolladores. Basada en esta retroalimentación, la especificación de requerimientos es actualizada y una segunda versión del producto es desarrollada y desplegada. El proceso se repite indefinidamente.

El desarrollo evolutivo no demanda una forma específica de observar el desarrollo de algún incremento. Todo lo que uno tiene que hacer es construir un subconjunto de requerimientos conocidos (incremental), y comprender al principio que muchos nuevos requerimientos es probable que aparezcan cuando el sistema sea desplegado o desarrollado.

El desarrollo de software en forma evolutiva requiere un especial cuidado en la manipulación de documentos, programas, datos, etc., desarrollados para distintas versiones del software. Cada paso debe ser registrado, la documentación debe ser recuperada con facilidad, los cambios deben ser efectuados de una manera controlada.

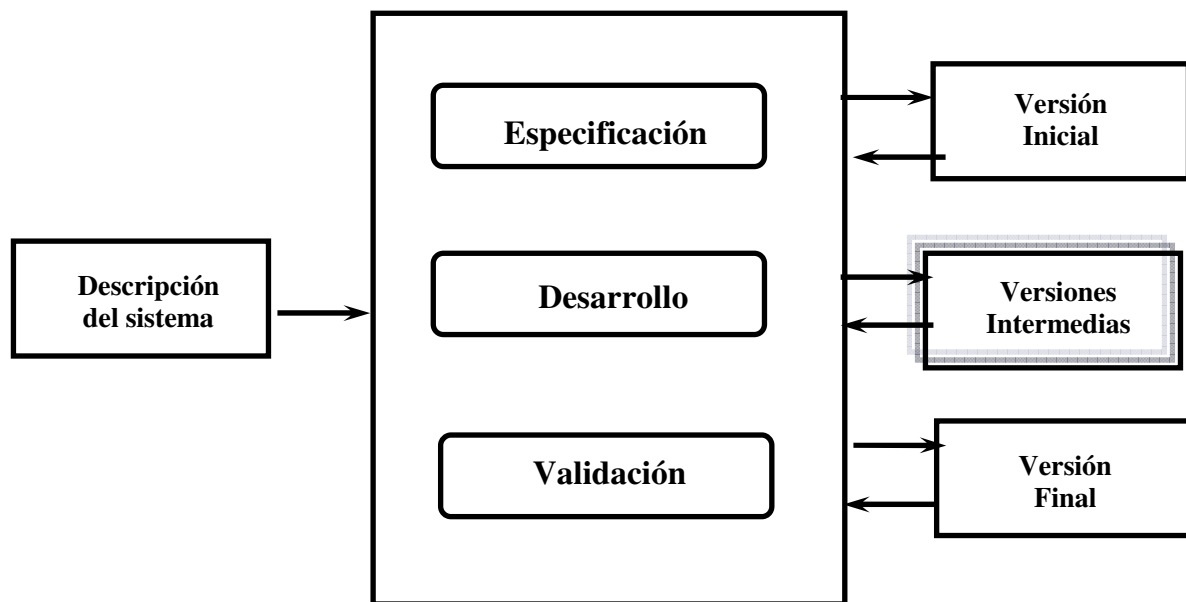


Figura 1.3 Esquema General de un Modelo Evolutivo

Considerados modelos evolutivos tenemos:

- Modelo Incremental
- Modelo en Espiral
- Modelo de Desarrollo Basado en Componentes
- Modelo WINWIN
- Modelo de Desarrollo Concurrente

## ➤ **PROTOTIPADO**

El prototipado consiste en la construcción de modelos de prueba, que simulen el funcionamiento que se pretende conseguir en el sistema.[1]

Los prototipos pueden ser:

- Ligeros: dibujos de pantallas de interfaz con simulación de funcionamiento por enlaces a otros dibujos...
- Operativos: Módulos de software con funcionamiento propio que se desarrollan sin cubrir las funcionalidades completas del sistema, normalmente en entornos RAD (rapid application development).

Esta forma de trabajo previo suele tener como principal objetivo la experimentación con un entorno similar al pretendido, para obtener retro-información del usuario o cliente que ayuda a los desarrolladores en la concreción de los requisitos.

Aunque ofrece muchas ventajas, deben conocerse los riesgos que implica el uso de prototipado:

- Como puede parecer que se ha desarrollado un interfaz de usuario sofisticado y elaborado, el cliente puede llegar a pensar que ya se ha realizado el grueso del trabajo.
- Si se trata de un prototipo operativo, puede empezar a crecer al margen de la planificación, más allá de los objetivos previstos, desbordando agendas y recursos.
- Si se trata de un prototipo ligero desarrollado fuera del departamento de desarrollo (ej. Marketing), puede mostrar al cliente funcionalidades no implementables.
- El prototipo puede llegar a ofrecer funcionalidades superiores a lo conseguible, por estar construido en un entorno diferente al de desarrollo, o no incluir toda la funcionalidad del sistema.

## ➤ **TRANSFORMACION FORMAL**

Un modelo matemático del sistema se transforma formalmente en la implementación. Esta aproximación involucra el desarrollo de una especificación formal del sistema de software y la transformación de dicha especificación, mediante transformaciones que preservan la correctitud, en un programa

El Método de Transformación formal o Métodos formales tiene las siguientes características:

- Usan notación rigurosa.
- Especificaciones sin ambigüedades.
- Útiles para sistemas críticos.
- Demostraciones formales de propiedades.
- Dificulta validación con cliente, combinación con otras técnicas semi-formales.

- Alto nivel de experticia en lógica y matemática.
- La naturaleza formal de la derivación puede proveer una forma de verificación matemático/lógica que demuestra que un paso es la correcta transformación del anterior.
- Cada paso, denominado *refinamiento*, disminuye el nivel de abstracción del modelo hasta llegar a una descripción ejecutable
- También es posible volver hacia atrás para obtener una implementación más eficiente

### ➤ **DESARROLLO BASADO EN REUTILIZACION**

El sistema es ensamblado a partir de componentes existentes. Esta técnica supone que los sistemas en su mayoría, están constituidos por componentes existentes. El proceso de desarrollo del sistema es más un ensamble que una creación.

## **I.2 SISTEMA DE BASES DE DATOS**

Un Sistema de Bases de Datos está conformado por los siguientes componentes:

- Una Base de Datos (BD) es un conjunto de datos relacionados entre sí que tienen un significado implícito es decir que pertenecen al mismo contexto y que están almacenados sistemáticamente para un posterior uso. Una base de datos debe representar algún aspecto del mundo real ser un conjunto de datos lógicamente coherente y con cierto significado inherente; por lo que se puede decir que toda base de datos se diseña y construye con datos para un propósito específico dirigida a un grupo de usuarios y que tiene ciertas aplicaciones que interesan a dicho conjunto de usuarios
- Un Sistema de Gestión de Bases de datos (SGBD): es un conjunto de programas que permite a los usuarios crear y mantener una base de datos. Por lo que un SGBD es un sistema de software de propósito general que facilita todo el proceso de definir, construir y manipular datos para diversas aplicaciones. Es un programa que proporciona la interfaz entre los datos de bajo nivel almacenados y los programas y aplicaciones.
- Usuarios de la BD:
  - Administrador se encarga del control global de la base de datos; otros usuarios: programadores de aplicaciones;
  - Usuarios directos que usan las funciones del SGBD para consultar información;
  - usuarios indirectos interactúan con la BD haciendo uso directamente del lenguaje que proporciona el SGBD;

- Usuarios indirectos: No saben nada de la BD, excepto que existe. Interaccionan con ella a través de las aplicaciones desarrolladas por los programadores.

En resumen: al conjunto formado por la base de datos y el software necesario para manipular dicha base(SGBD) y los usuarios se le llama *Sistema de Base de Datos*. [2]

### I.2.1 Sistema Gestor de Bases de Datos

Aparte de ser el encargado de proporcionar la interfaz necesaria entre los datos de bajo nivel almacenados en la base de datos y los programas o aplicaciones, tiene las siguientes funciones:

- Mantener la integridad. Reglas sobre los datos que han de ser admitidos como válidos para ser almacenados.
- Mantener la seguridad: control de acceso a los datos establecidos ciertos criterios.
- Permitir las copias de seguridad: respaldo a la información dentro de la base de datos.
- Controlar la concurrencia: Evitar conflictos entre los datos al ocurrir diferentes accesos a la base de datos que pongan en riesgo la actualización correcta de los datos.
- Suministrar Lenguajes de:
  - **Manipulación(DML):** utilizado para escribir programas que crean, actualizan y extraen información de la base de datos.
  - **Definición (DDL)** de datos: Utilizado para describir todas las estructuras reinformación y los programas que se usan para construir, actualizar e introducir la información que contiene una base de datos.
- Suministran **independencia de los datos**, en el sentido de que, a pesar de la evolución del esquema de los datos, las aplicaciones deben sufrir las mínimas modificaciones imprescindibles. [8]

### I.2.2. Sistema de Base de Datos.

Existen los siguientes niveles de abstracción en un SBD:

- **Nivel de Vista:** En este nivel se muestran visiones parciales de los datos a la que los usuarios tienen acceso y no a la totalidad que reencontraría representada en un esquema conceptual de la base de datos dentro del nivel lógico.
- **Nivel Lógico:** Datos organizados respecto a un modelo de datos adecuado tomando en cuenta una visión del mundo. Describe la estructura de toda la base de datos. Se trata con entidades u objetos que representan a sus equivalentes en el mundo real. En este nivel es donde se asocia el esquema conceptual necesario para poder hacer la implementación de una base de datos.

- **Nivel Físico:**  
Es el nivel en el que los datos son almacenados, tiene un esquema interno, que describe la estructura física de almacenamiento de la base de datos. El esquema interno emplea un modelo físico de los datos y describe todos los detalles para su almacenamiento, así como los caminos de acceso para la base de datos.

### I.2.3 Modelos de datos conceptuales

En el modelado de datos se utilizan una serie de herramientas conceptuales para realizar un esquema conceptual con una descripción concisa de los requerimientos de información de los usuarios, y contiene descripciones detalladas de los tipos de datos, los vínculos y las restricciones; éstas se expresan mediante los conceptos del modelo de datos de alto nivel.

Para el modelado de datos se tienen tres etapas de diseño:

- **Diseño Conceptual:** En el cual se parte de las especificaciones de requisitos del usuario. Es una descripción de alta nivel de la Base de Datos independiente del SGBD.
- **Diseño Lógico:** Se parte del diseño conceptual. Es una descripción de la BD que puede procesar un tipo específico de SGBD.
  - **Modelo Entidad-Relación.** Está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Este modelo distingue los siguientes elementos: *Entidad* (cosa u objeto abstracto que existe en la realidad y acerca del cual queremos almacenar información), *Interrelación* (Es la asociación, vinculación o correspondencia entre las diferentes entidades), *Dominio* y *Valor* (conjunto de posibles valores que puede tomar un tipo de entidad o interrelación), *Atributo* (cada una de las propiedades o características que tiene un tipo de entidad o un tipo de interrelación), *Claves* (es un conjunto de uno o más atributos que permiten identificar de forma única una entidad en el conjunto de entidades.[2])
  - **Uso de Teoría de la Normalización.** La normalización de datos es un proceso durante el cual los esquemas de relación insatisfactorios se descomponen repartiendo sus atributos entre esquemas de relación más pequeños que poseen propiedades deseables. El objetivo principal de la normalización es eliminar en lo posible las anomalías de actualización y las redundancias, valiéndose de, si es necesario, de todas las Formas Normales (1FN, 2FN, 3FN,4FN y 5FN) [14]
- **Diseño Físico.** Parte del diseño lógico. Es una descripción de la implementación de la BD en la memoria secundaria y depende enteramente del tipo de SGBD a utilizar.

Los modelos de datos son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

## **Base de datos jerárquicas**

En este modelo, se toma en cuenta que la información es almacenada en una estructura jerárquica. Los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

## **Base de datos de red.**

Este es un modelo ligeramente distinto del jerárquico; su diferencia radica en la modificación del nodo: se permite que un mismo nodo tenga varios padres.

Fue una gran mejora respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos;

## **Base de datos relacional**

El modelo más actualizado para modelar problemas reales y administrar datos dinámicamente. El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL (Structured Query Language o Lenguaje estructurado de consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales).

La estructura fundamental del modelo relacional es precisamente **la relación**, es decir una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos).

## **Base de datos Orientada a Objetos**

Este modelo es bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento) Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes de paradigma a objetos:

- Encapsulación
- Herencia
- Poliformismo

## **Base de datos distribuidas**

Las bases de datos distribuidas se están utilizando cada vez más en la medida en que se usan las arquitecturas de cliente-servidor. Los principales problemas que se generan por el uso de la tecnología de bases de datos distribuidas son en lo referente a la duplicidad de datos y a su integridad al momento de realizar actualizaciones a los mismos.

## Bases de Datos Cliente Servidor.

Con el Desarrollo de Internet el uso de Sistemas de Bases de Datos Web es ya habitual, por lo que uno de los esquemas más utilizados es el Cliente-Servidor que en el caso más simple corresponde a la situación en la cual una aplicación (el cliente) solicita un recurso y otra (el servidor) la atiende para brindarle el servicio de ser posible.

Para un Sistema de Bases de Datos Web en el que solo se tiene un servidor que atienda dichas solicitudes se dice que es un *Sistema Dorsal o Arquitectura Centralizada* (ver Figura 1.4) en el que hay un solo servidor y varios clientes, en el servidor se encuentra instalado el SGBD (Manejador o Sistema de Gestión de Bases de Datos) y residen las BD para diversas aplicaciones. Los clientes mediante su poder local de cómputo, presentan las interfases a los usuarios y preparan las transacciones solicitadas, las cuales son enviadas al servidor para su procesamiento. Se tiene la ventaja de repartir el trabajo en partes, lo cual mejora el tiempo de respuesta y reduce la latencia en las comunicaciones.[3]

- a. El servidor puede ser una máquina especializada de alto rendimiento para el manejo de BD y un sistema operativo estable multitarea, multiusuario y con seguridad.
- b. El Cliente puede ser una estación de trabajo personal (PC).

De esta forma varias máquinas cliente podrán acceder a la misma máquina que juega el papel de servidor. Por lo cual una base de datos puede ser compartida entre distintos clientes.

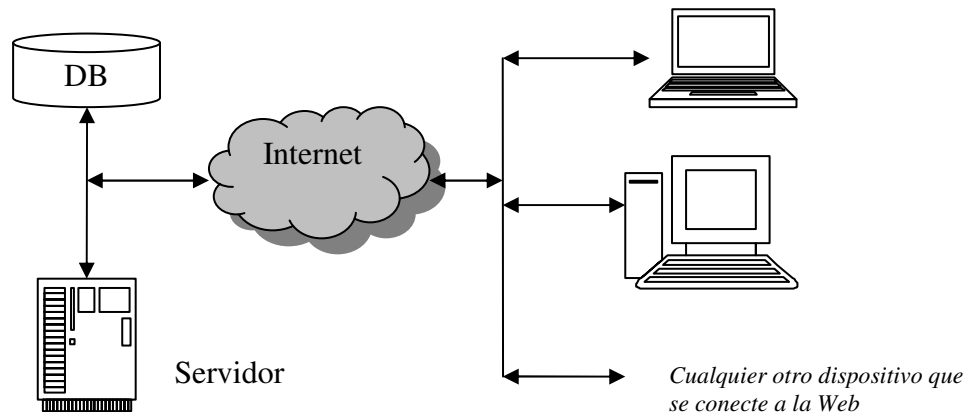


Figura 1.4 Esquema de Arquitectura Centralizada

Es notable que el servidor de Web debe ser multitarea para poder atender a varios clientes a la vez por un lado, por otro el servidor de Web debe contener mecanismos que permitan interpretar la solicitud como algo más que una simple página de web, es decir el servidor de web debe ser capaz de ejecutar programas en uno o varios lenguajes, los cuales mediante bibliotecas especiales y dedicadas se han de comunicar con el DBMS para que éste realice

la tarea de BD solicitada y finalmente el proceso, una vez hecha la tarea, se revierte para producir la respuesta y enviarla al cliente solicitante a través de la red.[8]

## **I.2.4 Herramientas para un Sistema de Bases de datos Web**

Para la creación y mantenimiento de un Sistema de Base de datos Web se pueden tener muchas herramientas de software dependiendo de la plataforma sobre la que se ha de trabajar.

Para el presente proyecto se eligió MySQL como Gestor de Bases de datos tipo SQL y PHP como lenguaje de programación.

En cuanto a Gestores de Bases de datos (SGBD) de tipo SQL podemos encontrar entre los más conocidos a MySQL y ASP. Aunque también existen otros como DB2, Firebird, Informix, Interbase, Oracle, SQL Server, Cold Fusion etc.

### **MySQL**

Dentro de los SGBD que permiten la creación de aplicaciones cliente servidor es uno de los más extendidos por ofrecer una solución integral, es un sistema con seguridad y opera bajo el modelo Open Source (de código abierto). Esto hace del manejador una opción interesante para la creación de aplicación cliente-servidor sobre Internet. La documentación pública y privada sobre el DBMS es amplia y existe soporte para el producto. Existen versiones para diversas plataformas y sistemas operativos (Windows, Linux, Mac OS X, varias versiones de UNIX y Netware, distribuidas como binarios o en código fuente).[3]

Los procedimientos de instalación y configuración son claros y generalmente si la plataforma cumple los requerimientos solicitados el sistema es funcional y estable.

Así como es necesario un software SGBD para el manejo de una BD, así también es necesario un lenguaje de programación para el desarrollo de aplicaciones en el lado del servidor sobre la web. Existen muchos lenguajes de este tipo, por ejemplo: PHP, Perl, Visual Basic.Net, Visual Java ++, Python, C#, etc.[13]

### **PHP**

Es uno de los lenguajes de servidor más extendidos en la web, es Open Source (código abierto), esta muy bien documentado, ligero, multiplataforma y relativamente simple de utilizar. PHP permite embeber pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados íntegramente en un lenguaje distinto al HTML.

Lo que distingue a PHP; de la tecnología Javascript por ejemplo la cual se ejecuta en la máquina cliente; es que el código PHP es ejecutado en el servidor. El cliente solamente recibe el resultado de su ejecución. El servidor Web puede ser incluso configurado para que procese todos los archivos HTML con PHP.

Aunque el desarrollo de PHP está concentrado en la programación de scripts en la parte de servidor, se puede utilizar para muchas otras tareas, en general se puede considerar un lenguaje de propósito general. Su estructura es tipo C, más no requiere de declaración de variables, para el lenguaje todas son cadenas, en caso de ser números las cadenas, éstos se pueden operar aritmética y lógicamente de manera normal. Tiene manejo de archivos locales, se pueden declarar funciones, es posible utilizar el paradigma de programación orientada a objetos y crear clases.[10]

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS etc. PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape y iPlanet, Orelly Website Pro Server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, Puede usarse como procesador CGI.

Con PHP no se está limitado a resultados en HTML. Entre otras habilidades de PHP se incluyen, creación de imágenes, archivos PDF y películas flash. También pueden presentar otros resultados, como XHTML y archivos XML. PHP puede generar estos archivos y grabarlos en el sistema de archivos en vez de presentarlos en la pantalla.[5]

Quizás la característica más potente y destacable de PHP es su soporte para una cantidad de bases de datos. Escribir una interfaz vía Web para una base de datos es una tarea simple con PHP.

Este lenguaje ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales y por ODBC a todas las bases de datos posibles en sistemas Microsoft.

En general se suele agrupar a un conjunto de subsistemas software necesarios para alcanzar una solución en el caso de configurar una base de datos en web.

Estas agrupaciones se conocen con sus acrónimos dependiendo del software que unen. Entre estas tecnologías se pueden tener por ejemplo:

- LAMP (Linux, sistema operativo; Apache, servidor web; MySQL, el gestor de base de datos; Perl, PHP o Python lenguajes de programación.)
- WAMP (Windows, sistema operativo; Apache, servidor web; MySQL, el gestor de base de datos; Perl, PHP o Python, lenguajes de programación)
- LAMDA (Linux, Apache, Mono, DB40, ActiveRecord)

Para sistemas basados en otros gestores de bases de datos se tienen otro tipo de agrupaciones por ejemplo: Windows 2003 Server, sistema operativo; Internet Information, Server, servidor Web; SQL Server 2003, gestor de base de datos; Visual Basic.Net y/o, Visual Java++, lenguajes de programación.

## **CAPITULO II**

### **ANALISIS DEL SISTEMA**

#### **II.1 Planteamiento del problema**

Actualmente y tomando en cuenta la necesidad de tener un sistema de información fácil de consultar y manejar al estar automatizado es que se propone un Sistema Web para la Facultad de Ciencias de la Computación que publique eventos existentes ya sean académicos, deportivos o anuncios de carácter general; y facilite la reservación o preinscripción a los mismos para los estudiantes y docentes de la Facultad.

Dicho sistema podrá ser manejado por tres tipos de usuarios: un administrador, un usuario registrado y usuario cliente; dependiendo del tipo de usuario serán las acciones que podrán realizar dentro del sistema; consultar eventos, dar de alta nuevos, modificación y baja de eventos existentes, creación de constancias a un evento específico; así como una administración de los usuarios que puedan tener acceso a todas estas opciones.

El Sistema sería una aplicación Web ya que de esta forma se podrá tener un mejor manejo por los distintos usuarios y de una gran disponibilidad para la comunidad de la F.C.C.

#### **II.1.1 Especificación de Requerimientos**

- En lo que sería el sistema se tendrá un manejo diferente dependiendo del usuario que ingrese ya que se tendrán tres diferentes:
  - Usuario Cliente: quien solo consultará la información sin opción a modificarla y podrá reservar o registrarse a un evento.
  - Usuario Registrado: que podrá hacer publicación de eventos y edición o cambio exclusivamente de su evento publicado, así como generar constancias a eventos realizados.
  - Administrador; que aparte de poder realizar todo lo que un usuario registrado puede hacer; y tendrá la habilidad de borrar y modificar los eventos de cualquier usuario registrado, así como gestionar los usuarios registrados: sus altas, bajas y modificaciones.
- Solo se hará una validación con un login y contraseña para los usuarios que requieran dar de alta, baja o modificación de eventos.
- Cuando se realice una consulta se podrá visualizar ya sea en una lista de eventos en orden cronológico o en una búsqueda de día, mes y año.
- Se podrá hacer consulta de eventos próximos a realizarse, y al elegir un evento hacer una preinscripción o reservación siempre y cuando dicho evento tenga la opción para ello y haya cupo.

- También se podrá consultar en una lista de orden cronológico todos los eventos que se hayan realizado.

### **II.1.2 Antecedentes**

Actualmente el proceso de promoción de eventos académicos en la Facultad se realiza mediante carteles y trípticos. Por otro lado los promotores de estos eventos no manejan un control sobre los eventos realizados durante el año. Inclusive los cupos dependen del orden en que los estudiantes y profesores lleguen al evento.

Así, con el objetivo de automatizar el proceso, surge el planteamiento de generar un Sistema Web que publique los eventos existentes y facilite la reservación de los mismos, además de que permite a los organizadores consultar los eventos tanto pasados como próximos a acontecer.

Aunque existen algunos otros sistemas de agendas similares se ha tenido en cuenta como una importante adición, la generación de constancias a eventos realizados con anterioridad a los cuales se haya hecho un registro o reservación.

### **II.1.3 Alcances**

- Un Sistema Web capaz de ser consultado que contenga la información de eventos realizados en la Facultad de Ciencias de la Computación en el que se incluirían las funcionalidades de:
  - Administración de eventos: Alta, baja y modificación de eventos.
  - Creación de Constancias de un evento específico.
  - Alta, baja y modificación de usuarios que pueden administrar eventos.

### **II.1.4 Metodología**

Utilizando la metodología del ciclo de vida clásico de la ingeniería de software y el diseño e implementación de una base de datos relacional, los pasos a considerar son:

- Elaborar la especificación de requerimientos del proyecto.
- Realizar el análisis de los requerimientos usando algunos elementos de Notación UML (Lenguaje Unificado de Modelado) dentro de todo el desarrollo del proyecto (Casos de uso, Diagramas de casos de uso, etc.)
- Diseñar la base de datos relacional.
- Diseñar la aplicación basada en Web.
- Implementar el sistema con PHP y Mysql.
- Realizar pruebas del sistema.

### **II.1.5 Definición de Requerimientos Funcionales**

- Cualquiera usuario puede acceder al sistema específicamente al área de consulta. Para hacer una alta o administración de eventos, el sistema verificará que tipo de usuario tiene acceso y dependiendo de eso se tendrán diferentes tareas que se podrán realizar.
- En el sistema se permitirán realizar las siguientes acciones dependiendo del tipo de usuario:
  - Consulta: Se tendrá la opción de realizar consulta de eventos por tipo: Académicos, Deportivos o Anuncios generales. Dentro de eventos académicos y deportivos se podrá hacer una búsqueda más especializada de eventos pasados o próximos. Para eventos próximos se podrá hacer una búsqueda por día, mes y año o una vista de todos los eventos a realizarse en orden cronológico. Para eventos pasados solo mostrará en orden cronológico todos los eventos. Al escoger algún evento dentro de la consulta, se podrá hacer un Registro o reservación a dicho evento si cuenta con esa característica.
  - Administración o alta de evento: Se podrá realizar alta, baja modificación y creación de constancias a eventos al mostrarse una lista de eventos. Dentro de esta opción se podrá hacer una Generación de constancias a un evento ya realizado dependiendo de una lista de registro o reservación.
  - Gestión de usuarios, con opción de: altas, bajas y modificaciones de usuarios registrados en el sistema. El Administrador será el único que podrá realizar estas tareas.

### **II.1.6 Definición de Requerimientos No Funcionales**

- Al ser un Sistema Web se debe tener un equipo cliente con opción para la conexión a un servidor web del cual se pueda visualizar las páginas que formen parte del sistema, si no hay conexión al servidor, no será posible acceder al sistema ya que será un acceso remoto.
- La velocidad de respuesta del sistema, dependerá de los componentes del hardware y en gran medida de la velocidad y tipo de conexión del equipo cliente y del servidor.

### **II.1.7 Diagrama de casos de Uso**

Dentro del análisis inicial realizado se llegó a determinar, usando un Modelo de Casos de uso, lo siguiente:

#### **Actores**

Representaciones de entidades externas que integran el sistema durante su ejecución:

- Usuario Registrado

- Usuario Cliente
- Administrador

### Casos de Uso

Representan o describen procesos a realizar por el sistema (Figura 2.1):

- Consulta de Eventos
- Registro o reservación a un evento
- Administración de Eventos
- Gestión de usuarios.

### Relación entre Actores y Casos de Uso

Se tiene una descripción de los servicios ofrecidos por el sistema bajo el Modelado de un Diagrama de Casos de uso, donde cada caso de uso es todo un procedimiento que se ha de llevar a cabo. En el Diagrama solo se muestra en forma general la interacción y relación de los diversos actores con los Casos de uso:

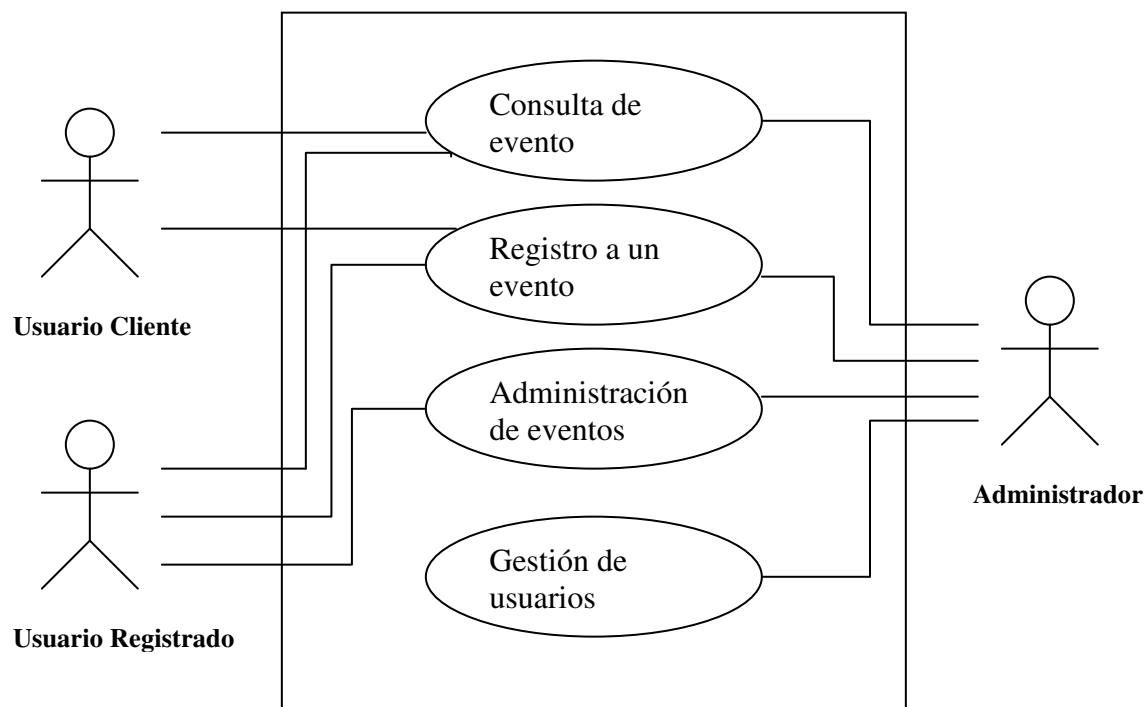


Figura 2.1 Diagrama General de Casos de Uso

Siendo una de las herramientas utilizadas los diagramas de casos de uso para el análisis del sistema y una vez determinados los principales casos de uso, se hace un análisis de las actividades que se realizarán dentro de cada uno de ellos y se realiza una serie de diagramas de subcasos de uso considerando cada uno de los casos de uso principales y los actores que intervienen en ellos.

Como el sistema tendrá diferentes acciones y opciones a seguir dependiendo del actor, se ha realizado un diagrama de subcasos de uso por cada uno de los actores que intervienen en el sistema.

## **II.1.7.1 CASOS DE USO Y SUS DESCRIPCIONES**

### **CONSULTA DE EVENTOS**

#### **Descripción**

En este caso de uso un usuario, no importa si es registrado, será capaz de consultar cualquier evento a realizar. El actor de este caso será el Usuario cliente o usuario sin registro dentro del sistema, aunque no de forma exclusiva y no excluye a los actores Usuario registrado y Administrador.

Será una de las principales funciones a realizar por el sistema.

#### **Flujo básico**

1. El usuario ingresará a Consulta de eventos.
2. El sistema le muestra las opciones de consulta: Eventos Académicos, Eventos Deportivos, y Anuncios en general. Si se escoge Eventos Académicos ir a paso 3, si se escoge Eventos Deportivos ir a paso 3 y Anuncios en general ir paso 10 .
3. Se tiene opción de elegir Eventos realizados o Eventos a realizar. Si usuario elige Eventos realizados ir a paso 4, si elige Eventos a Realizar ir a paso 7 y Compra-Venta ir al paso 9.
4. Dentro de Eventos a realizar usuario podrá escoger alguna de las siguientes subconsultas: De todo en forma cronológica ir a paso 5 y consulta por búsqueda día, mes y año ir a paso 6. Con posibilidad de regresar a la consulta del paso 2.
5. Para consultas de forma cronológica, el sistema mostrará la lista de todos los eventos realizados empezando por el último más reciente hasta el último más antiguo. Se dará al usuario la posibilidad de volver a las opciones descritas en el paso 3.
6. Usuario podrá hacer una consulta más específica introduciendo un día, mes y año; si existen eventos en esa fecha serán mostrados, de lo contrario mostrará un mensaje de no hay eventos. En esta fase se tendrá la opción a regresar a las subconsultas del paso 3.
7. Para búsqueda de eventos a realizar podrá escoger alguna de las siguientes subconsultas: De todo en forma cronológica ir a paso 8 y consulta por búsqueda día, mes y año ir a paso 9. Con posibilidad de regresar a la consulta del paso 2.

8. En consulta cronológica se hará una búsqueda semejante a la descrita en el paso 5 pero si se encuentra algún evento al cual se quiera y pueda registrar (solo académicos), se hará el correspondiente direccionamiento a Reservación de eventos (caso de uso).
9. El usuario podrá hacer una consulta semejante a la descrita en el paso 6, pero si se encuentra algún evento al cual se quiera y pueda registrar(solo académicos), se hará el correspondiente direccionamiento a Reservación de eventos (caso de uso).
10. Al consultar Anuncios Generales se mostrará todas las publicaciones en orden de la más reciente a la más antigua, con opción para regresar a las opciones del paso 2.

### **Flujos Alternativos**

- En el punto 2.  
En caso de que aún no haya ningún evento dado de alta, no mostrará ninguna de las opciones de consulta, solo el mensaje no hay eventos registrados
- En el punto 4  
En caso de que no hubiera ningún evento registrado, no habrá opciones de consulta, solo el mensaje no hay eventos registrados.
- En el punto 6  
En caso de que no hubiera ningún evento registrado, no habrá opciones de consulta, solo el mensaje no hay eventos registrados.
- En el punto 9  
En caso de que no hubiera ningún evento registrado, no habrá opciones de consulta, solo el mensaje no hay eventos registrados.

### **Precondiciones**

- El usuario ha ingresado correctamente al enlace para realizar la consulta de eventos.
- En el punto 7 si el estado del evento es activo, se mostrará su información, en caso de ser pendiente, dicho evento no aparecerá dentro de la lista resultado de la búsqueda.
- Para que un evento pase de estado pendiente a activo o realizado, debe hacerse la modificación de ese dato preciso dentro de Administración de eventos, opción disponible solo para Administrador.

### **Poscondiciones**

En caso de haber encontrado un evento al cual desea inscribirse, si la opción está dada, se hará referencia al módulo de Reservación de eventos.

## **ADMINISTRACIÓN DE EVENTOS**

### **Descripción**

En este caso de uso se tendrá acceso solo a los actores: Usuarios Registrado y

Administrador. Una vez validado alguno de estos dos usuarios se puede proceder a realizar alta, baja y modificación de un evento, así como editar lista de reservaciones a un evento si es se encuentra habilitada esta acción, como también realizar generación de constancias a un evento ya realizado, esto para los eventos dados de alta solo por el Usuario Registrado. En caso de Administrador, podrá hacer todas las anteriores acciones para cualquier evento que haya registrado dentro del sistema por cualquier Usuario Registrado

### **Flujo Básico**

1. El usuario entra al administrador de eventos.
2. Se hará una validación de usuario pidiendo login y contraseña. Si se valida como Usuario Registrado ir a paso 3, si se valida como Administrador ir a paso 15. Si no se encuentra registrado, se tendrá opción de registro para ser Usuario Registrado ir a paso 17.
3. Una vez ingresado al sistema como Usuario Registrado se tendrá las opciones de alta ir a paso 4, baja ir a paso 8, modificaciones ir a paso 6 y generación de constancias ir a paso 9 o salir de la sesión de usuario registrado ir a paso 14.
4. Para dar de alta un evento, se tendrá que escoger el tipo de evento: Académico, Cultural o Anuncio General. Ingreso de datos correspondientes al evento: Título, lugar, hora, fecha, Descripción. Con opción a activar un registro o reservación a dicho evento y un límite de reservaciones. Para los Anuncios Generales, solo se tendrán características de: Título, lugar, hora, fecha y descripción, sin opción de registro.
5. Si no se cumpliera con alguno de los campos o se ingresan datos inválidos, se pedirá rectificar los datos o con opción a cancelar la alta y volver al paso 3 en caso de Usuario Registrado, o ir a paso 15 en caso de Administrador. Si se aceptan los datos, se muestra un mensaje de alta realizada y regresa al paso 3 en caso de Usuario Registrado, o paso 15 en caso de Administrador.
6. Al ingresar a modificar se mostrarán todos los eventos dados de alta por el usuario y si se escoge un evento específico a modificar se mostrará todos los campos correspondientes al evento en modo edición para cambiar cualquier dato y una opción para aceptar los cambios ir a paso 7 o se puede regresar a cualquiera de las opciones para un eventos o regresar a Administración de eventos paso 3 en caso de Usuario Registrado, o paso 15 en caso de Administrador.
7. Al aceptar la modificación, se mostrará un mensaje de cambios realizados y opción para regresar a Administración de eventos paso 3 en caso de Usuario Registrado, o paso 15 en caso de Administrador.
8. En caso de ingresar a baja de evento, se presentará la lista de todos los eventos datos de alta por el usuario y al elegir un evento a eliminar. Si se elige eliminar un evento específico confirma la baja y regresa a Administración de eventos paso 3 en caso de Usuario Registrado, o paso 15 en caso de Administrador..
9. Al entrar a generación de constancias para un evento, se visualizará la lista de todos los eventos y su opción de constancia, al elegir un evento con la opción activa, muestra todas las personas que se han registrado y sus datos, con opción a modificar los datos de la persona paso 10, a eliminar a la persona de la lista paso 12, generar constancia de asistencia de esa persona al evento paso 13, o regresar a

Administración de evento paso 3 en caso de Usuario Registrado, o paso 15 en caso de Administrador.

10. Al ingresar a modificar datos de la persona, se tendrá en modo edición los campos correspondientes para cambiar cualquier dato y una vez cambiados muestra la confirmación ir a paso 11 o regresar a Generación de Constancias paso 9 o escoger cualquiera de las otras opciones para el usuario registrado (altas, bajas, modificaciones, constancias).
11. Al aceptar la modificación, se mostrará un mensaje de cambios realizados y opción para regresar a Generación de consultas paso 9 o escoger cualquiera de las otras opciones para el usuario registrado (altas, bajas, modificaciones, constancias).
12. En caso de elegir baja de reservación o registro se muestra confirmación de la acción. Opción de regresar a Generación de constancias paso 9 o a cualquiera de las otras opciones para usuario registrado (altas, bajas, modificaciones, constancias).
13. En caso de elegir Generar constancia de asistencia, se visualizará un formato de Constancia con el dato de la persona y opción para imprimir o guardar,
14. Al elegir Salir de regresará a la página principal del Sistema Administrador de Eventos terminando la sesión del usuario registrado (fin del caso de uso).
15. Al ingresar al sistema como Administrador se tendrá la visualización de las opciones eventos ir a paso 16, usuarios ir a paso 22 y salir 29.
16. En eventos se muestra todos los eventos dados de alta en forma cronológica con opciones de dar de baja paso 18, modificar paso 19, reservaciones y constancias ir a paso 20, dar de alta ir a paso 21. Si no hay ningún evento dado de alta solo se tendrá la opción para dar de alta ir a paso 21, usuarios ir a paso 22 y salir ir a paso 29.
17. Para registrarse por primera vez y ser Usuario Registrado, se pedirá llenar formato con datos principales: Nombre, dirección, número de teléfono, dirección de correo electrónico. Introducción de login y password a dar de alta y aceptar registro. Al aceptar el registro se habrán ingresado los datos del usuario y regresa a inicio de Administración de eventos paso 3. Si no se acepta el registro, se tendrá opción para cancelar y regresar a la página principal del Sistema.
18. Al dar de baja un evento solo aparece información de confirmación regresando al paso 16 o con opciones de usuarios 22 y salir 29
19. Al ingresar a modificar se mostrará todos los campos correspondientes al evento en modo edición para cambiar cualquier dato y una opción para aceptar los cambios o se puede regresar a cualquiera de las opciones para un evento o regresar a Administración de eventos paso 16.
20. Al entrar a generación de constancias para un evento, se visualizará la lista de todas las personas que se han registrado y sus datos, con opción a modificar los datos de la persona paso 26, a eliminar a la persona de la lista paso 27, generar constancia de asistencia de esa persona al evento paso 13, o regresar a Administración de evento paso 16. Dar de alta una reservación con direccionamiento a módulo de Reservación o Registro a un evento.
21. Para dar de alta un evento, se tendrá que escoger el tipo de evento: Académico, Cultural o Anuncio General. Ingreso de datos correspondientes al evento: Título, lugar, hora, fecha, Descripción. Con opción a activar un registro o reservación a dicho evento y un límite de reservaciones. Para los Anuncios Generales, solo se tendrán características de: Título, lugar, hora, fecha y descripción, sin opción de registro. Si no se cumpliera con alguno de los campos o se ingresan datos inválidos,

- se pedirá rectificar los datos o con opción a cancelar la alta y volver al paso 16 en Si se aceptan los datos, se muestra un mensaje de alta realizada y regresa al paso 16.
22. En usuarios se muestra todos los usuarios dados de alta con opciones de dar de baja paso 23, modificar paso 24, dar de alta un usuario nuevo ir a paso 25. Si no hay ningún usuario dado de alta solo se tendrá la opción para dar de alta ir a paso 25, eventos ir a paso 16 y salir ir a paso 29
  23. Al dar de baja un usuario solo aparece información de confirmación regresando al paso 22 o con opciones de eventos ir a paso 16 y salir 29
  24. Al ingresar a modificar se mostrará todos los campos correspondientes al usuario en modo edición para cambiar cualquier dato y una opción para aceptar los cambios y regresar a paso 22 o se puede regresar a cualquiera de las opciones para el administrador eventos ir a paso 16 y salir paso 29.
  25. Para dar de alta un usuario, se ingresarán los datos correspondientes al usuario: nombre, apellido paterno, apellido materno, calle, número colonia, estado, teléfono, email, login y contraseña. Si no se cumpliera con alguno de los campos o se ingresan datos inválidos o que el login ya existe, se pedirá rectificar los datos o con opción a cancelar la alta y volver al paso 22 Si se aceptan los datos, se muestra un mensaje de alta realizada y regresa al paso 22.
  26. Al ingresar a modificar datos de la persona, se tendrá en modo edición los campos correspondientes para cambiar cualquier dato y una vez cambiados muestra la confirmación se muestra mensaje de datos modificados y regresa a paso 20 o escoger cualquiera de las otras opciones eventos ir a paso 16 o salir ir a 29.
  27. Al aceptar la modificación, se mostrará un mensaje de cambios realizados y opción para regresar a Generación de constancias paso 20 o escoger cualquiera de las otras opciones eventos ir a paso 16 o salir ir a paso 29.
  28. En caso de elegir baja de reservación o registro se muestra confirmación de la acción. Opción de regresar a Generación de constancias paso 20 o a cualquiera de las otras opciones, eventos ir a paso 16 o salir ir a paso 29
  29. Termina la sesión del administrador y regresa a paso 1.

### **Flujos Alternativos**

- En el punto 2.  
En caso de que la validación de usuario y contraseña no se realiza por datos incorrectos, se tendrá la opción de intentarlo otra vez con opción a cancelar y salir de validación regresando a página principal del Sistema de Agenda

### **Precondiciones**

- El usuario ha ingresado correctamente al enlace para Alta y Administración de eventos.
- Al entrar a la administración de eventos, solo el Administrador puede cambiar el estado de un evento, de pendiente a activo o realizado.

### **Poscondiciones**

En caso de que al ingresar sea como Usuario Cliente, al ingresar y registrarse en el Sistema terminará siendo un Usuario Registrado.

## **REGISTRO O RESERVACIÓN A UN EVENTO**

### **Descripción**

En este caso de uso un usuario, no importa si es registrado será capaz de realizar una reservación o consulta a un evento próximo a realizarse.. El actor de este caso será el Usuario cliente o usuario sin registro dentro del sistema, aunque no de forma exclusiva y no excluye a los actores Usuario registrado y Administrador.

### **Flujo Básico**

1. El usuario ingresa a Registro o Reservación a un Evento, una vez encontrado el evento a registrarse en Consulta y elegir la opción correspondiente de Registro o Reservación. En caso de que no haya cupo (porque se ha llegado al límite de reservaciones) no se podrá realizar el registro ir a paso 4; si hay cupo ir a paso 2.
2. El sistema le indicará ciertos campos a llenar para realizar el registro o reservación: nombre, dirección, número de teléfono, correo electrónico, una vez ingresados dichos datos se procede a aceptar hacer reservación o registro.
3. Al realizarse el registro con éxito, se mostrará un anuncio de confirmación y opción para regresar a realizar una consulta de eventos o anuncios o terminar.
4. Al no haber cupo se tendrá la opción para salir de registro para regresar a realizar una consulta de eventos o anuncios o terminar.

### **Flujos Alternativos**

- En el punto 2  
En caso de que no se llenen los campos necesarios o se ingresen datos inválidos, se mostrará un mensaje de error y se dará la oportunidad para poder intentarlo de nuevo o cancelar el registro.

### **Precondiciones**

El usuario deberá hacer una consulta previa de los eventos a realizarse y haber seleccionado un evento con opción de registro.

### **Poscondiciones**

Una vez escritos sus datos y mandado la información, el usuario queda registrado a ese evento e incluido en la lista de reservaciones a la cual tiene acceso solo el Usuario Registrado que dio de alta el evento y el Administrador

## **GESTION DE USUARIOS**

### **Descripción**

En este caso de uso solo el Administrador será capaz de hacer modificación de los datos de los Usuarios Registrados dentro del sistema: altas, bajas, modificaciones. El actor de este caso, el Administrador, deberá validarse al inicio, para poder acceder a las acciones anteriores.

### **Flujo Básico**

1. Al ingresar se hará una validación del usuario requiriendo login y contraseña, al validar estos datos ir a paso 2, si no son correctos ir a paso 8.
2. Se mostrará la lista de usuarios registrados en el sistema con opciones de modificación, ir a paso 3; baja ir a paso 5; o dar de alta un usuario nuevo ir a paso 6; salir de gestión de usuarios ir a paso 7.
3. Al ingresar en modificación de un usuario en específico se mostrarán los campos correspondientes a los datos dados de alta por el Usuario Registrado en modo edición, opción de guardar las modificaciones, ir a paso 4, cancelar modificación ir a paso 2.
4. Las nuevas modificaciones son guardadas y regresa a paso 2.
5. Al elegir baja de usuario, aparece un mensaje de confirmación. Al elegir aceptar se hace la eliminación y regresa a paso 2. Si se desea, no elimina al usuario y regresa a paso 2.
6. Al entrar a dar de alta un usuario, se pedirá llenar formato con datos principales: Nombre, dirección, número de teléfono, dirección de correo electrónico, en caso de ser alumno número de matrícula, en caso de ser profesor número de empleado. Introducción de login y contraseña a dar de alta y aceptar registro. Al aceptar el registro se habrán ingresado los datos del usuario y regresa a inicio de Gestión de Usuarios paso 2. Si no se acepta el registro, se tendrá opción para cancelar y no guardar cambios regresando a principio de Gestión de Usuarios paso 2.
7. Al salir de Gestión de Usuarios se regresa a página principal de Agenda de Eventos.
8. En caso de no validar datos de Administrador, se mostrará un mensaje para introducirlos de nuevo. En caso que no se tengan se tendrá opción para salir de validación y regresar a página principal de Sistema de Agenda de Eventos.

### **Flujos Alternativos**

- En el punto 2
9. En caso de que aún no haya ningún Usuario Registrados, es decir, la lista se encuentre vacía; solo se mostrará un mensaje de no hay usuarios dados de alta y como únicas opciones, Alta de nuevo usuario (paso 6) y salir de Gestión de Usuarios a página principal de Agenda de Eventos

### **Precondiciones**

El usuario ingresado correctamente al enlace para Gestión de Usuarios .

**Poscondiciones** Ninguna.

## II.1.7.2 SUBCASOS DE USO

En base a lo anterior y tomando en cuenta que el sistema deberá la mayor parte de su funcionamiento dependiendo del usuario que haya ingresado en él; teniendo la siguiente clasificación: Usuario Cliente (Figura 2.2.), Usuario Registrado (Figura 2.3) y Administrador (Figura 2.4); se ha hecho la consideración de subdividir algunos de los principales casos de uso que contienen más procesos internos que se pueden agrupar como subcasos de uso tomando en cuenta a cada uno de los tres usuarios considerados anteriormente.

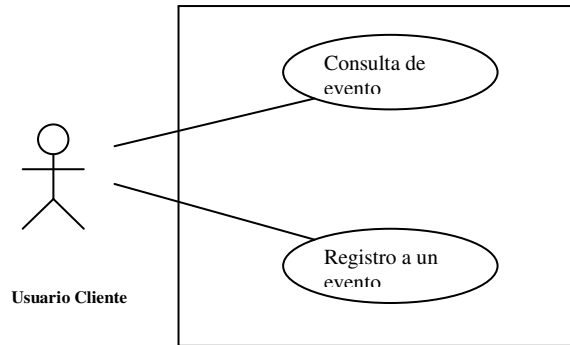


Figura 2.2 Diagrama de subcasos de Uso del Usuario Cliente

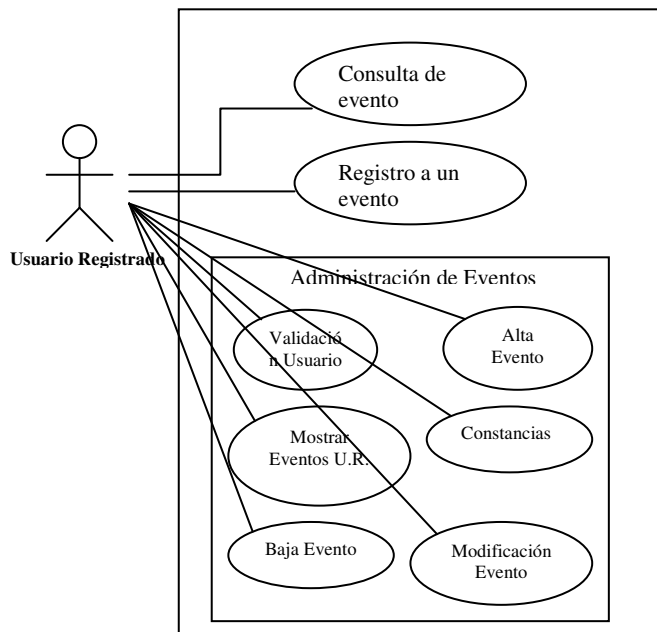


Figura 2.3 Diagrama de Subcasos de Uso Usuario Registrado

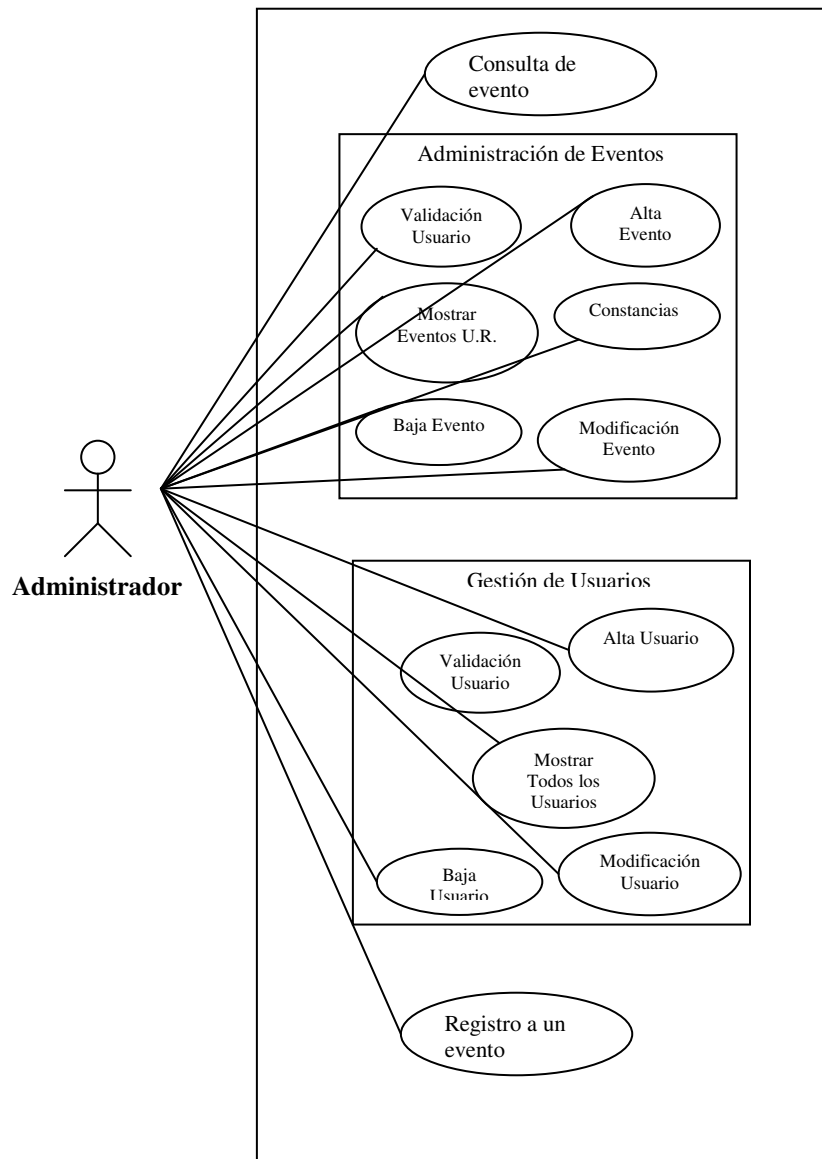


Figura 2.4 Diagrama de subcasos de uso del Administrador

## **SUBCASO DE USO VALIDACION USUARIO**

### **Descripción**

En este caso de uso se hará validación de usuario y contraseña para hacer ingreso como Usuario Registrado o como Administrador.

### **Flujo Básico**

1. El usuario al ingresar a validación se pedirá login y contraseña.
2. Si valida como Usuario Registrado o como Administrador termina validación. En caso de no estar registrado ir a paso 3. En caso cancelar validación ir a paso 4.
3. Para registrarse por primera vez y ser Usuario Registrado, se pedirá llenar formato con datos principales: Nombre, dirección, número de teléfono, dirección de correo electrónico. Introducción de login y contraseña a dar de alta y aceptar registro. Al aceptar el registro se habrán ingresado los datos del usuario y regresa a paso 1. En caso de no querer realizar registro ira paso 4.
4. Termina validación de usuario y regresa a página principal.

### **Flujos Alternativos**

- En el punto 2 en caso de no ser válidos los datos, pedirá una y otra vez por datos válidos con opción de salir de validación.

### **Precondiciones**

El Usuario debe haber ingresado previamente a Administración de Eventos o a Gestión de usuarios que es donde se tiene módulo de Validación.

### **Poscondiciones**

Una vez validado ser validado como Usuario Registrado o Administrador se puede ingresar a todas las funcionalidades asignadas a ese tipo de usuario, para Administración de Eventos o Gestión de usuarios.

## **SUBCASO DE USO ALTA DE EVENTOS**

### **Descripción**

En este caso de uso se tiene en cuenta la alta de un evento que puede ser hecha por un Usuario Registrado o el Administrador.

### **Flujo Básico**

1. Para dar de alta un evento, se tendrá que escoger el tipo de evento: Académico, Cultural o Anuncio General. Ingreso de datos correspondientes al evento: Título, lugar, hora, fecha, descripción. Con opción a activar un registro o reservación a dicho evento y un límite de reservaciones. Para los Anuncios Generales, solo se tendrán características de: Título, lugar, hora, fecha y descripción, sin opción de registro.
2. Si no se cumpliera con alguno de los campos o se ingresan datos inválidos, se pedirá rectificar los datos o con opción a cancelar la alta y volver al paso 1. Si se aceptan los datos, se muestra un mensaje de alta realizada y termina alta de evento.

### **Flujos Alternativos**

- Ninguno

### **Precondiciones**

El Usuario debe haber ingresado previamente a Administración de Eventos en donde luego de haberse validado como un Usuario Registrado o Administrador elegido la alta de un evento

### **Poscondiciones**

Al término se habrá dado de alta un evento dentro de la Agenda de Eventos.

## **SUBCASO DE USO BAJA DE EVENTOS**

### **Descripción**

En este caso de uso se tiene en cuenta la baja de un evento que puede ser hecha por un Usuario Registrado o el Administrador.

### **Flujo Básico**

1. En caso de ingresar a baja de evento, se pedirá una confirmación de baja. Si se acepta la confirmación de baja se elimina evento y va a paso 2. Si no se acepta la eliminación, no se realiza y de igual forma va a paso 2.
2. Termina baja de evento y regresa a Administración de eventos dentro del flujo normal.

### **Flujos Alternativos**

- Ninguno

**Precondiciones**

El Usuario debe haber ingresado previamente a Administración de Eventos en donde y luego haberse validado como un Usuario Registrado o Administrador y elegido la baja de un evento específico.

**Poscondiciones**

Al término de validación se habrá dado de baja el evento seleccionado dentro de la Agenda de Eventos.

**SUBCASO DE USO MODIFICACION DE EVENTO****Descripción**

En este caso de uso se tiene en cuenta la modificación de los datos de un evento dado de alta anteriormente por un Usuario Registrado o el Administrador.

**Flujo Básico**

1. Al ingresar a modificar algún evento se mostrará todos los campos correspondientes al evento en modo edición para cambiar cualquier dato y una opción para aceptar los cambios ir a paso 2 o cancelar la modificación ir a paso 3.
2. Al aceptar la modificación, se mostrará un mensaje de cambios realizados y opción para salir ir a paso 3.
3. Salir de modificación de evento y regresar a Administración de eventos flujo normal.

**Flujos Alternativos**

- En paso 1  
Al ingresar datos modificados si alguno de ellos no concuerda con el tipo de dato especificado, mostrará un mensaje de error y pedirá volver a introducir datos válidos.

**Precondiciones**

El Usuario debe haber ingresado previamente a Administración de Eventos en donde luego de haberse validado como un Usuario Registrado o Administrador elegido la modificación de evento específico.

**Poscondiciones**

Al término de modificación se habrá modificado datos de un evento seleccionado dentro de la Agenda de Eventos.

## **SUBCASO DE USO MOSTRAR EVENTOS U.R. (USUARIO REGISTRADO )**

### **Descripción**

Al entrar a Administración y Alta de Eventos, se hará una muestra de todos los eventos dados de alta por el Usuario registrado.

### **Flujo Básico**

1. Al ingresar a Administración de eventos se mostrará en una lista todos los eventos dados de alta por el Usuario Registrado en orden cronológico.
2. En caso de que Usuario Registrado no tenga eventos, se mostrará un mensaje de ningún evento.

### **Flujos Alternativos**

- En paso 2  
En caso de no haber eventos, se tendrá solo la opción de alta de un evento dentro de la administración de eventos.

### **Precondiciones**

El Usuario debe haber ingresado previamente a Administración de Eventos luego de haberse validado como un Usuario Registrado.

### **Poscondiciones**

Se tendrán opciones para dar de alta, baja, modificación o constancias para los eventos dados de alta, siempre y cuando haya algún mostrado.

## **SUBCASO DE USO MOSTRAR TODOS LOS EVENTOS**

### **Descripción**

Al entrar a Administración y Alta de Eventos, se hará una muestra de todos los eventos dados de alta por cualquier usuario dentro de la Agenda de Eventos.

### **Flujo Básico**

1. Al ingresar a Administración de eventos se mostrará en una lista todos los eventos dados de alta por cualquier Usuario Registrado en orden cronológico.
2. En caso que no haya ningún evento dado de alta aún, se mostrará un mensaje de ningún evento registrado.

## Flujos Alternativos

- En paso 2  
En caso de no haber eventos, se tendrá solo la opción de alta de un evento dentro de la administración de eventos.

## Precondiciones

El Usuario debe haber ingresado previamente a Administración de Eventos luego de haberse validado como Administrador.

## Poscondiciones

Se tendrán opciones para dar de alta, baja, modificación o constancias para los eventos dados de alta, siempre y cuando haya algún mostrado.

## SUBCASO DE USO CONSTANCIAS

### Descripción

En este caso de uso se tiene en cuenta la creación de modificación de los datos de un evento dado de alta anteriormente por un Usuario Registrado o el Administrador.

### Flujo Básico

1. Al entrar a generación de consultas para un evento, se visualizará la lista de personas que se han registrado y sus datos, con opción a modificar los datos de la persona paso 2, a eliminar a la persona de la lista paso 4, generar constancia de asistencia de esa persona al evento paso 5, o terminar Creación de constancias ir a paso 6.
2. Al ingresar a modificar datos de la persona, se tendrá en modo edición los campos correspondientes para cambiar cualquier dato y una opción para aceptar los cambios, ir a paso 3 o cancelar la modificación y regresar a Generación de Constancias paso 1.
3. Al aceptar la modificación, se mostrará un mensaje de cambios realizados y opción para regresar a Creación de constancias paso 1.
4. En caso de elegir baja de reservación o registro se pedirá una confirmación de la acción. Si se acepta la confirmación de la baja, se realiza la eliminación de la lista y regresa a Generación de constancias paso 1. Si se cancela la orden de baja no se realiza la eliminación de la lista y regresa a Generación de constancias paso 1.
5. En caso de elegir Generar constancia de asistencia, se podrá escoger uno o dos firmantes, sus nombres y puestos que se incluirán en la constancia, después se visualizará un formato de Constancia con el dato de la persona y opción para imprimir o guardar, y opción para regresar a Creación de constancias paso 1.
6. Salir de Creación de constancias y regresar a flujo normal en Administración de evento.

### **Flujos Alternativos**

- En paso 1  
Si no hubiera personas que hayan reservado para el evento, solo se mostrará un mensaje de no hay reservaciones y opción de salir de salir de creación de constancias.
- En paso 2  
En caso de no agregar datos correctos dentro de los campos al modificar datos de persona, se pedirá volver a ingresar datos hasta que se acepten o cancelar la modificación.

### **Precondiciones**

El Usuario debe haber ingresado y validado previamente como Administrador o Usuario Registrado al ingresar a Administración de Eventos.

### **Poscondiciones**

Al término de habrá creado una o varias constancias de participación, dependiendo si hubo o no personas que se hubieran registrado con anterioridad..

## **SUBCASO DE USO MOSTRAR USUARIOS REGISTRADOS**

### **Descripción**

En este caso de uso se tiene en cuenta el despliegue de todos los usuarios registrados dados de alta en el sistema solo pudiendo ser vistos por el Administrador.

### **Flujo Básico**

1. Al ingresar a Gestión de Usuarios se mostrará en una lista todos los Usuarios Registrados dados de alta dentro del sistema.
2. En caso que no haya ningún Usuario dado de alta, se mostrará un mensaje de ningún evento registrado

### **Flujos Alternativos**

- En paso 2  
En caso de no haber Usuarios, se tendrá solo la opción de alta de un Usuario dentro de Gestión de Usuarios

### **Precondiciones**

El Usuario debe haber ingresado previamente a Gestión de Usuarios y haberse validado como Administrador.

### **Poscondiciones**

Se tendrán las opciones de dar de alta, baja o modificación de información de Usuarios Registrados.

## **SUBCASO DE USO ALTA DE USUARIOS**

### **Descripción**

En este caso de uso se tiene en cuenta la alta de un nuevo usuario registrado efectuada por el Administrador.

### **Flujo Básico**

1. Al entrar a dar de alta un usuario, se pedirá llenar formato con datos principales: Nombre, dirección, número de teléfono, dirección de correo electrónico. Introducción de login y contraseña a dar de alta y aceptar registro. Al aceptar el registro ir a paso 2 si no se acepta registro termina Alta de Usuario.
2. Si no se acepta el registro, será porque los datos no se agregan correctos o ya existe el login que se requiere dar de alta, se podrá cancelar el flujo y no guardar cambios saliendo de alta de usuarios y siguiendo el flujo normal en Gestión de Usuarios. .

### **Flujos Alternativos**

- Ninguno

### **Precondiciones**

El Usuario debe haber ingresado previamente a Gestión de Usuarios y haberse validado como Administrador.

### **Poscondiciones**

Al término se habrá dado de alta un nuevo Usuario Registrado capaz de dar de alta nuevos eventos.

## **SUBCASO DE USO BAJA DE USUARIOS**

### **Descripción**

En este caso de uso se tiene en cuenta la baja de un usuario registrado efectuada por el Administrador.

### **Flujo Básico**

1. Al elegir baja de usuario, aparece un mensaje de confirmación. Al elegir aceptar se hace la eliminación ir a paso 2. Si se elige cancelar no elimina al usuario ir a paso 2.
2. Termina Baja de usuario y regresa al flujo normal en Gestión de Usuarios.

### **Flujos Alternativos**

- Ninguno

### **Precondiciones**

El Usuario debe haber ingresado previamente a Gestión de Usuarios y haberse validado como Administrador.

### **Poscondiciones**

Al término se habrá dado de baja un Usuario Registrado.

## **SUBCASO DE USO MODIFICACION DE USUARIOS**

### **Descripción**

En este caso de uso se tiene en cuenta la modificación de los datos de un usuario registrado efectuada por el Administrador.

### **Flujo Básico**

1. Al ingresar en modificación de un usuario en específico se mostrarán los campos correspondientes a los datos dados de alta por el Usuario Registrado en modo edición, opción de guardar las modificaciones, ir a paso 2, cancelar modificación ir a paso 3.
2. Las nuevas modificaciones son guardadas ir a paso 3.
3. Termina Modificación de Usuarios y regresa al flujo normal en Gestión de Usuarios.

### **Flujos Alternativos**

- En paso 1  
Al ingresar datos modificados si alguno de ellos no concuerda con el tipo de dato especificado, mostrará un mensaje de error y pedirá volver a introducir datos válidos.

**Precondiciones**

El Usuario debe haber ingresado previamente a Gestión de Usuarios y haberse validado como Administrador.

**Poscondiciones**

Al término se habrá hecho cambios en los datos de un Usuario Registrado realizados por el Administrador.

## **CAPITULO III**

### **DISEÑO DEL SISTEMA**

En el presente capítulo se tiene el diseño de la base de datos; el Modelo Conceptual que incluye las entidades sus atributos y dominios así como las relaciones entre ellas; el Diseño Lógico que agrupa el modelo relacional y la normalización a la base de datos.

#### **III. 1 MODELO CONCEPTUAL.**

Dentro del análisis realizado, el primer paso para la creación de la base de datos es producir el esquema conceptual en el que se toman en cuenta las distintas vistas que se tienen sobre la información a manejar.

Dentro del Modelo conceptual se identifica identidades, sus atributos y las relaciones que hay entre dichas entidades.

##### **Identificación de Entidades**

Las entidades que se identificaron se muestran en la tabla 3.1.

<b>ENTIDAD</b>	<b>ATRIBUTOS</b>
USUARIO	Nombre, Dirección, Teléfono, Email, Login, Password, level(de usuario: cliente, registrado, administrador)
EVENTOS	Tipo, Título, Fecha, Hora, Descripción, Registro(si/no), limite de registros
RESERVACIONES	Nombre, Dirección, Teléfono, Email, fecha de reservación, estado de la reservación( pendiente,activa)

Tabla 3.1 Entidades y atributos

##### **Identificación de Relaciones**

Las relaciones identificadas entre las entidades se describen en la tabla 3.2

<b>ENTIDAD</b>	<b>RELACION</b>	<b>CONECTIVIDAD</b>	<b>ENTIDAD</b>
USUARIO	REGISTRA	1aN	EVENTOS
EVENTO	TIENE	0aN	RESERVACIONES

Tabla 3.2 Relaciones entre las entidades

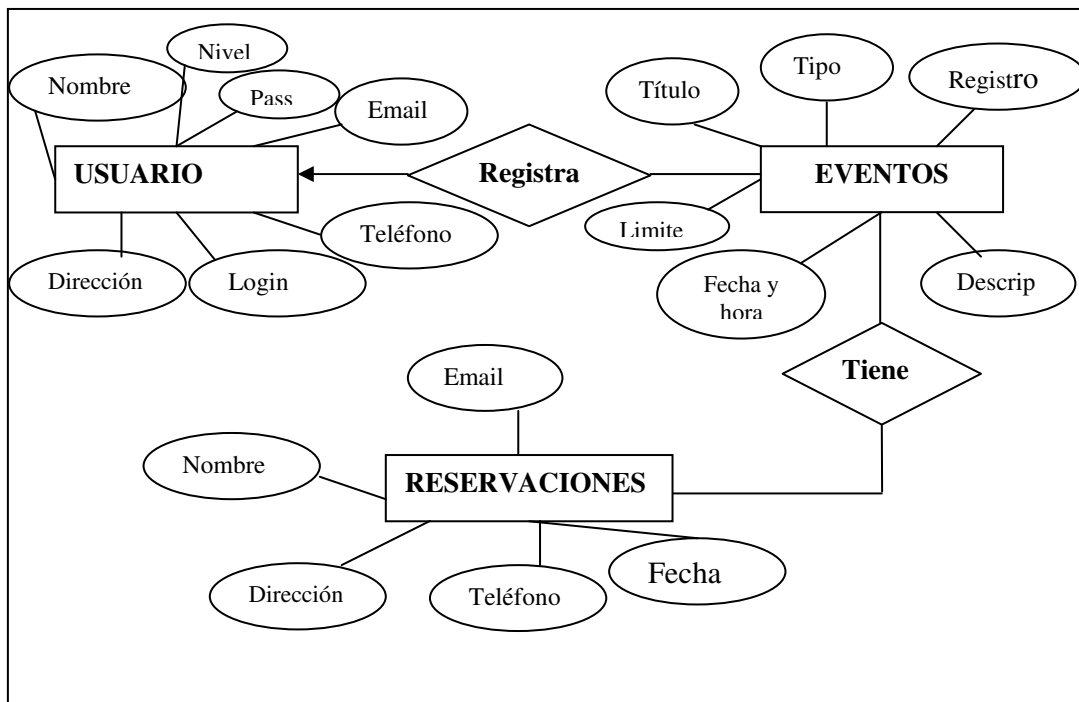


Figura 3.1 Diagrama conceptual preliminar de la Base de Datos para la Administración de Eventos

### III. 2 DISEÑO INICIAL DE LA BASE DE DATOS

Posteriormente al hacer varias reconsideraciones sobre el diseño inicial, por ejemplo: división de algunos de los atributos; en entidad Usuario el atributo dirección descompuesto en calle, colonia, número, estado, eliminación de algunos otros en el caso de la entidad Reservas; se obtiene un diseño inicial de la base de datos.

En el Diagrama Entidad-Relación, se puede apreciar que dentro de las reservas no es necesario tener los datos de la persona ya que comparte muchos de los atributos con los Usuarios Registrados, solo difiriendo en el usuario y password, ya que el nivel (level) se dará en función de si el que ingresa es cualquiera de los tres actores considerados dentro del análisis del sistema (Usuario Registrado, Usuario Cliente, Administrador)

Por lo tanto en el Diagrama Entidad-Relación (figura 3.2.) se puede identificar un supertipo el de Usuarios y dos subtipos: Usuario\_Reg (usuario registrado) y Usuario\_Reserva (Usuario Cliente)

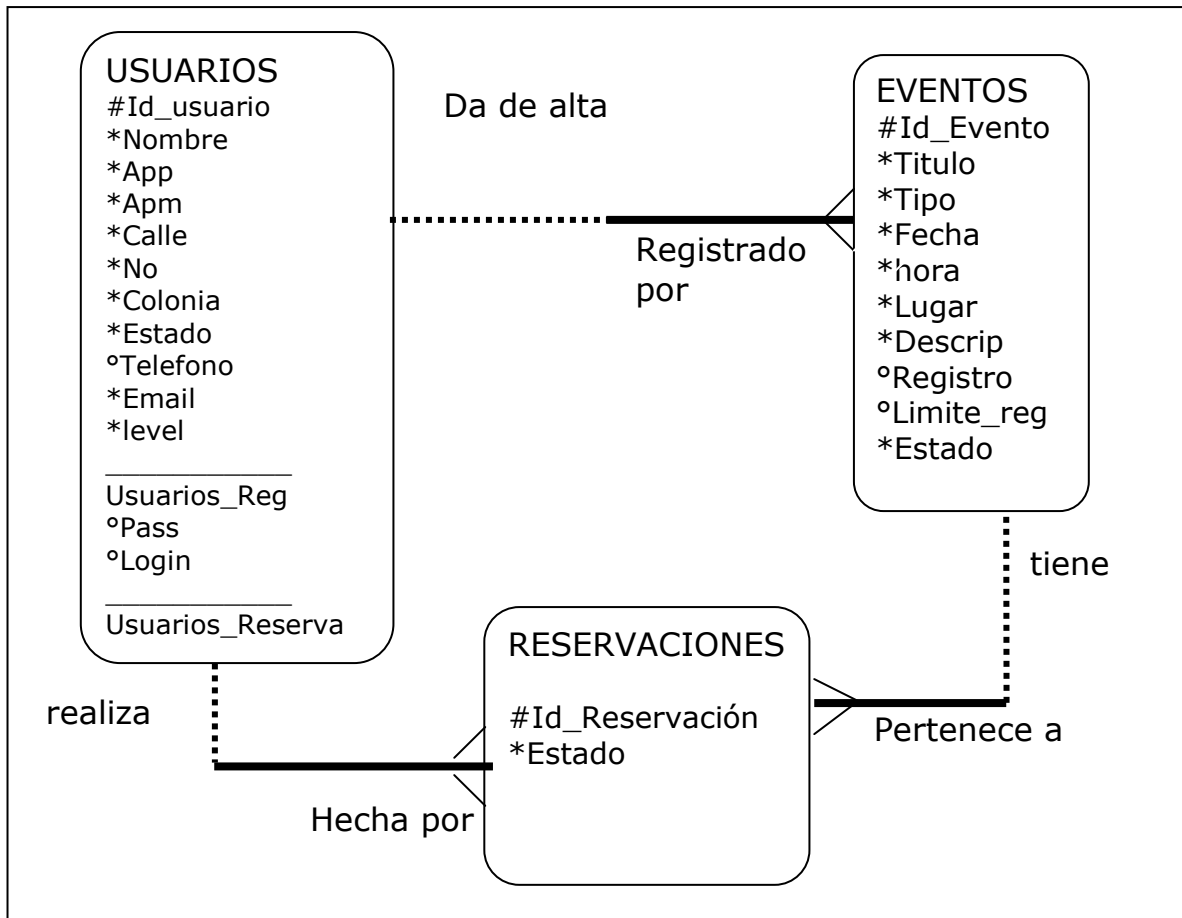


Figura 3.2 Diagrama Entidad-Relación

### III.2.1 NORMALIZACION

La teoría de la normalización proporciona un procedimiento riguroso para el diseño de bases de datos y ayuda a reconocer las cualidades no deseadas y la forma de corregirlas.

Una relación no-normalizada es una relación que contiene varias ocurrencias de algunos valores en cualquiera de sus campos. Por otro lado, una relación normalizada sólo permite una ocurrencia de un valor en cada campo. Las relaciones normalizadas se agrupan en cuatro categorías llamadas formas normales FN siendo cada nivel una descomposición más completa de una relación que la del nivel anterior.

Antes de describir el proceso de normalización, se debe descubrir la manera de determinar la forma normal de una relación a partir de su relación de atributos conocida como “dependencia funcional”.

El análisis de una relación de dependencia funcional entre los campos de una relación permite clasificar la relación en una de las cuatro formas normales. La dependencia funcional entre los atributos A y B en una relación se define como sigue: El atributo A es

funcionalmente dependiente del atributo B si el valor de A está determinado por el valor de B. Esta dependencia se puede simboliza como:  $B \rightarrow A$

La dependencia funcional para cada una de las entidades de nuestra base de datos se puede observar en la figura 3.3.

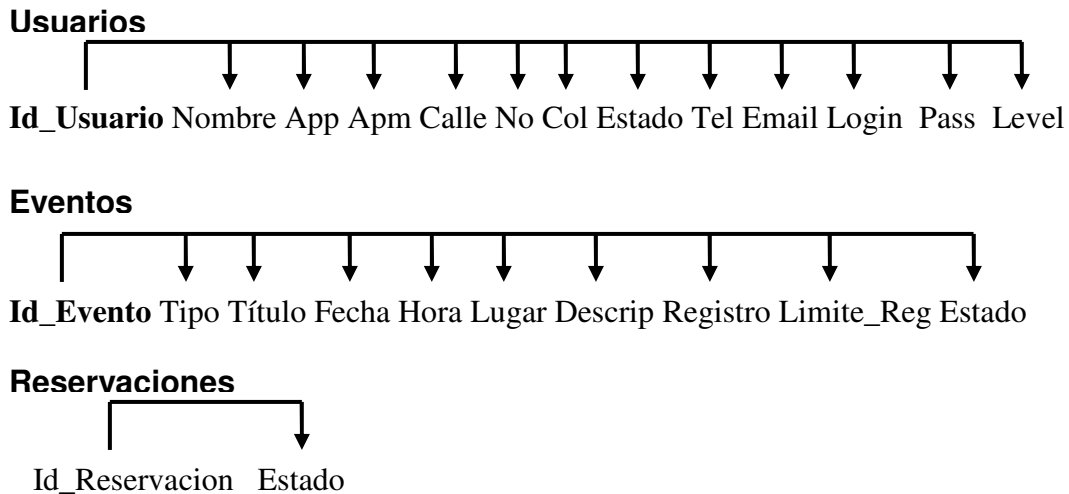


Figura 3.3 Diagrama de dependencias funcionales

Al hacer el proceso de normalizar una base de datos no siempre se llega a utilizar las cuatro formas normales; para la presente base de datos se hizo un análisis hasta la tercera forma normal para cada una de las entidades, ya que como se observa en las figuras 3.4, 3.5, 3.6, al final de la segunda forma ya se encontraba normalizada.

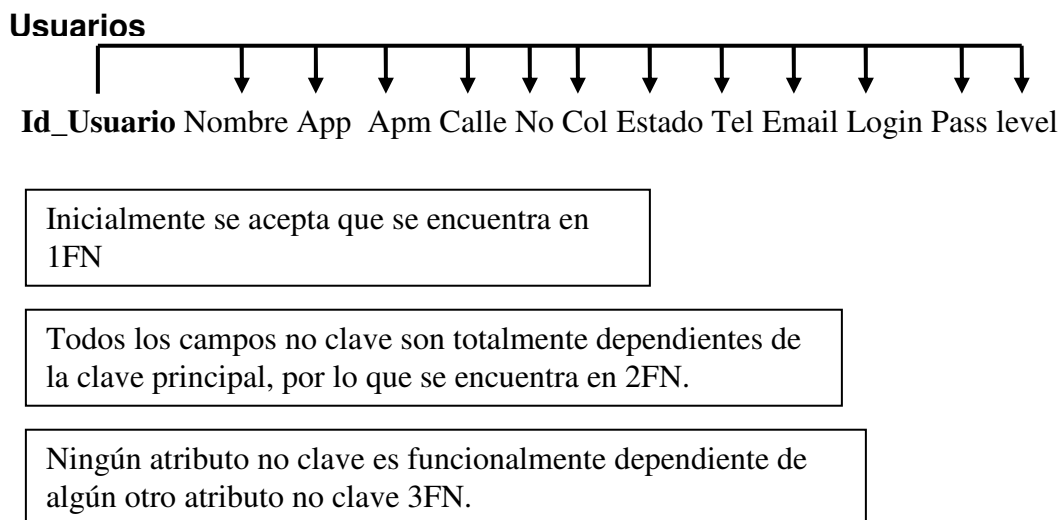


Figura 3.4 Normalización de "Usuarios".

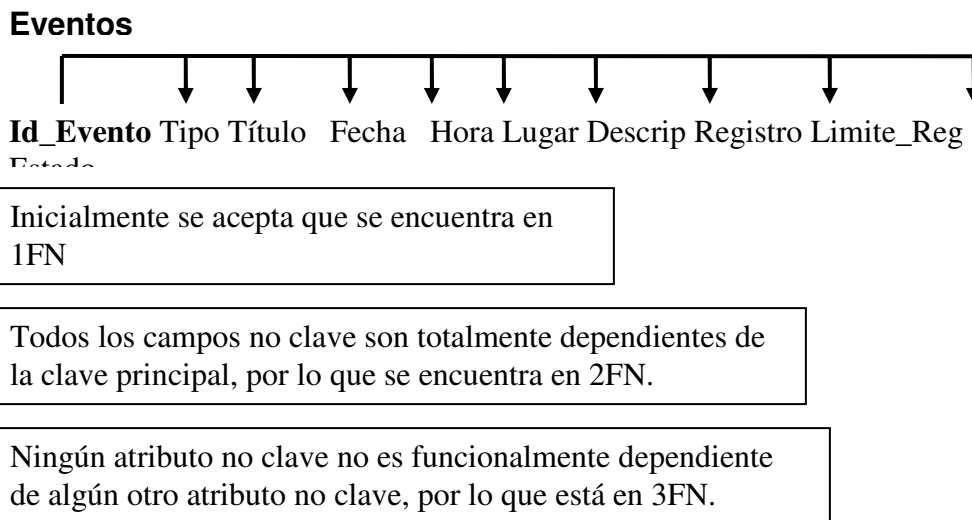


Figura 3.5 Normalización de “Eventos”

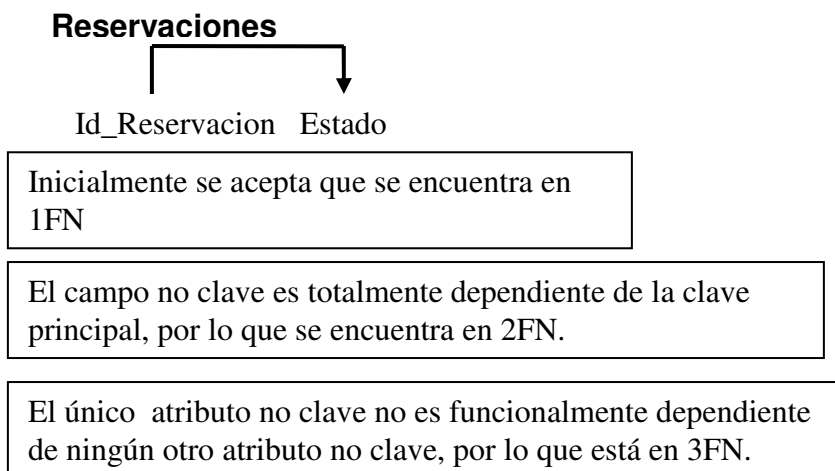


Figura 3.6. Normalización de “Reservaciones”

### III.2.2. DISEÑO DE LA BASE DE DATOS

En el Modelo Relacional todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura distinta.

La base de datos del este proyecto se diseño en base al modelo relacional, partiendo del la identificación de entidades, atributos y relación que llevaron a un modelo Entidad-Relación posteriormente se convierten en tablas según el modelo relacional del cual se partió y que están relacionadas entre si por medio de llaves foráneas.

Las Tablas 3.3., 3.4. y 3.5 muestran el esquema de la base de datos el cual ya incluye las relaciones existentes entre ellas con la inclusión de las llaves foráneas que hacen alusión a la correspondiente tabla relacionada.

### USUARIO

	Id_Usuario	Nombre	App	Apm	Calle	No	Col	Estado	Tel	Email	Login	Pass	level
<b>Key-type</b>	PK												
<b>Nulls</b>	NN <sup>♦</sup>	NN	NN	NN	NN		NN	NN		NN			NN

Tabla 3.3 Esquema de la Tabla Usuario ha implementarse en el Manejador de Base dedatos

### RESERVACIONES

	Id_Reservacion	Estado	Id_Evento	Id_usuario
<b>Key-Type</b>	PK <sup>♦</sup>		FK1 <sup>♦</sup>	FK2
<b>Nulls</b>	NN	NN	NN	NN

Tabla 3.4 Esquema de la Tabla Reservaciones

### EVENTOS

	Id_Evento	Titulo	Tipo	Fecha	Hora	Lugar	Descrip	Registro	Limite_Reg	Estado	Id_usuario
<b>Key-Type</b>	PK										FK
<b>Nulls</b>	NN	NN	NN	NN		NN	NN	NN		NN	NN

Tabla 3.5 Esquema de la Tabla Eventos

<sup>♦</sup> FK: Llave foránea, PK: llave primaria, NN: no nulo

### III.2.3 DICCIONARIO DE DATOS

Los principales datos se explican dentro de un Glosario de Términos o Diccionario de Datos, en el cual se explican y definen los elementos que conforman la base de datos del sistema

En la entidad Usuario se considera a las personas que ingresan al sistema, ya sea para registrarse y dar de alta eventos, o solo para hacer reservación a un evento específico.

Atributo	Tipo	Descripción	Dominio	Tamaño	Restricción
Id_usuario	Monovalorado Simple	Clave de identificación de usuario	Entero	10	No nulo, auto incrementable
nombre	Monovalorado Simple	Nombre de pila del usuario	Varchar	20	No nulo
app	Monovalorado Simple	Apellido paterno	Varchar	20	No nulo
apm	Monovalorado Simple	Apellido materno	Varchar	20	No nulo
calle	Monovalorado Simple	Calle (domicilio del usuario)	Varchar	20	No nulo
num	Monovalorado Simple	Número en el domicilio	Varchar	9	No nulo
colonia	Monovalorado Simple	Colonia	Varchar	20	No nulo
estado	Monovalorado Simple	Estado de la República Mexicana	Varchar	15	No nulo
tel	Monovalorado Simple	Teléfono	Varchar	10	No nulo
email	Monovalorado Simple	Dirección de correo electrónico	Varchar	30	No nulo
login	Monovalorado Simple	Nombre de usuario de registro	Varchar	10	
pass	Monovalorado Simple	Contraseña escogida por el usuario al registrarse	Varchar	10	
level	Monovalorado Simple	Nivel de usuario	Entero	2	No nulo

Tabla 3.6 Usuarios

<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>Dominio</b>	<b>Tamaño</b>	<b>Restricción</b>
Id_evento	Monovalorado Simple	Clave de identificación de usuario	Entero	10	No nulo, auto incrementable
Titulo	Monovalorado Simple	Nombre del evento	Varchar	50	No nulo
Tipo	Monovalorado Simple	Tipo de evento a realizarse (académico, general, deportivo)	Char	1	No nulo
Fecha	Monovalorado Simple	Fecha de realización	Varchar	11	No nulo
Hora	Monovalorado Simple	Hora de realización	Varchar	5	No nulo
Lugar	Monovalorado Simple	Lugar de realización	Varchar	50	No nulo
Descrip	Monovalorado Simple	Descripción del evento	Text		No nulo
Registro	Monovalorado Simple	Si al evento se puede registrar (0 no, 1 si)	Binary		
Limite_reg	Monovalorado Simple	Limite de registros permitidos para el evento	Int		
Estatus	Monovalorado Simple	Estado de activación del evento(activo, pendiente, realizado)	Char	1	No nulo

Tabla 3.7 Eventos

<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>Dominio</b>	<b>Tamaño</b>	<b>Restricción</b>
Id_reservación	Monovalorado Simple	Clave de identificación de reservación	Entero	10	No nulo, auto incrementable
estatus	Monovalorado Simple	Estado de la reservación (0 pendiente, 1 activa)	Binary		Defaul 0

Tabla 3.8. Reservaciones.

## **CAPITULO IV**

### **IMPLEMENTACION Y PRUEBAS DEL SISTEMA**

Después de llevado a cabo el proceso de análisis del sistema realizando el desglose de los diferentes casos de uso y el flujo básico de cada uno de ellos, y procediendo después con el diseño de la base de datos; ahora se procede a la implementación del sistema.

Para esta fase del desarrollo del presente proyecto, se utilizará un lenguaje de programación, un sistema administrador de bases de datos y herramientas para el diseño del entorno en el que se podrá manipular los datos que son la fuente principal sobre la que gira el Sistema Administrador de Eventos.

Como en el Capítulo I anterior se había señalado dentro del apartado I.2.4 de Herramientas para un Sistema de Bases de Datos Web; la implementación se lleva a cabo utilizando el sistema administrador de bases de datos MySQL, el lenguaje de programación PHP y HTML muy necesario para cualquier entorno web. Dicha implementación se realiza sobre una plataforma Windows.

Aunque la implementación se basa en plataforma Windows, dada la naturaleza del software utilizado para crear el sistema; no existe ningún inconveniente para migrar a otra plataforma este sistema, pudiendo ser poco necesario o innecesario realizar ajustes de programación.

#### **IV.1 HERRAMIENTAS PARA EL DESARROLLO DEL SISTEMA**

Dentro de este apartado solo se hace una recapitulación de la información sobre las herramientas usadas, de las que se había mencionado en el Capítulo I al considerar el marco teórico del presente trabajo.

##### **El manejador de Bases de Datos**

MySQL un sistema preferido de plataforma libre sencillo de usar y rápido. Teniendo dentro de sus características:

- La accesibilidad de su código fuente para cualquiera.
- Disponibilidad en gran cantidad de plataformas y sistemas
- Creado para manejar bases de datos relacionales, por lo que en él, las unidades básicas son tablas, columnas y filas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.

- Conectividad segura.
- Replicación.

MySQL tiene los dos tipos de comandos:

- DDL que crean y definen nuevas bases de datos, campos e índices.
- DML generan consultas para ordenar, filtrar y extraer datos.

### **Lenguaje de Programación.**

PHP al igual que MySQL se encuentra disponible para cualquiera. Su facilidad para embeber fragmentos de código dentro de HTML y realizar determinadas acciones sin generar programas en lenguaje distinto al HTML, hacen de esta una gran herramienta.

Como características primordiales en PHP tenemos:

- Gestión de bases de datos: PHP ofrece interfaces para el acceso a la mayoría de las bases de datos..
- Funciones de correo electrónico: envíos individuales o por grupos, con opciones para toda una serie de aspectos dentro de un email.
- Gestión de archivos: aperturas, borrados, modificaciones, etc.
- Interpretación y ejecución en el servidor web.

### **Herramientas adicionales**

Para realizar el diseño de la interfaz, se cuenta con HTML y hojas de estilo (CSS) que son conjunto de instrucciones que definen la apariencia de diversos elementos de un documento HTML. Permite indicar el tamaño de la fuente, color y estilo de cierto párrafo que se indique, mediante la definición de estilos y aplicación de los mismos.

Las hojas de estilo son la innovación más importante al HTML que le dan capacidades que no tenía anteriormente y también se usan en otros lenguajes de XML y SGML. Son la base de implementación estándar del HTML dinámico o DHTML.

## **IV.1.2 IMPLEMENTACION DE LA BASE DE DATOS EN MYSQL**

Dentro de MySQL se hizo la implementación de la base de datos que en lenguaje SQL es la siguiente:

Se crea la base de datos:

```
CREATE DATABASE admineventos;
```

Se crea la tabla usuario:

```
CREATE TABLE usuario (  
idusuario INT( 10 ) NOT NULL AUTO_INCREMENT primary key,  
nombre VARCHAR( 20 ) NOT NULL ,  
app VARCHAR( 20 ) NOT NULL ,  
apm VARCHAR( 20 ) NOT NULL ,  
calle VARCHAR( 20 ) NOT NULL ,  
num VARCHAR( 9 ) NOT NULL ,  
colonia VARCHAR( 20 ) NOT NULL ,  
estado VARCHAR( 15 ) NOT NULL ,  
tel VARCHAR( 10 ) NULL ,  
email VARCHAR( 30 ) NULL ,  
login VARCHAR( 10 ) NULL ,  
pass VARCHAR( 10 ) NULL ,  
level INT(2) NOT NULL  
) Engine innodb
```

Creación de tabla eventos y reservaciones:

```
CREATE TABLE eventos (  
idevento INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
titulo VARCHAR( 50 ) NOT NULL ,  
tipo CHAR( 1 ) NOT NULL ,  
fecha VARCHAR( 11 ) NOT NULL ,  
hora VARCHAR( 5 ) NULL ,  
lugar VARCHAR( 50 ) NOT NULL ,  
descrip TEXT NOT NULL ,  
registro BINARY NULL ,  
limite_reg INT UNSIGNED NULL ,  
estatus CHAR( 1 ) NOT NULL ,  
e_idusuario INT NOT NULL  
) engine innodb
```

```
CREATE TABLE reservaciones (  
idreservacion INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
estatus BINARY NOT NULL DEFAULT 0 ,  
r_idusuario INT NOT NULL ,  
r_idevento INT NOT NULL  
) engine innodb
```

Se agrega integridad referencial. Así, si se elimina el usuario se eliminará la reservación hecha por él si era un usuario cliente, pero si ese usuario era registrado, no se eliminarán los eventos que diera de alta.

```
Alter table reservaciones add constraint foreign key (r_idevento) referentes eventos(idevento) on  
update cascade
```

```
Alter table reservaciones add constraint Foreign Key (r_idusuario) references usuario(idusuario) on  
update cascade
```

```
Alter table eventos add constraint foreign key (e_idusuario) references usuario(idusuario) on update  
cascade
```

Esto asegura que los eventos queden guardados de forma histórica, aunque hayan sido realizados.

## IV.2 CONEXIÓN DE MYSQL CON PHP

Para realizar la conexión se realizó la función conectarse, que sirve para conectarse a la base de datos en MySQL:

```
function conectarse()
{
if (!($link=mysql_connect("localhost","root",""))) //información sobre el Host y el usuario
{
echo "Error conectando a la base de datos.";
exit();
}
if (!mysql_select_db("admineventos",$link)) // Nombre de la Base de Datos a utilizar
//se establece la conexión con la base de datos//
{
echo "Error seleccionando la base de datos.";
exit();
}
session_start(); // manejo de sesiones
return $link;
}
```

◆ Para realizar búsquedas de eventos a realizarse se tiene el siguiente código:

```
$result= mysql_query(" select * from eventos where (tipo='$_GET[tipo]') and
(estatus='$_GET[hecho]')",$link);
```

◆ Búsqueda por día mes y año de eventos académicos:

```
$result= mysql_query(" select * from eventos where (fecha='$fecha') and
(estatus='$_GET[hecho]')",$link);
```

◆ Código para dar de alta evento, solo si es académico se agrega registro y limite de registro:

```
if ($_POST[tipo]==a)
{$sql = " insert into eventos values( ";
$sql .= """,";
$sql .= $_POST[titulo].""",";
$sql .= $_POST[tipo].""",";
$sql .= $fecha.""",";
$sql .= $hora.""",";
$sql .= """.$_POST[lugar].""",";
$sql .= """.$_POST[descrip].""",";
$sql .= $_POST[registro].""";
```

```

$sql .= $_POST[limite_reg].",";
$sql .= "'a'";
$sql .= $_SESSION[idusuario].")";
}
else{$sql = " insert into eventos values( ";
$sql .= """,";
$sql .= $_POST[titulo].""";
$sql .= $_POST[tipo].""";
$sql .= $fecha.""";
$sql .= $hora.""";
$sql .= """.$_POST[lugar].""";
$sql .= """.$_POST[descrip].""";
$sql .= "0,";
$sql .= "0,";
$sql .= "'a'";
$sql .= $_SESSION[idusuario].")";
}
echo 'Evento agregado';
$res = mysql_query($sql);
◆ Código para dar de baja un evento:

```

```

$sql = "delete from eventos where idevento=".$_GET[b_idevento];
$res = mysql_query($sql);
echo 'Evento Eliminado';

```

◆ Código para modificar un evento:

```

$sql = "update eventos set ";
$sql .= " titulo=".$_POST[titulo].""";
$sql .= " tipo=".$_POST[tipo].""";
$sql .= " fecha=".$_POST[fecha].""";
$sql .= " hora=".$_POST[hora].""";
$sql .= " lugar=".$_POST[lugar].""";
$sql .= " descrip=".$_POST[descrip].""";
$sql .= " registro=".$_POST[registro].""";
$sql .= " limite_reg=".$_POST[limite_reg];
$sql .= " where idevento= ". $_POST[b_idevento];
$res = mysql_query($sql);

```

El código para eliminar usuario y eliminar reservación es semejante, dado que por la integridad referencial agregada a la base de datos, al eliminar el usuario que se registró a un evento, se elimina la entrada dentro de reservaciones.

```
$sql = "delete from usuario where idusuario="._GET[b_idusuario];  
$res = mysql_query($sql);  
echo 'Usuario Eliminado';
```

### IV.3 INTERFACES DEL SISTEMA DE ADMINISTRADOR DE EVENTOS.

El Sistema Administración de Eventos, comienza mostrando una pantalla que en su parte lateral derecha permite elegir alguna de las diferentes consultas de eventos que se pueden hacer como se ve en la figura 4.1 (mostrar eventos realizados con búsqueda por fecha, por realizarse; académicos, generales y deportivos).

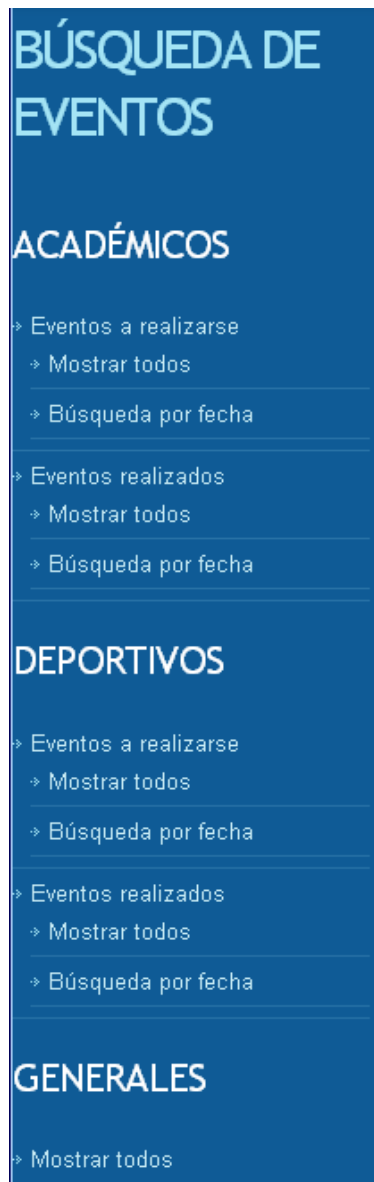


Figura 4.1 Interfaz Búsquedas en pantalla principal

También en la misma pantalla se puede hacer la validación de usuario registrado o administrador; y si no está registrado, con opción de registro, como lo muestra la figura 4.2.



The image shows a user login interface with a blue background. At the top, the title "Usuario Registrado" is displayed in white. Below the title, there are two input fields: "Login:" and "Contraseña:". The "Login:" field is a light blue rectangle, and the "Contraseña:" field is a light blue rectangle with a white border. Below the password field is a blue button with the text "Enviar" in white. At the bottom of the form, there is a link labeled "Registrarse" in white text.

Figura 4.2. Validación de usuario

En la figura 4.3, se presenta un diagrama de contenido del sistema que permite ser una guía de navegación del Sistema una vez que se abandona la pantalla principal y dependiendo de la validación que se haga de usuario, o en caso de sólo consulta como un usuario cliente.

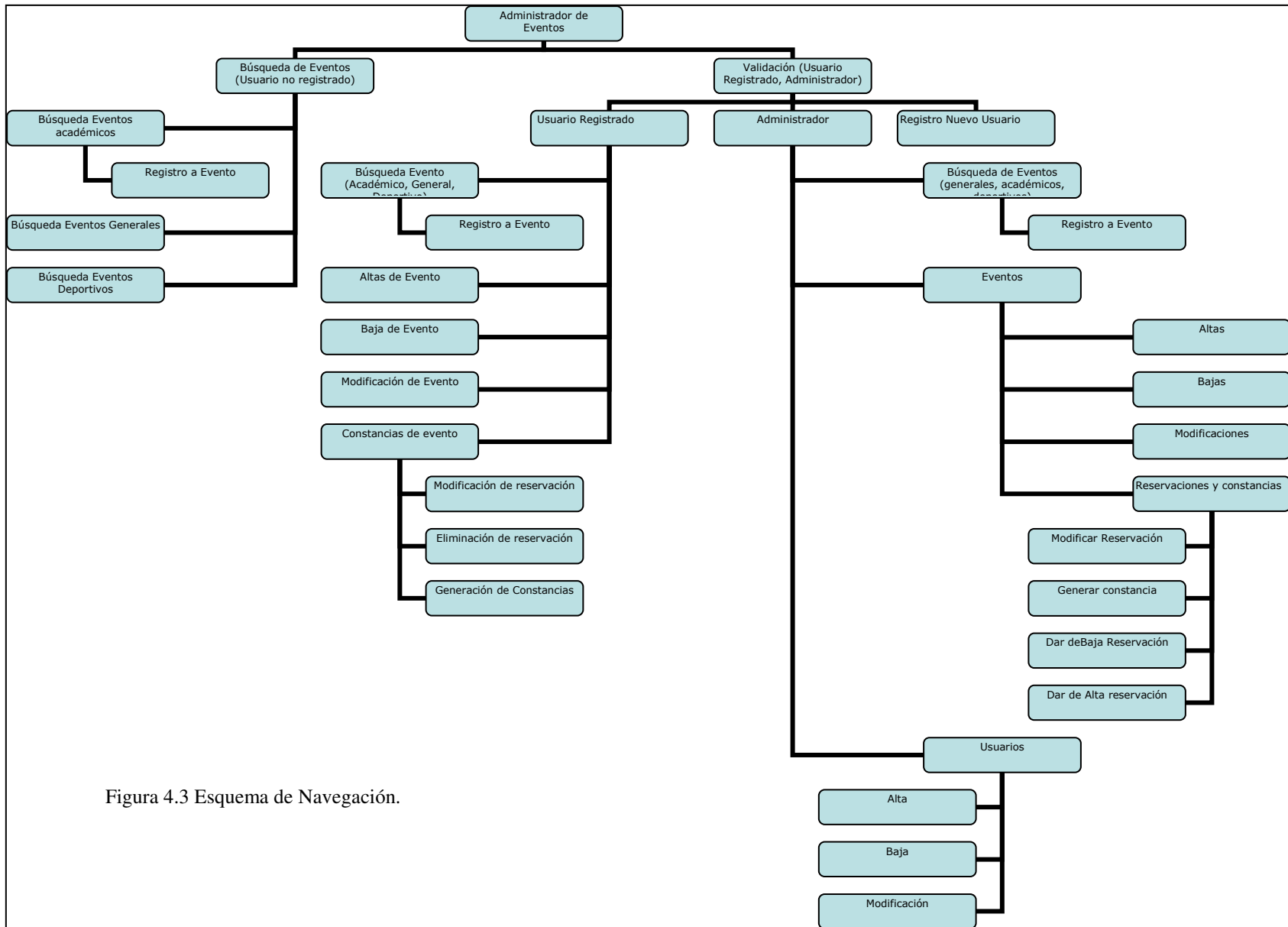


Figura 4.3 Esquema de Navegación.

### IV.3.1 INTERFACES DE USUARIO ADMINISTRADOR

La figura 4. presenta la interfaz una vez que se ha hecho la validación como Administrador, con sus distintas opciones. La figura 4.5 muestra la pantalla al escoger eventos, en el que se hace la administración de todos los eventos dados de alta por cualquier usuario, ver reservaciones y hacer constancias, dar de baja el evento o modificar sus datos.



Figura 4.4 Pantalla Principal del Administrador



Figura 4.5. Manejo de Eventos por Administrador

Un evento tiene tres diferentes estatus como se ve en la figura 4.6.




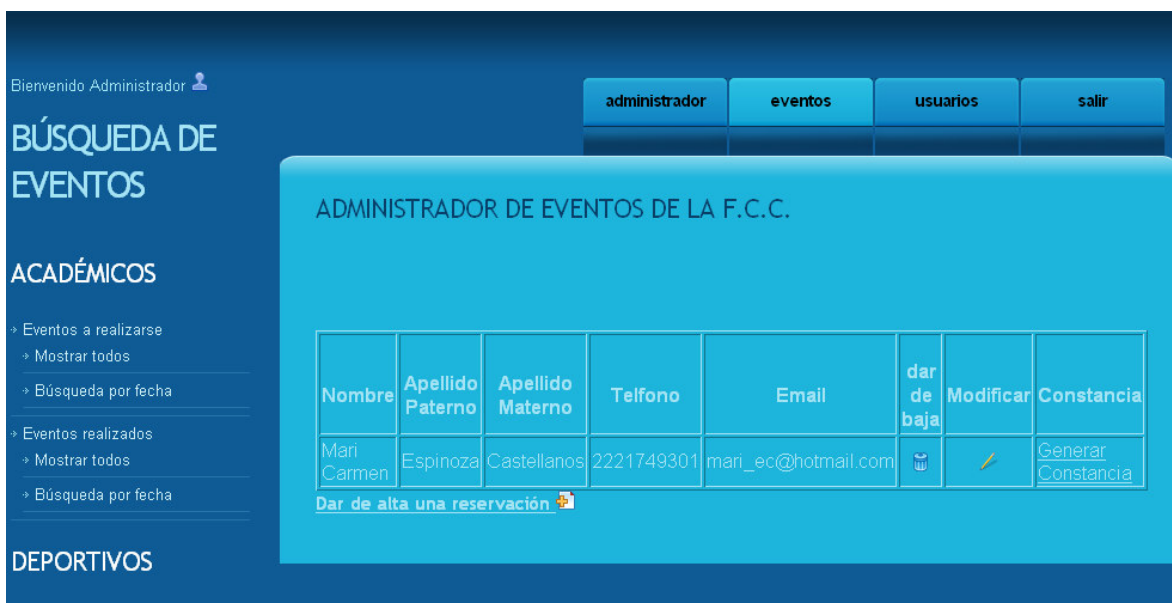

	<b>Activo:</b> El evento está próximo a realizarse
	<b>Pendiente:</b> Aún no ha sido aprobado por el administrador y no se muestra en las consultas por evento.
	<b>Realizado:</b> Evento que ya ha sido realizado

Figura 4.6.Estatus de un evento

Dentro de Reservas y Constancias de un evento en la figura 4.7, se ve todas las reservas hechas y las opciones para ellas: dar de baja, modificar la reservación y generar una constancia.



Bienvenido Administrador 

**BÚSQUEDA DE EVENTOS**

**ACADÉMICOS**

- Eventos a realizarse
- Mostrar todos
- Búsqueda por fecha



→ Eventos realizados

- Mostrar todos
- Búsqueda por fecha

**DEPORTIVOS**

administrador | eventos | usuarios | salir

ADMINISTRADOR DE EVENTOS DE LA F.C.C.

Nombre	Apellido Paterno	Apellido Materno	Telfono	Email	dar de baja	Modificar	Constancia
Mari Carmen	Espinoza	Castellanos	2221749301	mari_ec@hotmail.com			Generar Constancia


[Dar de alta una reservación](#) 

Figura 4.7 Reservas y Constancias

Al modificar la reservación, lo que se realiza es una modificación de los datos del usuario que reserva similar a modificar usuario en la Gestión de Usuarios dentro de las opciones del Administrador como se observa en las figuras 4.8. y 4.9



Figura 4.8 Administración de Usuarios



Figura 4.9 Modificación de Usuario

### IV.3.2 INTERFACES DE USUARIO REGISTRADO

Al validarse como usuario registrado se muestra la pantalla de la figura 4.10 con opciones para altas de evento, bajas, modificaciones y constancias.



Figura 4.10 Interfaz principal de usuario registrado



Figura 4.11 Altas de evento

En modificaciones (figura 4.12) al entrar a un evento específico se puede ver toda la información que puede cambiar el usuario registrado que había dado de alta al evento (figura 4.13), en la pantalla se muestran los datos de el evento y si se cambia alguno se debe pulsar el botón "Modificar" para que se haga efectiva la modificación.

Bienvenido(a)  
mari

**BÚSQUEDA DE EVENTOS**

**ACADÉMICOS**

- Eventos a realizarse
- Mostrar todos
- Búsqueda por fecha

→ Eventos realizados

- Mostrar todos
- Búsqueda por fecha

**DEPORTIVOS**

administrador    altas    bajas    **modificaciones**    constancias    salir

ADMINISTRADOR DE EVENTOS DE LA F.C.C.

Titulo	Tipo	Fecha	Hora	Lugar	Descripcion	Estatus	
Conferencia: Avances Recientes	a	2010-04-05	14:00	Auditorio	Impartida por el Dr. Martin Colton de la Universidad de Rutgers	●	✎
Seminario S.P.	a	2008-03-07	15:00	Auditorio	Impartida por el Dr. Michel Vallieres 2, 3y 4 de Diciembre del 2008	●	✎
Concierto de Guitarra Clasica	g	2008-03-22	17:00	Explanada FCC	Martin Saenz Ejecutante	●	✎
Venta de PC Dual Core	g	2008-05-16	17:00	2 Norte 505	Venta PC \$5,500 comunicarse con Javier P.rez	●	✎
Simposium de					Simposium de inteligencia		

Figura 4.12 Modificaciones de eventos

Bienvenido(a)  
mari

**BÚSQUEDA DE EVENTOS**

**ACADÉMICOS**

- Eventos a realizarse
- Mostrar todos
- Búsqueda por fecha

→ Eventos realizados

- Mostrar todos
- Búsqueda por fecha

**DEPORTIVOS**

administrador    altas    bajas    **modificaciones**    constancias    salir

ADMINISTRADOR DE EVENTOS DE LA F.C.C.

Titulo del Evento	Seminario S.P.
Tipo	Acadmico
Fecha	07 Marzo 2008
Hora	15 00
Lugar	Auditorio
Descripcion del evento	Impartida por el Dr. Michel Vallieres 2, 3y 4 de Diciembre del 2008
Registro al evento	Si
Limite de registro	20
<b>Modificar</b>	

Figura 4.13 Modificación de evento específico

En caso de Administrador y Usuario Registrado en el que se elija la opción de crear constancia, se creará un documento en PDF (ver figura 4.14 como ejemplo) solo para visualización en el navegador con los datos de la persona registrada anteriormente en Reservas y Constancias para el caso de Administrador y Constancias para Usuario Registrado que son los únicos capacitados para generarlas. Aquí se podrá escoger si se desea uno o dos firmantes, así como agregar sus nombres a la constancia generada.

Esta funcionalidad es una de las principales consideraciones hechas durante el análisis del sistema y que el proyecto cumple.

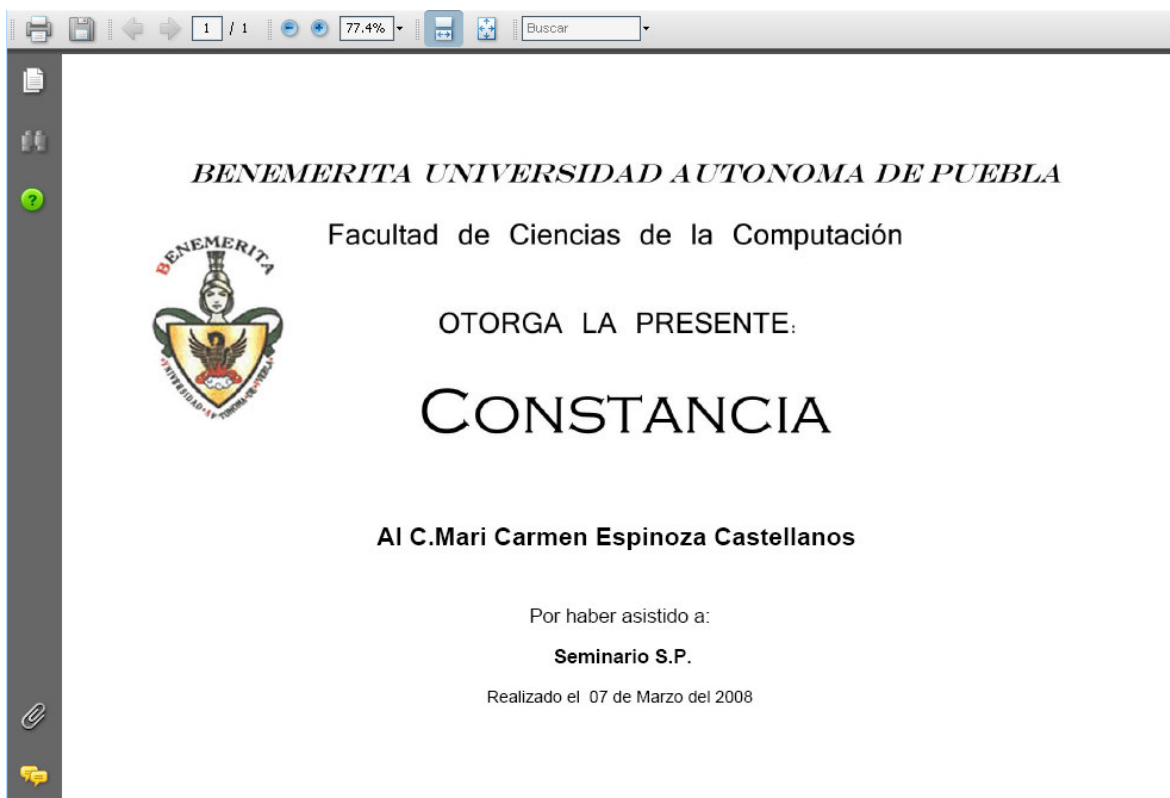


Figura 4.14 Constancia Generada

## **CONCLUSIONES**

Una vez hecha la implementación del sistema y realizadas las pruebas, el sistema cumple con los requisitos planteados al inicio del proyecto.

Se cuenta con una seguridad al acceso, con dos tipos de usuario para los cuales se debe contar con los datos válidos y así lograr acceder al sistema en alguna de las dos modalidades: Usuario Registrado y Administrador.

El sistema cubre principalmente dentro de todas las pantallas principales, el hecho de poder hacer una búsqueda dentro de los eventos dados de alta en el sistema dependiendo del tipo que sean.

En el Desarrollo del sistema hubo toda una revisión del procedimiento de análisis y desarrollo de una base de datos así como afirmación de conocimientos sobre el modelo cliente servidor y las herramientas para implantar uno.

Al Utilizar la Metodología de Ingeniería de Software; el Modelo clásico o cascada, se nota que el modelo es adecuado para este proyecto, ya que los requisitos y funcionalidades requeridas son entendidos sin mayores complicaciones al principio del desarrollo del software, dicho modelo es aplicable a sistemas similares a éste donde la información se encuentra acotada y la complejidad de éste no está sobrepasada por las etapas previstas por el modelo.

MySQL resulta ser la herramienta precisa en la cual se puede implantar toda la información que requiere el sistema y el manejo de ésta se realiza con relativa facilidad. Es de destacar que la administración de la información por parte de MySQL es óptima y el enlace a la base de datos transparente en el desarrollo del sistema.

Utilizando el lenguaje PHP se logra una excelente combinación con MySQL, a la vez que se puede hacer el agregado fácilmente de elementos en html o cualquier otra herramienta de desarrollo web sin que ello quite o dificulte las funcionalidades nativas de PHP.

El Sistema finalizado cubre la necesidad para la cual fue proyectado, hacer una Administración de Eventos para la Facultad de Ciencias de la Computación que pueda ser consultada por cualquier persona, con funcionalidades especiales para el usuario registrado (búsquedas, reservaciones, generación de constancias para determinados eventos , etc.), con las ventajas que ofrece una interfaz web de disponibilidad de información desde cualquier parte del mundo.

## **PERSPECTIVAS**

- Conveniencia de agregar la opción para imprimir el resultado de una búsqueda de eventos.
- Emisión de un email de confirmación al haberse validado una reservación por medio del Usuario Registrado o el Administrador.
- Al confirmar el estado de un evento pendiente activo, emisión de email para confirmar el nuevo estatus.

# **BIBLIOGRAFÍA**

[1] Ingeniería de Sistemas de Software

León Serrano

Isdefe

Primera Edición

1996

[2] Sistemas de Bases de Datos, conceptos Fundamentales.

Ramez Elmasri, Shakant B. Navathe

Segunda Edición

Addison-Wesley Iberoamericana

1999

[3] Aplicaciones de Bases de datos Cliente Servidor

Dr. Martín Ortiz Manuel

Notas del Curso Bases de Datos

FCC-BUAP

2004

[4] Ingeniería de Software. Un enfoque práctico

Roger S. Presuman

Quinta Edición

Mc Graw Hill

2002

[5] MySQL

Paull Dubois

Primera Edición

Developer's Library

2003

[6] Fundamentos y modelos de BASES de DATOS

Mario Piattini

Primera Edición

Alfaomega, Ra-Ma

2004

[7] Fundamentos de SQL

Forrest Houlette

Primera Edición

Mc Graw Hill

2001

**[8]** Introducción a los Sistemas de Bases de Datos

C.J. Dafe.

Séptima Edición

Pearson Educación

2001

**[9]** PHP 4 a través de ejemplos

Abraham Gutiérrez, Ginés Gravo

Primera Edición

Alfaomega, Ra-Ma

2004

**[10]** Creación de un Portal con PHP yMySQL

Jacobo Pavón Puertas

Tercera Edición

Alfaomega, Ra-Ma

2007

**[11]** Proyectos Profesionales Programación PHP 5

Francisco Chartre Ojeda

Primera Edición

Anaya Multimedia

2004

**[12]** MySQL para Windows y Linux

César Pérez

Primera Edición

Alfaomega, Ra-Ma

2004

**[13]** Fundamentos de Desarrollo Web con PHP, Apache y MySQL

Glass, Scouarnec, Naramore, Mailer, Solz, Gemer.

Primera Edición.

Ediciones Anaya Multimedia

2004

**[14]** Análisis y Diseño de Bases de datos

M.C. Ambrosio Vázquez Alma Delia

FCC - BUAP

2007

**[15]** Implementación de Bases de Datos Cliente- Servidor

M.C. Bello López Pedro

FCC- BUAP

2007