



---

---

**BENEMERITA UNIVERSIDAD AUTONOMA DE  
PUEBLA**

---

**FACULTAD DE CIENCIAS DE LA COMPUTACION**

**Sistema Web Para la Gestión de Mantenimiento  
Preventivo y Correctivo de Equipos de Cómputo**

**TESIS PROFESIONAL**

Que para obtener el título de

**LICENCIADA EN CIENCIAS DE LA COMPUTACION**

Presenta:  
Graciela López Rivera

Asesor:  
M.C. Mario Anzures García

Puebla, Pué.

Septiembre 2012



---

---

# CONTENIDO

<b>1. Introducción.....</b>	<b>5</b>
1.1 Descripción del Proyecto.....	6
1.2 Antecedentes del Proyecto.....	6
1.3 Objetivos Generales y específicos del proyecto.....	7
1.4 Resultados Esperados.....	7
1.5 Impacto Socioeconómico.....	8
1.6 Aportaciones.....	8
<b>2. Estado del Arte.....</b>	<b>9</b>
2.1 Análisis Comparativo.....	10
2.1.1 Easy Maint Software de Mantenimiento.....	10
2.1.2 Sistema Sitehelpdesk.....	12
2.1.3 Sistema Satnetwork.....	13
2.2 Ingeniería de Software.....	15
2.2.1 Definición del término Ingeniería de Software.....	15
2.3 Software.....	16
2.3.1 Características del software.....	16
2.3.2 Aplicaciones del Software.....	17
2.3.3 Visión General de Ingeniería de Software.....	17
2.3.4 Modelos de Procesos de Software.....	18
2.4 Metodología.....	19
2.4.1 Modelos Lineal Secuencial.....	19
2.5 El lenguaje de Modelo Unificado (UML).....	20
2.5.1 Diagramas de caso de Uso.....	22
2.5.2 Diagramas de clases.....	24
2.5.3 Diagramas de Estado.....	24
2.5.4 Diagrama de Actividades.....	25
2.5.5 Diagrama de Secuencia.....	27
2.5.6 Diagrama de Colaboración.....	29
2.6 Base de Datos.....	31
2.6.1 Abstracción de datos.....	33
2.6.2 Usuarios de la base de datos.....	34
2.6.3 Tipos de Base de Datos.....	34
2.7 Modelo de datos.....	35
2.7.1 Modelo lógico basado en Objetos.....	35
2.7.2 Modelo Entidad Relación.....	36
2.7.3 Tipos de Relaciones.....	37
2.7.4 Llaves Primarias.....	38
2.8 Normalización.....	38
2.8.1 Formas de Normalización.....	39
2.9 Desarrollo web.....	40
2.10 Apache.....	40
2.11 PHP.....	41
2.12 MYSQL.....	41



---

---

2.12.1 Comandos DDL.....	42
2.12.2 Comandos DML.....	43
<b>3. Desarrollo del Sistema.....</b>	<b>45</b>
3.1 Análisis de Requisitos.....	46
3.1.1 Justificación.....	46
3.1.2 Ámbito del Sistema.....	46
3.1.3 Características de los Usuarios.....	47
3.1.4 Recursos del Sistema.....	47
3.1.5 Requerimientos de la Interfaz.....	48
3.1.6 Restricciones.....	49
3.1.7 Funciones del Sistema.....	49
3.1.8 Casos de Uso.....	50
3.2 Diseño del Sistema.....	51
3.2.1 Diagrama Entidad-Relación.....	52
3.2.2 Diagrama de Clases.....	53
3.2.3 Diagrama de Secuencia.....	54
3.2.4 Diagrama de Actividad.....	59
3.2.5 Diagrama de Estado.....	60
3.3 Desarrollo del Sistema.....	61
3.3.1 Herramientas Del Sistema.....	61
3.3.2 Interfaces.....	61
3.4 Pruebas.....	82
<b>4. Conclusión.....</b>	<b>98</b>
<b>Trabajos a futuro.....</b>	<b>98</b>
<b>Bibliografía.....</b>	<b>99</b>



---

## ÍNDICE DE FIGURAS

Figura 2.1	Las Fases de un Bucle de Resolución de problemas.....	18
Figura 2.2	Modelo lineal secuencial.....	19
Figura 2.3	Ejemplo de Actores.....	23
Figura 2.4	Esquema de Una Clase.....	24
Figura 2.5	Ejemplo de Diagrama de Estado.....	25
Figura 2.6	Ejemplo de Diagrama de Actividades.....	26
Figura 2.7	Ejemplo de Diagrama de Secuencia.....	27
Figura 2.8	Ejemplo de Diagrama de Colaboración.....	30
Figura 2.9	Símbolos Usados en el Diagrama E-R.....	36
Figura 2.10	Relación Uno a Uno.....	37
Figura 2.11	Relación Uno a Muchos.....	37
Figura 2.12	Relación Muchos a Uno.....	37
Figura 2.13	Relación Muchos a Muchos.....	37
Figura 3.1	Diagramas de Casos de Uso.....	51
Figura 3.2	Diagrama Entidad-Relación.....	52
Figura 3.3	Diagrama de Clases.....	53
Figura 3.4	Diagrama de Secuencia para Ingresar al Sistema.....	54
Figura 3.5	Diagrama de Secuencia para Realizar Petición.....	55
Figura 3.6	Diagrama de Secuencia para Revisar Petición.....	55
Figura 3.7	Diagrama de Secuencia para Cerrar Petición.....	56
Figura 3.8	Diagrama de Secuencia para Elaborar Orden.....	56
Figura 3.9	Diagrama de Secuencia para Elaborar Aviso.....	57
Figura 3.10	Diagrama de Secuencia para Elaborar Reporte.....	58
Figura 3.11	Diagrama de Secuencia para Visualizar Reporte.....	58
Figura 3.12	Diagrama de Actividad.....	59
Figura 3.13	Diagrama de Estado.....	60
Figura 3.14	Menú Principal.....	62
Figura 3.15	Interfaz de Información sobre la Empresa.....	62
Figura 3.16	Interfaz Login y Opciones como Ingresar al Sistema.....	63
Figura 3.17	Interfaz de Usuario y Contraseña.....	63
Figura 3.18	Interfaz Principal de los Usuarios.....	64
Figura 3.19	Interfaz de Peticiones.....	65
Figura 3.20	Interfaz de Avisos.....	66
Figura 3.21	Interfaz de Contacto.....	66
Figura 3.22	Interfaz de Información para Usuarios.....	67
Figura 3.23	Interfaz Principal para Administradores.....	68
Figura 3.24	Interfaz Consulta de Datos.....	69
Figura 3.25	Interfaz Consulta de Datos Usuarios.....	69
Figura 3.26	Interfaz Consulta de Datos de Equipo.....	70
Figura 3.27	Interfaz de Insertar Información.....	70
Figura 3.28	Interfaz de Insertar Información en Usuarios.....	71
Figura 3.29	Interfaz de Insertar Información en Equipos.....	71
Figura 3.30	Interfaz de Eliminar Información.....	72
Figura 3.31	Interfaz de Eliminar Información en Usuarios.....	72
Figura 3.32	Interfaz de Eliminar Información en Equipos.....	73
Figura 3.33	Interfaz de Actualizar Información.....	73



---

Figura 3.34	Interfaz de Actualizar Usuarios.....	74
Figura 3.35	Interfaz de Actualizar Equipos.....	74
Figura 3.36	Interfaz de Revisar Peticiones.....	75
Figura 3.37	Interfaz de Revisar Peticiones2.....	75
Figura 3.38	Interfaz de Búsqueda.....	76
Figura 3.39	Interfaz de Búsqueda por Opciones.....	76
Figura 3.40	Interfaz de Reportes en Menú de Administradores.....	77
Figura 3.41	Interfaz de Reportes.....	77
Figura 3.42	Interfaz de Órdenes en Menú de Administradores.....	78
Figura 3.43	Interfaz de Órdenes de Compra.....	78
Figura 3.44	Interfaz de Avisos en Menú de Administradores.....	79
Figura 3.45	Interfaz de Avisos.....	79
Figura 3.46	Interfaz Principal de los Gerentes.....	80
Figura 3.47	Interfaz Visualizar Reportes.....	80
Figura 3.48	Interfaz Visualizar Ordenes.....	81
Figura 3.49	Interfaz Contacto en Menú Gerentes.....	81
Figura 3.50	Ingresar al Sistema como Usuario.....	82
Figura 3.51	Ingresar al Sistema como Administrador.....	83
Figura 3.52	Ingresar al Sistema como Gerente.....	83
Figura 3.53	Mensaje de Error de Password.....	84
Figura 3.54	Mensaje de Error de Login.....	84
Figura 3.55	Como Realizar una Petición.....	85
Figura 3.56	Dar de Alta un Usuario.....	85
Figura 3.57	Registro de un Nuevo Usuario.....	86
Figura 3.58	Dar de Alta un Nuevo Equipo.....	86
Figura 3.59	Registro de un Nuevo Equipo.....	87
Figura 3.60	Eliminar Registros de un Usuario.....	87
Figura 3.61	Comprobar la baja de un Usuario.....	88
Figura 3.62	Eliminar Registros de un Equipo.....	88
Figura 3.63	Comprobar la baja de un Equipo.....	89
Figura 3.64	Actualizar Información en los Usuarios.....	89
Figura 3.65	Registro Actualizado de Usuario.....	90
Figura 3.66	Actualizar Información en los Equipos.....	90
Figura 3.67	Registro Actualizado de los Equipos.....	91
Figura 3.68	Prueba Búsqueda de Datos.....	91
Figura 3.69	Búsqueda Satisfactoria.....	92
Figura 3.70	Búsqueda sin Resultados.....	92
Figura 3.71	Elaboración de Reportes.....	93
Figura 3.72	Comprobación que reporte se Genero.....	93
Figura 3.73	Eliminación de Reportes.....	94
Figura 3.74	Elaboración de Órdenes de Compra.....	94
Figura 3.75	Comprobación que Orden de Compra se Genero.....	95
Figura 3.76	Interfaz de Actualizar Equipos.....	95
Figura 3.77	Eliminación de Órdenes de Compra.....	96
Figura 3.78	Comprobar que los avisos se Generaron.....	96
Figura 3.79	Eliminación de Avisos.....	96



---

---

## INTRODUCCIÓN

En el presente capítulo, se introduce el tema sistema web de gestión para el mantenimiento preventivo y correctivo de equipos de cómputo, exponiendo el problema que busca resolver, de manera que se pueda colaborar con la comunidad empresarial.

Se ha redactado este documento con el propósito de determinar los aspectos generales y especificaciones que seguirá la aplicación de desarrollo, del sistema web de gestión para el mantenimiento correctivo y preventivo de equipos de cómputo. El presente documento va dirigido a los usuarios, administradores y gerentes de la empresa en la cual está pensado este sistema de mantenimiento.

También, se listan las motivaciones que llevaron a hacer este trabajo y los objetivos que se intentan alcanzar en el mismo. Finalmente, se explica brevemente como está conformada su estructura en capítulos y el contenido de cada uno de ellos.

Las necesidades aquí planteadas pueden ser atendidas con la implementación de un Sistema eficiente, flexible para el control, acceso y gestión de los mantenimientos de los equipos, y permita el acceso a la información del sistema, a usuarios, gerentes y administradores de la empresa.



---

## **1.1 DESCRIPCION DEL PROYECTO**

El sistema web de mantenimiento preventivo y correctivo de equipos de cómputo dará la oportunidad, al administrador del sistema de llevar un control acerca de los mantenimientos de los equipos de cómputo que se encuentren en la empresa, así como a los usuarios que estén registrados, de enviar peticiones al administrador sobre el funcionamiento de su equipo. Este sistema no realizara el mantenimiento físico, solo se llevara un control acerca de estos mantenimientos y gestión de las peticiones de los usuarios. El sistema permitirá a los usuarios ingresar con su nombre de usuario y contraseña, así como realizar las peticiones, que será llenando un formulario con una descripción breve en forma de texto de lo que requiere su equipo ya sea del tipo (red, software hardware u otra) y con una prioridad que podrá ser (baja, normal y urgente).

El sistema permitirá al administrador ingresar con su nombre de usuario y contraseña a revisar las peticiones realizadas por los usuarios conforme el grado de prioridad, una vez que el administrador termine de revisar el equipo tendrá que cerrar estas peticiones ya resueltas y enviar un aviso al usuario de que su equipo está listo. El administrador también tendrá otras funciones que el sistema le permitirá realizar, como elaborar las órdenes de compra del departamento de Soporte Técnico enviárselos al gerente general y una vez que esté apruebe las ordenes, se podrán surtir los materiales que se requieran para los mantenimientos, el administrador también deberá enviar avisos a los usuarios de las fechas que se tengan disponibles para los mantenimientos, una vez que el administrador termine de realizar los mantenimientos generales. El sistema le permitirá elaborar los reportes de esos mantenimientos y así tener un resultado sobre el estado de los equipos de cómputo.

El sistema también permitirá al gerente ingresar con su nombre de usuario y contraseña a visualizar los reportes de los mantenimientos que se realizaron y así tener un mejor control de estos equipos, también podrá visualizar las órdenes de compra que fueron elaboradas y enviadas por el administrador.

## **1.2 ANTECEDENTES DEL PROYECTO**

En los últimos años, en la empresa en la que laboro de nombre Gonvauto Puebla S.A. de C.V empresa dedicada a hacer los troquelados de la empresa Volkswagen, no se tiene un control acerca de los mantenimientos preventivos y correctivos de los equipos de cómputo que se encuentran dentro de la misma, una charla entre el gerente general y los gerentes de áreas me motivaron a trabajar en solucionar dicho problema y en pensar a realizar un sistema web de gestión para el mantenimiento correctivo y preventivo de computadoras, así dando mi punto de vista a mi gerente general sobre este sistema aprobó satisfactoriamente el proyecto de tesis que presento y se pueda convertir en un instrumento de uso diario para los usuarios de nuestra empresa. Por tanto, el sistema tendrá las siguientes metas:



- 
- El sistema sea una herramienta de uso diario para los usuarios de la empresa.
  - El sistema agilice la gestión del mantenimiento de equipos.

### **1.3 OBJETIVOS GENERALES Y ESPECIFICOS DEL PROYECTO**

Los objetivos que la presente tesis intenta alcanzar son los siguientes:

- Introducir un mejor manejo y control sobre los mantenimientos de equipos de cómputo de la empresa antes mencionada.
- Satisfacer los requerimientos que demande el cliente.
- Basarse en una metodología más factible para implementar el sistema.
- Hacer extensivos a la comunidad los resultados y conclusiones obtenidos, como así también a aquellos organismos posiblemente interesados.
- Aprobar satisfactoriamente este proyecto de tesis.

### **1.4 RESULTADOS ESPERADOS**

Entre los principales resultados que se esperan obtener con el desarrollo del sistema:

- Ahorrar el tiempo en que una petición de parte de los usuarios se resuelva lo más pronto posible.
- Llevar a cabo un mejor control cuando se realicen los mantenimientos de los equipos de cómputo de la empresa.
- Conocer el estado de cada equipo por parte del Gerente de la empresa.
- Llevar a cabo un control de los equipos que se encuentren en sus áreas.
- Promover entre los usuarios la cultura de cuidar los equipos de cómputo.
- Utilizar este sistema como una herramienta de uso diario para los usuarios de la empresa.



---

---

## 1.5 IMPACTO SOCIOECONOMICO

El sistema permitirá:

- Reducir costos en la empresa GONVAUTO PUEBLA S.A DE C.V. por ejemplo, se tendría un ahorro en papelería, ya que todos los documentos relacionados con el mantenimiento se llevaría de manera digital.
- Reducir tiempos de espera y solución entre las operaciones que realiza la empresa.
- Agilizar la gestión del mantenimiento de equipos, debido a que el control lo llevaría una sola persona a través del sistema.

## 1.6 APORTACIONES

La principal aportación, es un software de calidad que mejora y agiliza la gestión de los mantenimientos de la empresa así como los tiempos de espera y respuesta entre la comunidad de usuarios de la empresa.



---

---

# *CAPÍTULO II*

# *ESTADO DEL*

# *ARTE*



---

Este capítulo presenta un análisis comparativo de sistemas relacionados con el sistema de mantenimiento con la finalidad de determinar los aspectos importantes que debería de tener el sistema que se va a desarrollar, así como también se explican brevemente todos los conceptos que fundamentan el desarrollo de este proyecto de tesis.

## **2.1 ANALISIS COMPARATIVO**

Con en el Análisis comparativo se pretende realizar y analizar tres sistemas web que sean similares, al sistema de mantenimiento correctivo y preventivo de equipos de cómputo así como deducir porque es mejor el sistema de mantenimiento que los otros tres que se analizaran.

### **2.1.1 EASY MAINT SOFTWARE DE MANTENIMIENTO**

#### **DESCRIPCION:**

EasyMain es un sistema integrado para la Gestión del Mantenimiento, económico, y de fácil implementación. El software de Mantenimiento EasyMaint, es un programa diseñado para cualquier tipo de empresa, empresas industriales grandes y pequeñas de todo tipo, Plantas Industriales, Edificios, Hospitales, Hoteles, Flotas, entre otras.

EasyMaint ofrece una arquitectura abierta y es fácil de aprender, le ofrece reportes personalizables para ayudarle en la administración de todas sus operaciones, desde una solicitud de trabajo hasta el control total de la gestión del mantenimiento. Con su diseño intuitivo, EasyMaint provee una ventaja competitiva para la industria al ofrecer un completo control sobre todas las operaciones, eliminando el registro redundante de datos, y asegurando consistencia y veracidad en todo el proceso de Gestión, Control y Planificación del mantenimiento.

EasyMaint genera automáticamente las órdenes de trabajo para mantenimiento de sus equipos a partir del programa de mantenimiento y parámetros como fechas, de órdenes de trabajo previas, frecuencias de días, frecuencia en contadores de horas, unidades fabricadas. Kilómetros recorridos, entre otras.

EasyMaint programará las órdenes de trabajo de mantenimiento y calculará los costos estimados de cada orden de trabajo. Después cuando la orden de trabajo se termine EasyMaint calculará los costos reales.

#### **OBJETIVOS:**

- Reducir costos de producción
- Incrementar la capacidad y calidad de manufactura
- Cumplir con normas de calidad ISO y, QS
- Reducir reparaciones costosas y fallas de emergencia
- Optimizar niveles de inventario
- Tomar mejores decisiones



- 
- Reducir accidentes y mejorar la seguridad
  - Mejorar el servicio el cliente al entregar con calidad y oportunidad
  - Mejorar sus planes de Mantenimiento Preventivo MP, Predictivo y TPM
  - Crear Bitácoras y Análisis de Fallas
  - Establecer indicadores de Gestión

#### **VENTAJAS:**

- Económico
- Fácil de implementar
- Arquitectura abierta y fácil de aprender
- Consistencia y veracidad en procesos de gestión

#### **CLIENTES POTENCIALES:**

Cualquier tipo de empresa, empresas industriales grandes y pequeñas de todo tipo, Plantas Industriales, Edificios, Hospitales, Hoteles, Flotas, entre otras.

**ACTORES:** Administradores.

#### **DISEÑO DE LA INTERFAZ DE USUARIO:**

EasyMaint ofrece una interfaz abierta y es fácil de aprender, le ofrece reportes personalizables para ayudarle en la administración de todas sus operaciones, desde una solicitud de trabajo hasta el control total de la gestión del mantenimiento.

#### **COMPARACION CON EL SISTEMA DE MANTENIMIENTO PREVENTIVO:**

El sistema EasyMaint es un sistema que tiene funciones similares al sistema de mantenimiento pero no las mismas, es un sistema muy completo y maneja un enfoque a mantenimientos de todo tipo no solo el mantenimiento preventivo y el sistema de mantenimiento que aquí se presenta gestiona y controla además del mantenimiento de los equipos, las peticiones de los usuarios por lo que considero se tiene una mejor gestión.



---

---

## 2.1.2 SISTEMA SITEHELPDESK

### DESCRIPCION:

El sistema está diseñado para los servicios de apoyo interno, Gestión de servicios de TI o de atención a clientes en la web, monitor de correo electrónico integrado para el correo electrónico de dos vías de seguimiento dentro y fuera de la mesa de ayuda. Se integra con la auditoria de pcs líderes y herramientas de gestión de activos.

### OBJETIVOS:

- El establecimiento de un grupo de personas que den soporte a la consecución de las tareas del personal contratado.
- Identifica a todo el personal técnico dispuesto por la empresa para atender telefónica o personalmente (en el puesto de trabajo de cualquier empleado), cualquier duda informática sobre el software o hardware, así como reparación o reemplazo de piezas defectuosas.

### VENTAJAS:

- El conocimiento de Site Help Desk es amplio y no está limitado a un único producto, como puede ser el caso de los Call Center.
- Los Site Help Desk pueden ayudar con su diagnóstico a la resolución de problemas.
- El Site Help Desk deberá estar compuesto por un gerente o manager, un equipo de atención telefónica, los operadores, y otro equipo móvil de atención directa en el puesto de trabajo, también conocidos como backoffices.

### CLIENTES POTENCIALES:

Cualquier tipo de empresa, empresas industriales grandes y pequeñas de todo tipo, Plantas Industriales, Edificios, Hospitales, Hoteles, Flotas, entre otras.

**ACTORES:** Empleados, Gerentes, técnicos.

### DISEÑO DE LA INTERFAZ DE USUARIO:

El Site ofrece una interfaz amigable fácil entender y accionar, que le brinde tanto comodidad, como eficiencia.



---

---

## **COMPARACION CON EL SISTEMA DE MANTENIMIENTO PREVENTIVO:**

Comparando el sistema Site Help Desk con el sistema de mantenimiento preventivo, este sistema actualmente se utiliza en la empresa Gonvauto donde laboro, es un sistema amigable y fácil para la gestión de los mantenimientos, el problema que se tiene en la empresa, es que este sistema lo gestionan desde nuestra central en Madrid, España por lo que los tiempos de espera son muy tardados, las peticiones se realizan a destiempo y eso en ocasiones provoca conflictos tanto para la elaboración de los reportes como para la resolución de los mantenimientos, así como la saturación de nuestro servidor por el tráfico de datos, de aquí surge la idea de realizar nuestro sistema de mantenimiento preventivo y gestionar y controlar internamente nuestros mantenimientos sin descuidar a nuestra central en España.

### **2.1.3 SISTEMA SATNETWORK**

#### **DESCRIPCION:**

S.A.T. son las siglas de Sistema Administrativo para Talleres, dicho sistema fue pensado desde el principio como una solución para talleres de electrónica, sin embargo el mismo puede ser utilizado, personalizando los datos, para la administración de cualquier tipo de talleres (llámese línea blanca, automotriz, naval, etc.). El sistema ha demostrado ser altamente competitivo debido a su potencia, capacidad de almacenamiento de datos, seguridad, y sus cualidades de valor agregado entre las que se encuentran la consulta vía Web de ordenes de reparación, la copia de respaldo automatizada, el sistema de recuperación de fallos, el sistema de mailing, el sistema de chat local, el sistema de mensajería móvil SMS, etc. Gracias a que fue construido en una plataforma de 32 bits bajo el sistema operativo Windows y con lenguajes de programación visuales, el uso del programa resulta sumamente intuitivo y requiere un tiempo de aprendizaje mínimo, tarea ésta que se verá beneficiada por la utilización del manual de usuario, que detalla con precisión cada función del sistema.

#### **OBJETIVOS:**

- Tener control total del negocio
- Ofrecer un buen servicio a los clientes
- Agilizar la comunicación con los clientes.

#### **VENTAJAS:**

- Ahorrar tiempo y esfuerzo en el control de inventarios.
- Permitir hacer consultas a sus usuarios y revisar los estados reales de sus equipos desde internet.
- Utilizar el sistema desde distintas terminales.
- Administrar ordenes de servicio



---

---

## **CLIENTES POTENCIALES:**

Agente de servicios autorizados, sin dejar de lado a las pequeñas y medianas empresas que prestan servicios técnicos en forma particular.

**ACTORES:** Empleados, Gerentes, técnicos.

## **DISEÑO DE LA INTERFAZ DE USUARIO:**

El sistema SAT está diseñado estructuralmente para ser fácilmente adaptable a todo tipo de talleres, en la de artículos electrónicos, electrodomésticos, computación, motos, autos, fotografía, telefonía, relojería, máquinas herramientas, aeronáutica, naval, fotocopiadoras y electro medicina. Además, gracias a las capacidades de configuración y adaptabilidad del sistema, se podrá fácilmente aplicar a cualquier otro rubro de reparaciones y ventas.

En SatNetwork son conscientes del constante avance tecnológico y los cambios en la interacción con los clientes, por tal motivo el sistema cuenta con las más avanzadas herramientas de comunicación, con las mismas brindan información de los estados de los equipos a los clientes directamente en el sitio web, utilizar mensajería SMS, o enviar e-mails automatizados. Además el sistema SAT es compatible con Windows Live Messenger® y con el marcador telefónico de Windows.

## **COMPARACION CON EL SISTEMA DE MANTENIMIENTO PREVENTIVO:**

Comparando el sistema SAT con el sistema de mantenimiento preventivo, es un sistema más elaborado que el de mantenimiento preventivo puesto que está enfocado a diferentes rubros, manejando mas conceptos y funciones, el sistema SAT tiene la misma idea y los mismos objetivos que se tienen con el sistema de mantenimiento, pero considero que no es lo que la empresa realmente necesita, el sistema de mantenimiento sin embargo también es un sistema completo aunque no se tenga todas las funciones del sistema SAT puesto que se satisfacen todos los requerimientos de nuestro cliente y eso lo hace un software de calidad.



---

## 2.2 INGENIERÍA DEL SOFTWARE

La **Ingeniería de Software** es una disciplina o área de la informática o Ciencias de la Computación que ofrece métodos o técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Hoy día es cada vez más frecuente la consideración de la Ingeniería de Software como una nueva área de la ingeniería, y el ingeniero de software comienza a ser una profesión implantada en el mundo laboral internacional, con derechos, deberes y responsabilidad desde de cumplir con una, ya reconocida consideración social en el mundo empresarial y por suerte, para estas personas con un brillante futuro.

La Ingeniería de software trata con áreas muy diversas de la informática y de las Ciencias de la Computación tales como, construcción de compiladores, sistemas operativos o desarrollos en Intranet o Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información aplicables a una infinidad de áreas, tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, el mundo del derecho, la red de redes Internet, redes Intranet y Extranet, etc.[1]

### 2.2.1 DEFINICIÓN DEL TÉRMINO INGENIERIA DE SOFTWARE

El término Ingeniería se define en el Diccionario de la Real Academia Española de la Lengua como un conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía y el término ingeniero se define como persona que profesa o ejerce la ingeniería. De igual manera la Real Academia de Ciencias Exactas, Físicas, Naturales de España define el término Ingeniería como conjunto de conocimientos y técnicas cuya aplicación permite la utilización racional de los materiales y de los recursos naturales, mediante invenciones, construcciones y mediante otras realizaciones provechosas para el hombre.

Evidentemente la Ingeniería de Software es una nueva Ingeniería, parece lógico que reúna las propiedades citadas en las definiciones anteriores. Sin embargo, ni el Diccionario de la Real Academia Española de la Lengua ni la Real Academia Española de Ciencias han incluido todavía el término en sus últimas ediciones; en consecuencia vamos a recurrir para su definición más precisa a algunos de los autores más acreditados que comenzaron en su momento a utilizar el término o bien en las definiciones dadas por organismos internacionales profesionales de prestigio tales como IEEE o ACM. Así, se ha seleccionado las siguientes definiciones de Ingeniería de Software.

**Definición.** Ingeniería de Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadoras y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software [1].



---

## 2.3 SOFTWARE

El software de computadora es el producto que diseñan y construyen los ingenieros de software. Esto abarca programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, documentos que comprenden formularios virtuales e impresos y datos que combinan números, texto y también incluyen representaciones de información de audio, video, imágenes.

Construir software de computadora es como si construyéramos cualquier otro producto satisfactorio, aplicando un proceso que conduce a un resultado de alta calidad que satisface las necesidades de la gente que usará el producto.

El software es un elemento del sistema que es lógico en lugar de físico. Por tanto el software tiene unas características considerablemente distintas a las del hardware. [2].

### 2.3.1 CARACTERÍSTICAS DEL SOFTWARE

- *El software se desarrolla, no se fabrica en un sentido clásico.* Aunque existen similitudes entre el desarrollo del software y la construcción de hardware, ambas actividades son fundamentalmente diferentes. En ambas actividades la buena calidad se adquiere mediante un buen diseño y dependen de las personas, pero la relación entre las personas dedicadas y el trabajo realizado es completamente diferente para el software. Las dos actividades requieren de la construcción de un producto pero los enfoques son diferentes. Los costes del software se encuentran en la Ingeniería. Esto significa que los proyectos de software no se pueden gestionar como si fueran proyectos de fabricación.
- *El software no se estropea.* El software no es susceptible a los males del entorno que hacen que el hardware se estropee. Cuando un componente de hardware se estropea, se sustituye por una pieza de repuesto. No hay piezas de repuesto para el software, cada fallo en el software indica un error en el diseño máquina ejecutable, por tanto, el mecanismo del software tiene una complejidad considerablemente mayor que la del mantenimiento del hardware.
- *Aunque la industria tiende a ensamblar componentes, la mayoría del software se construye a medida.* A medida que la disciplina del software evoluciona, se crea un grupo de componentes de diseño estándar. El componente de software debería diseñarse e implementarse para que pueda volver a ser utilizado en muchos programas diferentes. Se ha extendido la visión de reutilización para abarcar no solo los algoritmos, sino también estructuras de datos. Los componentes reutilizables modernos encapsulan tanto datos como procesos que se aplican a los datos, permitiendo al ingeniero del software crear nuevas aplicaciones a partir de las partes reutilizables.



---

### 2.3.2 APLICACIONES DEL SOFTWARE

El software puede aplicarse en cualquier situación en que se haya definido previamente un conjunto específico de pasos procedimentales. El contenido y el determinismo de la información son factores importantes a considerar para determinar la naturaleza de una aplicación de software. El contenido se refiere al significado y a la forma de información de entrada y salida.

- *Software de sistemas.* El software de sistemas es un conjunto de programas que han sido desarrollados para servir a otros programas.
- *Software de tiempo real.* El software que mide/analiza/controla sucesos del mundo real conforme ocurren, se denomina de tiempo real.
- *Software de gestión.* El procesamiento de información comercial constituye la mayor de las áreas de aplicación del software.
- *Software de ingeniería y científico.* El software de ingeniería y científico está caracterizado por los algoritmos de manejo de números.
- *Software empotrado.* Reside en memoria de sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo.
- *Software de computadoras personales.* El procesamiento de textos, las hojas de cálculo, los gráficos por computadora, multimedia, gestión de base de datos, aplicaciones financieras, redes de acceso a base de datos externa.
- *Software de Inteligencia Artificial.* Hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo.

### 2.3.3 VISION GENERAL DE INGENIERÍA DE SOFTWARE

El trabajo que se asocia a la ingeniería de software se puede dividir en tres fases genéricas, con independencia del área de aplicación, tamaño, complejidad del proyecto. Cada fase se encuentra con una o varias cuestiones de las destacadas anteriormente.

La fase de definición se centra en el qué. Es decir, durante la definición, el que desarrolla el software intenta identificar que información ha de ser procesada, que función y rendimiento se desea, que comportamiento del sistema, que interfaces van a ser establecidas, que restricciones de diseño existen y que criterios de validación se necesitan para definir un sistema correcto.

La fase de desarrollo se centra en el cómo. Es decir, durante el desarrollo de un ingeniero del software intenta definir como han de diseñarse las estructuras de datos, como han de implementarse la función dentro de una arquitectura de software, como han de implementarse los detalles procedimentales, como han de caracterizarse



interfaces, como ha de traducirse el diseño de un lenguaje de programación y como ha de realizarse la prueba.

La fase de mantenimiento se centra en el cambio que se va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidas a las mejoras producidas por los requisitos cambiantes del cliente. Durante la fase de mantenimiento se encuentran cuatro tipos de cambios: corrección, adaptación, mejora, prevención.

### 2.3.4 MODELOS DE PROCESO DE SOFTWARE

Para resolver los problemas reales de una industria, un ingeniero del software o equipo de ingenieros deben incorporar una estrategia de desarrollo que acompañe al proceso, métodos, capas de herramientas y las fases genéricas. Esta estrategia a menudo se llama modelo de proceso o paradigma de ingeniería del software. Se selecciona un modelo de proceso para la ingeniería del software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, los controles y entregas que se requieren.

Todo el desarrollo del software se puede caracterizar como un bucle de resolución de problemas. En el que se encuentran cuatro etapas distintas: status quo, definición de problemas, desarrollo técnico e integración de soluciones. Status quo, representa el estado actual del suceso, la definición de problemas, identifica el problema específico a resolver, el desarrollo técnico, resuelve el problema a través de la aplicación de alguna tecnología, integración de soluciones, ofrece los resultados. La figura 2.1 muestra las fases de un bucle de resolución de problemas.

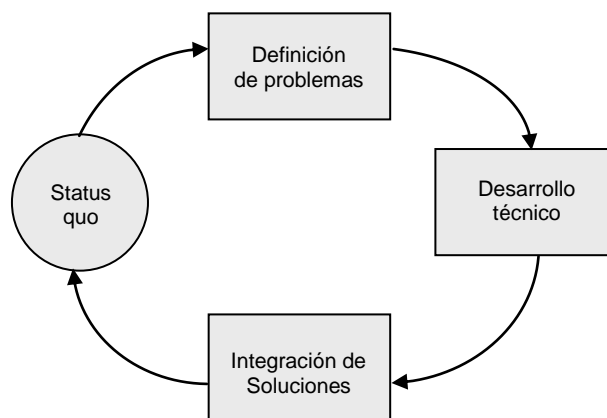


Figura 2.1 Las fases de un bucle de resolución de problemas



## 2.4 METODOLOGIA

Se usó el modelo lineal secuencial para organizar el proceso de desarrollo del sistema, es decir, se realizó siguiendo las etapas de análisis, diseño, codificación y prueba así como también la metodología UML y sus herramientas.

### 2.4.1 EL MODELO LINEAL SECUENCIAL

En la figura 2.2 se ilustra el modelo lineal secuencial para la ingeniería de software. El cliente utiliza el producto central (o sufre la revisión detallada). Como un resultado de utilización y/o evaluación, se desarrolla un plan para el incremento siguiente. El plan afronta la modificación del producto central a fin de cumplir mejor las necesidades del cliente y la entrega de cada incremento, hasta que se elabore el producto completo.

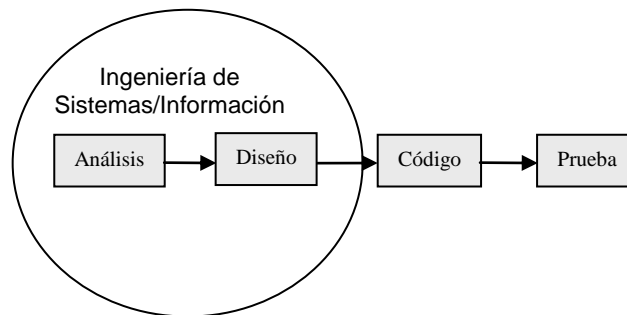


Figura 2.2 Modelo Lineal secuencial

Ingeniería y modelado de Sistemas/Información. Como el software siempre forma parte de un sistema mas grande, el trabajo comienza estableciendo requisitos de todos lo elementos del sistema y asignando al software algún subgrupo de estos requisitos.

Análisis de Requisitos. El proceso de reunión de requisitos se intensifica y se centra especialmente en el software.

Diseño. Este es realmente un proceso de muchos pasos que se centra en estructura de datos, arquitectura de software, representación de interfaz y detalle procedimental.

Generación de código. El diseño se debe traducir en una forma legible y por la máquina.

Pruebas. Una vez que se ha generado el código empiezan las pruebas del programa



---

---

Inicialmente se realizó la recopilación de la información y el análisis de requerimientos para comprender las necesidades de los usuarios, y así llegar a elegir la forma en que se dará solución al problema, y la creación del documento de requerimientos.

Posteriormente se usó el modelo entidad-relación, para definir la estructura, integridad, manipulación de los datos.

A continuación, se realizó la codificación en donde se genera el programa que compone el sistema, las interfaces, así mismo a la Base de Datos.

Finalmente, se realizaron las pruebas, para verificar que todos los componentes del sistema funcionan correctamente y asegurar que la entrada produce los resultados que realmente se requerían.

## **2.5 EL LENGUAJE DE MODELADO UNIFICADO (UML)**

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. [3]

En función de las diferentes vistas del modelo, en UML se definen los siguientes diagramas gráficos:

- Diagrama de casos de uso
- Diagrama de clases
- Diagramas de comportamiento:
  - Diagrama de estados
  - Diagrama de actividades
- Diagramas de interacción:
  - Diagrama de secuencia
  - Diagrama de colaboración
- Diagramas de implementación
  - Diagrama de componentes
  - Diagrama de despliegue

Por otro lado podemos ver el modelo de una forma estática o de una forma dinámica. Estas perspectivas nos dan la siguiente clasificación:



- **Modelo estático (estructural):**
  - Diagrama de despliegue
  - Diagrama de componentes
  - Diagrama de clases
  - Diagrama de objetos
  
- **Modelo dinámico (comportamiento):**
  - Diagrama de estados
  - Diagrama de actividades
  - Diagrama de secuencia
  - Diagrama de colaboración
  - Diagrama de casos de uso

Los cuatro diagramas estructurales de UML permiten visualizar, especificar, construir y documentar los aspectos estáticos de un sistema:

Un **diagrama de clases** presenta un conjunto de clases, interfaces y colaboraciones, y las relaciones entre ellas.

Un **diagrama de objetos** representa un conjunto de objetos y sus relaciones. Se utilizan para describir estructuras de datos, instantáneas de los elementos encontrados en los diagramas de clases. Cubre los mismos aspectos que los diagramas de clases pero desde una perspectiva de casos reales o prototípicos.

Un **diagrama de componentes** muestra un conjunto de componentes y sus relaciones. Los diagramas de componentes se utilizan para describir la vista de implementación estática de un sistema. Los diagramas de componentes se relacionan con los diagramas de clases, ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones.

Un **diagrama de despliegue** muestra un conjunto de nodos y sus relaciones. Los diagramas de despliegue se utilizan para describir la vista de despliegue estática de un sistema. Los diagramas de despliegue se relacionan con los diagramas de componentes, ya que un nodo normalmente incluye uno o más componentes.

Los cinco diagramas de comportamiento de UML se emplean para visualizar, especificar, construir y documentar los aspectos dinámicos de un sistema.

Los **diagramas de casos de uso** organizan los comportamientos del sistema. Un diagrama de caso de uso representa un conjunto de casos de uso y actores (un tipo especial de clases) y sus relaciones. [4]

Los **diagramas de interacción**: Se da este nombre colectivo a los diagramas de secuencia y los diagramas de colaboración. Ambos diagramas son isomorfos, es decir, se puede convertir de uno a otro sin pérdida de información.



---

Un **diagrama de secuencia** es un diagrama de interacción que resalta la ordenación temporal de los mensajes. Un diagrama de secuencia presenta un conjunto de objetos y los mensajes enviados y recibidos por ellos., Los objetos suelen ser instancias con nombre o anónimas de clases, pero también pueden representarse instancias de otros elementos, tales como colaboraciones, componentes y nodos.

Un **diagrama de colaboración** es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de colaboración muestra un conjunto de objetos, enlaces entre estos objetos y mensajes enviados y recibidos por estos objetos. Los objetos normalmente son instancias con nombre o anónimas de clases, pero también pueden representar instancias de otros elementos, como colaboraciones, componentes y nodos.

Un **diagrama de estado** representa una máquina de estados, constituida por estados, transiciones, eventos y actividades. Son especialmente importantes para modelar el comportamiento de una interfaz, una clase o una colaboración. Los diagramas de estados resaltan el comportamiento dirigido por eventos de un objeto, lo que es especialmente útil al modelar sistemas reactivos.

Un **diagrama de actividades** es un tipo especial de diagrama de estados que muestra el flujo secuencial o ramificado de actividades en un sistema. Conjunto de actividades, Son especialmente importantes para modelar la función del sistema, así como para resaltar el flujo de control entre objetos.

### 2.5.1 Diagramas de Casos de Uso

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones). Un caso de uso, denotando un requisito funcional exigido al sistema, se representa en el diagrama por una elipse y un nombre significativo. [4]

Un diagrama de casos de uso consta de los siguientes elementos:

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

#### Elementos

- **Actor:**

Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.



Como ejemplo a la definición anterior, tenemos el caso de un sistema de ventas en que el rol de Vendedor con respecto al sistema puede ser realizado por un Vendedor o bien por el Jefe de Local.

- **Caso de Uso:**

Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

- **Relaciones:**

- Asociación

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

- Dependencia o Instanciación

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

- Generalización

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>). Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

extends: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

uses: Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

De lo anterior cabe mencionar que tiene el mismo paradigma en diseño y modelamiento de clases, en donde está la duda clásica de usar o heredar. En la figura 2.3 se muestra un ejemplo de actores de un caso de uso.



Figura 2.3 Ejemplo de Actores



## 2.5.2 Diagramas de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido.

Un diagrama de clases está compuesto por los siguientes elementos:

- **Clase:** atributos, métodos y visibilidad.
- **Relaciones:** Herencia, Composición, Agregación, Asociación y Uso.

### Elementos

- **Clase**

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella se puede modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.).

En UML, una clase es representada por un rectángulo que posee tres divisiones como se muestra en la figura 2.4:

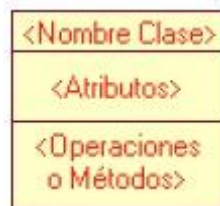


Figura 2.4 Esquema de una clase

En donde:

- **Superior:** Contiene el nombre de la Clase.
- **Intermedio:** Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).
- **Inferior:** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

## 2.5.3 Diagramas de Estados

Los diagramas de estado describen gráficamente los **eventos** y los **estados** de los objetos. Los diagramas de estado son útiles, entre otras cosas, para indicar los eventos del sistema en los casos de uso.



## Elementos

- Un **evento** es un acontecimiento importante a tomar en cuenta para el sistema.
- Un **estado** es la condición de un objeto en un momento determinado: el tiempo que transcurre entre eventos.
- Una **transición** es una relación entre dos estados, e indica que, cuando ocurre un evento, el objeto pasa del estado anterior al siguiente.

En UML, los estados se representan mediante óvalos. Las transiciones se representan mediante flechas con el nombre del evento respectivo. Se acostumbra poner un estado inicial (círculo negro). Por ejemplo como se muestra en la figura 2.5:

### Teléfono

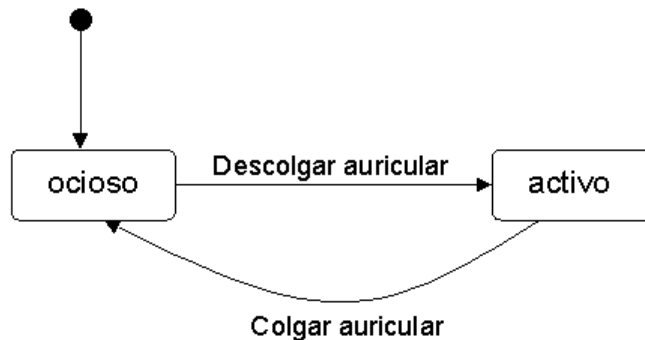


Figura 2.5 Ejemplo de Diagrama de Estado

Un diagrama de estado representa el ciclo de vida de un objeto: los eventos que le ocurren, sus transiciones, y los estados que median entre estos eventos.

En particular, es útil hacer diagramas de estado para describir la secuencia permitida de eventos en los casos de uso.

#### 2.5.4 Diagramas de Actividades

Un diagrama de actividades puede considerarse como un caso especial de un diagrama de estados en el cual casi todos los estados son estados acción (identifican una acción que se ejecuta al estar en él) y casi todas las transiciones evolucionan al término de dicha acción (ejecutada en el estado anterior). Un diagrama de actividades puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Permiten representar transiciones internas al margen de las transiciones o eventos externos.

La interpretación de un diagrama de actividades depende de la perspectiva considerada: en un diagrama conceptual, la actividad es alguna tarea que debe ser realizada; en un diagrama de especificación o de implementación, la actividad es un método de una clase. Generalmente se suelen utilizar para modelar los pasos de un algoritmo.



Los diagramas de actividad son similares a los diagramas de flujo procesales, con la diferencia de que todas las actividades están claramente unidas a objetos, siempre están asociados a una clase, a una operación o a un caso de uso. Soportan actividades tanto secuenciales como paralelas. La ejecución paralela se representa por medio de iconos de fork/espera, y en el caso de las actividades paralelas, no importa en qué orden sean invocadas (pueden ser ejecutadas simultáneamente o una detrás de otra).

Las actividades pueden formar jerarquías, lo que significa que una actividad puede estar formada de varias actividades «de detalle», en cuyo caso las transiciones entrantes y salientes deberían coincidir con las del diagrama de detalle.

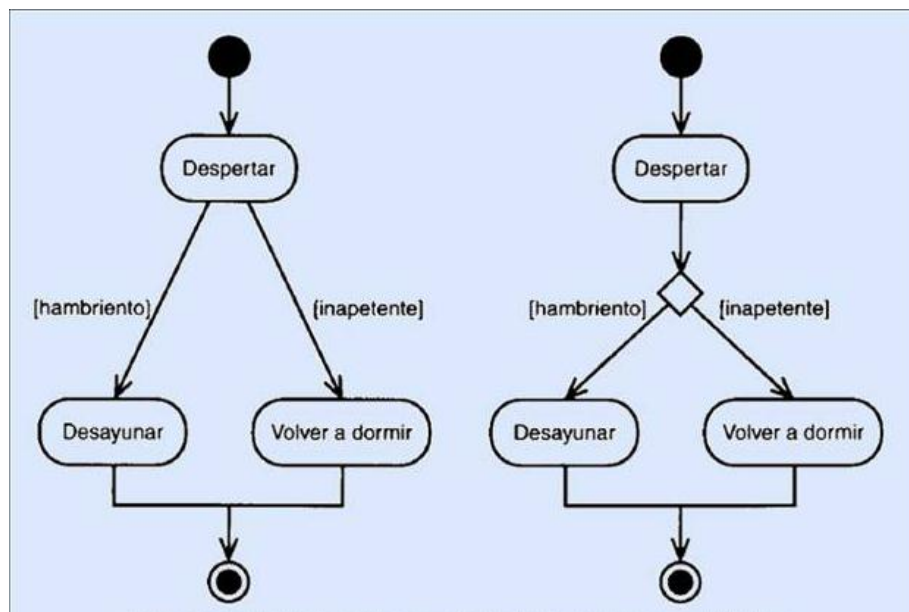


Figura 2.6 Diagrama de actividades

Un diagrama de Actividad demuestra la serie de actividades que deben ser realizadas en un caso de uso, así como las distintas rutas que pueden irse desencadenando en el caso de uso. Es muy similar en definición a un diagrama de flujo (típicamente asociado en el diseño de Software), estos no son lo mismo en la figura 2.6 se muestra un ejemplo de diagrama de Actividades.

## Elementos

- **Inicio:** El inicio de un diagrama de actividad es representado por un círculo de color negro sólido.
- **Actividad:** Una actividad representa la acción que será realizada por el sistema la cual es representada dentro de un ovalo.
- **Transición:** Una transición ocurre cuando se lleva a cabo el cambio de una actividad a otra, la transición es representada simplemente por una línea con una flecha en su terminación para indicar dirección.



- **Ramificación (Branch):** Una ramificación ocurre cuando existe la posibilidad que ocurra más de una transición (resultado) al terminar determinada actividad. Este elemento es representado a través de un rombo.
- **Unión (Merge):** Una unión ocurre al fusionar dos o más transiciones en una sola transición o actividad. Este elemento también es representado a través de un rombo.
- **Expresiones Resguardadas (Guard Expressions):** Una expresión resguardada es utilizada para indicar una descripción explícita acerca de una transición. Este tipo de expresión es representada mediante corchetes [...] y es colocada sobre la línea de transición.
- **Fork :** Un fork representa una necesidad de ramificar una transición en más de una posibilidad. Aunque similar a una ramificación (Branch) la diferencia radica en que un fork representa más de una ramificación obligada, esto es, la actividad debe proceder por ambos o más caminos, mientras que una ramificación (Branch) representa una transición u otra para la actividad (como una condicional). Un fork es representado por una línea negra sólida, perpendicular a las líneas de transición.
- **Join :** Una join ocurre al fusionar dos o más transiciones provenientes de un fork, y es empleado para dichas transiciones en una sola, tal y como ocurría antes de un fork .
- **Fin:** El fin de un diagrama de actividad es representado por un círculo, con otro círculo concéntrico de color negro sólido.
- **Canales (Swimlanes):** En determinadas ocasiones ocurre que un diagrama de actividad se expanda a lo largo de más de un entidad o actor, cuando esto ocurre el diagrama de actividad es particionada en canales (swimlines), donde cada canal representa la entidad o actor que está llevando a cabo la actividad.

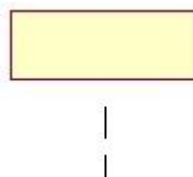
### 2.5.5 Diagramas de Secuencia

En un diagrama de secuencia participan varios de los objetos o clases que forman parte del programa y se describen las llamadas que van haciendo unos a otros para realizar una tarea determinada.

Se realiza un diagrama de secuencia por cada caso de uso o para una parte de un caso de uso (lo que llamo subcaso de uso). Por ejemplo en un ajedrez, se realizan diagramas de secuencia para "jugar partida" o bien para partes de "jugar partida", como puede ser "mover pieza".

#### Elementos

- **Objeto/Actor:**



El rectángulo representa una instancia de un Objeto en particular, y la línea punteada representa las llamadas a métodos del objeto.

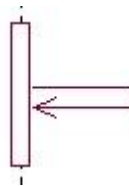


- **Mensaje a Otro Objeto:**



Se representa por una flecha entre un objeto y otro, representa la llamada de un método (operación) de un objeto en particular.

- **Mensaje al Mismo Objeto:**



No sólo llamadas a métodos de objetos externos pueden realizarse, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio.

En una primera fase de diseño se pueden definir clases grandes y ficticias, que representen un paquete/librería o, si el programa está compuesto por varios ejecutables corriendo a la vez, incluso clases que representen un ejecutable.

En cambio en una fase avanzada, se diseña el programa con los detalles necesarios para que los programadores tengan claro que métodos van a implementar, con sus parámetros respectivos. Incluso si es un diagrama para presentar al cliente, se puede realizar un diagrama de secuencia en el que sólo salga el actor "jugador" y una única clase "juego ajedrez" que represente el programa completo, de forma que el cliente vea qué datos y en qué orden los tiene que introducir en el programa y vea qué salidas y resultados le va a proporcionar el mismo.

El siguiente es un diagrama de secuencia del ejemplo del ajedrez a un nivel de diseño muy preliminar se muestra en la figura 2.7.

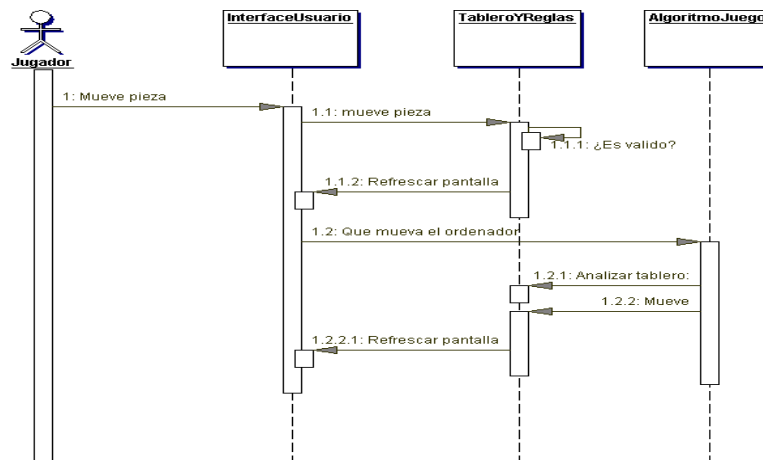


Figura 2.7 Ejemplo Diagrama de Secuencia

## 2.5.6 Diagramas de Colaboración

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas entorno a los objetos y los enlaces entre ellos.

### Elementos

- **Objeto:** Se representa con un rectángulo que contiene el nombre y la clase del objeto en un formato nombreObjeto: nombreClase.
- **Enlaces:** Un enlace es una instancia de una asociación en un diagrama de clases. Se representa como una línea continua que une a dos objetos, acompañada por un número que indica el orden dentro de la interacción. Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones. Se pueden utilizar estereotipos para indicar si el objeto que recibe el mensaje es un atributo, un parámetro de un mensaje anterior, si es un objeto local o global.
- **Flujo de mensajes:** Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cerca de un enlace.
- **Marcadores de creación y destrucción de objetos:** Puede mostrarse en la gráfica qué objetos son creados y destruidos, agregando una restricción con la palabra new o delete respectivamente.
- **Objeto compuesto:** Es una representación alternativa de un objeto y sus atributos. En esta representación se muestran los objetos contenidos dentro del rectángulo que representa al objeto que los contiene. Un ejemplo es el objeto Windows
- **Patrón de diseño:** Un diagrama de colaboración puede especificar un contrato entre objetos, parte esencial para la descripción de un patrón de diseño. Este diagrama contiene todos los elementos citados de un diagrama de colaboración, dejando libres posiblemente los tipos exactos de algunos objetos o con nombres



genéricos para los mensajes. Una “instanciación” del patrón se representa como una elipse unida mediante flechas punteadas a los objetos o clases que participan realmente en el patrón. Estas flechas pueden tener roles, indicando cuál es el papel de cada elemento dentro del patrón. Por ejemplo, una instanciación del patrón Observer.

- **Contexto:** Un contexto es una vista de uno o más elementos dentro del modelo que colaboran en el desarrollo de una acción. Se usa para separar los demás elementos en el modelo de este problema en particular y darle énfasis. Puede mostrar sólo los detalles relevantes de las clases u objetos que contiene, para resaltar su utilidad.

Los diagramas de secuencia proporcionan una forma de ver el escenario en un orden temporal - qué pasa primero, qué pasa después -. Los clientes entienden fácilmente este tipo de diagramas, por lo que resultan útiles en las primeras fases de análisis.

Por el contrario los diagramas de colaboración proporcionan la representación principal de un escenario, ya que las colaboraciones se organizan entorno a los enlaces de unos objetos con otros. Este tipo de diagramas se utilizan más frecuentemente en la fase de diseño, por ejemplo el caso de uso para servir un café como se muestra en la figura 2.8.

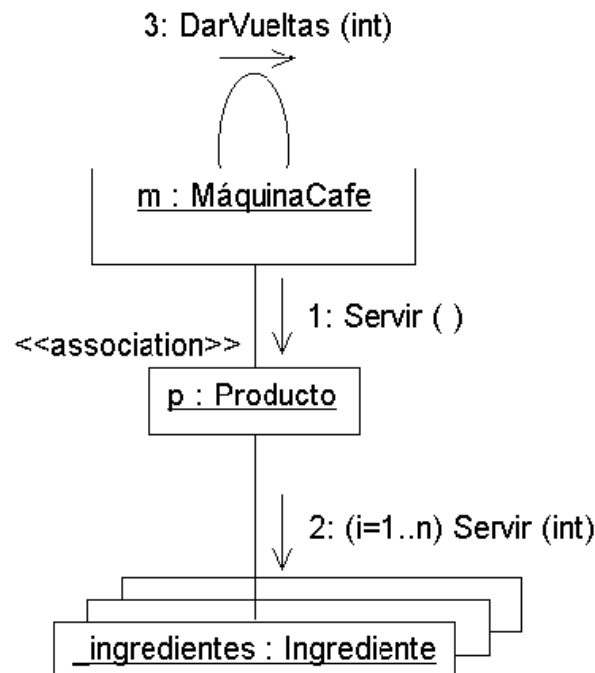


Figura 2.8 Ejemplo Diagrama de Colaboración

A diferencia de otras notaciones que muestran tanto el estado y el comportamiento de la clase en el diagrama de clases, UML separa el comportamiento de las clases en los diagramas de colaboración. Los diagramas de clase de UML no incluyen flujo de mensajes entre clases, es por esto que los diagramas de colaboración se deben crear en paralelo con los diagramas de clases. Aunque se puede indicar el orden del flujo de



---

mensajes en un diagrama de colaboración numerando los mensajes, no se suele hacer, ya que para este propósito son mejores los diagramas de secuencia.

## 2.6 BASE DE DATOS

Una **base o banco de datos** es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

La creación de una base de datos debe ser realizada cuidadosamente procurando cumplir con una serie de objetivos (entre otros) que se detallan a continuación:

- Permitir un fácil acceso a la información. El sistema debe ser facilitador en alto rendimiento, la velocidad es un factor esencial así como la consistencia de los datos.
- Evitar la redundancia. La información almacenada ocupará irremediablemente un espacio en memoria por lo cual es de vital importancia eliminar la posibilidad de almacenar datos repetidos ya que adicionalmente podrían llevarnos a inconsistencias en la información.

Un sistema de gestión de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información acerca de una empresa particular. El primer objetivo de un SGBD es proporcionar un entorno que sea tanto práctico como eficiente de usar en la recuperación y el almacenamiento de información de la base de datos. Las propiedades de los sistemas gestores de bases de datos se estudian en informática. [5]

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Los sistemas de bases de datos (SBD) se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la misma. Los SBD deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o del intento de acceso sin autorización.

Anteriormente se utilizaba el sistema de procesamiento de archivos en el cual toda la información de la empresa se almacenaba en archivos permanentes manejados por un sistema operativo convencional, se tenían que escribir diversos programas de aplicación para extraer registros y añadir registros a los archivos adecuados.



---

Los principales inconvenientes de la organización en un sistema de procesamiento de archivos son:

*Redundancia e inconsistencia de datos.* Como los programas y archivos son creados por diferentes personas a través del tiempo, estos pueden tener diferentes formatos y escritos en diferentes lenguajes de programación y la información puede estar duplicada en los diversos lugares. Esto conduce a un almacenamiento y coste de acceso más altos y puede conducir a inconsistencia de datos.

*Dificultad en el acceso de datos.* Algunas peticiones no previstas en el sistema original, no hay un programa de aplicación a la mano para hacerlo. Sin embargo hay programas para mostrar la lista total y las opciones son: primero de esa lista sacar la información manualmente y segundo, mandar a un programador a hacer un programa que haga esa consulta. Y por cada consulta se debería hacer lo mismo.

*Aislamiento de datos.* La creación de nuevos programas de aplicación es muy difícil ya que los archivos pueden estar en diversos formatos y se tiene que recopilar la información dispersa en cada uno de los archivos.

*Problemas de integridad.* Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de restricciones de consistencia. Por ejemplo, el saldo de ciertos tipos de cuentas bancarias no puede nunca ser inferior a una cantidad determinada. Los desarrolladores hacen cumplir esas restricciones en el sistema añadiendo el código correspondiente en los diversos programas de aplicación.

*Problemas de atomicidad.* Los sistemas informáticos, como cualquier otro dispositivo mecánico o eléctrico, está sujeto a fallos. En muchas aplicaciones es crucial asegurar que, si se produce algún fallo, los datos se restauren al estado consistente que existía antes del fallo. Es decir que si al existir algún fallo en el sistema de datos y el usuario esta haciendo alguna transacción, que ésta ocurra en su totalidad o que no ocurra en absoluto. Resulta difícil asegurar la atomicidad en los sistemas convencionales de procesamiento de archivos.

*Problemas de seguridad.* No todos los usuarios de un sistema de base de datos deben poder acceder a todos los datos. Por ejemplo, en un sistema bancario, el personal de nomina solo necesita ver la parte de la base de datos que contiene información acerca de los diferentes empleados del banco. No necesitan tener acceso a la información acerca de las cuentas de clientes. Pero, como los programas de aplicación se añaden al sistema de procesamiento de datos de una forma ad hoc, es difícil cumplir tales restricciones de seguridad.

Aunque las bases de datos pueden contener muchos tipos de datos, algunos de ellos se encuentran protegidos por las leyes de varios países. Por ejemplo en España, los datos personales se encuentran protegidos por la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD). [5]



---

## 2.6.1 ABSTRACCIÓN DE DATOS

El propósito principal de un sistema de bases de datos permite acceder y modificar estos archivos. El propósito principal de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

Para que el sistema sea útil, debe recuperar los datos eficientemente. Esto ha conducido al diseño de estructuras complejas para la representación de los datos en la base de datos. Los desarrolladores esconden la complejidad a través de los niveles de abstracción para simplificar la interacción de los usuarios con el sistema.

*Nivel físico.* El nivel más bajo de abstracción, describe como se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.

*Nivel lógico.* El siguiente nivel de abstracción describe como se almacenan en la base de datos y que relaciones existen entre esos datos.

*Nivel de vistas.* Este nivel sólo describe parte de la base de datos completa. A muchos usuarios no les importará el nivel lógico, solo querrán ver parte de la Información. Para la simplificación de la interacción del usuario se crea un nivel de vistas.

Los componentes de gestión de almacenamiento proporciona la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y el envío de consultas al sistema. El gestor de almacenamiento tiene:

- Gestor de autorizaciones de integridad que comprueba que se satisfagan las ligaduras de integridad y la autorización de los usuarios para acceder a los datos.
- Gestor de transacciones que asegura que la base de datos quede en un estado consistente a pesar de los fallos del sistema y que las ejecuciones de transacciones concurrentes ocurran sin conflictos.
- Gestor de archivos que gestiona la reserva de espacio de almacenamiento de disco y las estructuras de los datos usadas para representar la información almacenada en el disco.
- Gestor de memoria intermedia que es responsable de traer los datos del disco de almacenamiento a memoria principal y decidir qué datos traer a memoria caché.

También se necesitan diversas estructuras de datos como parte de la implementación física del sistema.

- Archivos de datos que almacenan la base de datos en sí.



- 
- Diccionario de datos que almacena los metadatos acerca de la estructura de la base de datos.
  - Índices que proporcionan acceso rápido a elementos de datos que tienen valores particulares.
  - Datos estáticos que almacenan información estadística sobre los datos en la base de datos, el procesador de consultas usa esta información para seleccionar las formas eficientes para ejecutar una consulta. [5]

## 2.6.2 USUARIOS DE LA BASE DE DATOS

Hay tres diferentes tipos de usuarios de un Sistema de Base de Datos, diferenciados por la forma en que ellos esperan interactuar con el sistema.

- Programadores de aplicaciones. Responsables de escribir los programas de aplicación de base de datos en algún lenguaje de alto nivel. Estos programas acceden a la base de datos emitiendo la solicitud apropiada al manejador de base de datos. Los programas pueden ser aplicaciones convencionales por lotes o pueden ser aplicaciones en línea, cuyo propósito es permitir al usuario final el acceso a la base de datos desde una estación de trabajo o terminal en línea.
- Usuarios finales. Interactúan con el sistema desde estaciones de trabajo o terminales en línea. Un usuario final puede acceder a la base de datos a través de las aplicaciones en línea mencionadas anteriormente, o bien puede usar una interfaz proporcionada como parte integral del software del sistema de base de datos.
- Administrador de la base de datos (DBA). Es la persona que tiene el control central del sistema. [6]

## 2.6.3 TIPOS DE BASES DE DATOS

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

*Bases de datos estáticas.* Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

*Bases de datos dinámicas.* Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.



---

*Bases de datos bibliográficas.* Sólo contienen un representante de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

*Bases de datos de texto completo.* Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

*Directorios.* Un ejemplo son las guías telefónicas en formato electrónico. [6]

## 2.7 MODELO DE DATOS

**Modelo.** Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un Sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos. [7]

- Modelo lógico basado en objetos.
- Modelo lógico basado en registro.
- Modelo físico de datos.

### 2.7.1 MODELO LÓGICO BASADO EN OBJETOS

Se basan para describir datos en el nivel conceptual y de visión. Se caracterizan porque proporcionan una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Los modelos son:

- a) Entidad Relación
- b) Orientado a Objeto
- c) Binario
- d) Semántica de Datos
- e) Infológico



f) Funcional de Datos

## 2.7.2 MODELO ENTIDAD – RELACIÓN (E-R)

Denominado por sus siglas como E-R. Este modelo representa la realidad a través de un esquema gráfico empleando la terminología de **entidades**, que son objetos, que existen y son los elementos principales que identifican, el problema a resolver con el diagrama y se distinguen de otros por sus características particulares denominadas **atributos**, el enlace que rige la unión de las entidades está representada por la **relación** del modelo.

Las entidades pueden ser de dos tipos:

**Tangibles.** Son todos aquellos objetos físicos que podemos ver, tocar o sentir.

**Intangibles.** Todos aquellos eventos u objetos conceptuales que no podemos ver, aun sabiendo que existen.

La figura 2.9 muestra los símbolos usados en el diagrama E-R.

Descripción	Simbólicamente se representan
Entidades	
Entidad débil	
Interrelación	
Interrelación débil	
Atributo	
Atributo Llave	
Atributo Multivaluado	
Atributo derivado	
Atributo compuesto	

Figura 2.9 Símbolos usados en el diagrama E-R



### 2.7.3. TIPOS DE RELACIONES

Existen 4 tipos de **relaciones** que pueden establecerse entre entidades, las cuales se establecen con: cuantas entidades de tipo B se puede relacionar una entidad de tipo A.

**Relación uno a uno.** Se presenta cuando existe una relación, como su nombre lo indica uno a uno, denominado también relación de matrimonio. Una entidad del tipo A solo se puede relacionar con una entidad del tipo B, y viceversa. Esto se puede observar en la figura 2.10

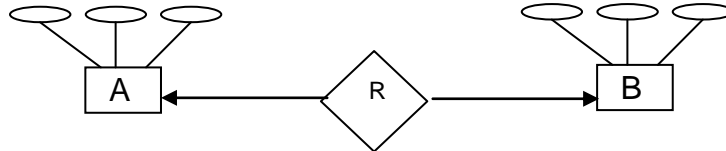


Figura 2.10 Relación uno a uno

A: Representa a una entidad de cualquier tipo diferente a una entidad B.

R: En el diagrama representa a la relación que existe entre las entidades.

El extremo de la flecha que se encuentra punteada indica el uno de la relación, en este caso, una entidad A ligada a una entidad B.

**Relación uno a muchos.** Significa que una entidad del tipo A puede relacionarse con cualquier cantidad de entidades del tipo B, y una entidad del tipo B solo puede estar relacionada con una entidad del tipo A. Esto se puede observar en la figura 2.11.

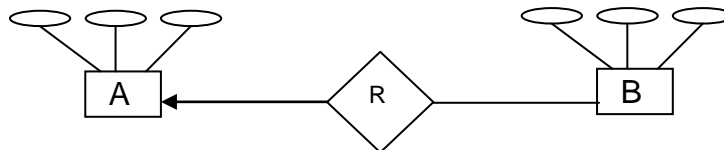


Figura 2.11 Relación uno a muchos

**Relación muchos a uno.** Indica que una entidad del tipo B puede relacionarse con cualquier cantidad de entidades del tipo A, mientras que cada entidad del tipo A solo puede relacionarse con solo una entidad del tipo B. Esto se puede observar en la figura 2.12.

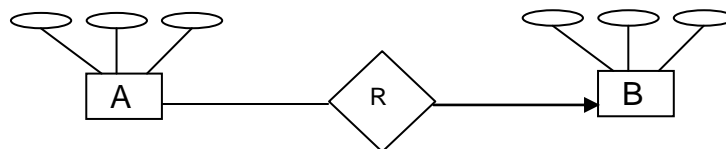


Figura 2.12 Relación muchos a uno

**Muchos a muchos.** Establece que cualquier cantidad de entidades del tipo A puede estar relacionados con cualquier cantidad de entidades del tipo B. Esto se puede observar en la figura 2.13

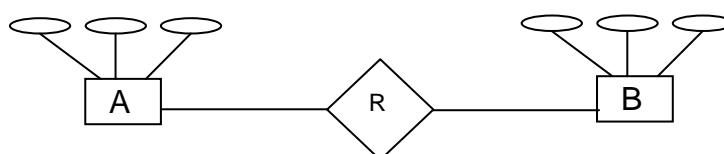


Figura 2.13 Relación muchos a muchos



---

A los tipos de relaciones antes descritos, también se le conoce como cardinalidad. La cardinalidad nos especifica los tipos de relaciones que existen entre las entidades en el modelo E-R y establecer con esto las validaciones necesarias para conseguir que los datos de la instancia (valor único en un momento dado de una base de datos) correspondan con la realidad.

Otra clase de limitantes lo constituye la dependencia de existencia. Refiriéndonos a las mismas entidades A y B, decimos que si la entidad A depende de la existencia de la entidad B, entonces A es dependiente de la existencia por B, si eliminamos a B tendríamos que eliminar por consecuente la entidad A, en este caso B es la entidad Dominante y A es la entidad subordinada.

#### 2.7.4 LLAVES PRIMARIAS

Como ya se ha mencionado anteriormente, la distinción de una entidad entre otra se debe a sus atributos, lo cual lo hacen único. Una llave primaria es aquel atributo el cual consideramos clave para la identificación de los demás atributos que describen a la entidad. Por ejemplo, si consideramos la entidad alumno, podríamos tener los siguientes atributos: nombre, semestre, especialidad, dirección, número de control, de todos estos atributos el que podremos designar como llave primaria es el número de control, ya que es diferente para cada alumno y este los identifica en la institución.

Claro que puede haber más de un atributo que pueda identificarse como llave primaria en este caso se selecciona la que consideremos más importante, los demás atributos son denominados **llaves secundarias**. Una clave o llave primaria es indicada gráficamente en el modelo E-R con una línea debajo del nombre del atributo.

## 2.8 NORMALIZACIÓN

La normalización es un proceso que consiste en comprobar que las tablas (también denominadas relaciones en terminología propia del modelo relacional de datos) definidas cumplen determinadas condiciones. Se pretende garantizar la no-existencia de redundancia y una cierta coherencia en la representación mediante un esquema relacional de las entidades y relaciones del modelo conceptual (diagrama E-R)

Mediante la normalización se puede solucionar diversos errores en el diseño de la base de datos. También facilita el trabajo posterior del administrador de la base de datos y de los desarrolladores de aplicaciones.

Su finalidad es reducir las inconsistencias y redundancias de los datos, facilitar el mantenimiento y evitar las anomalías en la manipulación de datos. El objetivo será obtener un modelo normalizado que represente las entidades normalizadas y las interrelaciones existentes entre ellas. Para ello se toman como punto teórico de partida el concepto de dependencia funcional que dice: “un atributo B depende funcionalmente de otro atributo A, de la misma entidad si a cada valor de A le corresponde sólo un valor de B”. Lo anterior se completa mediante la dependencia fundamental completa y la dependencia transitiva.



---

El procedimiento de normalización consiste en someter a las tablas que representan entidades a un análisis formal para ver si cumplen, o no las restricciones necesarias que aseguren evitar los problemas citados con anterioridad. A mayor nivel de normalización, mayor calidad en la organización de los datos y menor peligro para la integridad de los datos. Este procedimiento consiste en ir alcanzando formas normales.

Para lograr la consistencia e integridad de los datos, es necesario mejorar las relaciones. Estas mejoras deben dar como resultado tablas equivalentes, y poseer siempre tres propiedades: conservación de la información (de atributos y de tuplas), conservación de dependencias y mínima redundancia de los datos.

Las mejoras introducidas obligan a plantear hasta que Forma Normal es necesario llegar. Es decir, que nivel de depuración. Normalmente, es recomendable alcanzar la máxima Forma Normal, aunque luego es muy probable que restricciones existentes, de algún tipo, obliguen a retroceder a un nivel inferior de normalización, o incluso a cierto nivel de “desnormalización”. [6]

### **2.8.1 FORMAS DE NORMALIZACIÓN**

Primera forma normal (FN1). Se pretende garantizar la no-existencia de grupos repetitivos. Un grupo repetitivo es un conjunto de atributos de igual semántica en el problema y dominio, que se toman valores distintos para la misma clave. Cualquier esquema que tenga claves correctas está seguro en FN1.

Segunda forma normal (FN2). Si esta en FN1 y cada atributo de la tabla que no forma parte de la clave dependen funcionalmente de forma completa de la clave primaria. Es decir, depende de toda la clave y no de ningún subconjunto de ella. Se pretende garantizar una correcta elección de claves y eliminar redundancias. Si la clave está formada por un único atributo entonces ese esquema estará seguro en segunda forma normal.

Tercera forma normal (FN3). Si esta en FN2 y cada atributo no primo de la tabla no dependen funcionalmente de forma transitiva de la clave primaria.

BCFN (Forma Normal de Boyce-Codd). Una relación está en la forma normal de Boyce-Codd si, y sólo si, todo determinante es una clave candidata.

La 2FN y la 3FN eliminan las dependencias parciales y las dependencias transitivas de la clave primaria. Pero ese tipo de dependencias todavía puede existir sobre otras claves candidatas, si éstas existen. La BCFN es más fuerte que la 3FN, por lo tanto, toda relación en BCFN está en 3FN. [7]

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.



---

## 2.9 DESARROLLO WEB

Desarrollo web es el conjunto de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador en Internet a fin de realizar determinadas tareas o mostrar información [8].

Tradicionalmente un software departamental o incluso un ambicioso proyecto corporativo de gran envergadura es desarrollado en forma standalone, es decir, usando lenguajes ya sea compilados (C, C++, Delphi), semicompilados (.NET, Mono, Java), o interpretados (Python) para crear tanto la funcionalidad como toda la interfaz de los usuarios, pero cabe perfectamente un desarrollo orientado a web para dichos propósitos, siendo más homogéneo y multiplataforma, y dependiendo de las tecnologías utilizadas, más rápido y robusto tanto para diseñar, implementar y probar, como para su uso una vez terminado.

Funcionalmente, el desarrollador web, que es quien realiza esta labor, normalmente sólo se preocupa por el funcionamiento del software, es tarea del diseñador web preocuparse del aspecto final (layout) de la página y del webmaster el integrar ambas partes. En ocasiones el webmaster también se encarga de actualizar los contenidos de la página.

Los lenguajes de programación más usados en desarrollo web son principalmente: ASP.NET, PHP y JSP, aunque aún hay quienes usan ASP, Macromedia ColdFusion y Perl. Algunos desarrolladores web hablan muy bien de Ruby on Rails pero no está muy difundido todavía. También hay un proyecto para usar Pascal en web a través del lenguaje Pascal Server Pages - PSP pero aún no se han concretado desarrollos, hasta donde se conoce. [9]

La base de datos más popular en desarrollo web es MySQL, seguida por Oracle, SQL Server y PostgreSQL, también puede usarse perfectamente Firebird o HSQL.

## 2.10 APACHE

El nombre de Apache viene de **A PatCH** y **sErver**. Apache actúa como servidor Web. Su labor principal consiste en analizar cualquier archivo solicitado por un navegador y mostrar los resultados correctos en función del código del archivo. Apache es muy potente y puede realizar prácticamente cualquier tarea que se le pida [8].

Entre las funciones de las opciones de servidor que incluye destacamos las siguientes:

- Páginas protegidas con contraseñas para múltiples usuarios.
- Páginas de errores personalizados.



- 
- Representación de código en numerosos niveles de HTML y la posibilidad de determinar a qué nivel el navegador puede aceptar el contenido.
  - Registros de uso y de errores en diferentes formatos personalizables.
  - Alojamiento virtual de diferentes direcciones IP asignadas al mismo servidor.
  - Directivas Directory Index a varios archivos.
  - Reescritura de URL sin límite fijo.

Su flexibilidad, prestaciones y, evidentemente, su precio, lo han convertido en la principal opción. Se puede utilizar para alojar un sitio Web para el público general, una intranet de una empresa o simplemente para probar páginas personales antes de cargarlas a un servidor seguro en otro equipo.

## 2.11 PHP

PHP es un lenguaje programación (originario del nombre **PHP Tools**, o **Personal Home Page Tools**) de secuencias de comandos del lado del servidor que permite que un sitio Web sea realmente dinámico. PHP (preprocesador de hipertexto). Su flexibilidad y su curva de aprendizaje relativamente corta lo convierten en uno de los lenguajes de programación más conocidos. Su popularidad sigue en aumento debido a la acogida por parte de empresas y particulares como alternativa al lenguaje ASP de Microsoft y porque sus ventajas superan a los costes. [10]

El **lenguaje PHP** tiene la característica de poder mezclarse con el lenguaje HTML. **PHP**, al contrario que este último, se interpreta y ejecuta directamente en el servidor en el que está albergada la página Web, con lo que el visitante a la misma únicamente recibe el resultado buscado por el código en el que está escrito [10].

## 2.12 MYSQL

MYSQL surgió como un intento de conectar el gestor MYSQL a las tablas propias de MYSQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que MYSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a MYSQL. [11]

MYSQL empezó como un proyecto de programación hace unos 10 años, cuando un programador sueco decidió crear su propio gestor de datos para la aplicación que estaba desarrollando. Hasta ese momento usaba msql, pero vio que se le quedaba corto en algunos aspectos que él podría mejorar directamente [12]. Pronto MYSQL fue el número 1 en rendimiento en el ámbito de las bases de datos de código abierto. Posteriormente se creó una empresa sueca, llamada Tux que se dedicó a crear nuevas



---

versiones de MYSQL y comercializarlo en ciertos casos. Dentro de sus características se encuentran:

- Disponibilidad en gran cantidad de plataformas y sistemas.
- 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.
- Búsqueda e indexación de campos de texto.

Existen dos tipos de comandos en MYSQL:

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

### 2.12.1 COMANDOS DDL

Con estos comandos se especifican las funciones de definición de datos de ahí su nombre **DDL (Data Definition Language)** e incluye los comandos:

- **CREATE**  
Este comando es utilizado para crear nuevas Bases de Datos o nuevas tablas dentro de una Base de Datos específica.

Sintaxis:

```
DROP DATABASE IF EXISTS nombre_BD;  
CREATE DATABASE nombre_BD;
```

En el caso de la creación de tablas la sintaxis es la siguiente:

```
USE nombre_BD;  
DROP TABLE IF EXISTS nombre_tabla;  
CREATE TABLE nombre_tabla(  
var1 tipo1 [NOT NULL] [PRIMARY KEY],  
... varN tipoN  
);
```

En la creación de tablas es muy importante la generación de llaves foráneas. Los comandos para establecer llaves foráneas incluyen:

- Foreign Key es utilizado para designar llaves foráneas en una tabla.
- References establece cuál es la tabla de donde se obtiene la llave foránea.
- On delete cascade indica que al eliminar registros en la otra tabla se eliminan de la tabla actual.



- On update cascade indica que al actualizar registros en la otra tabla se actualizan en la tabla actual.

Sintaxis:

```
USE nombre_BD;  
DROP TABLE IF EXISTS nombre_tabla;  
CREATE TABLE nombre_tabla(  
var1 tipo1 [NOT NULL] [PRIMARY KEY],  
var_foranea tipo x [NOT NULL] ...  
varN tipoN  
FOREIGN KEY(var_foranea)  
REFERENCES otra_Tabla(var_foranea) ON DELETE CASCADE ON  
UPDATE CASCADE  
);
```

- **DROP**  
Este comando es empleado para eliminar Bases de Datos y tablas.

Sintaxis

```
DROP Database nombre_BD;  
DROP table nombre_tabla;
```

- **ALTER**  
Este comando es utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Sintaxis:

```
ALTER TABLE nombre_tabla  
ADD COLUMN campo1 tipo_campo1;
```

```
ALTER TABLE nombre_tabla  
DROP COLUMN campo1;
```

### 2.12.2 COMANDOS DML

Este tipo de comandos se encargan de la manipulación de los datos es por eso su nombre **DML(Data Manipulation Language)**, incluye los comandos:

- **INSERT**

Este comando es utilizado para cargar lotes de datos en la base de datos en una única operación.



---

Sintaxis:

Use nombre\_BD;

INSERT INTO Tabla (campo1, campo2, ..., campoN)

VALUES (valor1, valor2, ..., valorN);

- **SELECT**

Este comando es utilizado para consultar registros de la base de datos que satisfagan un criterio determinado

Sintaxis:

Use nombre\_BD;

SELECT [DISTINCT] <lista\_columnas>

FROM <lista\_tablas>

[WHERE <predicado>]

- **UPDATE**

Utilizado para modificar los valores de los campos y registros especificados

Sintaxis:

Use nombre\_BD;

UPDATE Tabla SET

Campo1=Valor1, Campo2=Valor2, ... CampoN=ValorN

WHERE Criterio;



---

---

# ***CAPÍTULO III***

# ***DESARROLLO***

# ***DEL***

# ***SISTEMA***



---

## **3.1 ANALISIS DE REQUISITOS**

El presente capítulo se ha redactado con el propósito de determinar los aspectos generales y especificaciones que seguirá la aplicación de desarrollo, del sistema web de gestión para el mantenimiento correctivo y preventivo de equipos de cómputo. El presente documento va dirigido a los usuarios, administradores y gerentes de la empresa en la cual está pensado este sistema de mantenimiento.

Así como determinar cuáles son las necesidades y la técnica o tecnología más factible a manera de satisfacer a los usuarios, tomando en cuenta todos sus requisitos e implementarlos en el sistema de mantenimiento preventivo y correctivo de equipos de cómputo.

### **3.1.1 JUSTIFICACION**

Los alcances que el sistema pretende son el de ser un sistema (software) de calidad y cumplir todas las expectativas y requerimientos estipulados por la empresa y los usuarios que interactúen con el sistema, así como un sistema o producto fácil de usar, seguro, fiable, etc.

Realizando la recopilación de la información y el análisis de requerimientos para comprender las necesidades de los usuarios, y así llegar a elegir la forma en que se dará solución al problema, y la creación del presente capítulo de análisis de requerimientos.

Posteriormente se utilizara el modelo entidad-relación, para definir la estructura, integridad, manipulación de los datos.

A continuación, se realizara la codificación en donde se genera el programa que compone el sistema, las interfaces, así mismo la Base de Datos.

Finalmente, se realizara las pruebas, para verificar que todos los componentes del sistema funcionan correctamente y asegurar que la entrada produce los resultados que realmente se requerían.

### **3.1.2 AMBITO DEL SISTEMA**

El sistema web de mantenimiento preventivo y correctivo de equipos de cómputo dará la oportunidad, al administrador del sistema de llevar un control acerca de los mantenimientos de los equipos de cómputo que se encuentren en la empresa, así como a los usuarios que estén registrados, de enviar peticiones al administrador sobre el funcionamiento de su equipo. Este sistema no realizara el mantenimiento físico solo se llevara un control acerca de estos mantenimientos y gestión de las peticiones de los usuarios.



---

Se espera que este sistema permita ahorrar el tiempo en que una petición de parte de los usuarios se resuelva lo más pronto posible y llevar a cabo un mejor control cuando se realicen los mantenimientos de los equipos de cómputo de la empresa. Que el Gerente de la empresa así como los gerentes de cada área estén enterados sobre el funcionamiento de cada equipo y llevar a cabo un control de los equipos que se encuentren en sus áreas y promover entre los usuarios la cultura de cuidar los equipos de cómputo.

### **3.1.3 CARACTERISTICAS DE LOS USUARIOS**

Usuarios, Administrador y Gerentes necesitan estar registrados en el sistema para que se les pueda proporcionar su nombre de usuario y su contraseña y puedan ingresar al sistema y utilizar las opciones que deseen así como un equipo de cómputo donde poder visualizar el sistema.

A través del Sistema de mantenimiento antes mencionado, los usuarios tendrán acceso a la información que ellos necesitan, y podrán consultarlo en cualquier momento desde una computadora conectada a la Web, para dar solución a sus necesidades.

De esta manera se pondrá en marcha el Sistema de Mantenimiento preventivo y correctivo de equipos de cómputo modo funcional y flexible.

### **3.1.4 RECURSOS DEL SISTEMA**

- La información fue almacenada en un servidor de Base de Datos que tendrá los servicios del Servidor Web o APACHE y se puede tener acceso mediante una computadora.
- El Sistema se desarrolló en un lenguaje enfocado en aplicaciones a la Web, como PHP, HTML, y MySQL.
- El Sistema cuenta con un entorno, sencillo y amigable en línea para consultar la Información acerca de los mantenimientos.
- Los requerimientos mínimos para el Servidor son:
  - Software
    - Tener instalado un Sistema Operativo
    - Tener instalado un Servidor Web o Apache
    - Tener instalado el programa PHP
    - Tener instalado el programa MySQL
  - Hardware
    - Una computadora Intel Pentium III o superior, puede ser alguna Compatible.
    - 128 Mb en RAM como mínimo



- 
- 1Gb de memoria libre en disco duro
  - Los requerimientos mínimos para el Cliente son:
    - Software
      - Tener instalado un Sistema Operativo
      - Tener instalado un navegador de Internet.
    - Hardware
      - Una computadora Intel Pentium III o superior, también puede ser compatible.
      - Una conexión para internet
      - 128 Mb en RAM como mínimo.

### **3.1.5 REQUERIMIENTOS DE LA INTERFAZ**

#### **Interfaz de usuario**

Como la aplicación fue desarrollada y dirigida a la ejecución en un entorno visual, la interacción entre el usuario y la aplicación se realizó mediante pantallas típicas en un entorno de tipo formulario con ventanas, botones, etiquetas, etc.

#### **Interfaces del Software**

El servidor debe contar con una base de datos, con un Servidor de la Web o APACHE y herramientas de programación para Internet como los lenguajes PHP, HTML y MySQL.

El Sistema debe utilizar un Administrador de Base de Datos

#### **Interfaces del Hardware**

No es necesario que la computadora que tome la función de servidor tenga un mínimo de espacio en el disco duro debido a que el Sistema funcionará mediante vía Web o APACHE, pero si deberá contar con una conexión a Internet.

#### **Interfaces de Comunicación**

La computadora donde se implantó el Sistema deberá contar con el protocolo necesario para una conexión al Servidor de la Web o APACHE como TCP/IP.



---

### 3.1.6 RESTRICCIONES

- El que no haya acceso al Servidor.
- No es posible garantizar el acceso adecuado por tráfico de la red.
- Que no se cuente en algún equipo de cómputo con entradas de disco CD puesto que en la empresa se cuenta con mini laptops y sería una limitación al instalar el sistema a los usuarios.
- Que no todos los usuarios cuenten con un equipo de cómputo donde ingresar al sistema.

### 3.1.7 FUNCIONES DEL SISTEMA

El sistema web de mantenimiento preventivo y correctivo de equipos de cómputo dará la oportunidad, al administrador del sistema de llevar un control acerca de los mantenimientos de los equipos de cómputo que se encuentren en la empresa, así como a los usuarios que estén registrados, de enviar peticiones al administrador sobre el funcionamiento de su equipo.

Este sistema no realizara el mantenimiento físico solo se llevara un control acerca de estos mantenimientos y gestión de las peticiones de los usuarios. El sistema permitirá a los usuarios ingresar con su nombre de usuario y contraseña y realizar las peticiones, que será una descripción breve en forma de texto de lo que requiere su equipo ya sea del tipo (red, software hardware u otra) y con una prioridad que podrá ser (baja, normal y urgente). Así como de poder visualizar los avisos de los próximos mantenimientos o cualquier otra noticia que el administrador publique.

El sistema permitirá al administrador ingresar con su nombre de usuario y contraseña a revisar las peticiones realizadas por los usuarios conforme el grado de prioridad, una vez que el administrador termine de revisar el equipo tendrá que cerrar estas peticiones ya resueltas y enviar un aviso al usuario de que su equipo está listo.

Además el administrador mediante el sistema tendrá las opciones de consultar, insertar, eliminar, actualizar o realizar búsqueda de información referente a los usuarios y equipos que se encuentren dados de alta en la base de datos.

El administrador también tendrá otras funciones que el sistema le permitirá realizar como elaborar las órdenes de compra de mantenimiento, enviárselos al gerente general y una vez que esté apruebe las órdenes, se podrá comprar el material que requieran para el mantenimiento que se realiza por periodos o en casos muy urgente y algún usuario requiera antes cualquier refacción.



---

También el administrador será el encargado de enviar avisos a los usuarios en cuanto a las fechas en que se realizara el mantenimiento general del equipo o cualquier otro aviso que desee publicar, una vez que el administrador termine de realizar los mantenimientos generales. El sistema le permitirá elaborar los reportes de esos mantenimientos y así tener un resultado sobre el estado de los equipos de cómputo.

Otra de las funciones que podrá realizar el administrador es imprimir los reportes realizados después del mantenimiento, dárselos a firmar a los usuarios y guardarlos en para futuras aclaraciones.

El sistema también permitirá al gerente ingresar con su nombre de usuario y contraseña a visualizar los reportes de los mantenimientos que se realizaron y así tener un mejor control de estos equipos.

### 3.1.8 CASOS DE USO

En la figura 3.1 se muestra el Diagrama de Casos de Uso, donde podemos visualizar cuatro Actores Usuarios, Administradores, Gerentes e Impresora en donde cada uno tendrá las siguientes funciones:

- **Usuarios (empleados):** Podrá realizar peticiones para el mantenimiento de sus equipos de Cómputo.
- **Administradores:** Podrá revisar y cerrar peticiones, elaborar órdenes de compra, reportes y poder enviarlos al gerente así como poder imprimirlos en caso necesario.
- **Gerentes:** Podrá visualizar los reportes de los mantenimientos y las órdenes de compra.
- **Impresora:** Tendrá la función de imprimir órdenes de compra y reportes cuando sea requerido.

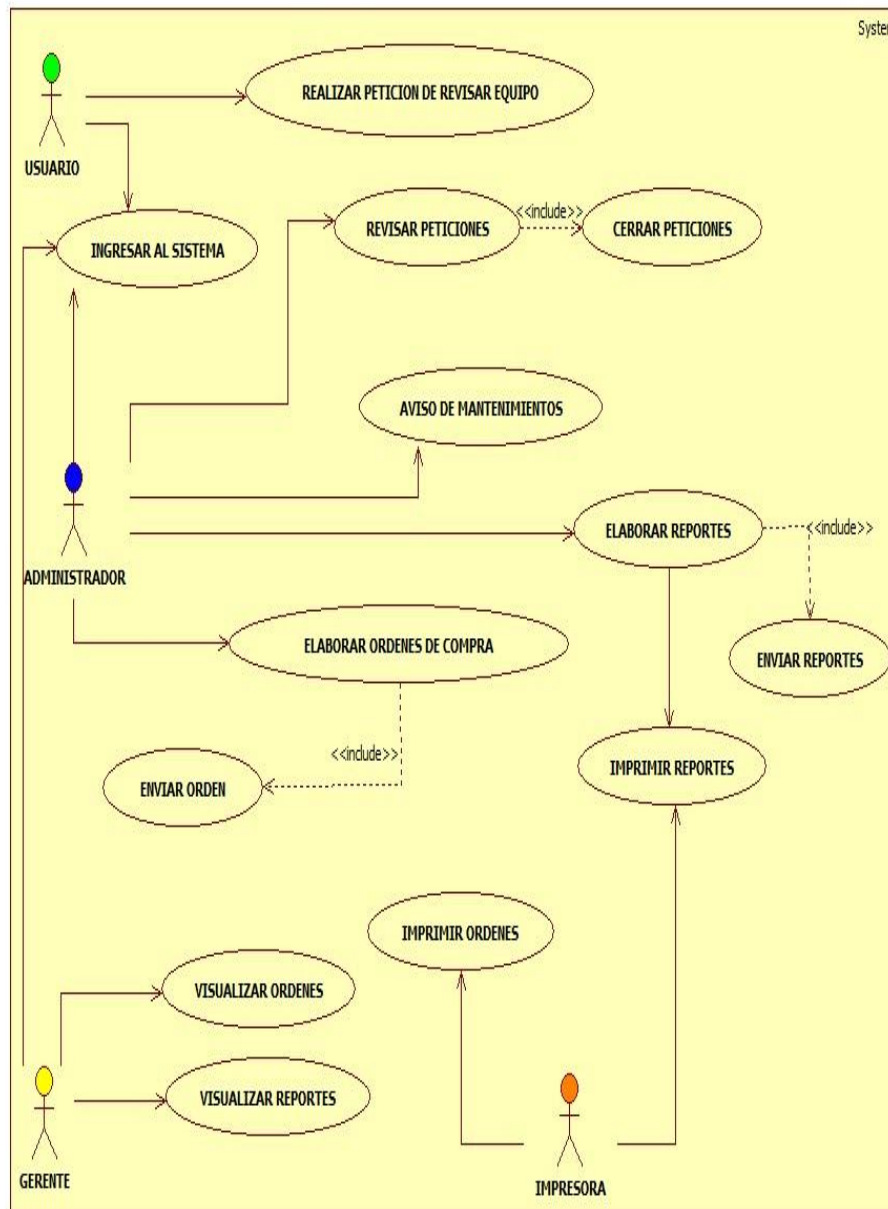


Figura 3.1 Diagrama de Casos de Uso

## 3.2 DISEÑO DEL SISTEMA

El diseño del sistema y de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. El diseño se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas.



### 3.2.1 DIAGRAMA ENTIDAD RELACION

El modelo entidad-relación que se usara es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas. En la figura 3.2 se muestra que después de identificar las entidades, atributos y relaciones se procedió al modelo.

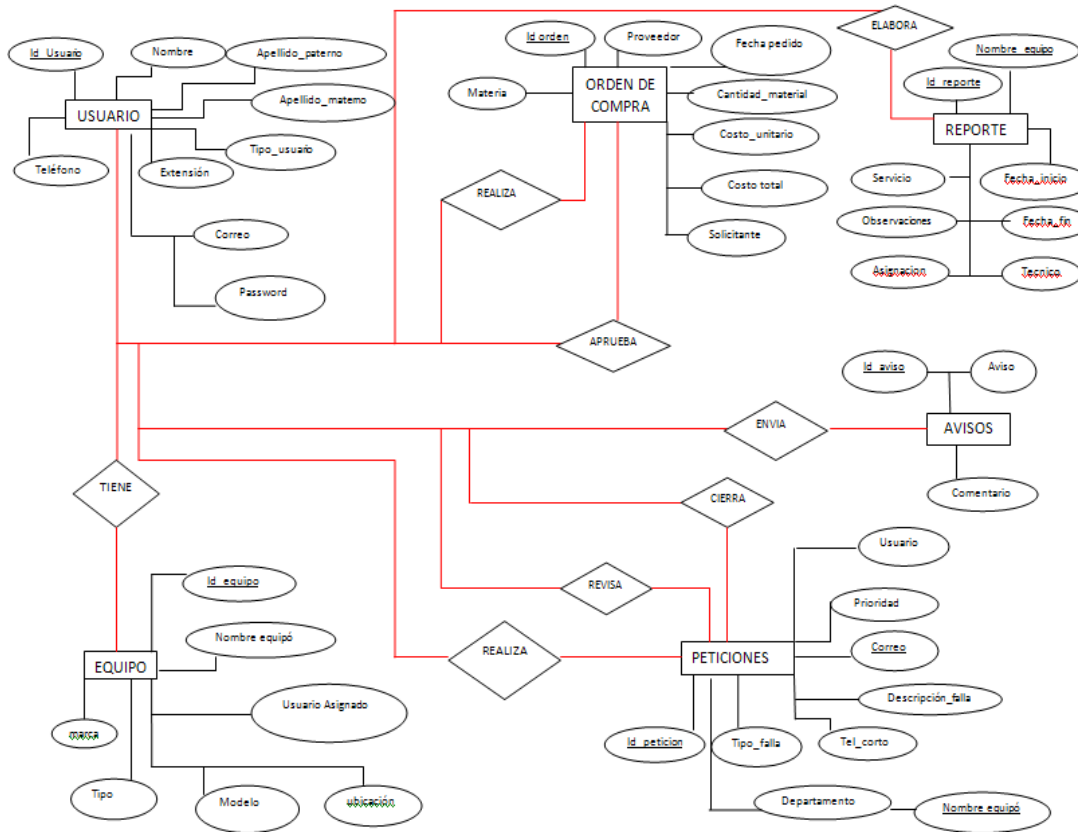


Figura 3.2 Diagrama Entidad-Relación



### 3.2.2 DIAGRAMA DE CLASES

En la figura 3.3 se muestra el diagrama de clases donde se muestran las entidades de usuario, equipo, reportes, peticiones, avisos y órdenes de compra, donde cada una muestra sus atributos y sus funciones que describen a continuación:

- **Usuarios:** pueden ser empleados (usuarios), administradores y gerentes en caso de ser un empleado podrá realizar peticiones y visualizar los avisos, en caso de ser el administrador es el que podrá visualizar las peticiones realizarlas y cerrarlas, así como elaborar reportes y órdenes de compra también podrá enviar avisos y por último en caso de ser Gerente podrá visualizar los reportes generados por el administrador y así tener conocimiento sobre el estado de los equipos, a su vez también podrá visualizar las órdenes de compra.
- **Equipo:** cada usuario tiene que tener un equipo de cómputo.
- **Reportes:** Se elaboran cuando se terminen los mantenimientos y se tendrán que enviar al gerente para saber el estado de cada equipo de cómputo y tener un mejor control.
- **Peticiones:** Son las solicitadas por los usuarios que son empleados y que requieren algún servicio para su equipo de cómputo.
- **Avisos:** Estos avisos son generados por el administrador y enviados a los usuarios (empleados) para que estén enterados de los próximos mantenimientos.
- **Órdenes de compra:** también son generados por el administrador y se generan cuando se requiera algún tipo de material para los mantenimientos, una vez generados se enviarán al gerente para su aprobación.

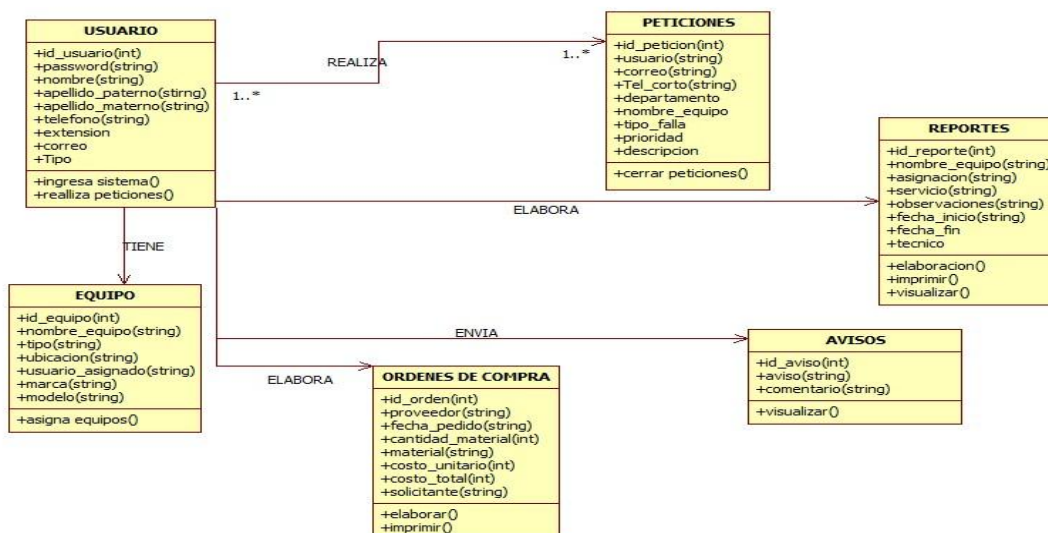


Figura 3.3 Diagrama de Clases



### 3.2.3 DIAGRAMA DE SECUENCIA

PARA EL CASO DE USO INGRESAR AL SISTEMA ACTOR USUARIO, ADMINISTRADOR Y GERENTE:

En la figura 3.4 se muestra un diagrama de secuencia que representa el ingreso al sistema para los usuarios, administradores y gerentes como se puede apreciar el usuario que desee ingresar al sistema tendrá que proporcionar su nombre de usuario y contraseña, el sistema realizara la validación de la información proporcionada y si es correcto se abrirá la ventana de menú en caso contrario regresara a la ventana de login para volver a proporcionar la información correcta.

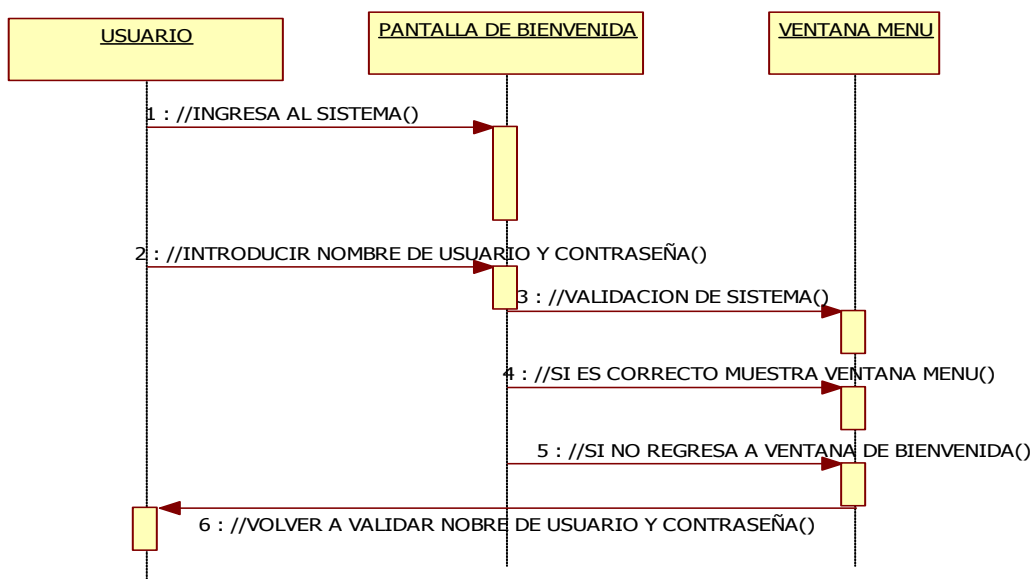


Figura 3.4 Diagrama de Secuencia Para Ingresar Al sistema

PARA EL CASO DE USO REALIZAR PETICION REVISAR EQUIPO ACTOR USUARIO (EMPLEADO).

En la figura 3.5 se muestra el diagrama de secuencia que representa como realizar las peticiones de revisar equipo, como podemos observar el usuario debe ingresar al sistema proporcionar su nombre de usuario y contraseña cuando el sistema valide la información proporcionada desplegara la ventana de menú para usuarios, estos tendrán acceso a la ventana de realizar peticiones, donde podrá llenar un formulario que será enviado al administrador, el usuario podrá salir del sistema, regresar al menú de usuarios o al menú principal.

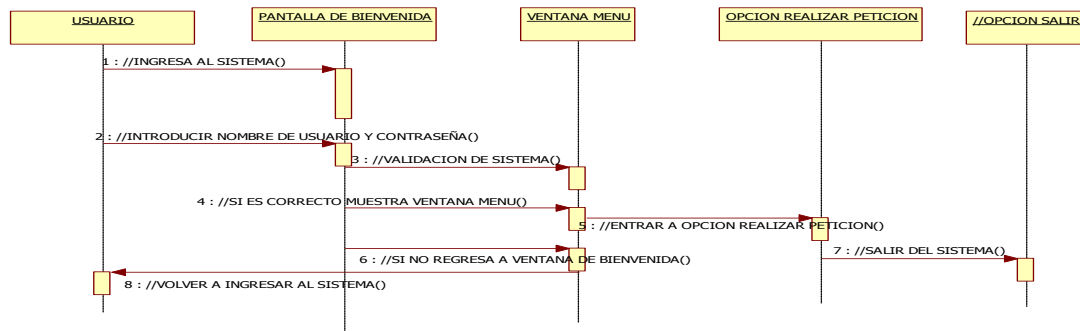


Figura 3.5 Diagrama de Secuencia Para Realizar Petición

### PARA EL CASO DE USO REVISAR PETICIONES ACTOR ADMINISTRADOR:

En la figura 3.6 se muestra el diagrama de secuencia que representa como revisar las peticiones en este caso es para los administradores que son los encargados de realizar esta función, como podemos observar el administrador debe ingresar al sistema con su nombre de usuario y contraseña cuando el sistema realice la validación de que la información proporcionada sea correcta, desplegara una ventana donde el administrador podrá revisar todas las peticiones realizadas por los usuarios(empleados) una vez revisadas las peticiones el administrador podrá salir del sistema o regresar al menú principal.

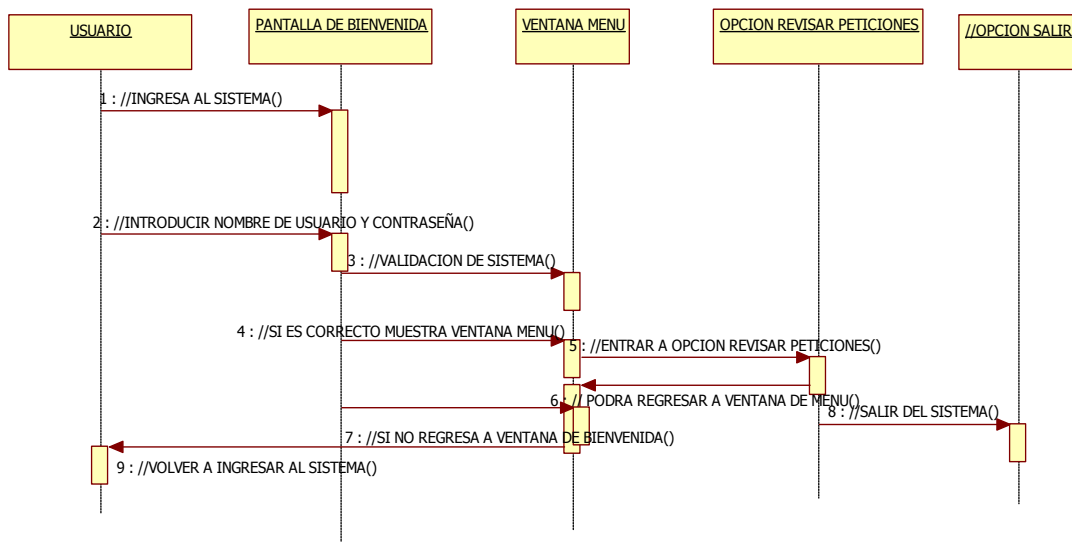


Figura 3.6 Diagrama de Secuencia Para Revisar Petición



## PARA EL CASO DE USO CERRAR PETICIONES ACTOR ADMINISTRADOR:

En la figura 3.7 se muestra el diagrama de secuencia que representa como cerrar las peticiones de revisar equipo, como podemos observar el administrador debe ingresar al sistema proporcionar su nombre de usuario y contraseña, cuando el sistema valide la información proporcionada desplegara la ventana de menú para administradores y tendrá acceso a la ventana de realizar peticiones, cuando termine de revisar las peticiones y las haya realizado el administrador podrá cerrar las peticiones, el usuario(administrador) podrá salir del sistema, regresar al menú de administradores o al menú principal.

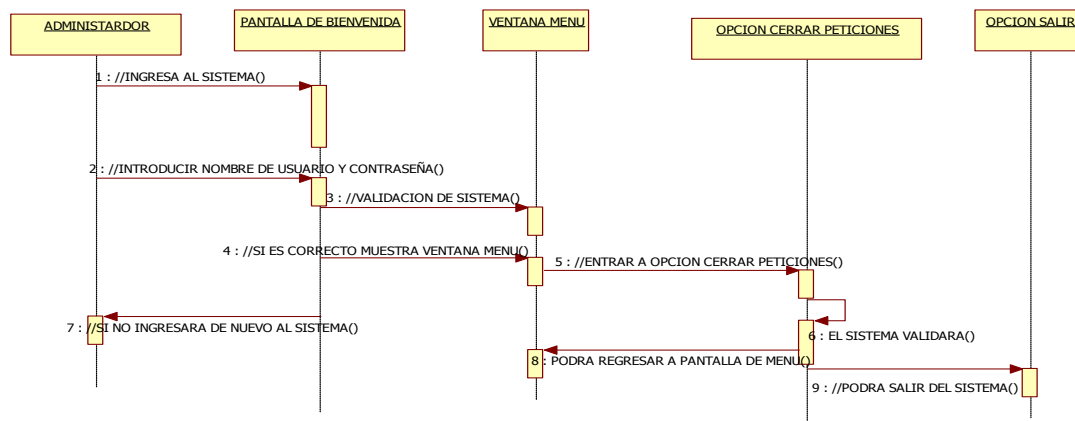


Figura 3.7 Diagrama de Secuencia Para Cerrar Petición

## PARA EL CASO DE USO ELABORAR ORDEN ACTOR ADMINISTRADOR:

En la figura 3.8 se muestra el diagrama de secuencia que representa como elaborar una orden de compra, podemos observar que el administrador debe ingresar al sistema proporcionar su nombre de usuario y contraseña, cuando el sistema valide la información proporcionada el sistema mostrara la ventana de menú para administradores tendrá acceso a la ventana de elaborar orden, cuando la orden de compra este lista podrá enviarlas al gerente y el administrador podrá salir del sistema, regresar al menú de administradores o al menú principal.

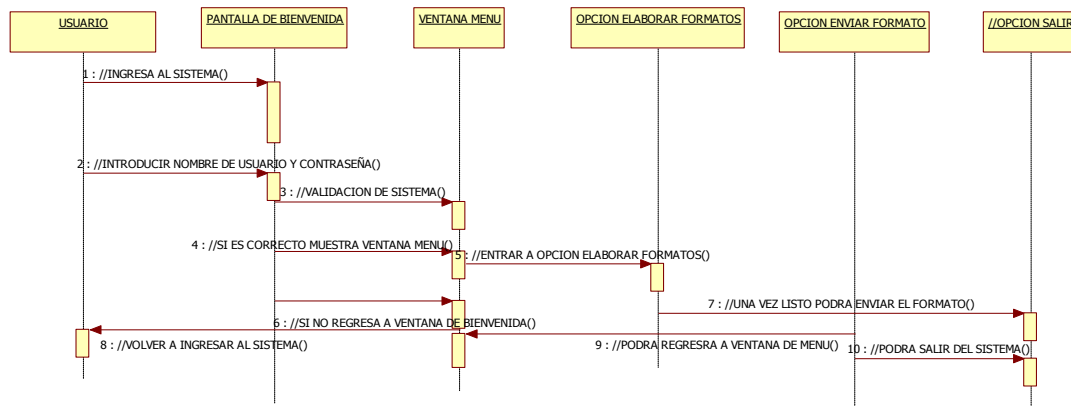


Figura 3.8 Diagrama de Secuencia Para Elaborar Orden



## PARA EL CASO DE USO ELABORAR AVISOS DE MANTENIMIENTO ACTOR ADMINISTRADOR:

En la figura 3.9 se muestra el diagrama de secuencia que representa como elaborar los avisos de mantenimiento, podemos observar que el administrador debe ingresar al sistema proporcionar su nombre de usuario y contraseña, cuando el sistema valide la información proporcionada desplegara la ventana de menú para administradores tendrá acceso a la ventana de Avisos, donde podrá elaborar los avisos sobre los próximos mantenimientos cuando termine de elaborarlos podrá enviarlos a los usuarios(empleados ) y así estén enterados de los próximos mantenimientos, el usuario(administrador) podrá salir del sistema, regresar al menú de administradores o al menú principal.

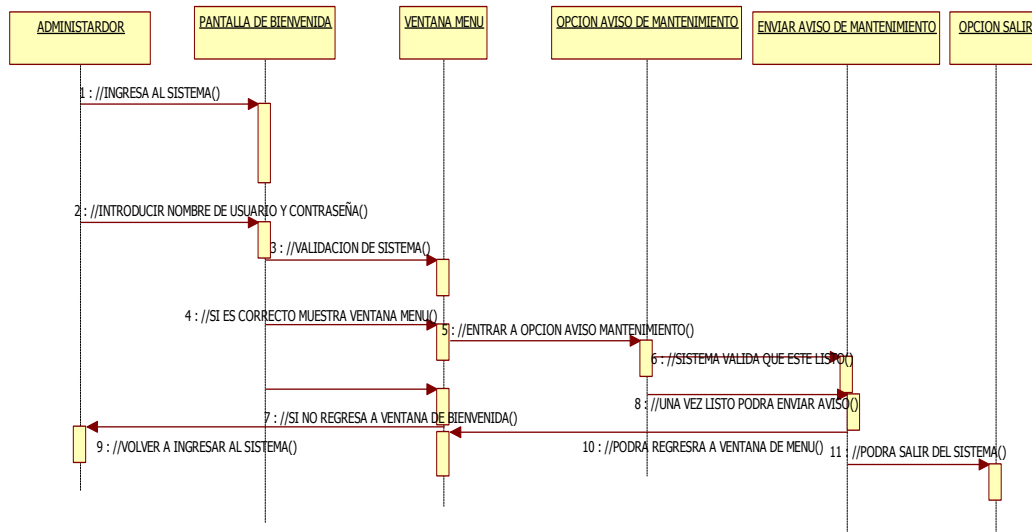


Figura 3.9 Diagrama de Secuencia Para Elaborar Aviso

## PARA EL CASO DE USO ELABORAR REPORTES ACTOR ADMINISTRADOR:

En la figura 3.10 se muestra el diagrama de secuencia que representa como elaborar una reporte, podemos observar que el administrador debe ingresar al sistema proporcionar su nombre de usuario y contraseña, cuando el sistema valide la información proporcionada el sistema mostrara la ventana de menú para administradores donde tendrá acceso a la ventana reportes, el administrador podrá elaborar los reportes de los mantenimientos, cuando los reportes estén listos podrá enviarlos al gerente y el administrador podrá salir del sistema, regresar al menú de administradores o al menú principal.

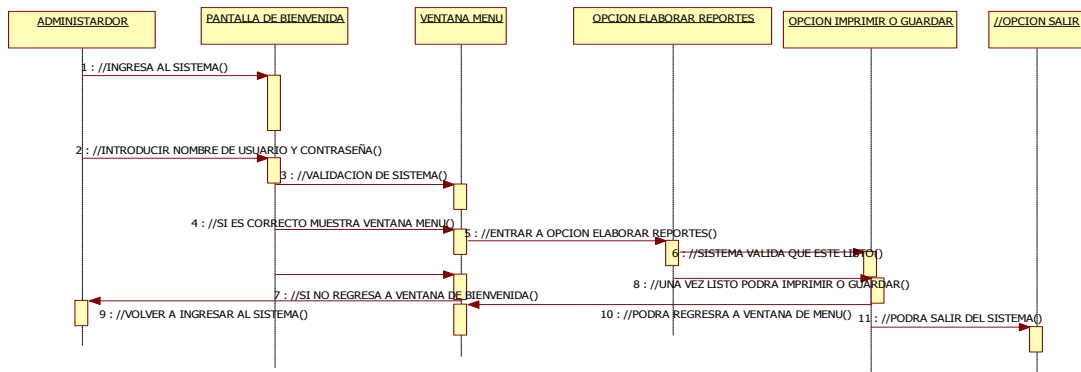


Figura 3.10 Diagrama de Secuencia Para Elaborar Reporte

### PARA EL CASO DE USO VISUALIZAR REPORTES Y ORDENES ACTOR GERENTE

En la figura 3.11 se muestra el diagrama de secuencia que representa como visualizar reportes y órdenes de compra, podemos observar que el administrador debe ingresar al sistema proporcionar su nombre de usuario y contraseña, cuando el sistema valide la información proporcionada el sistema mostrara la ventana de menú para administradores, tendrá acceso a la ventana de visualizar reporte y regresar a la ventana de menú cuando termine de visualizar los reportes, también podrá visualizar las ordenes el gerente podrá salir del sistema, regresar al menú de gerente o al menú principal.

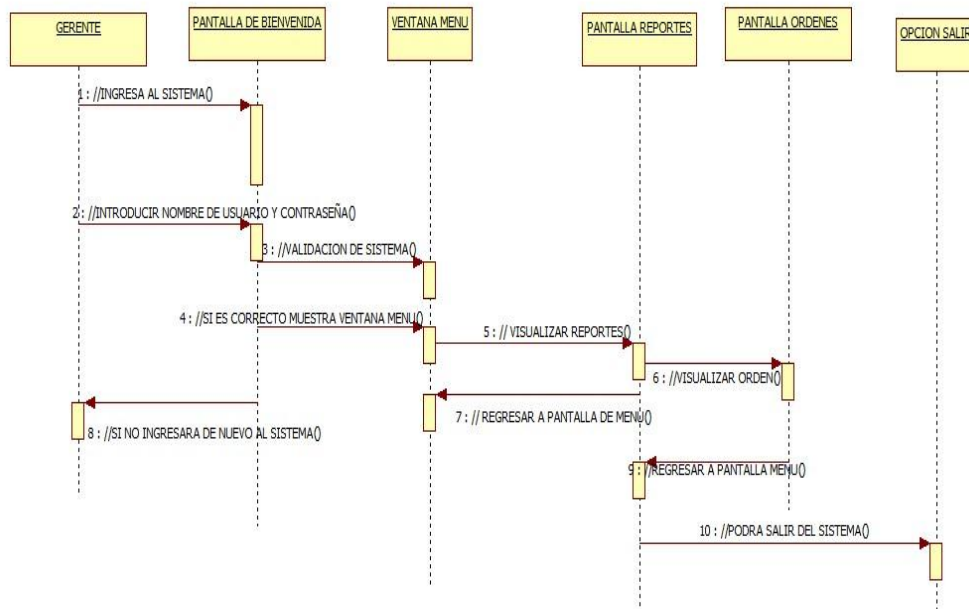


Figura 3.11 Diagrama de Secuencia Para Visualizar Reporte



### 3.2.4 DIAGRAMA DE ACTIVIDAD

En la figura 3.12 se muestra el diagrama de actividad donde podemos observar que para ingresar al sistema debemos de proporcionar un nombre de usuario y un login, dependiendo de cómo se desea ingresar si como Usuario (empleado), Administrador o gerente se tendrán distintas funciones y por lo tanto distintas pantallas.

- **Sesión como Usuario (empleado):** Donde el usuario podrá realizar peticiones y podrá revisar los avisos que el administrador envía, cuando el usuario no tenga otra actividad que realizar podrá salir del sistema.
- **Sesión como Administrador:** donde el administrador podrá realizar las actividades de revisar peticiones, cerrarlas, enviar avisos, elaborar órdenes de compra y reportes cuando no desee realizar más actividades podrá salir del sistema.
- **Sesión como Gerente:** Donde el gerente podrá realizar las actividades de visualizar los reportes y órdenes de compra, elaborados por el administrador cuando el gerente termine estas actividades podrá salir del sistema.

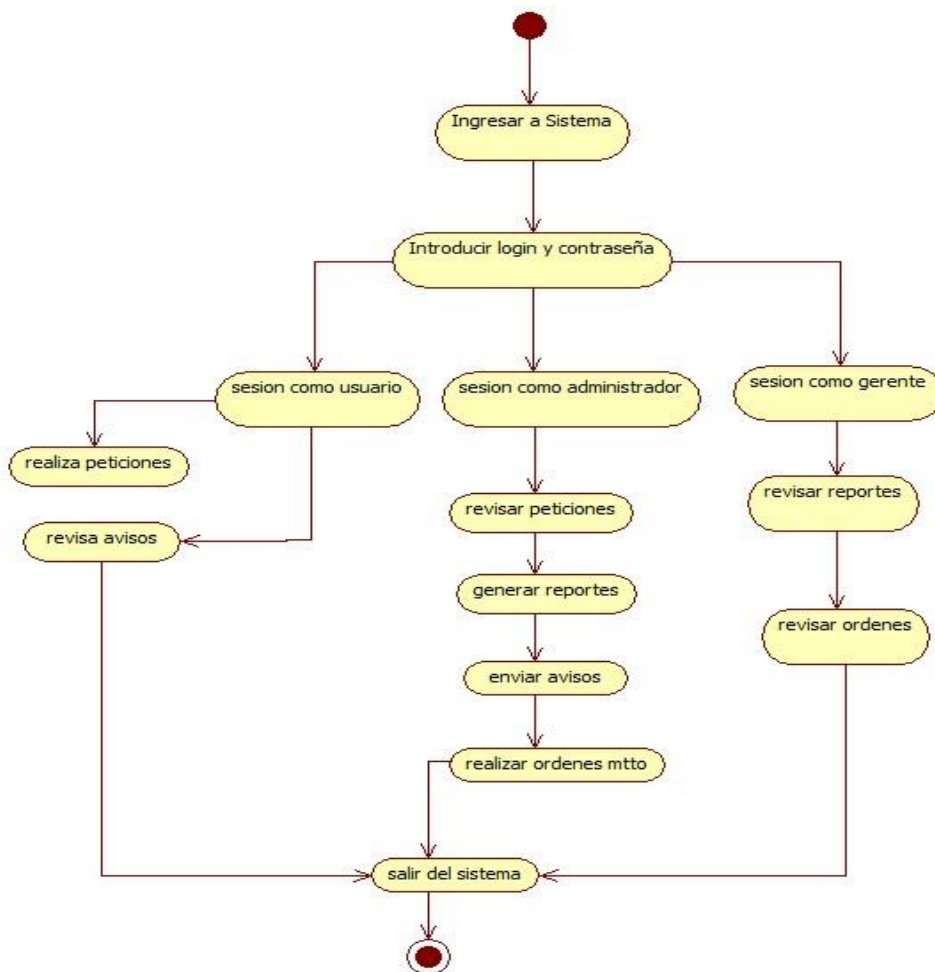


Figura 3.12 Diagrama de Actividad



### 3.2.5 DIAGRAMA DE ESTADO

En la figura 3.13 se muestra el diagrama de estado donde podemos observar que para ingresar al sistema debemos de proporcionar un nombre de usuario y un login, dependiendo de cómo se desea ingresar si como Usuario (empleado), Administrador o gerente, el sistema realizara la validación de la información proporcionada si es correcto ingresara con la sesión deseada, en caso contrario regresara a la pantalla de login para rectificar nuevamente los datos cuando y las sesiones como se puede acceder al sistema son las siguientes:

- **Sesión como Usuario (empleado):** Donde el usuario tendrá los siguientes estados realizar peticiones y revisar los avisos que el administrador envía, cuando el usuario termine podrá salir del sistema.
- **Sesión como Administrador:** donde el administrador tiene los estados de revisar peticiones, cerrarlas, enviar avisos, elaborar órdenes de compra y reportes cuando el administrador termine podrá salir del sistema.
- **Sesión como Gerente:** Donde el gerente tiene los estados de revisar los reportes y órdenes de compra, elaborados por el administrador cuando el gerente termine podrá salir del sistema.

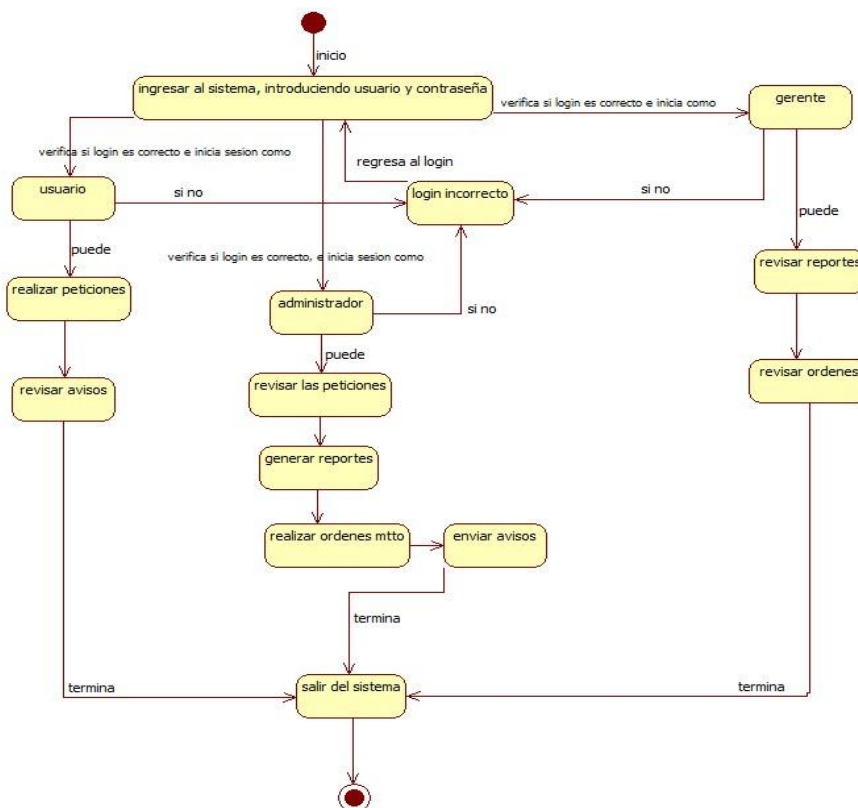


Figura 3.13 Diagrama de Estado



---

### 3.3 DESARROLLO DEL SISTEMA

En el desarrollo del sistema realizamos la programación del sistema, siguiendo el diseño elaborado anteriormente. Las tablas del modelo relacional se transforman en tablas de la base de datos, y de esta manera se permitirá agregar y manipular los datos.

En este capítulo se presenta la descripción de las principales funciones del sistema, así como el código. Además se muestra la interfaz de cada una de las pantallas principales del sistema.

#### 3.3.1 HERRAMIENTAS DEL SISTEMA

La implementación del sistema se realizó de la siguiente manera: la base de datos se creó en MySQL, la programación de las interfaces se realizó en PHP utilizando el servidor APACHE, instalando el programa WampServer que contiene lo mencionado anteriormente y de esta manera nos permite crear aplicaciones para visualizar editar y actualizar información de la base de datos.

#### 3.3.2 INTERFACES

Las interfaces se realizaron de la siguiente manera:

1. **Menú principal o de Bienvenida.** Es el menú que contiene información acerca del sistema, y de la empresa se aprecia en la figura 3.14 y 3.15 también se muestra la ventana de login en la figura 3.16 donde se presenta un menú con las opciones de cómo poder ingresar al sistema ya sea como usuario (empleado), administrador o gerente y al dar cualquiera de estas opciones aparecerá una pantalla con un formulario en donde se solicita el usuario y la contraseña como se muestra en la figura 3.17 dependiendo de cómo se desea ingresar si los datos son correctos se accederá a la página de usuarios, administrador o gerente, si no aparecerá un mensaje de error y regresará a la pantalla de login para poder ingresar nuevamente.



Figura 3.14 Menú Principal



Figura 3.15 Interfaz de información sobre La empresa

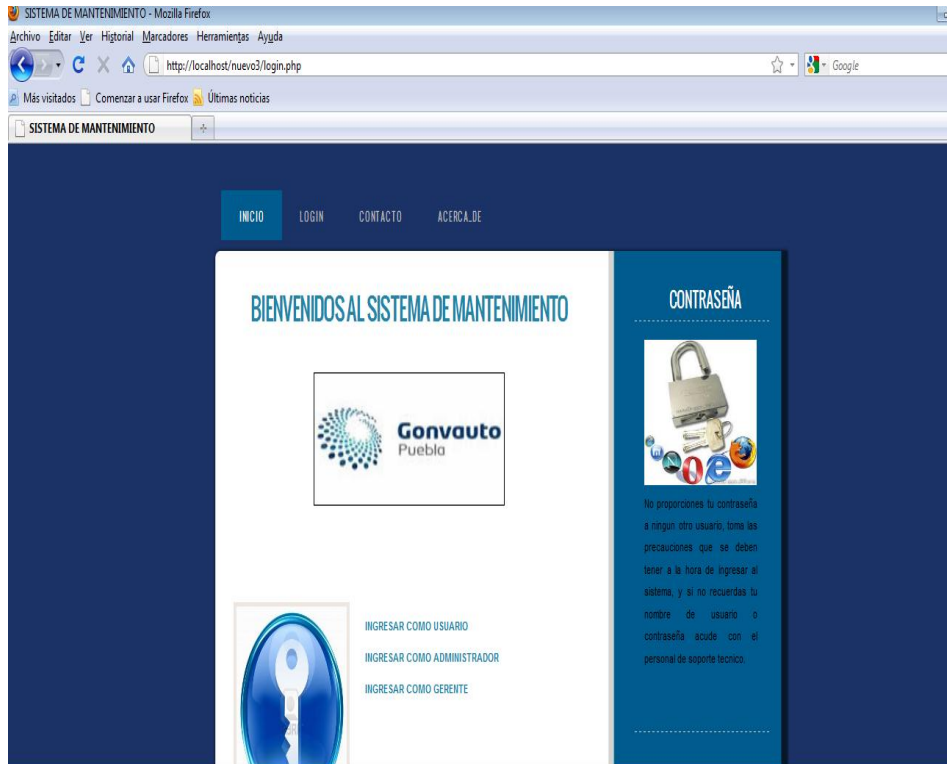


Figura 3.16 Interfaz login y opciones como ingresar al sistema



Figura 3.17 Interfaz de usuario y contraseña



2. **Menú de Usuarios (empleados).** En donde se presenta una interfaz con menús u opciones que tienen los usuarios figura 3.18, como primera interfaz se visualiza la de peticiones figura 3.19 que despliega un formulario, que el usuario debe llenar con información referente al estado de su equipo de cómputo y debe enviar al administrador para que pueda resolver su petición.

Posteriormente se tiene la interfaz de avisos en la figura 3.20 esta interfaz permite estar informados a los usuarios de los próximos mantenimientos, u otro tipo de aviso que el administrador requiera informar a los usuarios.

La interfaz de contacto figura 3.21 es una interfaz en donde los usuarios podrán contar con información de cómo poder contactar a los administradores, la interfaz de Acerca\_de figura 3.22 es la misma que la de la interfaz principal y se mantiene así para que nuestros usuarios tengan presentes las políticas de la empresa para futuras auditorias.



Figura 3.18 Interfaz principal de los Usuarios



Figura 3.19 Interfaz de peticiones

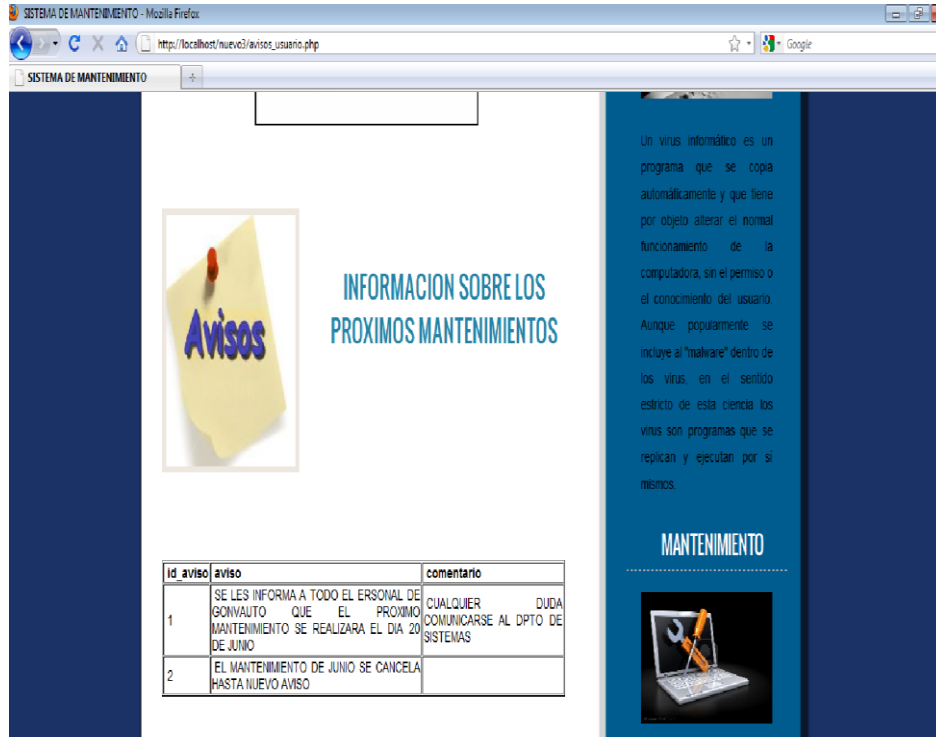


Figura 3.20 Interfaz de Avisos



Figura 3.21 Interfaz de Contacto

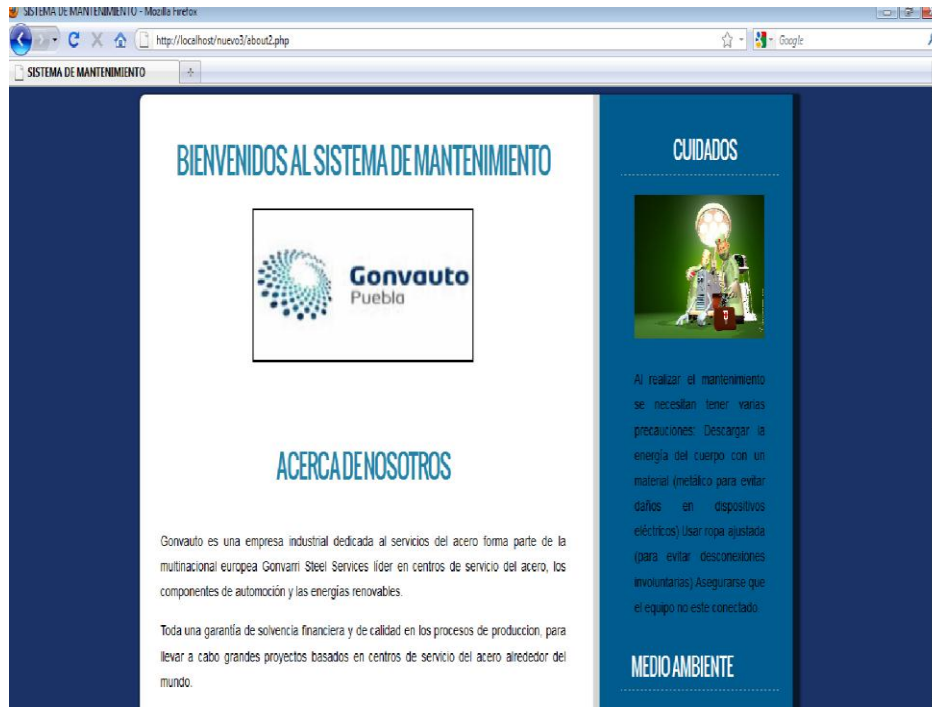


Figura 3.22 Interfaz de Información para Usuarios

- Menú de Administradores.** Se presenta una interfaz principal para los administradores figura 3.23 que son los que tendrán acceso a básicamente toda la información del sistema, posteriormente se muestra la interfaz de consultas en la cual se presentan las opciones de consultar los datos de equipo y de usuario figura 3.24 al dar clic en alguna de las opciones se muestra otra pagina donde contiene información de los datos de los usuarios figura 3.25 y de los equipos figura 3.26.

En la figura 3.27 se muestra la interfaz de insertar información de datos en la tabla de los usuarios y de los equipos al dar clic en alguna de estas opciones aparecerá otra pagina dependiendo de donde se va a insertar información si en la tabla de usuarios figura 3.28 o en la de equipos figura 3.29.

La interfaz de Eliminar datos se visualiza en la figura 3.30 donde también el administrador podrá elegir que información desea eliminar, si es en la tabla de usuarios como se muestra en la figura 3.31 o en la tabla de equipos como se visualiza en la figura 3.32.

Posteriormente se presenta la interfaz de Actualizar información en la figura 3.33 que también cuenta con las opciones de actualizar información de los usuarios como se aprecia en la figura 3.34 o actualizar información de los equipos como se muestra en la figura 3.35.

En las figuras 3.36 y 3.37 se presenta la interfaz de Revisar peticiones, estas peticiones son las que nuestros usuarios envían con información del equipo de cómputo que tenga algún problema y requiera solución de parte del administrador.



También se visualiza en el menú de los administradores la interfaz de Búsqueda figura 3.38 donde el administrador podrá buscar datos de los usuarios y lo podrá realizar por nombre, apellido o por su password como se muestra en la figura 3.39.

Se visualiza la interfaz de reportes figura 3.40 donde al dar clic el sistema le permite al administrador llenar un formulario figura 3.41 donde tendrá que llenar con información acerca de los mantenimientos y así poder realizar los reportes, que podrá visualizar los gerentes en el menú de Gerentes.

Posteriormente se presenta la interfaz de Órdenes figura 3.42 donde al dar clic al administrador se le presenta un formulario 3.43 donde tendrá que insertar información acerca del material que requiera surtir, así como una tabla donde podrá visualizar las anteriores ordenes.

En la figura 3.44 se muestra la interfaz de Avisos donde al dar clic se presenta un formulario figura 3.45 que el administrador tendrá que llenar con información sobre los próximos mantenimientos o cualquier otra información que desee publicar y enviar a los usuarios.



Figura 3.23 Interfaz principal de Administradores



Figura 3.24 Interfaz Consulta de Datos

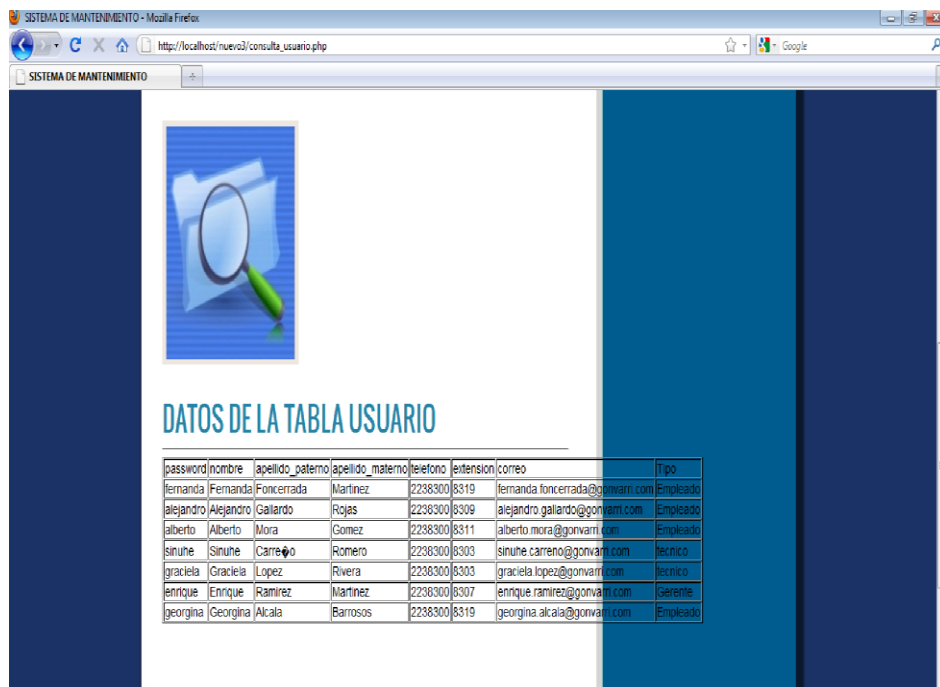


Figura 3.25 Interfaz Consulta de Datos de Usuario

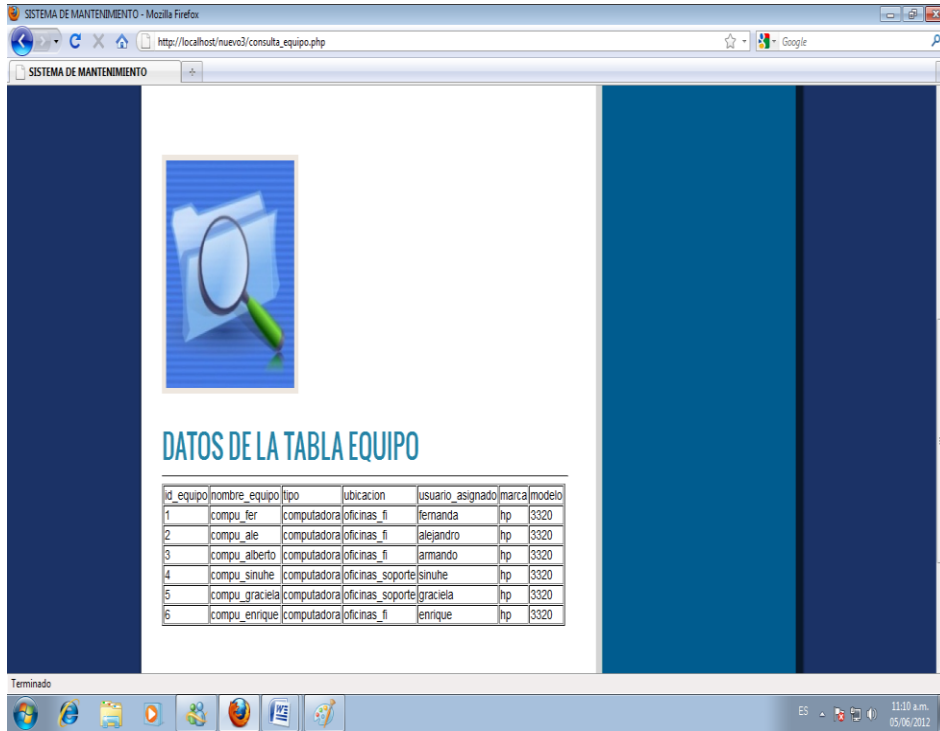


Figura 3.26 Interfaz Consulta de Datos de Equipo



Figura 3.27 Interfaz de Insertar Información

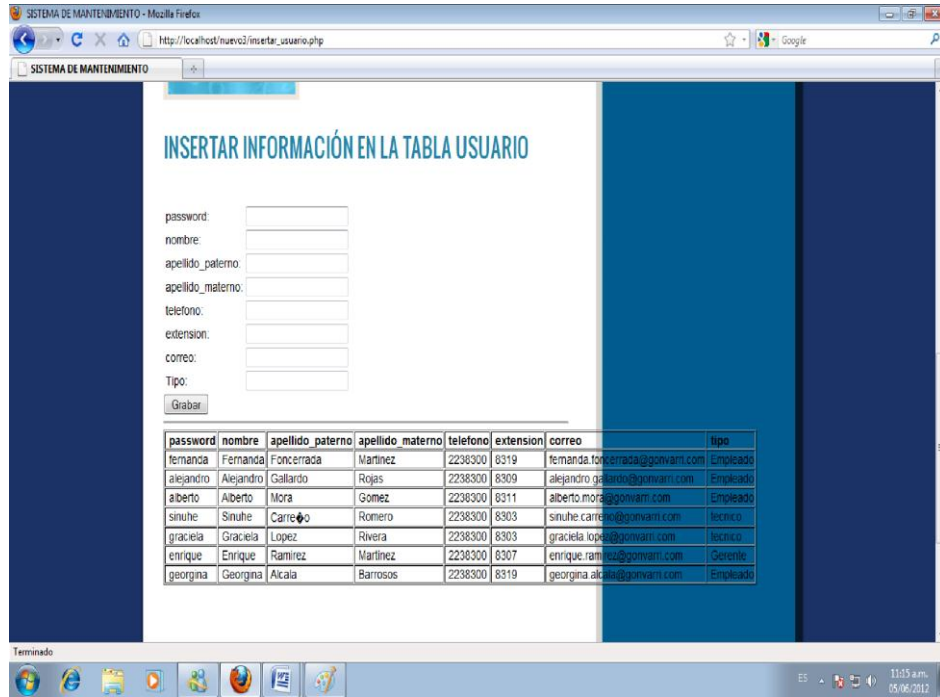


Figura 3.28 Interfaz de Insertar Información en Usuarios

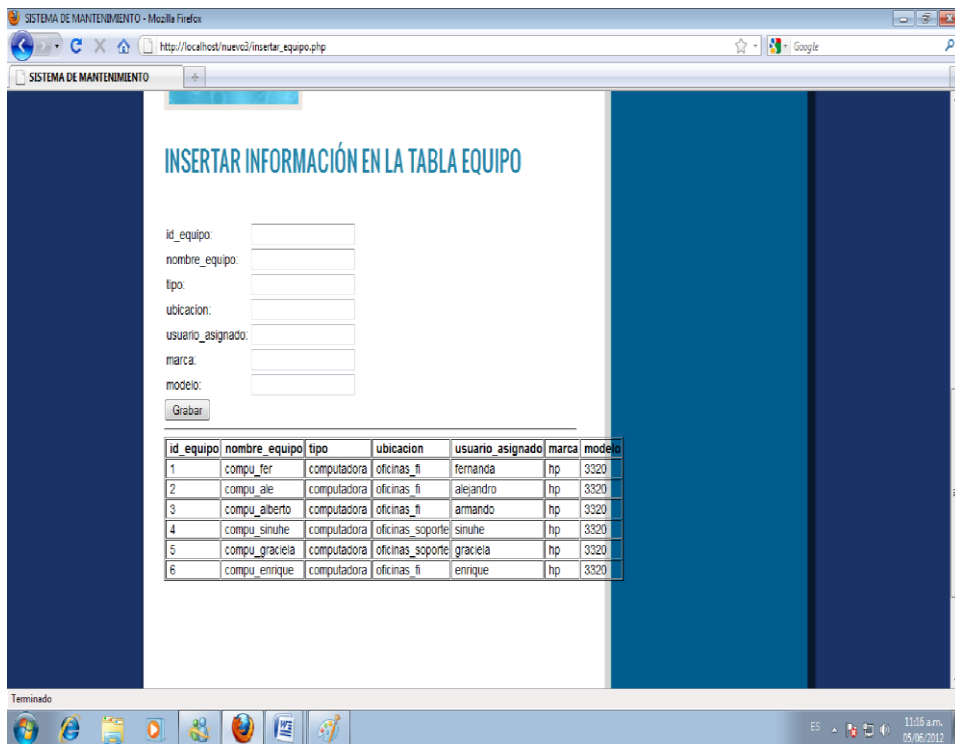


Figura 3.29 Interfaz de Insertar Información en Equipos

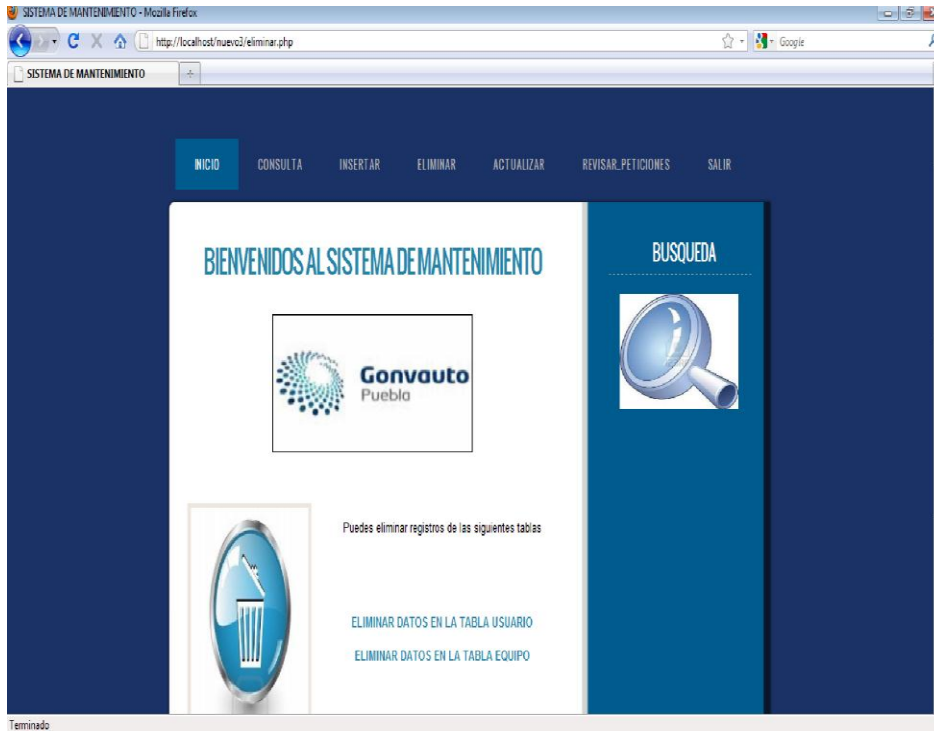


Figura 3.30 Interfaz de Eliminar Información

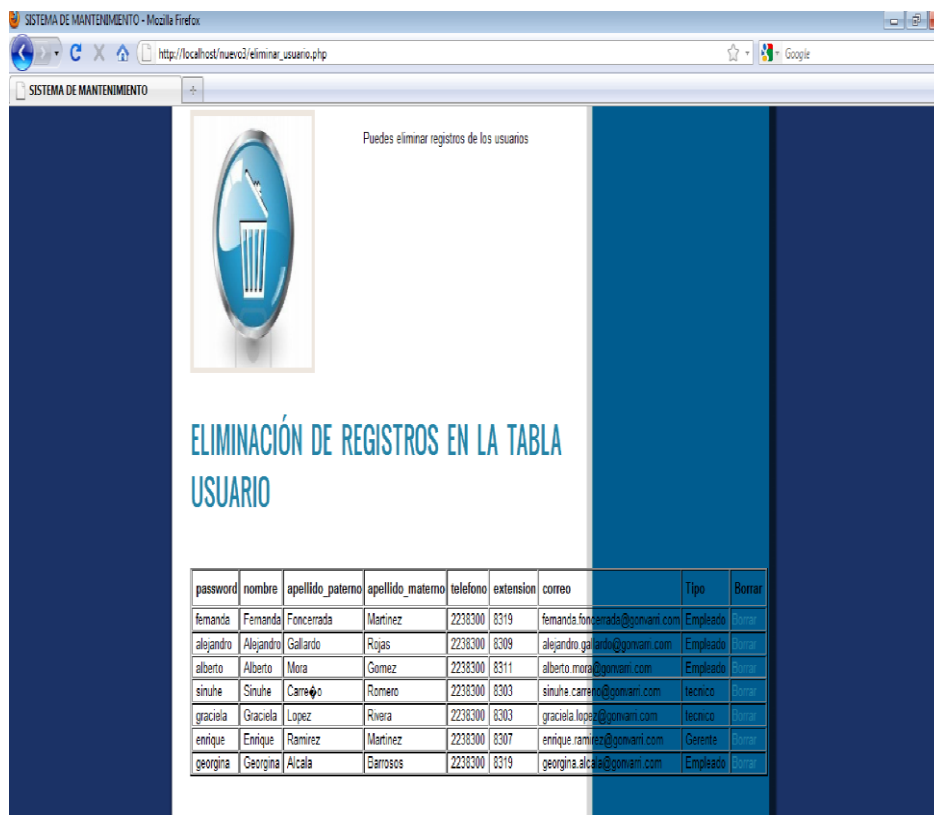


Figura 3.31 Interfaz de Eliminar Información de Usuarios



Figura 3.32 Interfaz de Eliminar Información de Equipos

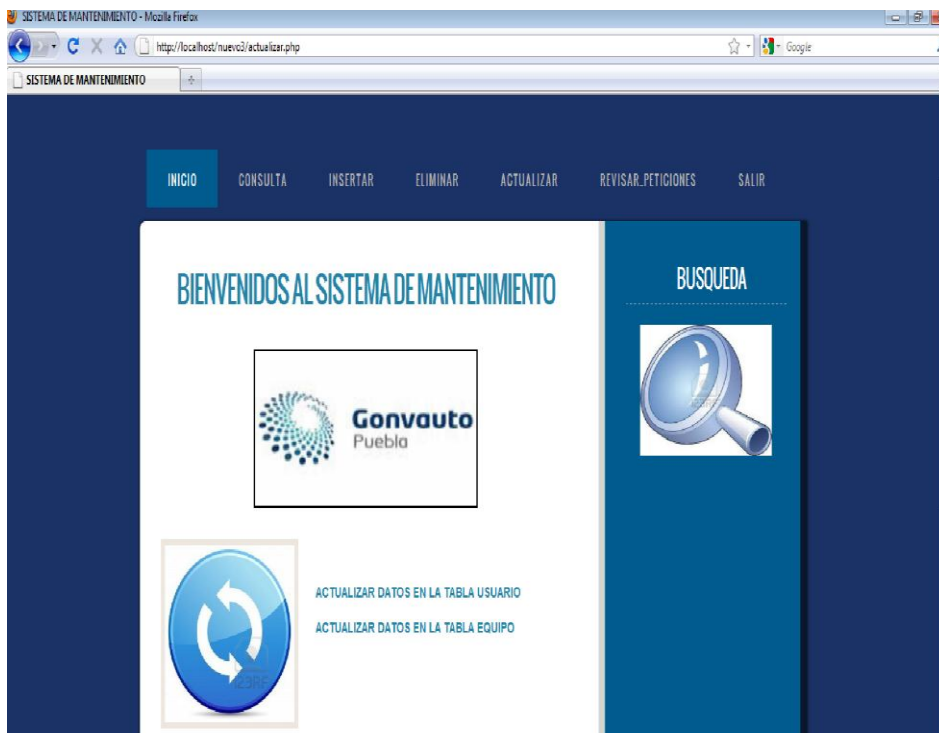


Figura 3.33 Interfaz de Actualizar Información



SISTEMA DE MANTENIMIENTO - Mozilla Firefox  
http://localhost/nuevo3/actualizar\_usuario.php

SISTEMA DE MANTENIMIENTO

Actualizar registros  
NOMBRE A ACTUALIZAR  
Alberto ▾  
Nuevo nombre  
 Actualizar

APELLIDO PATERNO A ACTUALIZAR  
Alcala ▾  
Nuevo apellido\_paterno  
 Actualizar

APELLIDO MATERNO A ACTUALIZAR  
Barros ▾  
Nuevo apellido\_materno  
 Actualizar

PASSWORD A ACTUALIZAR  
alberto ▾  
Nuevo password  
 Actualizar

Figura 3.34 Interfaz de Actualizar usuarios

SISTEMA DE MANTENIMIENTO - Mozilla Firefox  
http://localhost/nuevo3/actualizar\_equipo.php

SISTEMA DE MANTENIMIENTO

Actualizar registros  
NOMBRE DE EQUIPO A ACTUALIZAR  
compu\_alberto ▾  
Nuevo nombre  
 Actualizar

TIPO A ACTUALIZAR  
computadora ▾  
Nuevo tipo  
 Actualizar

UBICACION A ACTUALIZAR  
oficinas\_f ▾  
Nuevo ubicacion  
 Actualizar

USUARIO ASIGNADO A ACTUALIZAR  
alejandro ▾  
Nuevo Usuario  
 Actualizar

Figura 3.35 Interfaz de Actualizar Equipos

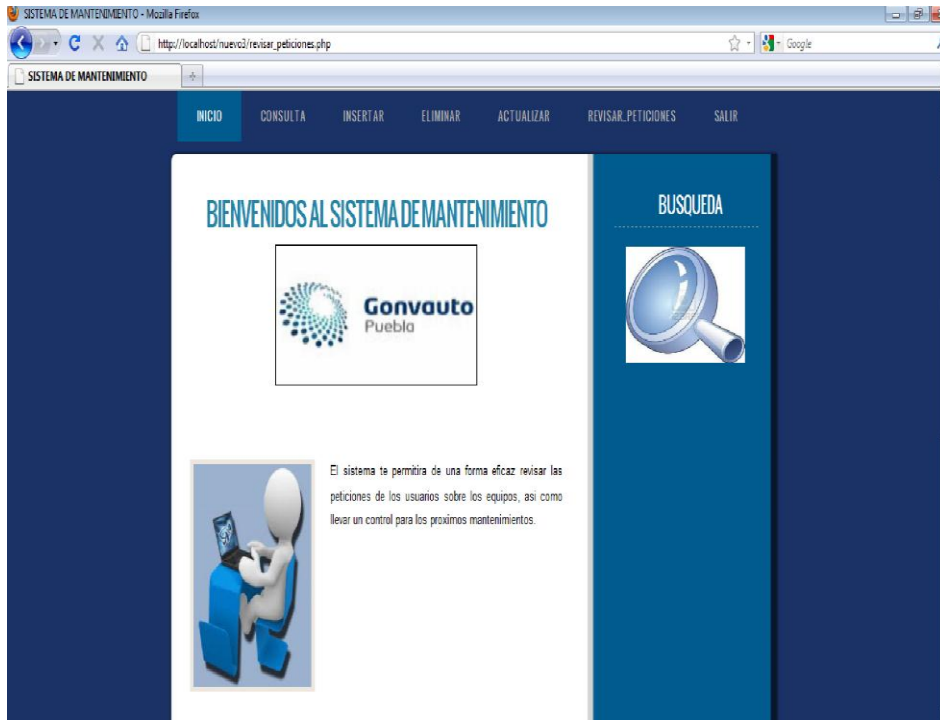


Figura 3.36 Interfaz de Revisar Peticiones

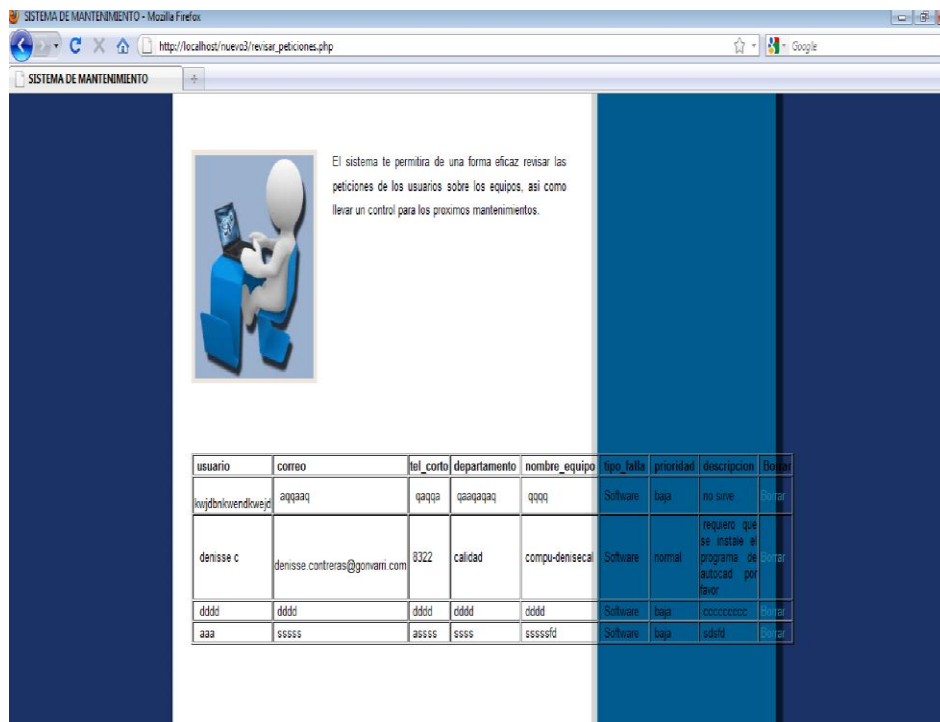


Figura 3.37 Interfaz de Revisar Peticiones 2



Figura 3.38 Interfaz de Búsqueda

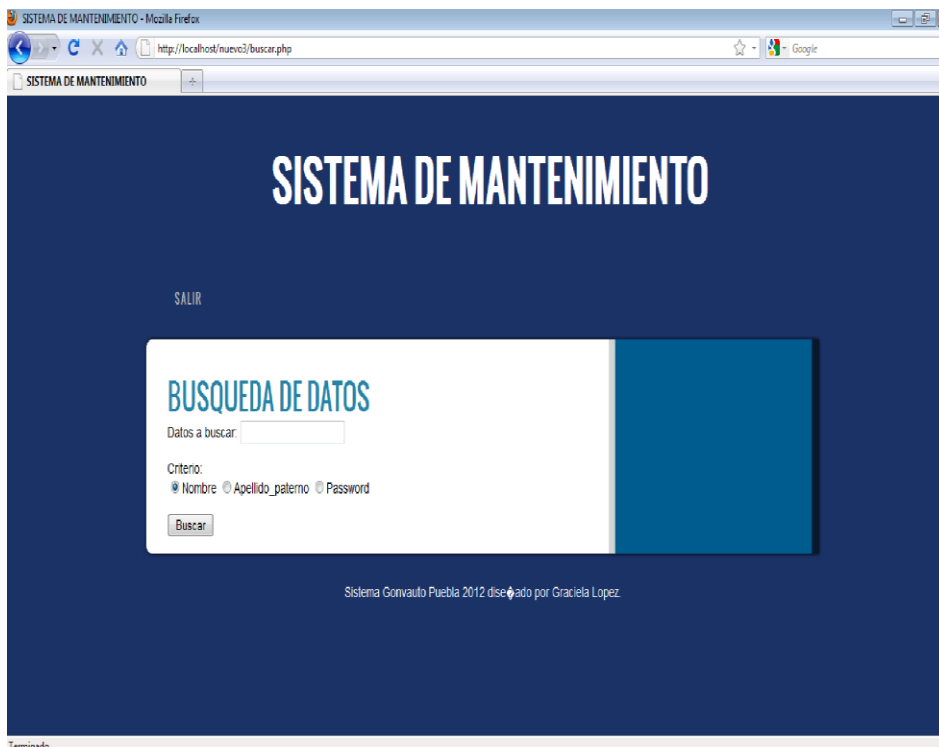


Figura 3.39 Interfaz de Búsqueda por opciones



Figura 3.40 Interfaz de Reportes en menú de administrador

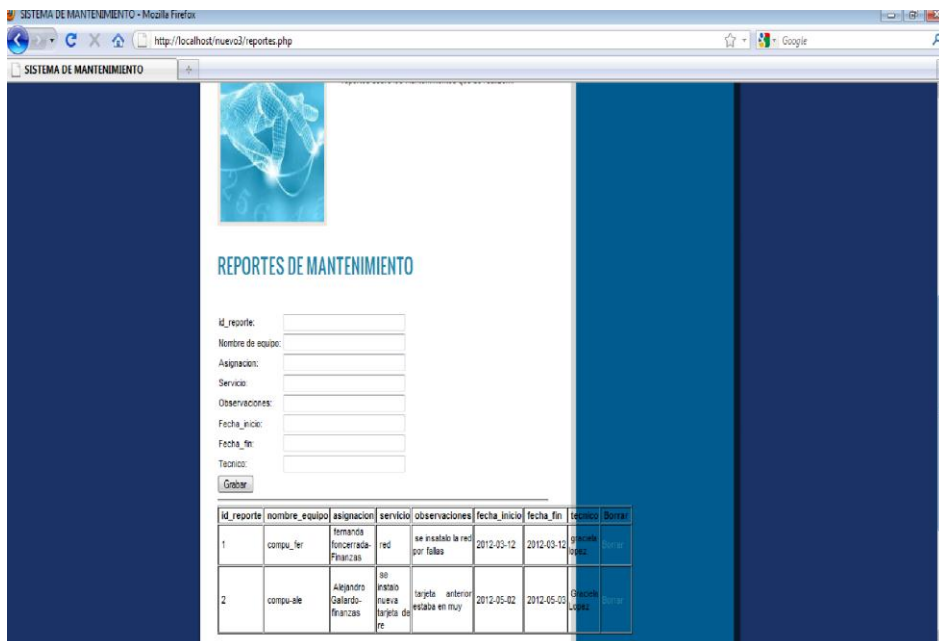


Figura 3.41 Interfaz de Reportes



Figura 3.42 Interfaz de Ordenes en Menú de Administrador

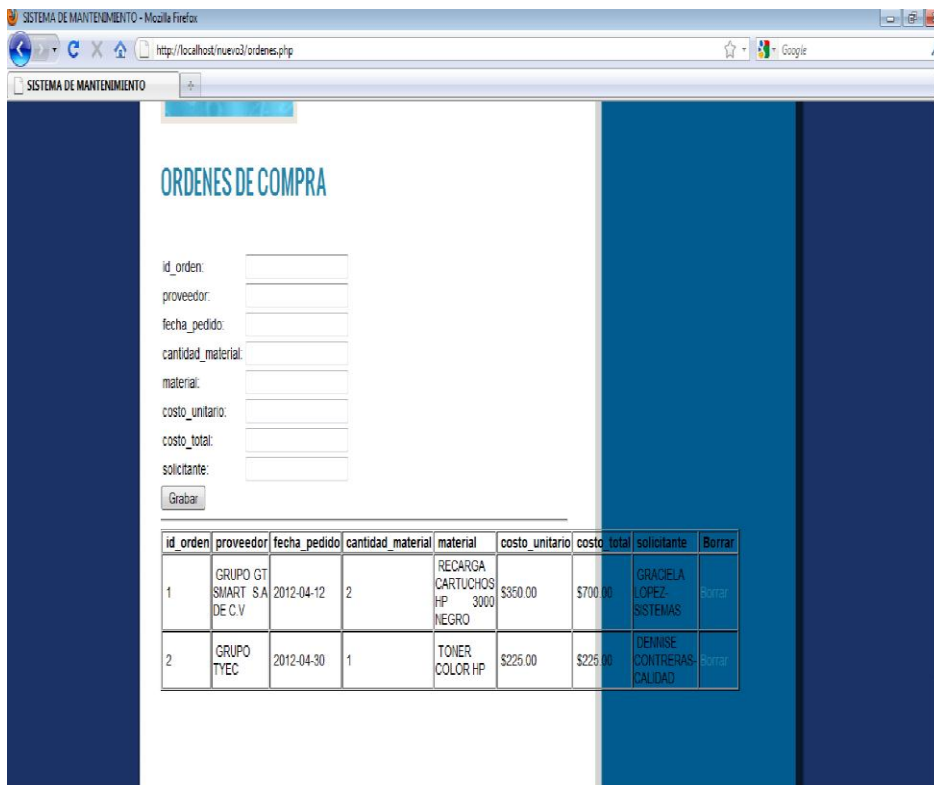


Figura 3.43 Interfaz de Órdenes de compra



Figura 3.44 Interfaz de Avisos en Menú de Administrador

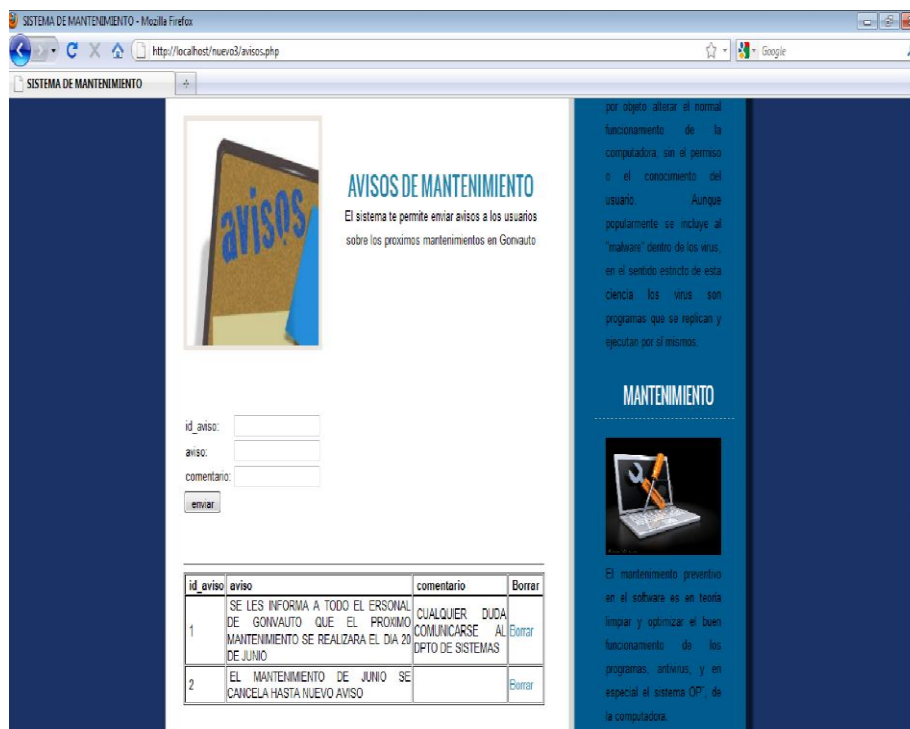


Figura 3.45 Interfaz de Avisos



4. **Menú de Gerentes.** Se presenta una interfaz principal para los gerentes figura 3.46 donde posteriormente se muestra la interfaz de visualizar reportes figura 3.47 estos reportes los genera el administrador y los envía al gerente para que pueda estar enterado del estado de todos los equipo de la empresa. En la figura 3.48 se presenta la interfaz de visualizar ordenes y Se mantiene la interfaz de Contacto figura 3.49 del menú principal al igual que la interfaz de Acerca\_ de.



Figura 3.46 Interfaz Principal de los Gerentes



Figura 3.47 Interfaz Visualizar Reportes



El sistema de mantenimiento te permite visualizar las ordenes de compra realizadas en Gonvauto.

### ORDENES

id_orden	proveedor	fecha_pedido	cantidad_material	material	costo_unitario	costo_total	solicitante
1	GRUPO GT SMART S.A DE C.V	2012-04-12	2	RECARGA CARTUCHOS HP 3000 NEGRO	\$350.00	\$700.00	GRACIELA LOPEZ-SISTEMAS
2	GRUPO TYEC	2012-04-30	1	TONER COLOR HP	\$225.00	\$225.00	DENNISE CONTRERAS-CALIDAD

Figura 3.48 Interfaz Visualizar Ordenes

**BIENVENIDOS AL SISTEMA DE MANTENIMIENTO**

**CONTACTO**

Cualquier duda contactanos en el departamento de soporte tecnico.  
Con gusto te ayudaremos.

**TELEFONOS**

(222) 223 83 03

**CORREOS**

santhe.carrero@gonvauto.com  
graciela.lopez@gonvauto.com

Figura 3.49 Interfaz Contacto en Menú de Gerentes



### 3.4 PRUEBAS

Después de implementar el sistema con los requerimientos planteados es necesario validarlo para tener la seguridad de que el Sistema funciona correctamente; para ello es necesario hacer las pruebas. A continuación se muestran las pruebas que se realizaron para verificar el funcionamiento.

En la figura 3.50 se muestra una prueba de cómo ingresa al sistema un usuario en la figura 3.51 se presenta una prueba sobre ingresar al sistema como administrador y la figura 3.52 muestra la prueba de cómo ingresa al sistema un gerente, si los datos son correctos de visualizara la interfaz de usuarios si no se apreciara una pantalla de error indicando que está mal, si el password como se muestra en la figura 3.53 o el login de usuario como se muestra en la figura 3.54 y regresara a la pantalla de login para que pueda volver a ingresar.



Figura 3.50 Ingresar al Sistema como Usuario



Figura 3.51 Ingresar al Sistema como Administrador



Figura 3.52 Ingresar al Sistema como Gerente

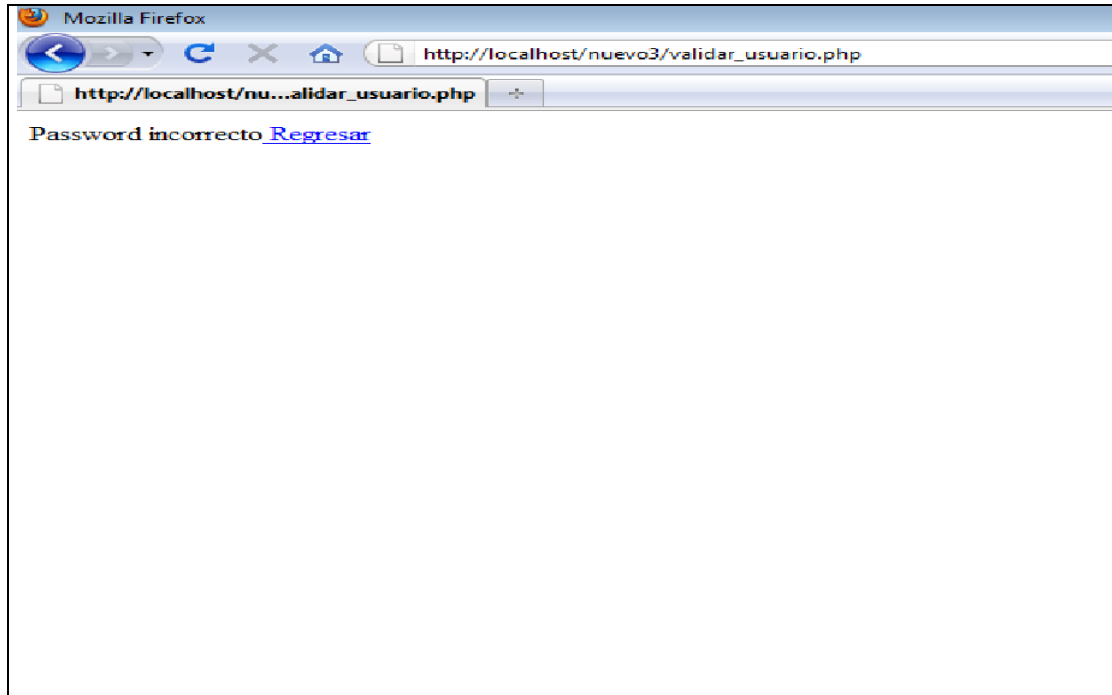


Figura 3.53 Mensaje de Error de Password

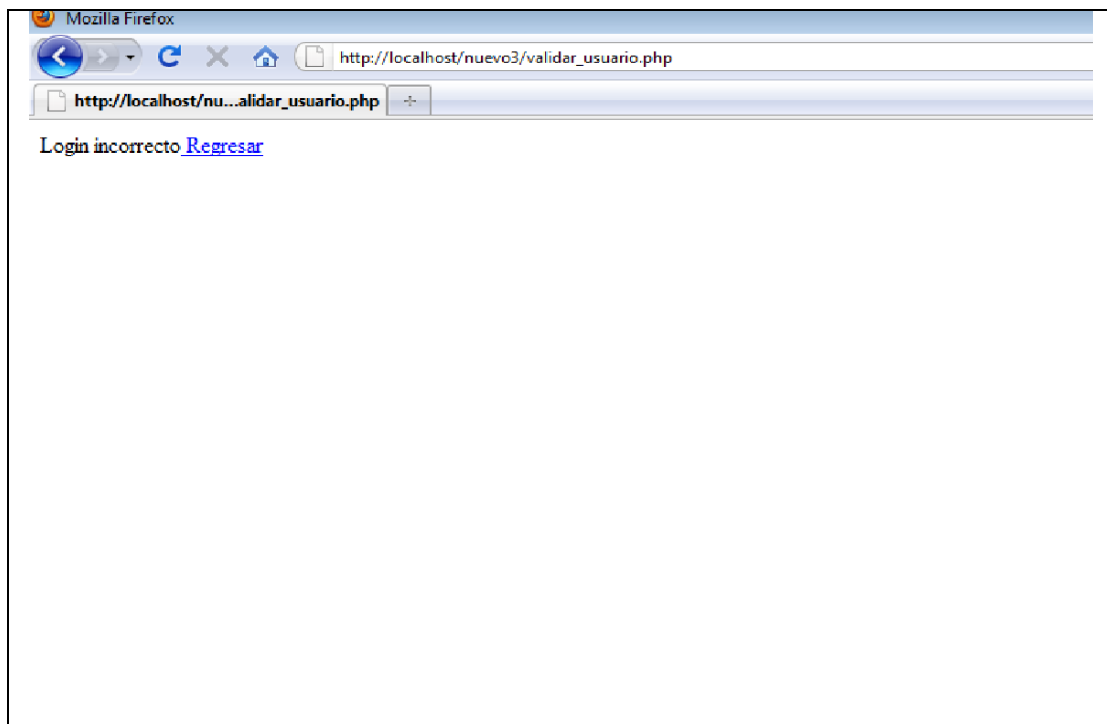


Figura 3.54 Mensaje de Error de Login



En la figura 3.55 se muestra una prueba de cómo los usuarios realizan una petición llenando el formulario con la información que se le solicita, para posteriormente enviarla al administrador.

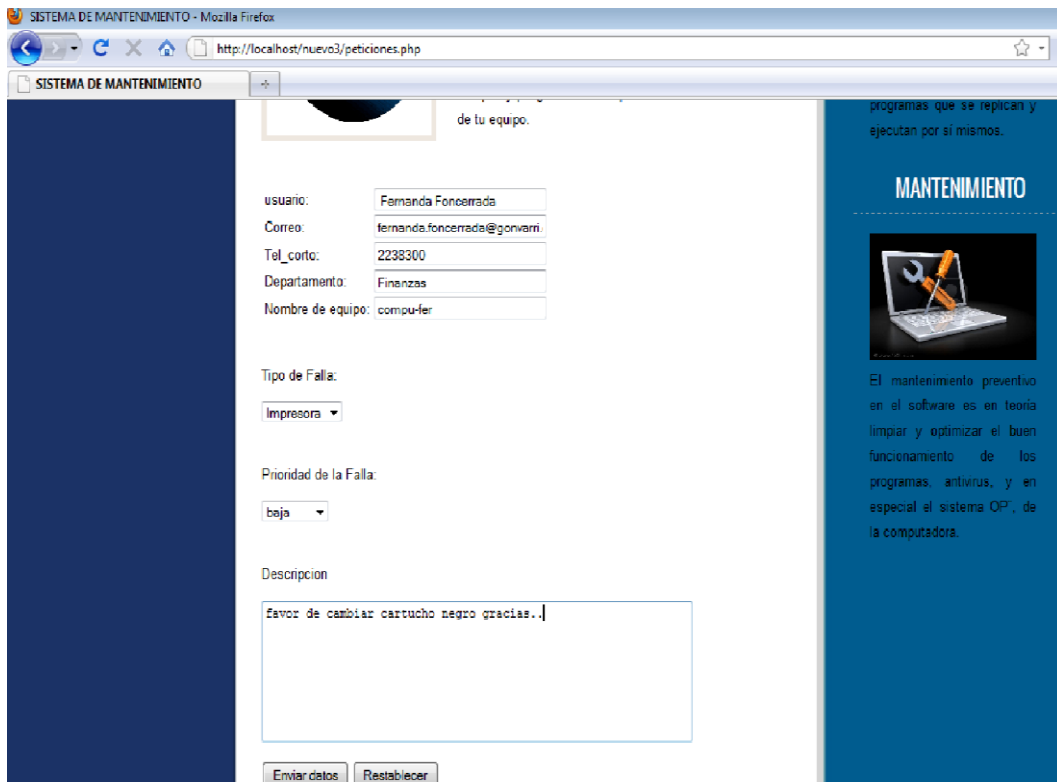


Figura 3.55 Como Realizar una Petición

En la figura 3.56 se muestra una prueba de cómo el administrador da de alta un usuario y apreciamos que antes de grabar efectivamente no se encuentra el usuario en la tabla que se muestra abajo del formulario y en la figura 3.57 se aprecia el registro del nuevo usuario.

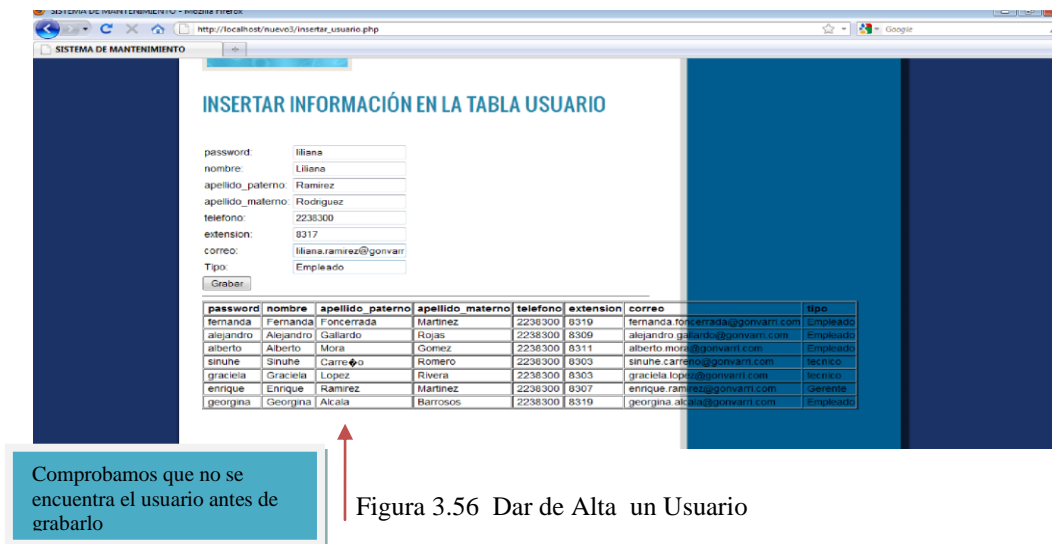


Figura 3.56 Dar de Alta un Usuario



password:

nombre:

apellido\_paterno:

apellido\_materno:

telefono:

extension:

correo:

Tipo:

password	nombre	apellido_paterno	apellido_materno	telefono	extension	correo	tipo
fernanda	Fernanda	Foncerrada	Martinez	2238300	8319	fernanda.foncerrada@gonviam.com	Empleado
alejandro	Alejandro	Galarzo	Rojas	2238300	8309	alejandro.galarzo@gonviam.com	Empleado
alberto	Alberto	Mora	Gomez	2238300	8311	alberto.mora@gonviam.com	Empleado
sinuhe	Sinuhe	Carreño	Romero	2238300	8303	sinuhe.carreno@gonviam.com	Tecnico
graciela	Graciela	Lopez	Rivera	2238300	8303	graciela.lopez@gonviam.com	Tecnico
enrique	Enrique	Ramirez	Martinez	2238300	8307	enrique.ramirez@gonviam.com	Gerente
georgina	Georgina	Alcala	Barrosos	2238300	8319	georgina.alcala@gonviam.com	Empleado
liliana	Liliana	Ramirez	Rodriguez	2238300	8317	liliana.ramirez@gonviam.com	Empleado

Se aprecia el nuevo registro

Figura 3.57 Registro de Nuevo Usuario

En la figura 3.58 se muestra una prueba de cómo el administrador da de alta un equipo y apreciamos que antes de grabar efectivamente no se encuentra el equipo en la tabla que se muestra abajo del formulario y en la figura 3.59 se aprecia el registro del nuevo equipo.

INSERTAR INFORMACIÓN EN LA TABLA EQUIPO

id\_equipo:

nombre\_equipo:

tipo:

ubicacion:

usuario\_asignado:

marca:

modelo:

id_equipo	nombre_equipo	tipo	ubicacion	usuario_asignado	marca	modelo
1	compu_fer	computadora	oficinas_fi	fernanda	hp	8320
2	compu_ale	computadora	oficinas_fi	alejandro	hp	3320
3	compu_alberto	computadora	oficinas_fi	armando	hp	3320
4	compu_sinuhe	computadora	oficinas_soporte	sinuhe	hp	3320
5	compu_graciela	computadora	oficinas_soporte	graciela	hp	3320
6	compu_enrique	computadora	oficinas_fi	enrique	hp	3320

Comprobamos que no se encuentra el usuario antes de grabarlo

Figura 3.58 Dar de Alta Nuevo Equipo

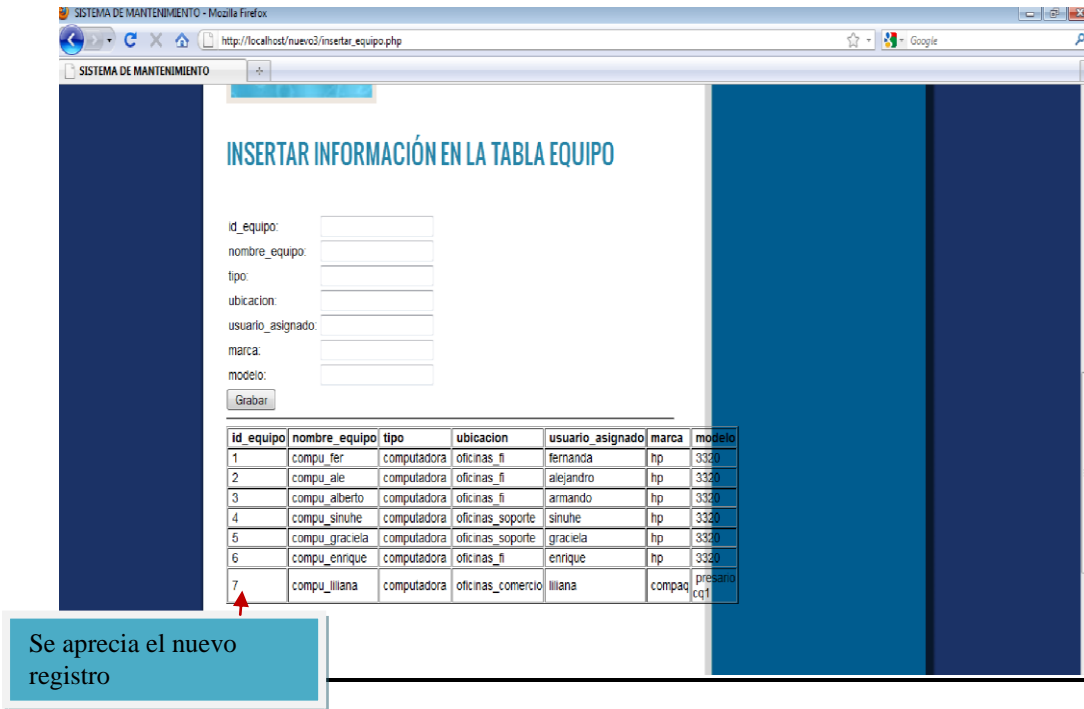


Figura 3.59 Registro de Nuevo Equipo

En las figuras 3.60 y 3.61 se aprecia una prueba de cómo el administrador elimina un usuario y se comprueba que en realidad se elimino el usuario, seguido de las figuras 3.62 y 3.63 donde se realiza la prueba de eliminar un equipo y comprobar que efectivamente se elimino en la tabla.



Figura 3.60 Eliminar registros de un Usuario



SISTEMA DE MANTENIMIENTO - Mozilla Firefox

http://localhost/nuevo3/eliminar\_usuario.php

### ELIMINACIÓN DE REGISTROS EN LA TABLA USUARIO

password	nombre	apellido_paterno	apellido_materno	telefono	extension	correo	Tipo	Borrar
fernanda	Fernanda	Foncerrada	Martinez	2238300	8319	fernanda.foncerrada@gonvarm.com	Empleado	Borrar
alejandro	Alejandro	Galiardo	Rojas	2238300	8309	alejandro.galiardo@gonvarm.com	Empleado	Borrar
alberto	Alberto	Mora	Gomez	2238300	8311	alberto.mora@gonvarm.com	Empleado	Borrar
sinuhe	Sinuhe	Carreño	Romero	2238300	8303	sinuhe.carreno@gonvarm.com	Tecnico	Borrar
graciela	Graciela	Lopez	Rivera	2238300	8308	graciela.lopez@gonvarm.com	Tecnico	Borrar
enrique	Enrique	Ramirez	Martinez	2238300	8307	enrique.ramirez@gonvarm.com	Gerente	Borrar
georgina	Georgina	Alcala	Barrosos	2238300	8319	georgina.alcala@gonvarm.com	Empleado	Borrar

Figura 3.61 Comprobar la baja de un Usuario

SISTEMA DE MANTENIMIENTO - Mozilla Firefox

http://localhost/nuevo3/eliminar\_equipo.php

### ELIMINACIÓN DE REGISTROS EN LA TABLA EQUIPO

id_equipo	nombre_equipo	tipo	ubicacion	usuario_asignado	marca	modelo	Borrar
1	compu_fer	computadora	oficinas_fi	fernanda	hp	332	Borrar
2	compu_ale	computadora	oficinas_fi	alejandro	hp	332	Borrar
3	compu_aberto	computadora	oficinas_fi	amando	hp	332	Borrar
4	compu_sinuhe	computadora	oficinas_soporte	sinuhe	hp	332	Borrar
5	compu_graciela	computadora	oficinas_soporte	graciela	hp	332	Borrar
6	compu_enrique	computadora	oficinas_fi	enrique	hp	332	Borrar
7	compu_liliana	computadora	oficinas_comercio	liliana	compaq	presario p41	Borrar

Figura 3.62 Eliminar Registros de un Equipo

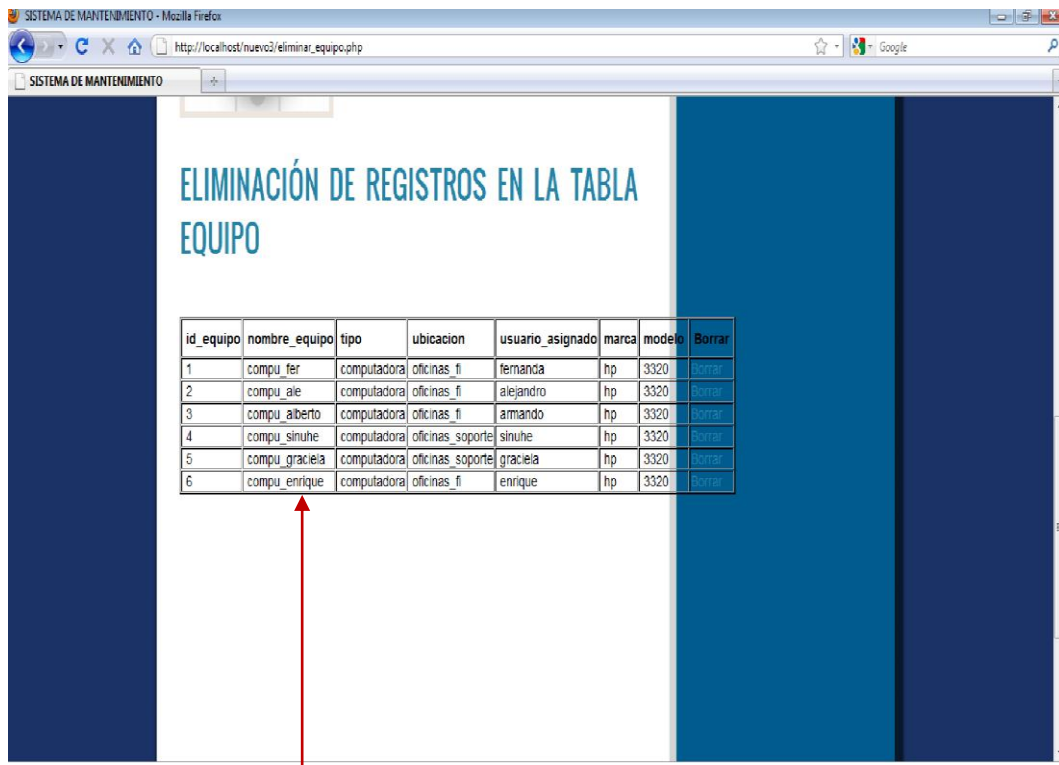


Figura 3.63 Comprobar la baja de un Equipo

Se presenta una prueba de Actualizar información en los usuarios En las figuras 3.64 y 3.65 y se comprueba que en realidad se actualizo el usuario, seguido de las figuras 3.66 y 3.67 donde se realiza la prueba de Actualizar información en un equipo y comprobar que efectivamente se actualizo la información del equipo.



Figura 3.64 Actualizar información en los Usuarios



SISTEMA DE MANTENIMIENTO - Mozilla Firefox  
http://localhost/nuevo3/verdatos.php

# SISTEMA DE MANTENIMIENTO

## DATOS DE LA BD MANTENIMIENTO

[SALIR](#)

password	nombre	apellido_paterno	apellido_materno	telefono	extension	correo	tipo
fernanda	Fernanda	Foncerrada	Martinez	2238300	8319	fernanda.foncerrada@gonvami.com	Empleado
alejandra	Alejandra	Gallardo	Rojas	2238300	8309	alejandra.gallardo@gonvami.com	Empleado
alberto	Alberto	Mora	Gomez	2238300	8311	alberto.mora@gonvami.com	Empleado
sinuhe	Sinuhe	Carreno	Romero	2238300	8303	sinuhe.carreno@gonvami.com	tecnico
graciela	Graciela	Lopez	Rivera	2238300	8303	graciela.lopez@gonvami.com	tecnico
enrique	Enrique	Ramirez	Martinez	2238300	8307	enrique.ramirez@gonvami.com	Gerente
georgina	Georgina	Alcala	Barrosos	2238300	8319	georgina.alcala@gonvami.com	Empleado

Figura 3.65 Registro Actualizado de Usuarios

SISTEMA DE MANTENIMIENTO - Mozilla Firefox  
http://localhost/nuevo3/actualizar\_equipo.php



Actualizar registros  
NOMBRE DE EQUIPO A ACTUALIZAR  
compu\_alberto

Nuevo nombre  
Compu\_alberto2

TIPO A ACTUALIZAR  
computadora

Nuevo tipo

Figura 3.66 Actualizar Información en los Equipos



The screenshot shows a web browser window with the URL `http://localhost/nuevoc3/verdatos2.php`. The page title is 'SISTEMA DE MANTENIMIENTO'. Below the title, there is a section titled 'DATOS DE LA BD MANTENIMIENTO' with a 'SALIR' button. A table displays the following data:

id equipo	nombre equipo	tipo	ubicacion	usuario asignado	marca	modelo
1	compu_fer	computadora	oficinas_fi	fernanda	hp	3320
2	compu_ale	computadora	oficinas_fi	alejandro	hp	3320
3	Compu_alba	computadora	oficinas_fi	armando	hp	3320
4	compu_sinuhe	computadora	oficinas_soporte	sinuhe	hp	3320
5	compu_graciela	computadora	oficinas_soporte	graciela	hp	3320
6	compu_enrique	computadora	oficinas_fi	enrique	hp	3320

Figura 3.67 Registro Actualizado de los Equipos

Se muestra una prueba de búsqueda de datos en la figura 3.68 donde se realizara la búsqueda por nombre aunque se tengan otras opciones, si la búsqueda es satisfactoria aparecerá la pantalla de que la búsqueda fue satisfactoria como se muestra en la figura 3.69 y en caso contrario aparecerá la pantalla de la figura 3.70 donde nos indica que no se ha encontrado ningún registro.

The screenshot shows a web browser window with the URL `http://localhost/nuevoc3/buscar.php`. The page title is 'SISTEMA DE MANTENIMIENTO'. Below the title, there is a 'SALIR' button and a search form titled 'BUSQUEDA DE DATOS'. The form contains a text input field with the value 'Graciela', a 'Criterio' section with radio buttons for 'Nombre' (selected), 'Apellido\_paterno', and 'Password', and a 'Buscar' button. At the bottom of the page, there is a footer: 'Sistema Gonvauto Puebla 2012 diseñado por Graciela Lopez.'

Figura 3.68 Prueba Búsqueda de Datos

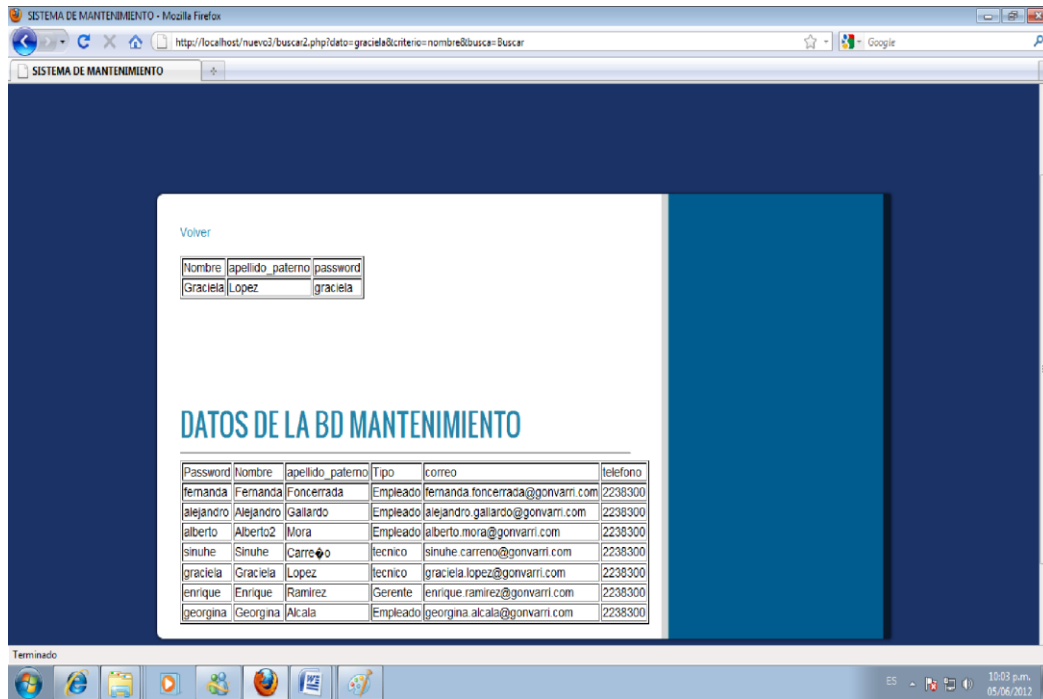


Figura 3.69 Búsqueda Satisfactoria



Figura 3.70 Búsqueda sin resultados



En la figura 3.71 se muestra una prueba de como se realizan los reportes que el administrador realiza y posteriormente el gerente puede visualizar en su menú, en la figura 3.72 se comprueba en la tabla que el reporte se genero, y por ultimo en la figura 3.73 se muestra una prueba de cómo eliminar los reportes que pudieran estar mal.



Figura 3.71 Elaboración de Reportes

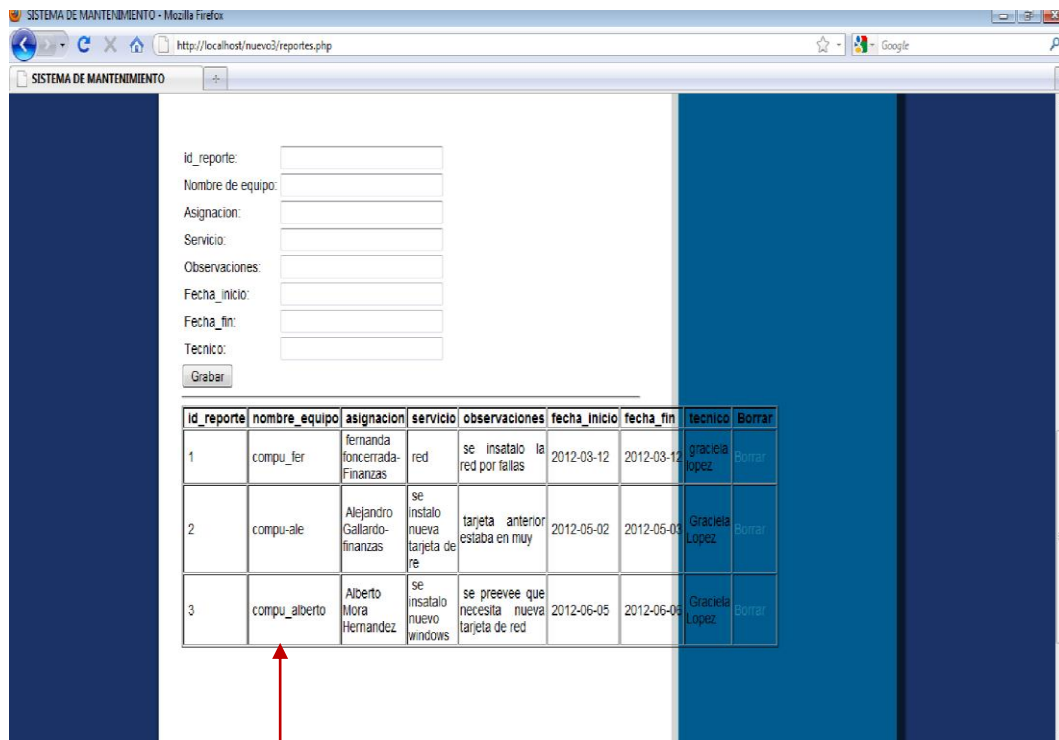


Figura 3.72 Comprobación de que Reporte se Genero

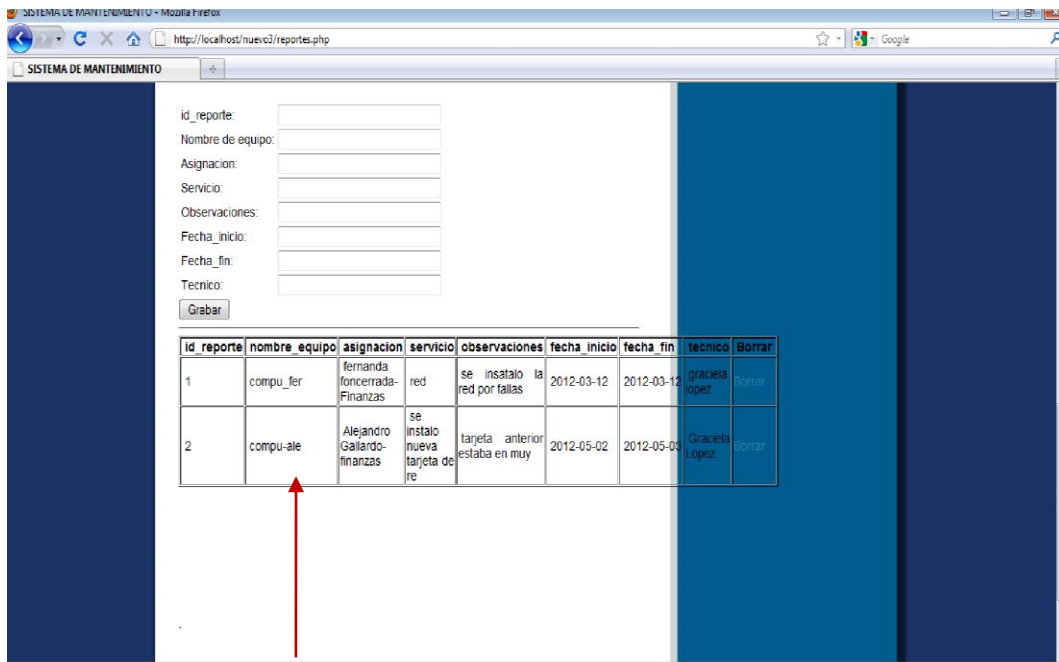


Figura 3.73 Eliminación de Reportes

En la figura 3.74 se muestra una prueba como se realizan las órdenes de compra que el administrador realiza, en la figura 3.75 se comprueba en la tabla que la orden se realizo, y por ultimo en la figura 3.76 se muestra una prueba de cómo eliminar las reportes que pudieran estar mal.

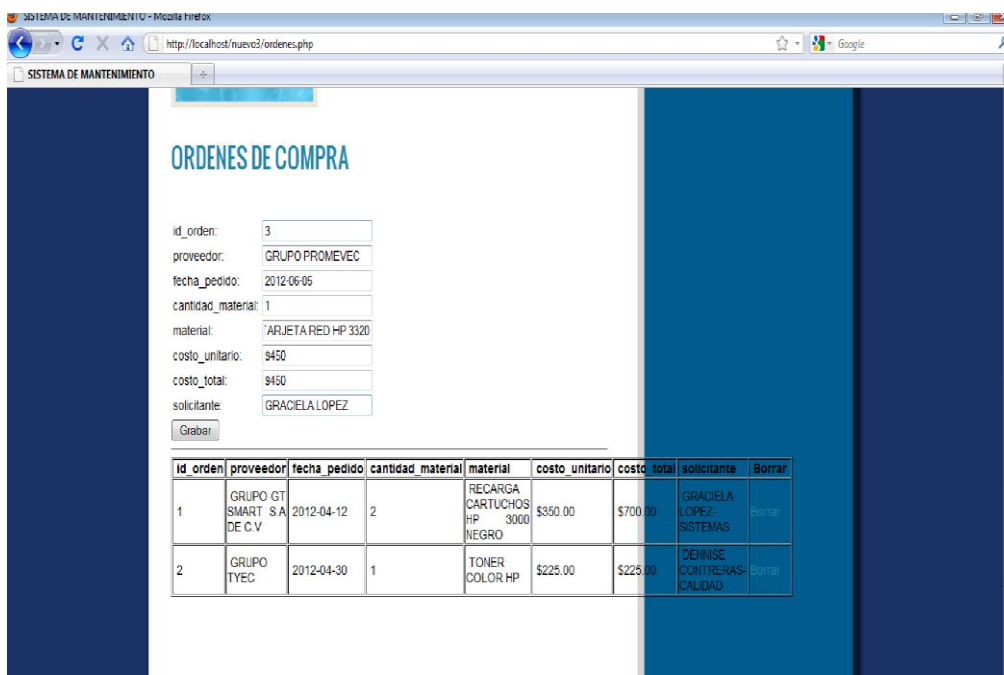


Figura 3.74 Elaboración de Órdenes de Compra

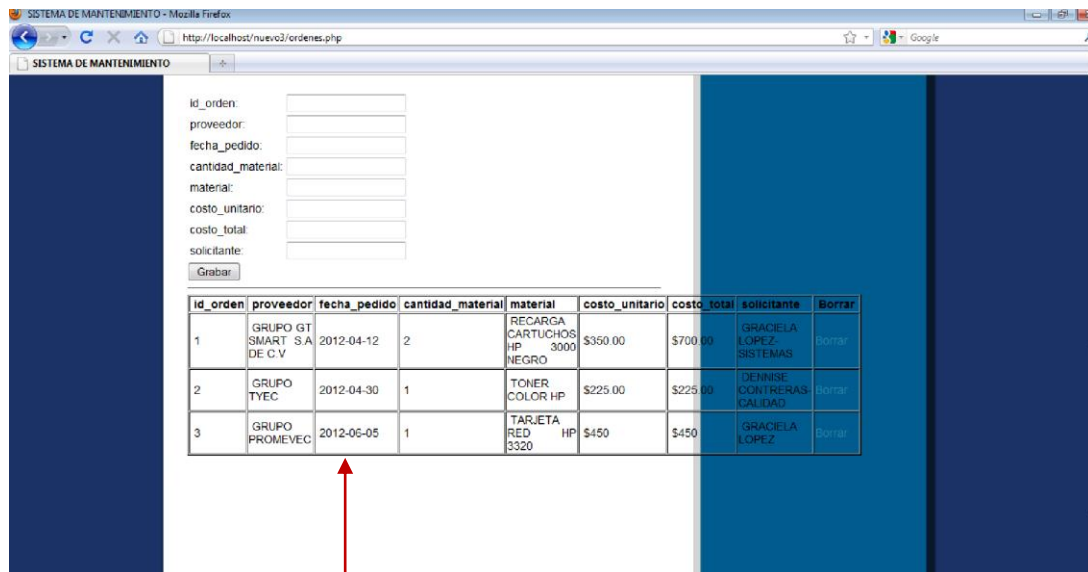


Figura 3.75 Comprobar que la Orden de Compra se Genero

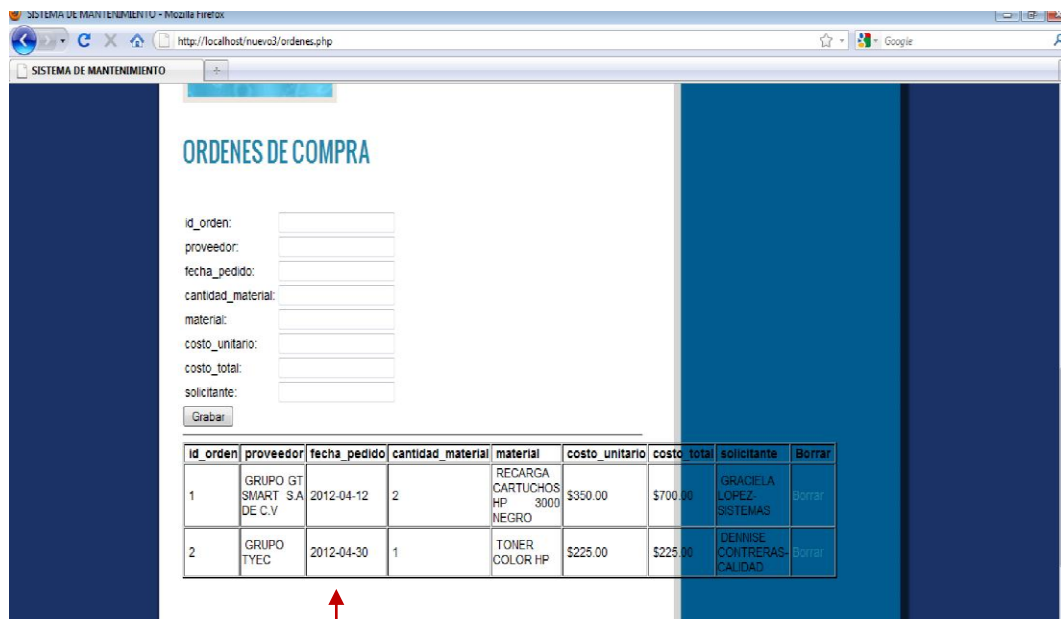


Figura 3.76 Eliminación de Órdenes de Compra

En la figura 3.77 se muestra una prueba como se realizan los avisos de mantenimiento que los usuarios podrán visualizar, en la figura 3.78 se comprueba en la tabla que el aviso se realizo, y por ultimo en la figura 3.79 se muestra una prueba de cómo eliminar los aviso que ya no sean necesarios.

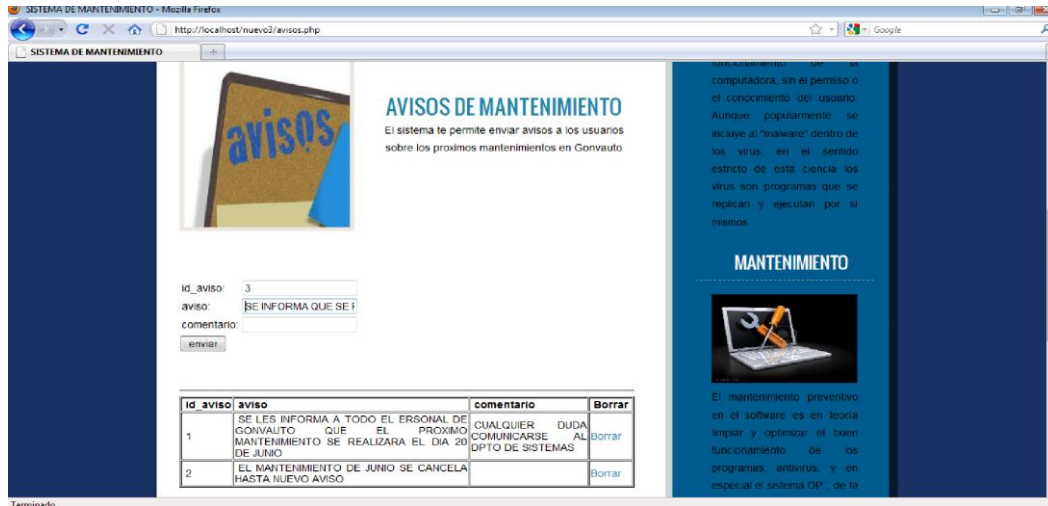


Figura 3.77 Elaboración de Avisos

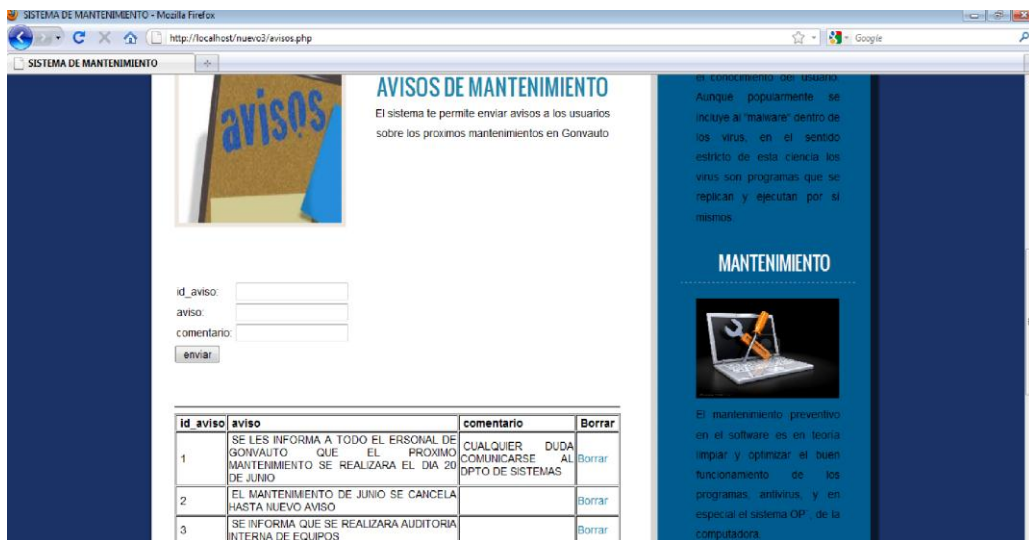


Figura 3.78 Comprobar que los Avisos se Generaron



Figura 3.79 Eliminación de Avisos



---

---

# *CAPÍTULO IV*

# *CONCLUSION*



---

---

## **CONCLUSION**

Se realizó el análisis de requisitos y se describió detalladamente, llevando una concordancia para obtener los resultados esperados.

El sistema cubrió las necesidades y las expectativas del cliente al realizar un sistema capaz de proporcionar servicios a los usuarios mediante las peticiones que se realizan y llevar así un buen control y gestión de los mantenimientos que se realizan en la empresa.

El sistema tal como se planteó se realizó mediante una página Web en donde se reconoció y complació los requerimientos del cliente. Para el desarrollo de éste, el usuario quedó satisfecho al visualizar todas las funciones que el sistema puede realizar.

El sistema aparte de hacer la gestión y el control de los mantenimientos, puede realizar las órdenes de compra del departamento de sistemas y llevar así un control interno del mismo.

Los resultados fueron los esperados ya que el sistema contiene todas las operaciones y funciones que requiere el cliente como la de poder realizar peticiones y revisarlas para dar seguimiento, enviar avisos, consultar, insertar, eliminar y actualizar información de los usuarios y equipos de cómputo así como los reportes de mantenimiento y las órdenes de compra todo esto para poder tener una buena gestión de los mantenimientos y a su vez se realice de manera eficaz.

El sistema que se realizó beneficia a los usuarios, administradores y gerentes de la empresa, así como a las personas que requieran tener una gestión y control de los mantenimientos de cómputo en sus empresas o pequeños negocios, a pesar de que en la Web existen sistemas similares al sistema de mantenimiento, este sistema también puede competir con algunos de ellos ya que es eficaz, flexible y cuenta con interfaces sencillas y amigables.

## **TRABAJOS A FUTURO**

1. Aumentar las funciones y operaciones para hacer una página más extensa.
2. Realizar el análisis de los mantenimientos y hacer graficas sobre estos análisis para poder tener un punto de referencia y saber cómo va evolucionando el estado de los equipos de cómputo.



---

---

## BIBLIOGRAFIA

- [1] Ingeniería de Software. Un enfoque practico. 5ª ed. Roger S. Pressman Mc Graw Hill.
- [2] Ingeniería del Software. Una perspectiva Orientada a Objetos. Eric Braude Alfaomega.
- [3] El lenguaje Unificado de Modelado. Grady Booch, James Rumbaugh, Ivar Jacobson, Prentice Hall.
- [4] Use Cases Requirements in Context. Daryl Kulak, Eamonn Guiney, Addison Wesley
- [5] Procesamiento de Base de Datos. Fundamentos, diseño e instrumentación. Favid M. Kroenke ed. Prentice Hall.
- [6] Introducción a los Sistemas de Bases de Datos 7ª ed. C. J. Date. Prentice Hall.
- [7] Diseño de Bases de Datos Adoración de Miguel, Paloma Martínez, Elena Castro. Alfaomega Ra-Ma.
- [8] Desarrollo Web con PHP, Apache y MySQL. Glass, Le Scouarnec, Naramore, Mailer, Stolz, Gerner. Anaya Multimedia.
- [9] Aprenda. Desarrollo de Bases de Datos WEB Ya. Jim Buyens, McGraw-Hill
- [10] Php 4 a través de ejemplos, Abraham Gutiérrez, Ginés Bravo. Alfaomega, Ra-Ma
- [11] MySQL. Larry Ullman, Prentice may
- [12] MySQL para Windows y Linux. César Pérez, Alfaomega Ra-Ma