



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

**FACULTAD DE CIENCIAS**

Sistema de control y automatización de procesos  
en el área de inventarios

**T E S I S**  
QUE PARA OBTENER EL TÍTULO DE:  
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN  
P R E S E N T A :  
**RUIZ ROJAS ERICK**



FACULTAD DE CIENCIAS  
UNAM

Directora de Tesis: LIC. EN C. C. KARLA RAMIREZ PULIDO

MÉXICO

2006



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**ACT. MAURICIO AGUILAR GONZÁLEZ**  
Jefe de la División de Estudios Profesionales de la  
Facultad de Ciencias  
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

Sistema de control y automatización de procesos en el área de inventarios

realizado por Erick Ruiz Rojas

con número de cuenta 09519683-9 , quien cubrió los créditos de la carrera de: Lic. en

Ciencias de la Computación

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis  
Propietario

Lic. en C.C. Karla Ramírez Pulido

Propietario

M. en C. José de Jesús Galaviz Casas

Propietario

Lic. en C.C. Francisco Lorenzo Solsona Cruz

Suplente

M. en C. Egar Arturo García Cárdenas

Suplente

M. en C. María Guadalupe Elena Ibarquengoitia González

Consejo Departamental de Matemáticas



*Francisco Hernández Quiroz*

Dr. Francisco Hernández Quiroz

FACULTAD DE CIENCIAS  
CONSEJO DEPARTAMENTAL  
DE  
MATEMÁTICAS

Gracias por ayudarme a culminar este trabajo, Max de Mendizábal, Martha Montes de Oca, Verónica Vanegas Lomas y Ruth Briones por parte de la Universidad Pedagógica Nacional. A mi familia y a Brenda de quien le he aprendido a ser mejor persona. En especial agradezco a mi directora de tesis por su apoyo.

# Contenido

<b>Lista de Figuras</b>	<b>vi</b>
<b>Lista de Tablas</b>	<b>ix</b>
<b>I Primera Parte</b>	<b>1</b>
<b>1 Introducción</b>	<b>3</b>
1.1 ¿Por qué usar PHP y MySQL? . . . . .	6
MySQL, PHP y sus logros . . . . .	7
1.2 Solución: Sistema de control y automatización de procesos . . . . .	7
<b>2 Linux</b>	<b>11</b>
2.1 Sistema Operativo . . . . .	11
2.1.1 Linux . . . . .	12
Free Software/Open Source . . . . .	13
GNU Software . . . . .	14
Red Hat™ . . . . .	15
2.2 Internet, Apache y Linux . . . . .	17
2.2.1 Internet . . . . .	17
HTTP . . . . .	17
2.2.2 Apache . . . . .	18
2.2.3 Linux . . . . .	19
Arquitectura Cliente-Servidor . . . . .	19
<b>3 PHP</b>	<b>23</b>
3.1 Lenguajes de Programación . . . . .	23
3.1.1 PHP . . . . .	24
Interacción de página HTML en el web . . . . .	25
Interacción de PHP en Web . . . . .	25

	Interacción de la Base de Datos através del Web . . . . .	20
3.2	Antecedentes . . . . .	21
3.3	PHP . . . . .	21
	Características . . . . .	21
	Zend . . . . .	3
	API . . . . .	3
<b>4</b>	<b>MySQL</b> . . . . .	<b>31</b>
	Conceptos de bases de datos . . . . .	31
	Arquitectura de los sistemas de bases de datos . . . . .	31
	ODBC . . . . .	31
4.1	Visión de los datos . . . . .	31
	4.1.1 Modelo entidad-relación . . . . .	31
	4.1.2 Modelo relacional . . . . .	41
4.2	MySQL . . . . .	41
	Características . . . . .	41
	4.2.1 Arquitectura de MySQL . . . . .	41
	4.2.2 Interfaces Web para MySQL . . . . .	41
<b>5</b>	<b>UML</b> . . . . .	<b>41</b>
5.1	Antecedentes . . . . .	41
5.2	Modelado . . . . .	41
5.3	Arquitectura de software . . . . .	41
5.4	UML . . . . .	51
5.5	Diagramas de UML . . . . .	51
	5.5.1 Diagramas de Caso de Uso . . . . .	51
	Elementos que componen a los casos de uso . . . . .	51
	5.5.2 Diagramas de Actividades . . . . .	51
	Elementos que componen a los diagramas de actividades . . . . .	51
<b>II</b>	<b>Segunda Parte</b> . . . . .	<b>61</b>
<b>6</b>	<b>Sistema Integral de Soporte Técnico</b> . . . . .	<b>61</b>
6.1	¿Qué es el SIST? . . . . .	61
	Objetivos del sistema . . . . .	61
	6.1.1 ¿Por qué es necesario hacer el SIST? . . . . .	61
6.2	Análisis del sistema . . . . .	71
	6.2.1 Diagramas de Caso de Uso . . . . .	71
	Orden de servicio . . . . .	71

	Inventario . . . . .	7
	Préstamo . . . . .	8
6.2.2	Diagramas de Actividades . . . . .	8
	Diagrama de actividades - Ordenes de Servicio . . . . .	8
	Diagrama de actividades - Inventario . . . . .	8
	Diagrama de actividades - Préstamo de equipo . . . . .	8
6.3	Diseño del sistema . . . . .	9
6.3.1	Diseño de la base de datos . . . . .	9
6.3.2	Inventario . . . . .	9
6.4	Implementación . . . . .	10
<b>7</b>	<b>Conclusiones y Trabajo a Futuro</b>	<b>10</b>
	<b>Bibliografía</b>	<b>10</b>



## Lista de Figuras

2.1	Esquema de programas de aplicación y de sistema. . . . .	12
2.2	Estructura gráfica de un sistema Unix. . . . .	16
2.3	Comparativo de Servidores Web, investigación hecha por Netcraft. . .	19
2.4	Arquitectura Cliente-Servidor de 2 capas. . . . .	20
2.5	Arquitectura Cliente-Servidor de 3 capas. . . . .	20
3.1	Esquema de lenguajes de programación. . . . .	24
3.2	Diagrama de interacción de páginas HTML en el Web. . . . .	25
3.3	Diagrama de interacción de aplicaciones en PHP en el web. . . . .	26
3.4	Interacción de la Base de Datos a través del Web. . . . .	27
3.5	Estructura de PHP. . . . .	32
4.1	Esquema de Sistema de Base de Datos. . . . .	36
4.2	Ejemplo de diagrama Entidad-Relación. . . . .	40
4.3	Ejemplo de modelo Relacional. . . . .	41
5.1	Representación gráfica de un caso de uso. . . . .	54
5.2	Representación gráfica de un actor. . . . .	55
5.3	Ejemplo de diagrama de caso de uso. . . . .	56
5.4	Representación gráfica de estados para los diagramas de actividades. .	57
5.5	Representación gráfica de eventos para los diagramas de actividades. .	57
5.6	Ejemplo de diagrama de actividades. . . . .	58
6.1	Organización del SIST. . . . .	65
6.2	Organización departamental de la subdirección de informática. . . . .	68
6.3	Caso de Uso - Actores involucrados en el sistema . . . . .	71
6.4	Caso de Uso - Realizar llamada . . . . .	72
6.5	Caso de Uso - Levantar orden de servicio . . . . .	73
6.6	Caso de Uso - Asignar orden de servicio . . . . .	74
6.7	Caso de Uso - Realizar orden de servicio . . . . .	75

6.8	Caso de Uso - Firmar término de orden . . . . .	70
6.9	Caso de Uso - Cerrar orden de servicio . . . . .	71
6.10	Caso de Uso - Llenar forma de inventario . . . . .	71
6.11	Caso de Uso - Firma de aceptación de captura . . . . .	71
6.12	Caso de Uso - Enviar formato para resguardo físico . . . . .	80
6.13	Caso de Uso - Archivar formatos de inventario . . . . .	81
6.14	Caso de Uso - Solicitar préstamo . . . . .	81
6.15	Caso de Uso - Llenar forma de préstamo . . . . .	81
6.16	Caso de Uso - Firma de préstamo . . . . .	81
6.17	Caso de Uso - Prestar equipo . . . . .	81
6.18	Caso de Uso - Entregar equipo . . . . .	81
6.19	Diagrama de Actividades de orden de servicio. . . . .	81
6.20	Diagrama de Actividades de inventario. . . . .	91
6.21	Diagrama de Actividades de préstamo de equipo. . . . .	91
6.22	Esquema del sistema basado en arquitectura Cliente/Servidor. . . . .	91
6.23	Entidad-Relación del módulo de inventario. . . . .	91
6.24	Modelo Entidad-Relación del Sistema Integral de Soporte Técnico. . . . .	91
6.25	Entidad-Relación del módulo de ordenes de servicio. . . . .	91
6.26	Flujo de acciones del módulo de inventario. . . . .	91
6.27	Inicio del módulo de inventario. . . . .	91
6.28	Verifica la existencia del usuario. . . . .	91
6.29	Se agregan personas que resguardan equipos de cómputo. . . . .	91
6.30	Datos generales del usuario a quien se le cargarán bienes. . . . .	91
6.31	Formato de captura para computadora personal. . . . .	91
6.32	Formato de captura para servidor. . . . .	91
6.33	Formato de captura para computadora portátil. . . . .	101
6.34	Formato de captura para impresora. . . . .	101

# Lista de Tablas

2.1	Respuesta de una petición de documento html por medio de http. . . . .	10
3.1	Código de PHP con conexión a MySQL. . . . .	30
3.2	Salida de código de PHP a HTML. . . . .	30
6.1	Caso de Uso - Realizar Llamada . . . . .	70
6.2	Caso de Uso - Levantar orden de servicio . . . . .	70
6.3	Caso de Uso - Asignar orden de servicio . . . . .	70
6.4	Caso de Uso - Realizar orden de servicio . . . . .	70
6.5	Caso de Uso - Firmar término de orden . . . . .	70
6.6	Caso de Uso - Cerrar orden de servicio . . . . .	70
6.7	Caso de Uso - Llenar forma de inventario . . . . .	70
6.8	Caso de Uso - Firma de aceptación de captura . . . . .	70
6.9	Caso de Uso - Enviar formato para resguardo físico . . . . .	80
6.10	Caso de Uso - Archivar formatos de inventario . . . . .	80
6.11	Caso de Uso - Solicitar préstamo . . . . .	80
6.12	Caso de Uso - Llenar forma de préstamo . . . . .	80
6.13	Caso de Uso - Firma de préstamo . . . . .	80
6.14	Caso de Uso - Prestar equipo . . . . .	80
6.15	Caso de Uso - Entregar equipo . . . . .	80

Parte I

Primera Parte



# Capítulo 1

## Introducción

En la actualidad toda institución privada o pública, en cualquiera que sea su giro tanto académico (escuelas, universidades) como empresarial en sus diversas formas (pequeñas, medianas y macro empresas<sup>1</sup> respectivamente), tiene a su cargo la tarea de llevar un control de bienes<sup>2</sup>; ya que a través de éstos la institución puede clasificar el nivel de producción y desempeño de los mismos.

La clasificación de dichos bienes permite conocer de manera precisa su estado. Por ejemplo, en un almacén de una tienda de autoservicio, se guardan tanto los productos nuevos que acaban de ser comprados, como aquellos que serán reemplazados o cambiados por otros. Éstos últimos pueden ser reemplazados por varias causas:

1. La fecha de caducidad.
2. Algún defecto de fábrica en el producto.
3. No se vendieron en el lapso de tiempo estimado, entre otras causas.

El conocer el estado del producto, permite a los encargados tener cierto control sobre éste, de tal forma que el almacén registre todos los aspectos relacionados con los productos y los procesos que ahí se manejan, para evitar así algún mal uso o desperdicio de los mismos.

---

<sup>1</sup>**Empresa:** Unidad autónoma de control y decisión que al utilizar insumos (objetos o productos para realizar otros) o factores productivos los transforman en bienes, servicios o en otros productos.

<sup>2</sup>**Bien:** Material y artículo en proceso de producción. Existen muchos tipos de bienes en el ámbito contable pero sólo nos ocuparemos de los bienes muebles. Aquellos que pueden trasladarse de un lugar a otro.

Tener este tipo de información actualizada, sin errores o duplicada, requiere de un enorme trabajo humano, por lo que hoy en día se facilita y optimiza dicha tarea con la ayuda de alguna herramienta computacional. Por lo general se requiere de la ayuda de un sistema, el cual permita resolver dicha tarea; de manera que los procesos y productos que se encuentren dentro del almacén sean registrados, de una manera eficiente y segura, facilitando y asegurando el manejo de cada producto. Entre otras cosas un sistema integraría y automatizaría las tareas comunes (por ejemplo, las de soporte técnico), asignaría responsabilidades a cada persona inmersa en cada proceso; desplegaría la cantidad de productos que se encuentran a su disposición así como el estado de los mismos, entre otras cosas. A éste proceso se le conoce como *inventario*.

El término inventario<sup>3</sup>, presenta una gran diversidad de significados dependiendo del giro empresarial al cual se esté aplicando dicho concepto. Para la parte industrial, ésta se constituye de materias primas, artículos en proceso y artículos terminados. Mientras que en una empresa comercial, ésta se encuentra presentada por las mercancías disponibles para la venta.

Los inventarios forman parte de los activos<sup>4</sup> circulantes y representan, en la mayoría de los casos, el activo importante del balance general<sup>5</sup> de una empresa comercial[23]. La importancia de un inventario, es minimizar la inversión, lo cual se logra estableciendo un sistema dinámico de control de los mismos. Además de conocer oportunamente la existencia física del almacén, su correcta rotación de inventarios<sup>6</sup>, determinar los artículos obsoletos y así evitar desperdicios, los cuales afectan las unidades de la empresa. Basado en una existencia mínima o margen de seguridad para cada artículo[1].

Uno de los objetivos más importantes de la contabilidad de inventarios, es determinar la utilidad a través del proceso de enfrentar los costos apropiados para los ingresos. Esto consiste, en determinar qué parte de artículos disponibles para la compra de un periodo deben disminuirse del ingreso proveniente para compras y cual debe incluirse como inventario inicial en el siguiente periodo para ser enfrentado con

---

<sup>3</sup>**Inventario:** Relación y valoración del conjunto de bienes y derechos pertenecientes a una persona física que forma su patrimonio, es decir, partida contable que recoge el nivel de existencias de una empresa de un momento determinado mediante el recuerdo físico de las mismas[1].

<sup>4</sup>Un **activo** es un bien, un servicio o un derecho sobre una empresa, es decir, cualquier cosa a favor del mismo[23].

<sup>5</sup>El **balance general** es el estado que muestra la situación económica de una empresa en una fecha determinada.

<sup>6</sup>Se define como **rotación de inventarios** a la renovación de las existencias de una empresa. Cada tipo de negocio tiene límites de rotación, dependiendo de la mayor o menor dificultad que haya para efectuar las adquisiciones en virtud de la clase de artículos que se manejen[23].

la compra de dicho periodo. De esta manera se puede clasificar a los inventarios en 3 tipos:

**Inventario de Materia Prima:** Aquellos que se adquieren de proveedores y sobre los cuales se efectuarán actos de transformación, consumo o adaptación posterior[1].

**Inventario de Productos en Proceso:** Son materiales y componentes cuya producción se encuentra en máquinas aún sin terminar, es decir, no son aún artículos terminados, sino en producción preventiva, semi-terminada o semi-elaborada[1].

**Inventario de Productos Terminados:** Éstos son el resultado final de una serie de actos de transformación para que estén listos para su venta, es decir, son bienes materiales comprendidos en el activo circulante y destinados a la venta después de haber sido elaborados o transformados[1].

De esta manera, la importancia de llevar un inventario, se vuelve una tarea difícil de controlar, porque debemos conocerlo, actualizarlo y mantenerlo para que sea confiable, útil y explotable. Tener el registro de productos, sin la ayuda de un sistema computacional resulta tedioso, difícil y propenso a fallas, ya que las tareas como relacionar y recordar las ubicaciones de dichos registros, resulta poco confiable. Es muy probable que no se recuerde el lugar o depósito del registro, provocando así que el inventario se encuentre en un estado inconsistente, erróneo y difícil de mantener. De esta manera el contar con un sistema computacional que maneje el inventario hace que el control de los productos sea: viable, eficiente, seguro, persistente, conciso y sin duplicación de datos.

En particular en este trabajo, nos referimos a la parte de inventario de productos terminados en su rubro empresarial, pero el que está formado por activos fijos<sup>7</sup>; específicamente en aquellos relacionados con las computadoras.

La Universidad Pedagógica Nacional contaba con un control de servicios que se tenía en papel. Este método de control era ineficiente y el proceso de búsqueda era tardado. Se analizó la forma de automatizar procesos de control, los cuales incluían control de equipo de cómputo inventariado en la Universidad Pedagógica Nacional Unidad Ajusco, además de solicitudes de mantenimiento técnico de los equipos. Se analizó y se propuso desarrollar un sistema que ayudara a agilizar estos procesos que tenía la universidad.

---

<sup>7</sup>Un **activo fijo** son los bienes que tienen una cierta permanencia, adquiridos con el propósito de usarlos y no venderlos.

## 1.1 ¿Por qué usar PHP y MySQL?

Siempre que se desarrollan páginas Web se deben considerar las siguientes características:

- Hardware<sup>8</sup> del servidor Web.
- Sistema Operativo, es aquel programa que controla todos los recursos de una computadora y determina la base sobre la que pueden escribirse los programas de aplicación.
- Servidor Web (Apache, IIS (Internet Information Service), entre otros).
- Sistema Manejadores de Bases de Datos.
- Lenguaje de Programación.

Los tipos de opciones a escoger dependen de lo que se quiera hacer. No todos los sistemas operativos trabajan correctamente con cierto hardware y no todos los lenguajes de programación pueden hacer conexiones a la mayoría de los sistemas manejadores de bases de datos.

En el caso de PHP se tiene la secuencia de páginas que un usuario visita en un sitio Web, desde que accede al sitio hasta que lo abandona. Varios usuarios pueden estar trabajando en la misma página Web y haciendo actualizaciones a la base de datos, con PHP podemos garantizar que todas las actualizaciones que se realicen en la página Web se harán en la base de datos ya que maneja sesiones.<sup>9</sup>

En éste caso trabajaremos bajo un sistema administrador de bases de datos (MySQL), el cual nos ayudará a agregar, acceder o procesar los datos almacenados. Es un sistema cliente-servidor que consta de un servidor SQL multi-thread<sup>10</sup> que soporta diferentes plataformas.

---

<sup>8</sup>Llamaremos hardware a la parte física de una computadora.

<sup>9</sup>Una sesión comienza cuando un usuario entra en la aplicación Web y termina cuando la abandona. Durante ese espacio de tiempo se pueden definir y manipular una serie de variables pertenecientes a la sesión que permite mantener cualquier tipo de información común (estado) para todas las páginas.

<sup>10</sup>MySQL es multiproceso, esto quiere decir que puede trabajar con varios procesadores si se encuentran disponibles, puede trabajar en distintas plataformas, sistemas de contraseñas y privilegios que hacen que sea más flexible y seguro.

## MySQL, PHP y sus logros

Una de las principales muestras de compatibilidad entre MySQL y PHP son los portales Web, éstos están realizados en MySQL y PHP, donde los portales sirven para la creación de páginas Web de manera dinámica, sin incluir demasiado código. Entre los portales que destacan son PHP-Nuke, Xaraya, Xoops, Drupal y Mambo.

## 1.2 Solución: Sistema de control y automatización de procesos

Para poder realizar un sistema que integre la información de bienes informáticos, es necesario tener las herramientas necesarias. Desde la infraestructura física como los clientes que utilizarán el sistema y el servidor donde estará ejecutándose dicho sistema; hasta el software adecuado que permita el perfecto manejo de la información de los bienes informáticos (inventario) de alguna institución. Es en éste último rubro donde por medio de éste proyecto se pretende incidir de manera positiva en el manejo de los bienes. Esto impactaría y mejoraría el tiempo de respuesta, para obtener la información valiosa acerca de cada uno de los bienes que se tengan.

La finalidad de hacer un inventario a través de un sistema mejoraría las búsquedas y los procesos relacionados con cada bien, la información ahí almacenada se presentaría de manera concisa y sin duplicados, permitiendo así obtener otro tipo de información relacionada con los datos previamente capturados, como la creación de estadísticas y listados de los bienes de una institución así como el ó los nombres de los responsables de cada proceso y así ayudar a la toma de decisiones de cada área involucrada en el proceso.

Se pensó en realizar este sistema a través de Internet debido a que es uno de los medios de comunicación masivo hoy en día y también debido a que cualquier modificación al sistema se vería reflejada de inmediato. Este sistema se hace con la integración de MySQL, PHP y Apache, alianza que ha sido muy comentada y utilizada para realizar este tipo de aplicaciones sobre Web. Esta combinación ha resultado ser eficaz, sencilla y robusta, además al ser "Open Source" esto es "todo el mundo puede verlo, todo el mundo puede probarlo" lo cual hace de esta combinación, una combinación sólida para aplicaciones en Web.

Las modificaciones que se le hagan al sistema se reflejan de inmediato en cuanto se renueva el contenido de la página modificada. Este desarrollo puede ampliarse a

nivel nacional, dado a que puede ampliarse para ser utilizada a todas sus unidades restantes, tanto a nivel local como nacional.

El sistema puede llegar a ser tan útil para la Universidad con el manejo de la información y así reportar sus bienes de manera rápida y clara. También es factible que este proyecto pueda unir a todas sus unidades (que forman parte de la Universidad) teniendo una base de datos más completa sobre los registros de usuarios y bienes de todas las Unidades de la Institución.

Por todas estas cuestiones se ha reforzado la idea de crear un sistema simple en su manejo y adaptabilidad, así como completo y complejo para tener el control de bienes en forma rápida, sencilla y clara. Las propuestas del sistema comenzaron a hacer de éste un sistema al cual se le puede sacar provecho y aumentar las capacidades de automatización simplificando, tareas y acciones que anteriormente llevaba a cabo la institución de manera manual, razón por la cual este sistema beneficia tanto al área de Subdirección de Informática como a la Universidad.

La tesis comprende principalmente dos partes, la primera referente al uso de tecnologías usadas y la segunda concierne al diseño e implementación del sistema.

En los siguientes capítulos de la primera sección se describen las tecnologías que se usarán para desarrollar el sistema en cuestión. Con un breve resumen del sistema operativo, el lenguaje de programación, el lenguaje de modelado usado, el manejador de base de datos, etcétera. Para así tener una visión completa y más amplia de las herramientas utilizadas en este trabajo.

En la segunda sección se da un análisis a profundidad del sistema de inventarios de los bienes informáticos de la Universidad Pedagógica Nacional. Se desarrolló específicamente para dicha institución haciendo uso de las tecnologías antes mencionadas.

El proyecto nace de tener un control de bienes manualmente a tener una alternativa de control automatizado de procesos informáticos. La importancia de tener un control de bienes es necesario para la toma de decisiones sobre que equipos adquirir.

En la Universidad año con año se hace presupuesto anual de compra de bienes informáticos, refacciones, etcétera; este proyecto de cierta forma ayuda en las decisiones de control sobre que equipos o partes tienen más fallas, que refacciones se tiene que

pedir para en caso de siniestro tener un banco de reserva.

Por esta razón es importante llevar un control de bienes, así como un control de problemas comunes sobre equipos de cómputo.

Este proyecto es un sistema de control de bienes informáticos y de control de procesos de soporte técnico.



# Capítulo 2

## Linux

La información contenida en este capítulo esta basada en la referencia “*Manual de referencia Linux*” [21].

Toda computadora de propósito general debe tener un sistema operativo que ejecute otros programas, algunas de las tareas que realiza son reconocer dispositivos de entrada y salida estándar, controlar diversos dispositivos periféricos, entre otros. Debido a esto no todas las computadoras tienen el mismo sistema operativo, existen una gran variedad de ellos: Windows (9x, 2000, XP), MacOS (9.0, X), Unix, Linux (Fedora Core, SuSE, Slackware, Debian), FreeBSD, etcétera.

### 2.1 Sistema Operativo

El software de computadora puede dividirse en:

- Programas del sistema, que controlan la operación de la computadora misma.
- Programas de aplicación, que realizan las tareas que el usuario desea.

El sistema operativo controla todos los recursos de la computadora y establece la base sobre la cual se pueden escribir los programas de aplicación.

Las computadoras (hablando de los componentes que interactúan con el sistema operativo) consisten de procesador, memoria RAM<sup>1</sup>(**R**andom **A**ccess **M**emory), discos, impresoras, interfaces de red y otros dispositivos de entrada/salida. Escribir programas que funcionen en todos estos componentes es difícil. La solución es tener el software encima del hardware de manera independiente, el cual se encargará de administrar todas las partes del sistema y posteriormente presentará al usuario una interfaz que sea fácil de entender y programar.

---

<sup>1</sup>**RAM**: Memoria de Acceso Aleatorio.

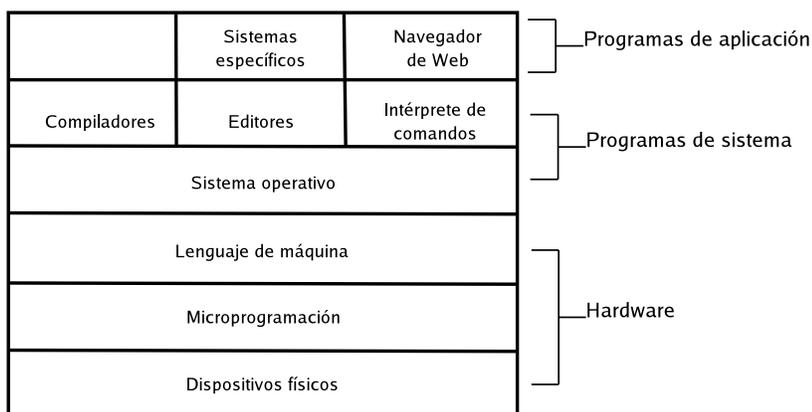


Figura 2.1: Esquema de programas de aplicación y de sistema.

En la Figura 2.1 se muestran las capas en las cuales se compone una computadora, la parte física (hardware) y los programas (software). En la parte inferior está el hardware. La capa más baja contiene a los dispositivos físicos. La capa siguiente es de software primitivo que controla directamente estos dispositivos y ofrece una interfaz más clara para la siguiente capa. Este software llamado **microprograma**, suele estar almacenado en memoria. Sobre esta capa se encuentra el resto del software del sistema. El intérprete de comandos, sistemas de ventanas, compiladores, editores y otros programas similares independientes de la aplicación. Por último se encuentran los programas de aplicación, escritos para resolver problemas particulares, como procesadores de texto, juegos, reproductores de audio, sistemas bancarios, etcétera.

### 2.1.1 Linux

En 1991 Linus Torvalds programó un núcleo<sup>2</sup> compatible con Unix en combinación con GNU acrónimo recursivo para *GNU Not Unix*, en 1992 crea el primer sistema operativo libre completo llamado *GNU/Linux*. Linux es de distribución múltiple, algunas de éstas son comerciales como Caldera Open Linux, pero la mayoría son gratuitos como Debian, Red Hat Linux particularmente solicitada debido a su sencillo proceso de instalación.

Las distintas distribuciones son creadas por grupos organizados, los cuales buscan una personalización del sistema operativo, añadiendo sus propios programas y características que consideren oportunas según su forma de entender la filosofía GNU/Linux.

---

<sup>2</sup>En el sistema operativo el núcleo es también conocido como *kernel*.

## Free Software/Open Source

Las características particulares del Software Open Source es la redistribución a voluntad sin ningún costo, su código fuente está disponible y puede ser modificado por cualquier persona con experiencia en programación. Este concepto es usado principalmente por los sistemas con licencias GPL (**G**eneral **P**ublic **L**icense)[2].

A continuación se describen brevemente los sistemas operativos Red Hat<sup>TM</sup> Linux y Fedora<sup>TM</sup> Core:

**Red Hat Linux:** Sistema operativo multiusuario y multitarea, es decir, que permite a varias personas usar el sistema sin molestarte entre sí, sin preocuparse por lo que otros hacen y a su vez asigna distintos segmentos de memoria virtual a diversos procesos, por lo cual si uno llegara a fallar no afecta al resto de los procesos.

**Fedora Core:** Será de hoy en adelante, la distribución que Red Hat<sup>TM</sup> pondrá a libre disponibilidad del público. No cuenta con soporte alguno por parte de la empresa[3]. El soporte que pudiera brindarse será solo a través de la propia comunidad de usuarios. FEDORA creado por Warren Togami[4] como una comunidad para construir paquetes RPM<sup>3</sup> de alta calidad para Red Hat<sup>TM</sup> Linux.

Linux es el sistema operativo que se utilizará en este proyecto de tesis por sus características más sobresalientes como: tiempo compartido, multiusuario y multitarea. El kernel es el responsable de iniciar, manejar peticiones de memoria, acceso a disco y gestionar las conexiones a red. Linux como se mencionó anteriormente es un sistema operativo de *tiempo compartido*, es decir, simula la ejecución de varios procesos simultáneamente. El término multiprogramación significa que más de un proceso puede estar listo para su ejecución; el sistema operativo primero escoge uno de estos procesos, lo ejecuta durante un cierto tiempo y cuando el proceso necesita esperar por un recurso, el sistema operativo guarda toda la información necesaria para continuar el proceso donde éste se quedó y entonces selecciona otro distinto para ejecutarlo.

Una de las razones del éxito de Linux se debe a su portabilidad. La característica de que todo, excepto pequeñas partes de los compiladores y el kernel, se ejecuten sin ningún cambio en cualquier otra computadora. Linux está escrito en C, por lo que el sistema operativo es portable (es decir, se ejecuta en una extensa variedad de computadoras). El código fuente está disponible y escrito en lenguaje de alto nivel

---

<sup>3</sup>RPM **R**ed **H**at **P**ackage **M**anager. Es software precompilado de fácil instalación y mantenimiento, no necesita compilarse para ser utilizado por el sistema operativo.

(lenguaje C), lo que lo hace que sea fácil de adaptar a exigencias particulares.

Por último, se argumenta que es un *buen* sistema operativo por su ambiente de programación de extraordinaria riqueza y productividad. Aún cuando el sistema operativo en sí introduce varios programas y métodos, su eficacia no se debe a un programa o idea en particular, sino a su enfoque en la programación. La potencia de un sistema depende más de las relaciones entre los programas que de los programas propiamente. Muchos programas hacen de forma aislada tareas triviales, pero al combinarse con otros se convierten en herramientas generales útiles y poderosas.

Existen algunas reglas de distribución de software<sup>4</sup>, esas reglas se basan en las limitaciones sobre código fuente, ejecución y modificación. Estas se definen como:

**Software propietario:** Su copia y distribución es un delito.

**Shareware:** Tras un período de prueba es necesario pagar por el sistema.

**Freeware:** Permiten la copia y distribución del sistema a todo el mundo.

**Software libre:**<sup>5</sup> Permiten la copia, distribución, modificación y posterior redistribución del software modificado.

Linux es parte de lo que se conoce como movimiento *Open Source* (código fuente abierto). El sistema operativo Linux es casi gratuito, pero lo realmente importante es que es abierto. Esto significa que está a disposición de cualquier persona, es decir que pueda leerlo, estudiarlo, modificarlo y sugerir o hacer cambios en él. Linux al ser *open source* todo el mundo puede verlo, puede probarlo y en el caso del kernel y de las aplicaciones de Linux, son miles las personas que hacen esta labor, por lo que el software se va mejorando y se corrigiendo aquellos detalles que fueron pasados por alto en etapas anteriores.

## GNU Software

El proyecto GNU nació en 1984 impulsado por Richard Stallman con la idea de llegar a ser algo parecido a lo que hoy es Linux: un sistema Unix totalmente público y gratuito[27].

Richard Stallman mencionó en su ensayo "*Manifiesto GNU*" que "un núcleo existe,

---

<sup>4</sup>Software: Secuencia de instrucciones o indicaciones destinadas a ser utilizadas en un sistema para realizar una función o tarea para obtener un resultado determinado.

<sup>5</sup>También conocido como Open Source.

pero se necesitan más programas para emular Unix”[43], donde adelantaba el hecho de no tener un núcleo compatible con Unix, además de ser un núcleo que sólo servía para equipos complicados y caros. En 1991, Linus Torvalds comenzó el núcleo de Linux (ver página 12) distribuido bajo la GPL 2.1.1. De ahí en 1992 se combinaron GNU y Linux dando como resultado el sistema operativo formado conocido como “GNU/Linux” [43].

Se habla de GNU/Linux, puesto que eliminando cualquiera de los componentes nos quedamos sin sistema operativo. El proyecto GNU acuñó el término **copy-left** (contrario a copy-right), es decir, que las fuentes son modificables y redistribuibles, con o sin beneficio económico pero en ningún momento, por muchas modificaciones que se realicen, pueden quedar ocultas.

### **Red Hat™**

Fué fundada en 1995 por Bob Young y Marc Ewing. La empresa no ha reportado beneficios, pero su marca ha dejado huella entre los profesionales que la prefieren por su probada fiabilidad y estabilidad. Mezcla una filosofía conservadora con aplicaciones y paquetes propios, por lo que Red Hat™ Linux se convirtió en la distribución Linux dominante en servidores en todo el mundo[2].

El cambio que optó Red Hat™ donde Red Hat Linux estaba orientada hacia usuarios caseros y estaciones de trabajo. Simplemente no producía las ganancias necesarias para mantener dicha distribución. Es por tal motivo, que decidió dar a la comunidad las herramientas y los recursos necesarios para hacerse cargo de una distribución que ocupe el lugar de Red Hat™ Linux. La noticia implica que los usuarios de Red Hat™ Linux deberán considerar un producto alternativo, como es Fedora, que da continuidad a la distribución que ahora Red Hat ha abandonado.

Al convertirse en un entorno cerrado y comercial, es necesario que ofrezca un producto y soporte de calidad. Es por ello que la empresa, comprometida desde el comienzo con la difusión de software libre y su filosofía, han creado el proyecto FEDORA™ en otras palabras, es la versión libre de su sistema operativo. La única diferencia que Red Hat™ probablemente resienta, serán los ciclos de vida del producto, serán más cortos y ahora se llamará Fedora Core, la cual ya no tendrá opción a servicio de soporte comercial.

Existen muchos programas que forman parte de este movimiento y algunos de ellos son los más usados en el mundo:

**Apache:** Un servidor Web que es usado en la mayoría de las sedes de Web de Internet.

**PHP:** Un lenguaje de alto nivel usado para diseñar páginas dinámicas Web.

**MySQL:** Un sistema de base de datos, usado en su mayoría por la relación sencilla que existe entre PHP y MySQL, entre otros.

Una de las cosas más importantes que tiene el software libre, la escribió Eric Raymond en 1997, en un documento titulado *The Cathedral and the Bazaar*, el cual dice: “si el software fuera libre, miles de programadores podrían verlo y al verlo encontrarían anotarían y corregirían cualquier error que presentase”[8].

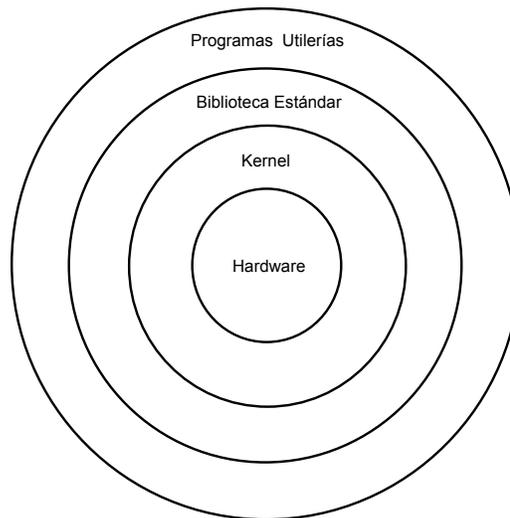


Figura 2.2: Estructura gráfica de un sistema Unix.

Las partes que constituyen la estructura del sistema operativo (como se muestra en la Figura 2.2) son:

**Manejadores de hardware:** Las partes físicas que se utilizan para la comunicación entre el kernel y sus medios.

**Kernel:** Encargado de controlar el hardware, administra la memoria, asigna recursos a todas las actividades que se realizan en la máquina.

**Biblioteca Estándar:** Encargado de crear y manejar los recursos del sistema. Para hacer uso de ellas se hace lo que se conoce como llamadas al sistema.

**Programas y utilerías:** Las aplicaciones que los usuarios requieren para ciertos propósitos.

## 2.2 Internet, Apache y Linux

Como se ha mencionado, Linux ofrece un servidor de páginas Web como herramienta del sistema operativo, a continuación se presentarán las características que presenta Apache en Linux así como su finalidad.

### 2.2.1 Internet

Uno de los protocolos más utilizados en Internet es HTTP<sup>6</sup> (**H**yper**T**ext **T**ransfer **P**rotocol), que es precisamente el protocolo que se emplea en la World Wide Web fundamentalmente para la transferencia de documentos HTML e imágenes entre clientes y servidores Web.

#### HTTP

Para que un paquete de datos viaje desde su origen hasta su destino en una red, es muy importante que todos los dispositivos de dicha red hablen el mismo lenguaje o protocolo. Un protocolo de red es un conjunto de normas las cuales hacen que sea posible y eficiente la comunicación en una red[39].

El protocolo HTTP es un protocolo del nivel de aplicación orientado hacia la comunicación entre sistemas de información distribuidos, colaborativos e hipermedia. El sistema se basa en el envío de mensajes entre un cliente (mediante un navegador Web) y un servidor. Dichos mensajes son la unidad básica de la comunicación HTTP. El cliente es siempre quien inicia la comunicación y para ello envía un mensaje que contiene una *petición*, a lo que el servidor contesta con otro mensaje que contiene la *respuesta*.

1. La petición, se compone de una línea la cual es la solicitud del cliente, subdividida ésta en:
  - Protocolo, por ejemplo http.
  - URL, como www.upn.mx.
  - Solicitud, como GET, POST o PUT.
2. La respuesta, se compone de un encabezado y del cuerpo de la respuesta, como se muestra en la Tabla 2.1.

---

<sup>6</sup>Protocolo de Transferencia de Hipertextos

```

telnet unidad095.upn.mx 80
Trying 200.23.113.55...
Connected to unidad095.upn.mx.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 20 Dec 2004 19:54:45 GMT
Server: Apache
X-Powered-By: PHP/4.3.8
Set-Cookie: lang=spanish; expires=Tue, 20-Dec-2005 19:54:45 GMT
Connection: close
Content-Type: text/html; charset=iso-8859-1

<pagina html>

```

Tabla 2.1: Respuesta de una petición de documento html por medio de http.

## 2.2.2 Apache

Un grupo del CERN<sup>7</sup> desarrolló el concepto cliente-servidor HTTP. En febrero de 1995 en el NCSA Rob McCool desarrolló uno de los más populares servidores Web de aquel entonces. Tomando como base el NCSA httpd 1.3 realizó la primera versión oficial pública de Apache 0.6.2 en abril de 1995. Después lanzó la versión 1.0 en diciembre del mismo año[9].

Apache (**A PAtCHy** server) debe su nombre a numerosos patches<sup>8</sup>, otra definición que le dan a los investigadores del proyecto es en memoria a los Apaches por su gran adaptabilidad al terreno[10]. Apache de acuerdo con un estudio hecho por Netcraft<sup>9</sup> es el servidor Web más popular del momento[11], como se muestra en la Figura 2.3.

Apache puede ser ejecutado en varios sistemas operativos como: Unix, Linux Win32, OS2. Carga sus módulos dinámicamente, tienes transacciones seguras mediante SSL<sup>10</sup>, creación de repositorios virtuales, además de ser de alto desempeño. Su gran rendimiento se logra manteniendo en memoria la compilación del código de

<sup>7</sup>Centro Europeo de Investigación Nuclear.

<sup>8</sup>Patch es una corrección temporal de los problemas o debilidades contenidas en la parte del software.

<sup>9</sup>Netcraft es una compañía de servicios de Internet.

<sup>10</sup>Secure Socket Layer, que es el protocolo de comunicación en Internet de forma encriptada.

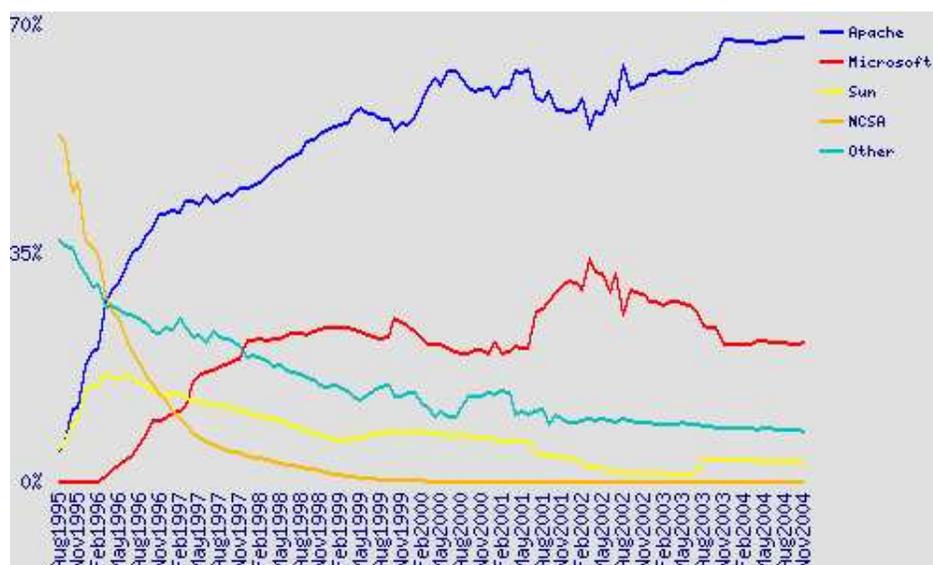


Figura 2.3: Comparativo de Servidores Web, investigación hecha por Netcraft.

bibliotecas que se van cargando, de esta manera llamadas repetidas a funciones ya cargadas provocan llamadas a código precompilado en memoria, por lo que su ejecución es muy rápida. De esta manera es posible servir páginas dinámicas eficientemente.

### 2.2.3 Linux

#### Arquitectura Cliente-Servidor

Algunas de las arquitecturas más usadas que existen en Web para la visualización de páginas Web son de:

**Dos Capas:** Servidores de páginas Web, donde el cliente a través del navegador Web le envía mensajes al servidor por medio del protocolo HTTP, el cual contiene los documentos que se hacen visibles en el navegador.

**Tres Capas:** Acceso a base de datos a través de Web, donde el cliente a través del navegador Web mediante el protocolo HTTP le envía mensajes al servidor el cual contiene los documentos y recoge la consulta. Pide los resultados al servidor de base de datos, el servidor de base de datos le proporciona al cliente los resultados que son enviados por medio del servidor Web al cliente, esto se hace por medio visual en el navegador del cliente.

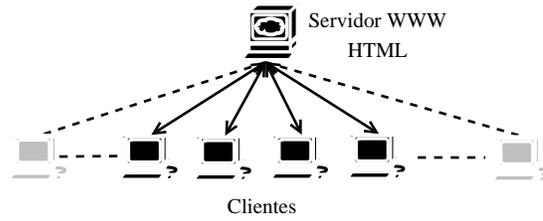


Figura 2.4: Arquitectura Cliente-Servidor de 2 capas.

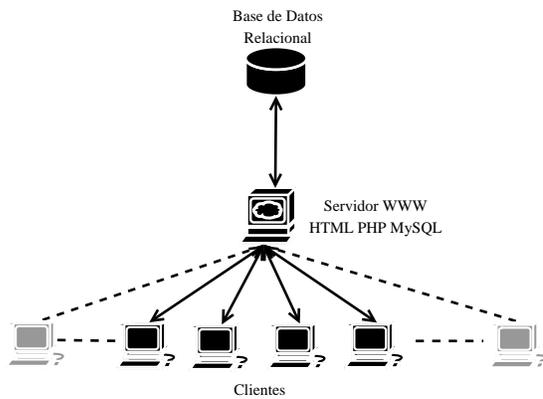


Figura 2.5: Arquitectura Cliente-Servidor de 3 capas.

Retomando los temas presentados en la Figura 3.2 donde el diagrama de HTML en el Web representa las dos capas de la arquitectura cliente-servidor, mostrado en la Figura 2.4; y la interacción de la base de datos a través del Web mostrado en la Figura 3.4 corresponde a la arquitectura cliente-servidor de 3 capas Figura 2.5.

Linux es el sistema operativo que ha tenido un amplio mercado para las instituciones de gobierno. En este capítulo se muestra a Linux como sistema operativo libre bajo el término de *copy-left*, el significado de GNU/Linux, su estructura y las herramientas

que linux tiene, usadas para la elaboración de nuestro proyecto basándose en la arquitectura cliente-servidor. Mostrando la comunicación a Internet mediante Apache que es el servidor Web de Linux.

Linux tiene muchas variantes (Fedora, Red Hat Enterprise, entre otros), se enfocará hacia Fedora el cual es una distribución libre y de instalación sencilla.



## Capítulo 3

# PHP

PHP (HyperText Preprocessor) es un lenguaje de programación de propósito general para hacer páginas Web de manera dinámica. PHP es un lenguaje el cual tiene las cualidades de generar de manera dinámica archivos PDF, imágenes en formato PNG, además de conexión a bases de datos como MySQL, enviar correo mediante el protocolo de SMTP lo cual permite el acceso a SNMP para redes y equipos entre otras cosas. La información contenida en este capítulo esta basada en la referencia [19].

### 3.1 Lenguajes de Programación

Todo sistema tiene que estar implementado en algún lenguaje de programación, este caso no es la excepción por lo que primero hablaremos un poco acerca de éstos.

Los lenguajes de programación se pueden clasificar en lenguajes de alto y bajo nivel, caracterizado por su nivel de abstracción entre el lenguaje y el hardware respectivamente. Podemos decir que un lenguaje de bajo nivel es muy cercano al “lenguaje de máquina<sup>1</sup>” y su código requiere de instrucciones con muchos detalles, un ejemplo de ello es el lenguaje ensamblador. Un lenguaje de alto nivel es el lenguaje natural utilizado para comunicarnos con la computadora, traducidos por compiladores o intérpretes para convertirse en lenguaje de bajo nivel.

Entre los lenguajes de bajo nivel estan los ensambladores y macroensambladores y como lenguajes de alto nivel tenemos a Java, Scheme, Fortran, etcétera.

Las computadoras trabajan con el lenguaje de máquina, por lo que al tener un programa hecho en un lenguaje de alto nivel la computadora lo transforma al lenguaje

---

<sup>1</sup>El lenguaje de máquina está constituida por arreglos de bits. Lenguaje fácil de comprender por la máquina pero no por el usuario.

de máquina pasando por varias etapas[5] como lo muestra la tabla 3.1.



Figura 3.1: Esquema de lenguajes de programación.

Para transformar un programa hecho en un lenguaje de alto nivel al lenguaje máquina se utilizan los procesadores de lenguajes como los intérpretes o traductores. Un intérprete es un programa que toma como entrada un programa escrito en algún lenguaje de programación llamado programa fuente, el cual se va ejecutando conforme va reconociendo el código escrito en él. Un traductor o intérprete es un programa que acepta como entrada un programa escrito en algún lenguaje de programación y va creando una versión del programa en lenguaje objeto<sup>2</sup> que es el lenguaje de máquina, en el cual la computadora va ejecutando en el momento dicho programa.

### 3.1.1 PHP

PHP<sup>3</sup> tiene su origen a principios de 1995. PHP es un lenguaje de programación de alto nivel que nos permite añadir dinamismo a las páginas Web. PHP es un lenguaje interpretado, el mecanismo de interpretación de PHP es repetir el ciclo de traducción y ejecución para cada enunciado del lenguaje fuente, es decir, si un enunciado se ejecuta más de una vez, ese enunciado se traducirá tantas veces como se ejecute. PHP se ejecuta en el servidor, lo que permite acceder a los recursos que este posee, por ejemplo: una base de datos. El intérprete reside y se ejecuta en la computadora donde se encuentra el servidor Web, así el cliente recibe el resultado generado por PHP (archivos HTML).

---

<sup>2</sup>Lenguaje objeto: Es el lenguaje máquina de una computadora, en cuyo caso el programa puede ser ejecutado inmediatamente.

<sup>3</sup>PHP conocido como **P**ersonal **H**ome **P**age, aunque en la actualidad se le conoce también como Hypertext Preprocessor, aunque la segunda definición es la más común de encontrar en los textos de PHP.

## Interacción de página HTML en el web

Las interacciones entre PHP y el Web se definen como los procesos que siguen las aplicaciones Web. Y se puede ver en la Figura 3.2, la interacción con páginas HTML:

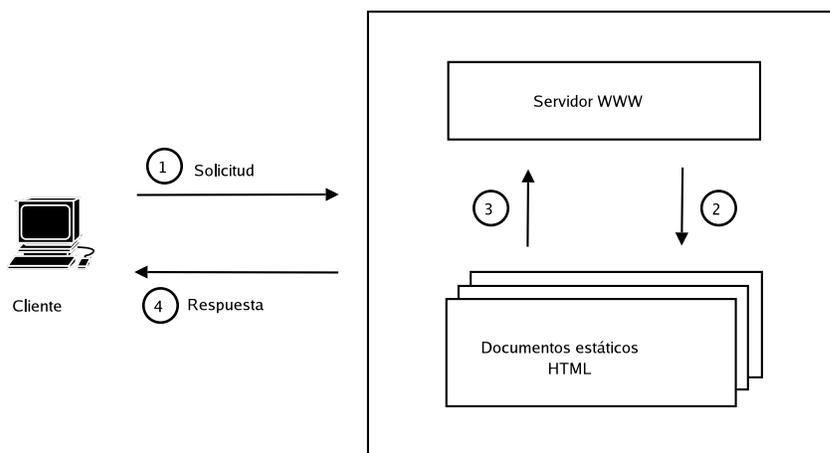


Figura 3.2: Diagrama de interacción de páginas HTML en el Web.

1. Se realiza una petición a un documento HTML, el navegador (*browser*) envía la petición al servidor utilizando el protocolo de transferencia de archivos *hypertext*<sup>4</sup>.
2. La solicitud llega hasta el servidor web correspondiente y localiza el documento solicitado.
3. El servidor lee el documento el cual es enviado al cliente.
4. El documento es recibido por el cliente y se visualiza a través del navegador.

## Interacción de PHP en Web

Esta interacción se diferencia principalmente de la arquitectura de las páginas HTML por el intérprete de PHP que tenemos en nuestro servidor, como nos lo muestra la Figura 3.3, la cual se explica a continuación:

<sup>4</sup>**Hipertexto:** Documentos que presentan la información mediante texto, imágenes, sonido. Además de tener la cualidad de enlazar a otros documentos[6].

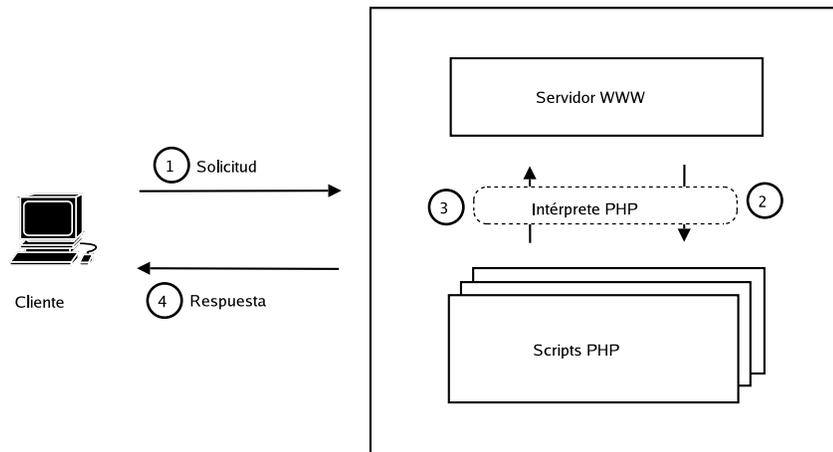


Figura 3.3: Diagrama de interacción de aplicaciones en PHP en el web.

1. Se realiza una petición a un documento PHP, a continuación el navegador envía dicha petición al servidor utilizando el protocolo de transferencia de archivos *hypertext*.
2. El servidor detecta que el archivo contiene código PHP y lo ejecuta el intérprete.
3. El intérprete ejecuta el script<sup>5</sup> solicitado y genera un resultado generalmente en HTML el cual regresará al servidor para que sea procesado y transferido al cliente.
4. El usuario visualiza el documento por medio de un navegador.

### Interacción de la Base de Datos através del Web

La arquitectura que siguen las páginas en PHP con las bases de datos, la podemos observar en la Figura 3.4, la cual sigue los siguientes pasos en su ejecución:

1. Se realiza una petición a un documento en PHP, después el navegador envía la petición al servidor utilizando el protocolo de transferencia de archivos *hypertext*

---

<sup>5</sup>Un **script** consiste de un conjunto de instrucciones para una aplicación, basada en reglas y sintaxis combinando estructuras de control. Éste código puede ser ejecutado por un programa que entienda el lenguaje en el cual ha sido escrito. No necesita ser compilado a código objeto para ser ejecutado. Además de ser un programa que se ejecuta en el servidor web respondiendo a las peticiones de la máquina cliente[7].

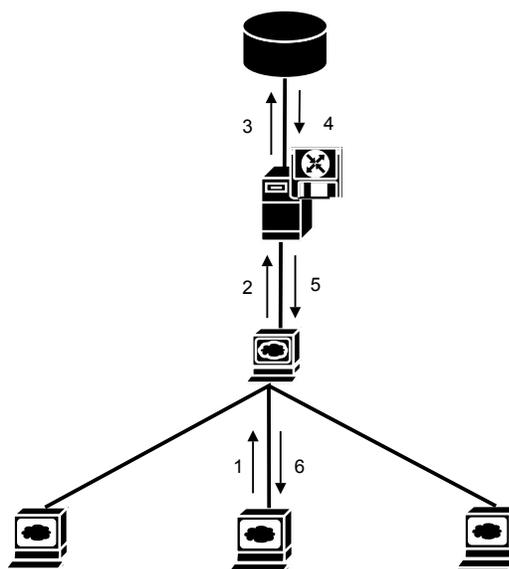


Figura 3.4: Interacción de la Base de Datos a través del Web.

2. Detecta que el archivo contiene código en PHP y lo envía al intérprete.
3. El intérprete ejecuta el script solicitado, el cual se conecta al sistema manejador de bases de datos (MySQL) y ejecutando una instrucción válida para la base de datos. A través de PHP se abre la conexión al servidor de MySQL y envía el resultado obtenido.
4. El servidor de MySQL recibe la consulta de la base de datos, la efectúa y envía el resultado al intérprete de PHP.
5. El intérprete ejecuta el script, lo devuelve al servidor Web como código HTML.
6. El servidor Web envía el documento HTML al navegador del cliente que realizó la petición.

## 3.2 Antecedentes

Rasmus Lerdorf construyó a principios de 1995 una serie de “*Herramientas para páginas Web personales*”<sup>6</sup>. Las cuales consistían de un analizador sintáctico simple.

<sup>6</sup>De ahí su nombre inicial de PHP como **P**ersonal **H**ome **P**ages **T**ools.

además de utilidades comunes en las páginas Web de ese entonces como lo eran: un libro de visitas y un contador, entre otras cosas.

Además del analizador sintáctico, se incluyó otro programa, el cual procesaba datos de formularios al cual se denominó FI (**F**orm **I**nterpreter)<sup>7</sup> y le añadió soporte para MySQL dando lugar a PHP/FI versión 2. *Una vez que se añadieron esas características al lenguaje, mucha gente involucrada en diversos proyectos pudo contribuir al desarrollo de PHP.* De ahí que parte de este lenguaje haya sido desarrollado por una gran comunidad de personas contribuyendo en el código.

En 1997, Andi Gutmans y Zeev Zuraski reescribieron PHP (tercera versión) por completo dando origen a un nuevo nombre, el cual eliminaba la implicación del uso personal limitado que tenía el nombre de PHP/FI, llamándose PHP, con el significado de ser un acrónimo para “*PHP: HyperText Preprocessor*” el cual tenía como característica primordial su **extensibilidad**, como lo es la infraestructura para la mayoría de las bases de datos, protocolos de Internet y API's.

Los manejadores de bases de datos a los que PHP puede acceder son: Adabas D, dbm, dBase, filePro, Hyperwave, Informix, Interbase, LDAP, Microsoft SQL Server, mSQL, MySQL, Oracle, PostgreSQL, Solid y Sybase. Además de tener soporte para el acceso a servidores IMAP (**I**nternet **M**essage **A**ccess **P**rotocol), envío de correo usando SMTP (**S**imple **M**ail **T**ransfer **P**rotocol), acceso a SNMP (**S**imple **N**etwork **M**anagement **P**rotocol) para gestión de redes y equipos, generación dinámica de gráficas PNG (**P**ortable **N**etwork **G**raphics) y de documentos PDF (**P**ortable **D**ocument **F**ormat ), entre otros[12].

Reescribieron el núcleo de PHP, el cual tenía como objetivo de diseño mejorar la ejecución de aplicaciones complejas y mejorar la modularidad del código base de PHP. Dando a su cuarta versión algunas mejoras como en la rapidez con su motor conocido como Zend (de los creadores **Z**eev y **A**ndi), ya que en la actualidad primero pasa una etapa de compilación y después se ejecuta; en sus versiones anteriores el código se ejecutaba mientras se interpretaba. También se mejoró la modularidad de sus componentes, en particular del servidor de Web; a su vez se creó un API más elaborado[13], el cual veremos más adelante en este apartado.

### 3.3 PHP

PHP es un lenguaje de programación que contiene muchos conceptos de C, Perl y Java. Es un lenguaje de programación de propósito general, es decir, que cuenta con variables, sentencias, condicionales, funciones y ciclos<sup>8</sup>.

---

<sup>7</sup>Intérprete de Formularios.

<sup>8</sup>Conocidos como loop (en inglés) o bucles (en español).

Para desarrollar sitios Web (empleando PHP) es necesario disponer de un servidor Web y del intérprete de PHP. PHP es un lenguaje sencillo, de sintaxis cómoda y similar a otros lenguajes, es rápido a pesar de ser interpretado, multiplataforma y dispone de una gran cantidad de bibliotecas, que facilitan el desarrollo de aplicaciones. PHP es modular debido a que cada parte de código de un sistema se puede clasificar en módulos los cuales cada componente hace acoplarse al código. PHP ha sido desarrollado para generar contenidos dinámicamente y uno de sus puntos fuertes, es su gran capacidad para interactuar con una multitud de gestores en bases de datos.

El código PHP está incluido en documentos HTML de manera que es muy fácil incorporar información actualizada en un sitio Web. Para incluir código PHP en una página HTML basta con activar el intérprete con la etiqueta '`<?php`' (como se muestra en la Tabla 3.1) e incluir dentro de la etiqueta las instrucciones correspondientes. Dentro de un mismo documento HTML se puede activar y desactivar el intérprete, tantas veces como sea necesario, construyendo el documento tanto con elementos estáticos HTML como con elementos generados dinámicamente.

El intérprete de PHP reside y se ejecuta dentro de la máquina en la que se encuentra el servidor de Web. De esta forma, el cliente lo único que recibe es el resultado generado (ver Tabla 3.2) por lo que no le afecta si ha sido producido por un programa CGI escrito en lenguaje C, por un script escrito en Perl o bien por una página con código PHP o ASP<sup>9</sup>.

La Tabla 3.1 representa el código de PHP, con incrustaciones de código HTML y solicitud de información a la base de datos MySQL y la Tabla 3.2 presenta la salida de la página con código HTML y resultado de peticiones del programa.

## Características

Un programa de PHP es una combinación de código HTML y código en PHP. El proceso de construcción de un programa es construir un formulario<sup>10</sup> o página dinámica, que contendrá los componentes, controles u objetos en HTML para resolver el problema; además de código o instrucciones en PHP.

Las cuatro grandes características que presenta PHP son:

**Velocidad:** No solo la velocidad de ejecución, sino también no crea demoras en la máquina. Por esta razón no debe requerir demasiados recursos del sistema.

---

<sup>9</sup>Active Server Pages, similar a PHP pero para Microsoft. Esto es, una tecnología del lado del servidor para páginas Web generadas dinámicamente sobre IIS[40].

<sup>10</sup>Parte de una página en HTML conocida como *form*, del que se puede describir como una página de encuesta.

```

<html>
<head>
  <title>Conexion a base de datos a traves de PHP</title>
</head>
<body>
<?php
$conn = mysql_connect("nombreservidor","usuario","contrasena");
if (!$conn) {
  echo "No pudo conectarse a la base de datos";
  exit;
}else{
  echo "Conexion exitosa";
  mysql_select_db("baseddatos");
  $SQL="SELECT nombre, paterno, materno
    FROM inventario_usuario WHERE nombre='Erick'";
  $result = mysql_query($SQL);
  $cn = mysql_result($result, 0, nombre);
  $cp = mysql_result($result, 0, paterno);
  $cm = mysql_result($result, 0, materno);
  print "<br>";
  print "$cn $cp $cm";
}
?>
</body>
</html>

```

Tabla 3.1: Código de PHP con conexión a MySQL.

**Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de errores, pero el hecho de ser un lenguaje conocido por muchos usuarios y programadores hacen de PHP un lenguaje que puede ser ayudado por esa comunidad para reportar, corregir errores además de aportar soluciones o virtudes al lenguaje.

**Seguridad:** PHP provee diferentes niveles de seguridad.

**Simplicidad:** Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente, lo cual facilita su aprendizaje.

```

<html>
<head>
  <title>Conexion a base de datos a traves de PHP</title>
</head>
<body>
  Conexion exitosa<br>ERICK RUIZ ROJAS
</body>
</html>

```

Tabla 3.2: Salida de código de PHP a HTML.

Dentro de las ventajas adicionales que PHP tiene son las siguientes:

- Se ejecuta en varias plataformas utilizando el mismo código fuente. Se utiliza el mismo código base y los scripts pueden ser ejecutados de manera independiente al sistema operativo.
- Expandible, compuesto de un sistema principal (Zend), un conjunto de módulos y una variedad de extensiones de código. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD[14].

Las características que destacan de PHP son su motor Zend, el cual aumenta entre 5 a 10 veces la velocidad de ejecución de páginas PHP. Se argumenta que debido a esto y al gran acoplamiento con Apache supera en rendimiento a IIS (Internet Information Service) con ASP (Active Server Pages) proyecto similar pero hacia plataformas Microsoft[15].

## Zend

Para implementar en Web un intérprete de código, se necesita de:

El **intérprete** que analiza, traslada y ejecuta el código de entrada.

La **funcionalidad** que maneja el lenguaje.

La **interfaz** que permite la comunicación con el servidor Web.

Zend toma la parte de intérprete y un poco de funcionalidad, PHP toma la funcionalidad y la interfaz. Juntos conforman PHP. Además de que contiene todos los módulos que actualmente crean las capacidades de entender al lenguaje. Así PHP se divide en:

**Módulos Externos:** Se llaman mediante un código de ejecución. Los módulos externos tienen la característica de que no requieren recompilarse.

**Módulos de construcción:** Aquellos que PHP compila y los lleva consigo durante el proceso de ejecución, además de tener la ventaja de contar con la disponibilidad de dichos módulos sólo cuando el programa este en ejecución.

**Zend:** En sí Zend es el intérprete de la programación orientada a objetos de PHP[44].

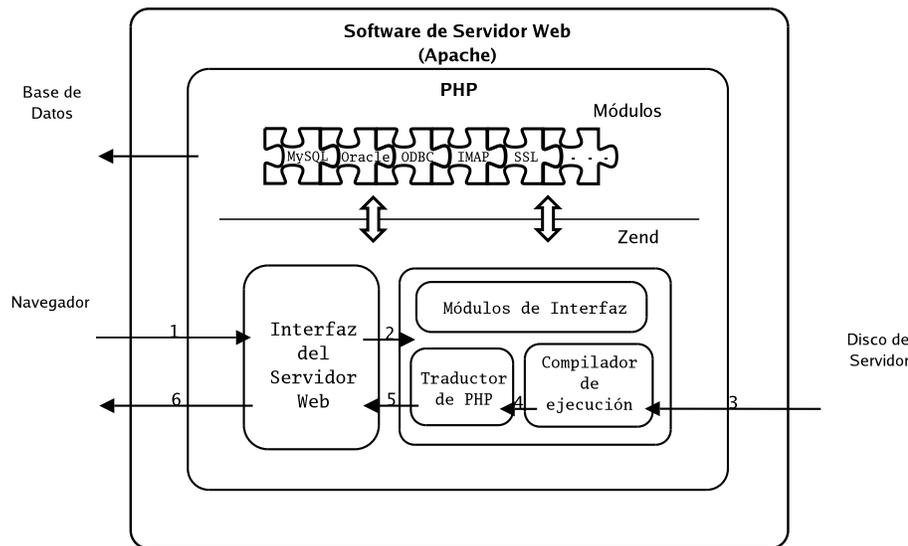


Figura 3.5: Estructura de PHP.

Dentro de la Figura 3.5 se detalla la estructura interna de PHP, con respecto a la Figura 3.4. Retomando la explicación de la Sección 3.1.1, donde:

1. El navegador pasando por Apache (el servidor de Web) le da entrada a PHP desde el momento que se lee la etiqueta '`<?php`'.
2. El código de PHP se interpreta utilizando todo lo que tiene alrededor, como son los módulos para conectividad a bases de datos, protocolos de comunicación, Zend.
3. Realiza las compilaciones necesarias para no volverlas a hacer PHP y guardarlas durante el proceso.

4. Realiza todas las interpretaciones de código correspondientes y utilizando los recursos que PHP le puede proporcionar al código dado. Devuelve los valores interpretados.
5. Devuelve las interpretaciones de PHP limpiando para que aparezca los valores para código HTML.

PHP 3 tuvo una gran cantidad de código basado en módulos de funciones (para la mayoría de los sistemas de bases de datos y protocolos de Internet), Zend dependió de estos módulos de extensión API. Esto significaba que tenían la compatibilidad API expuesta de la nueva ingeniería de scripting si querían que los módulos siguieran trabajando apropiadamente.

Finalmente una de las características en PHP 3 fué la fácil implementación de “ejecuto mientras analizo”, la cual ahora ha cambiado y mejorado por la de “compilo primero y ejecuto después”.

PHP 4 aparte de tener los mismos atributos de PHP 3 es completamente modular, es decir, cada módulo tiene su propia información de configuración de manera separada y permite al usuario construir de manera fácil la construcción de cada extensión como un módulo dinámico. Esto hace PHP sea más cercano a la posibilidad de hacer código (de alto nivel) hasta descender a código binario de distribuciones binarias sobre variantes UNIX, acompañados de extensiones precompiladas dinámicamente (MySQL, IMAP, SNMP, FTP, entre otros).

## API

API (**A**pplication **P**rogramming **I**nterface<sup>11</sup>) es un conjunto de especificaciones de comunicación entre componentes. Además de ser un método para abstraer los niveles software (Ver Figura 3.1). El objetivo principal es tener un conjunto de funciones de uso general. Haciendo de ello uso de su funcionalidad y evitar reprogramar las funciones “*básicas*”.

PHP un lenguaje de programación donde Rasmus Lerdorf unicia el proyecto para crear PHP/FI, después es reescrito por Gutmans y Zuraski. Crece a manera de

---

<sup>11</sup>Interface de Programación de Aplicaciones.

tener una gran extensibilidad en diferentes sistemas manejadores de bases de datos soporte a protocolos (correo y comunicación). PHP es un lenguaje que ha tenido mucha aceptación por ser un lenguaje sencillo y para su fundamental uso que es la generación de páginas dinámicas Web. La participación de PHP sobre nuestro proyecto es sobre Web y es el lenguaje más difundido aceptado y sencillo, es por eso que se optó usar este lenguaje debido a su gran aceptación e inclusive exportabilidad hacia otras plataformas, bases de datos, servidores Web.

# Capítulo 4

## MySQL

Un sistema de base de datos es en esencia una manera de administrar información. Manejar grandes volúmenes de información manualmente resulta difícil, por lo que utilizar un sistema de bases de datos permite hacer esa tarea más sencilla, como organizarla y administrarla.

La información contenida en este capítulo esta basada en la referencia [20].

Las características comunes para utilizar una buena base de datos son:

1. Almacenar datos por mucho tiempo.
2. Grandes volúmenes de datos.
3. Proteger datos contra desastres.
4. Proteger datos contra usuarios no autorizados.
5. Consultar datos actualizados.
6. Permitir acceso simultáneo.
7. Accesibilidad para modificar el esquema.

Una **base de datos** (**DataBase**) se define como una colección de datos relacionados. Donde los **datos** son los hechos que pueden registrarse y tienen un significado. Los **sistemas manejadores de bases de datos** (**DataBase Management System**), se define como el software que facilita el proceso de definición, construcción, manipulación y compartimento de datos a través de una base de datos en diversas aplicaciones, que sustentan con las siguientes facetas:

- Definición: Se especifica el tipo de datos.

- **Construcción:** Se refiere al llenado de las tablas.
- **Manipulación:** Se refiere a la actualización, borrado y consulta de los datos.

Los **sistemas de bases de datos** se constituyen de bases de datos y sistemas manejadores de bases de datos. Ver Figura 4.1.

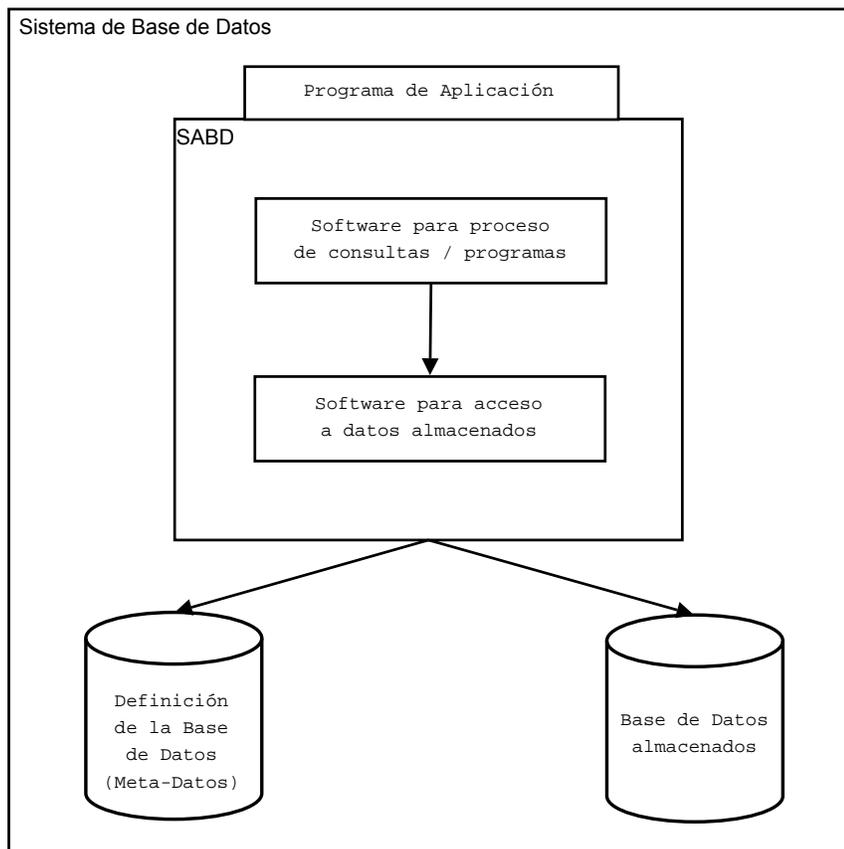


Figura 4.1: Esquema de Sistema de Base de Datos.

### Conceptos de bases de datos

Algunos conceptos comunes de bases de datos, sobre su estructura elemental son:

**Valores:** Cada renglón consiste de un conjunto de valores individuales que corresponden a las columnas. Cada valor debe tener el mismo tipo de dato especificado en la columna.

**Renglón:** Todos los renglones tienen los mismos atributos, debido a su forma tabular.

**Columna:** Cada columna tiene un único nombre y contiene diferentes datos.

**Tablas:** Constituidas por renglones y columnas, una base de datos relacional está hecha de relaciones conocidas como tablas.

### **Arquitectura de los sistemas de bases de datos**

La arquitectura de un sistema de base de datos cambia dependiendo de la forma en que el sistema se ejecuta, debido a la arquitectura de la computadora como la conexión de red, el paralelismo y la distribución.

- La conexión en red de varias computadoras permite que algunas tareas se ejecuten en un servidor y que otras se ejecuten en los clientes. Esta división de trabajo ha conducido al desarrollo de sistemas de bases de datos de tipo cliente-servidor.
- El procesamiento paralelo dentro de una computadora permite acelerar las actividades del sistema de base de datos. Las consultas pueden procesarse de manera que se explote el paralelismo ofrecido por el sistema informático. La necesidad del procesamiento paralelo de consultas ha conducido al desarrollo de los sistemas de bases de datos paralelos.
- La distribución de datos a través de las distintas sedes o departamentos de una organización permite que estos datos residan donde han sido generados o donde son más necesarios, para continuar siendo accesibles desde otros lugares o departamentos diferentes. El hecho de guardar varias copias de la base de datos, en diferentes lugares permite que puedan continuar las operaciones sobre la base de datos, aunque algún sitio se vea afectado por desastres naturales.

La mayoría de las personas que usan un sistema de base de datos no se encuentran (físicamente) junto al sistema de bases de datos, sino que se conectan a él a través de la red. Se pueden diferenciar entonces entre las máquinas cliente, en donde trabajan los usuarios remotos de la base de datos y las máquinas tipo servidor en las que se ejecuta el sistema de base de datos y las aplicaciones de bases de datos se dividen en dos o tres capas.

En la arquitectura de dos capas la aplicación se divide en un componente que reside en la máquina cliente, que manda pedir la funcionalidad del sistema de bases de datos en la máquina servidor mediante instrucciones del lenguaje de consultas y los estándares

de interfaces de aplicación como ODBC se usan para la interacción entre el cliente y el servidor.

En la arquitectura de tres capas, la máquina cliente actúa simplemente como frontal y no contiene una llamada directa a la base de datos. En su lugar, el cliente se comunica con un servidor de aplicaciones, usualmente mediante una interfaz de formularios. El servidor de aplicaciones, a su vez, se comunica con el sistema de bases de datos para acceder a los datos. La aplicación establece las acciones a realizar bajo determinadas condiciones, en lugar de tener aplicaciones distribuidas se podrían tener múltiples clientes. Las aplicaciones de tres capas son más apropiadas tanto para aplicaciones grandes como para aplicaciones que se ejecutan en la red.

## ODBC

ODBC (**O**pen **D**ata**B**ase **C**onnectivity)<sup>1</sup> es una de las normas que existen para la conexión a la base de datos SQL la cual realiza consultas y actualizaciones. Además de ser una interfaz para programas de aplicación para el lenguaje C.

Su manera de funcionar es: el “usuario” o aplicación se conecta a un servidor SQL, estableciendo una sesión SQL. Esta sesión debe tener algún mecanismo de ordenamiento para comprometer el trabajo realizado en la sesión o bien para retrocederlo.

ODBC define una forma para que un programa de aplicación se comunique con un servidor de bases de datos. A esta forma se le conoce como API (**A**pplication **P**rogram **I**nterface)<sup>2</sup> la cual usan las aplicaciones para poder abrir una conexión con una base de datos, enviar consultas, actualizaciones y obtener resultados. Las aplicaciones como las interfaces gráficas de usuario, los paquetes estadísticos y las hojas de cálculo pueden usar el mismo API para conectarse a cualquier servidor de base de datos compatible con ODBC. Para fines de este trabajo se utilizará el API de PHP para la conexión con MySQL, utilizando la interfaz de PHP.

## 4.1 Visión de los datos

Uno de los objetivos principales de un sistema de bases de datos es proporcionar una visión *abstracta* de los datos. Esto se refiere a que el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

Se dice que un sistema es útil si recupera datos de forma eficiente. Esto conlleva al diseño de estructuras de datos por la representación de los datos, en la base de datos

---

<sup>1</sup>Conectividad Abierta de Bases de Datos.

<sup>2</sup>Interfaz para Programas de Aplicación.

De esta manera se simplifica la abstracción de éstos, para aquellos usuarios que no están familiarizados con cómputo y les sea más fácil entenderla de manera gráfica.

Los diferentes niveles de abstracción que simplifican la interacción con los usuarios hacia el sistema son:

**Nivel Físico:** Nivel de abstracción que describe *cómo* se almacenan los datos.

**Nivel Lógico:** Describe *qué* datos se almacenan en la base de datos y *qué* relaciones existen entre éstos.

**Nivel de Vistas:** Describe la parte de la base de datos en vistas, es decir, para una determinada tarea y se requiere de determinadas tablas para satisfacer dicha tarea. El conjunto de vistas crean el nivel de vistas de la base de datos.

El diseño completo de la base de datos se conoce como **esquema** de la base de datos. Para darnos una idea, un esquema corresponde (en lenguajes de programación) a las declaraciones de variables en un programa. Cada variable tiene un valor particular en cierto momento (lo que se conoce como estado). Los sistemas de bases de datos tiene varios esquemas divididos de acuerdo a los niveles de abstracción. El esquema físico describe el diseño físico y el esquema lógico describe el diseño de la base de datos.

El modelo de datos es una colección de herramientas para describir los datos, las relaciones, la semántica y las restricciones de consistencia. El modelo de datos se clasifica en modelo entidad-relación o modelo relacional.

#### 4.1.1 Modelo entidad-relación

El modelo entidad-relación facilita el diseño de bases de datos, representando el significado de los datos haciendo corresponder su significado e interacción mediante un diagrama. El modelo ofrece diferentes decisiones de cómo representar gráficamente el modelo real de datos.

El modelo de datos entidad-relación, representado por sus siglas E-R; consta de una colección de objetos básicos, llamados *entidades* y de *relaciones* entre los objetos. Una **entidad** es una “cosa” u “objeto” distinguible de otros objetos. Las entidades se describen en una base de datos mediante un conjunto de **atributos**. Una **relación** es la asociación entre varias entidades. La estructura lógica general de una base de datos se puede expresar gráficamente mediante un *diagrama Entidad-Relación*, que consta de los siguientes componentes:

**Rectángulos:** Representan conjuntos de entidades.

**Elipses:** Representan atributos de las entidades.

**Rombos:** Representan relaciones entre conjuntos de entidades.

**Líneas:** Representa la unión de atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

En la Figura 4.2, se muestra un ejemplo de diagrama **Entidad-Relación**, donde la entidad *persona* tiene como atributos: *id\_persona*, se define como un número único asociado a la persona; *nombre*, *paterno* y *materno*, se define el nombre completo de la persona; *id\_area*, se asigna un número de una lista de áreas o departamentos de la dependencia; *tel\_ext*, se refiere al número telefónico para encontrar a la persona; *ubicacion*, esta parte es con respecto a la localización del área el cual se encuentra dicha persona. Las siguientes *entidades*: *persona* y *tipo\_persona*, están *relacionadas* con *perfil\_persona*, debido a que una persona puede ser clasificada dependiendo del tipo de persona registrada como: usuario, administrador, entre otros. Lo cual se detallará en el Capítulo 6.

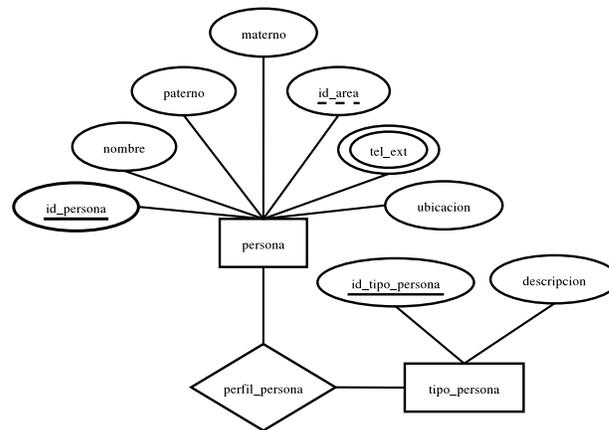


Figura 4.2: Ejemplo de diagrama Entidad-Relación.

### 4.1.2 Modelo relacional

El modelo relacional puede crear esquemas que pueda llevar (mediante un cálculo) la información correcta o duplicada ayudando a distinguir entre buenos y malos diseños de esquemas respectivamente. Los diseños de bases de datos se realizan en el modelo Entidad-Relación y después se traducen al modelo relacional[35].

En el modelo relacional se utiliza un grupo de tablas para presentar los datos y las relaciones entre ellos. El modelo relacional es un ejemplo de un modelo basado en registros. Esto es debido a que la base de datos se estructura en registros de formato fijo para varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo define un número fijo de campos o atributos. Las columnas de las tablas corresponden a los atributos del tipo de registro. En la Figura 4.3, se muestra la representación gráfica que se presentará a lo largo de esta tesis con el modelo relacional.

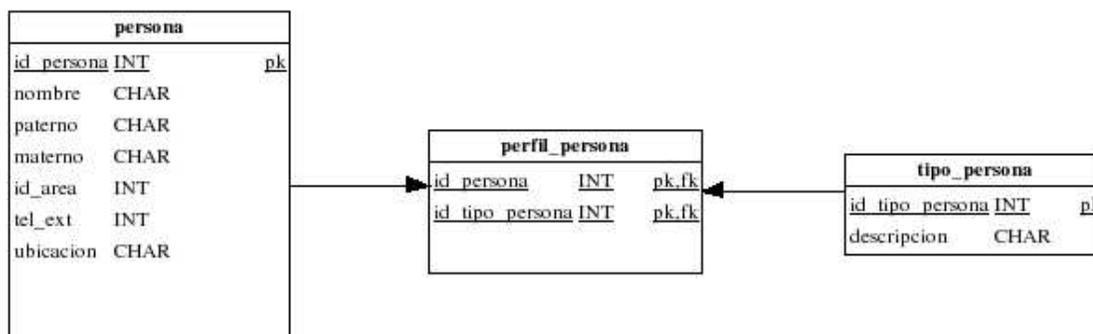


Figura 4.3: Ejemplo de modelo Relacional.

## 4.2 MySQL

Los orígenes de MySQL se remontan a 1979, cuando era utilizada la herramienta que hasta ese momento era la base de datos UNIREG la cual fue creada por Michael “Monthy” Widerius. En 1994, TcX una empresa sueca, comenzó a buscar un servidor SQL para utilizarlo en aplicaciones Web, Widerius se encontró con mSQL y creó una interfaz<sup>3</sup> de programación similar a la de mSQL, para así disponer de varias herramientas que eran gratuitas y que se podrían adaptar a lo que hoy se conoce como MySQL. La derivación de las siglas no es clara, se dice que MySQL se divide en dos partes, la primera que es el prefijo **My**, proviene del nombre de la hija del co-fundador de MySQL Monty Wideniu’s, y el sufijo SQL por **Structured Query Language**[42].

MySQL no es un producto Open Source pero está apegado a los términos de GNU/GPL (GNU **G**eneral **P**ublic **L**icence).

Para escribir programas de acceso a la base de datos de MySQL, que viene junto con diversos programas de utilidad, es necesario:

<sup>3</sup>Interfaz: Forma de comunicación entre el usuario y la máquina.

- Personalizar la manipulación de la entrada de datos, esto es, proporcionar métodos de entrada de datos para el usuario a través del programa adecuado ya que será más fácil de usar e intuitivo. Hacer invisible las instrucciones que se requieren para la entrada de instrucciones al servidor de MySQL mediante programas.
- Personalizar la salida, lo cual da un mejor entendimiento al informe obtenido de consultas. Algunos de los elementos especializados que incluiría el informe son: encabezados personalizados, supresión de valores repetidos, formatos numéricos y cálculos numéricos.

MySQL cuenta con las siguientes herramientas:

- Un servidor de SQL, el cual proporciona acceso a sus bases de datos.
- Programas “cliente” para poder acceder al servidor, compuesto de un programa interactivo para introducir consultas de manera directa al servidor y visualizar los resultados; además de programas administrativos y de utilidad para importar o exportar datos, y comprobar los permisos de acceso, entre otras cosas.

MySQL se clasifica como un sistema manejador de bases de datos relacional RDBMS (**R**elational **D**atabase **M**anagement **S**ystem), es decir:

- La base de datos *DB* (**D**ata**B**ase) es el depósito de la información a almacenar, estructurado de manera sencilla, ya que:
  1. El conjunto de datos de una base de datos está organizado en tablas.
  2. Cada tabla está organizada en filas y columnas.
  3. Cada fila de una tabla es un registro.
  4. Los registros pueden contener diversas partes de información, cada columna de una tabla corresponde a una de esas partes.
- El sistema de administración **MS** (**M**anager **S**ystem), es el software que le permite usar sus datos dejándole insertar, recuperar, modificar o borrar registros.
- La palabra **relacional** significa que sigue el modelo relacional capaz de extraer datos de tablas y de unir información para producir respuestas a preguntas que no pueden responderse sólo de tablas individuales.

La característica de un sistema manejador de base de datos relacional<sup>4</sup>, reside en su capacidad para extraer datos de tablas y de unir la información de cada una, para producir respuestas de preguntas que no pueden responder sólo desde tablas individuales.

### Características

MySQL tiene las siguientes características:

**Velocidad:** Se ha comprobado que MySQL es la base de datos más rápida.[27]

**Costo:** La licencia de MySQL es gratuita.

**Fácil de utilizar:** Es un sistema manejador de bases de datos sencillo, fácil de configurar y administrar en sistemas grandes.

**Capacidad:** Pueden estar conectados al servidor muchos clientes simultáneamente y llevar a cabo actualizaciones, consultas y visualización de la información de la base de datos.

**Compatibilidad:** Existen diferentes interfaces de programación para lenguajes como: C, Perl, Java, PHP, Python entre otros.

**Capacidad de gestión de lenguajes de consultas:** MySQL está comprendido en SQL (Structured Query Language) que es el lenguaje elegido por todos los sistemas de bases de datos actuales.

**Conectividad:** Desde cualquier lugar del mundo por medio de la Internet se puede acceder a las bases de datos y el trabajo en red.

**Multiplataforma:** MySQL se ejecuta en varios sistemas operativos tales como: variantes de UNIX, Windows y OS/2.

#### 4.2.1 Arquitectura de MySQL

MySQL opera usando una arquitectura cliente-servidor, es decir:

- El servidor de la base de datos es un programa situado en la computadora en la que se almacenan sus datos. Escucha las peticiones del cliente que entra en la red y accede a los contenidos de la base de datos según dichas solicitudes, para dar a los clientes la información que se solicita.

---

<sup>4</sup>RDBMS: Relational Database Manager System.

- Los clientes son programas que se conectan con el servidor de la base de datos y emiten consultas para indicar la información que requieren.

Esta arquitectura cliente-servidor de MySQL tiene las siguientes ventajas:

1. El servidor proporciona control de concurrencia para que dos usuarios no puedan modificar el mismo registro a la vez.
2. Todas las solicitudes del cliente se procesan a través del servidor, de tal forma que éste clasifica a los usuarios, sus actividades y privilegios.
3. No tiene que estar conectado a la computadora en la que está situada la base de datos.
4. MySQL está fuertemente acoplado a la Internet, así que cualquiera puede ejecutar un programa del cliente desde donde se encuentre y este conectado al servidor de red.

### 4.2.2 Interfaces Web para MySQL

La red se ha vuelto importante para las bases de datos debido a que los navegadores de Web ofrecen un lenguaje universal, para poder ver y capturar la información en cualquier parte del mundo. El sistema puede ejecutarse en cualquier sistema informático y no hace falta que el usuario descargue software específico para que tenga acceso a la información. Con el crecimiento de los servicios de información, las bases de datos empleadas para los servicios de procesamiento de operaciones resulta conveniente utilizar la interfaz de formularios HTML.

El uso de interfaces entre MySQL y el Web se ve reflejado en la presentación exclusiva de documentos fijos en un sitio Web, lo cual presenta limitaciones como la carencia de adaptabilidad para el usuario. Por ejemplo:

- Cuando se tienen artículos de noticias, es conveniente resaltar la noticia de mayor interés al usuario.
- Actualización de datos, no se hace con la misma rapidez que con documentos dinámicos, ya que en estos su actualización no resulta de manera simultánea. Y esto se complica si se trabajan varias páginas Web con una gran cantidad de datos a actualizar.

Por otro lado, mediante la generación dinámica de documentos en el Web a partir de una base de datos, cuando se solicita un documento se ejecuta un programa en el sitio

del servidor, que a su vez ejecuta consultas en una base de datos y genera el documento solicitado de acuerdo con los resultados de las consultas. HTML (**H**yper**T**ext **M**arkup **L**anguage) permite que el texto reciba formatos de manera ordenada y que se destaque la información importante. Es por ello que las interfaces Web resultan eficaces y atractivas visualmente.

Se presenta conceptos de bases de datos, la importancia de una base de datos (reside en la manera de administrar la información). Se presenta una forma de visualizar datos, el modelado de datos se hace a través del diagrama entidad-relación y el modelo relacional crea esquema de cálculo para realizar un buen diseño de base de datos. MySQL creado por Widerius en 1994, presenta sus características, su operación mediante la arquitectura cliente-servidor, permite la conexión en red de varias computadoras haciendo que algunas tareas se ejecuten en un servidor y que otras se ejecuten en los clientes.



# Capítulo 5

## UML

El lenguaje UML (**U**nified **M**odeling **L**anguage) es un lenguaje estándar que se utiliza para escribir planos de software. También se utiliza como método de desarrollo de software. UML es un lenguaje que sirve principalmente para visualizar, especificar, construir y documentar los planos del sistema para proyectos de software. La información contenida en este capítulo esta basada en la referencia “*El lenguaje unificado de modelado UML*” [17].

### 5.1 Antecedentes

Los lenguajes de modelado orientados a objetos aparecieron entre los años setenta y ochenta, cuando surgían nuevas y más complejas metodologías cuyos enfoques eran el análisis y el diseño. Las metodologías orientadas a objetos se incrementaron a más de 50 entre 1989 y 1994. El problema a resolver era el encontrar un lenguaje de modelado que cubriera las necesidades para modelar software.

Dentro de las metodologías destacadas se encuentran, el método de Booch, el OOSE<sup>1</sup> (**O**bject **O**riented **S**oftware **E**ngineering) de Jacobson y el OMT<sup>2</sup> (**O**bject **M**odeling **T**echnique) de Rumbaugh. Otros métodos importantes fueron Fusion, Shlaer-Mellor, Coad-Yourdon y cada metodología tenía sus ventajas y desventajas.

El método de Booch era expresivo en la fase de diseño y construcción de los proyectos; el de Jacobson (OOSE), da una buena descripción para casos de uso en requisitos, análisis y diseño de alto nivel y el método de Rumbaugh es útil para el análisis y sistemas de información de gran cantidad de datos[17].

---

<sup>1</sup>OOSE: Ingeniería de Software Orientado a Objetos.

<sup>2</sup>OMT: Técnica de Modelado de Objetos.

Debido a su alto potencial los tres métodos (Booch, OOSE, OMT), comenzaron a unificarse para establecer un estándar de modelado ofreciendo estabilidad, teniendo como objetivos los siguientes:

- Modelar sistemas desde su concepto hasta el programa ejecutable, utilizando técnicas orientadas a objetos.
- Satisfacer las necesidades de los sistemas más complejos.
- Un lenguaje de modelado utilizable tanto por personas como por máquinas.

Así nació UML en octubre de 1994 a partir de la unión entre Rumbaugh y Booch teniendo como objetivo unificar sus métodos (Booch y OMT), iniciando con el nombre de Método Unificado en 1995. Jacobson se incorporó después con OOSE para que en 1996 naciera UML.

## 5.2 Modelado

Con un modelo se pueden expresar mejor los procesos de un sistema desde una visión gráfica. A éstos modelos se les llaman *planos del sistema*, ya que nos muestran de forma gráfica cómo se está pensando y diseñando dicho sistema. Con éstos modelos se pueden involucrar planos detallados así como planos más generales, que ofrecen una visión global del sistema en consideración. Un buen modelo incluye aquellos elementos que tienen una gran influencia y omite aquellos elementos irrelevantes para el nivel de abstracción dado. A grandes rasgos se construyen modelos para una mejor comprensión del sistema que se va a desarrollar.

*“Un modelo es una simplificación de la realidad.”[17]*

A través del modelado se reduce la complejidad de comprender un sistema, centrándose en un aspecto a la vez. Éste es el enfoque que planteó Edsger Dijkstra: “divide y vencerás”, para resolver un problema difícil dividiéndolo en sub-problemas más pequeños que se pueden resolver de manera más sencilla. Modelar adecuadamente puede permitir al diseñador trabajar a mayores niveles de abstracción[17].

El modelado se define como:

- Una parte fundamental en la producción de software. Se construyen modelos para comunicar la estructura deseada o el comportamiento del sistema. Además de visualizar y controlar la arquitectura del mismo.

- Una técnica de ingeniería probada y aceptada. Como ejemplo de ello se tienen los modelos arquitectónicos de casa y rascacielos que ayudan a visualizar el producto final.
- Una comprensión del sistema. Y en cuanto más grande y más complejo es el sistema, el modelado se hace más importante, esto se debe a que es necesario construir modelos de sistemas complejos ya que es difícil comprender el sistema en su totalidad.

Los sistemas pueden ser descritos desde diferentes perspectivas utilizando diferentes modelos, y cada modelo es una abstracción.

Al modelar un sistema, los objetivos que se deben cubrir son los siguientes:

- Ayudan a visualizar qué es lo que se espera de un sistema.
- Permiten especificar la estructura o el comportamiento del sistema.
- Proporcionan plantillas que guían en la construcción del sistema.
- Documentan las decisiones que se han adoptado.

### 5.3 Arquitectura de software

La visualización, especificación, construcción y documentación de un sistema de grandes cantidades de software, requiere que el sistema sea visto desde varias perspectivas.

La arquitectura es el conjunto de decisiones significativas sobre los siguientes elementos:

- La organización de un sistema.
- La selección de elementos estructurales y las interfaces que constituyen el sistema.
- El comportamiento entre los elementos.
- La composición de elementos estructurales y de comportamiento de subsistemas progresivamente más grandes.

La arquitectura de software no sólo sirve para la estructura y el comportamiento del sistema, sino también para el uso, la funcionalidad, el rendimiento, la capacidad de adaptación y el poder reutilizar el código fuente del sistema, ya sea para desarrollar un nuevo sistema o mejorar el existente.

## 5.4 UML

El lenguaje UML debe servir para:

**Visualizar:** Al escribir modelos en UML, se facilita la comunicación de un modelo explícito. Algunas cosas se modelan mejor textualmente y otras de forma gráfica. UML es un lenguaje gráfico, por lo que un desarrollador puede escribir un modelo en UML y otro desarrollador, que conozca el lenguaje de modelado, en otro lugar puede interpretar ese mismo modelo sin ambigüedad.

**Especificar:** Construyendo modelos precisos, no ambiguos y completos. UML cubre la especificación de todas las decisiones de análisis, diseño e implementación que deben realizarse al desarrollar y desplegar un sistema.

**Construir:** Aunque no es un lenguaje de programación de propósito general, sus modelos pueden conectarse de forma directa a una gran variedad de lenguajes de programación. Esto significa que se puede establecer correspondencia entre un modelo de UML a un lenguaje de programación (como PHP o incluso a tablas en una base de datos relacional como MySQL).

**Documentar:** La arquitectura de un sistema y todos sus detalles además de proporcionar un lenguaje para expresar requisitos, pruebas, actividades de planificación de proyectos y gestión de versiones.

UML tiene cuatro tipos de elementos:

**Elementos estructurales:** Son partes estáticas de un modelo y representan conceptos o cosas materiales; por ejemplo, podría hablarse en una *ventana* como un objeto o la parte estática del modelo y el *colocar ventana en la pared* como la acción ejecutada. Donde se puede decir que ventana es un concepto y el colocar ventana es un elemento de cosas materiales.

**Elementos de comportamiento:** Representan comportamientos ligados a elementos estructurales.

**Elementos de agrupación:** Son composiciones de modelos o paquetes del sistema.

**Elementos de anotación:** Son comentarios aplicados para describir, clarificar y hacer observaciones sobre un elemento de un modelo, podría decirse que un elemento de anotación es una *nota*.

Un lenguaje de modelado es un lenguaje cuyo vocabulario y reglas se centran en la representación de conceptos y formas de un sistema.

## 5.5 Diagramas de UML

Un diagrama es una representación gráfica de un conjunto de elementos, visualizado como un gráfico de nodos llamados *elementos* y arcos llamados *relaciones*. Los diagramas se dibujan para visualizar un sistema desde diferentes perspectivas, ya que un diagrama es una proyección del sistema. El diagrama representa una gráfica resumida de los elementos que constituyen el sistema.

Los diagramas se utilizan para visualizar sistemas desde diferentes perspectivas, no sólo se limita a la industria de la construcción. Aparecen en cualquier ingeniería que implique la construcción de sistemas complejos como en la ingeniería civil, aeronáutica, naval, de manufacturación e ingeniería de software.

Los diagramas de UML se utilizan de dos formas: para especificar modelos a partir de los cuales construir un sistema ejecutable, conocido como *ingeniería directa* y reconstruir modelos a partir de partes de un sistema ejecutable conocido como *ingeniería inversa*. Al igual que un arquitecto, se tendrán que construir los diagramas incrementalmente.

Un sistema es una colección de subsistemas organizados para lograr un propósito descrito por un conjunto de modelos desde diferentes puntos de vista. Un subsistema es un grupo de elementos que constituyen, una especificación del comportamiento ofrecido por otros. Un modelo es una abstracción semánticamente cerrada de un sistema, es decir representa la simplificación completa y autoconsistente de la realidad.

A continuación se listan los tipos de diagramas con que cuenta UML:

**Diagramas estructurales:** Muestran los aspectos estáticos de un sistema (la arquitectura del sistema), es decir, la existencia y ubicación de clases, interfaces, colaboraciones y las relaciones entre ellas.

Este tipo de diagramas se subdividen en:

**Diagramas de clases:** Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema. Los diagramas de clases son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Los diagramas de clases contienen: clases, interfaces, colaboraciones y relaciones de dependencia, generalización y asociación.

**Diagramas de objetos:** Los diagramas de objetos modelan las instancias de los elementos contenidos en los diagramas de clases. Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. Se utilizan para modelar la vista de diseño estática o la vista de procesos estática de un sistema. Esto conlleva al modelado de un instante del sistema en un momento concreto y la representación de un conjunto de objetos, su estado y sus relaciones.

**Diagramas de componentes:** Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Esto implica modelar las cosas físicas que residen en nodo, tales como ejecutables, bibliotecas, tablas, archivos y documentos. Los diagramas de componentes son diagramas de clases que se centran en los componentes de un sistema.

**Diagramas de despliegue:** Un diagrama de despliegue muestra la configuración de nodos que participan en la ejecución de los componentes. Los diagramas de despliegue son diagramas de clases que se ocupan de modelar los nodos del sistema. Los diagramas de despliegue son importantes para visualizar, especificar y documentar en particular sistemas cliente/servidor.

**Diagramas de comportamiento:** Muestran los aspectos dinámicos de un sistema, por ejemplo el flujo de mensajes a lo largo del tiempo y el movimiento físico de componentes en una red.

Este tipo de diagramas se subdividen en:

**Diagramas de casos de uso:** Los diagramas de casos de uso son importantes para modelar el comportamiento de un sistema, un subsistema o una clase. Muestra un conjunto de casos de uso, actores y sus relaciones. Modelan el contexto del sistema o el modelado de los requisitos de comportamiento de esos elementos. Estos diagramas facilitan que los sistemas, subsistemas y clases sean abordables y comprensibles, al presentar una vista externa de cómo pueden utilizarse estos elementos en un contexto dado.

**Diagramas de interacción:** Un diagrama de interacción muestra una interacción que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se puedan enviar entre ellos. Un diagrama de secuencia es un diagrama de interacción, donde destaca la ordenación temporal de los mensajes; un diagrama de colaboración es un diagrama de interacción, donde destaca la organización estructural de los objetos que envían y reciben mensajes. La mayoría de las veces esto implica modelar instancias concretas de clases, interfaces, componentes y nodos, junto con los

mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Los diagramas de interacción documentan la dinámica particular de objetos o se pueden utilizar para modelar un flujo de control particular de un caso de uso.

**Diagrama de estados:** Un diagrama de actividades es un caso especial de diagrama de estados en el cual todos o la mayoría de los estados son estados de actividad y todas o la mayoría de las transacciones se disparan por la terminación de las actividades en el estado origen. Así, tanto los diagramas de actividades como los diagramas de estados son útiles para modelar la vida de un objeto. Sin embargo, mientras un diagrama de actividades muestra el flujo de control entre actividades, un diagrama de estados muestra el flujo de control entre estados.

**Diagrama de actividades:** Un diagrama de actividades es fundamentalmente ya que muestra el flujo de control entre actividades. Con un diagrama de actividades también se puede modelar el flujo de un objeto conforme pasa de un estado a otro en diferentes puntos del flujo de control. Mientras que en los diagramas de interacción destaca el flujo de control entre actividades.

### 5.5.1 Diagramas de Caso de Uso

Los diagramas de casos de uso se emplean para modelar la vista de los casos de uso de un sistema. Un diagrama de casos de uso se puede representar como un conjunto de casos de uso, en la Figura 5.3 se muestra un conjunto de los casos de usos dados. En las Figuras 6.4 hasta la 6.9 se muestran, los actores y sus relaciones.

Organiza los comportamientos del sistema, se utilizan para describir la vista de casos de uso estático de un sistema. Son importantes para organizar y modelar el comportamiento del sistema. En UML se puede modelar el contexto de un sistema con un diagrama de casos de uso, destacando los actores en torno al sistema.

Para modelar el contexto de un sistema debemos seguir los siguientes pasos:

- Identificar los actores en torno al sistema, considerando qué grupos requieren ayuda del sistema para llevar a cabo sus tareas, qué grupos son necesarios para ejecutar las funciones del sistema, qué grupos interactúan con el hardware externo o con otros sistemas de software y qué grupos realizan funciones secundarias de administración y mantenimiento.
- Organizar los actores similares en jerarquías de generalización o especialización.

- Proporcionar estereotipos para cada uno de esos actores, para ayudar a entender los roles de éstos en el sistema
- Introducir dichos actores en un diagrama de casos de uso y especificar las vías de comunicación de cada actor con los casos de uso del sistema.

Cuando se crean diagramas de casos de uso en UML, cada diagrama de casos de uso es una representación gráfica de la vista estática, de los casos de uso de un sistema. Esto significa que un único diagrama de casos de uso no necesita capturar toda la vista (de casos de uso estática) de un sistema. En conjunto, todos los diagramas de casos de uso representan la vista completa de un sistema, individualmente cada uno representa sólo un aspecto.

Los diagramas de casos de uso son importantes para visualizar, especificar y documentar el comportamiento de un elemento.

### Elementos que componen a los casos de uso

Cuando se dibuje un diagrama de casos de uso en UML, se deben mostrar sólo aquellos casos que son importantes para comprender el comportamiento del sistema o parte del sistema en su contexto, además de mostrar sólo aquellos actores relacionados al caso de uso.

**Nombre:** Es una cadena de texto que hace distinguir un caso de uso de otro.

**Caso de uso:** Es una descripción de un conjunto de secuencias de acciones que ejecuta un sistema para producir un resultado observable, el que adquiere un sentido para el actor en cuestión. Gráficamente un caso de uso se representa como una elipse como se muestra en la siguiente figura:



Figura 5.1: Representación gráfica de un caso de uso.

**Actor:** Se define como “algo” o “alguien” que puede interactuar con el sistema que se está desarrollando. Se representa como se muestra en la Figura 5.2 y puede ser

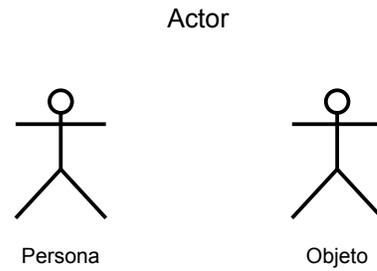


Figura 5.2: Representación gráfica de un actor.

representada tanto para las personas involucradas, tanto con el proceso como con los objetos.

Un ejemplo de una representación gráfica de un diagrama de caso de uso, se muestra en la Figura 5.3. Donde en el lado izquierdo se encuentran los actores, es decir las personas que estarán involucradas en el sistema, mientras que en la parte central están los casos de uso, los cuales se implementarán como acciones externas e internas del sistema. Al referirnos a acciones externas son aquellas que no están directamente involucradas en el sistema, sino a una precondition para poder realizar acciones internas.

### 5.5.2 Diagramas de Actividades

Un diagrama de actividades es un diagrama de flujo, que muestra el flujo de control entre actividades en un sistema. Una actividad muestra un conjunto de actividades el flujo secuencial o ramificado de actividades y los objetos que actúan para ilustrar la vista dinámica de un sistema. Los diagramas modelan los pasos de un proceso computacional. Además de modelar el flujo de control de una operación.

Una actividad es una ejecución en curso, dentro de una máquina de estados. Las actividades producen alguna acción, lo cual provoca un cambio en el estado del sistema o alguna devolución de un valor. Las acciones incluyen llamadas a operaciones, envíos de señales o cálculos, como la evaluación de una expresión. Gráficamente un diagrama de actividades es una colección de nodos y arcos.

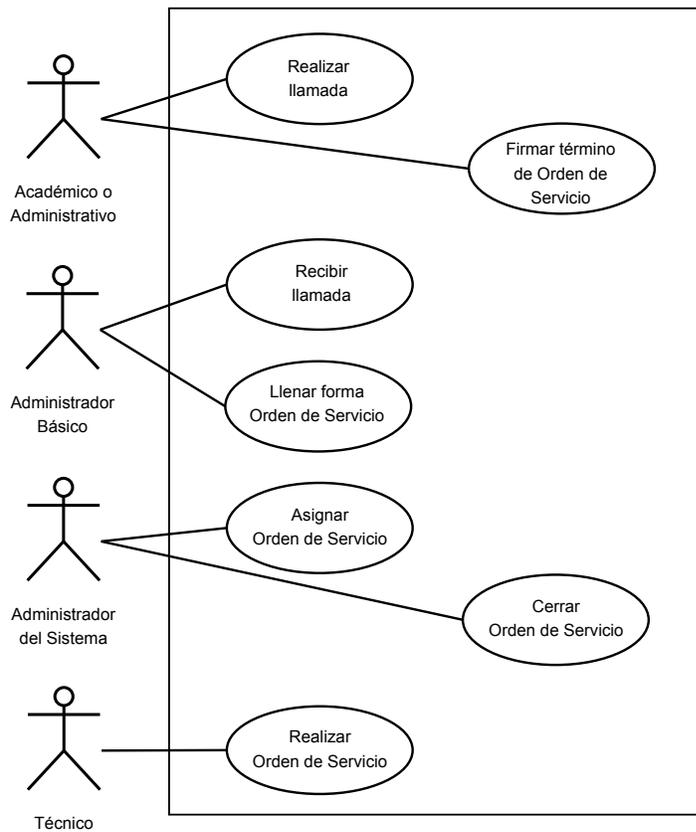


Figura 5.3: Ejemplo de diagrama de caso de uso.

### Elementos que componen a los diagramas de actividades

**Máquina de estados:** Es un comportamiento que especifica, las secuencias de estados por los que pasa un objeto a lo largo de su vida en respuesta a ciertos eventos.

**Estado:** Es una condición o situación en la vida de un objeto, durante la cual si satisface alguna condición, realiza alguna actividad o espera algún otro evento.

**Evento:** Es la especificación de un acontecimiento significativo que ocupa un lugar.

Cuando se modelan los aspectos dinámicos de un sistema, normalmente se utilizan los diagramas de actividades de dos formas:

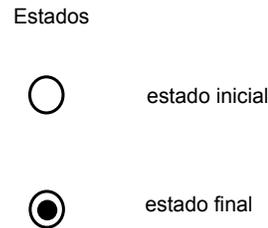


Figura 5.4: Representación gráfica de estados para los diagramas de actividades.



Figura 5.5: Representación gráfica de eventos para los diagramas de actividades.

1. Para modelar un flujo de trabajo.
2. Para modelar una operación.

Un diagrama de actividades:

- Se ocupa de modelar un aspecto de la dinámica de un sistema.
- Contiene los elementos esenciales para comprenderlo.
- Proporciona detalles de forma consistente con su nivel de abstracción.

Un ejemplo de una representación gráfica de un diagrama de actividades, se muestra en la Figura 5.6. En el cual a grandes rasgos se observa el estado de inicio de una tarea, pasando por diferentes eventos y por diferentes personas, hasta que se concluye la tarea en el estado final. Se inicia primero el diagrama donde el **Cliente** realiza el evento de **Realizar llamada**, el proceso sigue cuando un **Administrador simple** recibe el evento de **Recibir llamada**, continuando él mismo con **Llenar forma de orden de servicio**, pasando este evento a otra persona **Administrador del Sistema** realizando el evento de **Asignar orden de servicio**, el proceso pasa a otra persona que es el **Técnico** haciendo el evento de **Realizar orden de servicio**.

Este proceso regresa al Cliente quien Firma de término de servicio, y el Administrador del Sistema concluye el proceso de Cerrar orden de servicio. Para obtener más información al respecto, se puede referir a los Apéndices de este trabajo.

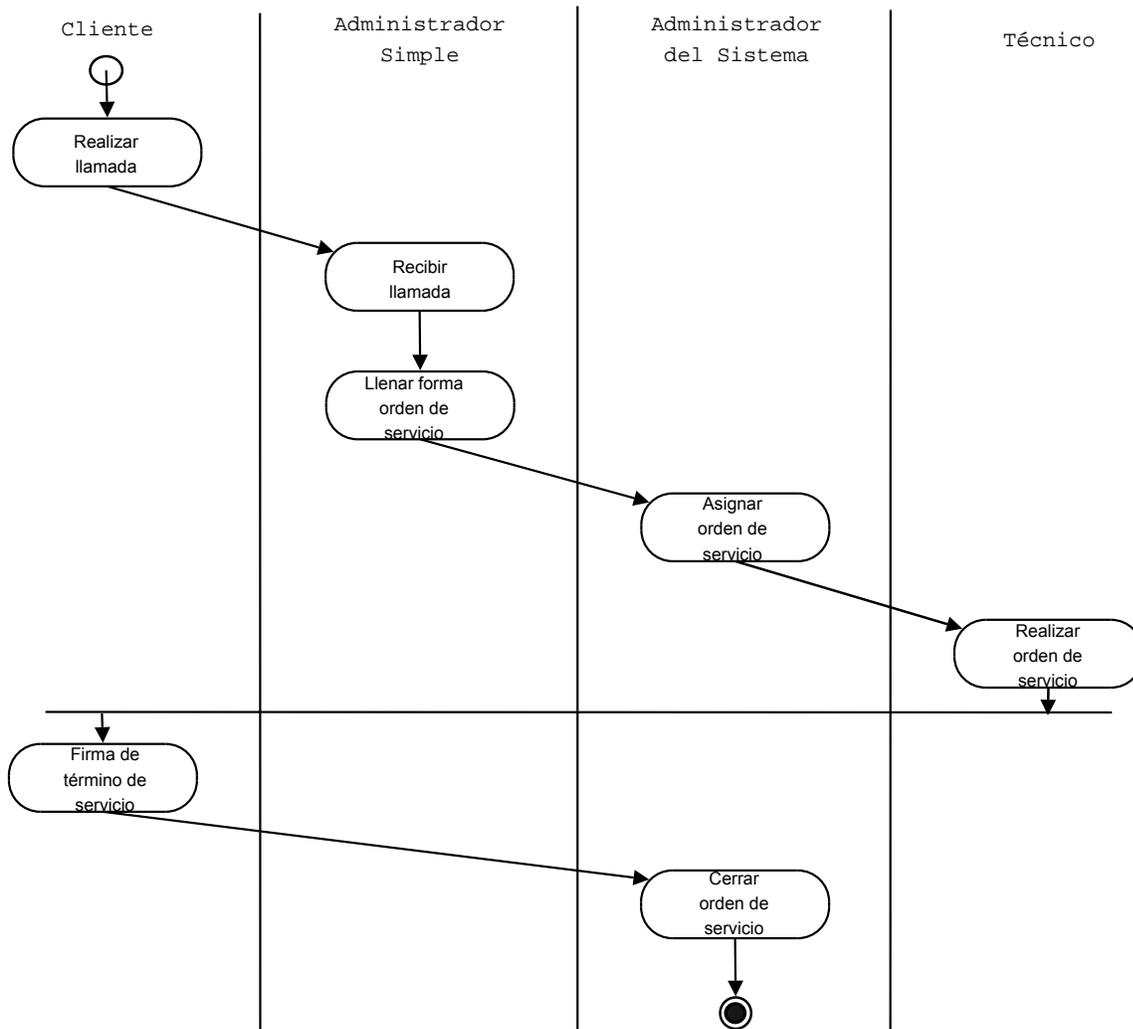


Figura 5.6: Ejemplo de diagrama de actividades.

UML es un lenguaje para visualizar, especificar, construir y documentar los planos del sistema. Se presentan definiciones de los diagramas de modelado pero para este proyecto se utilizan diagramas de comportamientos como los diagramas de casos de uso (modela el comportamiento de un sistema) y los diagramas de actividades (modela el flujo de control entre actividades). Estos diagramas se presentan en la Sección 6.2.1 y en la Sección 6.2.2 sobre el sistema.



Parte II  
Segunda Parte



## Capítulo 6

# Sistema Integral de Soporte Técnico

Recientemente todo el sector público y privado, hablando también de escuelas desde primarias, secundarias, preparatorias, universidades, incluso pequeña, mediana y macro empresa<sup>1</sup>, tienen todos algo en común. Deben tener un control de sus bienes<sup>2</sup>, con el objetivo de tener un mejor desempeño y producción, en cuanto a control de bienes se refiere. dependiendo de lo que ajuste para su finalidad u objetivo. Para ello este proyecto propone un sistema que conlleve los registros almacenados de los bienes informáticos, con la finalidad de obtener un sistema con un ambiente sencillo y útil para poder ingresar los datos, así como búsquedas predeterminadas sobre grupos departamentales o por alguna característica del equipo.

El objetivo principal del sistema, es automatizar las tareas comunes de la subdirección de informática, las cuales están dirigidas a una parte del personal de dicha área. Una de las tareas es por ejemplo, el registro de los bienes computacionales de manera automatizada. El proceso que se realizaba anteriormente era que un grupo de personas del área de Informática, llenaban en una hoja con pluma o lápiz, los datos que pedía el formato sobre el bien computacional. Al término del llenado, el usuario responsable del equipo firmaba el documento, esto significaba el sentido de que dicho documento avalaba el resguardo del equipo sobre el firmante. Este proceso se puede simplificar

---

<sup>1</sup>**Empresa:** Unidad autónoma de control y decisión que al utilizar insumos (objetos o productos para realizar otros) o factores productivos transforma en bienes y servicios o en otros productos que adquiere dicha unidad.

<sup>2</sup>**Bien:** Material y artículo en proceso de producción. Existen muchos tipos de bienes en el ámbito contable pero sólo nos ocuparemos de los bienes muebles, es decir, aquellos que pueden trasladarse de un lugar a otro.

haciendo el llenado a través de un sistema que presentará el formato de los bienes haciendo el proceso más rápido y claro. Otra tarea que el sistema podrá automatizar y mejorar es en el tiempo de consulta para conocer el estado del equipo computacional siendo éste más corto y confiable.

## 6.1 ¿Qué es el SIST?

El SIST (Sistema Integral de Soporte Técnico) es un sistema en el cual inicialmente se captura la información relacionada con los bienes informáticos, para posteriormente notificar las solicitudes de aquellas personas que requieran asistencia técnica en mantenimiento o reparación del equipo de cómputo en cuestión, perteneciente a la institución. Este sistema entonces se define como un sistema de automatización de oficina, ya que el propósito principal de éste es usar la información, manejarla, analizarla y explotarla, agregando bienes, órdenes de servicio, entre otras cosas.

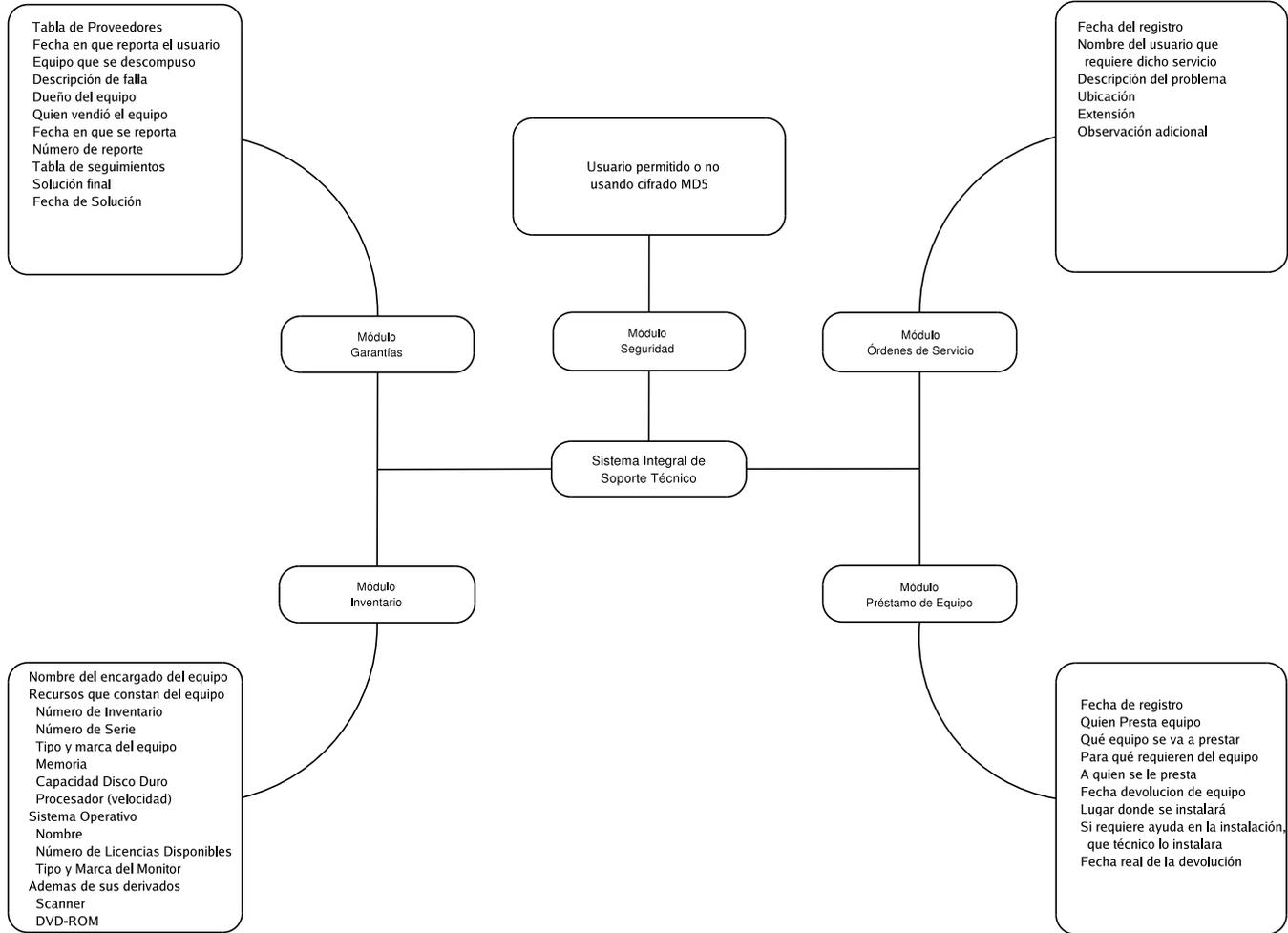
Una de las ventajas de tener dicha información, es que se encuentra disponible a través de uno de los medios masivos de propagación más utilizados en la última década, Internet. Aunque el manejo de información generada por computadora difiere del manejo de información producida manualmente, se dice que existe una mayor cantidad de información por computadora a administrar, en el que el sistema se apegará a los objetivos que se manejan.

### Objetivos del sistema

- El sistema deberá ser capaz de “informar” a través de correo electrónico a los técnicos encargados sobre las “órdenes” designadas por el administrador de soporte técnico.
- El sistema realizará una gráfica de desempeño de cada uno de los técnicos en cuanto a las órdenes realizadas.
- El sistema será capaz de informar mediante tablas, el contenido de inventario de bienes informáticos haciendo una búsqueda predeterminada a grupos departamentales o por alguna característica del grupo.

El sistema apoyará además en las tareas relacionadas con el almacenamiento de registros (o documentos) que facilitarán tanto las operaciones de conteo como la búsqueda de los equipos, además de permitir una fácil inserción y realización de dichos documentos para respaldar la captura y entrega del mismo. Esto se logra teniendo un sistema administrador de base de datos como MySQL (ver Capítulo 4) y una manera

Figura 6.1: Organización del SIST.



de visualizar la información de forma sencilla mediante un programa, ejecutándose sobre Internet (ver Capítulos 2 y 3).

Los objetivos requieren de un manejo de “módulos” (ver Figura 6.1), los cuales ayudarán a separar las tareas de automatización de procesos para la subdirección de Informática. Las tareas comunes a automatizar, son:

**Seguridad:** Su tarea permitir el acceso al sistema por medio de encriptación MD5<sup>3</sup>, que es el usado en MySQL para encriptación de claves.

**Inventario:** Son formas en las cuales se registran ciertos campos necesarios para llevar a cabo el control de bienes informáticos. Estas formas se detallarán en el diseño del sistema.

La institución contempla equipos de cómputo, específicamente computadoras personales de escritorio, impresoras, servidores y computadoras portátiles. Dentro del inventario de bienes está catalogado marcas y modelos de los bienes adquiridos para tener un control con lo que cuenta la institución.

**Ordenes de Servicio:** El objetivo es reportar al administrador, si existió algún tipo de complicación con el equipo. Para ello, el proceso inicia desde que un usuario llama a la persona encargada de informar los errores o fallas que presenta su computadora. Esta información es capturada en el sistema con los siguientes datos: nombre y apellido del usuario, además de la ubicación del equipo, el cual debe estar previamente dado de alta en la base de datos de usuarios activos del sistema.

Quienes se encargan de generar el reporte agregan al usuario en la base de datos en caso de que no este anexado a ella. Después se brinda una breve descripción del problema y de ahí se levanta la orden de servicio.

**Almacén:** Esta parte consiste del material que se tiene en almacén (stock<sup>4</sup>), así como la acción de asignar y quitar elementos dados en resguardo.

Por ejemplo, si un usuario tiene problemas con su computadora lo que tiene que hacer es lo siguiente; levantar una orden de servicio con el administrador básico; después el administrador avanzado asigna dicha orden al personal autorizado. El personal capacitado, consultando la falla del equipo determina que hace falta (reemplazar el monitor, por ejemplo); el área de almacén asigna un vale el cual

---

<sup>3</sup>MD5 acrónimo de *Message-Digest Algorithm 5*, que es un algoritmo de reducción criptográfico Diseñado por Ronald Rivest en 1991.

<sup>4</sup>Stock: es la cantidad de bienes con los cuales cuenta actualmente el almacén. Stock se refiere al número de refacciones almacenados.

reporta el elemento que salió, junto con la firma de quien va a tener a su cargo dicha refacción. Sin embargo, puede ser que el problema no era el que estaba previamente registrado sino que el problema fuera de la tarjeta de video (basándonos en nuestro ejemplo). Para ello, el almacén deberá cancelar el equipo que salió anteriormente, para levantar otro vale distinto por una tarjeta de video éste se hace válido cuando se obtiene la firma del usuario quien resguarda el equipo.

**Préstamo de Equipo:** El sistema apoyará a la generación de vales de los equipos que presta la subdirección de Informática. Informática da servicio de préstamo de equipos computacionales a maestros y alumnos de la universidad, este módulo genera una forma de captura de los equipos prestados para su control.

**Garantías:** Se muestra un calendario de fecha de vencimiento de garantía del equipo. En caso de que un equipo sufra alguna falla y en caso de estar dentro del rango de garantía, este módulo presenta si el equipo tiene o no garantía.

Estas tareas son las que conforman al sistema, el cual será operado por un grupo de personas organizadas de tal forma que el tipo de operación del sistema se restringa dependiendo del operador. El nombre que toman las personas que operarán el sistema se les llamará como *Administrador*.

Cada uno de estos administradores se diferencia por ser:

**Administrador Básico:** es aquella persona que sólo se dedica a levantar órdenes de servicio.

**Administrador Intermedio:** hace las operaciones de administrador básico, además tiene las funciones de agregar usuarios, agregar equipos de cómputo, hardware y software.

**Administrador Avanzado:** hace las operaciones del administrador intermedio y tiene la autoridad para asignar órdenes de servicio.

**Administrador Único:** él se encarga de operar y administrar el control del sistema siendo éste, quien se encarga del código, además se encarga de realizar las siguientes operaciones borrar usuarios, equipo de cómputo tanto hardware como software y sobre el control del sistema de almacén etc.

### 6.1.1 ¿Por qué es necesario hacer el SIST?

Para tener un control de bienes es necesario saber qué información es apropiada a escribir o almacenar. El SIST facilitará tanto la búsqueda de equipo almacenado así

como los respaldos seguros de la misma.

Para entender la importancia que tendrá el sistema se detallará la organización de la subdirección de informática a la cual se destina el proyecto.

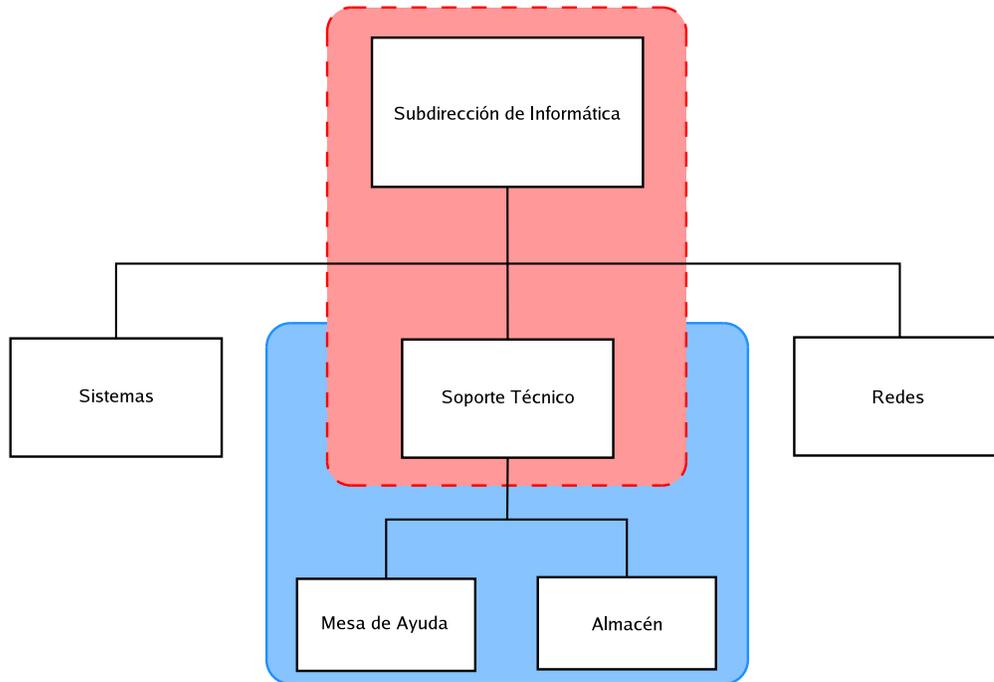


Figura 6.2: Organización departamental de la subdirección de informática.

El sistema apoyará de manera visible a los departamentos de “*Soporte Técnico*”, “*Mesa de Ayuda*” y “*Almacén*”. A su vez ayudará en la toma de decisiones a las áreas de “*Soporte Técnico*” y la “*Subdirección*” (Figura 6.2). A continuación se detallan las actividades que realizan cada uno de los departamentos:

**Sistemas:** Área cuyas tareas están basadas en (como su nombre indica) sistemas de control escolar. Nos referimos a los procesos de la parte administrativa relacionados con los alumnos, como anexar a la base de datos existente admisión de alumnos, actualización de matrícula, becas, etcétera; en sí todas las partes que componen la administración escolar.

**Soporte Técnico:** El área se desempeña en las actividades de reparación de equipos de cómputo. En particular se define (en ésta tesis) al sistema de cómputo como

una computadora personal (PC<sup>5</sup>), computadora portátil (laptop) o servidor. Debido a que es una institución educativa, ésta se limita a ofrecer las herramientas que cada departamento requiere.

Para ello, el área de soporte técnico requiere de un almacén con el fin de tener almacenado equipo para cambio, reparación o préstamo. Los accesorios con que cuenta el almacén son extensos pero para ello se listan los de mayor reemplazo: monitor, disco duro, lector de discos compactos o quemador de los mismos, memoria RAM, tarjetas de red o tarjeta inalámbrica. Otros accesorios comunes para préstamo son: monitor, computadora personal y proyector.

Accesorios como el teclado o el ratón, son reemplazados en caso de haber alguna falla.

**Redes:** Este departamento se encarga de la administración de servidores, así como la ampliación del enlace de transmisión, es decir, en cuanto a telecomunicaciones se refiere.

Por ejemplo, el tipo de transmisión que debe tener una videoconferencia o el incremento de ancho de banda, para abastecer la demanda de servicios en Internet en toda la universidad.

**Almacén:** En este subdepartamento se almacenan los distintos accesorios con los que se puede trabajar en el departamento de “*Soporte Técnico*”. Almacén, cuenta también con accesorios que requieren otros departamentos, como son: cables de red, pilas de interruptores de energía (no-break), herramienta para soporte técnico (desarmador, taladro, martillo, taquetes, etcétera) para el departamento de redes y los accesorios para el departamento de sistemas<sup>6</sup>.

Estos son los departamentos, que conforma la subdirección. De ahí, el sistema que se realizará deberá apoyar a los departamentos de “*Soporte Técnico*” y de “*Almacén*”. El sistema será capaz de mantener un contador, en lo que se refiere a los equipos de cómputo y sus derivados.

Finalmente, la importancia de llevar un inventario de manera manual se vuelve una tarea difícil de controlar. Tener el registro de productos, sin la ayuda de un sistema computacional resulta tedioso y propenso a fallas; ya que las tareas como, relacionar y recordar las ubicaciones de dichos registros, resulta poco confiable, provocando así que el inventario se encuentre en un estado inconsistente, erróneo y difícil de mantener.

---

<sup>5</sup>PC se entiende como computadora personal (del inglés **P**ersonal **C**omputer) ó computadora de escritorio se caracterizan por ser equipos que no requieren de accesorios caros para poder usarlos.

<sup>6</sup>Se refiere al software que la dependencia académica adquiere, como parte de accesorios para reparación o adquisición.

De esta manera el contar con el SIST hará que el control de los productos sea: viable, eficiente, seguro, persistente, conciso y sin duplicación de datos.

## 6.2 Análisis del sistema

La función del análisis será brindar soporte a las actividades de un negocio. Para conseguir este objetivo, un sistema basado en computadoras hace uso de seis elementos fundamentales[36]:

1. Software: Son programas de computadora, con estructuras de datos y su documentación que hacen efectiva la logica del negocio o controles de requerimientos del programa.
2. Hardware, dispositivos electrónicos y electromecánicos: Que brindan capacidad de cálculos y funciones rápidas, exactas y efectivas (computadoras, sensores, maquinarias, bombas, lectores, etc.), que proporcionan una función externa dentro de los sistemas.
3. Personal: Son los operadores o usuarios directos de las herramientas del sistema.
4. Base de Datos: Una gran colección de información organizada y enlazada al sistema.
5. Documentación: Manuales, formularios y otro tipo de información descriptiva que detalla o da instrucciones sobre el empleo y operación del programa.
6. Procedimientos: También conocidos como los pasos que definen el uso específico de cada uno de los elementos o componentes del sistema y las reglas para su uso y mantenimiento.

### 6.2.1 Diagramas de Caso de Uso

Los diagramas de casos de uso como hemos mencionado anteriormente en el capítulo de UML, presentamos los diagramas que componen al sistema.

El diagrama de caso de uso general, de las personas involucradas en el sistema (Figura 6.3), es el personal del área de informática que actúa directamente sobre el sistema, mientras que los usuarios (académicos y administrativos) actúan indirectamente sobre el sistema, a través de un administrador. La descripción de los usuarios del sistema se hizo en la sección 6.1.

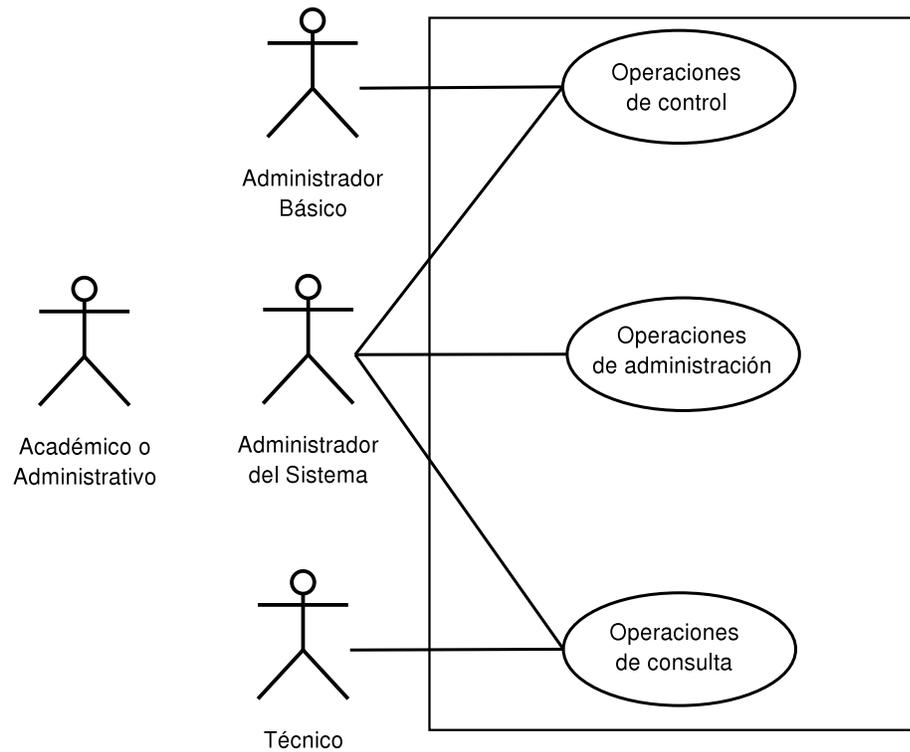


Figura 6.3: Caso de Uso - Actores involucrados en el sistema

Comenzaremos presentando los diagramas con respecto a las tareas que corresponden. Las tareas correspondientes son:

1. Orden de Servicio
2. Inventario
3. Préstamo

Éstas tareas son las que satisface el sistema y de ahí se detalla los participantes, las acciones y funcionamiento del sistema. Comenzaremos por las gráficas de *Orden de Servicio*, después de *Inventario* y por último las de *Préstamo*.

## Orden de servicio

Nombre:	Realizar llamada
Objetivo:	Realizar llamada al departamento de soporte técnico.
Actor:	Académico o administrativo de la Universidad Pedagógica Nacional.
Precondición:	Saber el número de extensión del departamento de soporte técnico o de mesa de ayuda ( <i>áreas de recepción correspondientes para levantar una orden de servicio</i> ). Además en caso de tener problemas con su equipo de cómputo, para solicitar apoyo con el mantenimiento preventivo o correctivo del equipo. También se asiste con la salida a Internet para un equipo o varios.
Postcondición:	Haber descrito su problema al personal del departamento encargado de recibir los datos del usuario.

Actor	Sistema
Realizar llamada	

Tabla 6.1: Caso de Uso - Realizar Llamada

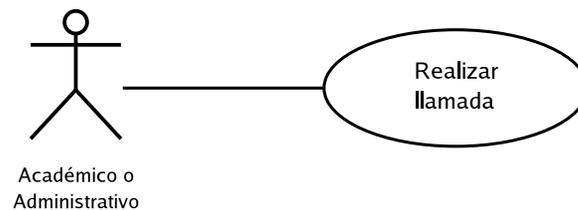


Figura 6.4: Caso de Uso - Realizar llamada

Nombre:	Levantar orden de servicio
Objetivo:	Tener datos concretos y concisos para levantar una orden de servicio.
Actor:	Administrador básico
Precondición:	Tener nociones básicas de operación del sistema. Saber sus datos para entrar al sistema (login y password). Recibir llamada para solicitar apoyo de acuerdo con la precondición de la tarea anterior.
Postcondición:	Llenar una forma de orden de servicio correctamente lo cual lleva a levantar una orden de servicio.

Actor	Sistema
Teclear datos para ingresar al sistema (login y password)	Despliega página con opciones para control del sistema.
Seleccionar opción de ordenes de servicio	Despliega forma de orden de servicio.

Tabla 6.2: Caso de Uso - Levantar orden de servicio

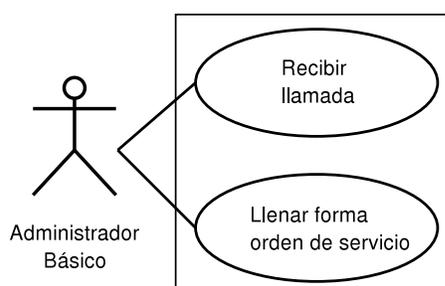


Figura 6.5: Caso de Uso - Levantar orden de servicio

Nombre:	Asignar orden de servicio
Objetivo:	El administrador del sistema deberá ingresar al sistema para asignar ordenes de servicio documentadas en sistema al personal técnico del departamento.
Actor:	Administrador del sistema
Precondición:	Saber su datos para entrar al sistema (login y password). Tener un conocimiento casi total del sistema.
Postcondición:	Haber asignado las ordenes pendientes a su personal.

Actor	Sistema
Teclear datos para ingresar al sistema (login y password)	Despliega página con opciones para control del sistema.
Seleccionar opción de seguimiento de ordenes	Despliega una lista de órdenes por asignar y asignadas a personal.

Tabla 6.3: Caso de Uso - Asignar orden de servicio

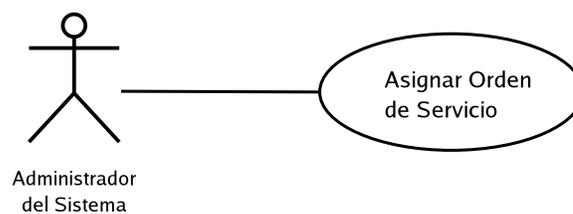


Figura 6.6: Caso de Uso - Asignar orden de servicio

Nombre:	Realizar orden de servicio
Objetivo:	Solucionar la problemática que se tuvo con el equipo de cómputo.
Actor:	Técnico del departamento de soporte.
Precondición:	El técnico deberá entrar al sistema con sus datos respectivos (login y password). La asignación de la orden deberá corresponder con la persona que entre al sistema.
Postcondición:	Correcto funcionamiento de la solicitud asignada.

Actor	Sistema
Teclear datos para ingresar al sistema (login y password)	Despliega página de inicio la cual muestra si tiene ordenes de servicio.
Seleccionar orden de servicio	Muestra todas las referencias para localizar la máquina y al usuario. Además de tener la opción de imprimir la orden para obtención de firmas.

Tabla 6.4: Caso de Uso - Realizar orden de servicio

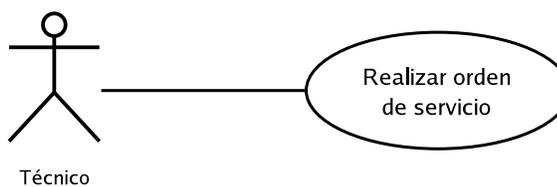


Figura 6.7: Caso de Uso - Realizar orden de servicio

Nombre:	Firmar el término de orden
Objetivo:	La persona que pidió asistencia deberá estar satisfecho con la conclusión de la orden de servicio.
Actor:	Académico o Administrativo
Precondición:	Haber terminado satisfactoriamente con la orden de servicio asignada al Técnico.
Postcondición:	La persona que pidió asistencia deberá firmar la orden de servicio de conformidad en reparación al problema que ordenó.

Actor	Sistema
Firmar término de orden	

Tabla 6.5: Caso de Uso - Firmar término de orden

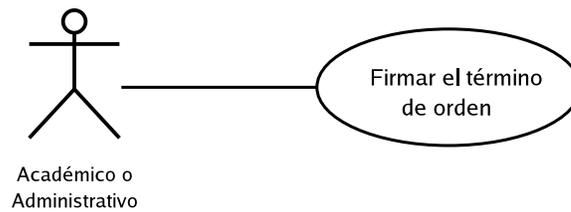


Figura 6.8: Caso de Uso - Firmar término de orden

Nombre:	Cerrar orden
Objetivo:	Finalizar con la orden de servicio eliminandola del sistema.
Actor:	Administrador del sistema
Precondición:	Tener la firma de conformidad de la orden de servicio. Presentar la orden de servicio al administrador del sistema.
Postcondición:	

Actor	Sistema
Teclear datos para ingresar al sistema (login y password)	Despliega página con opciones para control del sistema.
Buscar el número de orden de servicio correspondiente a la impresión	Se muestra un catálogo de las ordenes asignadas en el transcurso del día.
Seleccionar la opción de cerrar orden	Elimina de lista la orden seleccionada por el administrador del sistema. Para que, posteriormente, éste no sea reflejado en catálogo.

Tabla 6.6: Caso de Uso - Cerrar orden de servicio

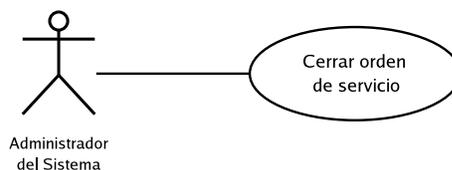


Figura 6.9: Caso de Uso - Cerrar orden de servicio

## Inventario

Nombre:	Llenar forma de inventario
Objetivo:	Llenar la forma con datos concretos y concisos de los equipos a capturar.
Actor:	Técnico del departamento de soporte.
Precondición:	El inventario se asigna por áreas previamente citadas para captura.
Postcondición:	Captura de equipo(s) de una persona.

Actor	Sistema
Ingresar al sistema, selecciona opción inventario	Despliega apartado de usuarios.
Buscar el nombre del usuario.	Verifica la validez del usuario en caso de existir se procede a llenado de formas de inventario, en otro caso, se pasa a forma de captura de usuario.

Tabla 6.7: Caso de Uso - Llenar forma de inventario

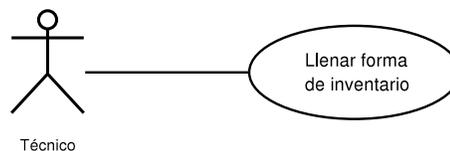


Figura 6.10: Caso de Uso - Llenar forma de inventario

Nombre:	Firma de aceptación de captura
Objetivo:	La persona a quién se le hace captura del bien debe estar satisfecha con el número de bienes capturados.
Actor:	Académico o administrativo de la Universidad Pedagógica Nacional..
Precondición:	Haber terminado satisfactoriamente la captura de bienes computacionales de una persona. Tener impresas las formas de captura.
Postcondición:	La persona a quién se le capturó los bienes deberá firmar la(s) forma(s).

Actor	Sistema
Firmar resguardo de bien	

Tabla 6.8: Caso de Uso - Firma de aceptación de captura

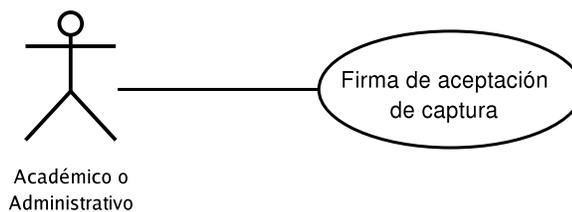


Figura 6.11: Caso de Uso - Firma de aceptación de captura

Nombre:	Enviar formato para resguardo físico
Objetivo:	Tener guardado el documento que avalado por el usuario que tiene un bien bajo su cargo.
Actor:	Técnico del departamento de soporte.
Precondición:	Tener firmado el resguardo.
Postcondición:	Almacenamiento de documentos de bienes.

Actor	Sistema
Tener firmados los resguardos capturados.	

Tabla 6.9: Caso de Uso - Enviar formato para resguardo físico

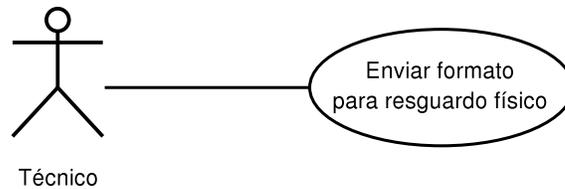


Figura 6.12: Caso de Uso - Enviar formato para resguardo físico

Nombre:	Archivar formatos de inventario
Objetivo:	Tener almacenado los bienes resguardados para aclaraciones posteriores.
Actor:	Administrador del sistema
Precondición:	Tener ordenado la documentación por área.
Postcondición:	

Actor	Sistema
Resguarda información de bienes.	Tiene capturada la información y ofrece un listado de bienes computacionales por área en la sección de Reportes.

Tabla 6.10: Caso de Uso - Archivar formatos de inventario

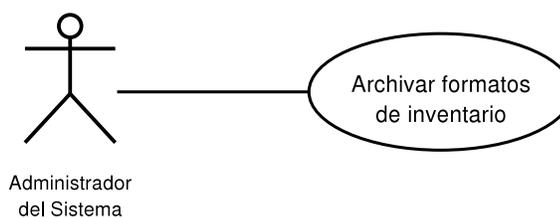


Figura 6.13: Caso de Uso - Archivar formatos de inventario

**Préstamo**

Nombre:	Solicitar préstamo
Objetivo:	Solicitar préstamo de equipo de cómputo.
Actor:	Académico, administrativo o alumno de la Universidad Pedagógica Nacional.
Precondición:	Necesitar del préstamo de equipo.
Postcondición:	Solicitar el préstamo de equipo a Informática.

Actor	Sistema
Solicitar préstamo	

Tabla 6.11: Caso de Uso - Solicitar préstamo

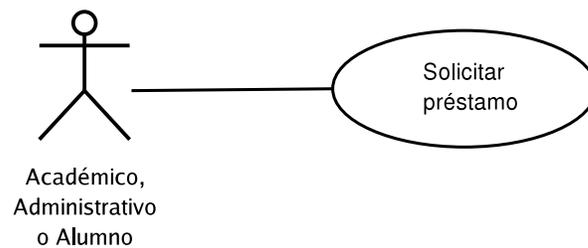


Figura 6.14: Caso de Uso - Solicitar préstamo

Nombre:	Llenar forma de préstamo
Objetivo:	Tener datos necesarios para levantar una orden de préstamo.
Actor:	Administrador básico
Precondición:	Recibir solicitud de préstamo.
Postcondición:	Llenar la forma de préstamo de equipo.

Actor	Sistema
Seleccionar opción de préstamo	Despliega forma para préstamo de equipo.

Tabla 6.12: Caso de Uso - Llenar forma de préstamo

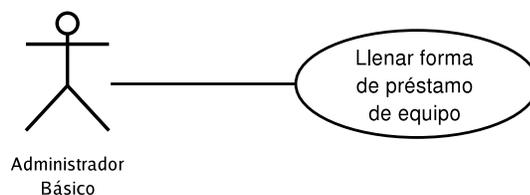


Figura 6.15: Caso de Uso - Llenar forma de préstamo

Nombre:	Firma de préstamo
Objetivo:	Obtener firma de responsiva y aprobación de equipo.
Actor:	Académico, administrativo o alumno Administrador o Almacén
Precondición:	Tener la forma de préstamo impresa.
Postcondición:	Tener la firma del usuario y de almacén.

Actor	Sistema
Firma de préstamo	

Tabla 6.13: Caso de Uso - Firma de préstamo

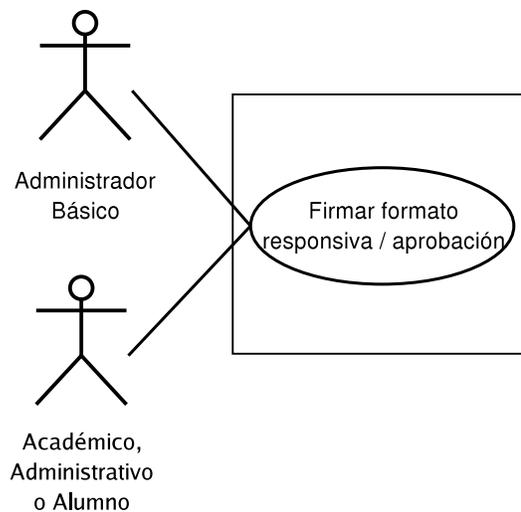


Figura 6.16: Caso de Uso - Firma de préstamo

Nombre:	Prestar equipo
Objetivo:	Prestar el equipo solicitado anteriormente.
Actor:	Almacén
Precondición:	Tener la forma de préstamo con sus firmas respectivas.
Postcondición:	Devolver el equipo cuando termine el préstamo de equipo.

Actor	Sistema
Prestar el(los) equipo(s) solicitado(s)	

Tabla 6.14: Caso de Uso - Prestar equipo

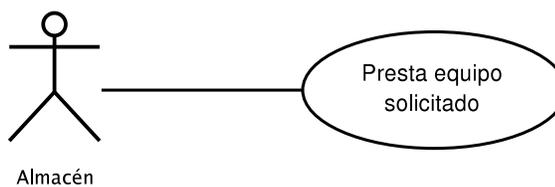


Figura 6.17: Caso de Uso - Prestar equipo

Nombre:	Entregar equipo
Objetivo:	Devolver el equipo prestado.
Actor:	Académico, administrativo o alumno Administrador o Almacén
Precondición:	Llevar consigo la forma de préstamo para cerrar la orden y tener el documento como informativo para reportes posteriores.
Postcondición:	Terminar con la orden para poder prestar ese equipo a solicitudes posteriores.

Actor	Sistema
Devuelve el equipo	
Recibir equipo	Se cierra orden de préstamo.

Tabla 6.15: Caso de Uso - Entregar equipo

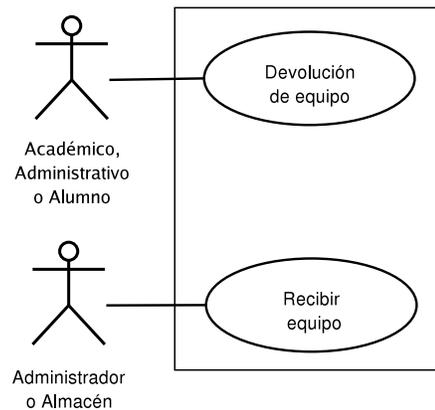


Figura 6.18: Caso de Uso - Entregar equipo

## 6.2.2 Diagramas de Actividades

Se presentan los diagramas de actividades que componen la sistema. Como ya se ha mencionado anteriormente (ver capítulo 5), estos diagramas sirven para tener una visión gráfica de las actividades que realizaran las personas involucradas en la actividad. Comenzaremos mostrando los diagramas con respecto a las tareas.

Las tareas correspondientes son:

1. Orden de Servicio
2. Inventario
3. Préstamo

Éstas tareas no sólo incluyen la participación del sistema sino también de las personas involucradas, son diagramas de acciones de las personas por las cuales pasa la actividad tanto para iniciarla como para terminar dicha tarea. Estos diagramas hacen una visión más concreta de la actividad que realiza cada persona.

Se explicará el desenvolvimiento del diagrama, así como algunas cuestiones en las que el sistema participe.

### Diagrama de actividades - Ordenes de Servicio

El diagrama se divide en las cuatro personas que se involucra en esta actividad **Cliente**, que es el usuario ya sea de tipo académico o administrativo de la institución; **Administrador Simple**, que es la persona que maneja el sistema pero sólo para captura de datos; **Administrador del sistema**, persona que controla la mayor parte del sistema teniendo privilegios de asignación de tareas para este acaso en particular; **Técnico**, persona que se encarga de darle mantenimiento a los equipos de cómputo y sobre el sistema sólo tiene la participación de ver las ordenes asignadas a él.

La actividad comienza cuando el cliente tiene un problema de carácter técnico de cómputo, es decir, se puede tener un problema de no saber respaldar información de virus, sobre conexión a red, sobre falla en el desempeño de su equipo, etcétera. El **cliente** realiza una llamada al departamento correspondiente (**Mesa de Ayuda**), para reportar su problema; después el **administrador simple** captura los datos del cliente sobre se ubicación y la descripción del problema. Una de las tareas constantes del **administrador del sistema** es revisar las nuevas ordenes asignadas y que él deberá de distribuir. El **técnico** por su parte es avisado de tener una nueva orden de servicio por correo y además se puede ver la asignación a través de internet.

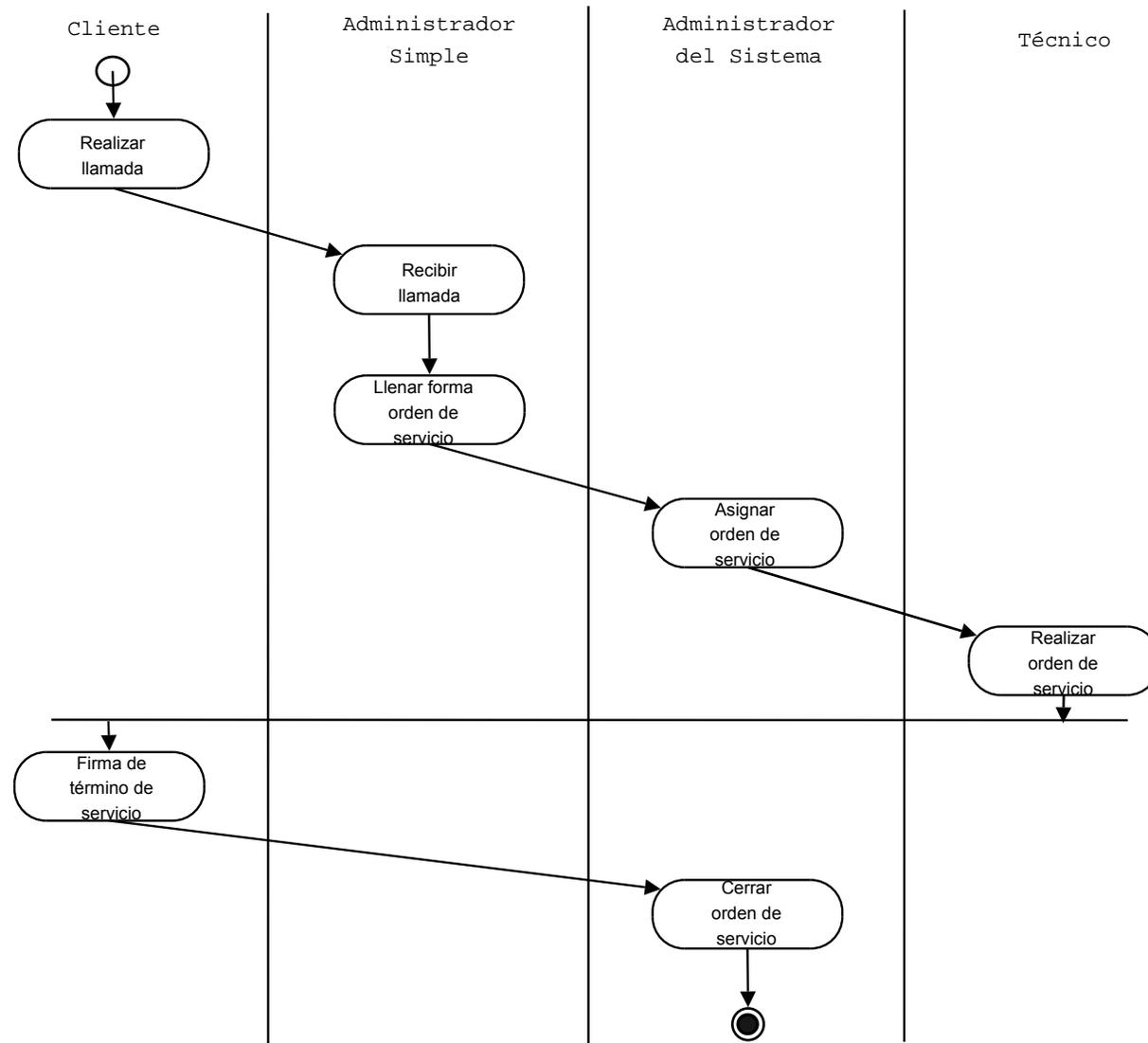


Figura 6.19: Diagrama de Actividades de orden de servicio

Se imprime esta orden para que una vez realizada, el usuario la firme de satisfacción de servicio. El técnico lleva la carta al administrador del sistema para que él cierre la orden y no tenga el técnico ordenes pendientes.

La Figura 6.19, muestra el Diagrama de Actividad correspondiente a la Orden de Servicio.

### **Diagrama de actividades - Inventario**

Ya una vez conocidas las personas involucradas del sistema, se menciona el desarrollo de la actividad. Para esta actividad, se tiene que el técnico, ingresa al sistema a través de internet y en la parte de inventario captura los datos pedidos por el sistema, de todo equipo computacional que exista, enfocándose a los bienes de CPU (de escritorio o servidor), Monitor, Scanner, Laptop, Regulador o No-Break. Después de haber capturado se imprime el formato para que el usuario responsable del equipo firme de que el equipo está bajo su resguardo. Esta tarea se hace por el inventario anual que se hace obligatoriamente en la institución. Después de ahí se archivan los formatos firmados y obviamente en la base de datos para después actualizar o como base para manejar el control que exista también en órdenes de servicio.

La Figura 6.20, muestra el Diagrama de Actividad correspondiente a Inventario.

### **Diagrama de actividades - Préstamo de equipo**

Para esta actividad Informática ofrece el préstamo de equipo de cómputo a cualquier persona que pertenezca a la Universidad. En este caso el cliente solicita préstamo de equipo a Informática, el administrador se encargará de llenar una solicitud de préstamo de equipo, una vez teniendo la forma impresa y de consultar si se tiene el equipo, se firma de responsiva y de préstamo del equipo. Almacén presta el equipo solicitado. En cuanto termina el préstamo de equipo, el usuario devuelve el equipo y se cierra la orden de préstamo.

La Figura 6.21, muestra el Diagrama de Actividad correspondiente a Préstamo.

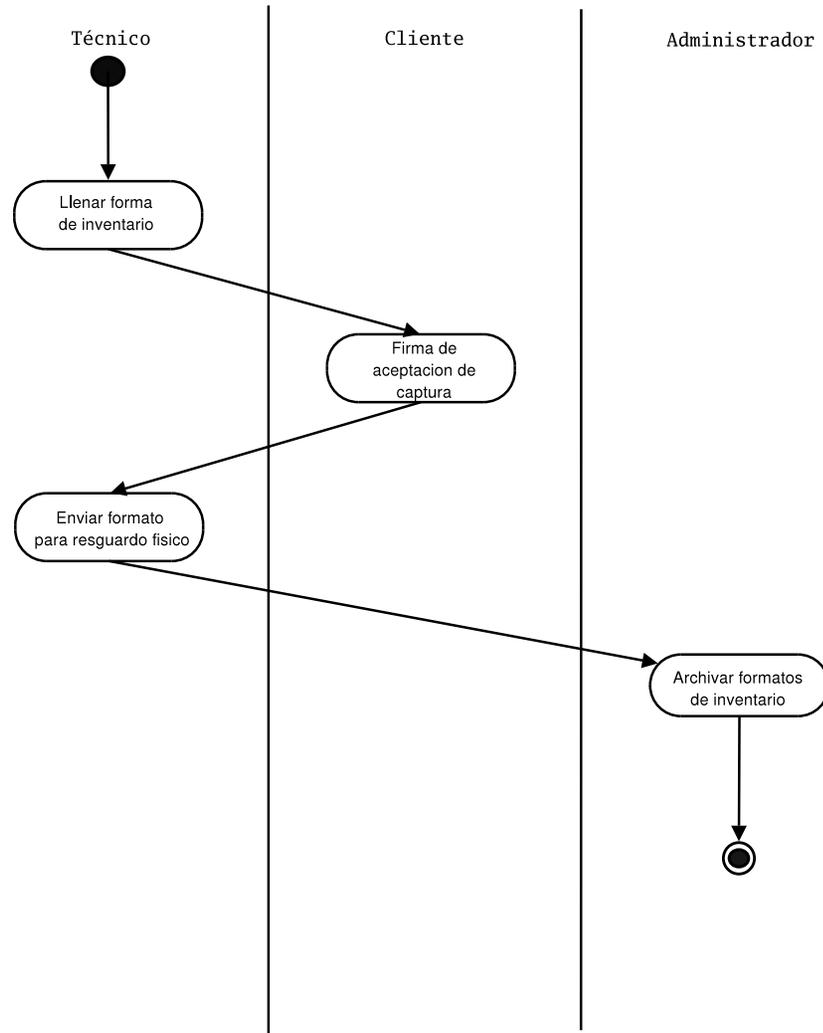


Figura 6 20: Diagrama de Actividades de inventario

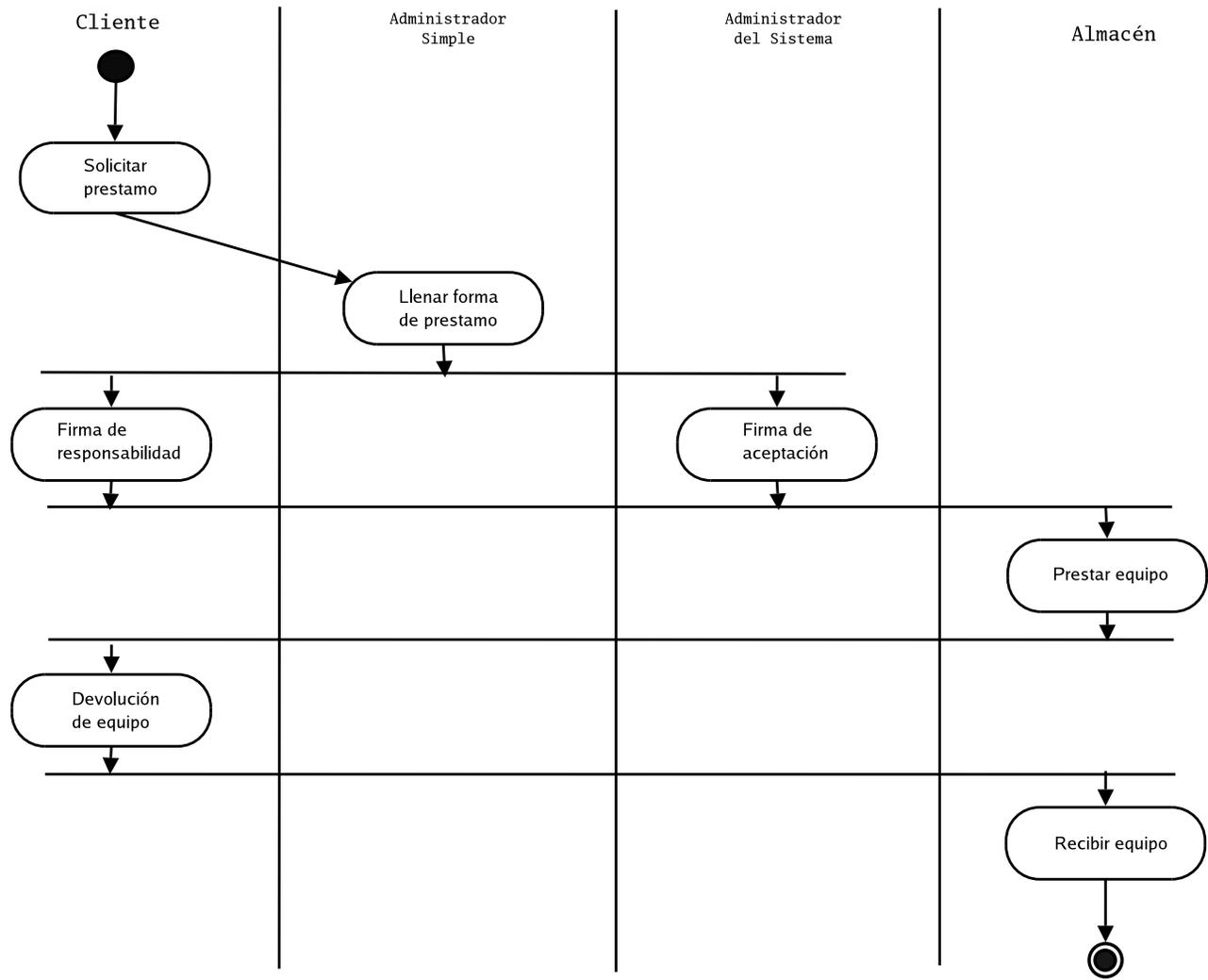


Figura 6.21: Diagrama de Actividades de préstamo de equipo.

## 6.3 Diseño del sistema

En muchas ocasiones para las organizaciones es necesario establecer una infraestructura de procesamiento de información, que cuente con los elementos para proveer información adecuada, exacta y oportuna.

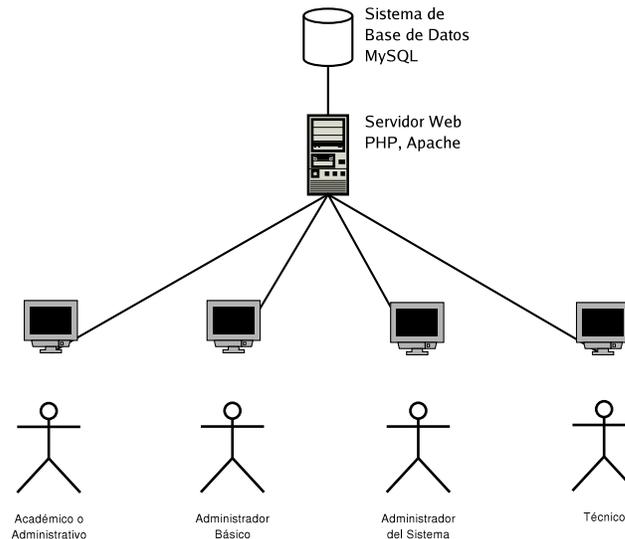


Figura 6.22: Esquema del sistema basado en arquitectura Cliente/Servidor.

El modelo Cliente/Servidor reúne las características necesarias para proveer esta infraestructura, independientemente del tamaño y complejidad de las operaciones de las organizaciones públicas o privadas y de forma consecuente desempeña un papel importante dentro del proceso de evolución de las organizaciones. IBM define al modelo Cliente/Servidor como la tecnología para proporcionar al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo, incluso a través de la organización en múltiples plataformas[29]. Los sistemas Cliente/Servidor se pueden ver de la siguiente manera, los clientes solicitan que una tarea sea realizada entonces el servidor realiza dicha tarea y regresa la información al cliente de la red. Cada componente dentro de estos sistemas se encarga de realizar su tarea exclusivamente. La arquitectura Cliente/Servidor que seguirá el SIST se muestra en la Figura 6.22. Donde se muestra a las personas involucradas en el uso y administración del sistema, además de la gráfica de arquitectura que se usará.

Para muchas organizaciones, los sistemas de información basados en computadoras son el corazón de las actividades cotidianas y objeto de gran consideración en la

toma de decisiones. Es por ello que se ha escogido la arquitectura Cliente/Servidor, debido a que PHP es un lenguaje de programación destinado para aplicaciones de red (Figura 3.3), en parte a MySQL como un manejador de base de datos acoplado a PHP y que puede utilizarse en red (Figura 3.4) y Apache quién es el mediador entre las peticiones que hace el *Cliente* y los resultados que envía el *Servidor* (ver apartado de Apache 2.2.2).

### 6.3.1 Diseño de la base de datos

Se presenta el diagrama de entidad-relación correspondiente al Sistema Integral de Soporte Técnico. Este diagrama muestran las tablas que componen al sistema así como las relaciones que existen de una tabla a otra mediante registros, describiendo su tipo.

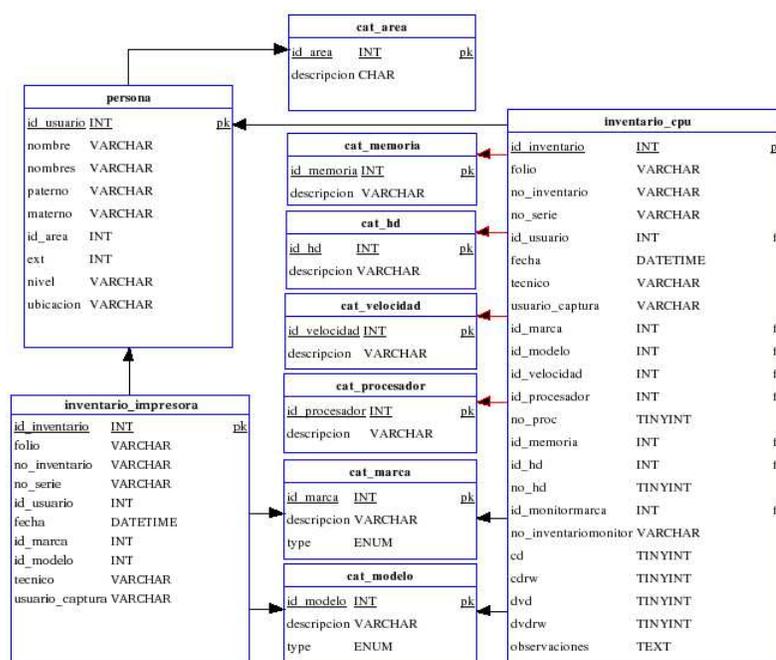


Figura 6.23: Entidad-Relación del módulo de inventario.

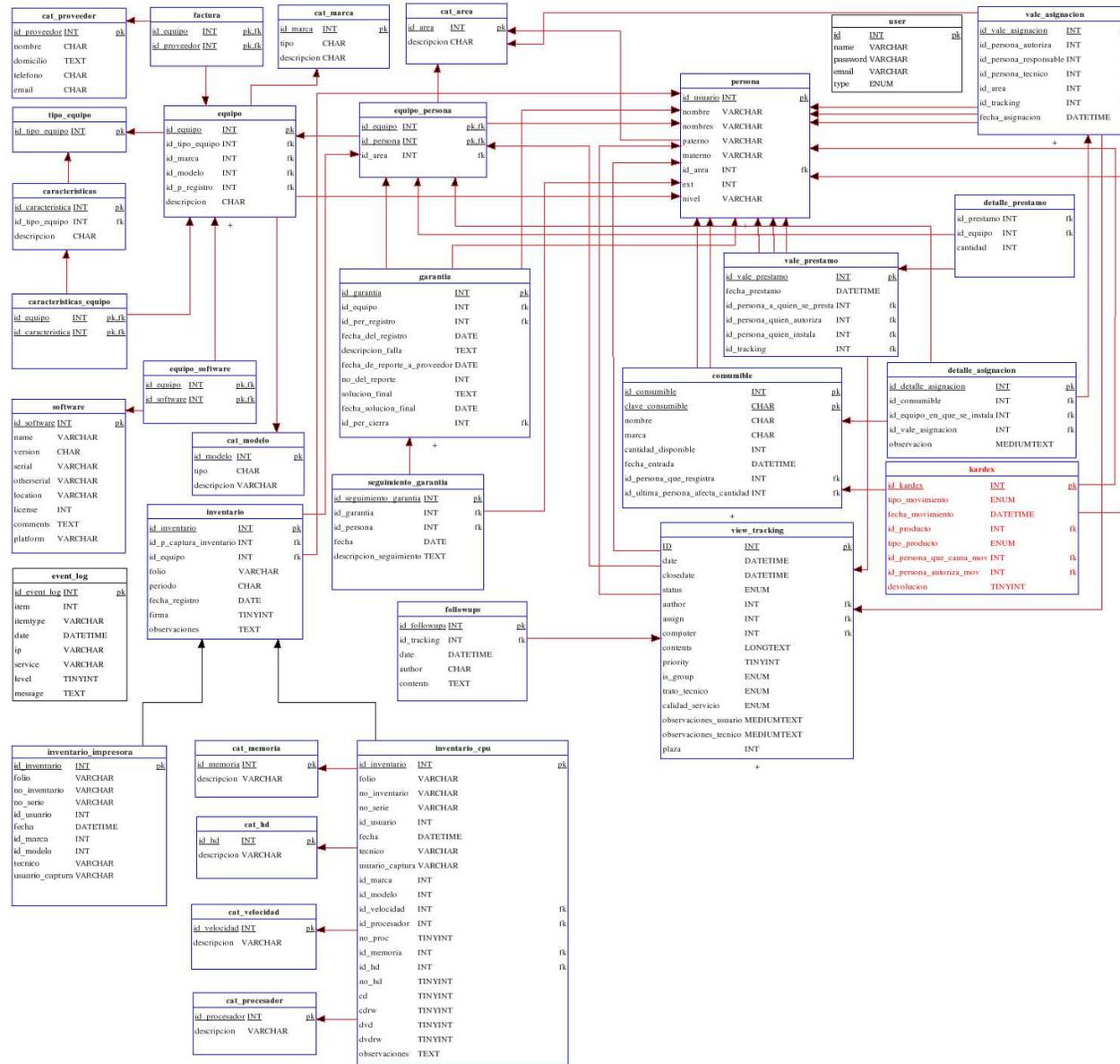


Figura 6.24: Modelo Entidad-Relación del Sistema Integral de Soporte Técnico

En la base de datos se integran los módulos presentados en la sección 6.2 (*Análisis de sistema*), el módulo de inventario está conformado por el siguiente modelo entidad-relación mostrado en la Figura 6.23.

En el módulo de inventario se registran los bienes informáticos particulares como: impresora, computadoras personales, portátiles y servidores. Las tablas que contienen la relación de registros de bienes se encuentra en las tablas de *inventario\_computadora* e *inventario\_impresora*. Desprendiendo en si dos cualidades globales, computadoras (*inventario\_computadora*) e impresoras (*inventario\_impresora*). Las computadoras pueden ser: personales, servidores y portátiles. Además de que cada equipo cuenta con una persona (tabla *persona*) a cargo del equipo. Se anexaron una serie de tablas que contendrá un catálogo particular debido a la variedad de productos que hay para los equipos de cómputo.

El módulo de órdenes de servicio esta conformado por el modelo entidad-relación de la Figura 6.25. Este módulo registra las órdenes de servicio que se tenga que realizar a equipos de cómputo de la universidad. Para esto es necesario que el equipo haya sido registrado previamente en el área de inventario, para poder tener las características del equipo al cual se le hará el servicio. Así como los datos personales de la persona a cargo del equipo. La tabla *view\_tracking* es la que muestra las órdenes de servicio agregadas al sistema que se encuentran es estado abierto, así como registros almacenados de órdenes concluidas en caso de necesitarlas para futuros reportes. La tabla de *equipo* toma datos de inventario para saber las características del equipo.

El diseño de la base de datos se muestra en la Figura 6.24, dentro de su modelo entidad-relación se encuentran los campos que componen al SIST.



un estado de transiciones el cual se muestra en la figura 6.26.

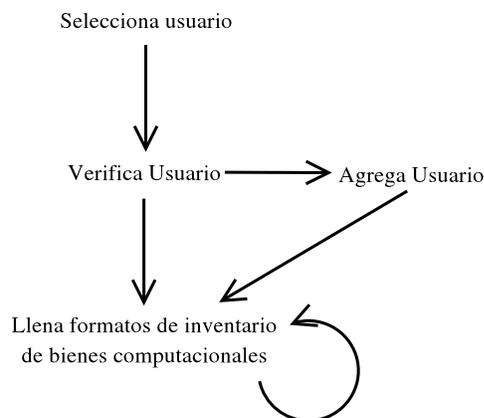


Figura 6.26: Flujo de acciones del módulo de inventario.

Esta transición (ver figura 6.26) es efectuada pasando por diferentes páginas que conllevan a la realización de esta tarea. A continuación se presentan las pantallas con respecto al flujo de acciones del módulo de inventario correspondientes a las Figuras 6.27, 6.28, 6.29 y 6.30.

#### Registro de Inventario

Bienvenidos al Sistema de Soporte Técnico. En esta sección se registra el Inventario de Hardware.

Nombre	Apellido Paterno	Apellido Materno
<input type="text" value="Elija una opción"/> <input type="text" value="Elija una opción"/>	<input type="text" value="Elija una opción"/>	<input type="text" value="Elija una opción"/>

[Si no esta el nombre en lista agregarlo aquí.](#)

Figura 6.27: Inicio del módulo de inventario.

Se selecciona de un catálogo el nombre de la persona, validando el usuario en la forma la Figura 6.28, en caso de ser usuario contenido en la base de datos se muestra su relación de datos del usuario como son nombre completo, área, departamento, y e. identificador que usuario tiene para relacionarlo con cualquier módulo del sistema. En caso de que el usuario no exista como tal en la base de datos se incluye otra forma de captura del usuario (Figura 6.29) para que una vez capturado se pase a la parte de llenado de inventario.

**El usuario existe**

1	Erick *	Ruiz	Rojas	Salón de Cómputo 334	Redes
---	---------	------	-------	----------------------	-------

Figura 6.28: Verifica la existencia del usuario.

**Agregar Usuario - Teclea los datos a anexar a lista**

Nombre de pila, nombres secundarios		Apellido Paterno		Apellido Materno	
<input type="text"/>		<input type="text"/>		<input type="text"/>	
Área		Ubicación		Nivel	Ext.
Elija una opción		<input type="text"/>		<input type="text"/>	<input type="text"/>
<input type="button" value="Continuar"/>			<input type="button" value="Limpiar"/>		

Figura 6.29: Se agregan personas que resguardan equipos de cómputo.

En esta forma sólo se contemplan datos de nombre, área es un catálogo en este caso estático ya que no hay áreas por una persona, en ubicación se interpreta como lugar ya sea por nombre del edificio, el nivel por el piso del edificio y extensión donde se le puede encontrar a la persona. Datos comunes para el resguardo del equipo y asignación de órden.

**Inventario - Agregar**

Registra los datos completos de CPU y /o Impresora. Pon atención en la captura de los No. de Inventario , No. de Serie, etc.

Datos Generales			
Nombre:	A. Paterno:	A. Materno:	
Erick *	Ruiz	Rojas	
Ubicación:	Área:	Nivel:	Extensión:
Redes	Salón de Cómputo 334	Sotano	1268
Tipo de inventario		<input type="checkbox"/> Impresora <input type="checkbox"/> Laptop <input type="checkbox"/> CPU <input type="checkbox"/> Servidor	

Figura 6.30: Datos generales del usuario a quien se le cargarán bienes.

En esta forma se ingresa el inventario de bienes de cómputo: computadora personal portátil, servidor o impresora. Este llenado de inventario la persona puede tener uno o más equipos de resguardo. Al término del llenado de las formas aparece un apartado enlistando los equipos que tenga el usuario. Donde las personas que capturan podrán editar, eliminar o imprimir el formato de resguardo del bien.

Comenzando con el formato de computadora personal tiene que considerarse el hecho de que toda computadora personal debe tener un monitor con marca y modelo. Así

Datos generales del CPU	
Folio de Inventario CPU: <b>20041C193581</b>	Tecnico: erick
Marca CPU Elija una opcion	Modelo CPU Elija una opcion
Procesador Elija una opcion	Memoria Elija una opcion
Velocidad Elija una opcion	Disco Duro Elija una opcion
Número de inventario CPU	Número de serie CPU
Marca Monitor Elija una opcion	Número de inventario Monitor
Pulgadas	
CD Si <input type="checkbox"/> No <input type="checkbox"/> Quemador Si <input type="checkbox"/> No <input type="checkbox"/> DVD Si <input type="checkbox"/> No <input type="checkbox"/> Scanner Si <input type="checkbox"/> No <input type="checkbox"/> No-Break Si <input type="checkbox"/> No <input type="checkbox"/>	<b>Otros Perifericos</b> Número de Inventario Scanner Marca del Scanner Número de Inventario No-Break Marca No-Break
Observaciones	
Firma del Usuario Si <input type="checkbox"/> No <input type="checkbox"/>	
<input type="button" value="Guardar"/> <input type="button" value="Limpiar Forma"/>	

Figura 6.31: Formato de captura para computadora personal.

como número de inventario. Los demás campos son comunes a excepción del folio que se distingue de los demás por tener una “C” que significa que es una computadora personal.

Datos generales del SERVIDOR	
Folio de Inventario Servidor: <b>200413193501</b>	Tecnico: erick
Marca Servidor Elija una opcion	Modelo Servidor Elija una opcion
Procesador Elija una opcion	Memoria Elija una opcion
Número de procesadores	Disco Duro Elija una opcion
Velocidad Elija una opcion	Número de Disco Duros
Número de inventario servidor	Número de serie servidor
Marca Monitor Elija una opcion	Número de inventario Monitor
Pulgadas	
CD Si <input type="checkbox"/> No <input type="checkbox"/> Quemador Si <input type="checkbox"/> No <input type="checkbox"/> No-Break Si <input type="checkbox"/> No <input type="checkbox"/>	Número de Inventario No-Break Marca del No-Break
Observaciones	
Firma del Usuario Si <input type="checkbox"/> No <input type="checkbox"/>	
<input type="button" value="Guardar"/> <input type="button" value="Limpiar Forma"/>	

Figura 6.32: Formato de captura para servidor.

El formato de servidor se tiene el numero de procesadores, además del número de unidades de disco que tiene el servidor. Puede tener un monitor o pertenecer a un

rack y que tenga un monitor común. Por eso se agregó el monitor aunque puede o no contener monitor o puede ser que tenga uno pero común para otros servidores. El folio se distingue por tener una “*S*” que significa que es un servidor.

Datos generales de Laptop	
Folio de inventario Laptop: <b>20041L193581</b>	Tecnico: erick
Marca Laptop <input type="text" value="Elija una opcion"/>	Modelo Laptop <input type="text" value="Elija una opcion"/>
Procesador <input type="text" value="Elija una opcion"/>	Memoria <input type="text" value="Elija una opcion"/>
Velocidad <input type="text" value="Elija una opcion"/>	Disco Duro <input type="text" value="Elija una opcion"/>
Número de inventario laptop <input type="text"/>	Número de serie laptop <input type="text"/>
CD <input type="checkbox"/> Si <input type="checkbox"/> No	Queemador <input type="checkbox"/> Si <input type="checkbox"/> No
DVD <input type="checkbox"/> Si <input type="checkbox"/> No	
Observaciones <input type="text"/>	
Firma del Usuario <input type="checkbox"/> Si <input type="checkbox"/> No	
<input type="button" value="Guardar"/> <input type="button" value="Limpiar Forma"/>	

Figura 6.33: Formato de captura para computadora portátil.

El formato de computadora portátil se descarta el hecho de tener un monitor, más sus respectivas características comunes. El folio se distingue por tener una “*L*” que significa que es una computadora portátil.

Datos generales de Impresora	
Folio de inventario Impresora <b>20041I76390</b>	Tecnico:erick
Marca de Impresora <input type="text" value="Elija una opcion"/>	Número de inventario Impresora <input type="text"/>
Modelo <input type="text" value="Elija una opcion"/>	Número de serie Impresora <input type="text"/>
Firma del Usuario <input type="checkbox"/> Si <input type="checkbox"/> No	
<input type="button" value="Imprimir"/> <input type="button" value="Limpiar"/>	

Figura 6.34: Formato de captura para impresora.

El formato de impresora sólo consta de datos como marca modelo y sus respectivos números de inventario y de serie. El folio se distingue por tener una “*I*” que significa que es una impresora.

## 6.4 Implementación

Se inicia el sistema, creando diagramas de casos de uso identificando las personas involucradas en el sistema diagramas de actividades de los módulos después se realizar

los diagramas entidad-relación sobre la base que contendrá los datos, para después crear programas en PHP teniendo una base con características operantes y funcionales de conectividad y consulta permitiendo arrojar datos de MySQL mediante PHP de manera sencilla y útil. Se inicia haciendo un módulo, teniendo en cuenta cuales son las necesidades de los módulos siguiendo las diferentes funcionalidades de otros módulos. Esa es la manera en cómo se ensambló el proyecto y en caso de haber una mejora o amplitud de algún módulo, se hace afectando solo las ramas necesarias del módulo en cuestión que en ciertas partes, ya que por su diseño modular resulta fácil identificar su independencia con otros módulos.



## Capítulo 7

# Conclusiones y Trabajo a Futuro

Para ampliar el sistema lo haremos desarrollando más automatizaciones para el área de Informática y para otras áreas, las cuales tengan la necesidad de utilizar uno o varios módulos que presenta el sistema; como por ejemplo, órdenes de servicio, no obstante que este sistema se refiere a la Subdirección de Informática existen otras áreas que tiene el mismo problema para realizar servicios o de llevar un control automatizado de los bienes o procesos; tomando como ejemplo un área que tenga que ver con el mantenimiento a la universidad. Este sistema no se enfocaría a hacer parte de una solicitud de reparación técnica de computadoras, sino que (bajo el mismo esquema) el sistema realizará órdenes de tipo eléctrico, de mantenimiento, etcétera, que finalmente son áreas similares, pero enfocadas hacia otras áreas con la misma particularidad, aplicando diferentes especificaciones necesarias para cada área. Este sistema propone la solución que el área requiere, sin embargo puede ser ampliada o volver a ser particularmente específica hacia otras áreas que tengan similitud a los módulos implementados en el sistema.

Los objetivos que se han propuesto desde un principio para la realización del sistema, han hecho de este proyecto, un proyecto a explotar en muchos rubros. Dentro de las propuestas de trabajo a futuro sobre el proyecto se encuentran las siguientes:

1. Las órdenes de servicio siguen el proceso de reportar el problema al departamento de *Mesa de Ayuda*, para que después el administrador en cuestión, reasigne la orden a un técnico en soporte. Otra manera que se sugiere para realizar el cumplimiento de este proceso es que cualquier usuario a través de Internet llene la forma en cuestión, para que posteriormente el administrador apruebe y canalice la orden a un técnico. Se propone esta alternativa que facilite o agilice el levantamiento de órdenes de servicio.

2. Existen dos áreas en la universidad que se pueden involucrar en los módulos de inventario, almacén e informática. El sistema basa su captura para informática mientras que el área de almacén tiene un control de inventario más extenso y completo, sobre todos los bienes de la universidad. Con el sistema, como se ha mencionado el objetivo es ser congruente con los registros capturados junto con los registros que contiene el área de almacén. Desafortunadamente el inventario que lleva a cabo el área de informática se realiza una vez al año. Para esto existe la posibilidad que el usuario responsable del equipo lo deje sin resguardo del mismo en el momento que se retire. Los registros del sistema no son actualizados debido a que el inventario se realiza una vez al año y no está ligado aún este sistema con el área de *Almacén*. Dicha área compara el listado con su propia base de datos. Lo cual hace que se tengan dos bases de datos permitiendo así inconsistencias pues los registros no son actualizados en caso de ausencia. Dado que las áreas son independientes, aún no existe una unificación, lo cual hace que nuestra base de datos de usuarios se renueve cada año, así como el de los equipos. Una de las tareas a realizar posteriormente será unificar las áreas para así tener registros actualizados diariamente para que el equipo no se quede sin una persona a cargo del resguardo y no exista duplicado de información ni recursos.
3. Actualmente el número de inventario es proporcionado por el área de *Almacén* con un formato constante. El formato del número de inventario consta de las siguientes partes: *número de inventario, consecutivo, año, cd y parte*. Este número aparece de la siguiente forma: *18000000660001 03 001 5206*. Este código es grabado físicamente en el equipo (rayado o en plumón), cuando entra a la institución por primera vez. Este tipo de grabados es muy arcaico debido a que pueden duplicarse los lotes de equipos que llegan en la universidad. Pero además otra empresa que ha hecho el mismo trabajo (de hacer un inventario sobre el equipo) ha agregado a este código una etiqueta numérica, la cual en ocasiones es totalmente distinta teniendo así 2 números de inventario diferentes para la misma cosa. La captura al sistema sobre el inventario de cómputo se realiza en base al número presentado en un principio, el establecido por el área de almacén para la universidad. Debido a ello y como propuesta para ampliar este proyecto, se propone la creación de códigos de barras para que el sistema tenga registro del equipo, eliminando así el problema de tener varios códigos y números de inventario.

# Bibliografía

- [1] Lorna Gomez Barrientos. *Control interno aplicable al inventario de una empresa familiar dedicada a la compra-venta de abarrotes*. Tesis Licenciatura, Universidad Villa Rica, 2003.
- [2] <http://www.portalmundos.com/mundoinformatica/linux/saboreslinux.htm>. de los Santos Sergio. *Los sabores de linux, un recorrido por sus distribuciones*. Internet.
- [3] <http://www.starlinux.net/staticpages/index.php?page=20031124005524899>. Pajuelo Vazquez Daniel. *El final de red hat gratis. ¿y ahora qué? fedora vs debian*. Internet, Noviembre 2003.
- [4] <http://fedora.redhat.com/about/name.html>. *The fedora name*. Internet, Diciembre 2003.
- [5] Viso Gurovich Elisa. *Autómatas y Lenguajes formales. Notas para el curso Teoría de la Computación*. Facultad de Ciencias UNAM, 1998.
- [6] <http://www.qwest.net/nav4/help/faqs/glossary.html>. *Glossary*. Internet, 2004.
- [7] [http://www.alphastarcomputer.com/fp\\_2000\\_glossary.htm](http://www.alphastarcomputer.com/fp_2000_glossary.htm). *Glossary*. Internet, 1999-2003.
- [8] [www.sindominio.net/biblioweb/telematica/catedral.html](http://www.sindominio.net/biblioweb/telematica/catedral.html). Raymond Eric S. y Soto Pérez José (Traductor). *La catedral y el bazar*.
- [9] [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html). Apache http server project. Internet, 1999-2004.
- [10] <http://es.tldp.org/LinuxFocus/pub/mirror/LinuxFocus/Castellano/May2000/article122.shtml>. Vidal Charles. *Apache: primera parte*. LUCAS.

- [11] <http://news.netcraft.com/archives/2004/11/index.html>. Netcraft. *El estudio de servidores www de netcraft, 1995-1999*.
- [12] <http://www.pecesama.net/php/index.php>. Santana Mancilla Pedro César. *Taller de php*. pecesama.net, Diciembre 2001.
- [13] <http://es.wikipedia.org/wiki/PHP>. Wikipedia. *PHP*. Wikipedia.
- [14] <http://www.programacion.com/php/articulo/porquephp/>. Dondo Agustin. *¿por qué elegir php?*. PHP en castellano.
- [15] <http://www.programacion.com/php/tutorial/php4/>. del Castillo Alvaro. *Webs dinámicas con PHP*. mailxmail.
- [16] <http://www.zend.com/apidoc/zend.overview.php>. Zend. *Extending PHP 4.0*. Zend, The *php* company.
- [17] Booch Grady, Rumbaugh James y Jacobson Ivar. *El lenguaje Unificado de Modelado UML*. Addison Wesley, 1999.
- [18] Welling Luke y Thomson Laura. *PHP and Mysql Web Development*. Developer's Library, 2da. edición, 2003.
- [19] Gil Rubio Francisco Javier, Tejedor Jorge A., Yague Panadero Cerbel Agustín, Villaverde Santiago Alonso y Gutiérrez Rodríguez Abraham. *Creación de Sitios Web con PHP 4*. Mc Graw Hill, 1ra. edición, 2001.
- [20] DuBois Paul. *MySQL*. Prentice Hall, A Simon AND Schuster Company, especial edición, 2000.
- [21] Petersen Richard. *Manual de referencia Linux*. Mc Graw Hill, 2da. edición, 2000.
- [22] Hughes Sterling. *PHP book Developer's Cookbook*. SAMS, 2001.
- [23] ICMP. *Principios de Contabilidad GENERALMENTE ACEPTADOS*. Instituto Mexicano de Contadores Públicos, A. C., 16ta. edición, 2001.
- [24] Alatraste Sealtiel. *Técnica de los costos*. Porrúa, 38ava. edición, 2001.
- [25] Tanenbaum Andrew S. and Woodhull Albert S. *Sistemas Operativos; Diseño e implementación*. Prentice Hall, A Simon AND Schuster Company, 2da. edición, 2002.

- [26] Robbins Kay A. y Robbins Steven. *UNIX Programación Práctica*. Prentice Hall, A Simon AND Schuster Company, 1era. edición, 1997.
- [27] Linux Free Magazine. *Manual de introducción al software libre en 8 pasos*. Linux Free Magazine, 2(5):37-38, 2004. <http://creativecommons.org/licenses/by-nc/1.0>.
- [28] <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>. Zavala Ruiz Jesús María. *Ingeniería de software*. Master's thesis, Universidad Autónoma Metropolitana-Azcapotzalco, 2000.
- [29] <http://www.monografias.com/trabajos13/cliser/cliser.shtml>. Zamacóna Cabra Gerardo. *Costos de los sistemas cliente servidor basado en actividades*. Internet, Octubre 2003.
- [30] <http://www.mandrakelinux.com/es/linux.php3>. *Linux mandrake - ¿qué es linux?*. Mandrake Linux, mayo 1999.
- [31] <http://www.maccare.com.ar/linux1.htm>. *Linux*. Internet, 2000.
- [32] <http://www.wmo.ch/web/www/WDM/Guides/Internet-glossary.html>. *Glossary*. Internet, Diciembre 2003.
- [33] <http://www.dartmouth.edu/~library/WWG/guide/Glossary.html>. *Www jargon glossary*. Internet.
- [34] <http://practice.findlaw.com/glossary.html>. *Glossary of technology*. Internet.
- [35] Silberschatz Avi, Korth Henry F. y Sudarshan S. *Fundamentos de Bases de datos*. Mc Graw Hill, 4ta. edición, 2002.
- [36] <http://www.monografias.com/trabajos7/resi/resi.shtml>. Rivas Yonekura Kenyiro. *Análisis y diseño de sistemas*. Internet.
- [37] <http://www.mmlabx.ua.es/mysql-postgres.html>. Aguilar Vicente y Suau Pablo. *Mysql vs. postgresql*. Internet, Agosto 2000.
- [38] <http://www.zend.com/zend/art/under-php4-hood.php>. Suraski Zeev. *Under the hood of php 4*. Zend.
- [39] Cisco Systems. *Academia de networking de Cisco Systems: Guía del primer año*. Cisco Press, 2da. edición, 2003.

- [40] [http://es.wikipedia.org/wiki/Active\\_Server\\_Pages](http://es.wikipedia.org/wiki/Active_Server_Pages). Wikipedia. *ASP* Wikipedia.
- [41] <http://es.wikipedia.org/wiki/API>. Wikipedia. *API*. Wikipedia.
- [42] <http://www.mysql.com>. MySQL Press. *History of MySQL*. Wikipedia.
- [43] <http://es.wikipedia.org/wiki/GNU/Linux>. Wikipedia. *GNU*. Wikipedia.
- [44] <http://www.zend.com/php5/articles/engine2-php-oo.php>. Zend. *Zend Engine II - PHP's OO Evolution*. Zend, The *php* company.