



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

---

**FACULTAD DE INGENIERÍA**

**OPTIMIZACIÓN Y REINGENIERÍA DEL SISTEMA WEB DENTIZT@**

TESIS PROFESIONAL  
QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN

PRESENTAN:  
BÁEZ ESCRIBANO ANUOAR  
TREVILLA ORTEGA CAROLINA

ASESOR:  
ING. JOSEFINA BÁRCENAS LÓPEZ



MÉXICO, D.F.

AGOSTO 2009



## Agradecimientos

### A LA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Por permitirnos ser parte de la máxima casa de estudios, disfrutar de sus instalaciones, ser nuestro hogar y brindarnos la formación universitaria que con orgullo siempre ejerceremos.

### A LA FACULTAD DE INGENIERÍA

Por brindarnos los conocimientos que hoy en día ponemos en práctica en nuestros trabajos, permitiéndonos ser útiles a la sociedad, formándonos un perfil analítico para dar soluciones ante cualquier obstáculo y persistencia para lograr nuestros objetivos.

### AL CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO

Por todas las facilidades en la realización de este proyecto y así poder servir a nuestra institución educativa, la UNAM, aportando nuestros conocimientos en este trabajo.

### A NUESTROS PROFESORES

Por cada una de sus enseñanzas y horas dedicadas a nuestra formación, porque lo que ahora somos es el reflejo de cada uno de ellos.

### A LA ING. JOSEFINA BÁRCENAS LÓPEZ

Por brindarnos la oportunidad de elaborar este proyecto, con el objetivo de fortalecer nuestra formación académica, gracias por transmitirnos sus conocimientos y por alentarnos en la culminación de este trabajo.

### AL ING. JOSÉ ANTONIO DOMÍNGUEZ HERNÁNDEZ

Por impulsarnos a ser excelentes ingenieros siendo para nosotros un gran ejemplo, por transmitirnos sus conocimientos y experiencia, gracias por estimularnos a culminar nuestra tesis.

### AL DR. JOSÉ S. TOLOSA

Por ser parte de este proyecto e impulsar el desarrollo tecnológico para el bien en la educación y por confiar en nosotros para la realización de este proyecto.

### AL ING. JESÚS RAMÍREZ ORTEGA

Por brindarnos todo su apoyo incondicionalmente, por confiar en nosotros, mil gracias.

### AL ING. JUAN GABRIEL SÁNCHEZ ROMERO

Por su disposición y sus valiosos consejos que nos brindó para la culminación de este proyecto, gracias.

### A NUESTROS AMIGOS

Por alentarnos durante el desarrollo de este proyecto, gracias por estar con nosotros en las buenas y en las malas, pero sobre todo gracias por todo su apoyo.

### A NUESTRAS FAMILIAS

Por ser parte esencial de este trabajo, porque sin el apoyo de ustedes esto no hubiera sido posible, gracias por la paciencia, la comprensión y el estímulo que de ustedes recibimos siempre.

### AL HONORABLE JURADO

**Anuoar y Carolina**



## Dedicatoria

Este trabajo quiero dedicarlo con todo mi amor y cariño a cada una de esas personas que siempre estuvieron detrás de mí alentándome todo el tiempo durante todo el desarrollo de este trabajo y confiaron en mí hasta el final.

A Dios a quien agradezco la oportunidad que me dio de concluir mis estudios, siempre dándome la sabiduría, el valor, la fuerza y la salud en los momentos difíciles.

A mis padres Celestina y Andrés, que me apoyaron en todo momento, a los cuales les debo la mayor parte de este trabajo, por su comprensión, su tolerancia y sobre todo su enorme amor hacia mí. Los amo.

A mi hermana Artemisa, quien siempre ha sido un ejemplo para mí, por sus consejos, su comprensión y por compartir sus experiencias. Gracias por todas tus palabras de aliento. Te quiero mucho hermanita.

A mi cuñado David, por ser un hermano para mí, comprenderme en las buenas y en las malas y por el apoyo que siempre me mostró.

A la Familia Trevilla Ortega, por el apoyo durante el desarrollo de este trabajo y alentarme a concluirlo.

Al amor de mi vida, Caro, por toda la comprensión, apoyo y amor que me brindo en todo momento, sin ti de verdad que no lo hubiera logrado. Gracias por hacerme participe de este proyecto. Te amo.

A mi hermano Leo, por estar siempre en las buenas y en las malas, gracias por tus palabras de aliento y por permitirme ser parte de tu familia. Te quiero mucho hermanito.

A mis amigos que fueron parte fundamental en este proyecto, Toño, Joselo, Ross, Otto, Dulce, María de Jesús, Evelin, Ruth, Gabriel y Bere. Gracias muchachos por alentarme siempre en los momentos más difíciles, por su apoyo incondicional y sus buenos consejos.

A la Auditoría Superior de la Federación, por todas las facilidades que me otorgó para la realización de este proyecto.

Al Prof. Carlos Cruz Cruz, por todo el apoyo durante el desarrollo de esta tesis y las facilidades que me brindó hasta el final.

Al C.P. Javier Pérez Saavedra, por todas las facilidades que me otorgó para la realización de esta tesis. Gracias por toda la confianza que me brindó durante el desarrollo de este trabajo y por alentarme en todo momento.

**Anuoar**



## Dedicatoria

Quiero dar gracias a Dios, por brindarme salud y una familia extraordinaria por las cuales he cumplido una de las metas más importantes de mi vida, ser una profesionista, gracias también por permitirme conocer a personas valiosas que han guiado mi camino, porque sé que sin tu ayuda ni la de ellos, no hubiera podido ser lo que soy ahora.

A mis padres, Femy y Feli, por todo el esfuerzo que hicieron para verme convertida en una persona de bien y sobre todo una profesionista, gracias por su amor, por todos y cada uno de sus consejos, sus regaños y halagos, gracias también por todos sus desvelos y momentos de angustia en espera de éxito a la entrega de mis trabajos escolares y exámenes, los amo.

A mi hermana que adoro Lore, porque siempre estás ahí para apoyarme en todos los aspectos de mi vida siendo una pieza fundamental en ella, gracias también por tus palabras de aliento para culminar este proyecto.

A mi abuelita Placi, por tus desvelos junto conmigo durante la carrera, por tus consejos, por tus bendiciones y oraciones para que siempre tenga éxito en mi vida.

A mis tíos Tere, Lolo, Toño y Chelo, por impulsarme a ser una persona exitosa desde mi niñez, por todo su apoyo, los quiero.

A mis primos, Teo, Sandy, Oscar, Katy, Toño y Jorge, porque con sus logros y éxitos me motivaron a llegar hasta esta etapa de mi vida, gracias.

Al amor de mi vida Any, por apoyarme y ser parte también de este maravilloso proyecto, por permitir que juntos culminemos una etapa de gran importancia en nuestras vidas. Gracias por tu amor, tolerancia y fortaleza, te amo.

A la familia Báez Escribano, gracias por alentarme para culminar este proyecto.

A la familia Fuerte Castro, por siempre brindarme su apoyo incondicional en cualquier momento de mi vida, saben que los considero parte de mi familia, los quiero muchísimo.

A mis jefas, Imelda Santos, María del Carmen Villarruel y Elizabeth Huerta, por apoyarme durante el desarrollo y culminación de mi tesis, brindándome todo su apoyo para que este proyecto se hiciera realidad.

Al ingeniero Carlos Vela, por ser un ejemplo para mí, gracias por brindarme todos sus consejos y conocimientos.

A Paty Fuerte, por ser mi gran amiga, por apoyarme y estar conmigo en todo momento, te adoro.

A mis amigos de la Facultad de Ingeniería, Paty, Leo, Faty, Ruth, Toño, Anabel y René por compartir todos los triunfos y las derrotas, así como desvelos durante la carrera y siempre ser un gran apoyo para terminar este proyecto de tesis.

A mis amigos del trabajo, Ceci, Ulises, Carmelita, Benjamín, Oli, Toñita, Susi y Gustavo, por su apoyo y estimulación para no dejar inconcluso este último paso de mi carrera, mi titulación.

**Carolina**



## ÍNDICE

<b>INTRODUCCIÓN</b>	<b>8</b>
<b>CAPÍTULO 1. ENTORNO INFÓRMATICO DENTIZT@</b>	<b>11</b>
1.1 Definiciones.	11
1.2 Software.	16
1.2.1 Sistema Operativo.	17
1.2.1.1 Dos de los Sistemas Operativos Más Conocidos: Windows y Linux.	18
1.2.2 Lenguaje de Programación.	23
1.2.2.1 Lenguajes de Programación para el Desarrollo de Sistemas Web.	24
1.2.3 Bases de Datos.	28
1.2.3.1 Modelos de Bases de Datos.	29
1.2.3.2 Sistema Manejador de Bases de Datos (DBMS).	35
1.2.3.3 Principales Manejadores de Base de Datos.	36
1.2.4 Servidores.	38
1.2.4.1 Servidores HTTP.	39
1.3 Metodología de Ingeniería de Programación.	41
1.3.1 Ciclo de Vida del Software.	42
1.4 Internet.	48
1.5 Seguridad Informática	49
1.5.1 Objetivos de la Seguridad Informática.	50
<b>CAPÍTULO 2. ANÁLISIS, DISEÑO Y OPTIMIZACIÓN DEL SISTEMA</b>	<b>51</b>
2.1 Desarrollo Orientado a Objetos.	52
2.1.1 Enfoque General de la Metodología.	54
2.2 Lenguaje Unificado de Modelado (UML). Una técnica para la especificación del Sistemas en todas sus fases.	56
2.2.1 Diagramas de Casos de Uso.	57



---

2.2.1.1 Estilos de Línea para los Conectores.	59
2.2.2 Diagramas de Actividades.	60
2.2.3 Diagramas de Clases.	64
2.2.4 Diagramas de Interacción.	64
2.2.5 Diagramas de Secuencia.	64
2.2.6 Diagramas de Colaboración.	66
2.2.7 Diagramas de Estado.	67
2.2.8 Diagrama de Componentes.	68
2.3 Fase de Planeación y Especificación de Requisitos.	69
2.3.1 Antecedentes del Sistema Web Dentizt@	69
2.3.1.1 Antecedentes Académicos del Proyecto.	71
2.3.1.2 Problemática del Sistema Existente.	72
2.3.1.3 Características de los Módulos Existentes.	73
2.3.2 Requisitos del Sistema.	75
2.3.3 Casos de Uso.	78
2.3.3.1 Identificación de Casos de Uso.	78
2.3.4 Construcción del Modelo de Casos de Uso.	79
<b>CAPÍTULO 3. CONSTRUCCIÓN E IMPLEMENTACIÓN DEL SISTEMA DENTIZT@</b>	<b>80</b>
3.1 Fase de Construcción.	80
3.1.1 Diseño de Alto Nivel.	81
3.1.1.1 Casos de Uso Expandidos.	82
3.1.1.2 Modelo Conceptual.	87
3.1.1.3 Diagramas de Secuencia del Sistema.	90
3.1.1.3.1 Construcción de Diagramas de Secuencia del Sistema Web Dentizt@.	90
3.1.2 Diseño de Bajo Nivel.	95
3.1.2.1 Diagramas de Interacción.	95

---



3.1.2.2 Diagramas de Colaboración.	96
3.1.2.3 Diagrama de Clases de Diseño.	99
3.1.2.3.1 Construcción de un Diagrama de Clases de Diseño.	100
3.1.3 Implementación y Puesta a Punto del Sistema Web Dentizt@.	101
3.1.3.1 Creación del Sistema en el Lenguaje de Programación.	101
3.1.3.2 Construcción de la Base de Datos.	107
3.1.4 Pruebas.	112
3.2 Fase de Instalación.	117
3.2.1 Especificación del Servidor.	117
<b>CAPÍTULO 4. EVALUACIÓN DEL SISTEMA WEB DENTIZT@.</b>	<b>128</b>
4.1 Pruebas y Resultados	128
<b>CONCLUSIONES</b>	<b>132</b>
<b>ANEXO. Código Fuente</b>	<b>134</b>
<b>GLOSARIO</b>	<b>144</b>
<b>BIBLIOGRAFÍA</b>	<b>163</b>
<b>REFERENCIAS</b>	<b>163</b>
<b>MESOGRAFÍA</b>	<b>166</b>



## INTRODUCCIÓN

Varios han sido los cambios que a finales del siglo pasado y hasta nuestros días se han producido en ámbitos tecnológicos y culturales, trayendo consigo transformaciones aceleradas en nuestra sociedad. Esta evolución ha producido un aprovechamiento de los avances tecnológicos en todos los ámbitos, particularmente en la implementación de tecnologías de información y comunicación (TIC's), que a partir del surgimiento de Internet ha motivado una interconexión global, trayendo consigo flujos de información cada vez más intensos y constantes.

Lo anterior ha hecho posible la creación y el uso de espacios virtuales, conformados en -sitios donde se almacena información de todo tipo, que han ido evolucionando hacia la especialización de diversos temas, en donde personas de cualquier índole interactúan entre sí, creando consecuentemente una comunicación fluida cada vez mayor y convergiendo en nuevas formas de aprovechamiento de los recursos telemáticos, como el desarrollo de sistemas computacionales que han beneficiado en las últimas tres décadas a la educación tanto presencial como a distancia.

En la educación superior, diversas han sido las instituciones educativas que han tomado esta alternativa; sin embargo uno de los errores más recurrente en estos sistemas es que son pensados y desarrollados para gente con conocimientos muy avanzados en computación, lo que ha provocado un desaprovechamiento de estas herramientas, dejándolas únicamente en un estado potencial sin abarcar el cien por ciento de su uso.

En nuestro país, una precursora en el diseño y desarrollo de sistemas cómputo para la educación, es la Universidad Nacional Autónoma de México de la cual se desprende el presente trabajo que tiene como objetivo optimizar el Sistema Web





Dentizt@ cuyo fin es permitir al usuario la administración de un Sistema Web por medio de una interfaz directa, sencilla, segura y de fácil acceso.

Esta propuesta tiene como objetivo desarrollar un sistema informático que permita realizar su administración a través de una interfaz desde la cual, cualquier persona ajena al medio informático pueda acceder.

La estructura de este trabajo de tesis se compone de cuatro capítulos, que van desarrollando el tema de manera gradual para que cualquier lector obtenga al final un panorama global de todo lo que involucra este Sistema Web y la metodología adoptada para su desarrollo.

El primer capítulo *Entorno Informático Dentizt@*, proporciona las bases y conceptos fundamentales de la Ingeniería de Sistemas, permitiendo conocer de manera general y puntual todos aquellos términos que serán necesarios durante el desarrollo de este trabajo y que se requieren para el fácil entendimiento de los diversos lectores y usuarios del Sistema Web.

En el segundo capítulo, que lleva por nombre *Análisis, Diseño y Optimización del Sistema* se realiza el análisis del Sistema Web original, con el fin de presentar la propuesta que dará pie a la Reingeniería y Optimización del Sistema Web Dentizt@. Se detalla de manera esencial los antecedentes, describiendo desde la primera versión hasta el estado actual, donde se retomó el sistema. Además se definen los requisitos que el Sistema necesita para su mejor funcionamiento y aprovechamiento. Asimismo se realiza la optimización del sistema utilizando una metodología para el Desarrollo Orientado a Objetos que utiliza como herramienta el Lenguaje Unificado de Modelado, el cual permite la definición del sistema en todas sus fases y presenta al lector un panorama detallado del software.

El tercer capítulo nombrado *Construcción e Implementación del Sistema Dentizt@*, detalla el modelado y construcción de cada uno de los módulos que el sistema



contiene, dando una descripción de las características que en el desarrollo se obtuvieron. Desde la interfaz que presenta el sistema al usuario final, pasando por las especificaciones del equipo de cómputo que fungirá como servidor, hasta la construcción de la Base de Datos. Para finalmente llevarlo al Lenguaje de Programación y ponerlo en marcha para su uso.

En el capítulo cuarto llamado *Evaluación del Sistema Web Dentizt@*; se refiere a los resultados arrojados en la implementación y evaluación del usuario del software con una carga mediana de trabajo, en donde se pone a prueba su funcionalidad.

Por último se presentan las conclusiones del proyecto y un anexo que forma parte complementaria en el desarrollo de este proyecto de tesis que contiene el código desarrollado en cada uno de los módulos.

Además, se agrega un glosario que contiene todos aquellos términos que pueden ayudar a un mejor entendimiento de este trabajo, dando una breve pero directa explicación de los conceptos, lo cual sirve de apoyo para la lectura y comprensión.

Con estos capítulos se pretende explicar la situación, desarrollo y aplicación del Sistema Web Dentizt@, haciéndolo funcional, útil y práctico a los usuarios para el cual fue creado, además de fomentar el uso de las tecnologías informáticas.



## CAPÍTULO 1. ENTORNO INFORMÁTICO DENTIZT@

El uso de términos tecnológicos, que parecen de lo más común para los profesionales de las tecnologías, debe ser complicado para las personas que no tienen mucha idea del tema y que están aprendiendo o sólo desean conocer más. Tomando en cuenta lo anterior es que se inicia aclarando los términos más frecuentes que se presentan en este trabajo.

Este capítulo se enfoca en introducir al lector en los términos más importantes que conlleva el desarrollo de un sistema web, para así facilitar la comprensión de la importancia de cada una de las fases del desarrollo y los resultados obtenidos.

### 1.1 Definiciones

Es cada vez más frecuente escuchar en el vocabulario cotidiano el término *Tecnologías de la Información y Comunicación* mejor conocido por sus siglas TIC, que se conciben como el universo de dos conjuntos, representados por las tradicionales Tecnologías de la Comunicación (TC) -constituidas principalmente por la radio, la televisión y la telefonía convencional- y por las Tecnologías de la Información (TI) caracterizadas por la digitalización de las tecnologías de registros de contenidos (informática, de las comunicaciones, telemática y de las interfases). “Las TIC son las tecnologías que se necesitan para la gestión y transformación de la información, y muy en particular el uso de computadoras y programas que permiten crear, modificar, almacenar, proteger y recuperar esa información” [1].

Tomando en cuenta las grandes ventajas que proporciona el uso de la computadora y que las TIC toman como herramienta fundamental, se han desarrollado nuevas aplicaciones siendo una de las más importantes las *Redes de Computadoras*, apoyadas por el área de las telecomunicaciones, mismas que implementaron una forma de poder comunicarse a través de estos dispositivos. Las que se definen como “la conexión entre dos o más computadoras, que



comparten algunos elementos del hardware, al igual que información, como diversos archivos de la organización” [2]. Esta aplicación ha ido evolucionando, ajustándose a las necesidades de la época para que igual sean usadas para ciertos servicios comunes, como salas de chat y mensajería instantánea, correos electrónicos, etc.

Esta evolución llevó a la implementación de la *Internet*, y con ello la explotación del uso de los recursos en red. Hecho con lo que se empezó a hablar del término *Sistema* con mayor frecuencia, que para la informática, se refiere a “cualquier colección o combinación de programas, procedimientos, datos y equipamiento utilizado en el procesamiento de información” [3]; en pocas palabras, es la síntesis de hardware y software.

En estos términos los “*Sistemas de Información* se definen como un conjunto de funciones o componentes interrelacionados que obtienen, procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en una organización. Igualmente apoya la coordinación, análisis de problemas y visualización de aspectos complejos, entre otros aspectos” [4].

Con la implementación de los sistemas de información comenzaron algunos problemas. No bastaba con solo ponerlos en marcha, pues con el paso del tiempo se tenía la necesidad de renovarlos, darles mantenimiento a las bases de datos y todo el sistema en general para actualizarlo de acuerdo a ciertas necesidades. Desgraciadamente hasta entonces muchos de los desarrolladores de sistemas no tenían una forma estructurada de hacerlo, por lo que la tarea para alguien ajeno al realizador original era verdaderamente difícil. Era obvio que se tenía que dar una solución a esto, por lo que se desarrolló una rama de la ingeniería llamada *Ingeniería de Software*, la cual se encarga desde el principio de un proyecto de software, de crear y mantener las aplicaciones de software empleando tecnologías



y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos.

Con esto se evitó que el procedimiento en la construcción de un sistema fuera exclusivo de un solo desarrollador y propició una metodología ordenada en la forma de desarrollar sistemas entre los diversos desarrolladores, dando paso a que cualquier otro desarrollador pudiera en cualquier momento realizar alguna modificación, con el único hecho de consultar los papeles de trabajo del desarrollador original.

La ingeniería del software se centra en los pasos de planificación y diseño del programa, mientras que antiguamente la realización de un programa consistía únicamente en escribir el código. Por lo que, el uso de técnicas que proporciona la Ingeniería de Software en el diseño de sistemas computacionales ha sido una ayuda substancial para los desarrolladores, lo que ha favorecido en tiempo y costos a empresas e instituciones.

Sin embargo, aún en la actualidad se siguen utilizando prácticas en el desarrollo de sistemas sin ninguna base metodológica, dando como resultado que los sistemas tengan un periodo de utilidad corto, lo que ha tenido impacto entre los desarrolladores y los clientes que solicitan una mejora de sus sistemas, trayendo como resultado la implementación de lo que hoy en día se conoce como *Reingeniería de Software*. Contrario a lo que se piensa, la reingeniería no toma algo existente para continuar con su desarrollo, ésta se usa en aquellos sistemas en donde es complicado su mantenimiento pero que por sus funciones es importante que se mantengan con vida, dejando a un lado lo ya existente, tomando únicamente la esencia, en términos coloquiales el "¿qué hace?", ya que para el "¿cómo?" se usarán nuevos procedimientos que permitan la realización del trabajo de manera más eficiente. Esta cadena de procesos trae consigo un rediseño del sistema, es decir un análisis profundo, para que los sistemas realicen



el trabajo programado de una manera diferente, que implique bajos costos tanto en desarrollo como en mantenimiento.

Hay que recalcar que el uso de la reingeniería, además de que en la mayoría de las ocasiones se convierte en la única alternativa para aquellos sistemas en los que resulta difícil su mantenimiento, si se implementa de manera adecuada traerá beneficios enormes y mejoras substanciales.

Respecto a aquellos conceptos que se refieren al ambiente que rodea a un sistema y que comúnmente son confundidos por su similitud en su definición, se muestran a continuación.

El término *Entorno*, que en el ámbito informático es un “conjunto de software para ofrecer al usuario de una computadora un ambiente amigable y cómodo” [5].

De importancia es diferenciar lo anterior con el concepto de *Plataforma* la cual “es la base, ya sea de hardware o software, sobre el cual un programa puede ejecutarse. Ejemplos típicos incluyen: arquitectura de hardware, sistema operativo, lenguajes de programación y sus librerías de tiempo de ejecución” [6].

A diferencia de las anteriores el concepto de *Ambiente* se refiere al tipo de presentación con que interactúa el usuario, y queda definido por el hardware y software usado en la PC.

En software, el término *Interfaz de Usuario* se refiere a la “parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario” [7]. Metafóricamente se entiende la Interfaz como conversación entre el usuario y el sistema.



Otro concepto que es importante detallar es el término *Web*, definido como el “sistema lógico de acceso y búsqueda de la información disponible en Internet, cuyas unidades informativas son las páginas web” [8].

Las *Páginas Web*, “son una fuente de información adaptada para la World Wide Web, accesible mediante un navegador de Internet. Ésta información se presenta generalmente en formato HTML (Lenguaje de Marcas de Hipertexto) y puede contener hiperenlaces a otras páginas web, constituyendo la red enlazada de la World Wide Web” [9].

A partir de la creación de estas páginas podemos ir construyendo algo más amplio que se denomina, *Sitio Web*, el cual “es un conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet” [10]. Un sitio está típicamente dedicado a algún tema particular o propósito. Cualquier sitio web puede contener un hiperenlace a cualquier otro sitio web, de manera que la distinción entre sitios individuales, percibido por el usuario, puede ser a veces borroso.

No debemos confundir sitio web con página web, esta última es solo un archivo html, y muchas veces las empresas o personas tienen más de un archivo publicados en Internet, lo que se constituye en sitio web.

Mientras un *Portal de Internet* es “un sitio web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente están dirigidos a resolver necesidades específicas de un grupo de personas o de acceso a la información y servicios a una institución pública o privada” [11].



Los portales normalmente tienen programación que requiere muchos recursos computacionales y por su alto tráfico generalmente se hospedan en servidores de Internet dedicados.

Como se puede observar, el mundo de las Tecnologías de la Información y Comunicaciones está relacionado con gran variedad de términos, por lo que no es nada difícil perderse en ellos y llegar a la confusión en su uso.

## 1.2 Software

“De acuerdo con el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), el *Software* es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación” [12].

Se puede entender como una serie de instrucciones efectuadas para el funcionamiento del hardware de las computadoras, comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica.

El software tiene las siguientes funciones:

- Administrar los recursos de cómputo.
- Proporcionar las herramientas para optimizar estos recursos.
- Actuar como intermediario entre el usuario y la información almacenada.

Se clasifica comúnmente en tres principales tipos.

*Software de Aplicación*, el cual es un conjunto de instrucciones de computadora escritas con un lenguaje de programación, las cuales dirigen al hardware para que efectúe actividades específicas de procesamiento de datos y de información que proporcionan funcionalidad al usuario. Los programas de aplicación satisfacen una necesidad.





*Software de Sistemas*, el cual actúa como intermediario entre el hardware de cómputo y los programas de aplicación. Es un conjunto de programas que administran los recursos de la computadora.

*Software de Usuario Final*, el cual permite el desarrollo de algunas aplicaciones directamente por los usuarios finales, el software del usuario final con frecuencia tiene que trabajar a través del software de aplicación y finalmente a través del software del sistema.

### 1.2.1 Sistema Operativo

“El *Sistema Operativo* es el programa o software más importante de un equipo de cómputo, porque contiene un conjunto de programas que administran los recursos de la computadora y que ayuda en el desarrollo y ejecución del software” [13].

Los sistemas operativos realizan tareas básicas, tales como reconocimiento de la conexión del teclado, enviar la información a la pantalla, no perder de vista archivos y directorios en el disco, y controlar los dispositivos periféricos tales como impresoras, escáner, etc. En sistemas grandes, el sistema operativo tiene incluso mayor responsabilidad y poder, se asegura de que los programas y usuarios que están funcionando al mismo tiempo no interfieran entre ellos. El sistema operativo también es responsable de la seguridad, certificando que los usuarios no autorizados no tengan acceso al sistema.

Los sistemas operativos proporcionan una plataforma de software encima de la cual otros programas, llamados aplicaciones, puedan funcionar. Las aplicaciones se programan para que funcionen encima de un sistema operativo particular, por tanto, la elección del sistema operativo determina en gran medida las aplicaciones que pueden ser utilizadas. En cuanto a los usuarios, normalmente interactúan con el sistema operativo a través de un sistema de comandos.



### 1.2.1.1 Dos de los Sistemas Operativos Más Conocidos: Windows y Linux

En estos días, los sistemas operativos modernos responden a nuevos desarrollos del hardware y nuevas aplicaciones. En los campos de aplicación que han influido en el diseño de los sistemas operativos están las aplicaciones multimedia, el acceso a Internet y páginas Web y la ejecución cliente/servidor.

Dos son los sistemas operativos más importantes por su uso y popularidad: Windows y Linux. Ambos son los más usados comúnmente por los desarrolladores, por tal motivo a continuación se presentan sus principales características.

#### ➤ Sistema Operativo Windows

Uno de los pasos más importantes que revolucionó los sistemas operativos de Microsoft fue el diseño y creación de un sistema operativo extensible, portable, fiable, adaptable, robusto, seguro y compatible con sus versiones anteriores, fue entonces que surgió los sistemas operativos Windows NT (Nueva Tecnología). Esta familia de sistemas operativos de Microsoft está constituida por versiones como Windows Vista, Windows Server 2003, Windows XP, Windows 2000 y Windows NT.

Windows tiene una arquitectura modular, compuesta por una serie de componentes separados donde cada cual es responsable de sus funciones y brindan servicios a otros componentes. Esta arquitectura es del tipo cliente – servidor ya que los programas de aplicación son contemplados por el sistema operativo como si fueran clientes a los que hay que servir, y para lo cual viene equipado con distintas entidades servidoras (Figura 1.1).

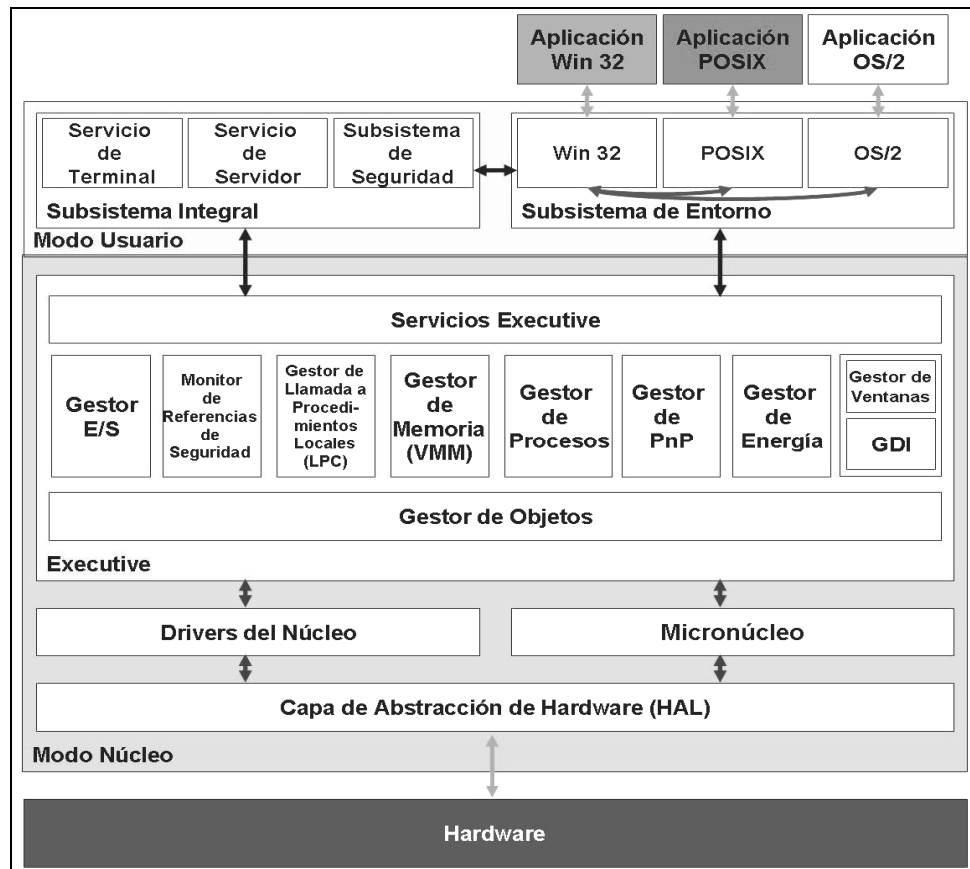


Figura 1.1 Arquitectura del Sistema Operativo Windows

Uno de las características que Windows comparte con el resto de los Sistemas Operativos avanzados es la división de tareas del Sistema Operativo en múltiples categorías, las cuales están asociadas a los modos actuales soportados por los microprocesadores. Estos modos proporcionan a los programas que corren dentro de ellos diferentes niveles de privilegios para acceder al hardware o a otros programas que están corriendo en el sistema.

La arquitectura de Windows es altamente modular y se basa en dos capas principales:

- *Modo Usuario:* Cuyos programas y subsistemas están limitados a los recursos del sistema a los que tienen acceso.



- *Modo Núcleo*: Tiene acceso total a la memoria del sistema y los dispositivos externos.

“Windows es un sistema que aprovecha la potencia de los procesadores, ha sido diseñado para adaptarse a las nuevas tecnologías, ofrece compatibilidad con varias plataformas (OS/2, Unix y versiones anteriores a el mismo), soporta el multiprocesamiento simétrico, buen rendimiento y conectividad, seguridad y al no estar encasillado en ningún modelo estándar de Sistema Operativo, que tiene la capacidad de combinar las ventajas del modelo cliente/servidor, puede correr además sobre múltiples arquitecturas con un mínimo de cambios, permite que varios procesos sean ejecutados simultáneamente en varios procesadores y estos no se apropien de recursos del sistema por tiempo indefinido, sino por tratamiento del sistema” [14].

### ➤ **Sistema Operativo Linux**

El sistema operativo Linux es un sistema operativo libre, lo que quiere decir que ha sido desarrollado por miles de usuarios de computadores a través del mundo, se genera inspirándose en dos sistemas operativos, el sistema abierto UNIX del cual se toman sus características, especificaciones y funcionamiento y el sistema educativo Minix del cual se toma la estructura y código del núcleo.

La parte central de Linux (conocida como núcleo o kernel) se distribuye a través de la Licencia Pública General GNU, lo que básicamente significa que puede ser copiado libremente, cambiado y distribuido. A todo el sistema se le da el nombre de GNU/Linux, que contiene el núcleo más las otras capas del sistema operativo y utilidades.

Cuenta con un amplio y robusto soporte para comunicaciones y redes, lo cual hace que sea una opción atractiva tanto para empresas como para usuarios individuales. Además da soporte a una amplia variedad de hardware y se puede



correr en una multitud de plataformas: computadoras personales (PC's) convencionales, computadoras Macintosh, así como costosas estaciones de trabajo. Cada usuario de un sistema Linux tiene su propia interfaz de usuario o Shell.

Es Multitarea porque presenta las tareas de forma intercalada para que se ejecuten varias simultáneamente. Por lo tanto en Linux es posible ejecutar varios programas a la vez sin necesidad de tener que parar la ejecución de cada aplicación.

Se dice que Linux es Multiusuario ya que varios usuarios pueden acceder a las aplicaciones y recursos del sistema Linux al mismo tiempo. Y, por supuesto, cada uno de ellos puede ejecutar varios programas a la vez (multitarea). Otra de las características referentes a este tema es que Linux es multiplataforma. Fue diseñada para plataforma Intel pero ha sido fácilmente exportado a diversos tipos de sistema.

Respecto a la seguridad de Linux, el sistema de contraseñas que protege el acceso al sistema se basa en el algoritmo DES (Data Encryption Standard), el más probado de los algoritmos de seguridad. DES cifra bloques de 64 bits, mediante permutación y sustitución y usando una clave de 64 bits, de los que 8 son de paridad (esto es, en realidad usa 56 bits), produciendo así 64 bits cifrados. Pero claro, por muy bueno que sea el algoritmo, si después permitimos a sus usuarios poner como contraseña su nombre de usuario, de nada servirá la contraseña y todos sus esfuerzos.

En Linux los controladores son tratados de forma independiente al núcleo del sistema y por lo tanto, se podrán añadir tantos controladores como dispositivos nuevos se vayan añadiendo a la computadora. Por otra parte todos los dispositivos son tratados de igual forma, y gracias a ello se podrá redirigir datos de la misma manera al disco duro o a la impresora.



Cuando se trabaja con Linux se está ante un sistema operativo orientado al trabajo de redes de computadoras. Linux dispone de varios protocolos como PPP, SLIP, TCP/IP, PLIP, etc., para la transferencia de archivos entre plataforma. Tiene a su disposición multitud de aplicaciones de libre distribución que permiten navegar a través de Internet y enviar y recibir correo electrónico.

Con Linux se puede montar un servidor en nuestra propia casa sin tener que pagar las enormes cantidades de dinero que piden otros sistemas. Linux es distribuido mediante una serie de distribuciones como RedHat, Slackware, Debían, etc., las cuales se diferencian por su método de instalación y por los paquetes (software) que viene incluido.

De manera funcional la arquitectura de Linux (Figura 1.2) para la interacción o solicitud de un servicio por parte de un programa de usuario, sigue los siguientes pasos:

1. El usuario en su programa solicita una función de la librería libc.
2. La función de libc, hace una llamada al núcleo a través de una llamada (syscall).
3. El núcleo recibe esa llamada mediante una función (System Call).
4. El núcleo redirige esta llamada a los módulos implicados que a su vez llaman a otros hasta llegar al hardware y realizar la solicitud del usuario.
5. El núcleo hace llegar a la aplicación del usuario el resultado de su solicitud.

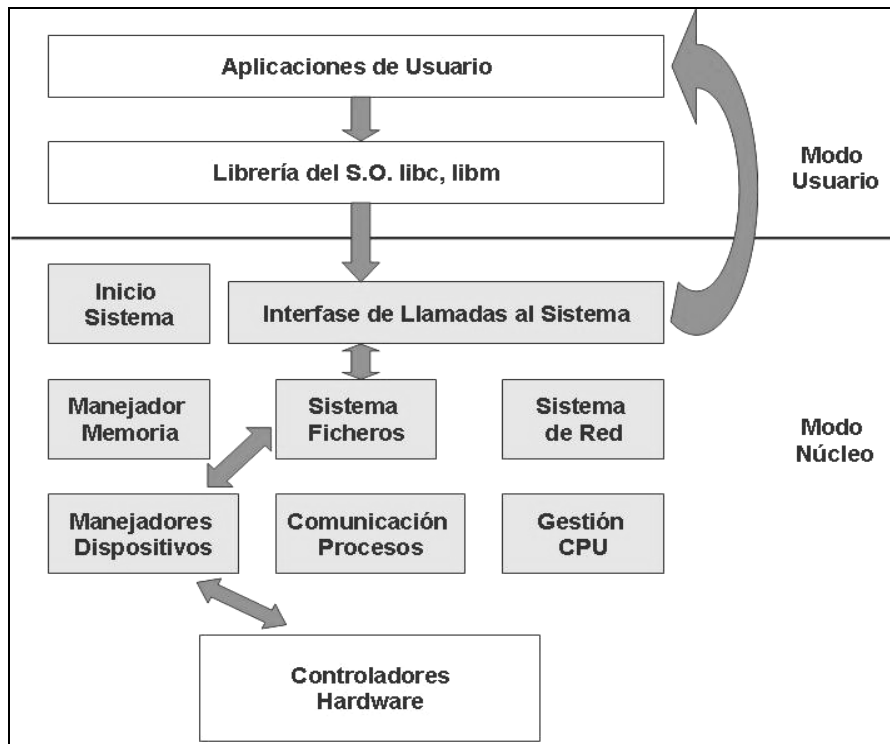


Figura 1.2 Arquitectura del Sistema Operativo Linux

En términos generales podemos decir que Linux dispone de varios tipos de sistema de archivos para poder acceder a archivos en otras plataformas. Incluye un entorno gráfico X Window (Interface gráfica estándar para máquinas UNIX). Está orientado al trabajo en red, con todo tipo de facilidades como correo electrónico por ejemplo. Posee cada vez más software de libre distribución, por lo que actualmente es preferido por la mayoría de los desarrolladores de software.

### 1.2.2 Lenguaje de Programación

Un *Lenguaje de programación* es un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura, el significado de sus elementos y expresiones, y es utilizado para controlar el comportamiento físico y lógico de una máquina.



Una característica relevante de los lenguajes de programación es precisamente que más de un programador puedan tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa.

### 1.2.2.1. Lenguajes de Programación para el Desarrollo de Sistemas Web

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a solucionar. Esto dio lugar a desarrollar lenguajes de programación dinámicos para la Web, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

Estos lenguajes a pesar de no ser visibles en su uso, son importantes, pues deben satisfacer las necesidades del propio sistema y al desarrollador debe facilitarle la programación, lo que tendrá como resultado la satisfacción del usuario final.

A continuación se da un panorama de los diferentes Lenguajes de Programación para el desarrollo de sistemas web.

#### ➤ Lenguaje JavaScript

Es un lenguaje interpretado, no requiere compilación. Utilizado principalmente en páginas Web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias.

Entre sus ventajas se destaca:

- Es un lenguaje de scripting (es decir, diseñado para ser ejecutado por medio de un intérprete) seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.





- El código Javascript se ejecuta en el cliente.

Sus principales desventajas son:

- Tiene código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (Cross Site Scripting). Sucede cuando hay un mensaje o una ruta en la URL del navegador o en una cookie, lo que permite meter comandos Javascript solo modificando una URL.

#### ➤ **Lenguaje ASP.NET**

Es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, Sitios Web. El ASP.NET fue desarrollado para resolver las limitantes que brindaba su antecesor ASP. Creado para desarrollar Web sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#.

Presenta ventajas como:

- Es completamente orientado a objetos.
- Tiene controles de usuario y personalizados.
- Presenta división entre la capa de aplicación o diseño y el código.
- Facilita el mantenimiento de grandes aplicaciones.
- Presenta incremento de velocidad de respuesta del servidor.
- Brinda mayor velocidad.
- Otorga mayor seguridad.

Su principal desventaja radica en:

- Su uso genera mayor consumo de recursos.



## ➤ Lenguaje Perl

Perl está basado en un estilo de bloques como los del C, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script. Es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo Web, programación en red, desarrollo de GUI (Interfaz Gráfica de Usuario) y más.

Sus principales características son:

- Es fácil de usar.
- Soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional.
- Tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles.

Tiene las siguientes ventajas:

- La gran ventaja, es la capacidad de crear aplicaciones CGI (Interfaz de Entrada Común, en inglés Common Gateway Interface) es una importante tecnología Web que permite a un cliente solicitar datos de un programa ejecutado en un servidor Web). Con esto se pueden crear programas amigables para los usuarios con una interfaz muy vistosa para el usuario.

Presenta ciertas desventajas como son:

- Puede ser más lento que otros lenguajes haciendo lo mismo, porque tiene que compilar el código fuente cada vez que corre el programa.
- El mal uso puede llegar a crear programas con grandes brechas de seguridad.



## ➤ Lenguaje PHP

Es un lenguaje de programación utilizado para la creación de Sitios Web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”. PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de Páginas Web Dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS (Servidores Web) con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

Entre sus ventajas destacan las siguientes:

- Es muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objetos, clases y herencias.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Presenta capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.



Sus principales desventajas son:

- Se necesita instalar un servidor Web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la Modularización.
- Dificulta la organización por capas de la aplicación.

### 1.2.3 Bases de Datos

El uso de sistemas de información requiere el almacenamiento de enormes cantidades de datos, ya sea para el uso mismo del sistema, para generar resultados o para compartir dicha información con otros sistemas. Lo anterior es posible gracias a un sistema de base de datos cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base a sus peticiones y necesidades.

“Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular” [15].

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.



- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

### 1.2.3.1 Modelos de Bases de Datos

Un Modelo de Datos es básicamente una descripción de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son objetos físicos, son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Los Modelos de Datos más conocidos son:

#### ➤ Modelo Entidad-Relación (E-R)

Denominado por sus siglas como: E-R; representa a la realidad a través de *entidades*, que son objetos que existen y que se distinguen de otros por sus características. Las entidades pueden ser de dos tipos:

- *Tangibles*: Son todos aquellos objetos físicos que podemos ver, tocar o sentir.
- *Intangibles*: Todos aquellos eventos u objetos conceptuales que no podemos ver, aun sabiendo que existen.

Las características de las entidades en base de datos se llaman *atributos*. A su vez una entidad se puede asociar o relacionar con más entidades a través de *relaciones*.

La estructura lógica de una base de datos puede expresarse gráficamente mediante un diagrama entidad – relación. La Tabla 1.1 muestra los componentes que conforman a los diagramas Entidad – Relación.




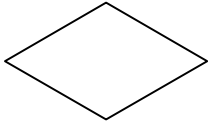
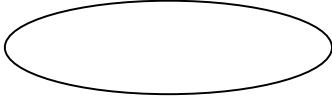

Símbolo	Descripción
	Entidad
	Relación
	Atributos
	Ligas

Tabla 1.1 Componentes de los Diagramas Entidad-Relación

Se muestra un ejemplo del Diagrama E-R (Figura 1.3), una tienda de ropa donde es necesario el control de los empleados y los artículos que se venden, estas entidades a estudiar tienen las siguientes características:

1. Empleado: Nombre, Puesto, Salario, Dirección, R.F.C.
2. Artículo: No. de inventario, Descripción, costo, talla, cantidad.

La *Relación* que existe entre estas dos entidades es la venta de artículos. Finalmente nuestro Diagrama Entidad/Relación quedaría de la siguiente forma.

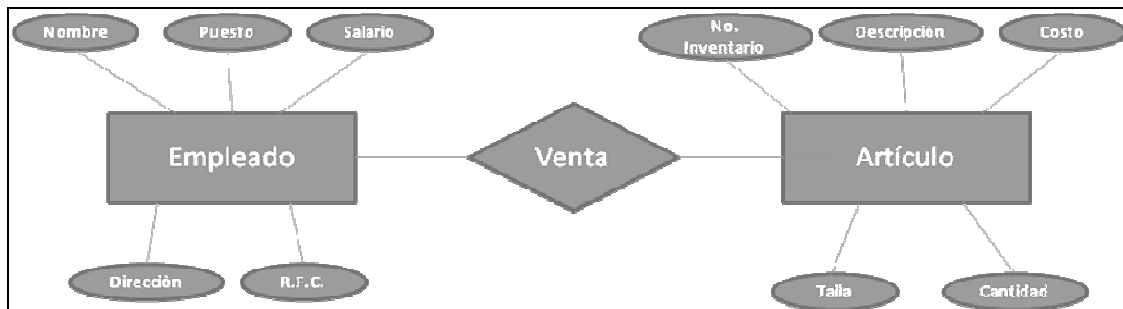


Figura 1.3 Diagrama del Modelo Entidad-Relación

### ➤ Modelo Orientado a Objetos (OO)

El modelo Orientado a Objetos es uno de los más recientes y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos, incorpora todos los conceptos importantes del paradigma de objetos:

- *Encapsulación*. Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- *Herencia*. Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- *Polimorfismo*. Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada Función) se especifica en dos partes. La interfaz de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos. La implementación de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y



argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

### ➤ Modelo Relacional

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características o atributos de cada registro.

Considerando el ejemplo de la tienda de ropa, el Diagrama del Modelo Relacional quedaría de la siguiente manera (Figura 1.4):

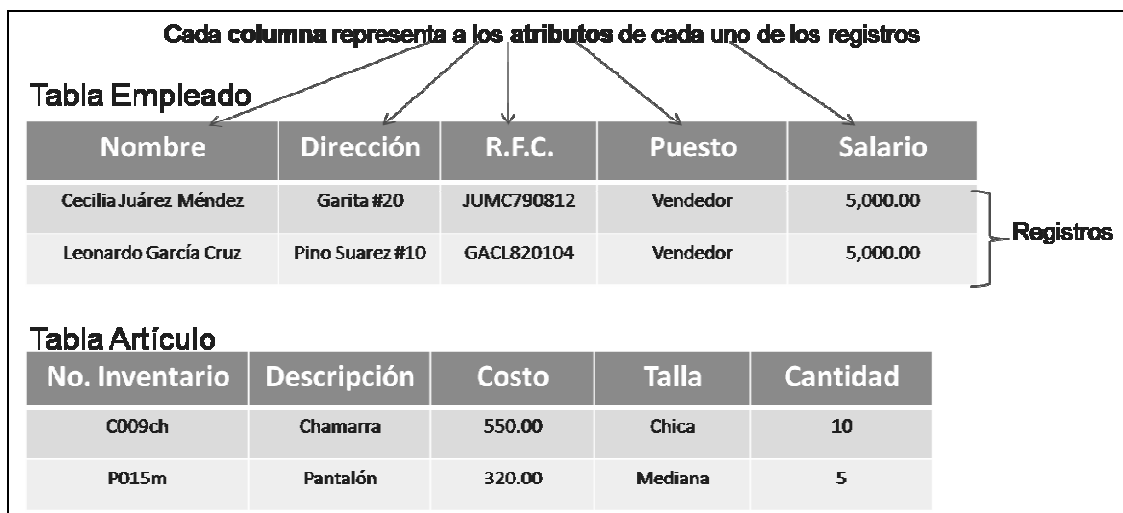


Fig. 1.4 Diagrama del Modelo Relacional

Existen dos formas de representar las relaciones en el modelo relacional; para ello es necesario definir el concepto de *llave primaria*, que es el atributo principal que identifica como única a una entidad. Por ejemplo, el RFC de un empleado se distingue de otro por que los RFC no pueden ser iguales.





La primera forma de representar las relaciones en este modelo, es definir la llave primaria en cada una de las tablas que conforman la base de datos, en el ejemplo propuesto, la llave primaria de la tabla Empleado seria el R.F.C., y la llave primaria de la tabla Artículo seria el No. Inventario. Posteriormente generar una tabla que contenga cada una de las llaves primarias elegidas, como se muestra en la Tabla 1.2 a continuación.

R.F.C.	No. Inventario
JUMC790812	C009ch
GACL820104	P015m

Tabla 1.2 Ejemplo de llaves primarias

La segunda forma es incluir en alguna de las tablas la llave principal de la otra tabla como se muestra a continuación en la Tabla 1.3.

Nombre	Dirección	R.F.C.	Puesto	Salario	No. Inventario
Cecilia Juárez Méndez	Garita #20	JUMC790812	Vendedor	5,000.00	C009ch
Leonardo García Cruz	Pino Suarez #10	GACL820104	Vendedor	5,000.00	P015m

Tabla 1.3 Ejemplo donde se incluye la Llave Principal

### ➤ Modelo de Red

Este modelo representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias. El Modelo de Red (Tabla 1.4) quedaría ejemplificado de la siguiente manera:



Cecilia Juárez Méndez	Garita #20	JUMC790812	Vendedor	5,000.00	C009ch	JUMC790812	C009ch
Leonardo García Cruz	Pino Suarez #10	GACL820104	Vendedor	5,000.00	P015m	GACL820104	P015m

Tabla 1.4 Modelo de Red

### ➤ Modelo Jerárquico

“El modelo jerárquico fue desarrollado para permitir la representación de aquellas situaciones de la vida real en las que predominan las relaciones de tipo 1:N (uno a muchos). Es un modelo muy rígido en el que las diferentes entidades de las que está compuesta una determinada situación, se organizan en niveles múltiples de acuerdo a una estricta relación *padre/hijo*, de manera que un padre puede tener más de un hijo, todos ellos localizados en el mismo nivel, y un hijo únicamente puede tener un padre situado en el nivel inmediatamente superior al suyo. Esta estricta relación *padre/hijo* implica que no puedan establecerse relaciones entre segmentos dentro de un mismo nivel” [16].

La representación gráfica de un modelo jerárquico se realiza mediante la estructura de árbol invertido, en la que el nivel superior está ocupado por una única entidad, bajo la cual se distribuyen el resto de las entidades en niveles que se van ramificando. Los diferentes niveles quedan unidos por medio de las relaciones. Las entidades se denominan en el caso particular del modelo jerárquico *segmentos*, mientras que los atributos reciben el nombre de *campos*. Los segmentos, se organizan en niveles de manera que en un mismo nivel estén todos aquellos segmentos que dependen de un segmento de nivel inmediatamente superior.

Los *segmentos*, en función de su situación en el árbol y de sus características, pueden denominarse como:

1. *Segmento Padre*: Es aquél que tiene descendientes, todos ellos localizados en el mismo nivel.



2. *Segmento Hijo*: Es aquél que depende de un segmento de nivel superior. Todos los hijos de un mismo padre están en el mismo nivel del árbol.
3. *Segmento Raíz*: El segmento raíz de una base de datos jerárquica es el padre que no tiene padre. La raíz siempre es única y ocupa el nivel superior del árbol.

El ejemplo de la tienda de ropa, quedaría expresado con el Modelo Jerárquico (Figura 1.5) de la siguiente manera:

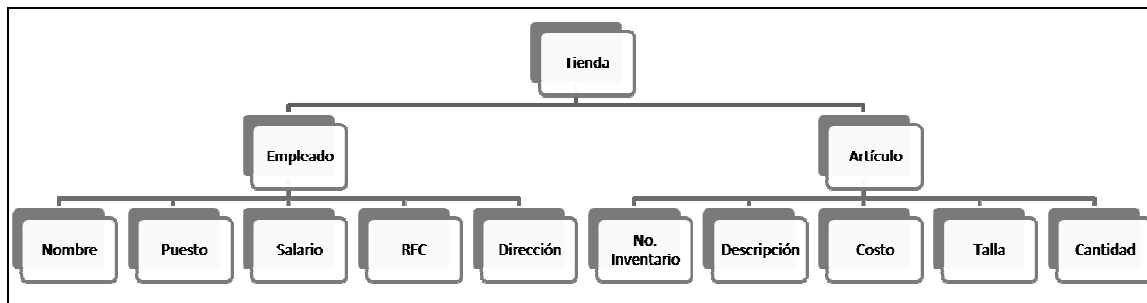


Figura 1.5 Diagrama del Modelo Jerárquico

### 1.2.3.2 Sistema Manejador de Bases de Datos (DBMS)

El DBMS se puede definir como: "El Conjunto de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad" [17].

Las funciones principales de un DBMS son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser localizados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.



- El respaldo y recuperación, que consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- El control de concurrencia que controla la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- La seguridad e integridad que cuenta con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

### 1.2.3.3 Principales Manejadores de Base de Datos

#### ➤ MySQL

*MySQL* es un sistema de gestión de bases de datos relacional, su diseño multihilo (Permite a un programa informático realizar varias tareas a la vez) le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.



6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

➤ **Oracle**

Es un sistema de gestión de base de datos relacional, desarrollado por *Oracle Corporation*. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

1. Soporte de transacciones.
2. Estabilidad.
3. Escalabilidad.
4. Soporte multiplataforma.

Las últimas versiones de *Oracle* han sido certificadas para poder trabajar bajo Linux. En la actualidad, la tecnología *Oracle* se encuentra prácticamente en todas las industrias alrededor del mundo. *Oracle* es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. *Oracle* es un sistema vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales.

➤ **PostgreSQL**

Es un sistema de gestión de base de datos relacional orientada a objetos de software libre, como muchos otros proyectos *open source*.

Dentro de sus principales características de este gestor de bases de datos se encuentran:



1. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
2. Incorpora una estructura de datos array.
3. Incorpora funciones de diversa índole: manejo de fechas, geométricas y orientadas a operaciones con redes.
4. Permite la declaración de funciones propias, así como la definición de disparadores.
5. Soporta el uso de índices, reglas y vistas.
6. Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
7. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

*PostGreSQL* es un magnífico gestor de bases de datos, capaz de competir con muchos gestores comerciales, aunque carezca de una herramienta que permita la fácil gestión de los usuarios y de las bases de datos que contenga el sistema. Por otro lado, la velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos grandes.

#### **1.2.4. Servidores**

Los Servidores suelen conocerse como una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes, en general este es el significado principal de este término.



En el mundo de la informática, existen unos equipos de cómputo conocidos genéricamente como *Servidores*, se les llama de esa manera porque son los que tienen almacenadas las páginas que visita el usuario, y se las muestra cuando éste las solicita. Estos equipos ejecutan programas que realizan algunas tareas en beneficio de otras aplicaciones llamadas clientes.

#### 1.2.4.1 Servidores HTTP

También conocidos como servidores de páginas web, ya que se encargan de alojar estas últimas.

Básicamente, “un *servidor web* sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI, seguridad SSL y páginas activas del servidor (ASP)” [18].

Estos últimos son los de mayor relevancia para los desarrolladores, los que a continuación se describen por su popularidad dentro del ámbito del desarrollo Web.

##### ➤ Servidor Cherokee

*Servidor HTTP Cherokee* es Servidor Web libre, multiplataforma, abierto bajo la licencia GPL. Apunta a ser un servidor Web bastante rápido que también soporta las funcionalidades más comunes de servidor. Está escrito completamente en C, es escalable y puede usarse como un Sistema integrado.

El servidor Cherokee puede realizar redirecciones y soporta la configuración de Servidores Virtuales así como ejecutar CGI, PHP tanto como PHPCGI o FastCGI. También soporta registro y autenticación de usuarios



Los métodos de registro son: NCSA,W3C y combinado. Los métodos de Autenticación: plain,htpasswd, htdigest y PAM.

### ➤ Servidor Apache

El *Servidor HTTP Apache* es un software libre de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual.

El servidor Apache es un software que esta estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- *Módulos Base*: Módulo con las funciones básicas del Apache
- *Módulos Multiproceso*: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones
- *Módulos Adicionales*: Cualquier otro módulo que le añada una funcionalidad al servidor.

A continuación se describen algunas de las funcionalidades del servidor apache:

1. Apache puede ejecutarse en modo híbrido multiproceso-multihebra. Esto mejora la escalabilidad para muchas configuraciones aunque no para toda.
2. Tiene la infraestructura necesaria para servir distintos protocolos.
3. Apache hace automáticamente la ordenación de módulos, lo que ofrece una mayor flexibilidad.
4. Los módulos de Apache pueden escribirse para que se comporten como filtros que actúan sobre el flujo de contenidos tal y como salen del servidor o tal y como son recibidos por el servidor.





5. Los mensajes de error que se envían a los navegadores están disponibles en diferentes idiomas, usando documentos SSI. Estos mensajes pueden personalizarse por el administrador del sitio web.

### 1.3. Metodología de Ingeniería de Programación

La Metodología para el Desarrollo de Software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Esta sistematización nos indica cómo dividir un gran proyecto en módulos más pequeños llamados etapas, y las acciones que corresponden en cada una de ellas, nos ayuda a definir entradas y salidas para cada una de las etapas y, sobre todo, normaliza el modo de administración el proyecto. Es decir, una metodología para el desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado.

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama “ciclo de vida del software” que comprende tres fases claramente definidas, que se detallan a continuación:

- *Planificación:* En esta etapa se obtienen los requerimientos del sistema y se lleva a cabo un planteamiento detallado que guíe la gestión del proyecto.
- *Implementación:* Es el conjunto de actividades que componen la realización del producto.
- *Pruebas:* Es la etapa donde el sistema ingresa en la etapa de definición, donde se presenta al cliente o usuario final, sabiendo que funciona correctamente y responde a los requerimientos solicitados por el mismo. Esta etapa es muy importante no sólo por representar la aceptación o no del proyecto por parte del cliente o usuario final, sino por las múltiples dificultades que suele presentar en la práctica, alargándose excesivamente y provocando costos no previstos.



### 1.3.1 Ciclo de Vida del Software

Existen distintos modelos de ciclo de vida, y la elección de un modelo para un determinado tipo de proyecto es realmente importante; el orden de las etapas es uno de estos puntos importantes.

#### ➤ **Modelo en Cascada**

Es un modelo base por contener un carácter metodológico que muestra rigurosamente las etapas del ciclo de vida del software, el cual presenta la característica que dichas etapas deben llevarse a cabo de manera sucesiva, de tal forma que no se comienza una etapa hasta no haber concluido la anterior, éstas a su vez requieren información de entrada, procesos y resultados.

Este modelo se constituye por una serie de etapas que contienen singulares características (Figura 1.6). En primer lugar se encuentra la *Etapa de Ingeniería de Sistemas*, en donde se obtienen y establecen los requerimientos globales del sistema a construir; posteriormente viene la *Etapa de Análisis*, en la que se recolectan las necesidades de los usuarios, permitiendo un entendimiento general del problema. Le sigue la *Etapa de Diseño*, en la cual se analiza detenidamente los requerimientos para que posteriormente se traduzcan a un lenguaje de programación. Subsecuentemente se encuentra la *Etapa de Codificación*, en la que el diseño se convierte en líneas de código de algún lenguaje de programación. Le prosigue la *Etapa de Pruebas*, que tiene la funcionalidad de realizar comprobaciones de cada uno de los módulos, probando que se ejecute cada una de las líneas de código sin ningún error. Finalmente, se encuentra la *Etapa de Mantenimiento*, que se presenta después de la entrega del sistema, realizando adecuaciones a nuevos requerimientos de los usuarios.

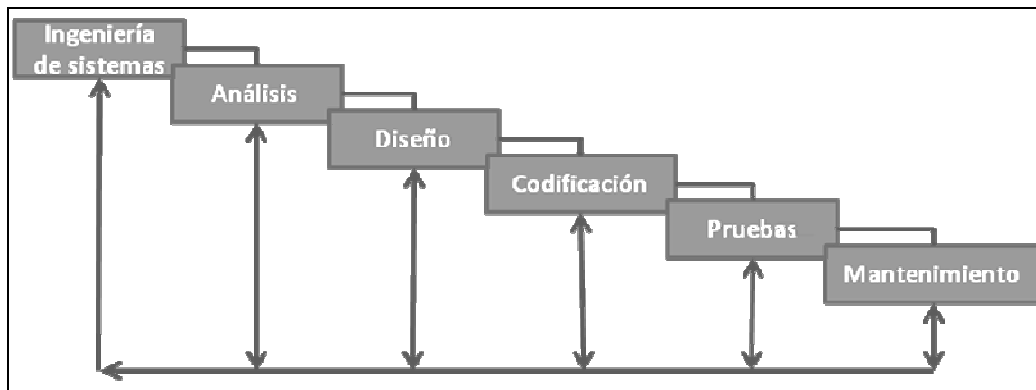


Figura 1.6 Modelo de Cascada

### ➤ Modelo de Prototipos

La importancia de este modelo radica en la iteración de la fase de análisis tantas veces como sea necesario, centrándose en el diseño rápido de aspectos del sistema que serán visibles para el cliente, conduciendo a la construcción de prototipos, el cual es evaluado por el mismo para tener una retroalimentación y poder satisfacer las necesidades exactas del cliente. Esto trajo como ventaja la mayor semejanza posible de lo que el cliente quería en su sistema en menor tiempo.

El Modelo de Prototipos o de Desarrollo Evolutivo (Figura 1.7), se rige por las siguientes etapas para su implementación:

1. Definición de objetivos globales del sistema.
2. Identificación de requisitos conocidos.
3. Identificación de áreas del esquema en donde se necesita más definición.
4. Construcción de un prototipo de baja fidelidad para clarificar los requerimientos iniciales.
5. Iterar con el equipo hasta que el cliente y desarrolladores estén de acuerdo con el acabado del prototipo en evolución.
6. Se finaliza la construcción del sistema tal como se requería.

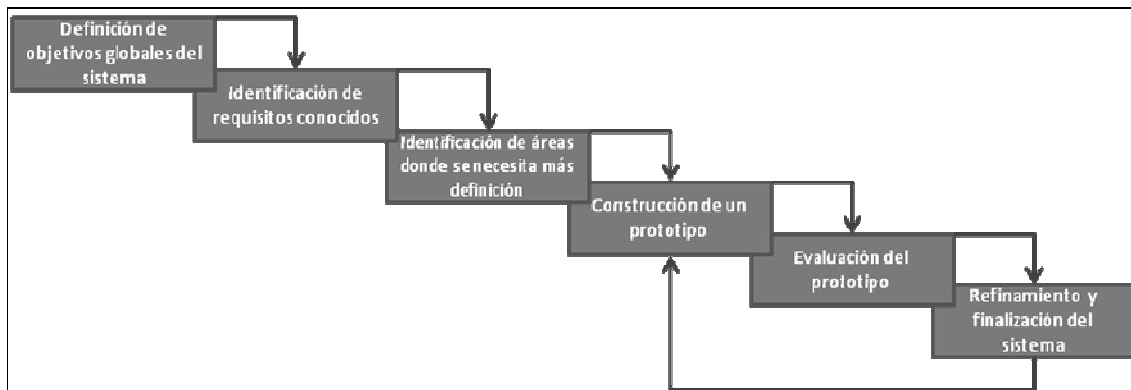


Figura 1.7 Modelo de Prototipos

### ➤ Modelo Espiral

Los modelos anteriores, representaron un punto de partida para la realización de nuevas tendencias y así los ingenieros del software se dedicaron a la tarea de perfeccionarlo gradualmente, lo cual trajo el surgimiento, de otro de los modelos más sonados en el ámbito de la Ingeniería de Software que es el Modelo Espiral (Figura 1.8); que presenta como característica fundamental el concepto de análisis de riesgo. Este modelo no es más que la unión de los modelos anteriormente citados. Se observa que las fases de este modelo se encuentran en forma de un espiral, y cada bucle es una fase. Así como los dos anteriores, este modelo presenta una serie de fases en donde se auxilian los desarrolladores para obtener resultados, teniendo como primer paso la *determinación de objetivos*, la cual es de vital importancia pues es el primer paso para empezar a construir el sistema, por tal motivo se deben tener claros los requerimientos, restricciones, etc. La segunda etapa llamada *análisis de riesgo*, considera todos y cada uno de los riesgos más probables por los cuales el sistema pueda fracasar y se consideran una o varias alternativas, lo que nos beneficia para reducir o eliminar estos fracasos en mayor porcentaje. Posteriormente se pasa a la *etapa de desarrollo, verificación y validación*, que abarca la realización del sistema, haciendo las respectivas comprobaciones. Por último se encuentra la *etapa de la planificación* representa la



parte final de nuestra implementación, ejerciendo la revisión y evaluación de todo lo anterior para saber si se prosigue con las siguientes fases.



Figura 1.8 Modelo Espiral

En respuesta a las nuevas tendencias de software, a finales de la década de los 80's y principios de los 90's, hombres como Ivar Jacobson, James Rumbaugh y Grady Booch, empezaron a definir los elementos del análisis y diseños modernos del software, lo que ahora conocemos como UML. En estos años, las guerras sobre la notación del modelado estaban plenamente entabladas, con diferentes círculos de desarrolladores apoyando a Jacobson, Rumbaugh o Booch. Estos tres personajes, cada uno por su lado, usaron símbolos y reglas diferentes para crear modelos. [25]

A mediados de la década de los 90's, se fusionaron los elementos de modelado de Rumbaugh (OMT, Object Modeling Technique), Booch (Método de Booch) y Jacobson (OOSE, Object Oriented Software Engineering), para formar el Proceso Unificado de Modelado, estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. Poco tiempo después, se eliminó Proceso de la especificación del modelado y nació el UML (Unified Modeling Language –



Lenguaje Unificado de Modelado). En conclusión, UML unifica esencialmente los métodos de Booch, Rumbaugh (OMT) y Jacobson.

Aprovechando el uso de este lenguaje como herramienta para abordar el *Desarrollo Orientado a Objetos*, Craig Larman, un especialista en el desarrollo iterativo e incremental y en el análisis, diseño y modelado orientado a objetos, propone un método sistemático que abarca todo el abanico de tamaños y complejidades. El cual se ajusta a un ciclo de vida iterativo e incremental dirigido por casos de uso.

Su método de Desarrollo Orientado a Objetos se sirve de los modelos de UML y se decidió usarlo para el desarrollo del Sistema Web Dentizt@ ya que no fija una metodología estricta, sino que define una serie de actividades que pueden realizarse en cada fase, las cuales deben adaptarse según las condiciones del proyecto que se esté llevando a cabo. Se ha escogido seguir este proceso debido a que aplica los últimos avances en Ingeniería del Software, ya que adopta un enfoque eminentemente práctico, aportando soluciones a las principales dudas y/o problemas con los que se enfrentan los desarrolladores. Su mayor aportación consiste en atar los cabos sueltos que otros métodos dejan.

La notación que se usa para los distintos modelos, es la proporcionada por UML. El uso de UML permite integrar con mayor facilidad en el equipo de desarrollo a nuevos miembros y compartir con otros equipos la documentación. Además la incursión de UML, trae como ventajas:

- Mejores tiempos totales de desarrollo.
- Modelar sistemas utilizando conceptos orientados a objetos.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.



En general, UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Además, nos permite: Visualizar, Especificar, Construir y Documentar.

En la implementación del método se va a abarcar todo el ciclo de vida, empezando por los requisitos y acabando en el sistema funcionando, proporcionando así una visión completa y coherente de la producción de sistemas software. El enfoque que toma es el de un ciclo de vida iterativo incremental, el cual permite una gran flexibilidad a la hora de adaptarlo a un proyecto y a un equipo de desarrollo específicos. El ciclo de vida está dirigido por casos de uso, es decir, por la funcionalidad que ofrece el sistema a los futuros usuarios del mismo. Así no se pierde de vista la motivación principal que debería estar en cualquier proceso de construcción de software: el resolver una necesidad del usuario/cliente.

El proceso está formado por una serie de actividades y subactividades, cuya realización se va repitiendo en los tiempos aplicados a distintos elementos. Las tres fases de las que esta formado son las siguientes:

- Planeación y Especificación de Requisitos: Planeación, definición de requisitos, construcción de prototipos, etc.
- Construcción: La construcción del sistema.
- Instalación: La transición de la implementación del sistema a su uso, en otras palabras, la puesta en marcha del sistema en el entorno previsto de uso.

Para llevarla a cabo se va adoptar un enfoque iterativo, tomando en cada iteración un subconjunto de los requisitos y llevándolo a través del diseño de alto y bajo nivel hasta la implementación y pruebas (Figura 1.9). El sistema va creciendo incrementalmente en cada ciclo. Con esta aproximación se consigue disminuir el grado de complejidad que se trata en cada ciclo, y se tiene pronto en el proceso una parte del sistema funcionando que se puede contrastar con el usuario/cliente.

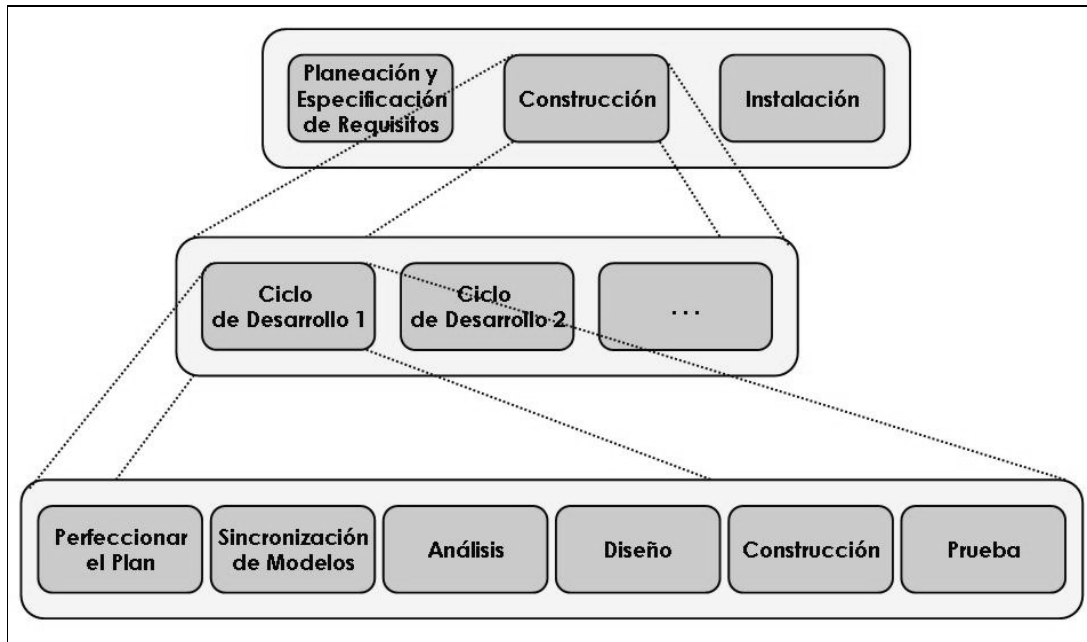


Figura. 1.9 Ciclo de Vida Iterativo, presenta un Desarrollo Iterativo en la Construcción

#### 1.4 Internet

Se define como, “una red telemática internacional, que procede de una red militar norteamericana (Arpanet, creada en 1969), que es fruto de la interconexión de múltiples redes que utilizan un mismo protocolo de comunicación (TCP-IP)” [19]. Todo usuario con una computadora personal provista de un módem se puede conectar a Internet a través de un servidor. Los servicios que ofrecen son la consulta de información (sitios web), la mensajería electrónica, el comercio electrónico, etc.

En pocas palabras, *Internet* es simplemente, una red de computadoras autónomas que se encuentran interconectadas entre si y que tienen como principal función intercambiar información a nivel mundial.

En Internet, las comunicaciones concretas se establecen entre dos puntos: uno es la computadora desde el que uno accede y el otro es cualquiera de los servidores





que hay en la Red y facilitan información. La forma en que realizan esa comunicación se debe al uso de protocolos.

### 1.5 Seguridad Informática

Podría definirse la seguridad de la información como la habilidad para proteger la información y los recursos respecto a la confidencialidad y la integridad. En este sentido es preciso garantizar la autenticación, confidencialidad, integridad y disponibilidad. No se puede defender el sistema con un concepto de fortaleza donde levantemos murallas empleando distintos productos, pues los sistemas en la actualidad tienen múltiples vías de entrada y salida de información que además cada vez deben ser más ágiles.

Así pues, las características principales de la seguridad son:

- Confidencialidad, es la propiedad de prevenir la divulgación de información a personas o sistemas no autorizados.
- Integridad, es la propiedad que busca mantener a los datos libres de modificaciones no autorizadas.
- Disponibilidad, es la característica, cualidad o condición de la información de encontrarse a disposición de quienes deben acceder a ella, ya sean personas, procesos o aplicaciones.

Cabe resaltar que la seguridad no es algo que deba implementarse momentáneamente, es más bien un proceso continuo que hay que gestionar conociendo siempre las vulnerabilidades y las amenazas que rodean a nuestra información, teniendo siempre en cuenta las causas de riesgo y la probabilidad de que ocurran, así como el impacto que puede tener. Una vez conocidos todos estos puntos, y nunca antes, deberán tomarse las medidas de seguridad oportunas.



### 1.5.1 Objetivos de la Seguridad Informática

El objetivo primordial de la Seguridad Informática es no permitir que la información sufra cualquier tipo de daño. Sin embargo además, tiene como objetivo proteger todos aquellos recursos del sistema de información o relacionado con éste, necesarios para que la organización funcione correctamente y alcance los objetivos propuestos, los que se les conoce como **Activos**.

Son tres elementos que los conforman:

- Información. Es el objeto de mayor valor, el objetivo es el resguardo de la información, independientemente del lugar en donde se encuentre registrada, en algún medio electrónico o físico.
- Equipos que la soportan. Software, hardware y organización.
- Usuarios. Individuos que utilizan la estructura tecnológica y de comunicaciones que manejan la información.

Queda claro que la seguridad es un estado que nos indica que ese sistema está libre de peligro, daño o riesgo. Se entiende como peligro o daño todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo.

Por tanto, un sistema puede considerarse como seguro cuando presenta las características conocidas como la Tríada CIA (Confidencialidad, Integridad, Disponibilidad - **Confidentiality, Integrity, Availability**), estos son los principios básicos de la seguridad de la información.



## CAPÍTULO 2. ANÁLISIS, DISEÑO Y OPTIMIZACIÓN DEL SISTEMA

El creciente uso de Internet y la utilización de nuevas tecnologías de la información han ido modificando el estilo de vida mundial, hoy en día es muy común ver los diferentes usos que se le da a la Internet, ya sea realizando pagos de diversos tipos, transmisiones en directo de eventos importantes, foros de todo tipo de temas, redes sociales y muchos más que diariamente se unen a esta tendencia tecnológica. El motivo es, que la Internet permite una manera más organizada de acceder a la información, presentando una interfaz amigable con el usuario mediante navegadores (Netscape, Microsoft Internet Explorer, etc.), sin límites de tiempo ni distancias.

Sin embargo todo este universo tecnológico no sería posible sin el uso conjunto de todas aquellas tecnologías de software que permiten integrar todas sus funcionalidades en un mismo sitio Web, de lo cual se ha encargado el Desarrollo Web. En este ámbito deben tomarse en cuenta algunos puntos clave para lograr el éxito en la construcción de nuestro Sistema Web, como por ejemplo un *Buen Contenido*, el cual dependerá directamente de la temática de nuestro sistema. Debemos tomar en cuenta que debemos aportar contenido al usuario, además es importante considerar que el entorno Web es un medio de comunicación distinto, pues los usuarios pasan de página en página Web a un ritmo muy veloz y generalmente pasan poco tiempo en una misma página. Por lo que se debe moderar el uso de la información textual y además complementar el contenido con el uso de imágenes.

Otro de los puntos que se debe considerar es el tener un *Buen Diseño*, colores, iconos e imágenes adecuados que hagan agradable la experiencia de visitar nuestras páginas. Lo importante en este aspecto es el facilitar la navegación al usuario mediante menús, iconos, y otros elementos para evitar que nuestros visitantes se pierdan dentro de nuestro sistema. Incluso puede pensarse en la



utilización de mapas, lo cuales son esquemas con enlaces de hipertexto que representan todo el contenido de nuestro sistema para contar con una referencia de nuestro sistema. Por último hay que tomar en cuenta el uso de *Elementos Interactivos* para así evitar al usuario la experiencia de una lectura simple y tediosa, lo cual involucra la actualización constante de contenido.

El balance en el uso de estos elementos del diseño brindará al usuario una buena experiencia al navegar a través de nuestro sistema y de esta forma habremos cumplido uno de los puntos más importantes del sistema.

## 2.1 Desarrollo Orientado a Objetos

Para construir un Sistema de Información no solo basta con conocer un lenguaje de programación, hay que tomar en cuenta además a los principales interesados cuando este se desarrolla como lo es el *Cliente*, quien paga por el sistema de información que se va a desarrollar, los usuarios futuros y los desarrolladores de ese sistema de información. La finalidad de esto es debido a que los desarrolladores tienen como tarea determinar las necesidades del cliente y desarrollar un sistema de información que satisfaga dichas necesidades y pueda ser utilizado por los usuarios.

Se debe además, analizar el problema y dar una solución diseñada para que nuestro sistema pueda ser robusto y soportado. Se debe seguir un proceso robusto, que incluya las actividades principales. Si se sigue un proceso de desarrollo que se ocupa de plantear cómo se realiza el análisis, el diseño y cómo se relacionan los productos de ambos, entonces la construcción va a poder ser planificada y repetitiva, y la probabilidad de obtener un sistema de mejor calidad al final del proceso aumenta considerablemente, especialmente cuando se trata de un equipo de desarrollo formado por varias personas.



Para poder llevar el desarrollo de esta manera, es necesario conocer el ciclo de vida de nuestro sistema de información, que no es más que la manera en que se construye dicho sistema. Debido a que casi siempre es más fácil realizar una secuencia de tareas pequeñas que una tarea grande, el ciclo de vida general se divide en una serie de pasos pequeños llamados fases.

Siguiendo el camino estructurado para el desarrollo de software nos encontramos con el dilema de utilizar una metodología que integre lo más posible de cada una de las etapas más importantes del ciclo de vida de nuestro software y que además cubra las particularidades necesarias de un sitio Web para un entorno académico. La elección no es sencilla ya que ésta debe estar sustentada por un análisis integral previo, tomando en cuenta las condiciones en que el desarrollo se lleve a cabo.

Las herramientas que se utilizarán, el propio lenguaje de programación, la proyección que se tendrá del propio software y las necesidades propias de un software que será utilizado por grupos académicos de profesores; característica muy particular dado que generalmente este tipo de sistemas está diseñado para usuarios que tienen un conocimiento previo del programa computacional, situación que se trata de solucionar en este caso.

En el desarrollo del Sistema Web Dentizt@ se tomará el método de desarrollo orientado a objetos que propone Craig Larman. Este proceso no fija una metodología estricta, sino que define una serie de actividades que pueden realizarse en cada fase, las cuales deben adaptarse según las condiciones del proyecto que se esté llevando a cabo.

Se ha escogido seguir este proceso debido a que el ciclo de vida proporciona una visión completa y coherente de la producción de sistemas software, empezando por los requisitos y terminando con el sistema funcionando.



El enfoque que toma Dentizt@ es el de un ciclo de vida iterativo incremental, el cual permite una gran flexibilidad al momento de adaptarlo a un proyecto y a un equipo de desarrollo específicos. El ciclo de vida está dirigido por casos de uso, es decir, por la funcionalidad que ofrece el sistema a los futuros usuarios del mismo. Así no se pierde de vista la motivación principal que debería estar en cualquier proceso de construcción de software: el resolver una necesidad del usuario/cliente.

Dentizt@ es un sistema que se desarrolla bajo la filosofía de la programación orientada a objetos, siendo la metodología que se usa para la notación de los modelos orientados a objetos la de UML (Unified Modeling Language – Lenguaje Unificado de Modelado).

El sistema esta pensado para trabajar bajo el sistema operativo Linux, por la ventaja de ser software libre, lo que además de reducir costos permite el ingreso al código fuente del Sistema Operativo para poder realizar mejoras. Además, este ha sido elegido por contar con un esquema de seguridad basado en un sistema de permisos de lectura, escritura y ejecución establecidos a los archivos y directorios con lo que los usuarios únicamente podrán modificar sus propios archivos a menos que el administrador le haya otorgado los permisos correspondientes.

### **2.1.1 Enfoque General de la Metodología**

La Metodología está formada por una serie de actividades y subactividades, cuya realización se va repitiendo en el tiempo, aplicadas a distintos elementos. Las tres fases de las que esta formado son las siguientes:

- Planeación y Especificación de Requisitos: Esta fase comprende la planeación de la solución del problema, la definición de los requisitos que el sistema presenta, así como, la construcción de prototipos.
- Construcción: Esta fase se refiere a la construcción del sistema.



Las fases dentro de esta etapa son las siguientes:

- Diseño de Alto Nivel: Se analiza el problema a resolver desde la perspectiva de los usuarios y de las entidades externas que van a solicitar servicios al sistema.
- Diseño de Bajo Nivel: El sistema se especifica en detalle, describiendo cómo va a funcionar internamente para satisfacer lo especificado en el Diseño de Alto Nivel.
- Implementación: Se lleva lo especificado en el Diseño de Bajo Nivel a un lenguaje de programación.
- Pruebas: Se llevan a cabo una serie de pruebas para corroborar que el software funciona correctamente y que satisface lo especificado en la etapa de Planificación y Especificación de Requisitos.
- Instalación: La puesta en marcha del sistema en el entorno previsto de uso.

De ellas la fase de Construcción es la que consumirá la mayor parte del esfuerzo y del tiempo en el desarrollo del sistema. Para llevarla a cabo se adoptará un enfoque iterativo, tomando en cada iteración un subconjunto de los requisitos y llevándolo a través del diseño de alto y bajo nivel hasta la implementación y pruebas (Figura 2.1). El sistema va creciendo incrementalmente en cada ciclo. Con esta aproximación se consigue disminuir el grado de complejidad que se trata en cada ciclo y se tiene pronto en el proceso, una parte del sistema funcionando que se puede contrastar con el usuario/cliente.

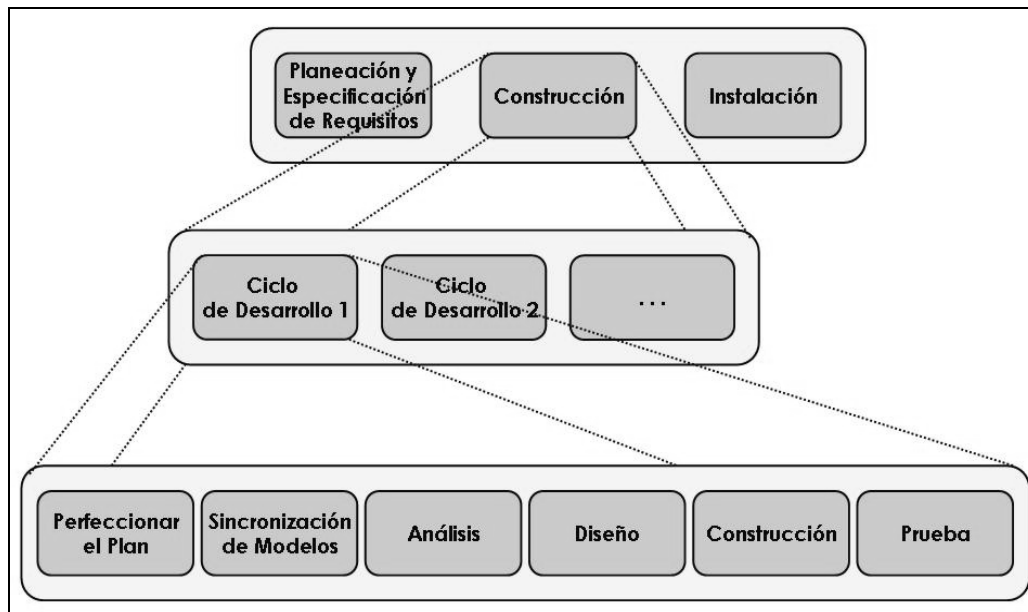


Figura 2.1 Ciclo de Vida Iterativo, presenta un Desarrollo Iterativo en la Construcción

Finalmente, lo más importante a considerar en el método, es el punto de vista desde el cual se analiza el proceso de diseño, el cual está pensado no desde la interrelación usuario-tecnología, sino como interacción humana mediatizada por la tecnología. Es decir, entender en este caso a la Internet como medio de comunicación, lo que tiene como fin pensar permanentemente en los usuarios finales para quienes se diseña un sistema.

## 2.2 Lenguaje Unificado de Modelado (UML): Una técnica para la especificación del Sistema en todas sus fases

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje gráfico para: visualizar, especificar, construir y documentar un sistema de software.

UML ha puesto fin a las llamadas “guerras de métodos”, que se mantuvieron a lo largo de los años 90, en que los principales métodos mostraban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de





éstas técnicas para formar una herramienta compartida entre todos los ingenieros de software que trabajan en el desarrollo del mismo.

Uno de los objetivos principales de la creación de UML, fue posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado.

El UML es una definición oficial de un lenguaje pictórico con símbolos y relaciones usuales que tienen un significado común. Es importante reiterar que es más rápido y más sencillo resolver problemas con imágenes que con código. El UML comprende símbolos y una gramática que define la manera en que se pueden utilizar estos símbolos. A continuación se definirán los tipos de diagramas que se pueden crear con este lenguaje.

### **2.2.1 Diagramas de Casos de Uso**

Los diagramas de casos de uso son el equivalente al arte rupestre moderno. Este diagrama es el responsable principalmente de documentar los macro requisitos del sistema, en otras palabras muestra la relación entre los actores y los casos de uso del sistema. Los símbolos principales de este diagrama son el actor, el conector y el ovalo del caso de uso, mismos que se muestran en la Tabla 2.1.

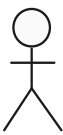


Símbolo	Definición
 <b>Actor</b>	El <b>actor</b> , representa los participantes en los casos de uso. Los actores son algo con comportamiento, pueden ser personas o cosas, son descubiertos como resultado del análisis y conforme se vayan identificando los macroúsos del sistema, identificara quienes son los participantes de esos casos de uso.
	Los <b>conectores</b> se utilizan para indicar la manera en que los actores y los casos de uso están asociados.
	Un <b>caso de uso</b> es una descripción de la secuencia de interacciones que se producen entre un actor y un sistema, es decir, se utiliza para representar capacidades del actor, al caso de uso se le da un nombre y una descripción mediante un texto. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo en el sistema.

Tabla 2.1 Símbolos de los Diagramas de Casos de Uso [1]

A continuación se muestra un ejemplo de un caso de uso en la figura 2.2.

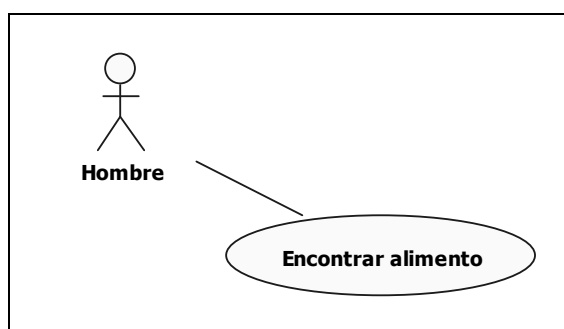


Figura 2.2 Ejemplo de Caso de Uso



### 2.2.1.1 Estilos de Línea para los Conectores

Los estilos de línea de los conectores cambian para transmitir más información acerca de la relación entre los actores y los casos de uso. Existen tres estilos básicos de líneas para los conectores mostrados en la Tabla 2.2:

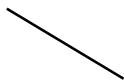


Estilo de línea para el conector	Uso
	Un <b>conector de línea simple</b> se llama asociación y se utiliza para mostrar cuales actores están relacionados con cada caso de uso (Figura 2.3).
	Un <b>conector de línea punteada</b> y con una flecha direccional se le llama dependencia. La flecha apunta hacia el caso de uso del que se depende (Figura 2.3).
	Un <b>conector de línea dirigida con un triangulo hueco</b> , se le conoce como generalización, que significa en UML “herencia”. Cuando se muestra una relación de generalización entre dos actores o dos casos de uso, se está indicando que el actor o caso de uso generalizado puede ser substituido por cualquiera de los elementos hijos ya que comparte su estructura y comportamiento (Figura 2.4).

Tabla 2.2 Estilos de líneas para los conectores [1]

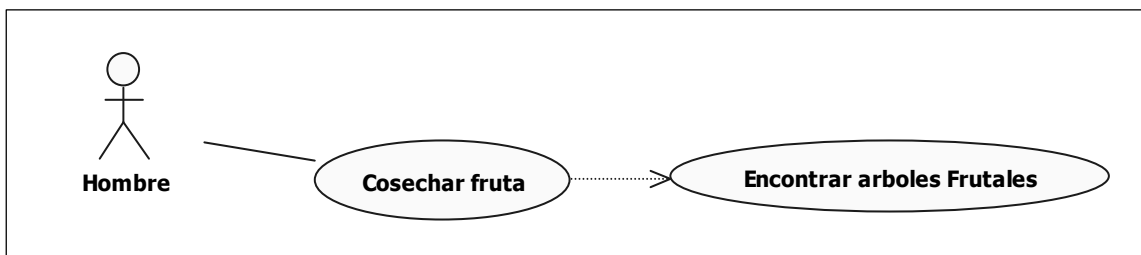


Figura 2.3 El caso de uso “cosechar fruta” depende de encontrar árboles frutales

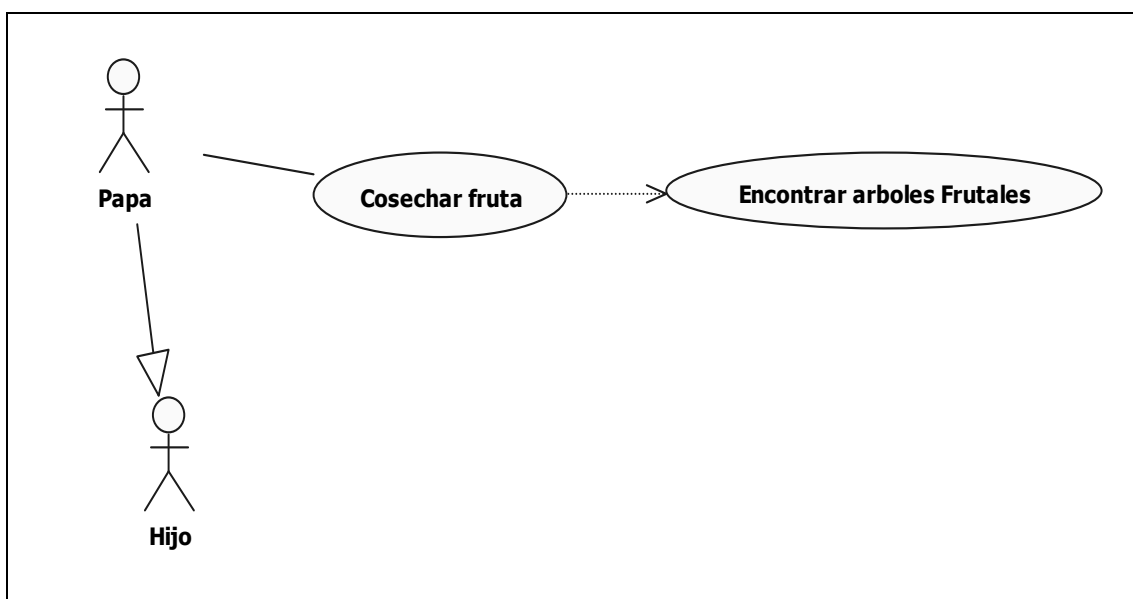


Figura 2.4 Diagrama de Casos de Uso en el que se muestra la relación de generalización entre dos actores

## 2.2.2 Diagramas de Actividades

Un diagrama de actividades es la versión UML de un diagrama de flujo. Los diagramas de actividades se usan para analizar los procesos, modelar el flujo de control de actividades y, si es necesario, volver a realizar la ingeniería de los procesos.

Básicamente un diagrama de actividades contiene: estados de actividad, estados de acción, transiciones y objetos.



Los estados de actividad y de acción se representan con un rectángulo con puntas redondeadas, cuyo interior representa una actividad o una acción como se muestra en la Tabla 2.3 a continuación:

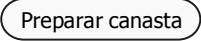

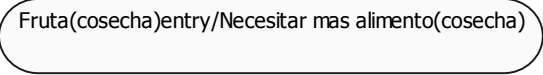
	Estado de acción con una acción simple.
	Estado de acción con una expresión.
	El estado de actividad, puede descomponerse en más sub-actividades representadas por otros diagramas de actividades. En los estados de actividad podemos encontrar elementos adicionales como: acciones de entrada (entry) y de salida (exit) del estado en cuestión.

Tabla 2.3 Estados de Actividad y de Acción



Otros símbolos que forman el diagrama de actividades son los mostrados en la Tabla 2.4 a continuación.


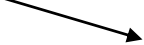
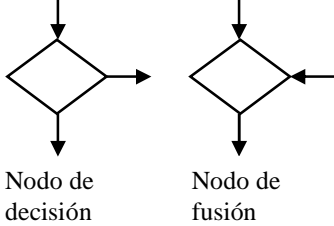


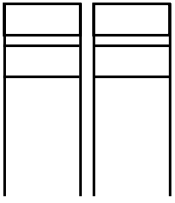
Símbolo	Definición
	El nodo inicial, es un círculo relleno que puede tener una línea de transición saliendo de él.
	El flujo de control es una flecha dirigida, también suele llamarse flujo o estímulo. Reflejan el paso de un estado a otro.
 <p>Nodo de decisión      Nodo de fusión</p>	Los nodos de decisión y fusión utilizan el símbolo con forma de diamante, la diferencia entre estos es que los nodos de decisión tienen un estímulo que entra al nodo y múltiples estímulos saliendo de este. El nodo de fusión tiene múltiples estímulos entrando y solamente uno saliendo del nodo.
	UML representa gráficamente el proceso de división, que representa la concurrencia, y el momento de la unión de nuevo al flujo de control secuencial, por una línea horizontal ancha.
	El nodo final de actividad indica el fin del flujo.
	Los carriles sirven para mostrar quién o qué es responsable de una parte de una actividad dentro del sistema.

Tabla 2.4 Símbolos del Diagrama de Actividades [1]



A continuación se muestra un diagrama de actividades utilizando los símbolos antes mencionados. En la Figura 2.5 se ejemplifica el diagrama de actividades donde el hombre camina para encontrar alimento.

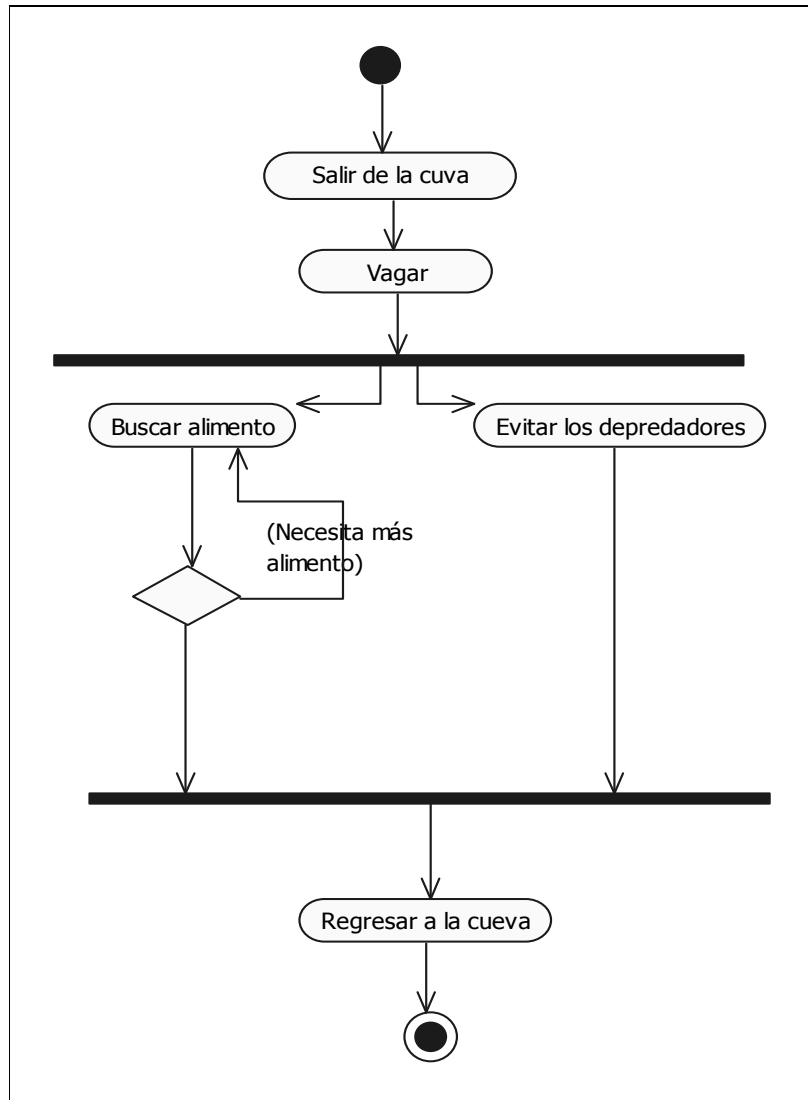


Figura 2.5 Ejemplo de Diagrama de Actividades



### 2.2.3 Diagramas de Clases

Los diagramas de clases se utilizan para mostrar las clases de un sistema y las relaciones entre ellas.

Los diagramas de clases muestran una vista estática del sistema (Figura 2.6); no describen los comportamientos o cómo interactúan entre ellas.

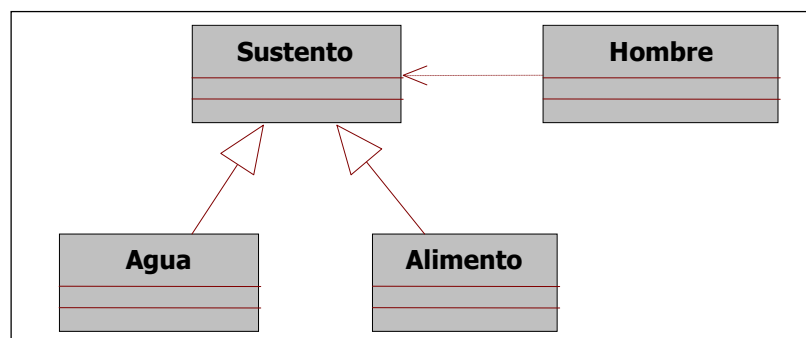


Figura 2.6 Ejemplo de Diagrama de Clases

### 2.2.4 Diagramas de Interacción

Existen dos tipos de diagramas de interacción: el diagrama de secuencia y el diagrama de colaboración. Ambos transmiten la misma información, empleando una perspectiva diferente.

Las secuencias tienen un ordenamiento explícito en el tiempo y son lineales. Las colaboraciones tienen un ordenamiento rotulado en el tiempo y son geométricas

### 2.2.5 Diagramas de Secuencia

Los diagramas de secuencia muestran las clases a lo largo de la parte superior y los mensajes enviados entre esas clases, modelando un solo flujo a través de los objetos del sistema. En particular, muestra los objetos participantes en la interacción y los mensajes ordenados que intercambian según su secuencia en el





tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción (Figura 2.7).

A continuación, en la Tabla 2.5 se muestran los símbolos utilizados para generar el diagrama de secuencias.

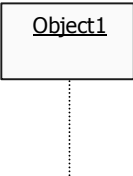

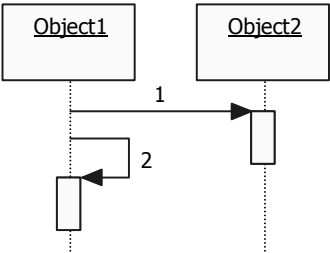
Símbolo	Descripción
	La <b>línea de vida</b> es un rectángulo con una recta vertical que desciende de ese rectángulo. La línea de vida representa un actor, un objeto o una clase, y la línea que desciende en forma vertical es un lugar conveniente para sujetar mensajes entrantes y salientes.
	El símbolo de activación es un rectángulo vertical que reemplaza la línea de vida en el transcurso de la duración de la existencia de ese caso.
	Los mensajes son líneas dirigidas que conectan líneas de vida. La línea se inicia en una línea de vida, y la flecha apunta hacia aquella línea que contenga el mensaje invocado. El mensaje puede empezar y finalizar en la misma línea de vida; a esto se le conoce con el nombre de llamada anidada.

Tabla 2.5 Símbolos del Diagrama de Secuencias [1]

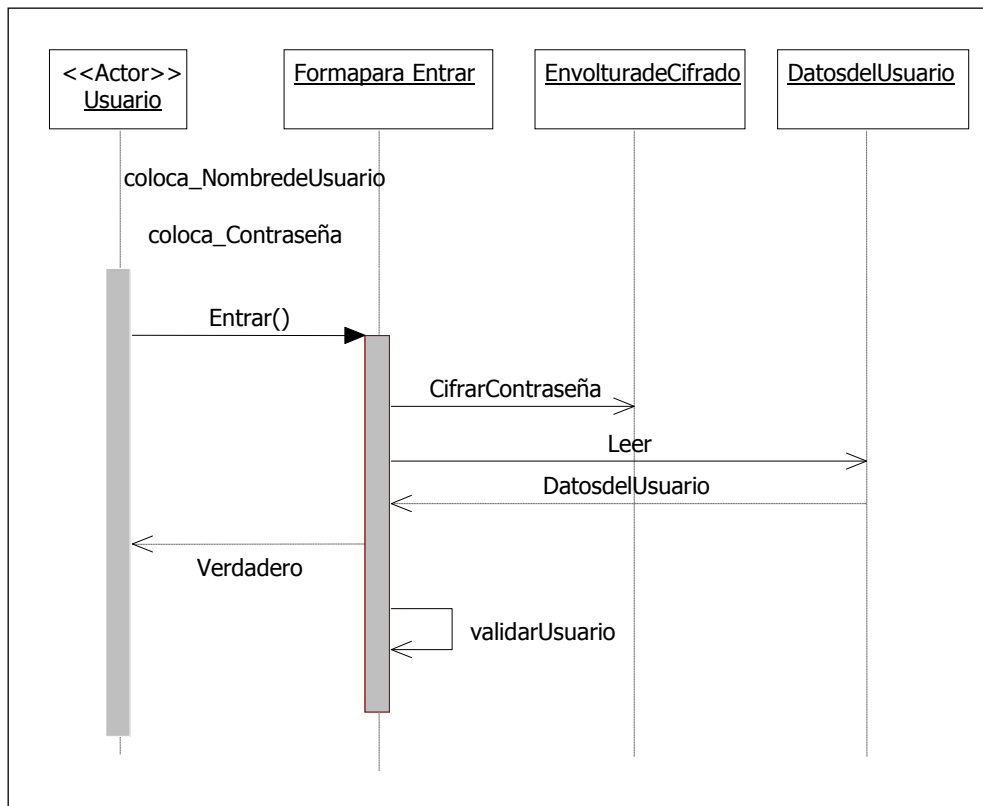


Figura 2.7 Ejemplo de Diagrama de Secuencia

### 2.2.6 Diagramas de Colaboración

Un diagrama de colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos. A diferencia de los Diagramas de secuencia, los diagramas de colaboración muestran las relaciones entre, los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

A continuación se mencionan los símbolos que generan un diagrama de colaboración, mostrados en la Tabla 2.6.



Símbolo	Descripción
	El rectángulo llamado papel clasificador representa los objetos.
	Línea que indica el mensaje, llamada conector.

Tabla 2.6 Símbolos de un Diagrama de Colaboración

A continuación se muestra un ejemplo del diagrama de colaboración (Figura 2.8).

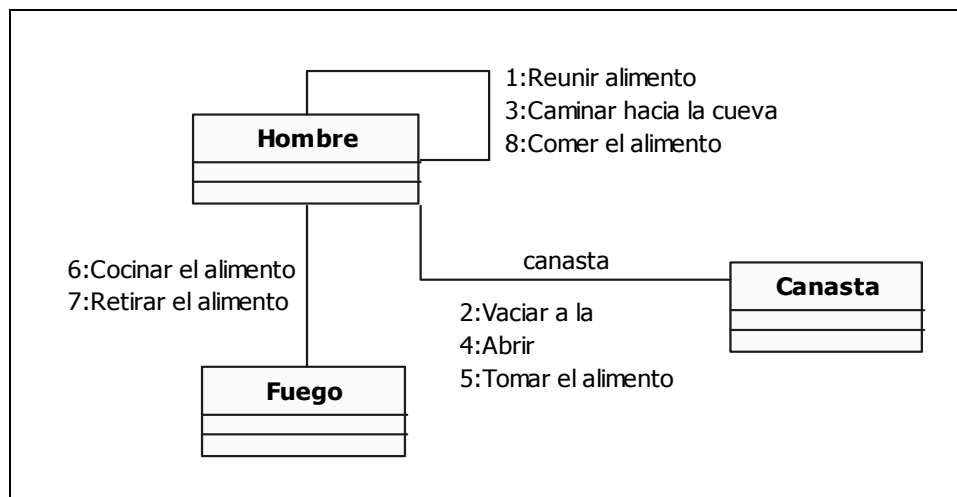


Figura 2.8 Ejemplo de Diagrama de Colaboración

## 2.2.7 Diagramas de Estado

Un diagrama de estado muestra el estado cambiante por los que pasa un solo objeto, un caso de uso o un sistema a lo largo de su vida.

En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos.

A continuación se muestra un ejemplo del diagrama de estado (Figura 2.9).

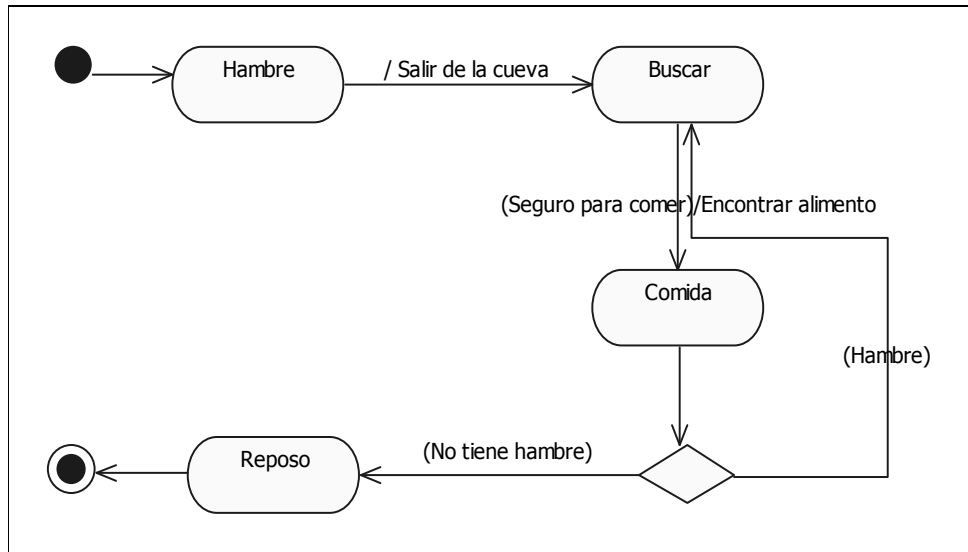


Figura 2.9 Ejemplo de Diagrama de Estado

### 2.2.8 Diagrama de Componentes

Los componentes pertenecen al mundo físico, es decir, representan un bloque de construcción al modelar aspectos físicos de un sistema. Una característica básica de un componente es que: debe definir una abstracción precisa con una interfaz bien definida, y permitiendo reemplazar fácilmente los componentes más viejos con otros más recientes y compatibles.

En UML todos los elementos físicos se modelan como componentes. UML lo representa gráficamente de la siguiente manera (Figura 2.10), donde xxxx.dll es el nombre del componente.

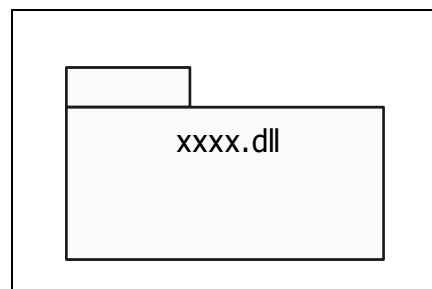


Figura 2.10 Representación Gráfica de un Componente



## 2.3 Fase de Planeación y Especificación de Requisitos

Esta fase corresponde la Especificación de Requisitos tradicional ampliada con un Borrador de Modelo Conceptual y con una definición de Casos de Uso de alto nivel.

Las actividades que comprenden a esta fase son las siguientes:

- Definir el Plan-Borrador.
- Crear el Informe de Investigación Preliminar.
- Definir los Requisitos.
- Definir Casos de Uso (de alto nivel y esenciales).
- Refinar el Plan.

### 2.3.1 Antecedentes del Sistema Web Dentizt@

Debido al veloz avance en la aplicación y desarrollo de las tecnologías de información y las comunicaciones en la educación, la investigación científica y en múltiples actividades dentro de la sociedad, ha causado que se emprendan acciones que permitan la adecuación de las mismas a las demandas sociales y educativas.

Dentro del ámbito educativo, una de las acciones más importantes en los últimos años, ha sido la incorporación de las TIC en la educación, apreciándose de esta manera nuevas formas de experimentar la enseñanza-aprendizaje, donde el componente tecnológico se manifiesta de manera relevante.

En este contexto, donde la característica principal es el uso de la tecnología informática, la electrónica y las comunicaciones para el control y optimización de la información, se presenta un hecho relevante en la que los límites espaciales, los tiempos y las formas de organizarse para transmitir el conocimiento se han superado.



De esta manera los grupos de trabajo académico que antes utilizaban la tecnología para enviarse información ahora también la usan para difundir el conocimiento producto de la interacción de los miembros del grupo y/o del grupo con otros grupos, teniendo en Internet una herramienta que les permite rebasar los límites impuestos por la distancia y el tiempo. Y es que no está por demás decir que el campo más desarrollado en Internet es el de la información, que se manifiesta fundamentalmente a través de páginas Web, lo que posibilita la comunicación enviando y recibiendo mensajes en diferentes formas en directo mediante vídeo-conferencia, audio-conferencia y texto-conferencia, ventajas que han sido aprovechadas por las llamadas comunidades virtuales de aprendizaje.

Una Comunidad de Aprendizaje puede definirse como “una comunidad humana organizada que construye y se involucra en un proyecto educativo y cultural propio para educarse a sí misma, a sus niños, jóvenes y adultos, en el marco de un esfuerzo endógeno, cooperativo y solidario, basado en un diagnóstico no sólo de sus carencias sino, sobre todo, de sus fortalezas para superar tales debilidades” [20].

Complementado con el uso de la Internet, como herramienta para cubrir las necesidades de comunicación e interacción, permite compartir intereses comunes. Lo que diferencia a estas comunidades de las demás.

Sin embargo, ante este escenario, se perciben algunos obstáculos que enfrentan estas comunidades y que tienen que ver con la falta de conocimiento de cómo crear éstos espacios en la Internet, dedicados a la interacción de los miembros de estos grupos con otros sectores académicos que les permita compartir, intercambiar y colaborar en proyectos para el diseño de propuestas que contribuyan al mejoramiento del proceso de enseñanza-aprendizaje; aprovechando de esta manera la información, actividades y experiencias de los participantes de estas comunidades.



Dadas las condiciones actuales de las comunidades que trabajan a través de la Internet, del proceso enseñanza-aprendizaje y de las tecnologías de información, los campos del conocimiento ahora tienen en común el uso de la tecnología como una de las herramientas primarias para el desarrollo de sus actividades.

### **2.3.1.1 Antecedentes Académicos del Proyecto**

Desde 1999 el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET) conjuntamente con la Facultad de Estudios Superiores Iztacala de la UNAM, ha realizado actividades de exploración y análisis del valor de la tecnología informática dentro del modelo de enseñanza que se aplica en el aula. La realización de estas actividades ha tenido como fin primordial considerar la utilización de la telemática como una de las diversas herramientas que se pueden utilizar en cualquier programa educativo, destacando la necesidad de planificar cuidadosamente los objetivos y estrategias tecnológicas que actualmente usan alumnos y profesores en la enseñanza-aprendizaje, enfocando este objetivo a la enseñanza de la Odontología.

El proyecto “Sistema Web Dentizt@”, surge en 1999 por la inquietud de algunos profesores de la Facultad de Estudios Superiores Iztacala (FES Iztacala) de integrar las Tecnologías de la Información y las Comunicaciones como una herramienta de apoyo en sus actividades docentes así como para el desarrollo de materiales didácticos multimedia que les permitía contribuir al mejoramiento del proceso de enseñanza-aprendizaje.

Muchos de los materiales diseñados y desarrollados por el grupo de docentes (manuales, tutoriales, videos, presentaciones) son importantes para los miembros de la comunidad, ya que éstos describen procedimientos y actividades de interés para el grupo, por lo que la falta de un espacio para publicar este material, ocasiona inconvenientes tales como el tiempo invertido para poder analizar y buscar de manera grupal el material, la distribución de éste a los miembros de la



comunidad y la interacción y comunicación del grupo con otros grupos académicos. Por lo que la necesidad de contar con un sistema que proporcione un espacio de comunicación donde se realice la interacción de profesores, alumnos y otros sectores académicos era más notoria y cobraba relevancia.

Para cubrir esta necesidad a través de la colaboración entre el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET) y la Facultad de Estudios Superiores Iztacala (FESI) de la Universidad Nacional Autónoma de México, se creó el proyecto “Sistema Web Dentizt@”, en el que los usuarios tienen la posibilidad de publicar su propio material para ser consultado por estudiantes y profesores.

### **2.3.1.2 Problemática del Sistema Existente**

En resumen, Dentizt@, es un sistema de información que trabaja en un entorno Web que tiene como herramienta principal el uso de Internet y que fue creado con la idea de disminuir los problemas de distancia y espacio que se presentaron para el trabajo multidisciplinario de grupos de académicos, ubicados geográficamente a grandes distancias, en el momento en que se inició el proyecto bajo el cual fue diseñado.

Este sistema Web fue creado para que sus diferentes módulos trabajarán para un fin común: el apoyo a la docencia. Capaz de realizar las actividades básicas necesarias que se requerían al momento de ser creado: entrada, almacenamiento, procesamiento y salida de la información, el sistema ha llegado a una etapa en que su funcionamiento se ha convertido en un problema, truncándose su uso y expansión debido al crecimiento en las necesidades de autonomía de los académicos que conforman las comunidades que hacen uso del mismo.

Actualmente el Sistema Dentizt@ se encuentra carente de autonomía para el usuario en la creación y eliminación de nuevos espacios, por lo que se requiere de





una interfaz más sencilla que le permita al usuario un uso fácil del sistema, además de la incorporación de elementos valiosos que faciliten su uso y manipulación.

Otro problema de gran relevancia que presenta el sistema actualmente, es la poca documentación que los desarrolladores dejaron para su mantenimiento. La insuficiente información del cómo había sido desarrollado el sistema, la falta de una metodología y el diseño poco estructurado han sido las principales barreras para concebir una actualización del mismo.

### **2.3.1.3 Características de los Módulos Existentes**

El marco que presenta actualmente el sistema (Figura 2.11), es un ambiente Web que permite gestionar, organizar y administrar los materiales desarrollados por los profesores que forman parte de estas comunidades telemáticas, así como un espacio de comunicación donde interactúan tanto profesores como alumnos y pueden contribuir en la adquisición y construcción de conocimientos. Es un sitio en el cual tienen la posibilidad de publicar su propio material (manuales, tutoriales, videos, entre otros) para ser consultado por los estudiantes y maestros de la carrera de Cirujano Dentista, dentro y fuera de las aulas. Teniendo un Administrador capaz de generar, actualizar y eliminar comunidades.

La interfaz del Sistema Web está basada en diversos programas de diseño y edición de imágenes, creando un ambiente gráfico y atractivo para los usuarios en donde se encuentran pantallas de acceso y de registro, como las siguientes:

- Menú principal del Sistema Dentizt@
- Solicitud de ingreso a la comunidad o de login y password
- Documentos de la ciberoteca y acceso al foro de la comunidad
- Lista de los integrantes de cada comunidad
- Registro de evento para publicar en Internet



- Solicitud de registro de comunidad al Administrador del sistema
- Módulo del Administrador del sistema de comunidades

Respecto a las condiciones técnicas en las que se encuentra Dentizt@, estas son:

- Linux, como sistema operativo en donde se aloja el servidor.
- Apache, como servidor http.
- MySQL, como manejador de bases de datos.
- PHP, como lenguaje de programación.

Dado este marco, se encuentra un sistema carente de autonomía para el usuario en la creación y eliminación de nuevos espacios. Donde, además, se pretende obtener una interfaz más sencilla para la fácil utilización del usuario.

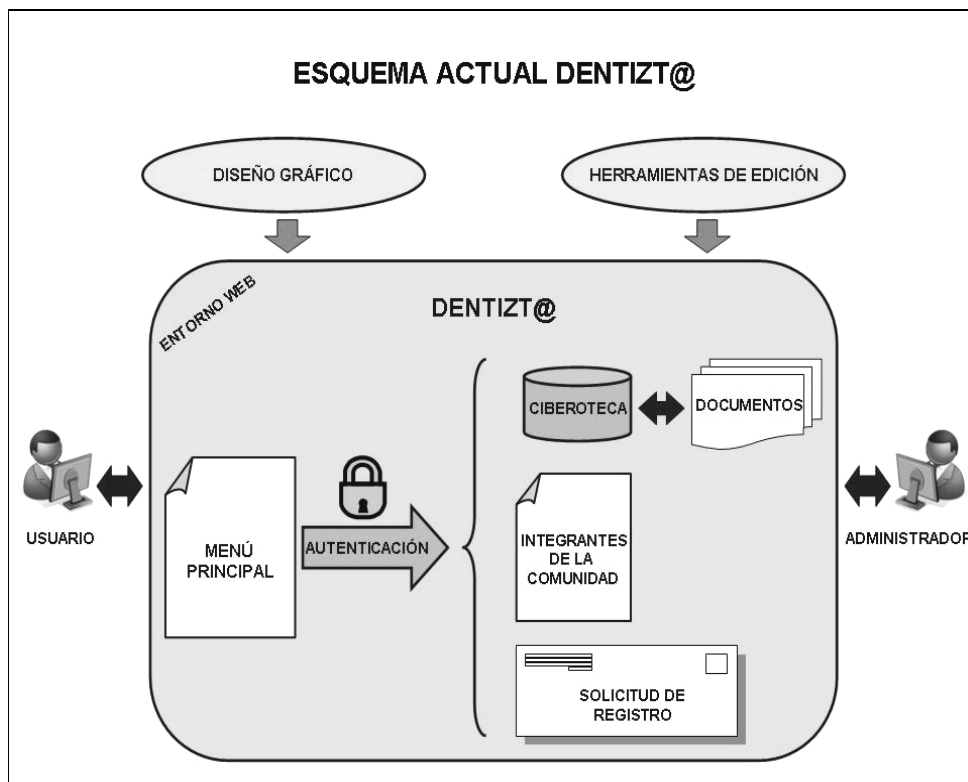


Figura 2.11 Estructura actual del Sistema Web Dentizt@



### 2.3.2 Requisitos del Sistema

El Sistema Web Dentizt@ requiere de la organización y administración de comunidades virtuales de docentes de la FES Iztacala, así como el material requerido por y para ellos, permitiendo hacer uso de las tecnologías informáticas aplicadas a la docencia, como un espacio de apoyo a las actividades de los docentes permitiendo abrir canales de interacción, comunicación y retroalimentación entre los participantes, eliminando las limitaciones de comunicación asíncrona entre ellos.

Estas comunidades deben tener la capacidad de publicar su propio material, a través de un administrador de la comunidad, para ser consultado por los usuarios del sistema (otros profesores y/o estudiantes) dentro y fuera de las aulas, además de cubrir objetivos generales, tales como, facilitar y fomentar el desarrollo de las relaciones profesionales entre colegas en el ámbito académico, superando las divisiones de tiempo, geográficas y campos disciplinarios.

Como producto de la reingeniería del sistema se pretende:

- a) Obtener una interfaz más sencilla para la fácil utilización del sistema que le proporcione a los profesores sencillez en el acceso y uso.
- b) Posibilitar al profesor-usuario una interfaz que le permita administrar autónomamente los grupos virtuales así como la creación y disposición de los espacios para publicar en la red.

En conclusión, la optimización y reingeniería del Sistema Web Dentizt@ pretende la incorporación de elementos valiosos que faciliten su uso y manipulación, apoyándonos en herramientas computacionales como: lenguajes de programación, elementos multimedia, diseño e implementación de nuevas bases de datos, para que permitan al usuario una comunicación directa, sencilla, segura y rápida con los integrantes de las diversas comunidades. Además debe ser capaz de fomentar en los usuarios la construcción de nuevas comunidades, para integrar



a su vez a nuevos usuarios. Esto beneficiará a los usuarios que no están identificados con el mundo de la tecnología computacional, poniendo a su alcance un mundo lleno de información, útil y necesaria para la educación del mismo, tomando en cuenta sus necesidades e intereses, siendo su mejor característica su interfaz amigable.

Por último, es necesario remarcar la importancia de la utilización de las Tecnologías de la Información y la Comunicación (TIC's) en el ámbito de la educación, no solo como herramientas de comunicación y de apoyo para la elaboración de materiales educativos, sino como medio de producción de información y conocimiento.

La autonomía que proporcionará al usuario, permitiéndole administrar el sistema de manera directa, sencilla y segura por medio de una interfaz de fácil acceso, sin que sea necesario ser un experto en cómputo, permitirá abrir un canal de acceso a usuarios no expertos, lo que permitirá mayor productividad e innovación tecnológica-educativa de los docentes.

Con base a lo anterior y dado que el sitio Web es un sistema conocido e identificado por docentes y grupos de la disciplina Odontológica es necesario mantener la estructura actual, pero considerando un nuevo modelo que permita mantener un diseño actualizado de forma constante y continua, por lo que se consideró darle un nuevo enfoque al sitio por medio de un proceso de Reingeniería (Figura 2.12), optimizando de esta manera la estructura del sistema desde su interfaz, el uso de los recursos que se tienen en el sitio para la comunicación y colaboración de los grupos que participan en el proyecto y la comunicación entre grupos y usuarios del recurso Web.

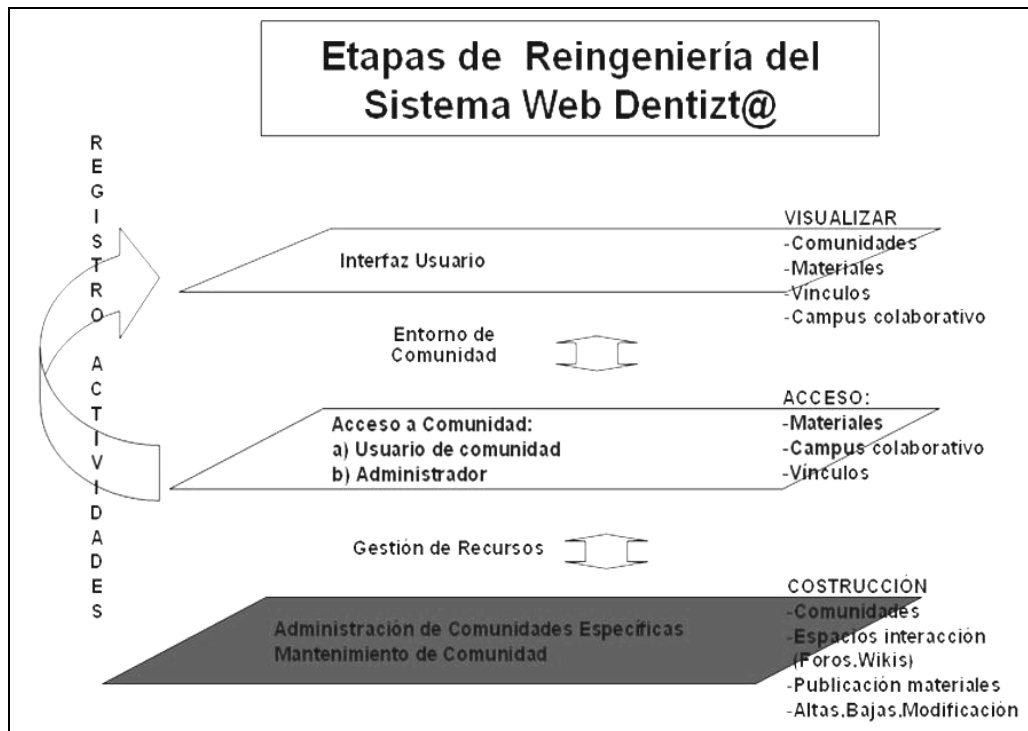


Figura 2.12 Procesos resultado del Análisis del Sistema Web Dentizt@ 1.0

La reingeniería del sistema muestra un nuevo enfoque del sitio; basado en un lenguaje orientado a objetos, proporciona una filosofía válida en la construcción del sistema. Abarcando todo el abanico de situaciones y complejidades que se presentan en un sitio web de las características de Dentizt@.

El formato del documento de Especificación de Requisitos no está definido en UML, pero se ha incluido este punto para resaltar que la actividad de definición de requisitos es un paso clave en la creación de cualquier producto software. Para entender y describir los requisitos la técnica de casos de uso constituye una valiosa ayuda.



### 2.3.3 Casos de Uso

Como se había mencionado anteriormente los casos de uso son documentos narrativos que describen a los actores utilizando un sistema para satisfacer un objetivo, en otras palabras, son requisitos funcionales.

Un escenario es un camino concreto a través del caso de uso, en primera instancia lo que interesa es abordar un caso de uso desde un nivel de abstracción alto, utilizando el formato de alto nivel. Cuando se requiere describir un caso de uso con más detalle se utiliza el formato expandido.

En un caso de uso de alto nivel se describen los requisitos de manera muy general pero es muy útil para comprender el ámbito y el grado de complejidad del sistema. El formato de casos de uso expandidos se utiliza en los casos de uso que se consideren más importantes, es decir, que influyan al resto. La principal diferencia con un caso de uso de alto nivel, está en que incluye un apartado de Curso Típico de Eventos.

A continuación se describen los apartados del formato de casos de uso:

1. Caso de uso: Nombre del Caso de Uso.
2. Actores: Lista de actores, indicando quien inicia el caso de uso.
3. Curso típico de eventos: Descripción de la interacción entre los actores y el sistema mediante las acciones numeradas de cada uno.
4. Cursos alternativos: Puntos en los que se puede surgir una alternativa, junto con la descripción de la excepción.

#### 2.3.3.1 Identificación de Casos de Uso.

La identificación de los casos de uso requiere un conocimiento medio acerca de los requisitos, y se basa en la revisión de los documentos de requisitos existentes.



Existen dos maneras para identificar los casos de uso:

a) *Basado en actores*

1. Identificar los actores relacionados con el sistema y/o la organización.
2. Para cada actor, identificar los procesos que inicia o en los que participa.

b) *Basado en eventos*

1. Identificar los eventos externos a los que el sistema va a tener que responder.
2. Relacionar los eventos con actores y casos de uso.

### 2.3.4 Construcción del Modelo de Casos de Uso

Para construir el modelo de casos de uso se tienen que identificar los actores y los casos de uso de alto nivel. Para la reingeniería del Sistema Web Dentizt@ se identificaron los siguientes actores y sus respectivos casos de uso (Figura 2.13).

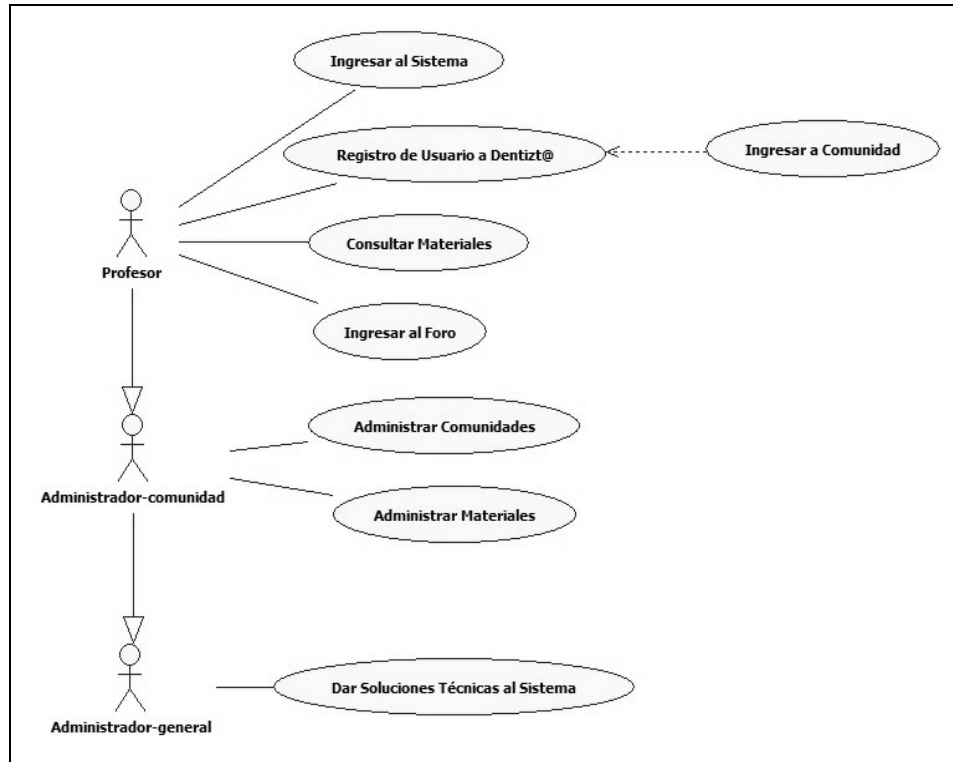


Figura 2.13 Modelo de Casos de Uso del Sistema Web Dentizt@



## CAPÍTULO 3. CONSTRUCCIÓN E IMPLEMENTACIÓN DEL SISTEMA DENTIZT@

Una vez identificados los requisitos y de haber definido los casos de uso de alto nivel de nuestro sistema, se inicia la siguiente fase llamada *Construcción*. Esta fase comprende el paso para el desarrollo de nuestro sistema, ésta es la que va a consumir la mayor parte del esfuerzo y del tiempo en nuestro proyecto de desarrollo. Para llevarla a cabo se va adoptar un enfoque iterativo, tomando en cada iteración un subconjunto de los requisitos y llevándolo a través del diseño de alto y bajo nivel hasta la implementación y pruebas. Finalmente en esta fase, se llevan a cabo una serie de pruebas para validar que el software funciona correctamente y que satisface lo especificado en la etapa de Planeación y Especificación de Requisitos.

Por último, una vez que se han cumplido con todos los requerimientos del sistema y que se tiene una versión estable se pasará a la fase final llamada *Instalación* en donde pondremos en marcha el sistema en el entorno previsto para su uso.

El objetivo es que al final de este proceso, se obtenga el Sistema Web Dentizt@ optimizado que permita al usuario la administración de nuevos espacios de comunicación interactiva, por medio de una interfaz de fácil acceso y que además permitan al usuario final una comunicación directa, sencilla, segura y rápida con los integrantes de las diversas comunidades, de manera autónoma para no depender de un administrador.

### 3.1 Fase de Construcción

Esta fase se refiere a la construcción del Sistema Web Dentizt@, se comienza una vez que se han definido los requisitos, donde se tiene un panorama general del mismo.





Las actividades que comprende la fase de Construcción del Sistema Web Dentizt@ son las siguientes:

- **Diseño de Alto Nivel:** Se analiza el problema a resolver desde la perspectiva de los usuarios y de las entidades externas que van a solicitar servicios al Sistema Dentizt@.
- **Diseño de Bajo Nivel:** El Sistema Dentizt@ se especifica en detalle, describiendo cómo va a funcionar internamente para satisfacer lo especificado en el Diseño de Alto Nivel.
- **Implementación:** Se lleva lo especificado en el Diseño de Bajo Nivel a un lenguaje de programación.
- **Pruebas:** Se llevan a cabo una serie de pruebas para corroborar que el software funciona correctamente y que satisface lo especificado en la etapa de Planificación y Especificación de Requisitos.

Para llevarla a cabo se adopta un enfoque iterativo, tomando en cada iteración un subconjunto de los requisitos y llevándolo a través del diseño de alto y bajo nivel hasta la implementación y pruebas. Como resultado de las iteraciones, el sistema va creciendo incrementalmente en cada ciclo. Con esta aproximación se consigue disminuir el grado de complejidad que se trata en cada ciclo, y se tiene rápidamente funcionando una parte del Sistema Web Dentizt@.

### **3.1.1 Diseño de Alto Nivel**

La fase de Diseño de Alto Nivel de un ciclo de desarrollo comprende la investigación sobre el problema, sobre los conceptos relacionados con el subconjunto de casos de uso que tenemos en Dentiz@. Tiene como objetivo llegar a una buena comprensión del problema como desarrolladores del sistema, sin entrar en cómo va a ser la solución en cuanto a detalles de implementación.



Después del primer ciclo de desarrollo, antes de la fase de Diseño de Alto Nivel hay una serie de actividades de planeación. Con estas actividades actualizamos los modelos que se tengan según lo que se haya implementado, tomando en consideración que se producen desviaciones entre lo que se ha analizado y diseñado y lo que finalmente se construye. Finalmente, una vez que se tienen los modelos acordes con lo implementado, se inicia el nuevo ciclo de desarrollo con la fase de Diseño de Alto Nivel.

En esta fase se trabaja con los modelos de Diseño de Alto Nivel del Sistema Dentizt@ construidos en la fase anterior, ampliando los conceptos correspondientes a los casos de uso que se traten en el ciclo de desarrollo actual.

Las actividades de la fase de Diseño de Alto Nivel del Sistema Web Dentizt@ son las siguientes:

1. Definir Casos de Uso Esenciales en formato expandido.
2. Refinar los Diagramas de Casos de Uso.
3. Refinar el Modelo Conceptual.
4. Definir los Diagramas de Secuencia del Sistema Dentizt@.

### **3.1.1.1 Casos de Uso Expandidos**

Los casos de uso del Sistema Web Dentizt@ que se consideran los más importantes y que son los que más influencia tienen sobre al resto, son descritos a un nivel más detallado en el formato expandido. La principal diferencia con un caso de uso de alto nivel está en que incluye un apartado de Curso Típico de Eventos.

Para la construcción de los casos de uso de Dentizt@ el significado de cada apartado es como sigue:



- Caso de Uso: Nombre del Caso de Uso
- Actores: Lista de actores (agentes externos), indicando quién inicia el caso de uso. Los actores son normalmente roles que un ser humano desempeña, pero puede ser cualquier tipo de sistema.
- Propósito: Intención del caso de uso.
- Visión General: Repetición del caso de uso de alto nivel, o un resumen similar.
- Tipo: Se refiere a la priorización de casos de uso que identifiquemos, de los cuales pueden distinguirse:
  1. Primario, Secundario u Opcional.
    - Primarios. Representan los procesos principales, los más comunes.
    - Secundarios. Representan casos de uso menores, que van a necesitarse raramente.
    - Opcionales. Representan procesos que pueden no ser abordados en el presente proyecto.
  2. Esencial o Real.
    - Esencial. Es un caso de uso definido a nivel abstracto. En este ámbito, los casos de uso definidos a alto nivel son siempre esenciales por naturaleza, debido a su brevedad y abstracción.
    - Real. Describe concretamente el proceso en términos del diseño real, de la solución específica que se va a llevar a cabo. Se ajusta a un tipo de interfaz específica, y se baja a detalles como pantallas y objetos en las mismas.
- Referencias: Casos de uso relacionados y funciones del sistema que aparecen en los requisitos.
- Curso Típico de Eventos: Descripción de la interacción entre los actores y el sistema mediante las acciones numeradas de cada uno. Describe la secuencia más común de eventos, cuando todo va bien y el proceso se completa satisfactoriamente. En caso de haber alternativas con grado similar de probabilidad se pueden añadir secciones adicionales a la sección principal.



- Cursos Alternativos: Puntos en los que puede surgir una alternativa, junto con la descripción de la excepción.

La definición de los casos de uso en formato expandido describe los procesos más a fondo. Para el Sistema Web Dentizt@ los casos de uso en formato expandido se definen como sigue en las Tablas 3.1 a 3.8, tomando en cuenta que estos son los de mayor relevancia en el sistema por las funciones que en él representa.

Caso de uso:	Ingresar al sistema
Actores:	Profesor, Administrador-comunidad, Administrador-general
Propósito:	Ingresar a la página del Sistema Web Dentizt@
Resumen:	Un Profesor, Administrador-comunidad o Administrador-general debe teclear la página Web del sistema Web dentizt@ en una computadora que tenga acceso a Internet. Mostrará la página de inicio del sistema.
Tipo:	Primario-esencial

Tabla 3.1 Caso de Uso Ingresar al Sistema

Caso de uso:	Registro de Usuario a Dentizt@
Actores:	Profesor, Administrador-comunidad, Administrador-general
Propósito:	Obtener usuario y contraseña para que el actor ingrese a las comunidades del Sistema Web Dentizt@
Resumen:	Un Profesor, Administrador-comunidad o Administrador-general se registra en el sistema Web Dentizt@, proporcionando sus datos, así como el nombre de usuario y contraseña para ser parte del sistema y así poder tener los privilegios que brinda el mismo.
Tipo:	Primario-esencial

Tabla 3.2 Caso de Uso Registro de Usuario a Dentizt@



Caso de uso:	Ingresar a Comunidad
Actores:	Profesor, Administrador-comunidad, Administrador-general
Propósito:	Ingresar a la comunidad del usuario proporcionado
Resumen:	Un Profesor, Administrador-comunidad o Administrador-general ingresa su usuario y contraseña para que el Sistema Web Dentizt@ proporcione los privilegios y comunidades correspondientes a dicho usuario
Tipo:	Primario

Tabla 3.3 Caso de Uso Ingresar a Comunidad

Caso de uso:	Administrar Materiales
Actores:	Administrador-comunidad, Administrador-general
Propósito:	Publicar, modificar y eliminar materiales de la comunidad
Resumen:	Un Administrador-comunidad o Administrador-general puede publicar, modificar o eliminar los materiales de su comunidad
Tipo:	Primario

Tabla 3.4 Caso de Uso Administrar Materiales

Caso de uso:	Consultar Materiales
Actores:	Profesor, Administrador-comunidad, Administrador-general
Propósito:	Consultar Materiales
Resumen:	Un Profesor, Administrador-comunidad o Administrador-general puede consultar los materiales de su comunidad
Tipo:	Secundario

Tabla 3.5 Caso de Uso Consultar Materiales



Caso de uso:	Ingresar al Foro
Actores:	Profesor, Administrador-comunidad, Administrador-general
Propósito:	Ingresar al foro para comunicarse entre integrantes de las comunidades
Resumen:	Un Profesor, Administrador-comunidad o Administrador-general puede ingresar al foro para comunicarse con otros integrantes de la comunidad
Tipo:	Secundario

Tabla 3.6 Caso de Uso Ingresar al Foro

Caso de uso:	Administrar Comunidades
Actores:	Administrador-comunidad, Administrador-general
Propósito:	Crear o eliminar comunidades y aceptar o rechazar peticiones de usuarios
Resumen:	Un Administrador-comunidad o Administrador-general puede eliminar o crear comunidades conforme a sus necesidades, así como aceptar o rechazar usuarios
Tipo:	Primario

Tabla 3.7 Caso de Uso Administrar Comunidades

Caso de uso:	Dar Soluciones Técnicas al Sistema
Actores:	Administrador-general
Propósito:	Tener el Sistema Web Dentit@ funcionando correctamente
Resumen:	Un Administrador-general recibirá peticiones de fallas en el sistema y deberá resolverlos lo más pronto posible
Tipo:	Secundario

Tabla 3.8 Caso de Uso Dar Soluciones Técnicas al Sistema



### 3.1.1.2 Modelo Conceptual

Como parte de la investigación sobre el dominio del problema se deben identificar los conceptos que lo conforman. Para representar estos conceptos se usa un Diagrama de Estructura Estática de UML, que se denomina Modelo Conceptual.

En el Modelo Conceptual de Dentizt@ se tiene una representación de conceptos del mundo real, no de componentes de software. El objetivo de la creación de un Modelo Conceptual es aumentar la comprensión del problema. Por lo que es importante remarcar que en la construcción de este modelo se trata de incluir todos los conceptos posibles con el fin de no quedarse corto y olvidar algún concepto importante.

Para el Sistema Web Dentizt@ los modelos conceptuales son los que se presentan de la Figura 3.1 a la Figura 3.8.

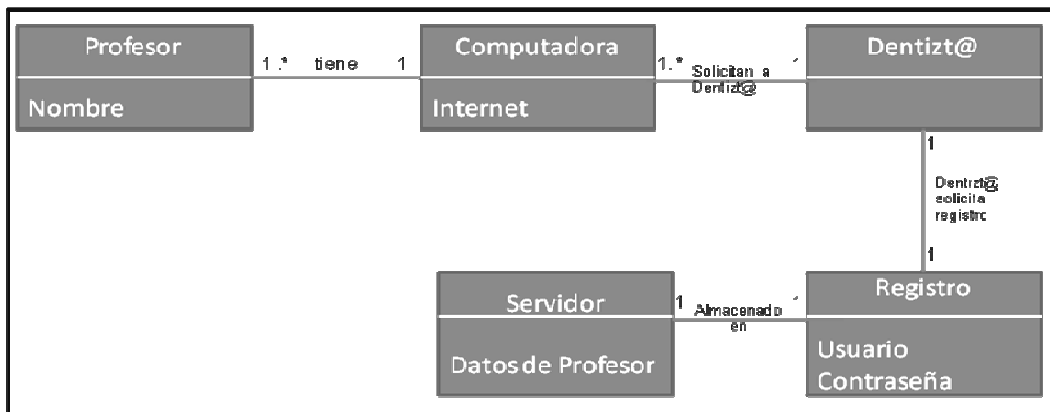


Figura 3.1 Modelo conceptual del caso de uso Registro de Usuario a Dentizt@

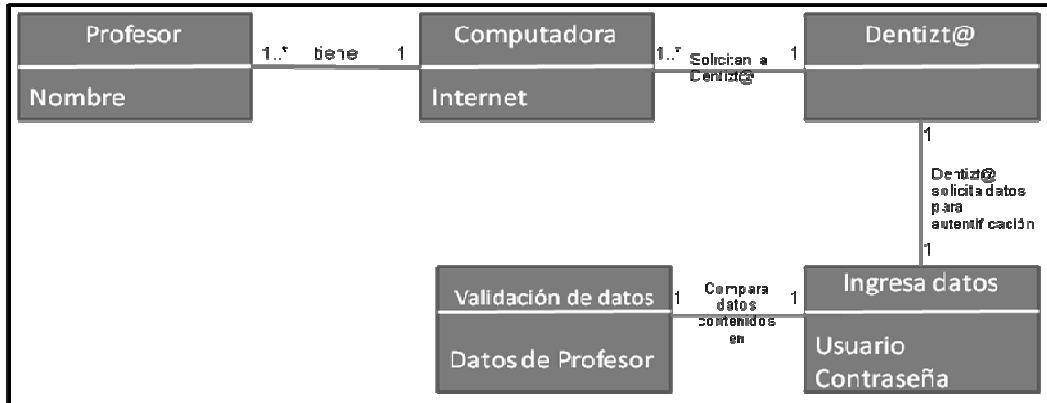


Figura 3.2 Modelo conceptual del caso de uso Ingresar al Sistema

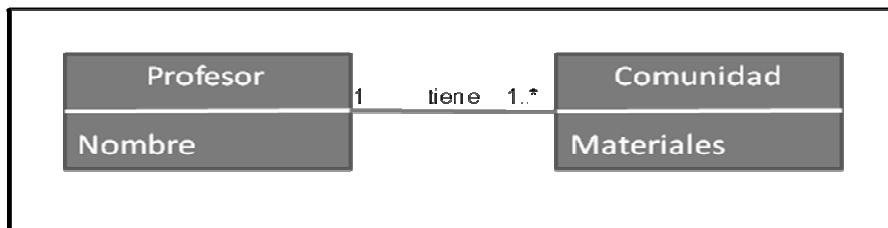


Figura 3.3 Modelo conceptual del caso de uso Ingresar a Comunidad

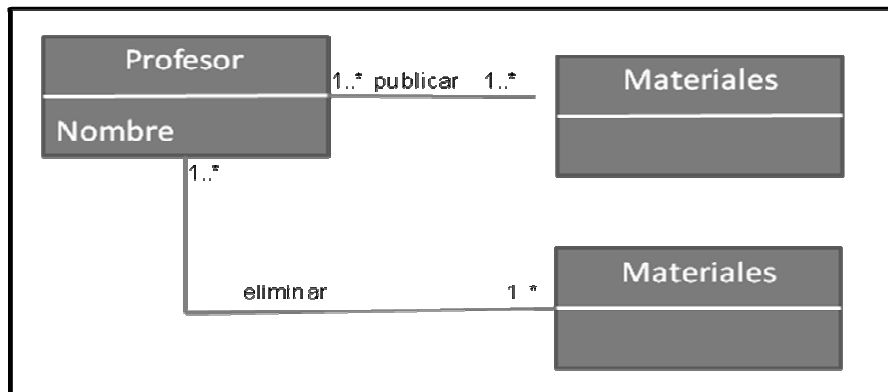


Figura 3.4 Modelo conceptual del caso de uso Administrar Materiales



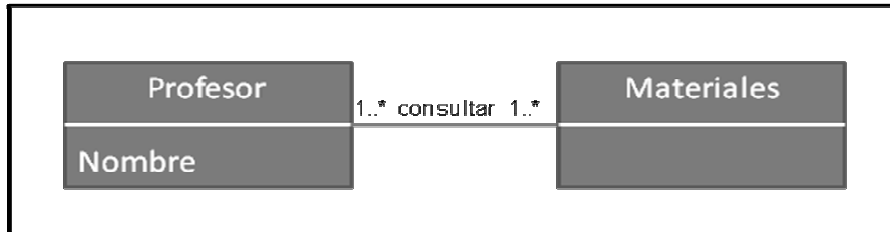


Figura 3.5 Modelo conceptual del caso de uso Consultar Materiales

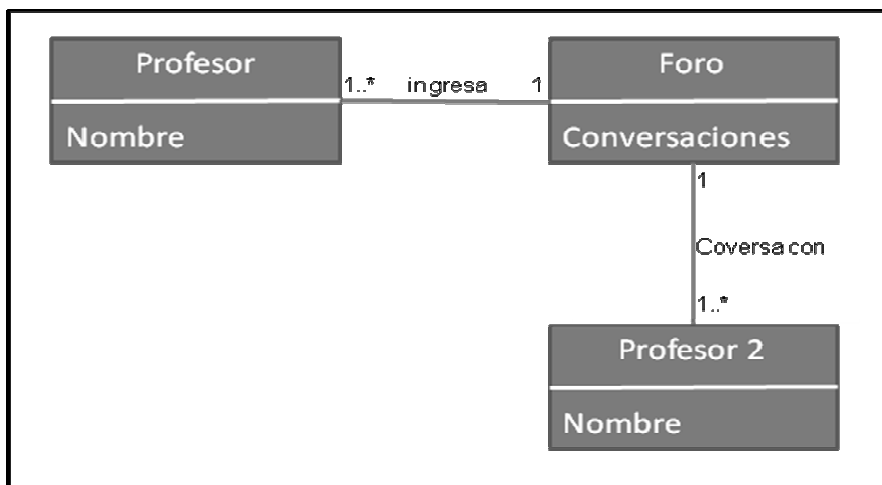


Figura 3.6 Modelo conceptual del caso de uso Ingresar al Foro

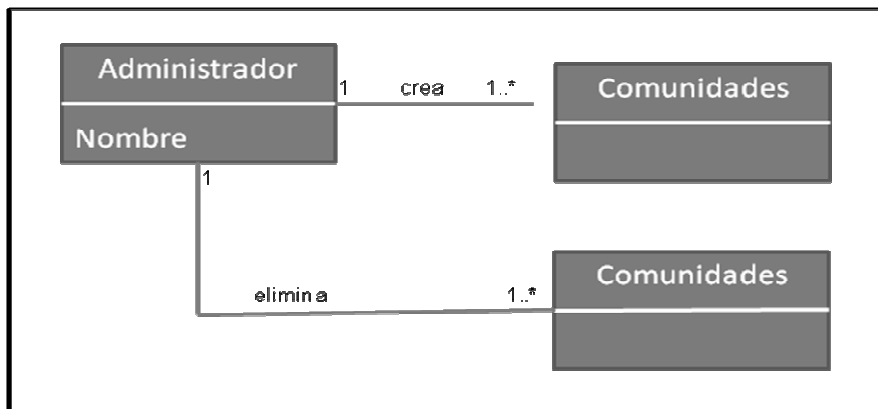


Figura 3.7 Modelo conceptual del caso de uso Administrar Comunidades

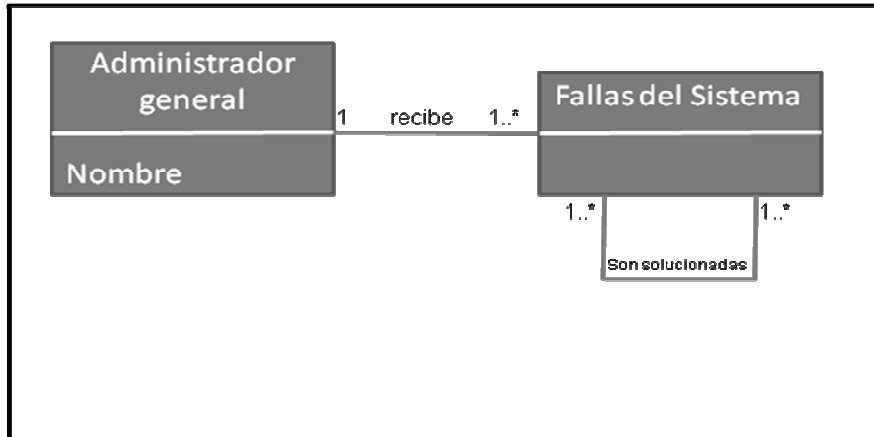


Figura 3.8 Modelo conceptual del caso de uso Dar Soluciones Técnicas al Sistema

### 3.1.1.3 Diagramas de Secuencia del Sistema

Además de investigar sobre los conceptos del Sistema Dentizt@ y su estructura, también se investigó en el Diseño de Alto Nivel sobre el comportamiento del mismo, visto éste como una caja negra. Una parte de la descripción del comportamiento del sistema se realizó mediante los Diagramas de Secuencia.

#### 3.1.1.3.1 Construcción de Diagramas de Secuencia del Sistema Web Dentizt@

Para la construcción de los Diagramas de Secuencia del Sistema Dentizt@, se siguieron los siguientes pasos:

1. Representamos al sistema como un objeto.
2. Identificamos los actores que directamente operan con el sistema, y dibujamos una línea para cada uno de ellos.
3. Partiendo del texto del curso típico de eventos del caso de uso, identificamos los eventos (externos) del sistema que cada actor genera y los representamos en el diagrama.

En resumen, los diagramas de secuencia nos muestran gráficamente los eventos que influyen de los actores al sistema.



A continuación se muestran los diagramas de secuencia para el Sistema Web Dentizt@ de la Figura 3.10 a la Figura 3.17.

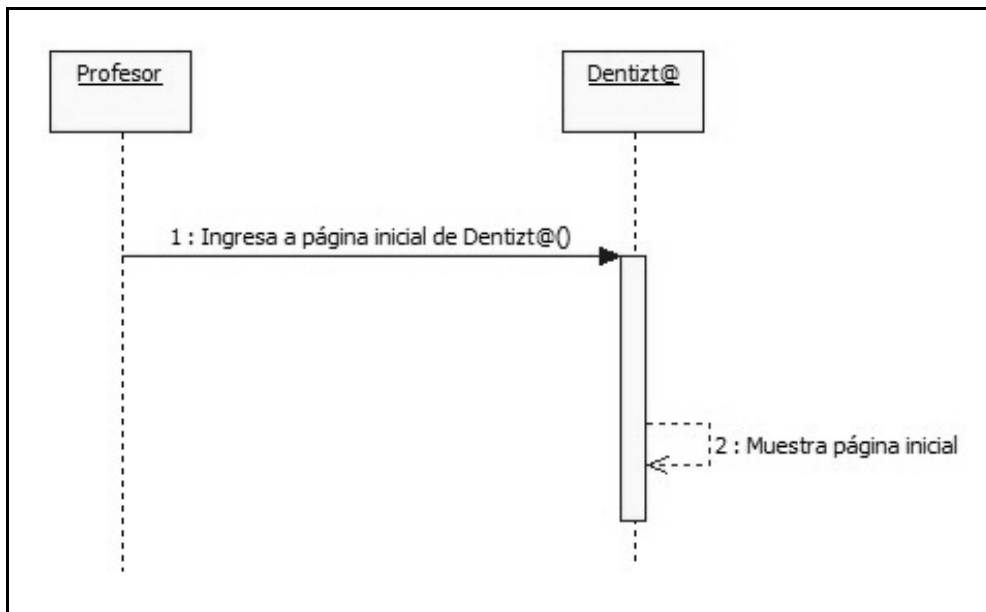


Figura 3.10 Diagrama de secuencia para Ingresar al Sistema Web Dentizt@

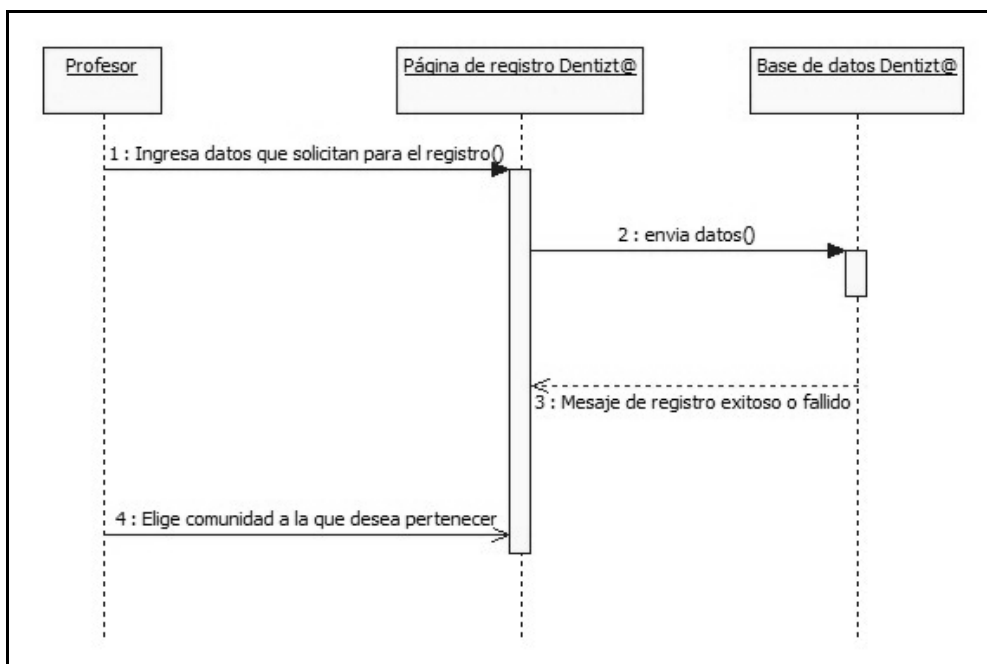


Figura 3.11 Diagrama de secuencia para Registro de Usuario al Sistema

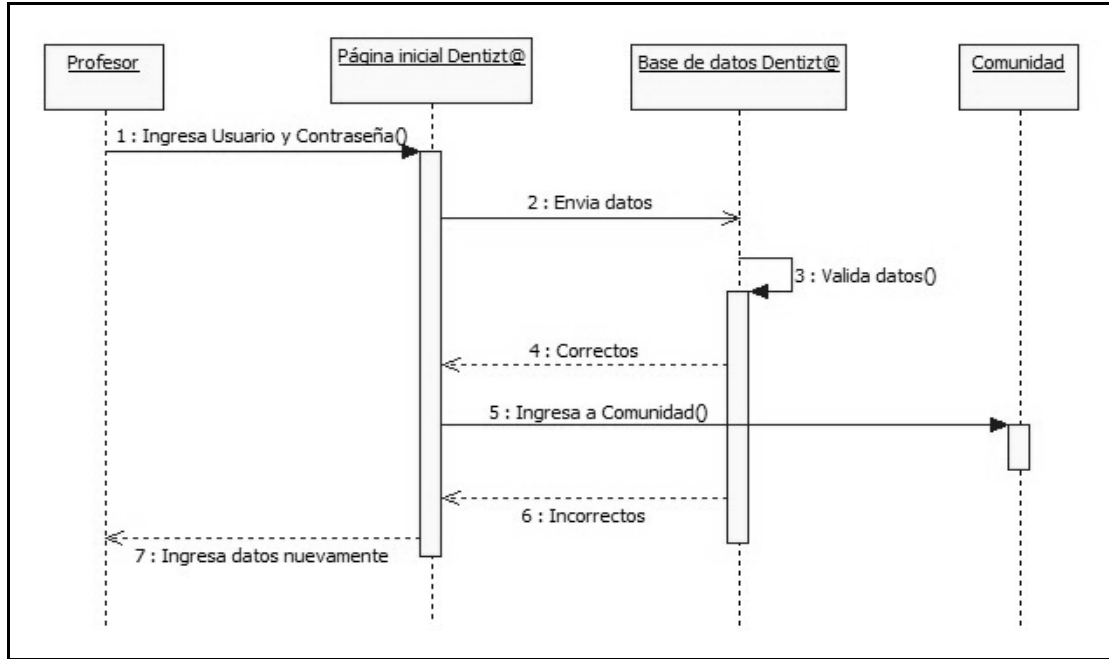


Figura 3.12 Diagrama de secuencia para Ingresar a Comunidad

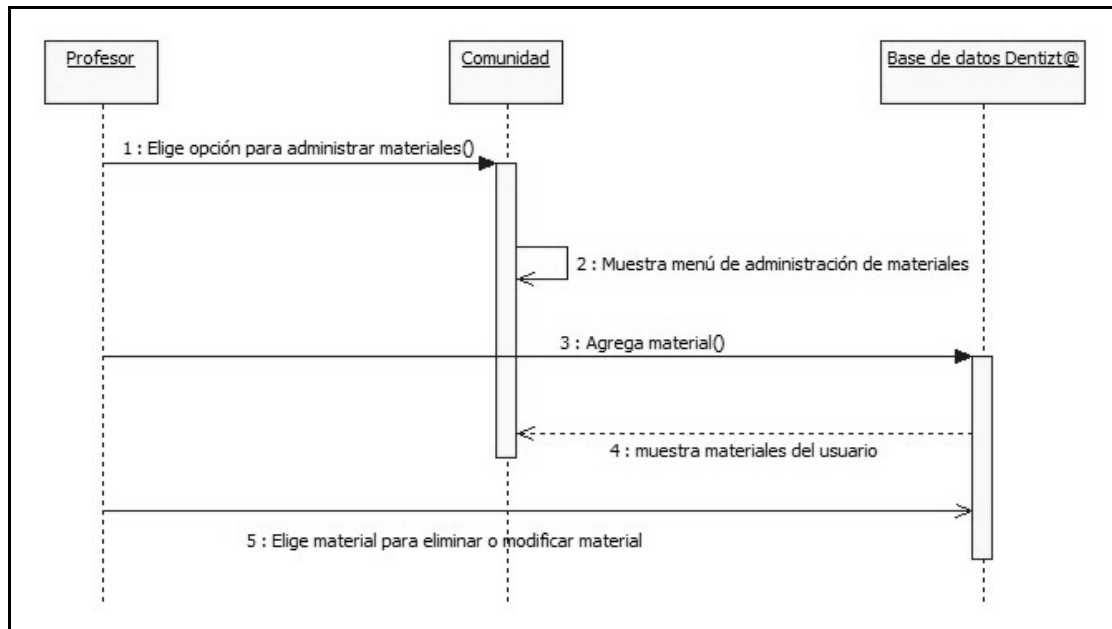


Figura 3.13 Diagrama de secuencia para Administrar Materiales

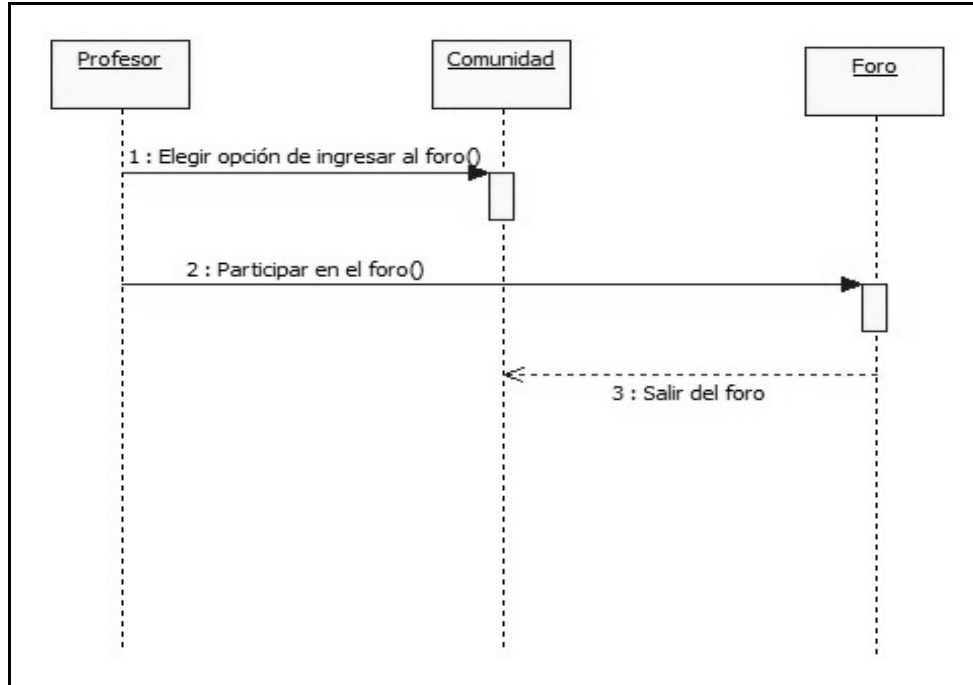


Figura 3.14 Diagrama de secuencia para Ingresar al Foro

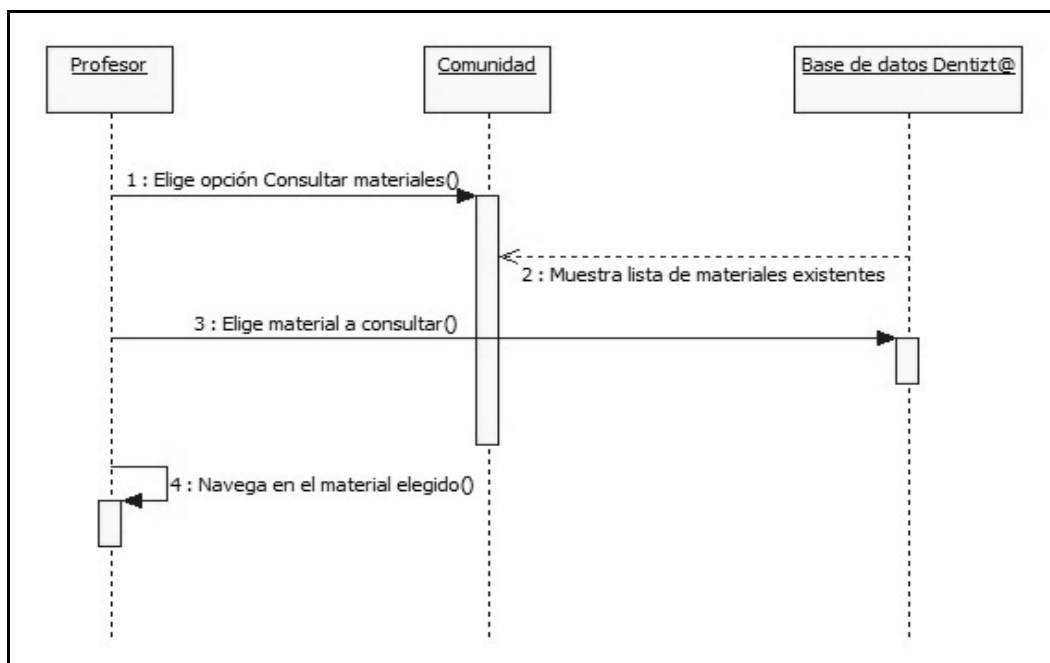


Figura 3.15 Diagrama de secuencia para Consultar Materiales

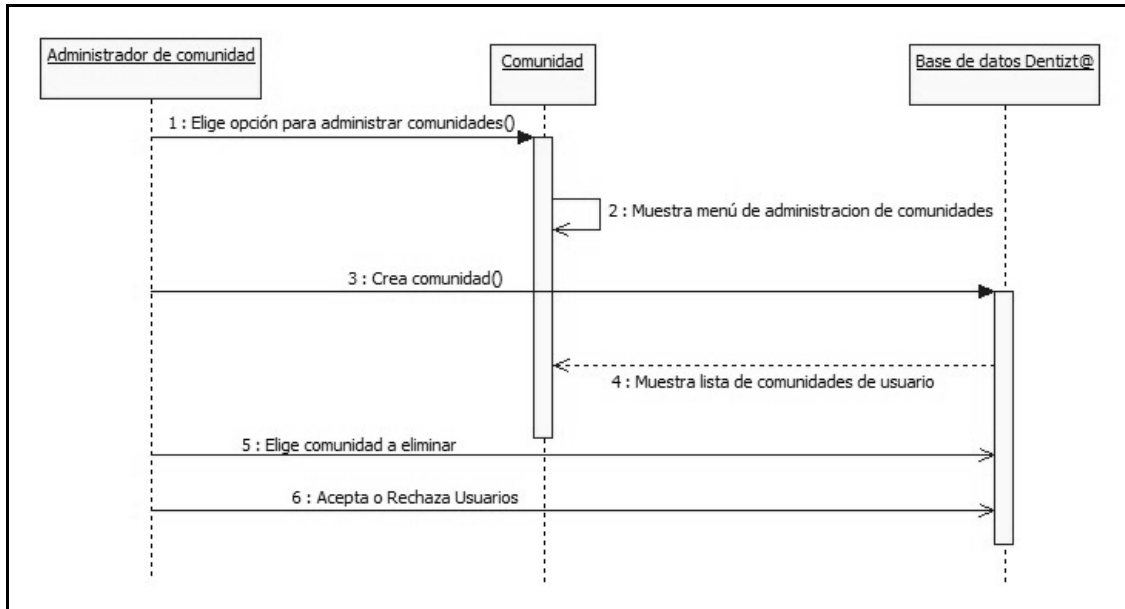


Figura 3.16 Diagrama de secuencia para Administrar Comunidades

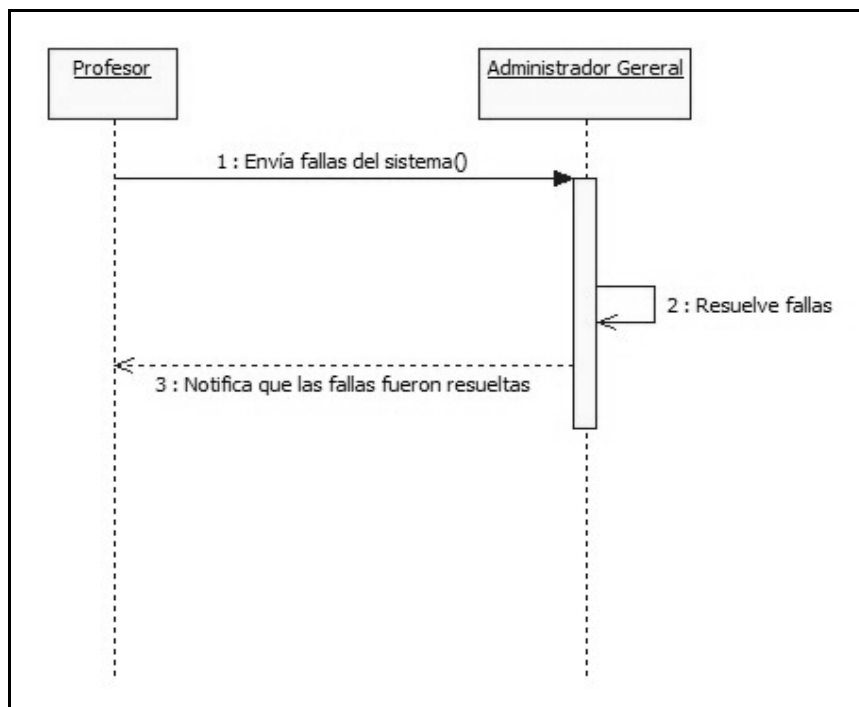


Figura 3.17 Diagrama de secuencia para Dar soluciones técnicas al sistema



### 3.1.2 Diseño de Bajo Nivel

En la fase de Diseño de Bajo Nivel se creó una solución a nivel lógico para satisfacer los requisitos, basándose en el conocimiento reunido en la fase de Diseño de Alto Nivel. Los modelos más importantes en esta fase son el Diagrama de Clases de Diseño y los Diagramas de Interacción, que se realizaron en paralelo y que definen los elementos que forman parte del sistema orientado a objetos que se va a construir (clases y objetos).

Las actividades que se realizaron en la etapa de Diseño de Bajo Nivel fueron las siguientes:

1. Definimos Informes e Interfaz de Usuario.
2. Refinamos la Arquitectura del Sistema.
3. Definimos los Diagramas de Interacción.
4. Definimos el Diagrama de Clases de Diseño.
5. Definimos el Esquema de Base de Datos.

#### 3.1.2.1 Diagramas de Interacción

Los Diagramas de Interacción nos muestran el intercambio de mensajes entre instancias del modelo de clases para cumplir con las post-condiciones establecidas en un contrato.

Hay dos clases de Diagramas de Interacción:

1. Diagramas de Colaboración.
2. Diagramas de Secuencia.

La creación de los Diagramas de Interacción del sistema Dentizt@ fue una de las actividades más importantes en el desarrollo porque al construirlos se tomaron las decisiones clave acerca del funcionamiento del sistema. La creación de estos diagramas, por tanto, ocupó un porcentaje significativo en el esfuerzo dedicado al proyecto entero.



### 3.1.2.2 Diagramas de Colaboración

Los diagramas de colaboración nos describen las interacciones entre los objetos en un formato de grafo o red. A continuación se muestran los diagramas de colaboración para el Sistema Web Dentizt@ de la Figura 3.19 a 3.26.

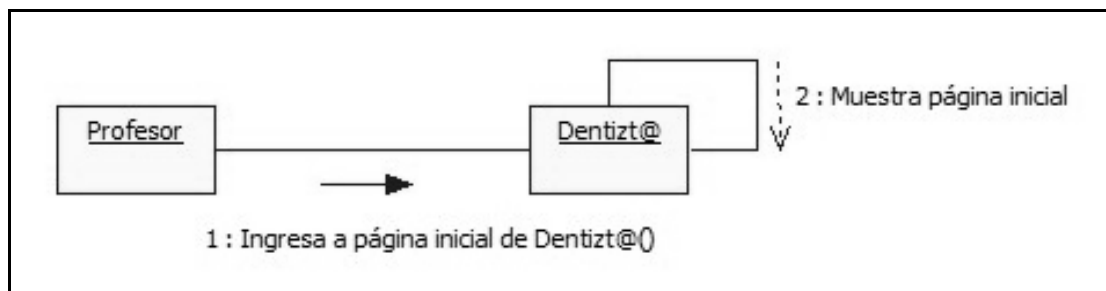


Figura 3.19 Diagrama de colaboración para Dar Ingresar al Sistema Web Dentizt@

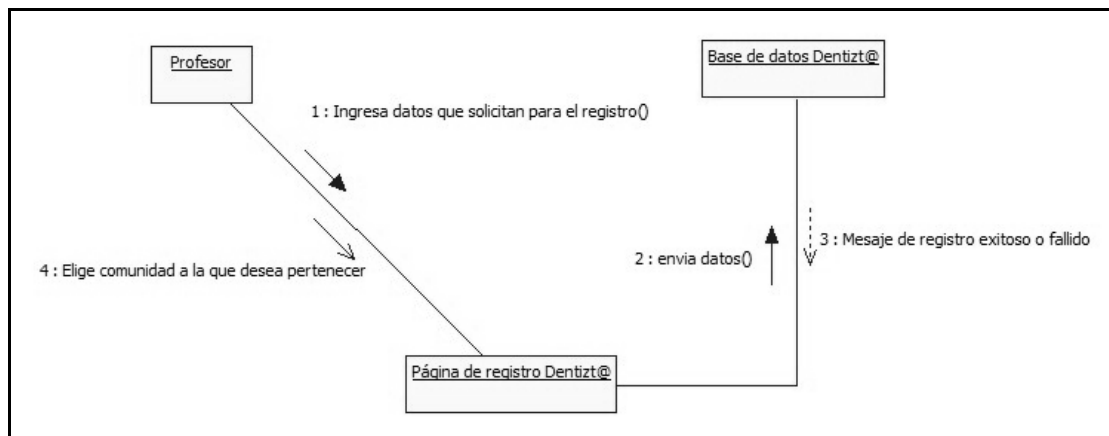


Figura 3.20 Diagrama de colaboración para Registro de Usuario al Sistema



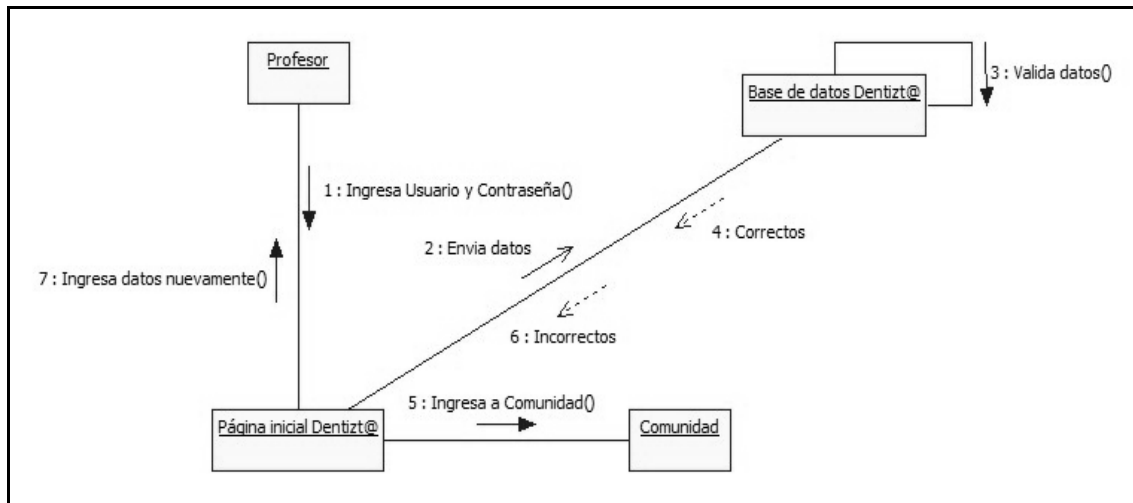


Figura 3.21 Diagrama de colaboración para Ingresar a Comunidad

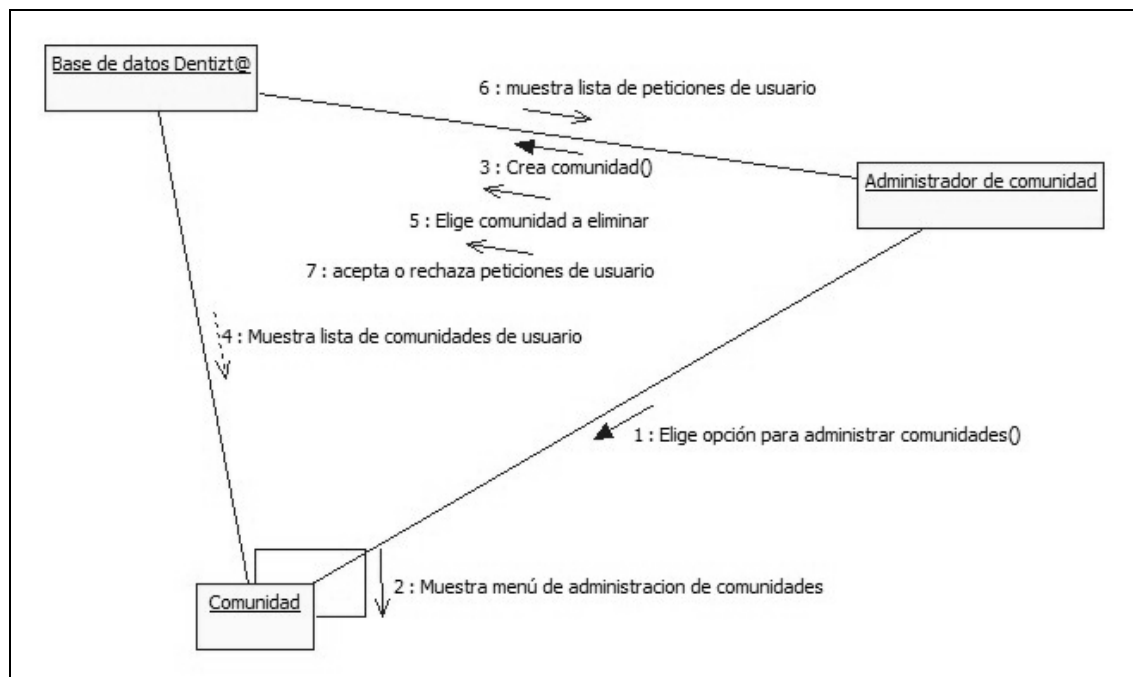


Figura 3.22 Diagrama de colaboración para Administrar Materiales

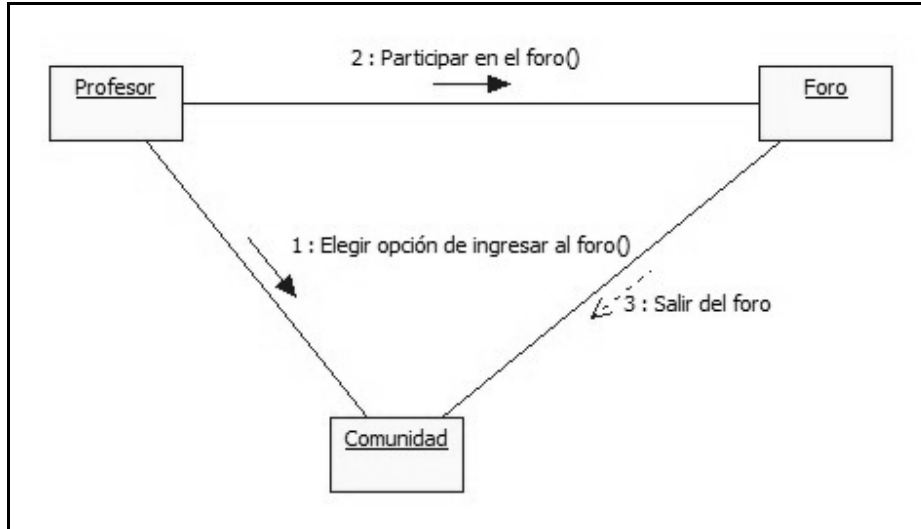


Figura 3.23 Diagrama de colaboración para Ingresar al Foro

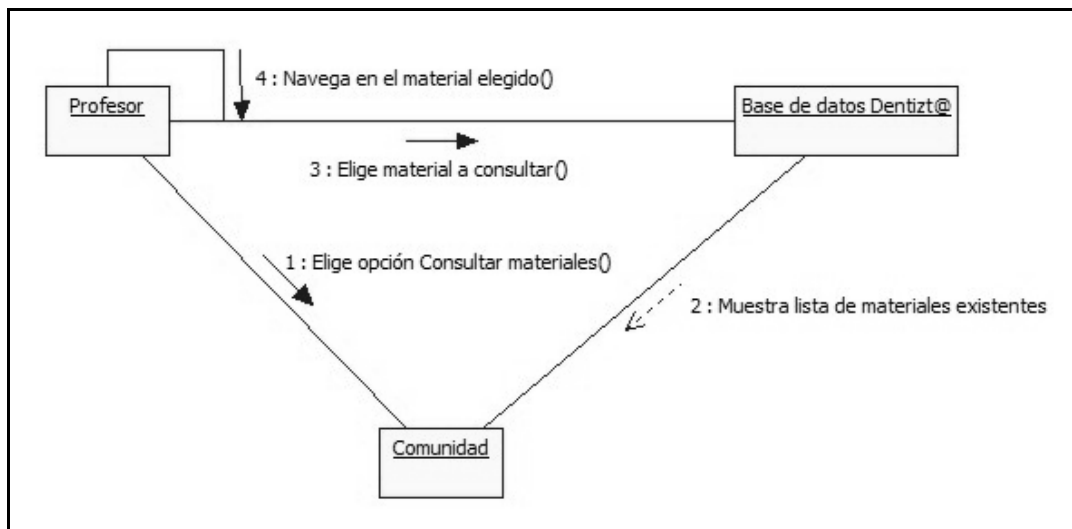


Figura 3.24 Diagrama de colaboración para Consultar Materiales

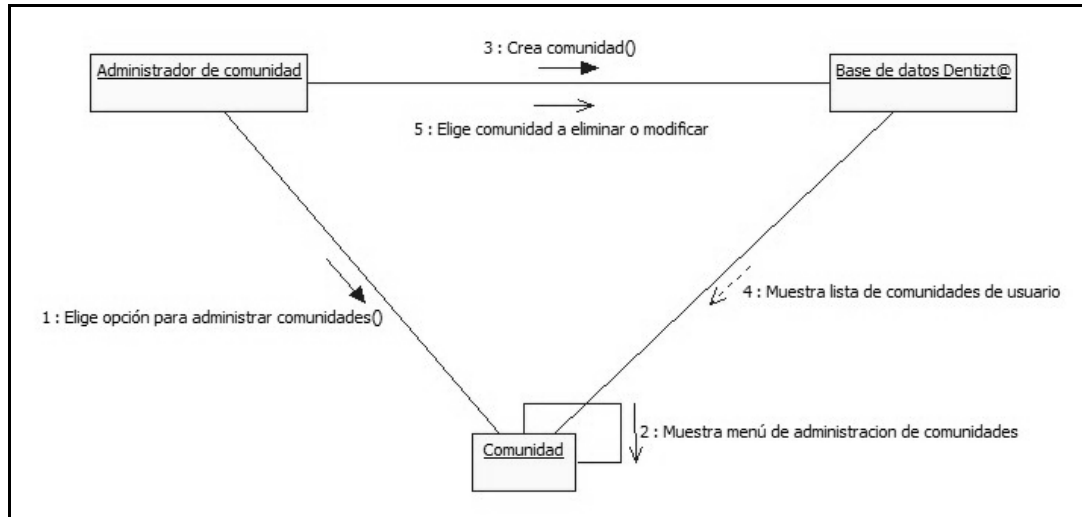


Figura 3.25 Diagrama de colaboración para Administrar Comunidades

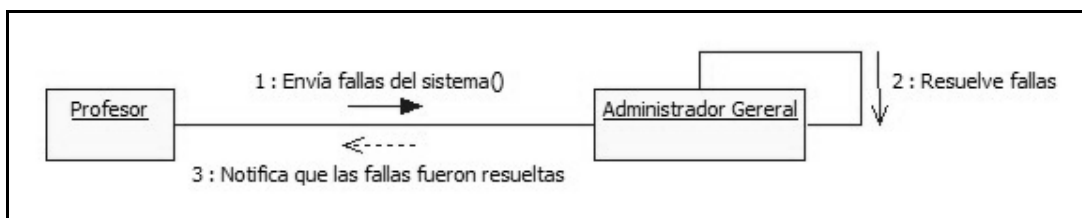


Figura 3.26 Diagrama de colaboración para Dar soluciones técnicas al sistema

### 3.1.2.3 Diagrama de Clases de Diseño

El Diagrama de Clases de Diseño del Sistema Web Dentizt@ incluye la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Navegabilidad.
- Dependencias.

A diferencia del Modelo Conceptual, el Diagrama de Clases de Diseño muestra definiciones de entidades de software más que conceptos del mundo real.



### 3.1.2.3.1 Construcción de un Diagrama de Clases de Diseño

Los diagramas de clases de diseño describen gráficamente las especificaciones de las clases de software de las interfaces del Sistema.

A continuación se muestran los diagramas de clases de diseño para el Sistema Web Dentizt@ de la Figura 3.28 a la Figura 3.30.

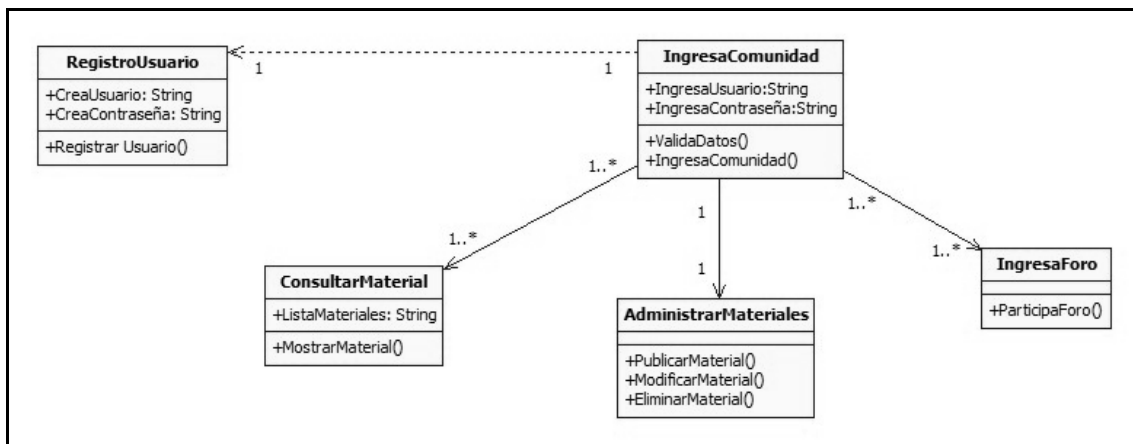


Figura 3.28 Diagrama de clases para el Profesor

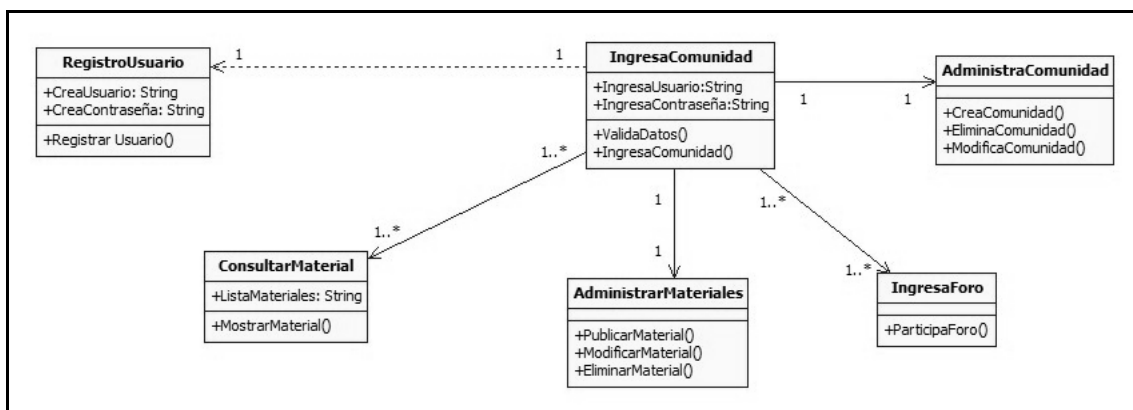


Figura 3.29 Diagrama de clases para el Administrador-Comunidad

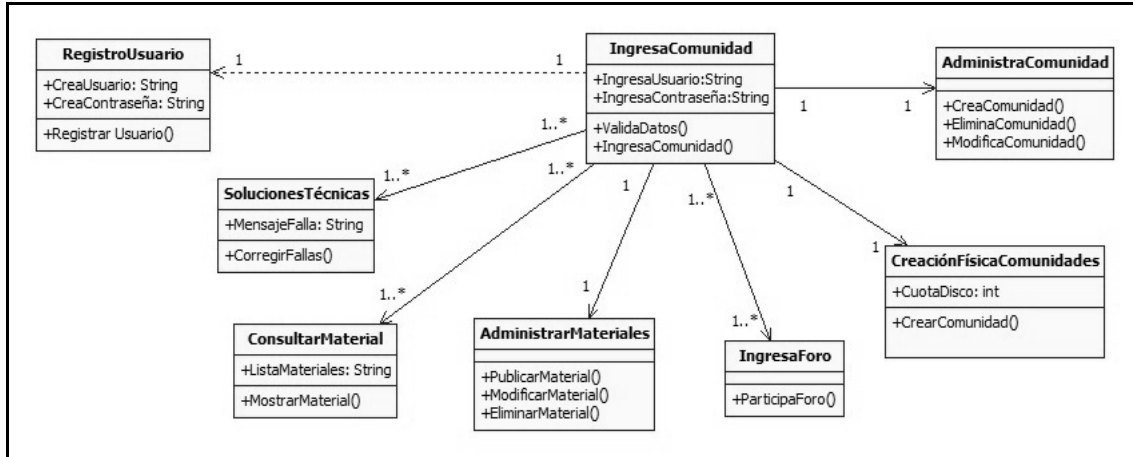


Figura 3.30 Diagrama de clases para el Administrador-General

### 3.1.3 Implementación y Puesta a Punto del Sistema Web Dentizt@

La implementación es la etapa dentro, de la fase de Construcción, en donde se lleva lo especificado en el Diseño de Bajo Nivel del Sistema Web Dentizt@ al lenguaje de programación PHP, previamente elegido por sus características para el desarrollo en la Web.

Además es aquí en donde se desarrolla la parte lógica de nuestro Sistema, pues se realiza el diseño de las bases de datos que se utilizarán para el registro de usuarios, comunidades y materiales que en conjunto proporcionarán al usuario una buena experiencia en la navegación por nuestro Sistema.

Por último, se realizarán las pruebas en donde se comprobará la funcionalidad del Sistema y el fácil manejo, para lo cual se dispondrá de las evaluaciones de una muestra de usuarios.

#### 3.1.3.1 Creación del Sistema en el Lenguaje de Programación

Una vez que concluimos los diagramas de clases del diseño, disponemos de suficientes detalles para generar un código que utilizaremos en la capa del



dominio de los objetos. Para que esto sea posible, lo que hasta ahora hemos creado con ayuda del UML en la fase de diseño, es decir, los diagramas de colaboración y los de clases del diseño, servirán de entrada en el proceso de generación del código.

Hay que tomar en cuenta que para reducir el riesgo y aumentar la probabilidad de conseguir una aplicación adecuada para nuestro sistema, el desarrollo debe basarse en un buen modelado de análisis y diseño antes de iniciar la codificación.

Para el Sistema Web Dentizt@, la parte esencial se determinó de mejor manera en una investigación formal y en el proceso de diseño, así mismo se evitó el apresurarse a codificar sin realizar antes un análisis y un diseño previo. Evitando así, que nuestro sistema pueda ser difícil de entender, ampliar o darle mantenimiento. De esta manera, una ventaja del análisis, del diseño y de la programación, es que ofrece una guía completa de principio a fin para realizar la codificación a partir de los requerimientos.

A continuación se muestran partes de código en lenguaje de programación php del Sistema Web Dentizt@, que fue generado a partir de todos los diagramas construidos en la fases de diseño y análisis; lo que permitió tener claro las características del sistema y generar funciones que satisficieran nuestras necesidades primero como programadores y posteriormente como usuarios.

- **Código que introduce los datos proporcionados por el usuario en su registro del Sistema Web Dentizt@.**

```
<?php
//Función que permite la conexión con la Base de Datos
include("conex.phtml");
$link=Conectarse();
```



/\*Se declaran las variables que contendrán los valores proporcionados por los usuarios\*/

```
$nom_Usu=$_POST['nom_Usu'];
```

```
$app_Usu=$_POST['app_Usu'];
```

```
$apm_Usu=$_POST['apm_Usu'];
```

```
$mail_Usu=$_POST['mail_Usu'];
```

```
$tel_Usu=$_POST['tel_Usu'];
```

```
$ocup_Usu=$_POST['ocup_Usu'];
```

```
$tipo_Usu=$_POST['tipo_Usu'];
```

```
$login_Usu=$_POST['login_Usu'];
```

```
$pass_Usu=$_POST['pass_Usu'];
```

```
// Se inicia una sesión
```

```
session_start();
```

```
if ($_POST["textoDelmagen"]==$_SESSION["cadena"])
```

```
// Se insertan los valores proporcionados por el usuario a la Base de datos
```

```
{
```

```
    mysql_query("insert into tbPreregistro
```

```
    (nom_Usu,app_Usu,apm_Usu,mail_Usu,tel_Usu,ocup_Usu,tipo_Usu,login_Usu  
    ,pass_Usu) values
```

```
    ('$nom_Usu','$app_Usu','$apm_Usu','$mail_Usu','$tel_Usu','$ocup_Usu','$tipo_Usu','$login_Usu','$pass_Usu')",$link);
```

```
    header("Location: index2.php");
```

```
}
```

```
else
```

```
{
```

```
    echo ("TUS DATOS NO HAN SIDO ENVIADOS CON EXITO");
```

```
}
```

```
?>
```



- **Código que permite el acceso al Sistema Web Dentizt@, identificando el tipo de usuario que ingresa.**

```
<?php
session_start ();
include("conex.phtml"); //hace la conexion con la base de datos
$link=Conectarse();

//DECLARACION DE VARIABLES ASIGNANDOLE EL VALOR DEL FORMULARIO
$username=$_POST['username'];
$password=$_POST['password'];
$_SESSION["username"]=$username;
// Para proteger de MySQL injection
$username = stripslashes($username);
$password = stripslashes($password);
$username = mysql_real_escape_string($username);
$password = mysql_real_escape_string($password);

//CONSULTA PARA NIVEL DE USUARIO CON USUARIO Y PASSWORD IGUALES
$query="SELECT tipo_Usu FROM tbUsuarios WHERE login_Usu='$username'
and pass_Usu='$password'";
$result=mysql_query($query,$link);
$count=mysql_num_rows($result);
if(mysql_num_rows($result) == 0)
{ // si no hay resultados
header("location: ingreso2.phtml");
}
$count2=mysql_result($result,0,"tipo_Usu");
if($count==1 && $count2=="Administrador")
```





```
{
    session_register("myusername");
    header("location:menu_Admcom.phtml");
}
elseif($count==1 && $count2=="Admgral")
{
    session_register("myusername");
    header("location:menu_Super.phtml");
}
elseif($count==1 && $count2=="Usuario")
{
    session_register("myusername");
    header("Location: menu_Usuario.phtml");
}
?>
```

- **Código que permite al administrador de comunidad publicar un material.**

```
<?php
include("conex.phtml");
$link=Conectarse();
$descrip_Mat=$_POST['descrip_Mat'];
$id_Com=$_GET['id_Com'];
$consulta="SELECT id_AdmCom FROM tbRegistroCom
        WHERE id_Com='$id_Com'";
$ejecutaConsulta=mysql_query($consulta,$link);
$count=mysql_fetch_assoc($ejecutaConsulta);
$id_AdmCom=$count['id_AdmCom'];

$directorio_archivo = "/opt/lampp/htdocs/Dentizta/doc_Ser/";
$nombre = strstr($_HTTP_POST_FILES['archivo']['name'],
```



```
"ñ'ÁÉÍÓÚÀÈÌÒÙáéíóúäëïöüàèìòù ", "n-AEIOUAEIOUaeiouaeiouaeiou_");
$tamano = $HTTP_POST_FILES['archivo']['size'];
$ext_Mat = $HTTP_POST_FILES['archivo']['type'];

if (move_uploaded_file($HTTP_POST_FILES['archivo']['tmp_name'],
    $directorio_archivo.$nombre))
{
    mysql_query("insert into tbMateriales
        nom_Mat,tam_Mat,ext_Mat,descrip_Mat) values
        ('$nombre','$tamano','$ext_Mat','$descrip_Mat')",
        $link);
    $consulta2="SELECT id_Mat FROM tbMateriales
        WHERE nom_Mat='$nombre' AND tam_Mat='$tamano' AND
        ext_Mat='$ext_Mat' AND descrip_Mat='$descrip_Mat'";
    $ejecutaConsulta2=mysql_query($consulta2,$link);
    $count2=mysql_fetch_assoc($ejecutaConsulta2);
    $id_Mat=$count2['id_Mat'];
    mysql_query("insert into tbRegistroMat (id_AdmCom,id_Mat)
        values ('$id_AdmCom','$id_Mat')",$link);
    mysql_query("insert into tbCM (id_Com,id_Mat) values
        ('$id_Com','$id_Mat')",$link);
    header("Location: men_up_archivo.phtml");
}
else
{
    header("Location: nomen_up_archivo.phtml");
}
?>
```



### 3.1.3.2 Construcción de la Base de Datos

Como producto del análisis en los requerimientos del Sistema Web Dentizt@ determinamos que requería de la implementación de una base de datos, con la finalidad de tener un conjunto de información organizada a nuestra disposición y poseer el control en el registro de los usuarios, comunidades y materiales con los que contará y de esta forma hacer eficiente su operación.

Para Dentizt@, el desarrollo de la base de datos toma un enfoque en el cual se transforman los requerimientos de información en una base de datos operacional.

#### Requerimientos

Como primer paso, se realizó el *Análisis* de los requerimientos de información de cada uno de los usuarios que operarán el sistema, los cuales fueron proporcionados por ellos mismos y con esto se desarrolló un Modelo Entidad-Relación para expresar esos requerimientos.

El Sistema Web Dentizt@ necesita tener la información de cada uno de sus *Usuarios*, nombre, apellidos, correo electrónico, teléfono y ocupación, los cuales deben crear su nombre de usuario y una contraseña para su acceso a Dentizt@, a cada usuario se le asignará un número único de identificación en el sistema.

Dentizt@ está dividida en *Comunidades*. Cada comunidad tiene un número único de identificación, un nombre y su descripción. Solo algunos usuarios podrán crear comunidades lo cuales se denominarán *Administradores de Comunidad* y serán identificados con un número único. Se requiere conocer quien es el responsable de cada una de las comunidades así como la fecha de creación.

Cada *Administrador de Comunidad* tiene la posibilidad de publicar *Materiales Educativos* en la comunidad de la que son responsables, los cuales deben tener un número único de identificación, nombre y descripción. Se necesita tener



identificada la comunidad a la que pertenece cada uno de los materiales así como su fecha de publicación y el responsable de dicho material.

Todos los usuarios podrán registrarse a una o varias comunidades como miembros de estas, y podrán consultar los materiales que en ellas están publicadas.

### Modelo Entidad-Relación del Sistema Web Dentizt@

En el diseño de la base de datos de Dentizt@ se mapearon los requerimientos de información y fueron reflejados en un modelo E-R, el cual se muestra a continuación en la figura 3.31.

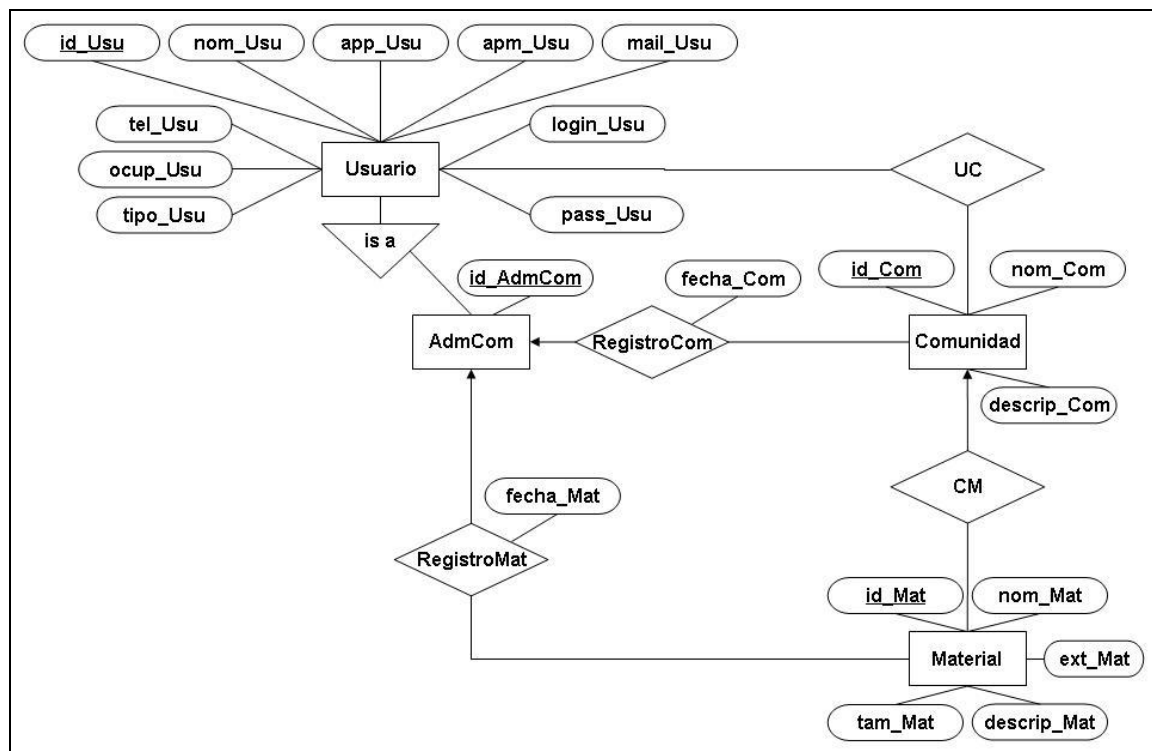


Figura 3.31 Modelo Entidad-Relación de la Base de Datos Dentizt@



En el diagrama se observan que para nuestro diseño existen 4 entidades las cuales se consideraron por su relevancia en el registro de la información, estas son *Usuario*, *Administrador de Comunidad*, *Comunidad* y *Material*. Dichas entidades presentan relaciones, las cuales se les ha dado el nombre de *Relación Registro de Comunidad (RegistroCom)*, *Relación Registro de Material (RegistroMat)*, *Relación Usuario-Comunidad (UC)* y *Relación Comunidad-Material (CM)*. Unidas a través de sus atributos y en especial de aquellos denominados llaves primarias, atributos que definen a cada entidad de manera única.

En la figura 3.32 se muestra también el diseño de las tablas para la base de datos y la relación entre ellas de manera gráfica, lo que ayuda a comprender de otra manera lo antes mencionado.

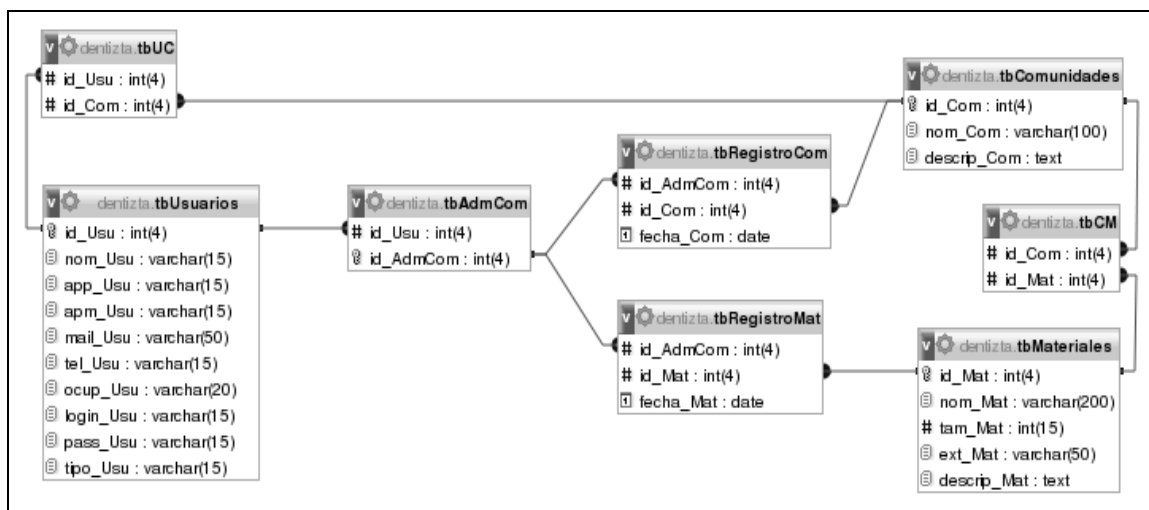


Figura 3.32 Diseño de las Tablas de la Base de Datos Dentizt@

Se construyeron 8 tablas las cuales se presentan a continuación en las Tablas 3.9 a 3.16, en donde se describen los campos que cada una contiene.



### tbUsuarios

Campo	Tipo	Nulo	Descripción
id_Usu	int(4)	No	Identificador único del usuario.
nom_Usu	varchar(15)	No	Nombre del usuario.
app_Usu	varchar(15)	No	Apellido paterno del usuario.
apm_Usu	varchar(15)	No	Apellido materno del usuario.
mail_Usu	varchar(50)	No	Correo electrónico del usuario.
tel_Usu	varchar(15)	No	Teléfono del usuario.
ocup_Usu	varchar(20)	No	Ocupación del usuario.
login_Usu	varchar(15)	No	Nombre de usuario del usuario.
pass_Usu	varchar(15)	No	Contraseña del usuario.
tipo_Usu	varchar(15)	No	Tipo de usuario.

Tabla 3.9 Tabla de Usuarios

### tbAdmCom

Campo	Tipo	Nulo	Descripción	Enlaces a
id_Usu	int(4)	No	Identificador único del usuario.	tbusuarios → id_Usu
id_AdmCom	int(4)	No	Identificador único del administrador de comunidad.	

Tabla 3.10 Tabla de Administradores de Comunidades

### tbComunidades

Campo	Tipo	Nulo	Descripción
id_Com	int(4)	No	Identificador único de la comunidad.
nom_Com	varchar(100)	No	Nombre de la comunidad.
descrip_Com	text	No	Descripción de la comunidad.

Tabla 3.11 Tabla de Comunidades



### tbMateriales

Campo	Tipo	Nulo	Descripción
id_Mat	int(4)	No	Identificador único del material.
nom_Mat	varchar(200)	No	Nombre del material.
tam_Mat	int(15)	No	Tamaño del material.
ext_Mat	varchar(50)	No	Tipo de extensión del material.
descrip_Mat	text	No	Descripción del material.

Tabla 3.12 Tabla de Materiales

### tbRegistroCom

Campo	Tipo	Nulo	Descripción	Enlaces a
id_AdmCom	int(4)	No	Identificador único del administrador de comunidad.	tbadmcom → id_AdmCom
id_Com	int(4)	No	Identificador único de la comunidad.	tbcomunidades → id_Com
fecha_Com	date	No	Fecha de registro a la comunidad.	

Tabla 3.13 Tabla de Registro de Comunidades

### tbRegistroMat

Campo	Tipo	Nulo	Descripción	Enlaces a
id_AdmCom	int(4)	No	Identificador único del administrador de comunidad.	tbadmcom → id_AdmCom
id_Mat	int(4)	No	Identificador único del material.	tbmateriales → id_Mat
fecha_Mat	date	No	Fecha de registro del material.	

Tabla 3.14 Tabla de Registro de Materiales



### tbUC

Campo	Tipo	Nulo	Descripción	Enlaces a
id_Usu	int(4)	No	Identificador único del usuario.	tbusuarios → id_Usu
id_Com	int(4)	No	Identificador único de la comunidad.	tbcomunidades → id_Com

Tabla 3.15 Tabla de Relación Usuario-Comunidad

### tbCM

Campo	Tipo	Nulo	Descripción	Enlaces a
id_Com	int(4)	No	Identificador único de la comunidad.	tbcomunidades → id_Com
id_Mat	int(4)	No	Identificador único del material.	tbmateriales → id_Mat

Tabla 3.16 Tabla de Relación Comunidad-Materiales

### 3.1.4 Pruebas

La última etapa de la fase de Construcción del Sistema Web Dentizt@ culmina con las pruebas, las cuales se llevaron a cabo para corroborar que el software funciona correctamente y que satisface lo especificado en la etapa de Planificación y Especificación de Requisitos.

Dichas Pruebas, fueron realizadas a una muestra de los usuarios que utilizarán el sistema, y fueron de ayuda para tomar en cuenta algunas adecuaciones que en una versión posterior pueden implementarse.

A continuación se muestran algunas imágenes de funcionalidad y usabilidad del Sistema Web Dentizt@ de la Figura 3.33 a la Figura 3.40.





“OPTIMIZACIÓN Y REINGENIERÍA DEL SISTEMA WEB DENTIZT@”

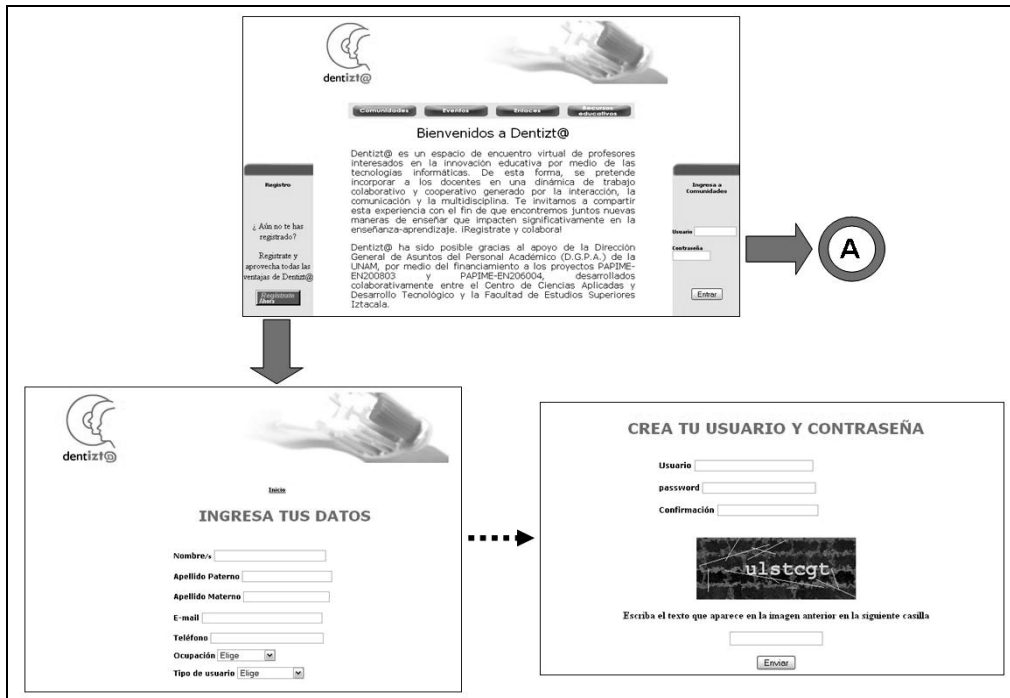


Figura 3.33 Página Principal de Dentizt@ y Registro de Usuarios

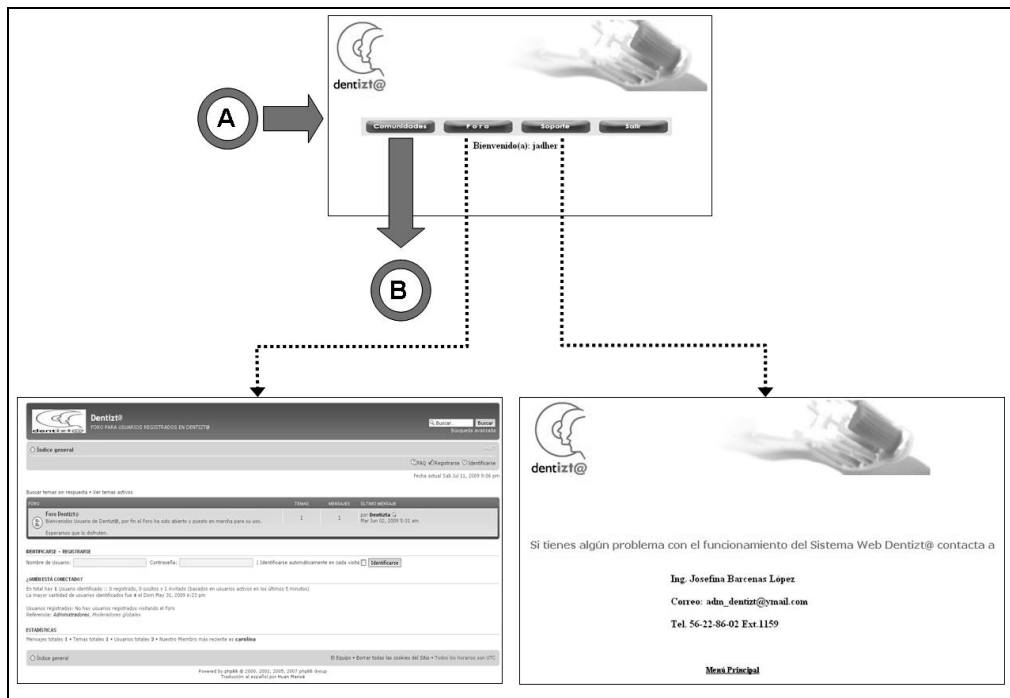


Figura 3.34 Menú de Usuarios de Dentizt@ con acceso a Foro y Soporte

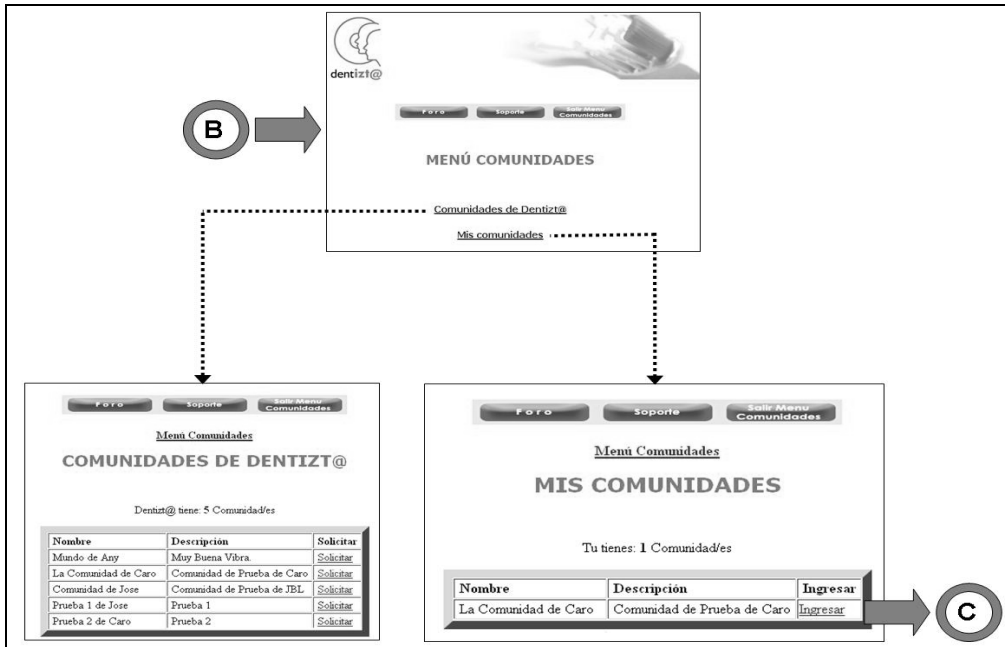


Figura 3.35 Menú de Comunidades de Dentizt@ y Comunidades del Usuario

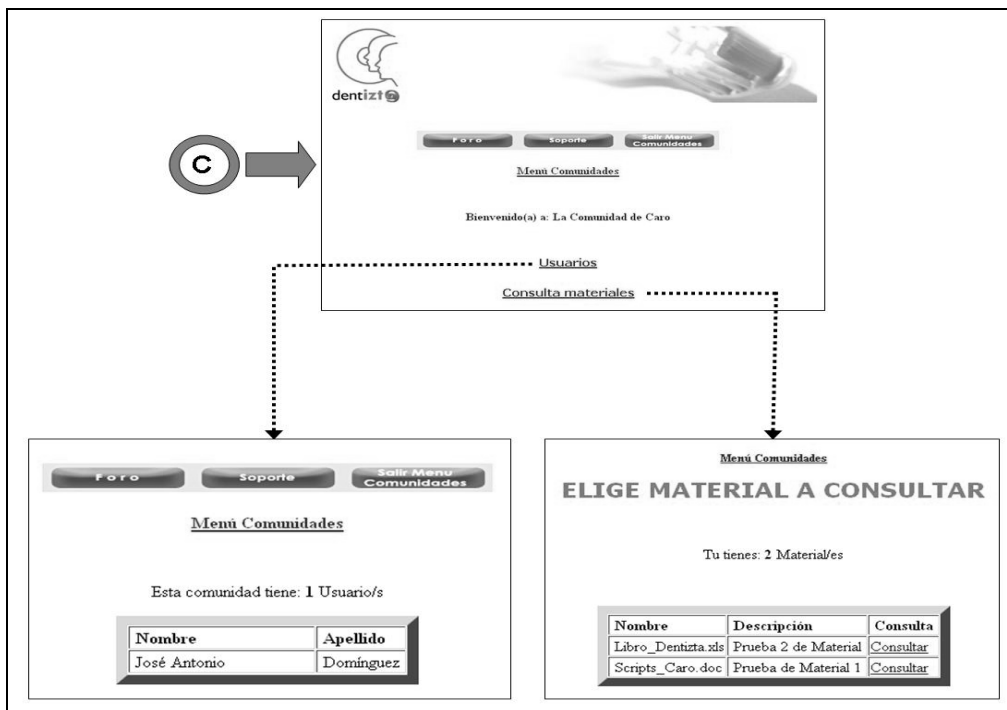


Figura 3.36 Comunidades de Dentizt@

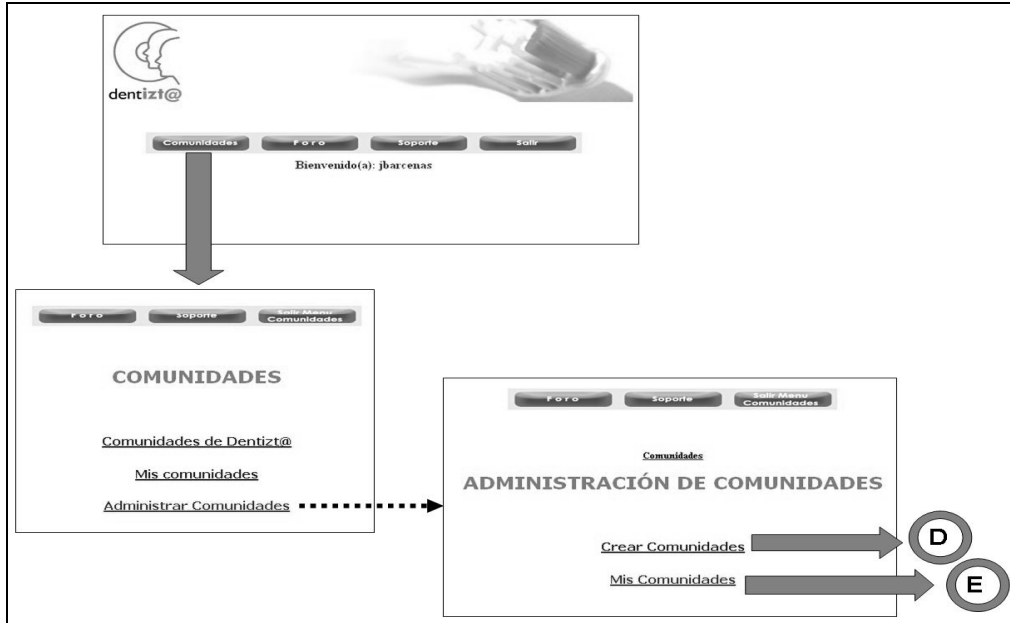


Figura 3.37 Menú de Comunidades de Dentizt@ para Administradores

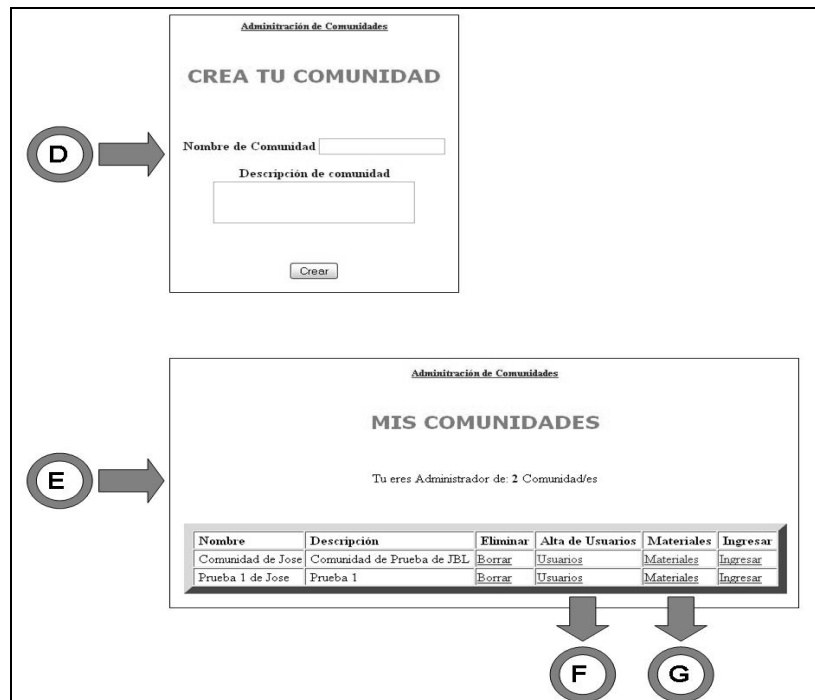


Figura 3.38 Administración de Comunidades de Dentizt@



“OPTIMIZACIÓN Y REINGENIERÍA DEL SISTEMA WEB DENTIZT@”

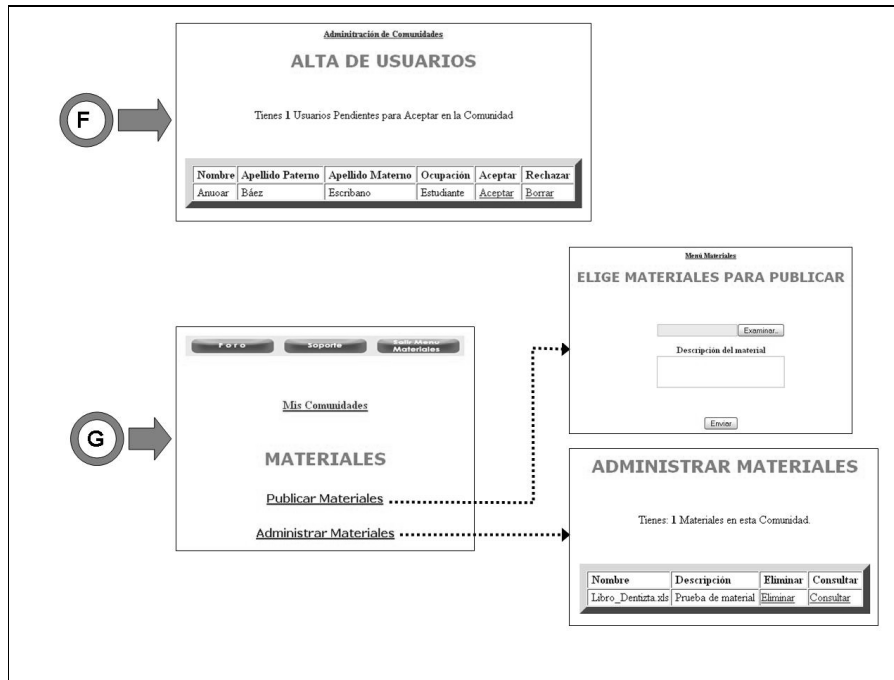


Figura 3.39 Administración de Usuarios y Materiales en las Comunidades

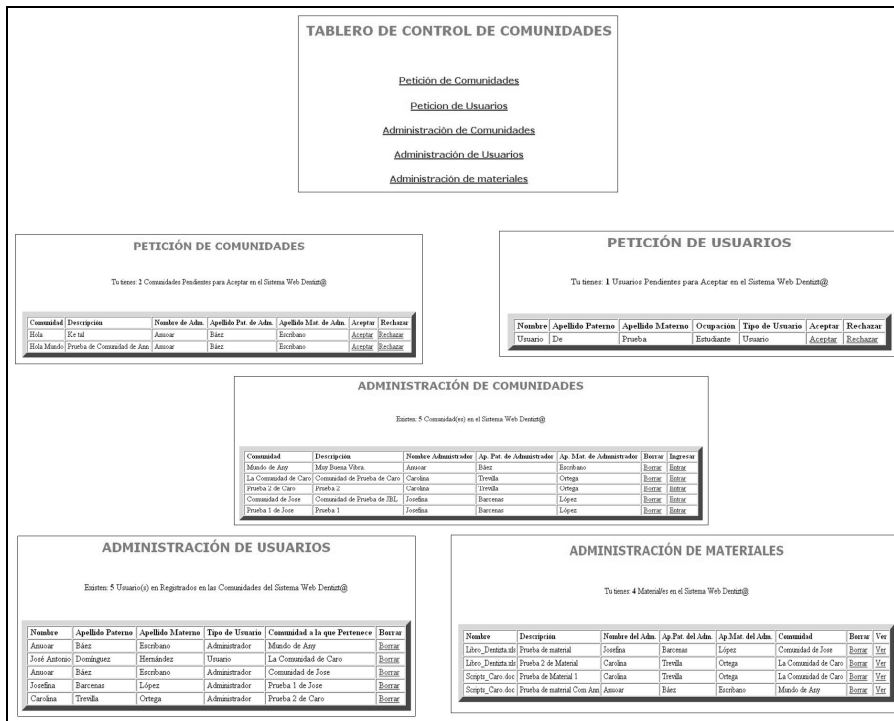


Figura 3.40 Tablero de Control para el Administrador General de Dentizt@



### 3.2 Fase de Instalación

En esta fase se puso la puesta en marcha del Sistema Web Dentizt@ en el entorno previsto de uso. Es decir, se liberó al público de manera *On Line* a través de la Web, y se comenzó a operar en un ambiente real para los usuarios finales. Es hasta este punto en donde se toma en cuenta las especificaciones técnicas del equipo a utilizar, así como del software necesario dentro del Servidor del CCADET en donde se alojara nuestro sistema.

#### 3.2.1 Especificación del Servidor

El paso inicial fue la instalación del sistema operativo bajo el cual trabajará el servidor, para lo cual se utilizó *Fedora Core 6* la cual es una distribución de Linux, el cual es software libre y de código abierto. Que entre sus características cuenta con los entornos Gnome 2.16 y KDE 3.5.4, X.org 7.1 y el kernel 2.6.18.

Respecto a la seguridad, Fedora posee la arquitectura de seguridad SELinux ("Security-Enhanced Linux") la cual esta integrada en el kernel 2.6.18 usando los módulos de seguridad Linux (o Linux Security modules, LSM). Entre sus características destaca la implementación de una gran variedad de políticas de seguridad, incluyendo control de acceso obligatorio (MAC "Mandatory Access Control"), a través de los Módulos de Seguridad de Linux.

Al ejecutar un kernel SELinux MAC se protege al sistema de aplicaciones maliciosas o dañadas que pueden perjudicar o destruir el sistema. SELinux define el acceso y los derechos de transición de cada usuario, aplicación, proceso y archivo en el sistema. SELinux gobierna la interacción de estos sujetos y objetos usando una política de seguridad que especifica cuán estricta o indulgente una instalación de Red Hat Enterprise Linux dada debería de ser.



Entre los requerimientos mínimos que se consideraron para la instalación de Fedora Core 6 se encuentran las especificaciones del procesador, el cual requería para su funcionamiento óptimo de un procesador Pentium 4 o posteriores. Respecto a la memoria RAM el mínimo a considerar son 256 MB, para el manejo de gráficos. Sobre el disco duro, se tomó en cuenta que la instalación mínima requería de 90MB y 1750MB para la instalación que incluía casi todos los paquetes. Adicionalmente tomamos en cuenta que se necesita espacio adicional para los datos del usuario, y que de acuerdo a las recomendaciones se debe reservar al menos un 5% de espacio libre para el funcionamiento adecuado del sistema.



Figura 3.41 Métodos de Instalación del Sistema Operativo Fedora

Fedora presenta 2 formas de instalación, el modo gráfico y el modo texto (Figura 3.41), siendo la primera la más recomendada, por su facilidad de uso, ya que presenta una interfaz en la que la configuración es muy sencilla y similar a la de



Windows que por su popularidad es más conocida. Dentro de sus características está su modo paso a paso, que lleva de la mano a los usuarios en la instalación del sistema operativo. Dentro de sus configuraciones más relevantes, está la elección del idioma, la creación de las particiones en las que se instalará el software y la configuración de la tarjeta de red (Figuras 3.42 y 3.43). Cada una de las cuales presenta una forma predeterminada la cual es la optima, sin embargo también da opción a personalizar cada una de las configuraciones.

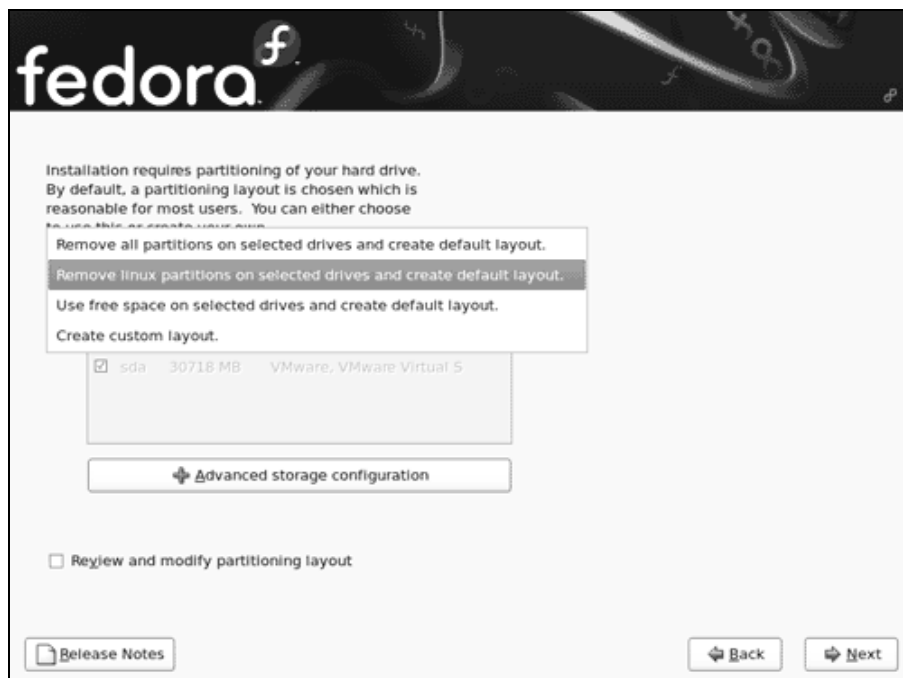


Figura 3.42 Creación de las particiones para la instalación de Fedora



Figura 3.43 Instalación y configuración de la Tarjeta de Red en Fedora

Fedora también presenta la posibilidad de seleccionar el software que queremos instalar, un ejemplo de esto son los paquetes para Desarrollo de Software y para el Servidor Web como se muestra en la figura 3.44.

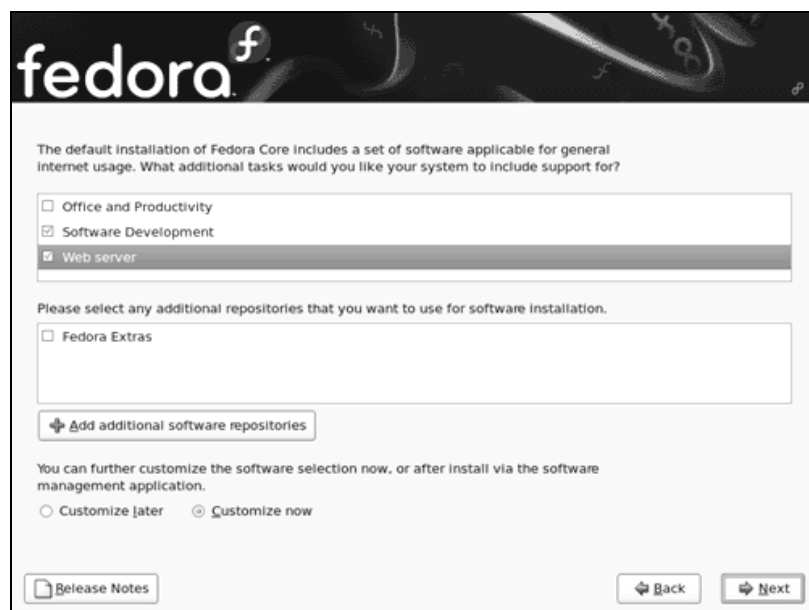


Figura 3.44 Instalación de paquetes para el Desarrollo Web y el Servidor Web





Por último, se instaló y configuró el software necesario para poner en funcionamiento el Servidor Web, para este fin se utilizó el paquete conocido como XAMPP derivado del acrónimo **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ySQL, **P**HP, **P**erl. Actúa como un Servidor Web libre, independiente de plataforma y catalogado como software libre, consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

Para su instalación sólo fue necesario su descarga a través de la pagina [www.apachefriends.org](http://www.apachefriends.org) y ejecutar un archivo tar. Esta herramienta también incluye otro módulo llamado phpMyAdmin, para la administración de las bases de datos. La instalación de XAMPP es muy sencilla y no requiere de gran cantidad de tiempo.

Se creó un servidor LAMP, esto se refiere a la mezcla de **L**inux+**A**pache+**M**ysql+**P**HP en este caso el servidor se configuró con los siguientes componentes:

- Apache 2.2.11 - Servidor Web de Linux
- MySQL 5.1.33 - Servidor de base de datos MySQL
- PHP 5.2.9 - Lenguaje interpretado PHP
- phpMyAdmin 3.1.3.1 - Software de administración de base de datos vía Web

Una vez instalado el XAMPP, se debe iniciar los servicios, para esto solo fue necesario llamar el comando: `/opt/lampp/lampp start`

En automático los servicios se inician y en pantalla podemos ver lo siguiente:

```
Starting XAMPP 1.7.1  
LAMPP: Starting Apache...  
LAMPP: Starting MySQL...  
LAMPP started.
```



Ready. Apache and MySQL are running.

XAMPP tiene predeterminada la configuración para trabajar offline, como un servidor local, y una vez que ha sido instalado pudo comprobarse la funcionalidad mediante el navegador tecleando la dirección `http://localhost` en la barra de direcciones, con lo cual se abre la página inicial en donde se da la bienvenida (Figura 3.45).

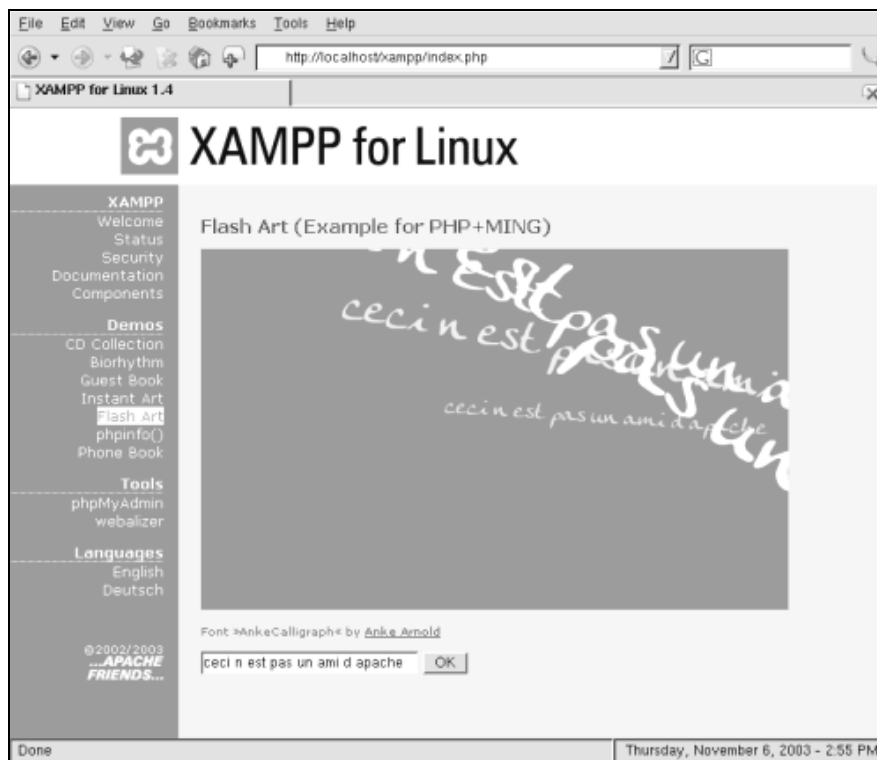


Figura 3.45 Página de inicio de XAMPP

Debido a que XAMPP esta preconfigurado para trabajar de manera local, fue necesario editar el archivo de configuración de Apache `httpd.conf` que encontramos en la carpeta **etc** dentro del directorio del XAMPP.

Como primer punto se pudo observar que el directorio predeterminado para guardar nuestras páginas es el `htdocs` del Apache, ubicado dentro del directorio



xampp. Sin embargo, fue posible cambiarlo, simplemente modificando las líneas del *DocumentRoot* de la siguiente manera:

```
#  
# DocumentRoot: The directory out of which you will serve your  
# documents. By default, all requests are taken from this directory, but  
# symbolic links and aliases may be used to point to other locations.  
#  
DocumentRoot "/opt/lampp/htdocs/"
```

Se sustituye la carpeta *htdocs* del Apache por la carpeta de nuestro sistema llamada *dentizta*. Considerando además, que todos los archivos del sistema siempre deberán estar localizados en esta carpeta. Por lo que la línea se modificó quedando de la siguiente manera: `DocumentRoot "/opt/lampp/htdocs/dentizta"`

Posteriormente, se buscan las líneas del *DirectoryIndex*, en donde aparece lo siguiente:

```
#  
# DirectoryIndex: sets the file that Apache will serve if a directory  
# is requested.  
#  
# The index.html.var file (a type-map) is used to deliver content-  
# negotiated documents. The MultiViews Option can be used for the  
# same purpose, but it is much slower.  
#  
DirectoryIndex index.html index.html.var
```

Aquí se modifica la última línea, en donde ponemos el nombre de nuestra página de inicio del sistema para que se auto ejecute y evitar algún mensaje de error.



Algo importante fue el verificar que debajo de estas líneas se encontrara las extensiones con las que trabajará Apache. Por lo cual debe tener lo siguiente, de no ser así se colocan:

```
AddType application/x-httpd-php .php .php3 .php4 .php5 .phtml
```

```
AddType application/x-httpd-php-source .phps
```

La primera indica las extensiones que serán interpretadas por el Apache. La segunda sirve para que en una página con extensión *.phps* muestre el código PHP a color, muy útil si queremos por ejemplo mostrar el código fuente a color en una página. De tal forma que al final todo quedó de la siguiente manera:

```
#  
# DirectoryIndex: sets the file that Apache will serve if a directory  
# is requested.  
#  
# The index.html.var file (a type-map) is used to deliver content-  
# negotiated documents. The MultiViews Option can be used for the  
# same purpose, but it is much slower.  
#  
DirectoryIndex index.phtml index.htm index.php index.php3 index.php4 index.phtml  
index.html.var  
AddType application/x-httpd-php .php .php3 .php4 .php5 .phtml  
AddType application/x-httpd-php-source .phps
```

Para poder cambiar la configuración de local a online, se editan las líneas siguientes:

```
#  
# Listen: Allows you to bind Apache to specific IP addresses and/or  
# ports, instead of the default. See also the <VirtualHost>  
# directive.  
#
```



```
# Change this to Listen on specific IP addresses as shown below to  
# prevent Apache from glomming onto all bound IP addresses.  
#  
#Listen 12.34.56.78:80  
#Listen 80
```

Donde se ingresa la *ip* del servidor se aloja nuestro sistema, por lo que finalmente queda:

```
#  
# Listen: Allows you to bind Apache to specific IP addresses and/or  
# ports, instead of the default. See also the <VirtualHost>  
# directive.  
#  
# Change this to Listen on specific IP addresses as shown below to  
# prevent Apache from glomming onto all bound IP addresses.  
#  
Listen 132.248.36.50:80
```

Una vez realizado estas modificaciones se guarda el archivo `httpd.conf`, obteniendo el servidor en modo online, listo para usarse vía Web.

Finalmente, para poner en marcha el servidor, fue necesario instalar y configurar el software que nos proporciona la posibilidad de crear un foro. Para el Sistema Web Dentizt@ se utilizó *phpBB*. Dicho software es en la actualidad la solución más popular para crear Foros, Paneles de Anuncios o Paneles de Discusión (Figura 3.46). La herramienta proporciona de una manera fácil y con amplias posibilidades de personalización, un sistema de foros sobre bases de datos basadas en el lenguaje SQL como MySQL.

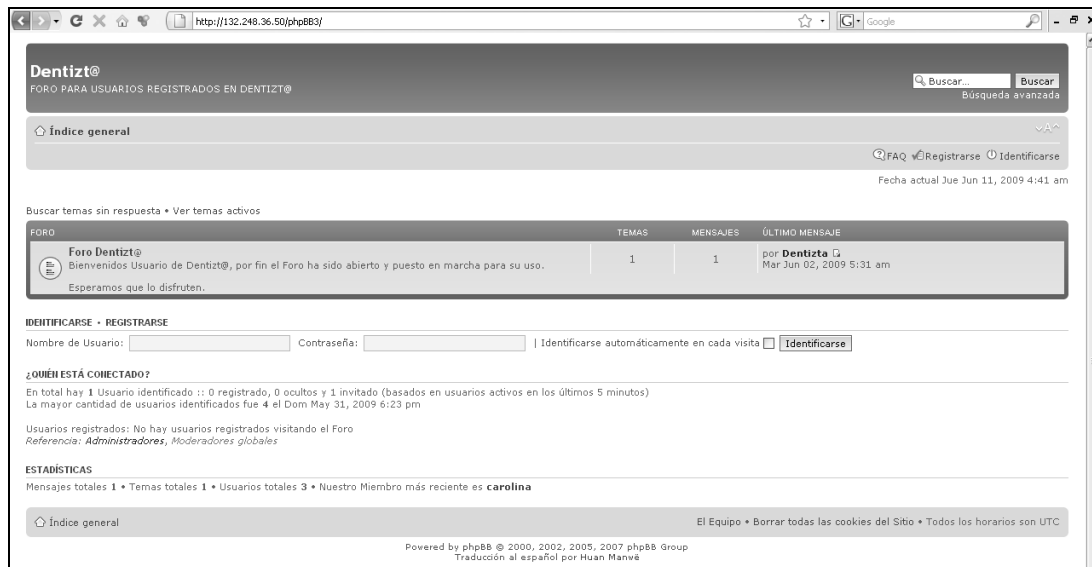


Figura 3.46 Página de inicio de foro para Dentizt@

Dicho software cuenta entre sus características con mensajes públicos o privados, un sistema modular para incorporación de extensiones, vista detallada de foros, mensajes y temas, permisos de administración individuales, permisos de usuarios individuales, estadísticas de usuarios, moderadores globales, paneles de control para moderadores y administradores, una interfaz de usuario avanzada y múltiples idiomas.

La instalación de phpBB, es realmente sencilla, ya que cuenta con una interfaz de modo gráfico para los usuarios, por lo que realmente solo basta que el servidor cumpla con los requerimientos mínimos, como soporte para PHP y un sistema de base de datos SQL.

El primer paso es descomprimir el archivo phpBB3, en una carpeta local en nuestro servidor. Usando el navegador, hay que dirigirse a la ubicación donde se encuentra phpBB3, añadiendo a la url `/install/index.php` para dirigirse directamente a la instalación del foro.



Se selecciona la pestaña *INSTALAR*, siguiendo los pasos y completando todos los datos requeridos. Se rellenan los campos de la *Base de Datos*, con la información de la base de datos que destinamos para tal efecto.

Además se llenan los datos del administrador quien tendrá acceso total al foro, con el cual se administrara y configurara. Reemplazamos el archivo config.php vacío. Posteriormente, aparece la configuración avanzada del foro. Estos datos son configurables luego desde el Panel de Administración del foro por si se tiene que hacer un ajuste en le futuro.

Finalmente se elimina la carpeta */install/*, para de esta forma iniciar en el foro y acceder al Panel de Administración.

De esta forma se pone en marcha el servidor con todos los servicios necesarios para que Dentizt@ funcionara.



---

## CAPÍTULO 4. EVALUACIÓN DEL SISTEMA WEB DENTIZT@.

### 1.1 Pruebas y Resultados

Cuando se habla de evaluar software generalmente los factores que son objeto de análisis y estudio en primera instancia por los usuarios finales, son aquellos relacionados con la funcionalidad y la facilidad de interacción del usuario con el sistema (usabilidad). Esto no quiere decir que el diseño, la estructura y la funcionalidad del sistema que nos marca la Ingeniería de Software, sean menos importantes, sin embargo estas son cuestiones que se dejan fuera por el momento ya que de acuerdo con Gándara (2000) los resultados de los criterios de Usabilidad serán en buena medida, las definidas por las características tanto de la interfaz como de la documentación; de hecho, un programa usable no requiere de grandes manuales, ni de grandes inversiones económicas.

De acuerdo con lo anterior, los criterios establecidos para evaluar la funcionalidad y usabilidad del sistema Web Dentizt@ se basaron en parámetros definidos por:

a) *Facilidad de navegación.* Que define la facilidad con la que el usuario puede revisar los contenidos del software: Legibilidad de la información, calidad estética de la interface y la actitud del usuario al utilizar el software.

b) *Criterios Técnicos:*

- **Compatibilidad.** Define si el software es compatible y puede ser usado en cualquier equipo de cómputo, si requiere de hardware especial o software exclusivo.
- **Recursos de comunicación del sistema.** Determina si el equipo de cómputo no requiere de recursos (hardware) especiales para el funcionamiento del software.





- Interacción con el usuario principiante o experto. Determina si el software requiere de capacitación avanzada para utilizar el software.
- Flujo de información. Define si el contenido del software se presente de manera rápida y fluida.

a) *Criterios Económicos.*

Relación costo/beneficio, considerando el costo total de propiedad (que incluye mantenimiento, soporte y actualización), Accesibilidad (facilidad de adquisición y soporte) obsolescencia.

La población “muestra” a la que fue aplicado la prueba de evaluación esta formada por un grupo de 20 profesores, tanto de carrera como de asignatura, de la carrera de Cirujano Dentista de la Facultad de Estudios Superiores de Iztacala.

La escala de medición seleccionada para el instrumento evaluador de los materiales, es una escala Likert a la cual se le asignó un valor ordinal, lo que permitió representar categorías ordenadas, de manera que tanto el estudiante como el profesor que contestara el cuestionario de evaluación tuviera “a-priori” una respuesta inmediata y que sobre todo, nos proporcionara la información necesaria para el análisis, de acuerdo con un criterio definido por un sentido que no permitiría introducir alteraciones en la secuencia de los valores. Así los intervalos “Muy bueno”, “Bueno”. “Regular o Normal” y “Malo” son valores específicos que permiten medir los aspectos a estudiar del material que nos ocupa, permitiendo por otro lado hacer un análisis crítico y fundamentado del recurso educativo diseñado.

El instrumento para la evaluación del Sistema Web Dentiz@ fue el siguiente:



Facultad de Estudios Superiores Iztacala

Carrera de Cirujano Dentista



CENTRO DE CIENCIAS APLICADAS Y  
DESARROLLO TECNOLÓGICO

**CUESTIONARIO DE EVALUACIÓN PARA EL SISTEMA WEB DENTIZT@**

**NORMAS PARA CONTESTAR**

Marque con una X una opción en cada una de las preguntas según el número que corresponde a la calificación. Procurando no equivocarse

Escala: 1 = Muy bueno; 2 = Bueno; 3 = Normal; 4 = Malo y 5 = Muy Malo

1. Es fácil de usar, no requiere de conocimientos avanzados en computación.	1—2---3---4---5
2. Es de fácil exploración y navegabilidad.	1—2---3---4---5
3. Permite la difusión e intercambio de información.	1—2---3---4---5
4. Permite conocer los puntos de vista y comentarios de los usuarios respecto a un tema o material publicado.	1—2---3---4---5
5. Contribuye al proceso enseñanza-aprendizaje utilizando la tecnología como principal herramienta.	1—2---3---4---5

OBSERVACIONES:

---



---



---



Tomando en cuenta a esta población, se llevaron a cabo las pruebas del sistema, obteniéndose los siguientes resultados.

Criterios	Resultados
1. Es fácil de usar, no requiere de conocimientos avanzados en computación.	91.2%
2. Es de fácil exploración y navegabilidad.	92.5%
3. Permite la difusión e intercambio de información.	95.0%
4. Permite conocer los puntos de vista y comentarios de los usuarios respecto a un tema o material publicado.	95.0%
5. Contribuye al proceso enseñanza-aprendizaje utilizando la tecnología como principal herramienta.	96.2%

Con los resultados anteriores se puede concluir que el usuario puede navegar en los diferentes contenidos del Sistema Web Dentizt@ sin complicaciones, porque no necesita de conocimientos avanzados en la rama de la computación, generando en los usuarios una actitud de aceptación y motivación para seguir utilizando el mismo como medio de difusión de materiales y puntos de vista, utilizando a la tecnología como principal herramienta.



## CONCLUSIONES

Para la optimización del Sistema Web Dentizt@ fue necesario utilizar una metodología que permitiera desarrollar de manera ordenada y sencilla cada uno de los objetivos de su funcionamiento.

Lo anterior se logró gracias a la utilización de la metodología propuesta por Craig Larman que utiliza como herramienta el Lenguaje Unificado de Modelado, y que permitió eliminar desde el análisis, la ambigüedad, teniendo como resultado un sistema fácil de entender, claro de desarrollar, implementar y probar, con la ventaja de poder brindar en un futuro mantenimiento o implementar nuevos módulos propiciando la mejora continua del Sistema Web por tener la documentación necesaria.

Dado que el problema primario que se observó fue precisamente la falta de documentación para realizar la actualización del Sistema Web Dentizt@, se consideró hacer uso de la reingeniería para la implementación de la metodología. Por lo que, tomando el lenguaje UML como apoyo, se obtuvo como resultado los diagramas del sistema implementados con símbolos, los cuales resultaron más sencillos de comprender que el código que se tenía del Sistema anterior. Estos diagramas contribuyeron de manera importante al análisis sobre la funcionalidad y necesidades que los usuarios requerían, además de ser una fuente importante para tomar decisiones en su mejora.

El desarrollo de Dentizt@ como una aplicación de tres capas vía web, se consideró debido a que en la actualidad estamos viviendo en una sociedad que depende cada vez más de la creación, la administración y la distribución de la información a través de la Internet. Pues es a través de estos Sistemas que se logra apoyar diversas actividades de la humanidad como son: la entrada, el almacenamiento y la difusión de la información, con ayuda de usuarios y las



computadoras como principal herramienta, quitando las limitaciones de distancia, espacio y tiempo.

Con la optimización y reingeniería del Sistema Web Dentizt@, se obtuvo una aplicación con interfaces sencillas que permite que las personas sin conocimientos avanzados en computación puedan utilizarlo sin complicaciones.

De esta forma se logró corregir la dependencia de un administrador experto que existía en la versión anterior, junto con la baja funcionalidad y alta complejidad en su manejo. Dentizt@ permite ahora tener a usuarios administradores de comunidades y administrar, dentro de la comunidad a su cargo, usuarios y materiales de manera sencilla. Esto sin tener necesariamente conocimientos avanzados en computación y solo dando clic con el ratón en la opción que quiere ver o realizar, siendo transparente la programación para el usuario y guardando su complejidad únicamente en su programación.

Al satisfacer todas las necesidades para los usuarios, se justifica el esfuerzo realizado cumpliendo con los objetivos establecidos para este proyecto y se resalta la importancia que la ingeniería tiene en la sociedad al ver la utilidad que para la comunidad académica representa este Sistema. Cumpliendo así con uno de los principales de objetivos de la disciplina: desarrollar en beneficio de la sociedad y de las personas.



## ANEXO. Código Fuente

### 1) Script para la conexión a la Base de Datos “dentizta”

```
<?php
function Conectarse()
{
    if (!( $link=mysql_connect("localhost","root","") ))
    {
        echo "Error conectando a la base de datos.";
        exit();
    }
    if (!mysql_select_db("dentizta",$link))
    {
        echo "Error seleccionando la base de datos.";
        exit();
    }
    return $link;
}

function getResultado()
{
    return ($this->result);
}

?>
```

### 2) Script para ingresar al Sistema Web Dentizt@

```
<?php
session_start ();
include("conex.phtml"); //hace la conexion con la base de datos
$link=Conectarse();

//DECLARACION DE VARIABLES ASIGNANDOLE EL VALOR DEL FORMULARIO
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];
$_SESSION["myusername"]=$myusername;
// Para proteger de MySQL injection
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);
```



```
$mypassword = mysql_real_escape_string($mypassword);

//CONSULTA PARA NIVEL DE USUARIO CON USUARIO Y PASSWORD IGUALES

$consulta="SELECT tipo_Usu FROM tbUsuarios WHERE login_Usu='$myusername'
and pass_Usu='$mypassword'";
$ejecutaConsulta=mysql_query($consulta,$link);
$count=mysql_num_rows($ejecutaConsulta);
if(mysql_num_rows($ejecutaConsulta) == 0){ // si no hay resultados
header("location: ingreso2.phtml");
}
$count2=mysql_result($ejecutaConsulta,0,"tipo_Usu");

if($count==1 && $count2=="Administrador")
{
    session_register("myusername");
    header("location:menu_Admcom.phtml");
}

elseif($count==1 && $count2=="Admgral")
{
    session_register("myusername");
    header("location:menu_Super.phtml");
}

elseif($count==1 && $count2=="Usuario")
{
    session_register("myusername");
    header("Location: menu_Usuario.phtml");
}

?>
```



### 3) Script que muestra las Comunidades de Dentizt@

```
<?php
    include("conex.phtml");
    $link=Conectarse();
    $sql = "SELECT * FROM tbComunidades";
    $resultado = mysql_query($sql);
    $filas = mysql_num_rows($resultado);
    echo "<p>Dentizt@ tiene: <b>$filas</b> Comunidad/es</p>\n";
    echo "<table border='10' cellspacing='1' cellpadding='1'>\n";
    // Empezar tabla
    echo "<tr><td>&nbsp;<b>Nombre</b></td>
    <td>&nbsp;<b>Descripción</b>&nbsp;</td>
    <td>&nbsp;<b>Solicitar</b>&nbsp;</td></tr>"; // Crear fila
    while($row = mysql_fetch_array($resultado))
    {
        printf("<tr><td>&nbsp;%s&nbsp;</td>
        <td>&nbsp;%s&nbsp;</td><td>&nbsp;
        <a href=\"comu.phtml?id=%d\">Solicitar</a>&nbsp;</td></tr>",
        $row["nom_Com"], $row["descrip_Com"], $row["id_Com"]);
    }
    mysql_free_result($resultado);
    mysql_close($link);
?>
```

### 4) Script que muestra las comunidades en las que está registrado el usuario

```
<?php
    include("conex.phtml");
    $link=Conectarse();
    $usu = $_SESSION["myusername"];
    $consulta="SELECT id_Usu FROM tbUsuarios WHERE login_Usu='$usu'";
    $ejecutaConsulta=mysql_query($consulta,$link);
    $count=mysql_fetch_assoc($ejecutaConsulta);
    $id_Usu= $count['id_Usu'];
    $sql = "SELECT * FROM tbComunidades JOIN tbUC
```





```
WHERE tbUC.id_Com=tbComunidades.id_Com AND tbUC.id_Usu='$id_Usu';
$resultado = mysql_query($sql);
$filas = mysql_num_rows($resultado);
echo "<p>Tu tienes: <b>$filas</b> Comunidad/es</p>\n";
echo "<table border='10' cellspacing='1' cellpadding='1'>\n";
// Empezar tabla
echo "<tr><td>&nbsp;<b>Nombre</b></td>
      <td>&nbsp;<b>Descripci&oacute;n</b>&nbsp;</td>
      <td>&nbsp;<b>Ingresar</b>&nbsp;</td></tr>"; // Crear fila
while($row = mysql_fetch_array($resultado))
{
    printf("<tr><td>&nbsp;<td>&nbsp;<td>&nbsp;</td><td>&nbsp;<td>&nbsp;</td><td>
          <a href=\"comsusu.phtml?id=%d\">Ingresar</a></td></tr>",
          $row["nom_Com"], $row["descrip_Com"], $row["id_Com"]);
}
mysql_free_result($resultado);
mysql_close($link);
?>
```

## 5) Script que muestra a los usuarios que pertenecen a una comunidad

```
<?php
define ("salto", "<br>\n");
session_start();
include("conex.phtml");
$link=Conectarse();
$id_Com=$_GET['id'];
$sql = "SELECT * FROM tbUsuarios JOIN tbUC
WHERE tbUsuarios.id_Usu=tbUC.id_Usu AND tbUC.id_Com= '$id_Com' ";
$resultado = mysql_query($sql);
$filas = mysql_num_rows($resultado);
echo "<p>Esta comunidad tiene: <b>$filas</b> Usuario/s</p>\n";

echo "<table border='10' cellspacing='1' cellpadding='1'>\n";
// Empezar tabla
echo "<tr><td>&nbsp;<b>Nombre</b></td>
```



```
        <td>&nbsp;<b>Apellido</b>&nbsp;</td></tr>"; // Crear fila
while($row = mysql_fetch_array($resultado))
{
    printf("<tr><td>&nbsp;<b>%s</b></td><td>&nbsp;<b>%s</b>&nbsp;</td></tr>",
        $row["nom_Usu"], $row["app_Usu"]);
}
mysql_free_result($resultado);
mysql_close($link);
?>
```

## 6) Script que muestra los materiales de una comunidad

```
<?php
define ("salto", "<br>\n");
session_start();
include("conex.phtml");
$link=Conectarse();
$id_Com=$_GET['id'];
$sql = "SELECT * FROM tbMateriales JOIN tbCM
WHERE tbMateriales.id_Mat=tbCM.id_Mat AND tbCM.id_Com= '$id_Com' ";
$resultado = mysql_query($sql);
$filas = mysql_num_rows($resultado);
echo "<p>Tu tienes: <b>$filas</b> Material/es</p>\n";
echo "<table border='10' cellspacing='1' cellpadding='1'>\n";
// Empezar tabla
echo "<tr><td>&nbsp;<b>Nombre</b></td>
      <td>&nbsp;<b>Descripción</b>&nbsp;</td>
      <td>&nbsp;<b>Consulta</b>&nbsp;</td></tr>"; // Crear fila
while($row = mysql_fetch_array($resultado))
{
    printf("<tr><td>&nbsp;<b>%s</b></td><td>&nbsp;<b>%s</b>&nbsp;</td><td>
      <a href=\"consulta_mat.phtml?id=%d\">Consultar</a>
      </td></tr>",
        $row["nom_Mat"], $row["descrip_Mat"], $row["id_Mat"]);
}
mysql_free_result($resultado);
```



```
mysql_close($link);  
?>
```

## 7) Script que permite consultar materiales de una comunidad

```
<?php  
include("conex.phtml");  
$link=Conectarse();  
$id=$_GET['id'];  
$directorio_archivo = "/opt/lampp/htdocs/Dentizta/doc_Ser/";  
$ejecutaConsulta=mysql_query("select nom_Mat from tbMateriales  
    where id_Mat = $id",$link);  
$nombre=mysql_result($ejecutaConsulta,0,"nom_Mat");  
$ejecutaConsulta2=mysql_query("select ext_Mat from tbMateriales  
    where id_Mat = $id",$link);  
$ext=mysql_result($ejecutaConsulta2,0,"ext_Mat");  
  
if($ext=="application/pdf")  
{  
    header('Content-type: application/pdf');  
    filename=".$nombre");  
    $manejadorArchivo=fopen("$directorio_archivo$nombre","r+");  
    fpassthru($manejadorArchivo);  
    fclose($manejadorArchivo);  
}  
elseif($ext=="application/msword")  
  
{  
    header("Content-type: application/msword");  
    $manejadorArchivo=fopen("$directorio_archivo$nombre","r+");  
    fpassthru($manejadorArchivo);  
    fclose($manejadorArchivo);  
}  
elseif($ext=="application/vnd.ms-excel")  
{  
    header("Content-type: application/vnd.ms-excel");
```



```
$manejadorArchivo=fopen("$directorio_archivo$nombre", "r+");  
fpassthru($manejadorArchivo);  
fclose($manejadorArchivo);  
}  
?>
```

## 8) Script que permite solicitar la creación de una comunidad

```
<?php  
    session_start();  
    include("conex.phtml");  
    $link=Conectarse();  
    $usu = $_SESSION["myusername"];  
    $nom_Com=$_POST['nom_Com'];  
    $descrip_Com=$_POST['descrip_Com'];  
    $id_Usu=$_GET['id_Usu'];  
    $id_AdmCom=$_GET['id_AdmCom'];  
  
    mysql_query("insert into tbprecomunidades  
        (nom_Com,descrip_Com,id_AdmCom) values  
        ('$nom_Com','$descrip_Com','$id_AdmCom')", $link);  
    header("Location: aviso_preregcom.phtml");  
?>
```

## 9) Script que permite la administración de una comunidad

```
<?php  
    define ("salto", "<br>\n");  
    session_start();  
    include("conex.phtml");  
    $link=Conectarse();  
    $usu = $_SESSION["myusername"];  
    $consulta="SELECT id_Usu FROM tbUsuarios WHERE login_Usu='$usu';"  
    $ejecutaConsulta=mysql_query($consulta, $link);  
    $count=mysql_fetch_assoc($ejecutaConsulta);  
    $id_Usu= $count['id_Usu'];
```



```
$consulta2="SELECT id_AdmCom FROM tbAdmCom WHERE id_Usu='$id_Usu'";
$ejecutaConsulta2=mysql_query($consulta2,$link);
$count2=mysql_fetch_assoc($ejecutaConsulta2);
$id_AdmCom= $count2['id_AdmCom'];
$sql = "SELECT * FROM tbComunidades JOIN tbRegistroCom
        WHERE tbRegistroCom.id_Com=tbComunidades.id_Com AND
        tbRegistroCom.id_AdmCom='$id_AdmCom' ";
$resultado = mysql_query($sql);
$filas = mysql_num_rows($resultado);
echo "<p>Tu eres Administrador de:
        <b>$filas</b>Comunidad/es</p>\n";
echo "<table border='10' cellspacing='1' cellpadding='1'>\n";
// Empezar tabla
echo "<tr><td>&nbsp;<b>Nombre</b></td>
        <td>&nbsp;<b>Descripción</b>&nbsp;</td>
        <td>&nbsp;<b>Eliminar</b>&nbsp;</td> <td>&nbsp;<b>
        Alta de Usuarios</b>&nbsp;</td>
        <td>&nbsp;<b>Materiales</b>&nbsp;</td>
        <td>&nbsp;<b>Ingresar</b>&nbsp;</td></tr>"; // Crear fila
while($row = mysql_fetch_array($resultado))
{
    printf("<tr><td>&nbsp;<td>&nbsp;<td>
        <a href=\"borra_Com.phtml?id=%d\">Borrar</a></td><td>
        <a href=\"alta_usu.phtml?id=%d\">Usuarios</a></td><td>
        <a href=\"menu_materiales_admin.phtml?id=%d\">
        Materiales</a></td><td>
        <a href=\"comsusuac.phtml?id=%d\">Ingresar</a>
        </td></tr>",
        $row["nom_Com"],$row["descrip_Com"],$row["id_Com"],
        $row["id_Com"],$row["id_Com"],$row["id_Com"]);
}
mysql_free_result($resultado);
mysql_close($link);
?>
```



## 10) Script para publicar materiales en una comunidad

```
<?php
    include("conex.phtml");
    $link=Conectarse();
    $descrip_Mat=$_POST['descrip_Mat'];
    $id_Com=$_GET['id_Com'];
    $consulta="SELECT id_AdmCom FROM tbRegistroCom
        WHERE id_Com='$id_Com'";
    $ejecutaConsulta=mysql_query($consulta,$link);
    $count=mysql_fetch_assoc($ejecutaConsulta);
    $id_AdmCom=$count['id_AdmCom'];

    $directorio_archivo = "/opt/lampp/htdocs/Dentizta/doc_Ser/";
    $nombre = strstr($_HTTP_POST_FILES['archivo']['name'],
        "ñ'ÁÉÍÓÚÀÈÌÒÛäéíóúääëïöüàèìòù ", "n-AEIOUAEIOUaeiouaeiouaeiou_");
    $tamaño = $_HTTP_POST_FILES['archivo']['size'];
    $ext_Mat = $_HTTP_POST_FILES['archivo']['type'];

    if (move_uploaded_file($_HTTP_POST_FILES['archivo']['tmp_name'],
        $directorio_archivo.$nombre))

    {
        mysql_query("insert into tbMateriales
            nom_Mat,tam_Mat,ext_Mat,descrip_Mat) values
            ('$nombre','$tamaño','$ext_Mat','$descrip_Mat')",
            $link);
        $consulta2="SELECT id_Mat FROM tbMateriales
            WHERE nom_Mat='$nombre' AND tam_Mat='$tamaño' AND
            ext_Mat='$ext_Mat' AND descrip_Mat='$descrip_Mat'";
        $ejecutaConsulta2=mysql_query($consulta2,$link);
        $count2=mysql_fetch_assoc($ejecutaConsulta2);
        $id_Mat=$count2['id_Mat'];
        mysql_query("insert into tbRegistroMat (id_AdmCom,id_Mat)
            values ('$id_AdmCom','$id_Mat')", $link);
        mysql_query("insert into tbCM (id_Com,id_Mat) values
```



```
        ('$id_Com', '$id_Mat'), $link);  
    header("Location: men_up_archivo.phtml");  
}  
else  
{  
    header("Location: nomen_up_archivo.phtml");  
}  
?>
```



## GLOSARIO

<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Ambiente:	Se refiere al tipo de presentación con que interactúa el usuario, y queda definido por el hardware y software usado en la PC.
Archivo:	Unidad de información manipulable por el usuario desde el sistema operativo; se utiliza para guardar información en discos, cintas, etc.
Base de datos:	Conjunto de archivos o tablas almacenados y manejados en orden y significancia por un DBMS.
BBS:	Bulletin Board System. Sistema de tablero de anuncios, en Internet se utiliza para unir a usuarios con intereses similares.
BIT:	Binary Digit. Abreviación de dígito binario. Unidad mínima de información con que trabajan internamente las computadoras, puede ser un cero (0) o un uno (1).
Boot:	Área ubicada en el disco (floppy o disco duro); donde se ubica el software de arranque de la PC. Lugar donde suelen almacenarse el código de los virus informáticos.
bps:	Bits per second. Bits por segundo. Unidad para medir la velocidad de transferencia de datos en un sistema de comunicaciones.





<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Browser:	Navegador o visualizador, es un programa que permite navegar mediante hipertexto.
BUS:	Un canal o ruta común entre dispositivos del hardware, ya sea internamente entre componentes del computador o externamente entre estaciones de una red de comunicaciones.
Byte:	Unidad de medición de memoria, compuesta por 8 bits. El conjunto de 8 bits o byte representa un carácter (signo, letra o número).
Caballo de Troya:	Agente destructivo de software o virus informático; aparece asociado en aparentes programas de cómputo.
Carácter:	Letra, dígito o símbolo especial; es equivalente a un byte, el cual es reconocido por una computadora.
Clic:	Acción del usuario al presionar algún botón del mouse.
Compilador:	Programas de software que permiten traducir a código de máquina el conjunto de instrucciones escritas en un lenguaje de programación.
Computadora:	Máquina de propósito general que procesa datos de acuerdo con las instrucciones almacenadas en un programa.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Concentrador:	Dispositivo que concentra todas las señales de los nodos y servidores de una red y las envía, una por una, a la computadora central y recibe de ésta la respuesta que envía al nodo correspondiente de acuerdo con un código de dirección.
Correo electrónico:	Electronic mail o e-mail. Medio de comunicación entre los usuarios de redes de computadoras.
CPU:	Control Processing Unit. Unidad central de procesamiento. Parte central de una computadora. Véase: Procesador.
DBMS:	Database Management System. Sistema manejador de bases de datos. Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos. Acepta pedidos de datos desde un programa de aplicación y le ordena al sistema operativo transferir los datos apropiados.
Dirección electrónica:	Identificador único y asociado a cada computadora que forma parte de una red.
DNS:	Domain Name Service. Servicio de nombres de dominio. Es un servicio de Internet y de TCP/IP, que traduce una dirección alfabética en dirección numérica.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
DOS:	Disk Operating System. Sistema operativo en disco; conjunto de programas que permite la administración del equipo, es un sistema monousuario, comúnmente usado en PC.
Driver:	Programa controlador que manejan la operación de dispositivos periféricos (hardware).
Enlace:	Es el establecimiento de comunicación entre dos puntos; esto es, cuando existe conexión física (medio de transmisión) y conexión lógica (software).
Ethernet:	Estándar de redes LAN con transmisión en topología de BUS y transmisión de 10 Mbps, con capacidad de 1,024 nodos.
FAT:	File Allocation Table. Tabla de distribución de archivos en un disco duro o flexible. Método de organización de archivos usado en DOS y OS/2.
Freeware:	Software que se distribuye sin ningún cargo y bajo ninguna condición. La propiedad la retiene el desarrollador que tiene el control de su distribución, incluyendo la capacidad de cambiar la siguiente versión del freeware a payware (software que se distribuye a cambio de dinero).



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
FTP:	File Transfer Protocol. Protocolo de TCP/IP para transferir archivos.
Gateway:	Dispositivos de comunicación de redes que permite comunicar dos redes con protocolos distintos.
GB:	GigaByte, se utiliza para representar 230 caracteres o bytes = 1,073'741,824 caracteres.
GIF:	Graphics Interchange Format. Formato de intercambio de gráficos, se utiliza para almacenar gráficos en archivos.
Gusanos:	Variante de agente destructivo (virus informático) exclusivo de las redes.
Hardware:	Componentes físicos de un sistema de cómputo.
HD:	Hard Disk. Disco duro que gira a gran velocidad y que permite almacenar y recuperar información. Es el medio de almacenamiento básico de las computadoras. También se utiliza como High Density, para designar la densidad de grabación en un disco flexible.
HTML:	Hyper Text Marcking Languaje. Lenguaje de etiquetas para definir hipertexto.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
HTTP:	Hyper Text Transport Protocol. Protocolo para la transmisión de hipertexto.
Hypertexto:	Se refiere a texto con apuntadores a otros textos, lo que permite al usuario ir navegando a otros documentos para más información de un tema dado, y luego regresar fácilmente al documento original. El hypertexto es un subconjunto de hypermedia.
Iconos:	Representación gráfica de un objeto asociado a un dispositivo, software o servicio, se utilizan en: Windows, paquetes, sistemas de cómputo y de información y en Internet.
Instalación:	En cómputo es el procedimiento relativo a colocar un software en condiciones de operación en una computadora.
Interactivo:	Término usado para referirse a la comunicación hombre-máquina, con tiempo de respuesta razonable.
Interfaz:	Referentes a conexiones e interacción de hardware, software o usuarios.
Internet:	Es una malla mundial de computadoras y redes de computadoras de todo tipo de topologías interconectadas.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
IRC:	Internet Relay Chat. Servicio de conversación sobre Internet.
ISA:	Industry Standard Architecture. Arquitectura normalizada de industria, es una especificación propuesta por Microsoft e Intel para conexión y trabajo, que proporciona configuración automática de tarjetas adaptadoras.
ISP:	Integrated Software Package. Paquete de software integrado. Véase: Paquetes de software integrado.
Java:	Lenguaje de programación orientado a objetos frecuentemente usado para aplicaciones de HTTP, Java fue creado para construir aplicaciones que saltarán a través del World Wide Web (WWW).
JPG:	Joint Photographic experts Group. Formato de archivos para desplegar imágenes.
KB:	KiloByte equivale a 210 caracteres = 1,024 bytes.
Kernel:	Núcleo o parte fundamental de los sistemas operativos que reside en memoria RAM todo el tiempo. Gestiona la memoria, los archivos, los periféricos y los recursos del sistema. Típicamente ejecuta procesos y proporciona las comunicaciones entre procesos.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
LAN:	Local Area Network. Red de área local. Es un segmento de red con estaciones de trabajo o nodos y servidores enlazados, o un conjunto de segmentos de red interconectados, por lo general dentro de la misma área, como por ejemplo un edificio.
Lenguaje de programación:	Un lenguaje usado para escribir instrucciones para la computadora. Permite que el programador exprese el procesamiento de datos de una forma simbólica sin tener en cuenta los detalles específicos de máquina. Las instrucciones que escribe el programador se denominan lenguaje fuente, y son traducidas al lenguaje de máquina de la computadora por medio de programas llamados ensambladores, compiladores e intérpretes. Como los lenguajes humanos, cada lenguaje de programación tiene su propia gramática y sintaxis. Existen muchos dialectos del mismo lenguaje, y cada dialecto requiere su propio sistema de traducción.
Login:	Nombre identificador del usuario ante un sistema de cómputo. Es de carácter público.
Macromedia:	Empresa dedicada al desarrollo de software para la elaboración de aplicaciones multimedia como el Authorware Professional.
MAN:	Metropolitan Area Network. Red de área metropolitana.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
MB:	MegaByte, se utiliza para representar 220 caracteres o bytes = 1,048,576 caracteres.
Memoria caché de disco:	Es una porción reservada de la memoria normal o memoria adicional en la tarjeta controladora del disco y que usan algunos programas con el fin de evitar múltiples lecturas al disco.
Memoria RAM-Caché:	Son bancos de memoria de alta velocidad entre la memoria normal y la CPU. Los bloques de instrucciones y datos se copian en la memoria caché, y la ejecución de las instrucciones y la actualización de los datos son llevados a cabo en la memoria de alta velocidad.
Mhz:	Megahertz. Unidad de medida para la velocidad del reloj interno de una computadoras, representa millones de ciclos por segundo.
Microprocesador:	También llamado CPU, es un chip constituido de transistores, diodos, resistencias y condensadores. La CPU, el reloj y la memoria constituyen formalmente una computadora.
Módem:	Modulador-demodulador. Dispositivo que adapta una terminal o computadora a una línea telefónica. Convierte los pulsos digitales de la computadora a frecuencias dentro del rango de audio del teléfono y los vuelve a convertir en pulsos en el lado opuesto.





<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Modular:	Proceso de modificar una señal con el fin de que pueda viajar a través de un medio físico (compare con demodular).
Monitor:	Dispositivo de salida principal en una computadora. Pantalla de cristal líquido o de rayos catódicos donde la computadora presenta información e imágenes.
Motherboard:	Es la tarjeta principal de la computadora. Cuenta con ranuras de expansión (Slots) o enchufes que ocupan un determinado número de puertos de comunicación, por los cuales la computadora se comunica con los periféricos o dispositivos de entrada y salida.
Mouse:	Dispositivo de entrada que facilita el manejo en el ambiente gráfico Windows.
MPEG:	Moving Pictures Expert Group. Formato de archivos para contener video.
Multimedia:	Es el uso de la computadora como medio de comunicación. También se define como las aplicaciones de computadora para el usuario, que integran tres o más de los cinco tipos de datos siguientes: audio, imagen fija, imagen en movimiento, texto y gráficas.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Navegar:	Referente a la acción de un usuario de buscar información a través de la red.
ORACLE:	Manejador de bases de datos para equipos micro, minis y mainframes.
Paquete:	En el contexto de redes, se refiere a los segmentos empaquetados de información que viajan a través de la red.
Paquetes:	Conjunto de programas para uso específico: procesador de palabras, hoja de cálculo, manejador de bases de datos, generador de cálculos estadísticos, generador de presentaciones, etc.
Paquetes de cómputo:	Software de uso general para aplicaciones específicas, compárese con sistemas de información.
Paquetes de software integrado:	Conjunto de paquetes de cómputo que cubren en su totalidad necesidades específicas y poseen portabilidad entre ellos.
Password:	Es una palabra clave o contraseña, para identificar si un usuario está autorizado para entrar a una computadora. Sirven como una medida de seguridad contra el acceso no autorizado a los datos. Es el identificador de carácter personal del usuario.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
PC:	Personal Computer. Computadora personal. Las PC se usan independientes o como estaciones de trabajo y servidores de archivos en una red de área local. Se usan predominantemente como sistemas de usuario único en DOS; se usan también como un computadora central en un entorno de multiusuario en UNIX y otros sistemas operativos de multiusuario, como PC-MOS de The Software Link, Inc., y Concurrent DOS de Digital Research Inc.
Periféricos:	Dispositivos hardware de entrada/salida (I/O, Input/Output) de información asociados a una computadora.
Piratería:	Acción de plagio en software en cuanto a los derechos de autor.
Pixel:	PIX [picture] Element. Elemento de imagen, pixel. El elemento más pequeño en una pantalla de presentación de video (monitor). Una pantalla se divide en miles de pequeños puntos, y un pixel es uno o más puntos que se tratan como una unidad. Un pixel puede ser un punto en una pantalla monocromática, tres puntos (rojo, verde y azul) en pantallas de color, o una agrupación de esos puntos.
Plataforma:	Arquitectura de software y hardware que soporta una aplicación.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Polimorfismo:	Característica de algunos virus informáticos en cuanto a su código de identificación cambiante; los virus con esta característica tienen la capacidad de cambiar la cadena que los identifica y por medio de la cual la vacuna los identifica y elimina.
Procesador:	Parte de la computadora que realiza el cómputo, constituido por la ALU y la unidad de control. También puede referirse a un procesador de palabras.
Programa:	Secuencia de instrucciones para una computadora que están dispuestas con una sintaxis lógica y reglamentada; a este juego de instrucciones y sintaxis definida se le llama lenguaje de programación. Véase: Lenguaje de programación.
Programa ejecutable:	Código de instrucciones listo y compilado para su ejecución en la computadora.
Programa fuente:	Programa en su forma original, tal y como lo concibió el programador en un lenguaje de programación.
Programa objeto:	Código generado a partir de un programa fuente que servirá de base para que una vez ligado se obtenga el programa ejecutable.
Protocolos:	Conjunto de reglas para lograr la comunicación entre computadoras en una red.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Puente:	Bridge. Elemento de hardware en una red de cómputo cuya función es conectar dos o más segmentos de red.
Puerto:	Conector externo en una computadora que permite la conexión de los dispositivos periféricos.
RAM:	Random Access Memory. Memoria de acceso aleatorio o memoria de trabajo, que se borra cuando falta la energía eléctrica en una computadora.
Realidad virtual:	Realidad simulada a través de la computadora.
Red:	Organización de equipos de cómputo conectados mediante un medio de conexión guiado o no para compartir recursos.
Repetidores:	Elemento de red que se encarga de restaurar las señales que sean atenuadas.
Respaldo:	Copia de software hecha con el fin de poder reponerlo en caso de daño físico o lógico.
RJ45:	Conector para cable telefónico, utilizado para armar redes de par trenzado.
Ruteador:	Router. Elemento de red que se encarga de buscar el camino más corto para los paquetes de información.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Servidor:	En una red, es una computadora compartida por múltiples usuarios.
Shareware:	Software compartido o software distribuido a través de servicios de información. Dicho software es libre de cargo sobre una condición de prueba. Si se utiliza, se paga por él, recibiendo a cambio una documentación adicional, un soporte o posiblemente la siguiente versión mejorada sin costo alguno. Se requieren licencias de pago para su distribución.
Sintaxis:	Referente a un conjunto de reglas que marcan una disposición correcta de instrucciones en un lenguaje de programación.
Sistema de computación:	También llamado sistema informático. Es el formado por una o varias computadoras (red), todos los dispositivos periféricos conectados a ella y los sistemas operativos de las computadoras de la red.
Sistema de información:	Es una aplicación para un área de una empresa o institución desarrollada en una computadora. Está constituido por las bases de datos, los programas de aplicación, los procedimientos manuales y automatizados, y abarca a los sistemas de computación o informáticos que llevan a cabo el procesamiento.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Sistema integral de información:	Es el constituido por varios sistemas de información y mediante el cual se puede apreciar en cualquier momento tanto el estado actual de la empresa o institución como el estado de cada una de sus áreas.
Sistema operativo:	Conjunto de programas que permite la administración y disposición del equipo y periféricos. Se ejecuta inmediatamente después de encender la máquina.
Slots:	Ranuras o enchufes de expansión existentes en la motherboard para conectar las tarjetas manejadoras de dispositivos periféricos.
Software:	Instrucciones o componentes lógicos para una computadora.
Tarjeta madre:	Tarjeta principal de la computadora. Véase: Motherboard.
TB:	TeraByte, se utiliza para representar 240 caracteres o bytes = 1'099,511'627,776 bytes caracteres.
TCP/IP:	Transmision Control Protocol/Internet Protocol. Protocolo de control de transmisión/protocolo Internet. Los protocolos definen las reglas de comunicación. TCP/IP se diseñó específicamente para la interconexión de diferentes tipos de equipos de computadoras.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Teclado:	Dispositivo principal de entrada de la computadora, similar en su disposición al teclado de una máquina de escribir, existen en español o inglés de 101 y 86 teclas.
Tiempo real:	Respuesta inmediata. El término puede referirse a sistemas de procesamiento rápido de transacciones para aplicaciones comerciales; no obstante, siempre se utiliza para aludir a aplicaciones de control de procesos. También puede referirse a la operación electrónica que se ejecuta en el mismo marco temporal que su contrapartida del mundo real. Por ejemplo, se requiere una computadora veloz para simular en pantalla el movimiento de modelos sólidos complejos, al mismo ritmo con que se mueven en el mundo real. La transmisión de vídeo en tiempo real produce una emisión en vivo.
Topología de red:	Clasificación de las redes de computo en cuanto a su estructura física. Se puede considerar una topología de red como un mapa de la distribución de cables. La topología define cómo se organiza el cable en las nodos individuales y desempeña un papel importante en la decisión que se tome sobre el cable. Se debe considerar la topología de una red cuando se tomen decisiones sobre qué tipo de red instalar. La topología equivale a cómo se instalará el cable a través de las paredes, pisos y techos de un edificio.





<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
UNIX:	Sistema operativo multiusuario y multitarea.
URL:	Uniform Resource Locator. Localizador uniforme de recursos. Es un método para especificar el lugar donde residen los documentos y recursos de la red en INTERNET.
Usuario:	Individuo que interactúa con la computadora al ejecutar alguna aplicación.
Utilerías:	Programas que facilitan la administración de los recursos del equipo.
Vacunas:	Programas que rastrean y eliminan los virus informáticos contenidos en los programas.
Virus informático:	Programa parásito de cómputo que puede destruir información sin que el usuario se percate en el momento.
WAN:	Wide Area Network. Red de área amplia. Son redes que cruzan límites municipales, estatales e internacionales. Los enlaces se realizan con los servicios públicos y privados de telecomunicaciones, además con los enlaces por satélites y microondas. Las WAN están constituidas por redes LAN, de CAMPUS y MAN.



<b>TÉRMINO</b>	<b>DEFINICIÓN O SIGNIFICADO</b>
Workstation:	Estación de trabajo. Microcomputadora para un usuario, de alto rendimiento, que ha sido especializada para gráficos, diseño asistido por computadora, ingeniería asistida por computadora o aplicaciones científicas. En una red de área local, una computadora personal que sirve a un usuario, a diferencia de un servidor de archivos, que sirve a todos los usuarios de la red.
WWW:	World Wide Web, Red de telaraña mundial. Dada la capacidad que tiene http para enlazarse con otros servidores, ha nacido el concepto de World Wide Web (WWW), también llamado Web o W3, que se refiere a tener a todas las computadoras compartiendo su información y poder cambiar de un servidor a otro servidor, eligiendo mediante el apuntador del mouse un botón o una frase, para tener acceso a una gran cantidad de información de una manera sencilla a través de servidores en todo el mundo.



---

## BIBLIOGRAFÍA

1. Bárcenas J., Domínguez, Tolosa. (2008). “Evaluación de la funcionalidad didáctica de software multimedia en la carrera de cirujano dentista”. Memorias SOMECE 2008.
2. Date, C. J. “Introducción a los sistemas de bases de datos”. Prentice Hall, México 2001.
3. Gándara, M. (1999) “¿Qué son los programas multimedios de aplicación educativa y cómo se usan?: una introducción al modelo “NOM” En Turren A. Coord. “Uso de nuevas Tecnologías y aplicación en la Ecuación a Distancia”. Módulos IV. V. y VI ULSA. México. ILCE: México, pp 129-152
4. Larman, Craig. “UML y Patrones”. Prentice Hall, Primera Edición, Madrid 1999.
5. Kimmel, Paul. “Manual de UML”. McGraw-Hill, México 2007,
6. Lopez Quijado, José. “Domine PHP y MySQL”. Alfaomega, México 2007.

## REFERENCIAS

[1] Daccach T., J.C. (1997-2007). Tecnologías de la Información y Comunicaciones (TIC). (sin fecha).

<http://www.gestiopolis.com/delta/term/TER434.html>

Consultado el 05 de octubre de 2008.



[2] ¿Qué es una red? (Sin fecha).

<http://www.misrespuestas.com/que-es-una-red.html>

Consultado el 05 de octubre de 2008.

[3] Sistema (informática). (2009).

[http://es.encarta.msn.com/encnet/refpages/RefArtTextonly.aspx?refid=761560074  
&find=1](http://es.encarta.msn.com/encnet/refpages/RefArtTextonly.aspx?refid=761560074&find=1)

Consultado el 21 de marzo de 2009.

[4] García Holguín, P.F. (1997). Informática Contable. (Sin fecha)

<http://www.monografias.com/trabajos48/contabilidad-informatica/contabilidad-informatica.shtml>

Consultado el 05 de octubre de 2008.

[5] Orea Cordero, I., & Díaz Rodríguez, N. (Sin fecha). Apuntes sobre redes informáticas.

<http://www.monografias.com/trabajos42/redes-informaticas/redes-informaticas2.shtm>

Consultado el 05 de octubre de 2008.

[6] Tecnología, evolución y desarrollo. (1 de Mayo de 2008).

<http://margiolet.espacioblog.com/post/2008/05/01/glosario-terminos>

Consultado el 05 de octubre de 2008.

[7] Interfaz. (2003). <http://es.wikipedia.org/wiki/Interfaz>

Consultado el 05 de octubre de 2008.



[8] Glosario de Teminos TiCs Regimen Especial. (Sin fecha).  
<http://jesusguerrero43.nireblog.com/post/2009/05/19/glosario-de-terminos-tics-regimen-especial>

Consultado el 05 de octubre de 2008.

[9] Web Sites-Portales. (Sin fecha).

<http://www.xynco.com/productos/web-sites-portales.html>

Consultado el 05 de octubre de 2008.

[10] Últimas tecnologías: sitio web. (Sin fecha).

[www.aliamultimedia.com/tecnologias/sitio\\_web.html](http://www.aliamultimedia.com/tecnologias/sitio_web.html)

Consultado el 05 de octubre de 2008.

[11] Portal (Internet). (Sin fecha). [http://es.wikipedia.org/wiki/Portal\\_\(Internet\)](http://es.wikipedia.org/wiki/Portal_(Internet))

Consultado el 05 de octubre de 2008.

[12] Quinteros Martinez, A. & Cubillos Silva, L. (Julio 2008). Software educativo.

[https://www.u-cursos.cl/filosofia/2008/1/EDU301/1/material\\_alumnos/objeto/436](https://www.u-cursos.cl/filosofia/2008/1/EDU301/1/material_alumnos/objeto/436)

Consultado el 05 de octubre de 2008.

[13] Sistema Operativo y conceptos básicos. (Sin fecha).

<http://nayar.uan.mx/~iavalos/OS.htm>

Consultado el 11 de octubre de 2008.

[14] Díaz Bastida, M. (1997).

<http://www.monografias.com/trabajos26/arquitectura-windows/arquitectura-windows.shtml>

Consultado el 26 de octubre de 2008.



[15] Bases de datos. (Sin fecha).

<http://www.monografias.com/trabajos12/basdat/basdat.shtml>

Consultado el 10 de noviembre de 2008.

[16] Ruíz, F. & Moraga de la Rubia, M. (26 de abril de 2001). El modelo de datos jerárquico.

[http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T0001\\_MAMoraga.pdf](http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T0001_MAMoraga.pdf)

Consultado el 10 de noviembre de 2008.

[17] Sistemas de bases de datos. (Sin fecha). [www.mailxmail.com](http://www.mailxmail.com)

Consultado el 20 de noviembre de 2008.

[18] ¿Qué es un servidor web (Web Servers)? - Definición de servidor web. (Sin fecha).

<http://www.masadelante.com/faqs/servidor-web>

Consultado el 24 de noviembre de 2008.

[19] Definición de INTERNET - Definición de la Palabra. (Sin fecha).

<http://www.definiciones.com.mx/definicion/I/internet/>

Consultado el 24 de noviembre de 2009.

[20] “Simposio Internacional sobre Comunidades de Aprendizaje”, Barcelona Forum 2004, Barcelona, 5-6 Octubre 2001.

## **MESOGRAFÍA**

1. ¿Qué es un servidor web?. (Sin fecha).

<http://www.masadelante.com/faq-servidor-web.htm>

Consultado el 01 de diciembre de 2008.



2. ¿Qué es un sistema operativo? (Sin fecha).

<http://www.masadelante.com/faq-sistema-operativo.htm>

Consultado el 11 de octubre de 2008.

3. Ciclo de vida del software. (Sin fecha).

[http://www.cepeu.edu.py/LIBROS\\_ELECTRONICOS\\_3/lpcu097%20-%2001.pdf](http://www.cepeu.edu.py/LIBROS_ELECTRONICOS_3/lpcu097%20-%2001.pdf)

Consultado el 01 de diciembre de 2008.

4. Colorado, D. (1998-2009). Sistema Operativo Linux. (Sin fecha).

<http://www.ilustrados.com/publicaciones/EpyppFEAVkbbGelqJB.php>

Consultado el 02 de noviembre de 2008.

5. Concepto y definición de sistemas operativos (Sin fecha).

<http://www.escolar.com/article-php-sid=52.html>

Consultado el 05 de octubre de 2008.

6. Desarrollo web. (Sin fecha). [http://es.wikipedia.org/wiki/Desarrollo\\_web](http://es.wikipedia.org/wiki/Desarrollo_web)

Consultado el 16 de diciembre de 2008.

7. Lenguajes de Programación para la Web. (Sin fecha).

[http://www.wikilearning.com/curso\\_gratis/lenguajes\\_de\\_programacion\\_para\\_la\\_web-comparacion\\_entre\\_lenguajes/6074-1](http://www.wikilearning.com/curso_gratis/lenguajes_de_programacion_para_la_web-comparacion_entre_lenguajes/6074-1)

Consultado el 02 de noviembre de 2008.

8. Los diferentes lenguajes de programación para la web. (Sin fecha).

<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>

Consultado el 10 de noviembre de 2008.



9. Mercado, W. (31 de marzo de 2007). Lenguaje de programación y software propietario.

[http://html.rincondelvago.com/lenguajes-de-programacion\\_22.html](http://html.rincondelvago.com/lenguajes-de-programacion_22.html)

Consultado el 05 de octubre de 2008.

10. MySql vs. PostgreSQL. (Sin fecha). <http://www.fedora-es.com/node/189>

Consultado el 27 de noviembre de 2008.

11. Proceso unificado. (Sin fecha). [http://es.wikipedia.org/wiki/Proceso\\_Unificado](http://es.wikipedia.org/wiki/Proceso_Unificado)

Consultado el 16 de diciembre de 2008.

12. Seguridad de la información. (Sin fecha).

[http://es.wikipedia.org/wiki/Seguridad\\_de\\_Informaci%C3%B3n](http://es.wikipedia.org/wiki/Seguridad_de_Informaci%C3%B3n)

Consultado el 01 de diciembre de 2008.

13. Seguridad informática. (Sin fecha).

[http://es.wikipedia.org/wiki/Seguridad\\_inform%C3%A1tica](http://es.wikipedia.org/wiki/Seguridad_inform%C3%A1tica)

Consultado el 01 de diciembre de 2008.

14. Seguridad informática. (Sin fecha). <http://www.jbex.net/seguridad-informatica>

Consultado el 01 de diciembre de 2008.

15. Servidor apache. (Sin fecha). <http://jakarta.apache.org/site/binindex.cgi>

Consultado el 27 de noviembre de 2008.

16. Sistemas informáticos multiusuario y en red. (2007).

<http://www.myplick.com/view/6zrVFU8OvBZ/UD01>

Consultado el 05 de octubre de 2008.





17. Sistema Operativo. (Sin fecha).

<http://www.mflor.mx/materias/comp/cursosos/sisope1.htm>

Consultado el 11 de octubre de 2008.

18. Windows y linux. (Sin fecha). <http://www.temas-estudio.com/windows-linux/>

Consultado el 14 de enero de 2009.