

77



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

## “SISTEMA DE COTIZACIÓN E INVENTARIO POR MEDIO DE UNA BASE DE DATOS Y UN SITIO WEB”

T E S I S

Que para obtener el título de :  
INGENIERO EN COMPUTACIÓN  
p r e s e n t a n :

**Miriam Morán Pacheco /**  
**Edgar Iván Romero Celis**

Director de tesis: M.I. Benjamín Valera Orozco

Sinodales:

Ing. Alberto Templos Carbajal  
Ing. Ana María Vázquez Vargas  
Ing. Lucila Patricia Arellano Mendoza  
Ing. Gabriela B. Lizárraga Ramírez



MÉXICO, D. F.

TESIS CON  
FALLA DE ORIGEN

2002.



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

# PAGINACION DISCONTINUA

Agradezco a Dios por bendecirme con unos maravillosos padres que me han dado su amor, ejemplo y apoyo para alcanzar mis metas.

A mi familia por su cariño, palabras de aliento, paciencia y alegría.

A Israel por su amor y entrega.

A mis amigos de toda la vida Octavio, Iván, Gustavo, Doroteo, Manuel y Georgina por sus sonrisas y compañía en la Universidad.

A Raúl y Carmen por su motivación para que terminara la tesis.

A Guillermo por darme tiempo para concluir mi tesis.

A Benjamín por su amistad y paciencia.

Al Ing. Rigoberto Nava Sandoval por el apoyo brindado en el Centro de Instrumentos.

Miriam

Le doy gracias a DIOS por darme la oportunidad de seguir con vida y la salud que me ha permitido finalizar este proyecto.

A mis padres por su apoyo, paciencia y fe.

A mi hermano por estar ahí cuando me ha hecho falta y ser mi compañero y amigo en este mundo.

A Lila por todo el amor, cariño, paciencia, apoyo y confianza que siempre me ha dado.

A mi abuelita por su amor, cariño, comprensión y por estar siempre en mi vida.

A mi familia por su apoyo, palabras, y fe que siempre me han tenido.

A la Universidad por la formación, que me ha dado y el espíritu que ha inculcado en mí.

A mis maestros que con su ejemplo me han ayudado a mejorar y siempre dar lo mejor.

A Benjamín por su increíble apoyo y paciencia y la culminación del proyecto.

Al Ing. Rigoberto Nava Sandoval por sus consejos y apoyo.

A mis amigos Gerardo, Esther, Armando, Benjamín, Gina, Manuel, Zelene, Brenda, Julio, Octavio, Miguel, Marú, Erika por todos los momentos divertidos, por la solidaridad pero sobretodo por su amistad.

A Miriam por su entusiasmo, capacidad y humor que sin ellos no hubiese sido igual.

Edgar Iván

# Í N D I C E

<b>INTRODUCCIÓN .....</b>	<b>I</b>
<b>OBJETIVOS .....</b>	<b>I</b>
<b>DEFINICIÓN DEL PROBLEMA .....</b>	<b>I</b>
<b>ENTORNO ACTUAL Y TENDENCIAS .....</b>	<b>I</b>
<b>SISTEMAS EN SITIOS WEB .....</b>	<b>II</b>
<b>PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>III</b>
<b>MÉTODO.....</b>	<b>III</b>
<b>ALCANCE.....</b>	<b>IV</b>
<b>RELEVANCIA Y JUSTIFICACIÓN.....</b>	<b>V</b>
<b>RELACIÓN CON OTRAS ÁREAS.....</b>	<b>V</b>
<b>INVENTARIO DE MATERIAS .....</b>	<b>V</b>
<b>RESULTADOS ESPERADOS .....</b>	<b>VI</b>
<b>RESUMEN DE LA TESIS .....</b>	<b>VI</b>
<b>CAPITULO 1. CONCEPTOS BÁSICOS .....</b>	<b>1</b>
<b>1.1 INGENIERÍA DE SOFTWARE.....</b>	<b>1</b>
1.1.1 EL PRODUCTO Y EL PROCESO .....	1
1.1.2 CRONOLOGÍA DE SOFTWARE .....	2
1.1.3 EL SOFTWARE.....	5
1.1.4 EL PROCESO .....	9
<b>1.2 CONCEPTOS GENERALES DE BASES DE DATOS RELACIONALES.....</b>	<b>36</b>
1.2.1 ARQUITECTURA DE LA BASE DE DATOS.....	36
1.2.2 ESQUEMA CONCEPTUAL DE DATOS .....	37
1.2.3 CLASIFICACIÓN DE LOS SISTEMAS DE BASES DE DATOS.....	39
1.2.4 BASE DE DATOS RELACIONAL .....	40

<b>1.3 INTERNET</b> .....	<b>47</b>
1.3.1 ANTECEDENTES .....	47
1.3.2 SERVICIOS A TRAVÉS DE INTERNET.....	51
1.3.3 EL MODELO CLIENTE-SERVIDOR.....	52
1.3.4 MODELO INTERNET .....	53
1.3.5 TCP/IP.....	54
1.3.6 WORLD WIDE WEB.....	55
<b>1.4 SOFTWARE LIBRE EN INTERNET</b> .....	<b>63</b>
1.4.1 SISTEMAS OPERATIVOS GNU. ....	64
1.4.2 MANEJADOR DE BASES DE DATOS RELACIONALES GNU. ....	65
1.4.3 LENGUAJES DE PROGRAMACIÓN GNU. ....	66
1.4.4 SERVIDORES WEB.....	67
<b>CAPITULO 2. DISEÑO DEL SISTEMA</b> .....	<b>68</b>
<b>2.1 ANÁLISIS DE LA PROBLEMÁTICA ACTUAL</b> .....	<b>68</b>
2.1.1 PROCEDIMIENTO DE SOLICITUD-COTIZACIÓN-INVENTARIO.....	69
<b>2.2 DIAGRAMAS DE FLUJO DE DATOS</b> .....	<b>76</b>
<b>2.3 DISEÑO Y MODELADO DE LA BASE DE DATOS</b> .....	<b>81</b>
2.3.1 ENTIDADES Y ATRIBUTOS .....	81
2.3.2 PROCESO DE NORMALIZACIÓN .....	84
2.3.3 MODELO ENTIDAD/RELACIÓN .....	90
2.3.4 DICCIONARIO DE DATOS .....	91
2.3.5 PLANIFICACIÓN.....	93

<b>CAPITULO 3. IMPLEMENTACIÓN DEL SISTEMA.....</b>	<b>94</b>
3.1 IMPLANTACIÓN Y ADECUACIÓN DE LAS HERRAMIENTAS DE SOFTWARE.....	96
3.1.1 INSTALACIÓN DE LINUX.....	96
3.1.2 INSTALACIÓN DE APACHE.....	116
3.1.3 INSTALACIÓN DE MYSQL.....	118
3.1.4 INSTALACIÓN DEL LENGUAJE DE PROGRAMACIÓN PHP4.....	119
3.1.5 INTEGRACIÓN DE APACHE-MYSQL-PHP4.....	120
3.2 DISEÑO Y DESARROLLO DE PÁGINAS DEL SISTEMA.....	122
3.2.1 PÁGINAS DE MANTENIMIENTO.....	123
3.2.2 PÁGINAS PÚBLICAS.....	155
3.3 PRUEBAS DEL SISTEMA.....	166
<b>CAPITULO 4. CONCLUSIONES.....</b>	<b>174</b>
<b>BIBLIOGRAFÍA.....</b>	<b>177</b>
<b>ANEXO. MANUAL DE USUARIO.....</b>	<b>178</b>
ACCESO AL SISTEMA.....	178
CONSULTAR SERVICIOS.....	179
AGREGAR SERVICIO.....	181
CAMBIAR VIGENCIA DE SERVICIO.....	183
MODIFICAR SERVICIO.....	184
CAMBIAR INFORMACIÓN DE LA SECCIÓN.....	186
LISTAR METRÓLOGOS.....	188
AGREGAR METRÓLOGO.....	189
CAMBIAR VIGENCIA DE UN METRÓLOGO.....	191
MODIFICAR DATOS DEL METRÓLOGO.....	193
CONSULTAR COTIZACIONES.....	194
GENERAR ORDENSM MANUALMENTE.....	198
GENERAR ORDENSM CON UNA COTIZACIÓN DE INTERNET.....	200
BUSCAR ORDENSM.....	202
IMPRIMIR ORDENSM.....	204
MODIFICAR STATUS DE LA ORDENSM.....	207



---

## INTRODUCCIÓN

### OBJETIVOS

- Implementar un sistema para la difusión de los proyectos y servicios que se realizan en el Laboratorio de Metrología del Centro de Instrumentos (CI) en la Universidad Nacional Autónoma de México (UNAM) mediante un sitio Web.
- Automatización de los servicios de cotización e inventario de las órdenes de servicio del Laboratorio de Metrología, apoyándose en un sitio de Web y base de datos relacional.

### DEFINICIÓN DEL PROBLEMA

#### ENTORNO ACTUAL Y TENDENCIAS

En la actualidad en el CI, particularmente en el Laboratorio de Metrología, se desempeñan actividades de medición dimensional certificadas que requieren un procedimiento administrativo que comprenden las siguientes tareas:

- Recepción de solicitudes de cotización
- Elaboración de cotizaciones
- Recepción de equipo y registro de órdenes de servicio
- Elaboración de certificados de medición o calibración
- Entrega de equipo y certificados

Dicho procedimiento, *solicitud-cotización-inventario*, se realiza en forma manual, teniendo registrado el procedimiento únicamente en documentos escritos. El procedimiento requiere gran apoyo del personal administrativo para atender solicitudes, elaborar inventarios, cotizar, y el seguimiento del proceso.

Las dificultades que se presentan en el proceso administrativo de atención a clientes son:

- Durante la recepción de solicitudes de cotización el tiempo de respuesta, al usar los medios convencionales como mensajería o fax, es ineficiente, poniendo en riesgo la confiabilidad del proceso de atención al cliente.
- El tiempo de respuesta a una solicitud de cotización es aproximadamente de 48 horas, lo que implica una pérdida de tiempo tanto para el cliente como para el mismo laboratorio.
- El proceso manual de registro de órdenes de servicio se convierte en un proceso redundante de duplicación de la información de las solicitudes de cotización.
- Se invierte una cantidad considerable de recursos humanos en el proceso de *solicitud-cotización-inventario*, tanto del personal administrativo como técnico.
- El seguimiento del proceso *solicitud-cotización-inventario* se realiza con el apoyo de bitácoras escritas, entorpeciendo búsquedas eficientes. La administración y manejo de documentos escritos presenta serias carencias frente a la administración electrónica.
- La descentralización de la información genera retrasos y duplicidades que hacen ineficiente la respuesta esperada por el cliente.

## SISTEMAS EN SITIOS WEB

Las dificultades expuestas anteriormente son ampliamente conocidas y las empresas prestadoras de servicios o bienes de consumo las enfrentan diariamente. Ante tal situación, la red de Internet y las bases de datos proporcionan una herramienta que en conjunto ofrecen una solución eficiente. Un ejemplo elegante, pionero y exitoso es el sitio Web [www.amazon.com](http://www.amazon.com), que ha logrado beneficios increíbles en la venta de libros, discos compactos, videos y regalos.

## PLANTEAMIENTO DEL PROBLEMA

Hoy en día las empresas con presencia en Internet les da la oportunidad de alcanzar un nuevo mercado, además de poder intercambiar documentos en línea, eliminando gastos y problemas de impresión de información obsoleta.

Actualmente el Laboratorio de Metrología no ofrece servicios en un sitio Web, lo que ayudaría a optimizar la participación de recursos humanos, aumentaría la difusión de los servicios que presta el laboratorio, reduciría los tiempos de operación e incrementaría la productividad al tener mayor captación de clientes potenciales.

En éste sentido se plantea la creación de un sistema de cotización e inventario utilizando las tecnologías de programación en Internet y el manejo de bases de datos para agilizar el proceso de *solicitud-cotización-inventario* del Laboratorio de Metrología. El problema enfrentado entonces es crear un sistema electrónico que reemplace el proceso manual llevado a cabo hasta la fecha.

## MÉTODO

Se aplicará la metodología del modelo de espiral de la ingeniería de software para la creación de un sistema de información acorde al Sistema de Calidad del Laboratorio de Metrología. Para ello se optimizarán los recursos tecnológicos y humanos con que se cuenten.

Dicha metodología comprende cuatro etapas:

1. *Determinación de objetivos, alternativas y restricciones:* Determinación de la problemática.
2. *Análisis de riesgos:* Evaluación de riesgos técnicos y entrega de prototipos.
3. *Desarrollo y pruebas:* Construir una o más representaciones de la aplicación.
4. *Planeación:* Definir recursos, tiempos y otra información relacionada con el proyecto.

Para eliminar costos al Laboratorio de Metrología se instalará el siguiente software GNU (GNU es un acrónimo recursivo para "GNU No es Unix") para soportar el sistema:

- Sistema operativo LINUX
- Servidor Web APACHE
- Servidor de base datos MySQL
- Lenguaje de programación PHP.

Finalmente se desea una etapa de fácil mantenimiento que pueda ser implementada en una interfaz amigable para el usuario, preferentemente en una página Web.

## ALCANCE

El sistema de cotización e inventario esta restringido a la automatización del proceso *solicitud-cotización-inventario* que forma parte del Sistema de Calidad específico para el Laboratorio de Metrología del CI UNAM.

Aprovechando la tecnología de Internet y con el fin de agilizar los procesos administrativos del laboratorio se buscarán las siguientes características:

- Mayor difusión de los servicios con la mínima inversión de recursos.
- Centralización de la información involucrada en el proceso.
- Intercambio de información electrónica entre cliente y proveedor.
- Acceso a la información de la base de datos desde diferentes localidades.
- Automatización completa del proceso *solicitud-cotización-inventario*.

Con la implementación del sistema de solicitud, cotización e inventario que se propone en éste trabajo de tesis, se pretende lograr un sistema informático en línea (a través de la Web) capaz de brindar la atención esperada por el cliente de los servicios de calibración que se ofrecen en el Laboratorio de Metrología, disponible las 24 horas del día.

## RELEVANCIA Y JUSTIFICACIÓN

La necesidad en el Laboratorio de Metrología de automatizar las tareas de oficina surge porque hoy en día las empresas difícilmente se pueden permitir no tener una presencia en Internet. Una razón es el costo de los anuncios tradicionales ya sea difusión impresa, por radio o televisión.

Otra razón digna de tomarse en cuenta para estar en un sitio Web es que vivimos en una economía global muy competitiva que ningún negocio puede darse el lujo de ignorar. Con un sitio Web se puede llegar a las personas más rápido que con cualquier otro medio. Hay muchas empresas de diversos campos profesionales que utilizan diariamente Internet para adquirir productos y servicios. Además ofrece la ventaja de poder intercambiar los documentos en línea con un efecto instantáneo, eliminando el gasto y los problemas potenciales de imprimir información obsoleta.

## RELACIÓN CON OTRAS ÁREAS

Existe una gran relación con las áreas de control de calidad, diseño de software, sistemas de gestión de bases de datos, administración de empresas, estrategias de mercado electrónico, metrología dimensional.

## INVENTARIO DE MATERIAS

- Computadoras y programación.
- Técnicas de investigación de operaciones.
- Ingeniería de programación.
- Introducción a la economía.
- Bases de datos.
- Redes de Computadoras.
- Recursos y necesidades de México.
- Organización y administración de centros de cómputo.

## RESULTADOS ESPERADOS

Se espera contar con un sistema de cotización e inventario capaz de automatizar las tareas *solicitud-cotización-inventario* que se realizan en el Laboratorio de Metrología por medio de la integración de un sistema de gestión de bases de datos y un sitio Web.

Se pretende dar mayor difusión a los servicios y proyectos que se desarrollan en el Laboratorio de Metrología.

IncurSIONAR en el ambiente de comercio electrónico para propiciar la mejora continua de la calidad.

## RESUMEN DE LA TESIS

El contenido de la tesis consta de cuatro capítulos divididos de la siguiente forma: (1) Conceptos básicos, (2) Diseño del sistema, (3) Implementación del sistema, (4) Conclusiones.

El capítulo uno, abarca los conceptos básicos involucrados en los temas en los que se basó el trabajo de tesis y que fueron fundamentales en el desarrollo de la misma. Tales conceptos comienzan con la teoría de ingeniería de software, así como los conceptos generales de las bases de datos, conceptos de un sitio Web. En este capítulo se describen las bases de programación para Internet y bases de datos.

El capítulo dos, abarca el diseño del sistema, pasando por el análisis de la problemática actual, los procedimientos *solicitud-cotización-inventario*, el procedimiento de cotización e inventario, el diseño, modelado de datos, modelado de la base de datos.

El capítulo tres contiene la parte sustancial del desarrollo de la tesis. En este se describe la implementación del sistema, incluyendo los preparativos, la implementación del sitio Web, las pruebas del sistema y el mantenimiento del mismo.

El capítulo cuatro, resalta los resultados obtenidos en el sistema de cotización e inventario, en el que se puede apreciar la automatización de los servicios de cotización e inventario de las órdenes de servicio del Laboratorio de Metrología, apoyados en un sitio Web y en una base de datos relacional. Al final de la tesis se incluye un anexo con la documentación del manual de usuario.

## **CAPITULO 1. CONCEPTOS BÁSICOS**

### **1.1 INGENIERÍA DE SOFTWARE.**

Actualmente las economías de los países desarrollados (en el mayor de los casos y en menor proporción los países no desarrollados) dependen en gran parte del software, de ahí la importancia del tema. Así mismo, el ambiente en que se desarrolla la sociedad ha ido exigiendo la utilización de software para optimización y automatización en los procesos ya establecidos, por eso en la actualidad más sistemas son controlados por software.

La ingeniería de software es una disciplina de la computación que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelva confiablemente problemas de todo tipo de modo rentable. La ingeniería de software trata con áreas muy diversas de la computación, tales como la construcción de compiladores, sistemas operativos o desarrollos de Intranet/Internet/Extranet, abordando todas las fases del ciclo de vida del desarrollo de cualquier sistema de información aplicables a una infinidad de áreas tales como: empresarial, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, etc.

#### **1.1.1 EL PRODUCTO Y EL PROCESO**

En esta parte de la ingeniería de software, el producto se considera que va a ser tratado con ingeniería y que el proceso proporciona un marco de trabajo para la tecnología del software.

#### **EL PRODUCTO**

El software de computadora se ha vuelto indispensable. Está inmerso en sistemas de todo tipo: de transportes, médicos, de telecomunicaciones, militares, de procesos industriales, entrenamientos, productos de oficina, en fin, la lista sería interminable. Todo esto ha cambiado la percepción pública del software. Los programas informáticos están en todas partes y la gente los ve como un hecho tecnológico de la vida

Podemos clasificar a los productos de software como:

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

La mayor parte del gasto del software es en productos genéricos, pero hay más esfuerzo en el desarrollo de los sistemas hechos a medida.

## CARACTERÍSTICAS DE LOS PRODUCTOS DE SOFTWARE

Los productos de software deben de contar con ciertas características básicas las cuales se listan a continuación:

- *Mantenibles:* Debe ser posible que el software evolucione y que siga cumpliendo con sus especificaciones.
- *Confiable:* El software no debe causar daños físicos o económicos en caso de fallas.
- *Eficiencia:* El software no debe desperdiciar los recursos del sistema.
- *Utilización adecuada:* El software debe contar con una interfaz de usuario adecuada y documentación.

### 1.1.2 CRONOLOGÍA DE SOFTWARE

Durante los primeros años de la era de la computadora, el software se veía como un añadido. La programación de computadoras era muy casera, existían pocos métodos sistemáticos. El desarrollo del software se realizaba virtualmente sin ninguna planificación.

Durante los primeros años, lo común era que el hardware fuera de propósito general. Por otra parte el software se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña. El software como producto (es decir, programas desarrollados para ser vendidos a uno o más clientes) estaba aún en sus inicios. La mayoría del software se desarrollaba y era utilizado por la misma persona u organización. La misma persona lo escribía, lo ejecutaba y si fallaba, lo depuraba.



Debido a este entorno personalizado del software, el diseño era un poco implícito, realizado en la mente de alguien y la documentación normalmente no existía.

La segunda era en la evolución de los sistemas de computadora se extiende desde la mitad de la década de los 60 hasta mediados de los 70.

La multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre-máquina. Las técnicas interactivas abrieron un nuevo mundo de aplicaciones y nuevos niveles de sofisticación del hardware y del software. Los sistemas de tiempo real podían recoger, analizar y transformar datos de múltiples fuentes, controlando así los procesos y produciendo salidas en milisegundos en lugar de minutos. Los avances en los dispositivos de almacenamientos en línea condujeron a la primera generación de sistemas de gestión de datos.

La segunda era se caracterizó también por el establecimiento del software como producto y la llegada de "las casas de software". El software ya se desarrollaba para tener una amplia distribución en un mercado multidisciplinario. Los programas se distribuían para computadoras grandes y para mini computadoras, a cientos e incluso a miles de usuarios.

La gente de la industria, del gobierno y de la universidad se disponía a "desarrollar" el mejor paquete de software y ganar dinero.

Conforme crecía el número de sistemas informáticos, comenzaron a extenderse las bibliotecas de software de computadoras. Todos esos programas, todos esos códigos fuente, tenían que ser corregidos cuando se detectaban fallos, modificarlos cuando cambiaban los requerimientos de los usuarios o adaptarlos a nuevos dispositivos de hardware que se hubieran adquirido. Estas actividades se llamaron colectivamente mantenimiento del software. La naturaleza personalizada de muchos programas los hacía virtualmente imposibles de mantener. Había comenzado una crisis del software.

La tercera era de la evolución de los sistemas de computadora comenzó a mediados de los años 70 y continuó más allá de una década. El sistema distribuido de múltiples computadoras, cada una ejecutando funciones concurrentemente y comunicándose una con la otra, incrementó notablemente la complejidad de los sistemas informáticos. Las redes de área local y de área global, las comunicaciones digitales de alto ancho de banda y la creciente demanda de acceso "instantáneo" a los datos, pusieron una fuerte presión en los desarrolladores de software.

La conclusión de la tercera era se caracterizó por la llegada y amplio uso de los microprocesadores. El microprocesador ha producido un extenso grupo de productos inteligentes, desde automóviles hasta hornos de microondas, desde robots industriales a equipos de diagnósticos de suero sanguíneo, pero ninguno tan importante como la computadora personal.

La cuarta era de la evolución de los sistemas informáticos se aleja de las computadoras individuales y de los programas de computadoras, dirigiéndose al impacto colectivo de las computadoras y del software. Poderosas máquinas personales controladas por sistemas operativos sofisticados, en redes globales y locales, acompañadas por aplicaciones de software avanzadas, se han convertido en la norma. Las arquitecturas informáticas están cambiando de entorno descentralizados cliente/servidor.

La industria del software es parte importante de la economía del mundo, han comenzado a surgir tecnologías nuevas, tecnologías orientadas a objetos, que están desplazando rápidamente los enfoques de desarrollo de software más convencionales en muchas áreas de aplicaciones. Las técnicas de cuarta generación para el desarrollo del software están cambiando en la forma en que la comunidad del software construye programas informáticos.

En la figura 1.1 se muestra la evolución de los sistemas de ingeniería de software.

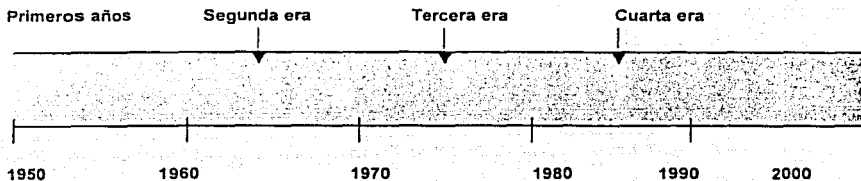


Figura 1.1 Evolución de los sistemas de ingeniería de software

### 1.1.3 EL SOFTWARE

En los años de 1970, el 1% de las personas podría haber descrito inteligentemente lo que significaba "software de computadora". Hoy, la mayoría de los profesionales y muchas personas en general piensan que comprenden el software. ¿Pero lo entienden realmente?

La descripción de software en un libro de texto podría tomar la forma siguiente:

El software es:

1. Instrucciones (programas de computadora) que cuando se ejecutan proporcionan la función y el rendimiento deseados.
2. Estructuras de datos que permiten a los programas manipular adecuadamente la información.
3. Documentos que describen la operación y el uso de programas.

### CARACTERÍSTICAS DEL SOFTWARE

Para poder entender que es el software y consecuentemente que es la ingeniería de software, es importante examinar las características del software que lo diferencian de otras cosas que los hombres pueden construir. Cuando se construye hardware, el proceso creativo humano (análisis, diseño, construcción, prueba) se traduce finalmente en una forma física. Si construimos una nueva computadora, nuestro boceto inicial, diagramas formales de diseño y prototipo de prueba, evolucionan hacia un producto físico (circuitos impresos, fuentes de potencia, etc.).

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

- 1.- El software se desarrolla, no se fabrica en un sentido clásico. En ambas actividades la buena calidad se adquiere mediante un buen diseño, pero la fase de construcción de hardware puede introducir problemas de calidad que no existen o que son fáciles de corregir en el software. Ambas actividades dependen de personas pero la relación entre las personas dedicadas y el trabajo realizado es diferente en el software. Ambas actividades requieren la construcción de un producto, pero los métodos son diferentes.

Los costos del software se encuentran en la ingeniería. Esto significa que los proyectos de software no se pueden gestionar como si fueran proyectos de fabricación

2.- El software no se estropea. El software no es susceptible a los males del entorno que hacen que el hardware se estropee. Por tanto, en teoría, los defectos no detectados harán que falle el programa durante las primeras etapas de su vida. Sin embargo, una vez que se corrigen, suponiendo que no se introducen nuevos errores, la curva de fallos se aplana. Sin embargo aunque el software no se daña, si se deteriora. Es decir, durante su vida, el software sufre cambios, es bastante probable que se introduzcan nuevos defectos. Lentamente el nivel mínimo de fallos comienza a crecer, el software se va deteriorando debido a los cambios.

Otro aspecto de ese deterioro ilustra la diferencia entre el hardware y el software. Cuando un componente de hardware se estropea, se sustituye por una pieza de repuesto. No hay piezas de repuesto para el software. Cada fallo en el software indica un error en el diseño o en el proceso mediante el que se tradujo el diseño a código máquina ejecutable. Por tanto el mantenimiento del software tiene una complejidad considerablemente mayor a la del mantenimiento de hardware.

3.- La mayoría del software se construye a medida, en vez de ensamblar componentes existentes. Es decir, la parte que se necesita no se puede pedir por catálogo, sino que se tiene que programar en el momento. Se puede comprar software ya desarrollado, pero solo como una unidad completa, no como componentes que pueden reensamblarse en nuevos programas.

## COMPONENTES DE SOFTWARE

En el mundo del hardware, la reutilización de componentes es una parte natural del proceso de ingeniería. En el mundo del software, es algo que todavía se tiene que lograr a una escala amplia.

La reutilización es una característica importante para un componente de software de alta calidad. El componente debería diseñarse e implementarse para que pueda vivir y ser utilizado en muchos programas diferentes. Los componentes reutilizables modernos encapsulan tanto datos como procesos que se aplican a los datos, permitiendo al ingeniero del software crear nuevas aplicaciones a partir de las partes reutilizables.

Los componentes de software se construyen mediante un lenguaje de programación que tiene un vocabulario limitado, una gramática definida explícitamente y reglas bien formadas de sintaxis y semántica. En un nivel más bajo el lenguaje refleja el conjunto de instrucciones de hardware, en el nivel medio los lenguajes de programación tales como C o Smalltalk, se usan para crear

descripción procedimental del programa, en el nivel más alto el lenguaje utiliza íconos gráficos u otra simbología para representar los requisitos para una solución.

Los lenguajes máquina son una representación simbólica del conjunto de instrucciones de la CPU. Si un buen programador produce programas de fácil mantenimiento y bien documentados, puede utilizar el lenguaje máquina para hacer un uso extremadamente eficiente de la memoria y para optimizar la velocidad de elección del programa.

Los lenguajes de alto nivel, permiten al programador y al programa independizarse de la máquina. Cuando se utiliza un traductor sofisticado, el vocabulario, la gramática, la sintaxis y la semántica de un lenguaje de alto nivel pueden ser mucho más sofisticados que los lenguajes máquina. De hecho los compiladores e intérpretes de los lenguajes de alto nivel producen lenguaje máquina como salida.

El código máquina, los lenguajes ensambladores (nivel máquina) y los lenguajes de programación de alto nivel, son normalmente considerados como las tres primeras generaciones de lenguajes de computadora. Con cualquiera de estos lenguajes, el programador ha de preocuparse tanto de la especificación de la estructura de la información como de la de control del propio programa. Por ello, los lenguajes de las tres primeras generaciones se llaman lenguajes procedimentales.

Los lenguajes de la cuarta generación son también denominados lenguajes no procedimentales, llevan al desarrollador del software mucho más allá del hardware. En vez de requerir que quien desarrolla el software especifique los detalles procedimentales, un programa en un lenguaje no procedimental es la especificación del resultado deseado, en vez de la especificación de la acción requerida para conseguir el resultado. El software de soporte que traduce la especificación del resultado en un programa máquina ejecutable.

## APLICACIONES DE SOFTWARE

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimentales, es decir un algoritmo.

El contenido y determinismo de la información son factores importantes a considerar para definir la naturaleza de una aplicación de software. El contenido se refiere al significado y a la forma de una aplicación de entrada y de salida.

El determinismo de la información se refiere a la predicción del orden y del tiempo de llegada de los datos. Un sistema operativo multiusuario, por otra parte acepta entradas que tienen un contenido variado y que se producen en instantes arbitrarios, ejecuta algoritmos que pueden ser interrumpidos por condiciones

externas y produce una salida que depende de una función del entorno y del tiempo. Las aplicaciones con estas características se dice que son indeterminadas.

## CLASIFICACIÓN DEL SOFTWARE

### Software de Sistemas

El software de sistemas es un conjunto de programas que han sido escritos para servir a otros programas. Algunos programas de sistemas (por ejemplo: compiladores, editores y utilidades de gestión de archivos) procesan estructuras de información complejas pero determinadas. Por ejemplo otras aplicaciones de sistemas (por ejemplo, ciertos componentes del sistema operativo, utilidades de manejo de periféricos, procesadores de telecomunicaciones) procesan datos en gran medida indeterminados. En cualquier caso, el área de sistemas se caracteriza por una fuerte interacción con el hardware de la computadora, una gran utilización por múltiples usuarios, una operación concurrente que requiere una planificación, compartir recursos y una sofisticada gestión de procesos, unas estructuras de datos complejas y múltiples interfaces externas. Un ejemplo del software de sistemas es el sistema LINUX figura 1.2.



Figura 1.2 LINUX

### Software de tiempo real

El software que mide, analiza, y/o controla sucesos de tiempo real conforme ocurren, se denomina de tiempo real. Entre los elementos del software de tiempo real se incluyen: un componente de adquisición de datos, que recolecta y da formato a la información recibida del entorno externo, un componente de análisis que transforma la información según lo que requiera la aplicación, un componente de control-salida que responda al entorno externo y un componente de monitorización que coordina todos los demás componentes, de forma que pueda mantenerse la respuesta en tiempo real (típicamente en el rango de un milisegundo a un minuto). Hay que tomar en cuenta que el término de "tiempo real" tiene un significado diferente de "interactivo" o "compartido".

TESIS CON  
FALLA DE ORIGEN

### Software de gestión

El procesamiento de información comercial constituye la mayor de las áreas de aplicación del software. Los sistemas discretos (por ejemplo, nóminas, cuentas de débitos, inventarios, etc.) han evolucionado hacia el software de sistemas de información de gestión (SIG), que acceden a una o más bases de datos grandes que contienen información comercial tales como Oracle, Sybase, Mysql, SQL Server por mencionar algunos.

### Software de ingeniería y científico

El software de ingeniería y científico, está caracterizado por el manejo de números. Las aplicaciones van desde la astronomía a la vulcanología, desde la biología molecular a la fabricación automática.

### Software de Inteligencia Artificial

El software de inteligencia artificial (AI) hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo, actualmente el área más activa de la AI es la de los sistemas expertos, también llamados sistemas basados en el conocimiento. Otras áreas de aplicación para el software de AI son el reconocimiento de patrones (imágenes y voz), la prueba de teoremas y los juegos. En los últimos años se ha desarrollado una nueva rama del software de AI llamada redes neuronales artificiales. Una red neuronal, simula la estructura de proceso del cerebro (las funciones de la neurona biológica) y a la larga puede llevar a una clase de software que pueda reconocer patrones complejos y aprender de "experiencia" pasada.

## 1.1.4 EL PROCESO

Definimos al proceso de software como un marco de trabajo de las tareas que se requieren para construir software de alta calidad. Un proceso de software define el enfoque que se toma cuando el software es tratado por la ingeniería, y la tecnología del software contempla las tecnologías que llenan el proceso (métodos técnicos y herramientas automatizadas).

## Ingeniería de software, una tecnología estratificada

La ingeniería de software es una tecnología multicapa a fin de obtener económicamente software que sea fiable y eficiente. Cualquier enfoque de ingeniería (incluida ingeniería de software) debe descansar sobre un empeño de organización de calidad. La gestión total de calidad y las filosofías similares fomentan una cultura continua de mejoras de procesos y es esa cultura la que conduce últimamente al desarrollo de enfoques cada vez más robustos para la ingeniería de software. Los cimientos que son la base de la ingeniería de software están orientados hacia la calidad. En la figura 1.3 se muestran las capas de la ingeniería de software.

### Ingeniería del software

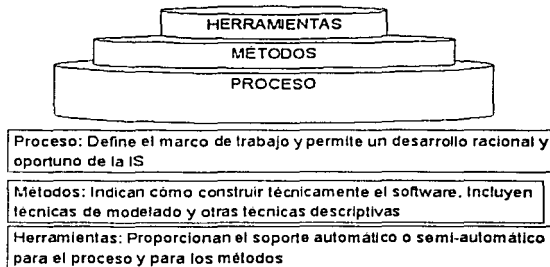


Figura 1.3 Capas de la Ingeniería de Software

El fundamento de la ingeniería de software es la capa proceso. El proceso de la ingeniería de software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería de software. El proceso que define un marco de trabajo para un conjunto de áreas clave de proceso que se debe establecer para la entrega efectiva de la tecnología de la ingeniería de software. Las áreas clave del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se producen resultados del trabajo (modelos, documentos, datos, informes, formularios, etc.), se establecen objetivos, se asegura la calidad y el cambio se gestiona adecuadamente.

Los métodos de la ingeniería de software indican como construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Los métodos de la ingeniería de software dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.



Las herramientas de la ingeniería de software proporcionan un soporte automático o semi-automático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado Ingeniería de software Asistida por Computadora (Computer Aided Software Engineering, CASE). Las herramientas CASE combinan software, hardware y una base de datos de ingeniería de software (un depósito que contiene información importante sobre el análisis, el diseño, la construcción de programas y pruebas) para crear un entorno de ingeniería de software que sea análogo para el hardware.

Una visión general de la ingeniería de software

La ingeniería es el análisis, diseño, construcción, verificación y gestión de entidades técnicas (o sociales). Con independencia de la entidad a la que se va aplicar ingeniería, se deben cuestionar y responder las siguientes preguntas:

- ¿Cual es el problema a resolver?
- ¿Cuales son las características de la entidad que utiliza para resolver el problema?
- ¿Cómo se realizará la entidad y la solución?
- ¿Como se construirá la entidad?
- ¿Que enfoque se utilizará para no contemplar los errores que se cometieron en el diseño y en la construcción de la entidad?

Para construir la ingeniería de software adecuadamente, se debe definir un proceso de desarrollo de software. En esta sección se consideran las características genéricas del proceso de software. El trabajo que se asocia a la ingeniería de software se puede dividir en tres fases genéricas, con independencia del área de aplicación tamaño o complejidad del proyecto. Cada fase se enfrenta con una o varias cuestiones de las destacadas anteriormente.

## FASES DEL PROCESO DEL SOFTWARE

### Fase de definición

La fase de *definición* se centra sobre *el qué*. Es decir, durante la definición, el que desarrolla el software intenta identificar que información ha de ser procesada, que función y rendimiento se desea, que comportamiento del sistema, que interfaces van a ser establecidas, que restricciones de diseño existen y que criterios de validación se necesitan para definir un sistema correcto. Por tanto, han de identificarse los requisitos clave del sistema y del software. Aunque los métodos aplicados durante la fase de definición variarán, de cualquier forma, estos tendrán lugar tres tareas principales: ingeniería de sistemas de información, planificación del proyecto de software, y análisis de los requisitos.

### Fase de desarrollo

La fase de *desarrollo* se centra en *el cómo*. Es decir, durante el desarrollo un ingeniero de software intenta definir como han de diseñarse las estructuras de datos, como ha de implementarse la función como una arquitectura del software, como han de implementarse detalles procedimentales, como han de caracterizarse las interfaces, como ha de traducirse el diseño en un lenguaje de programación (o en un lenguaje no procedimental) y como ha de realizarse las pruebas. Los métodos aplicados durante la fase de desarrollo variarán, aunque las tareas específicas técnicas deberían ocurrir siempre: diseño del software, generación de código y prueba del software.

### Fase de mantenimiento

La fase de *mantenimiento* se centra en el *cambio* que va asociado, a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. La fase de mantenimiento vuelve a aplicar los pasos de las fases de definición y de desarrollo, pero en el contexto del software ya existente.

Durante la fase de mantenimiento se encuentran cuatro tipos de cambios:

- *Corrección*: Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra defectos en el software. El mantenimiento correctivo modifica el software para corregir los defectos.
- *Adaptación*: Con el paso del tiempo es probable que cambie el entorno original (por ejemplo, el CPU, el sistema operativo, las reglas de la empresa, las características externas del producto) para el que se desarrolló el software. El mantenimiento de adaptación produce

modificación en el software para acomodarlo a los cambios de su entorno externo.

- *Mejora:* Conforme se utilice el software, el cliente/usuario puede descubrir funciones adicionales que van a producir beneficios. El mantenimiento perfecto lleva al software más allá de sus requisitos funcionales originales.
- *Prevención:* El software de computadora se deteriora debido al cambio, y por esto el mantenimiento preventivo también llamado ingeniería del software, se debe conducir para permitir que el software sirva para las necesidades de los usuarios finales. En esencia el mantenimiento preventivo hace cambios en programas de computadora a fin de que se puedan corregir, adaptar y mejorar más fácilmente. Hoy una fábrica de software que envejece está forzando a muchas compañías a que persigan estrategias de reingeniería de software. En un sentido global, la reingeniería de software se considera a menudo como una parte de la reingeniería de procesos comerciales.

Las fases y los pasos relacionados descritos en nuestra visión genérica de la ingeniería de software se complementan con un número de actividades protectoras. Entre las actividades típicas de esta categoría se incluyen:

- Seguimiento y control del proyecto de software
- Revisiones técnicas formales
- Garantía de calidad del software
- Gestión de configuración del software
- Preparación y producción de documentos
- Gestión de reutilización
- Mediciones
- Gestión de riesgos

## EL PROCESO DEL SOFTWARE

Un proceso de software se puede caracterizar como se muestra en la figura 1.4. Se establece un marco común del proceso, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia de su tamaño o complejidad. Un conjunto de tareas es una colección de tareas de ingeniería de software, objetivos de proyecto, posibles entregas y productos de trabajo del software, y puntos de garantía de calidad que permiten que las actividades del marco de trabajo se adapten a las características del proyecto del software y a los requisitos de equipo del proyecto.

Finalmente, las actividades de protección, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

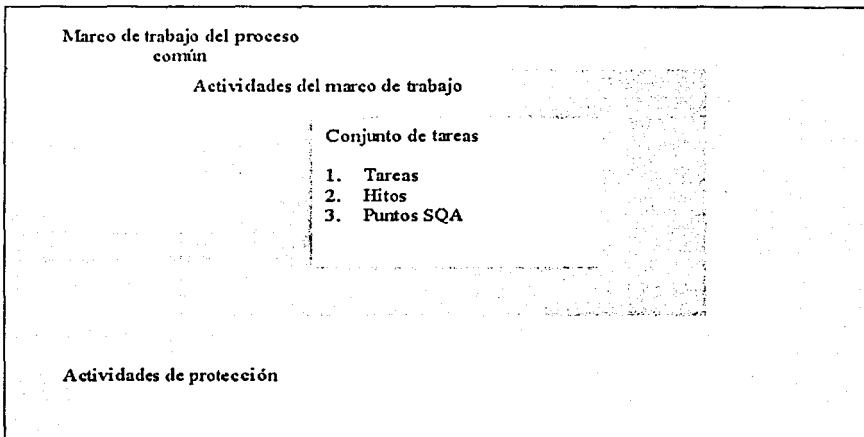


Figura 1.4 El proceso de software

En los últimos años se ha hecho mucho hincapié en la "madurez de los procesos". El Software Engineering Institute (SEI) ha desarrollado un concepto completo que se basa en un conjunto de funciones de ingeniería de software que deberían estar presentes conforme las organizaciones alcanzan diferentes niveles de madurez del proceso. Para determinar el estado actual de madurez del proceso, el SEI

utiliza un cuestionario de evaluación y un esquema de cinco grados. El esquema determina la conformidad con un modelo de capacidad de madurez que define las actividades clave que se requieren en los diferentes niveles de madurez del proceso.

## MODELOS DE PROCESO DE SOFTWARE

Para resolver los problemas reales de una industria, un ingeniero del software o un equipo de ingenieros deben incorporar una estrategia de desarrollo. Esta estrategia a menudo se denomina modelo de proceso o paradigma de ingeniería de software. Se selecciona un modelo de proceso para la ingeniería de software según la naturaleza del proyecto, de la aplicación, los métodos, las herramientas a utilizarse, los controles y las entregas que se requieren. Todo el desarrollo del software se puede caracterizar como un ciclo de resolución de problemas en el que se encuentran cuatro etapas distintas: entorno actual, definición de problemas, desarrollo técnico e integración de soluciones.

Entorno actual, representa el estado actual de sucesos la definición de problemas identifica el problema específico a resolver, el desarrollo técnico resuelve el problema a través de la aplicación de alguna tecnología y la integración de soluciones ofrece los resultados (por ejemplo, documentos, programas, datos, nueva función comercial, producto nuevo), a los que solicitan la solución en primer lugar. Las fases y los pasos genéricos de ingeniería de software definido, se describe fácilmente en la figura 1.5.

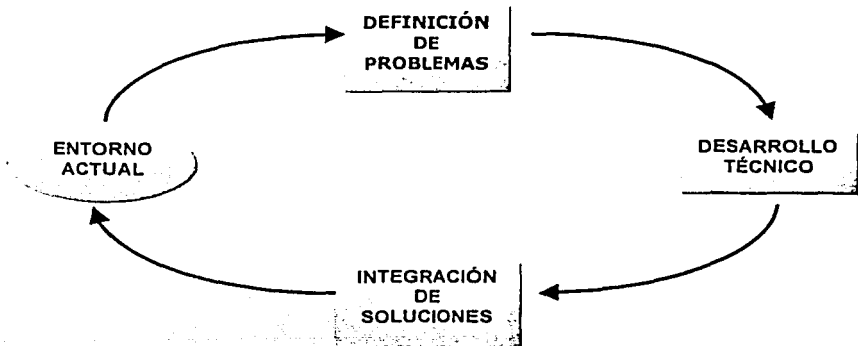


Figura 1.5 Las fases de un ciclo de resolución d problemas

El ciclo de resolución de problemas descrito anteriormente se aplica al trabajo de ingeniería de software en muchos niveles diferentes de resolución. Se puede utilizar en el macro nivel cuando se tiene en consideración la aplicación entera, en un nivel medio cuando se están considerando los componentes del programa e incluso en la línea del nivel del código.

En la figura 1.5, cada etapa del ciclo de resolución de problemas es un ciclo idéntico de solución de problemas, el cual contiene otro ciclo. Todas las etapas del entorno actual, definición de problemas, desarrollo técnico e integración de soluciones coexisten simultáneamente en algún nivel del detalle.

En las secciones siguientes, se tratan diferentes modelos de procesos para la ingeniería de software. Cada uno representa un intento de ordenar una actividad inherente caótica. Es importante recordar que cada uno de los modelos se ha caracterizado de forma que ayuden al control y a la coordinación de un proyecto de software real. Y a pesar de eso, en el fondo todos los modelos exhiben características del modelo del caos.

### El modelo lineal secuencial

La figura 1.6 ilustra el modelo lineal secuencial para la ingeniería de software. Llamado algunas veces "ciclo de vida básico" o "modelo en cascada", el modelo lineal secuencial sugiere un enfoque sistemático, secuencial del desarrollo del software que comienza en un nivel de sistemas y progresa, con el análisis, diseño, codificación, pruebas y mantenimiento. Modelado según el ciclo de ingeniería convencional, el modelo lineal secuencial acompaña a las actividades siguientes:



Figura 1.6 Modelo Lineal Secuencial

### Análisis

El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Para comprender la naturaleza del (los) programa (s) a construirse, el ingeniero de software debe de comprender el dominio de información del software, así como la función requerida, comportamiento, rendimiento e interconexión. El cliente documenta y repasa los requisitos del sistema y del software.

## Diseño

El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de un programa: estructura de datos, arquitectura del software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso de diseño traduce requisitos en una representación del software que se pueda evaluar por calidad antes de que comience la generación del código. Al igual que los requisitos, el diseño se documenta y se hace parte de la configuración del software.

## Generación de código

El diseño se debe de traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente.

## Pruebas

Una vez que se ha generado un código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores y el sentirse seguro de que la entrada definida produzca resultados reales de acuerdo con los resultados requeridos.

## Mantenimiento

El software indudablemente sufrirá cambios después de ser entregado al cliente. Se producirán cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios de su entorno externo, o porque el cliente requiere mejoras funcionales o de rendimiento. El mantenimiento vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo.

El modelo lineal secuencial es el paradigma más antiguo y más extensamente utilizado en la ingeniería de software. Entre los problemas que se encuentran a menudo en el modelo lineal secuencial se incluyen:

- 1.- Los proyectos reales raras veces siguen el modelo secuencial que propone el modelo. Aunque el modelo lineal puede acoplar interacción, lo hace indirectamente. Como resultado, los cambios pueden causar confusión cuando el equipo del proyecto comienza.
2. - A menudo es difícil que el cliente exponga explícitamente todos los requisitos. El modelo lineal secuencial lo requiere y tiene dificultades a la hora de acomodar la incertidumbre natural al comienzo de muchos proyectos.

3.- El cliente debe ser paciente. Una versión del trabajo del programa no estará disponible hasta que el proyecto esté muy avanzado. Un grave error puede ser desastroso, si no se detecta hasta que se revisa el programa.

4.- Los responsables del desarrollo del software siempre se retrasan innecesariamente. Esto es, se producen estados de bloqueo, en el que algunos miembros del equipo tienen que esperar a otros miembros del equipo para completar tareas dependientes.

#### El modelo de construcción de prototipos

Un cliente a menudo define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida. En otros casos, el responsable del desarrollo del software puede no estar seguro de la eficacia de un algoritmo, de la capacidad de adaptación de un sistema operativo, o de la forma en que debería tomarse la interacción hombre-máquina. En estas circunstancias, un paradigma de construcción de prototipos puede ofrecer el mejor enfoque.

El paradigma de construcción de prototipos de la figura 1.7, comienza con la recolección de requisitos. El desarrollador y el cliente se encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente (por ejemplo formatos de entrada y de salida). El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y lo utiliza para refinar los requisitos del software a desarrollar. La interacción ocurre cuando el prototipo satisface las necesidades del cliente, a la vez que permite que el desarrollador comprenda mejor lo que se necesita hacer. Lo mejor sería que el prototipo sirviera como un mecanismo para identificar los requisitos del software.

El prototipo puede servir como un primer sistema. Sin embargo, la construcción de prototipos también puede ser problemática por las siguientes razones:

El cliente ve lo que parece ser una versión de trabajo del software, sin saber que con la prisa de hacer que funcione no se ha tenido en cuenta la calidad del software global o la facilidad a largo plazo. Cuando se informa que el producto ha de construirse otra vez para que puedan mantener los niveles altos de calidad, el cliente no lo entiende y pide que se apliquen solo pequeños ajustes para que se pueda hacer del prototipo un producto final.

El desarrollador a menudo hace compromisos de implementación para hacer que el prototipo funcione rápidamente. Se puede utilizar un sistema operativo o lenguaje de programación inadecuado simplemente por que está disponible o es conocido por el desarrollador.



Aún con estos problemas, la clave es definir las reglas del juego al comienzo, es decir, el cliente y el desarrollador se deben de poner de acuerdo en que el prototipo se construya para servir como un mecanismo de definición de requisitos.

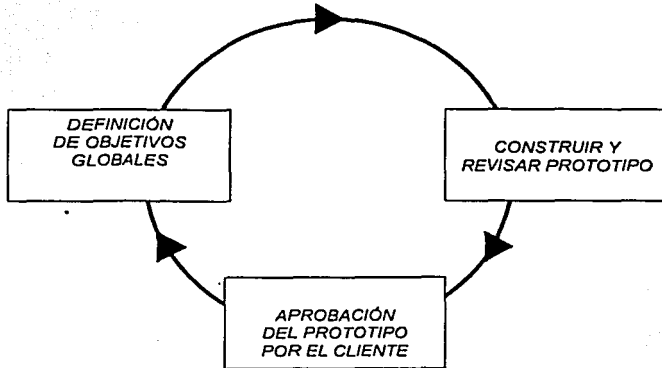


Figura 1.7 Modelo de construcción de prototipos

### El modelo de desarrollo rápido de aplicaciones (DRA)

El modelo DRA, es un modelo de proceso del desarrollo del software lineal secuencial, que enfatiza un ciclo de desarrollo extremadamente corto. El modelo DRA es una adaptación a "alta velocidad" del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. Si se comprenden bien los requisitos del proyecto, el proceso DRA permite al equipo de desarrollo crear un sistema completamente funcional, dentro de períodos cortos de tiempo (60 a 90 días).

El enfoque DRA comprende las siguientes fases:

#### Modelo de gestión

El flujo de la información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿qué información conduce al proceso de gestión?, ¿qué información se genera?, ¿quién la genera? ¿a dónde va la información? y ¿quién la procesa?

## Modelado de datos

El flujo de la información definida como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar a la empresa. Se definen las características, llamadas atributos, de cada uno de los objetos y las relaciones entre los objetos.

## Modelo del proceso

Los objetos de datos definidos en la fase del modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir o recuperar un objeto de datos.

## Generación de aplicaciones

El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de cuarta generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.

## Pruebas y entrega

Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben de probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

El modelo del DRA se ilustra en la figura 1.8. Si una aplicación de gestión puede modularse de forma que permita completarse cada una de las funciones principales en menos de tres meses, es un candidato para el DRA. Cada una de las funciones puede ser afrontada por un equipo DRA diferente y ser integradas en un solo conjunto.

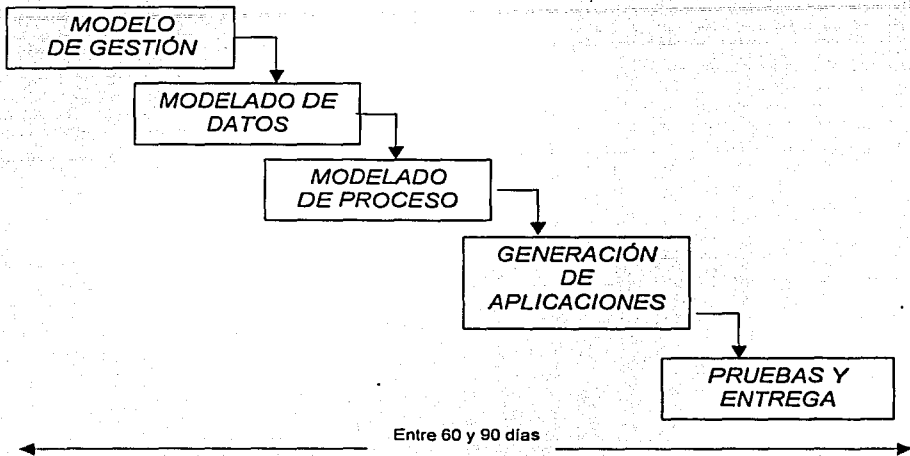


Figura 1.8 Modelo DRA

Al igual que todos los modelos de proceso, el enfoque DRA tiene inconvenientes:

Para proyectos grandes aunque por escalas, el DRA requiere recursos humanos suficientes como para crear el número correcto de equipos DRA.

DRA requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado. Si no hay un compromiso, por ninguna de las dos partes, constituyentes, los proyectos DRA fracasarán.

No todos los tipos de aplicaciones son apropiados para DRA. Si un sistema no se puede modularizarse adecuadamente, la construcción de los componentes necesarios para DRA será problemático. Si está en juego el alto rendimiento y se va a conseguir el rendimiento convirtiendo interfaces en componentes de sistemas, el enfoque DRA, puede que no funcione. DRA no es adecuado cuando los riesgos técnicos son altos, esto ocurre cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el nuevo software requiere un alto grado de interoperabilidad con programas de computadora ya existentes. DRA enfatiza el desarrollo de componentes de programas reutilizables.

## Modelos de procesos evolutivos de software

Se reconoce que el software al igual de todos los sistemas complejos, evoluciona con el tiempo. Los requisitos de gestión y de productos a menudo cambian conforme a que el desarrollo proceda haciendo que el camino que lleva el producto final no sea real.

Los ingenieros de software necesitan un modelo de proceso que haya diseñado explícitamente para acomodarse a un producto que evolucione con el tiempo.

El modelo lineal secuencial se diseña para el desarrollo en línea recta. En esencia, este enfoque en cascada, asume se va entregar un sistema completo una vez que la secuencia lineal se haya finalizado. El modelo de construcción de prototipos, se diseña para ayudar al cliente o al que desarrolla a comprender los requisitos. En general, no se diseña para entregar un sistema en producción. En ninguno de los paradigmas de ingeniería de software se tiene en cuenta la naturaleza evolutiva del software.

Los modelos evolutivos son iterativos. Se caracterizan por la forma en que permiten a los ingenieros de software desarrollar versiones cada vez más completas de software.

### El modelo incremental

El modelo incremental combina elementos del modelo lineal secuencial (aplicados repetitivamente) con la filosofía interactiva de construcción de prototipos. Como muestra la figura 1.9, el modelo incremental aplica a secuencias lineales de la misma forma que avanza el tiempo en el calendario.

Cada secuencia lineal produce un incremento del software. Por ejemplo el software de tratamiento de textos desarrollados con el paradigma incremental podría extraer funciones de gestión de archivos básicos y de producción de documentos en el primer incremento, funciones de edición más sofisticadas y de producción de documentos en el segundo incremento, corrección ortográfica y gramatical en el tercero; y una función avanzada de esquema de página en el cuarto. Se debería tener en cuenta que el flujo del proceso de cualquier incremento puede incorporar el paradigma de construcción de prototipos.

Cuando se utiliza un modelo incremental, el primer incremento, a menudo es un producto esencial (núcleo). Es decir, se afrontan requisitos básicos, pero muchas funciones suplementarias (algunas conocidas, otras no) quedan sin extraer. El cliente utiliza el producto central (o sufre la revisión detallada). Como un resultado de utilización y/o evaluación, se desarrolla un plan para el incremento siguiente. El plan afronta la modificación del producto central a fin de cumplir mejor las necesidades del cliente y la entrega de funciones y características adicionales.

Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo.

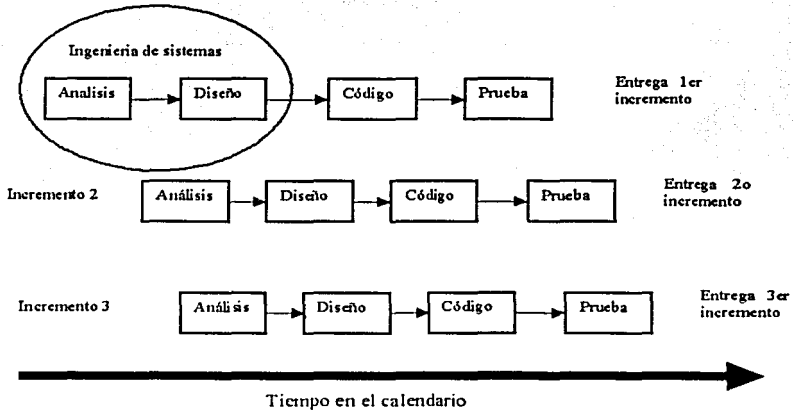


Figura 1.9 Modelo Incremental

El modelo del proceso incremental, es interactivo por naturaleza. Pero a diferencia de la construcción de prototipos, el modelo incremental se centra en la entrega del producto operacional con cada incremento. Los primeros incrementos son versiones del producto final, pero proporcionan la capacidad que sirve al usuario y también proporciona una plataforma para la evolución por parte del usuario.

El desarrollo incremental es particularmente útil cuando la dotación de personal no está disponible para una implementación completa en cuanto a la fecha límite de gestión que se haya establecido para el proyecto. Los primeros incrementos se pueden implementar con menos personas. Si el producto central es bien recibido, se puede añadir más personal (si se requiere) para implementar el incremento siguiente.

## El modelo en espiral

El modelo en espiral, propuesto originalmente por Boehm, ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como la de la creación de prototipos añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgos que falta en esos paradigmas.

La figura 1.10 define las cuatro actividades principales del modelo en espiral:

- *Determinación de objetivos, alternativas y restricciones:* Las tareas requeridas para determinar la problemática.
- *Análisis de riesgos:* Las tareas requeridas para evaluar riesgos técnicos y entrega de prototipos.
- *Desarrollo y pruebas:* Las tareas requeridas para construir una o más representaciones de la aplicación.
- *Planificación:* Las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto.

La primera espiral comienza con identificar los requerimientos incluyendo presupuestos, restricciones y alternativas para diseñar y desarrollar el sistema, el siguiente paso es evaluar riesgos y alternativas de prototipos antes de un documento de conceptos de operaciones, que describa a alto nivel como el sistema debe funcionar. Así el concepto de operaciones es el producto de la primera iteración, y los requerimientos son el principal producto de la segunda iteración, en la tercera iteración el desarrollador del sistema produce el diseño y en la cuarta se habilitan las pruebas.

Cada terminación de espiral se utiliza para desarrollar un prototipo y sucesivamente versiones más sofisticadas del software.

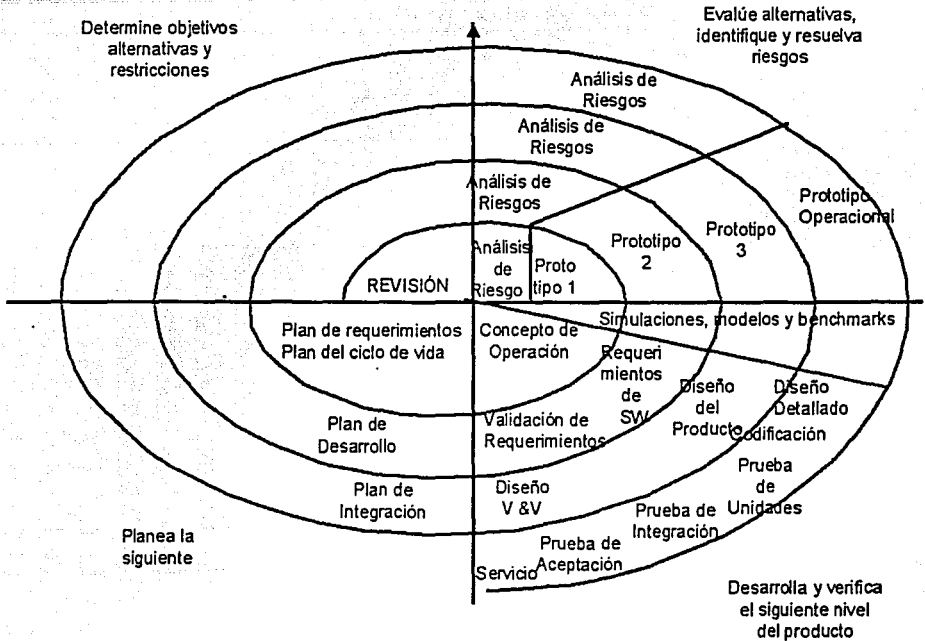


Figura 1.10 Modelo en Espiral.

Cada paso de la región de planificación produce ajustes en el plan de proyecto. El costo y la planificación se ajustan según la reacción ante la evaluación del cliente. El gestor del proyecto ajusta el número no planificado de iteraciones requeridas para completar el software.

A diferencia del modelo de proceso clásico que termina cuando se entrega el software, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.

El modelo en espiral es un enfoque realista del desarrollo de sistemas y de software en gran escala. Como el software evoluciona, a medida que progresa el proceso, el desarrollador y el cliente lo comprenden y reaccionan mejor ante el riesgo en cada uno de los niveles evolutivos. El modelo en espiral utiliza la

construcción de prototipos como mecanismo de reducción de riesgos, pero lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.

Pero al igual que otros paradigmas., el modelo en espiral no es la panacea. Puede resultar difícil convencer a grandes clientes de que el enfoque evolutivo es controlable. Requiere una considerable habilidad para la evaluación del riesgo y se podrá contar con el éxito. Si un riesgo importante no es descubierto y gestionado, surgirán problemas.

#### El modelo de ensamblaje de componentes

Las tecnologías de objetos proporcionan el marco de trabajo técnico para un modelo de proceso basado en componentes para la ingeniería de software. El paradigma de orientación a objetos enfatiza la creación de clases que encapsulan tanto a los datos como los algoritmos que se utilizan para manejar los datos. Si se diseñan y se implementan adecuadamente, las clases orientadas a objetos son reutilizables por las diferentes aplicaciones y arquitecturas basados en computadora.

El modelo de ensamblaje de componentes incorpora muchas de las características del modelo en espiral. Es evolutivo por naturaleza y exige un enfoque interactivo para la creación del software. Sin embargo, el modelo ensamblador de componentes configura aplicaciones desde componentes preparados de software (algunas veces llamadas clases).

La actividad de la ingeniería comienza con la identificación de clases candidatas. Los datos y algoritmos correspondientes se empaquetan en una clase. Las clases (llamadas componentes) creadas en los proyectos de ingeniería de software anteriores se almacenan en una biblioteca de clases o depósito. Una vez identificadas las clases candidatas, la biblioteca de clases se examina para determinar si estas clases ya existen. En caso de que así fuera, se extraen de la biblioteca y se vuelven a utilizar. Si una clase candidata no reside en la biblioteca, se aplican los métodos orientados a objetos.

Se compone así la primera interacción de la aplicación a construirse. El flujo del proceso se vuelve a la espiral y volverá a introducir por último la iteración ensambladora de componentes a través de la actividad de ingeniería. El modelo reensamblador de componentes, lleva a la reutilización del software y la reutilización proporciona beneficios a los ingenieros de software.



## Modelo de desarrollo concurrente

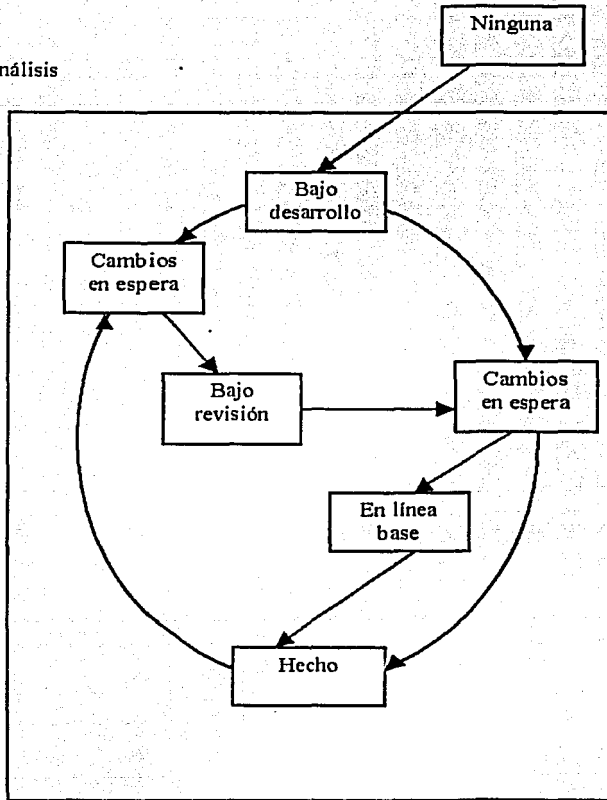
El modelo de desarrollo concurrente se puede representar en forma de esquema como una serie de actividades técnicas importantes, tareas y estados asociados a ellas. Por ejemplo, la actividad de ingeniería definida para el modelo en espiral, se lleva a cabo invocando las tareas siguientes: modelado de construcción de prototipos y/o análisis, especificación de requisitos y diseño. La figura 1.11 proporciona una representación esquemática de una actividad dentro del modelo de proceso concurrente.

El análisis se puede encontrar en uno de los estados, destacados anteriormente en cualquier momento dado. De forma similar, otras actividades, (por ejemplo diseño o comunicación con el cliente) se puede representar de una forma análoga. Todas las actividades existen concurrentemente, pero residen en estados diferentes. Por ejemplo, al principio del proyecto, la actividad de comunicación con el cliente (no mostrada en la figura) ha finalizado su primera interacción y existe en el estado de cambios en espera. La actividad de análisis mientras que comenzaba la comunicación inicial con el cliente ahora hace una transición al estado bajo desarrollo.

Sin embargo si el cliente indica que se deben de hacer cambios en requisitos, la actividad análisis cambia del estado bajo desarrollo al estado cambios en espera. El modelo de proceso concurrente define una serie de acontecimientos que disparan transiciones de estado a estado para cada una de las actividades de la ingeniería de software

El modelo del desarrollo concurrente se utiliza con frecuencia como el paradigma de desarrollo de aplicaciones cliente/servidor. Un modelo cliente servidor, está compuesto de un conjunto de componentes funcionales, cuando se aplica a cliente /servidor, el modelo de proceso concurrente define actividades en dos dimensiones: una dimensión de sistemas y una dimensión de componentes.

Actividad de Análisis



Representa un estado de una actividad en ingeniería de software

Figura 1.11 Un elemento del modelo de desarrollo concurrente

Los aspectos de nivel de sistemas se afrontan mediante tres actividades:

- Diseño
- Ensamblaje
- Uso

La dimensión de componentes se afronta con dos actividades:

- Diseño
- Realización

La concurrencia se logra de dos formas:

Las actividades de sistemas y de componentes ocurren simultáneamente y pueden modelarse con el enfoque orientado a objetos descrito anteriormente.

Una aplicación cliente/servidor típica se implementa con muchos componentes, cada uno de los cuales se puede diseñar y realizar concurrentemente.

En realidad el modelo de desarrollo concurrente es aplicable a todo tipo de desarrollo de software y proporciona una imagen exacta del estado actual de un proyecto. En vez de confinar todas las actividades de ingeniería de software a una secuencia de sucesos, define una red de actividades. Todas las actividades de la red existen simultáneamente con otras. Los sucesos generados dentro de una actividad dada o en algún otro lugar en la red de actividad inician las transiciones entre los estados de una actividad.

Modelo de métodos formales

El modelo de métodos formales acompaña a un conjunto de actividades que conducen a la especificación matemática del software de computadora. Los métodos formales permiten que un ingeniero de software, especifique desarrolle y verifique un sistema basado en computadora aplicando una notación rigurosa y matemática. Algunas organizaciones de desarrollo de software actualmente aplican una variación de este enfoque, llamado ingeniería de software de sala amplia. Cuando se utilizan métodos formales durante el desarrollo, proporcionan un mecanismo para eliminar muchos de los problemas que son difíciles de superar con paradigmas de la ingeniería de software. La ambigüedad, lo incompleto y la inconsistencia se descubren y se corrigen fácilmente, no mediante una revisión a propósito para el caso, sino mediante la aplicación del análisis matemático.

Aunque todavía no hay un enfoque establecido, los modelos de métodos formales ofrecen la promesa de un software libre de defectos. Sin embargo existe una gran preocupación sobre su aplicabilidad en su entorno de gestión:

El desarrollo de modelos formales, actualmente es bastante caro y lleva mucho tiempo. Se requiere un estudio caro porque pocos responsables del desarrollo de software tienen los antecedentes necesarios para aplicar métodos formales. Es difícil realizar los modelos como un mecanismo de comunicación con clientes que no tienen muchos conocimientos técnicos.

No obstante es posible que gane adeptos, sobre todo entre los desarrolladores de software que deben de construir software de mucha seguridad (desarrolladores de aviónica y dispositivos médicos).

#### Técnicas de cuarta generación

El término Técnicas de cuarta generación (TG4) abarca un amplio espectro de herramientas de software que tienen algo en común: todas facilitan al ingeniero de software, la especificación de algunas características del software de alto nivel. Luego, la herramienta genera automáticamente el código fuente basándose en la especificación del técnico.

El paradigma TG4 para la ingeniería de software se orienta hacia la posibilidad de especificar el software usando formas de lenguaje especializado o notaciones gráficas que describen el problema a resolver en términos que los entienda el cliente. Actualmente, el paradigma TG4 debe incluir todas o algunas de las siguientes herramientas:

- Lenguajes no procedimentales de consulta de bases de datos
- Generación de informes
- Manejo de datos
- Interacción
- Definición de pantallas
- Generación de códigos
- Capacidades gráficas de alto nivel
- Capacidades de hoja de cálculo

Inicialmente muchas de las herramientas estaban disponibles solo para ámbitos de aplicación muy específicos, pero actualmente los entornos TG4 se han extendido a todas las categorías de aplicaciones de software.

Al igual que otros paradigmas TG4 comienza con el paso de reunión de requisitos. Idealmente el cliente debe de describir los requisitos, que son posteriormente traducidos directamente a un prototipo operativo. Sin embargo en la práctica, no se puede hacer eso. El cliente puede que no esté seguro de lo que necesita; puede ser ambiguo en la especificación de hechos que le son conocidos, y puede que no sea capaz, o puede que no esté dispuesto a especificar la información en la que pueda utilizar una herramienta TG4. Por esta razón el diálogo cliente-desarrollador descrito por los otros paradigmas siguen siendo una parte esencial del enfoque TG4.

Para aplicaciones pequeñas se puede ir directamente desde el paso de recolección de requisitos al paso de implementación, usando un lenguaje de cuarta generación no procedimental (LG4). Sin embargo, es necesario un mayor esfuerzo para desarrollar una estrategia de diseño para el sistema, incluso si se utiliza un LG4. El uso de TG4 sin diseño (causará las mismas dificultades, poca calidad, mantenimiento pobre, mala aceptación por el cliente) que se encuentran cuando se desarrolla software mediante los enfoques convencionales.

La implementación mediante LG4 permite, al que desarrolla el software concentrarse en la representación de los resultados deseados, que es lo que se traduce automáticamente en un código fuente que produce dichos resultados. Obviamente, debe de existir una estructura de datos con información relevante y a la que el LG4 pueda acceder rápidamente.

Para transformar una implementación TG4 en un producto, el que lo desarrolla debe de dirigir una prueba completa, desarrollar con sentido una documentación y ejecutar el resto de las actividades de implementación requeridas en los otros paradigmas. Además, el software desarrollado con TG4 debe de ser construido de modo que facilite la realización del mantenimiento de una forma eficiente.

El modelo TG4 tiene ventajas e inconvenientes. Los defensores aducen reducciones drásticas en el tiempo de desarrollo del software y una mejora significativa en la productividad de la gente que construye el software. Los detractores aducen que las herramientas actuales de TG4 no son más fáciles de utilizar que los lenguajes de programación, que el código fuente producido por tales herramientas es "ineficiente" y que el mantenimiento de grandes sistemas de software desarrollados mediante TG4, es cuestionable.

El uso de TG4 ha crecido considerablemente en los últimos años y ahora es un enfoque viable para muchas de las diferentes áreas de aplicación. Junto con las herramientas de ingeniería de software asistida por computadora (CASE) y los

generadores de código, TG4 ofrece una solución fiable a muchos problemas de software.

Resumiendo, las técnicas de TG4 ya se han convertido en una parte importante del desarrollo de software. Cuando se combinan con enfoques de ensamblaje de componentes, el paradigma TG4 se puede convertir en el enfoque dominante hacia el desarrollo del software.

## TECNOLOGÍA DE PROCESOS

Los modelos de procesos tratados en las secciones anteriores se deben de adaptar para utilizarse por el equipo del proyecto de software. Para conseguirlo se han desarrollado herramientas de tecnología de procesos para ayudar a las organizaciones de software a analizar los procesos actuales, organizar tareas de trabajo, controlar y supervisar el progreso y gestionar la calidad técnica. Las herramientas de tecnología de procesos permiten que una organización de software construya un modelo automatizado de la estructura común del proceso, conjunto de tareas y actividades de protección ya tratadas anteriormente. El modelo presentado normalmente como una red se puede analizar para determinar el flujo de trabajo típico y para examinar estructuras alternativas de procesos que pudieran llevar a un tiempo de coste de desarrollo reducidos.

Una vez que se ha creado un proceso aceptable, se pueden utilizar otras herramientas de tecnología de procesos para asignar, supervisar e incluso controlar todas las tareas de ingeniería de software definidas como parte del modelo del proceso.

La ingeniería de software es una disciplina que integra métodos, procesos y herramientas para el desarrollo del software de computadora. Se han propuesto paradigmas diferentes, cada uno exhibiendo ventajas e inconvenientes pero todos tienen una serie de fases genéricas en común.

## DIAGRAMA DE FLUJO DE DATOS (DFD)

Para lograr un desarrollo y un diseño estructurado de sistemas la Ingeniería de Software se apoya en el DFD que es una herramienta de modelado que muestra un sistema como una red de procesos conectados a través de flujos de datos, haciendo una transformación de los datos de entrada para producir los datos de salida. Entre los precursores del DFD se destacan Stevens Myers, Constantine y Yourdon.

Al utilizar el DFD es posible:

- Comprender la relación existente entre los procesos del sistema.
- Esquematzación a los usuarios del sistema a través de los diagramas.

Para la elaboración eficaz del DFD se debe contemplar las siguientes características del mismo:

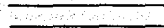
- Muestra el contexto del sistema.
- Muestra lo que hace el sistema.
- Presenta la interfaz entre sus componentes.

Los componentes del DFD usando la nomenclatura de Yourdon:

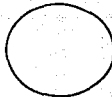
FLUJO DE DATOS



ARCHIVO/DEPOSITO DE INFORMACIÓN



PROCESO



ENTIDAD EXTERNA/FUENTE DE INFORMACIÓN/DESTINO



En la figura 1.12 se muestra la relación del DFD con el modelo de diagrama Entidad/Relación y con el diccionario de datos como se muestra:

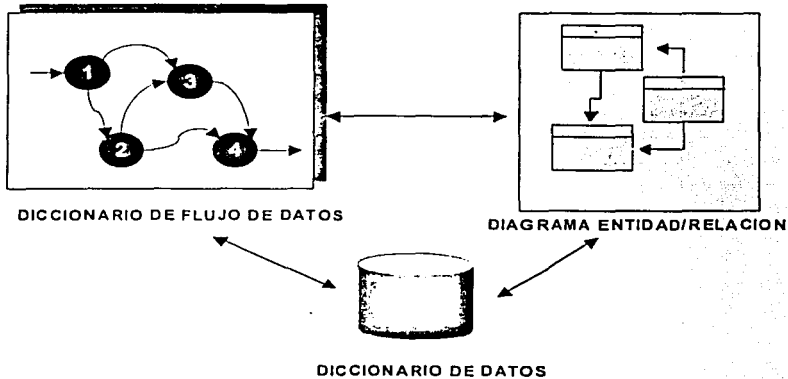


Figura 1.12 Relación DFD- E/R -DD

### DICCIONARIO DE DATOS (DD)

El diccionario de datos es una referencia útil para localizar nombres y atributos de los datos utilizados en el sistema, este deberá actualizarse cada vez que se hagan cambios en el sistema.

En la tabla 1.1 se muestra la notación de representación de la sintaxis del sistema:

SÍMBOLO	REPRESENTACION
=	Está compuesto de
+	Para unir campos elementales
( )	Campos opcionales
[ ]	Selección de una entre varias
{ }	Iteraciones
**	Comentarios
@	Identificación o clave de un almacenamiento
	Separación de un almacenamiento

Tabla 1.1 Notación de sintaxis del sistema



---

Los registros del diccionario de datos deben contener información referente a:

*El nombre y alias del dato:* Es necesario incluir la manera más común de denominar el dato en los programas del sistema, además de su alias. El registro de estos datos facilita la comunicación y la referencia cruzada entre esos programas.

*La descripción del dato:* Debe incluir un texto breve, pero informativo del dato.

*Los flujos de datos que se relacionan con el término:* Establece el origen y el destino de la transferencia de datos.

*La longitud del campo:* Establece la longitud permitida para el acceso de un dato.

*Cualquier otra información pertinente:* Debe agregarse al diccionario de datos cualquier información útil extra.

## 1.2 CONCEPTOS GENERALES DE BASES DE DATOS RELACIONALES

Las bases de datos surgieron durante los años 60 cuando se conoció un primer desarrollo por Bachman que constituyó el fundamento para el modelo de datos de red.

En 1965 surgió la primera generación de sistemas de administración de base de datos (DBMS) que estaba caracterizada por la separación de la descripción de los datos de los programas de aplicación y la llegada de lenguajes de consulta.

El término base de datos se define como:

Una base de datos es un conjunto de datos relacionados entre sí. Por datos se entiende que son hechos conocidos que pueden registrarse y que tienen un significado implícito. Una base de datos tiene las siguientes propiedades:

Una base de datos representa algún aspecto del mundo real. Las modificaciones del mundo real se reflejan en la base de datos.

Una base de datos es un conjunto de datos lógicamente coherente, con cierto significado inherente.

Toda base de datos se diseña, construye y prueba con datos para un propósito específico.

### 1.2.1 ARQUITECTURA DE LA BASE DE DATOS

La arquitectura de una base de datos tiene tres niveles de abstracción: interno, conceptual y externo.

#### NIVEL INTERNO.

El modelo interno es la representación del nivel inferior. Mapeo a la base lógica hacia el almacenamiento físico y establece trayectorias de datos para acceso a la base de datos.

## NIVEL CONCEPTUAL

En este nivel el administrador de datos (DA) define el modelo conceptual, éste representa la visión organizacional de la base que se obtiene al integrar los requerimientos de todos los usuarios de la empresa. Un esquema conceptual consta de las siguientes definiciones:

*Definición de los datos:* Planeación del tipo de datos, longitud de campos, atributos y entidades de acuerdo con las necesidades reales de la empresa.

*Relaciones entre datos:* Definir las relaciones de la información.

Un esquema conceptual se formula sin importar el almacenamiento físico, reflejando información sobre las actividades de la empresa.

El DA debe aprender, comprender y conceptualizar el mundo real transformándolo en definiciones con procesos de análisis y síntesis. Una vez definido el modelo conceptual el DA lo transforma en una descripción de datos, atributos, entidades y relaciones.

## NIVEL EXTERNO

Corresponde a la manera en como los usuarios visualizan la base de datos.

### 1.2.2 ESQUEMA CONCEPTUAL DE DATOS

Se debe empezar por acotar que parte del mundo real nos interesa representar. El DA debe aprender, comprender y conceptualizar el mundo transformándolo, a este proceso se le conoce como modelo conceptual. Para construir un buen modelo, el analista debe utilizar una gran dosis de procesos mentales de abstracción, análisis y síntesis. Necesitara además la colaboración de personal directivo.

Una vez definido el modelo conceptual, el analista lo transforma en una descripción de datos, atributos y tablas, incluyendo las posibles relaciones entre estos elementos y su significado. A esta descripción se le conoce como esquema conceptual de datos.

Al proceso de transformar el modelo conceptual a un esquema conceptual se le conoce como diseño lógico de datos.

Una vez definido el esquema conceptual, hay que traducirlo a estructuras que se almacenen en dispositivos físicos controlados por un ordenador, a esta transformación se le denomina diseño físico de datos.

La figura 1.13 muestra el proceso de transformación de la interpretación del mundo real a un esquema conceptual

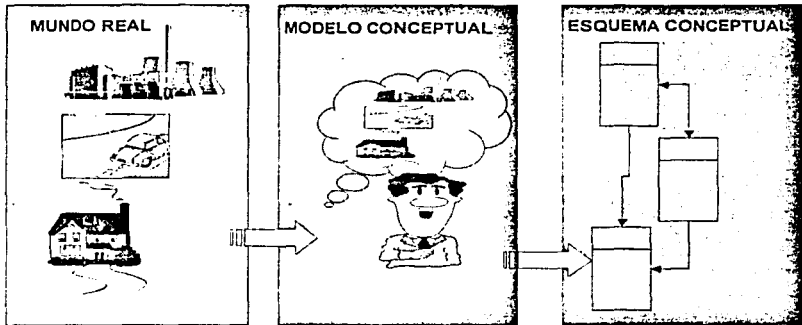


Figura 1.13 Proceso de transformación de la interpretación del mundo real.

Un sistema de administración de bases de datos es una herramienta que permite insertar, modificar y buscar eficientemente datos específicos entre un volumen masivo de información compartida por todos los usuarios

El DBMS crea una base de datos como un repositorio estructurado de información en archivos interrelacionados compartidos, para satisfacer los requerimientos de los usuarios de una manera selectiva y en tiempo oportuno.

Los tres principales componentes de un DBMS son:

#### Hardware

Se compone de los volúmenes de almacenamiento secundario (discos, cintas) donde reside la base de datos, así como de la memoria principal que se utiliza principalmente como almacén de información temporal.

#### Software

Entre la base de datos y los usuarios del sistema existe un software como interfaz que interpreta y procesa las peticiones del usuario denominado sistema de

administración de base de datos DBMS. La interpretación de las solicitudes se hace por medio del lenguaje de definición de datos (LDD) y lenguaje de manipulación de datos (LMD).

El DBMS protege a los usuarios de la tediosa programación que tendrían que hacer para organizar la información.

### Usuarios

Se consideran en general tres tipos de usuarios:

- *Programador de aplicaciones* quien es el encargado de desarrollar programas para la aplicación.
- *Usuario final* que acceda a la base de datos para consultar o manipular los datos.
- *Administrador de la base de datos (DBA)* especialista en el DBMS responsable de planear la organización de la información además de mantenerla consistente y segura.

### 1.2.3 CLASIFICACIÓN DE LOS SISTEMAS DE BASES DE DATOS

El diseño de los lenguajes de manipulación de datos debe definirse de acuerdo con las estructuras y operadores que soporte el sistema. Es conveniente clasificar los sistemas de bases de datos de acuerdo con el enfoque que adoptan. Los tres enfoques conocidos son:

#### ENFOQUE DE RED

Los datos en el enfoque de red se representan por medio de "registros" y las relaciones entre los datos se representan con ligas, que pueden considerar apuntadores.

#### ENFOQUE JERÁRQUICO

Es similar al modelo de red en cuanto a que los datos y las relaciones entre los datos se representan por medio de registros y ligas, respectivamente. El modelo jerárquico difiere del de red en que los registros están organizados como conjuntos de árboles en vez de gráficas arbitrarias.

## ENFOQUE RELACIONAL

Este deriva directamente del concepto de relación de teoría de conjuntos del producto cartesiano.

Ejemplo:

$A = \{\text{tornillo, tuerca}\}$

$B = \{100, 20\}$

$AXB = \{(\text{tornillo}, 100), (\text{tornillo}, 20), (\text{tuerca}, 100), (\text{tuerca}, 20)\}$

En donde (tornillo, 100) es una relación.

De una forma más sencilla las relaciones se representan a través de tablas que se organizan en renglones y columnas:

Tornillo	100
Tornillo	20
Tuerca	100
Tuerca	20

Cada renglón se denomina registro y son atributos que caracterizan la entidad.

Cada columna se denomina campo y es información de un solo tipo para todas las instancias.

### 1.2.4 BASE DE DATOS RELACIONAL

Una base de datos relacional es un conjunto de relaciones cuyas ocurrencias son tuplas de estas relaciones.

A la herramienta de administración de base de datos relacionales se denomina RDBMS.

### OBJETIVOS DEL RDBMS

Independencia de datos.

Se refiere a la protección contra los programas de aplicación en el momento de ocurrir modificaciones y/o alteraciones en la organización física de la estructura lógica de la base de datos.

### Independencia física.

La independencia física se refiere a la protección contra las modificaciones de aplicación debido a cambios en la estructura del archivo, sin modificar los programas ni los accesos a los datos.

### Independencia lógica.

La independencia lógica se refiere a la protección de algunos programas de aplicación y permite aumentar claves y objetos sin modificar los programas.

### Inconsistencia de datos

El guardar datos redundantes, evidentemente es un desperdicio de espacio de almacenamiento, sin embargo el problema es más serio asociado con dicha redundancia es la inconsistencia de datos. Esto ocurre generalmente como resultado de actualizaciones incompletas de datos duplicados.

En un ambiente dinámico, las operaciones de inserción, eliminación y actualización se efectúan frecuentemente, es entonces imperativo que el sistema evite la inconsistencia de los mismos.

### Integridad de los datos

El término de integridad de datos se refiere a medidas de seguridad usadas para mantener correctos los datos en la base de datos. Tal vez la mejor manera de explicar la integridad de datos sea ejemplificando la forma en que ocurren los errores en la misma.

- Falla del equipo (hardware)
- Actualización incompleta.
- Defectos de software.
- Inserción de datos inválidos.
- Errores de usuarios.

Una falla en el sistema tal como: fallas de disco, memoria, CPU, entre otras, pueden destruir físicamente los datos. Otra causa de error dramática es una caída del sistema al momento de un proceso de escritura de datos, lo que puede resultar en una actualización incompleta causando una inconsistencia de datos. Los errores de software pueden deberse a errores de lógica (bugs), estos pueden existir tanto en el software o en programas de aplicación. Es responsabilidad del

programador controlar estos errores antes de implementarlos en un ambiente productivo.

Para asegurar la integridad de los datos no es suficiente la confiabilidad del hardware y software, sino también los datos correctos para controlar estos errores es necesario recurrir a la validación a la entrada de los datos.

## VALIDACIÓN DE DATOS E INTEGRIDAD REFERENCIAL

Validación del tipo de los datos.

Así como en los lenguajes de programación los elementos de los datos de una base de datos se pueden especificar como carácter, decimal fijo, punto flotante, número complejo etc. El contenido de cada elemento debe coincidir con sus características de su definición.

Validación del valor de los datos.

El contenido de un dato de entrada puede validarse para cierto rango de valores numéricos o un número finito de valores alfa-numéricos. Por ejemplo, el campo ESTACIÓN puede contener solo cuatro valores posibles: PRIMAVERA, VERANO, OTOÑO O INVIERNO.

Validación sobre los valores únicos.

Cada registro de la base de datos se puede representar únicamente por un valor clave que se utiliza para identificar de manera única a cada dato.

El RDBMS debe asegurar que el valor de la clave sea único y no nulo mientras se introduce la información a la base de datos.

Integridad referencial

La idea básica de la integridad referencial es asegurar que toda la información este correctamente relacionada, por ejemplo, un contrato sin tener un cliente y un vendedor asociados a este, de igual manera la integridad referencial implica que la eliminación de datos conlleve a la eliminación de las relaciones correspondientes.

Control de concurrencia

Dado que en un sistema de base de datos la información es compartida, es común que los usuarios traten de utilizar la misma información simultáneamente. Es trabajo del RDBMS emplear métodos de bloqueo para mantener la consistencia de la información.



## SEGURIDAD DE LOS DATOS

La seguridad de los datos está relacionada con la integridad de los mismos. La seguridad de los datos se refiere a la protección de la base de datos contra accesos o modificaciones no autorizados. En la base de datos la información es vulnerable a las intromisiones accidentales o malintencionadas. Un acceso ilegal puede tener como resultado la destrucción total o parcial de la base de datos. Sin un control de seguridad los usuarios no tendrían privacidad en sus datos confidenciales y el sistema no podría mantener la integridad de los mismos.

Por estas razones el RDBMS proporciona mecanismos que refuerzan la seguridad algunos de ellos, con privilegios de los usuarios, claves de acceso y esquemas.

Una de las tareas del administrador de base de datos (DBA) es mantener la seguridad de la base de datos, controlando los derechos, limitando los privilegios de cada usuario de lectura, escritura o incluso consulta sobre los datos a los que tiene permiso.

## MODELADO Y DISEÑO DE UNA BASE DE DATOS RELACIONAL

El principal objetivo del diseño a nivel conceptual o esquema conceptual, es capturar datos de los requerimientos del mundo real, simples y manejables para los usuarios finales. El usuario final es quien consulta y actualiza la base de datos a través de RDBMS y no estará muy involucrado con el diseño conceptual.

Un buen diseño debe producir un esquema conceptual que sea una imagen fiel y completa del modelo conceptual incluyendo algunas interrelaciones y condiciones semánticas, es decir aquellas que son consecuencia del significado de los datos.

## CICLO DE VIDA DE UNA BASE DE DATOS

### Análisis de requerimientos

Los requerimientos de la base de datos están determinados por la interpretación correcta de las políticas de la empresa, incluyendo las relaciones y restricciones con respecto a la integridad y seguridad de la información. Esto incluye definir el hardware y la plataforma para la implementación de la base de datos.

### Diseño lógico

El esquema global que muestra los datos y su relación, está desarrollado usando técnicas de modelado tal como el diagrama entidad/relación (E/R).

Modelando E/R: Los requerimientos de los datos son analizados y modelados usando el diagrama E/R.

Transformación del modelo E/R a relaciones en un modelo E/R

Normalización de las relaciones candidatas.

Distribución de datos

Fragmentación y asignación del diseño físico, esto es considerar configuración de red.

Diseño del esquema físico

En esta fase es para decidir el DBMS para definir el esquema del usuario final.

Implementación, monitoreo y modificación de la base de datos

Una vez que el diseño de la base de datos esta terminado entonces puede ser implementado. Ya que la base de datos este en operación debe monitorearse para mejorar su rendimiento.

## CONCEPTOS DEL DIAGRAMA E/R

El modelado de datos entidad/relación se basa en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones.

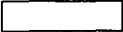



**Entidades:** Son principalmente datos de objetos de los cuales se recolecta información, usualmente denotan personas, lugares, cosas o eventos de interés.

**Relaciones:** Son acciones que realiza una entidad que describen el estado de las instancias así como la relación entre ellas

**Atributo:** Son propiedades que caracterizan a una entidad.

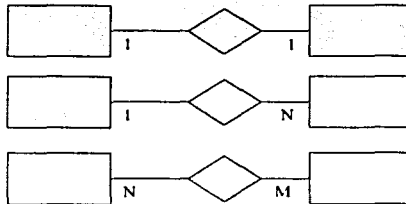
**Valor:** Firma un atributo de la ocurrencia de una entidad.

A continuación se muestra la representación de los conceptos mencionados anteriormente:

CONCEPTO	REPRESENTACIÓN
Entidad	
Relación	
Atributo	
Valor	

### Cardinalidad

Se refiere al número de elementos que están asociados entre entidades uno-a-uno, uno-a-muchos, o muchos-a-muchos



Una vez que se propone un diagrama Entidad/Relación se procede a el proceso de normalización.

### NORMALIZACIÓN

La normalización es un método a través del cual se puede verificar si el modelo propuesto es funcional y listo para ser implementado. Es un método basado en una serie de reglas.

Para explicar la normalización es necesario determinar la dependencia funcional.

El análisis de una relación de dependencia funcional entre los campos de una relación permite clasificar la relación en una de las cuatro formas normales. El concepto de dependencia funcional DF proviene de las matemáticas.

Se dice que Y es función de X:

$$Y = f(x)$$

Si el valor de Y está siempre determinado por el valor de X.

Aplicando el concepto de DF entre atributos A y B de una relación se define:

El atributo A es funcionalmente dependiente del atributo B si el valor de A está determinado por el valor de B:



Las relaciones normalizadas se agrupan en cuatro categorías llamadas formas normales. La meta final de la normalización es la agrupación de todos los atributos (o columnas) de una base de datos en relaciones adecuadas para que la base se pueda almacenar con el mínimo de datos redundantes.

Estas cuatro formas normales se enuncian a continuación.

#### Primera Forma Norma 1FN

Una relación esta en 1FN si todos los atributos de la relación contienen valores atómicos.

#### SEGUNDA FORMA NORMAL 2FN

Una relación está en 2FN si todos los atributos no llave dependen totalmente de la llave primaria.

#### TERCERA FORMA NORMAL 3FN

Una relación esta en 3FN si los atributos no llave son mutuamente independientes y dependientes por completo de la llave primaria.

#### CUARTA FORMA NORMAL 4FN

Una relación esta en 4FN si esta en tercera forma normal y todas las dependencias multivaluadas en la relación son de hecho dependencias multivaluadas.

## 1.3 INTERNET

### 1.3.1 ANTECEDENTES

Para comenzar con los conceptos de un sitio Web hay que tener en cuenta que cualquier información almacenada en una red a la que se puede acceder utilizando un Web browser se llama sitio Web ó Web site. Todos los antecedentes y conceptos que hay detrás de un sitio Web se incluyen en esta sección.

La utilización de las comunicaciones en las empresas apoyadas en la informática ha pasado, en poco menos de medio siglo, por diversas etapas:

Los primeros sistemas informáticos se basaban en ordenadores centrales mainframes, que proporcionaban soporte a una red de terminales locales o remotos. Fue la era de los Departamentos de Proceso de Datos centralizados.

En los inicios de la década de los 80 surgieron dos nuevos concepto que intentaban eliminar el control y centralización de las mainframes: la informática distribuida, y la aparición y despliegue del ordenador personal o PC. Las aplicaciones de mainframe se trasladan a los ordenadores personales, en un intento de mejorar la flexibilidad, la eficiencia y la productividad de los usuarios. Este movimiento se conoce internacionalmente con el nombre de downsizing. Pronto se vio la conveniencia de comunicar las PCs mediante redes para desarrollar funciones como el intercambio de mensajes, aplicaciones y recursos compartidos, etc. Estas redes, de alta velocidad y bajo costo, se denominan Redes de Área Local o LANs, y ofrecen ventajas apreciables, frente a las mainframes (bajo coste de la tecnología, reducción del ciclo de desarrollo de aplicaciones, delegación y gestión de las aplicaciones por el usuario, entre otras)

A pesar de todo, las redes distribuidas no han sustituido, ni mucho menos, a las mainframes. La elección entre ambas ha de considerarse bajo el modelo de negocio de cada empresa, y de las necesidades de información y comunicación impuestas por el mismo. La mayor ventaja que proporciona el modelo de mainframe, la centralización de los procesos empresariales, se ha conseguido con la integración de las aplicaciones basadas en LAN en un entorno corporativo mediante las Redes WAN que proporcionan acceso a los datos del sistema central corporativo. Es el denominado upsizing.

Además de todos estos desarrollos de comunicaciones dentro de la empresa, y en conjunción con el enfoque sistémico que gobierna el mercado actual, se han desarrollado aplicaciones a nivel más global, siendo su máximo exponente lo que conocemos como Internet. La expansión de esta "red de redes" ha traído grandes ventajas a la empresa.

Más recientemente, la filosofía y tecnología subyacentes a Internet se han trasladado a las redes locales de las empresas, en forma de lo que se conoce como Intranet, revolucionando los conceptos clásicos de comunicación empresarial. De hecho, se predice que el uso de estas Webs internas superará el uso de Internet.

#### Definición y cronología de Internet.

Lo que hoy conocemos como Internet es una red WAN mundial, que permite hacer conexiones punto por punto entre usuarios remotos.

Los orígenes de Internet se remontan a 1957 cuando el gobierno de los Estados Unidos crea la Agencia de Proyectos de Investigación Avanzada (ARPA) para acelerar las investigaciones en defensa y temas afines. En 1969 el ARPA organizó un plan para comunicar a cuatro universidades mediante una red experimental de computadoras, ARPANET.

La Red de ARPANET se compone de cuatro nodos:

- Nodo 1: UCLA - Universidad de Los Angeles, California.
- Nodo 2: SRI - Instituto de Investigaciones de Stanford.
- Nodo 3: UCSB Universidad de California en Santa Bárbara.
- Nodo 4: Universidad de Utah.

En Octubre de 1969 se envía el primer mensaje de nodo a nodo entre UCLA y SRI.

El gobierno seleccionó una tecnología de comunicaciones para ARPANET llamada conmutación de paquetes, que permitía transferir datos mediante pequeños paquetes junto con su dirección destino. Como los paquetes eran pequeños, requerían poco ancho de banda, se podían transmitir por líneas telefónicas normales, dichos paquetes se enviaban por la red pasando de una computadora a otra hasta su destino final. Como las computadoras en las universidades eran diferentes, tuvieron que ser modificadas para que los paquetes, las direcciones y los datos usaran el mismo conjunto de reglas, conocido como protocolo.

En 1972, el Gobierno de Estados Unidos adoptó oficialmente ARPANET como la red de datos de la Defensa, y un año más tarde el gobierno requería que todos los nodos en ARPANET utilizaran el Protocolo de transmisión/Protocolo de Internet (TCP/IP) como protocolo estándar. Así pues, estas investigaciones dieron como resultado el protocolo TCP/IP (Transmission Control Protocol/Internet Protocol) un sistema de comunicaciones muy sólido y robusto bajo el cual se integran todas las

redes que conforman lo que se conoce actualmente como Internet. Durante el desarrollo de este protocolo se incrementó notablemente el número de redes locales de agencias gubernamentales y de universidades que participaban en el proyecto, dando origen así a la red de redes más grande del mundo, las funciones militares se separaron y se permitió el acceso a la red a todo aquel que lo requiriera sin importar de que país provenía la solicitud siempre y cuando fuera para fines académicos o de investigación (y por supuesto que pagara sus propios gastos de conexión), los usuarios pronto encontraron que la información que había en la red era por demás útil y si cada quien aportaba algo se enriquecería aún más el acervo de información existente.

Posteriormente entre 1973 y 1975, la red global es una realidad, ya que se realizan las primeras conexiones internacionales a la ARPANET: University College of London (Inglaterra) y el Royal Radar Establishment (Noruega). La figura 1.14 muestra la primera conexión internacional.

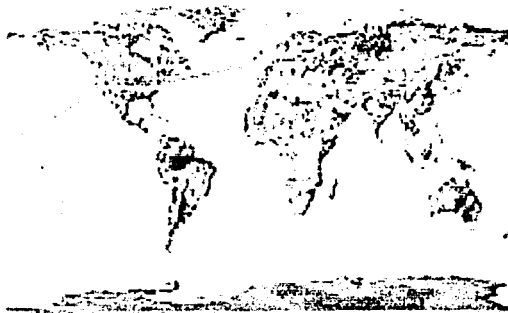


Figura 1.14 Primer conexión internacional

Ya entre los años de 1980 y 1983 se desarrolla el sistema de denominación de dominios ó Domain Name System (DNS), y se le da la responsabilidad al Instituto de Ciencias de la Información (ISI) en USC para administrar los dominios, así pues nace NIC.

Además se crean nuevas redes: La BITNET, la CSNET, FidoNet, la EARN y la EUnet. Un poco después se conectan los Países Bajos, Dinamarca y Suecia, haciendo aun más extensa el área que Internet abarcaba.

ARPANET se divide en ARPANET y MILNET. Después de que las funciones militares de la red se separaron en una sub-red de Internet, la tarea de coordinar el desarrollo de la red recayó en varios grupos, uno de ellos la National Science Foundation fue el que promovió bastante el uso de la red ya que se encargó de conectar cinco centros de supercómputo que podían ser accesados desde

cualquier nodo de la red. Eso funcionó bien al principio, pero pronto fueron superadas las cargas de tráfico previstas, fue entonces que se dio la concesión a Merit Network Inc. para que administrara y actualizara la red, se mejoraron las líneas de comunicación dando un servicio mucho más rápido, pero este proceso de mejora nunca termina debido a la creciente demanda de los servicios que se encuentran en la red.

Internet empieza a tomar relevancia, para 1986 hay más de 5,000 hosts y es cuando se da inicio a la comercialización del Web. En 1987 hay más de 28,000 hosts. Es en este periodo que México se integra al Web, junto con países como Alemania, Finlandia, Dinamarca, Francia, Islandia, Australia, Israel, Italia, nueva Zelanda, Puerto Rico y China.

Hacia el inicio de 1990, la expansión de Internet continúa, ahora ARPANET deja de existir y se sustituye por el backbone de la NSFNET y las líneas originales de 50 Kbps fueron eliminadas, en este momento los hosts superan ya los 300,000. Así mismo se empieza a brindar el servicio comercial de acceso telefónico dial-up, (world.std.com) se implementa el sistema de hipertexto parte fundamental del WWW y se da origen al protocolo ftp (File Transfer Protocol). En este periodo se conectan Argentina, Austria, Bélgica, Brasil, Chile, Grecia, India, Irlanda, Corea, España y Suiza.

La evolución de Internet en la década de los noventa y principios del nuevo siglo, ha sido espectacular, surgiendo entre 1991 y 1992 otros servicios avanzados como los Servidores de Información amplia conocido como WAIS (Wide Area Information Service), Gopher, y sobre todo el World Wide Web, que reúne, a nivel mundial, un gran número de documentos, datos y multimedia, que ha popularizado Internet.

El enorme crecimiento de Internet se debe en parte a que es una red basada en fondos gubernamentales de cada país que forma parte de Internet lo que proporciona un servicio prácticamente gratuito. A principios de 1994 comenzó a darse un crecimiento explosivo de las compañías con propósitos comerciales en Internet, dando así origen a una nueva etapa en el desarrollo de la red, es en esta etapa que comienzan las transmisiones de radio por Internet y los Centros Comerciales entran a Internet.

De entre 1996 y 1998 inicia la competencia entre navegadores (browsers) del WWW, entre Netscape y Microsoft. Ya para ese entonces muchos sitios del Gobierno de los EUA son atacados por hackers quienes cambian el contenido; como les sucedió a la CIA, el Departamento de Justicia y a la Fuerza Aérea.

Dentro de este periodo la Sociedad de Internet (Internet Society), quien se encarga de controlar el Internet, esta buscando una nueva versión del TCP/IP para poder ampliar el ahora limitado número de direcciones.



Por extraño que parezca no existe una autoridad central que controle el funcionamiento de la red, aunque existen grupos y organizaciones que se dedican a organizar de alguna forma el tráfico en ella

Actualmente, los desarrollos son numerosos: la llegada de la videoconferencia, por citar un ejemplo y otras aplicaciones pronostican que Internet es, hoy por hoy, el sistema de información mundial por excelencia.

### 1.3.2 SERVICIOS A TRAVÉS DE INTERNET

Lo interesante de Internet no es la red en sí, sino los servicios a los que se pueden acceder a través de ella. Entre los servicios principales se encuentran:

- Correo electrónico
- Grupos de discusión
- Acceso a la información que alguien haya introducido en la red.

El correo electrónico es -sin duda- el más utilizado de los servicios. Permite enviar mensajes, archivos, carpetas con información a cualquier otra persona que esté conectada. El correo electrónico ha sustituido -en gran parte- al teléfono en comunicaciones empresariales y en el mundo académico y científico. Sus características principales son: su bajo costo (mas barato que el fax), el que no es necesario que el emisor y el receptor coincidan en el tiempo (se llega a perder mucho tiempo llamando a personas que no están) y su velocidad (el correo electrónico llega en cuestión de minutos). Su gran problema en este momento es la falta de seguridad: el correo debe pasar por ordenadores intermedios que no están controlados por el emisor del mensaje ni por el receptor.

Los grupos de discusión e información son algo parecido a un pizarrón de anuncios electrónicos, es decir cualquiera puede poner información en uno de estos grupos y todos los interesados pueden leer las noticias de los grupos que le interesen. Hay más de 10.000 grupos de discusión en el mundo. La mayoría de ellos están en inglés. Aunque poco a poco los grupos de discusión en español han ido en aumento.

Un ejemplo de grupo de discusión sería uno especializado en "clonación humana". En el los médicos e investigadores interesados podrían poner la información relevante (e interesante para los lectores) con la que cuentan, y leer lo que todos los demás escribieran sobre el tema, así se establece un foro de discusión a través de Internet que permite mayor difusión de la información, así como el intercambio de ideas.

El tercer servicio interesante es acerca del acceso de la información que alguien haya introducido en la red, consiste en el que se permite publicar información en Internet y a su vez se permite que otras personas la encuentren y la lean. Su nombre: el World Wide Web .Normalmente se le conoce como "el Web" o como "WWW".

Por otra parte, cabe mencionar que Internet utiliza la tecnología Cliente-Servidor.

### 1.3.3 EL MODELO CLIENTE-SERVIDOR

La tecnología denominada Cliente -Servidor es utilizada por todas las aplicaciones de Internet/Intranet:

Un servidor es un ordenador remoto, en algún lugar de la red, que proporciona información según petición.

Un cliente funciona en su ordenador local, se comunica con el servidor remoto, y pide a éste información.

Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

Los sistemas Cliente-Servidor pueden ser de muchos tipos, dependiendo de las aplicaciones que el servidor pone a disposición de los clientes. Entre otros, existen:

- Servidores de Impresión, mediante el cual los usuarios comparten impresoras.
- Servidores de Archivos, con el cual los clientes comparten discos duros
- Servidores de Bases de Datos, donde existe una única base de datos.
- Servidores de Lotus Notes, que permite el trabajo simultáneo de distintos clientes con los mismos datos, documentos o modelos.

Los Servidores Web también utilizan la tecnología Cliente- Servidor, aunque añaden aspectos nuevos y propios a la misma.

### 1.3.4 MODELO INTERNET

El modelo Internet basado fundamentalmente en la serie de protocolos TCP/IP no consta de las mismas capas que el modelo OSI (Open Systems Interconnection), sin embargo, existe una correspondencia con el modelo OSI. En el modelo Internet tenemos cuatro capas mostradas en la figura 1.15:

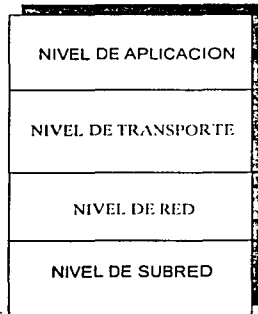


Figura 1.15 Modelo Internet

#### NIVEL DE APLICACIÓN.

Esta compuesto por las capas de aplicación, presentación y sesión del modelo OSI. Este consiste de las aplicaciones y de los procesos que utilizan la red. Este consta de los servicios DARPA: TELNET, FTP, SMTP y del sistema de archivos de red NFS, del DNS (Domain Name Server), y del ruteo de información RIP (Routing Information Protocol).

#### NIVEL DE TRANSPORTE.

Esta compuesto por la capa de transporte del modelo OSI. Ofrece servicios de entrega de datos de punto a punto. Aquí se sitúan los protocolos TCP y UDP.

#### NIVEL DE RED

Esta compuesto de la capa de red del modelo OSI. Aquí se encuentra el protocolo IP, su datagrama maneja el ruteo de los datos. IP ofrece el servicio de entrega del

paquete básico. Tenemos también como parte de este nivel el protocolo Internet Control Message Protocol (ICMP). Este efectúa tareas de control y reporte de errores.

### NIVEL DE SUBRED DE COMUNICACIONES O ACCESO A RED.

Está compuesto de la capa de enlace y la capa física del modelo OSI. Consiste de rutinas para el acceso a las redes físicas. En esta capa se utiliza el Address Resolution Protocol (ARP), el cual asocia direcciones IP a direcciones Ethernet.

Internet utiliza, aunque no exclusivamente el Protocolo de Control de Transmisión/Protocolo Internet (TCP/IP).

#### 1.3.5 TCP/IP

*TCP:* Transmision Control Protocol. El protocolo TCP ofrece un conjunto de mecanismos para establecer comunicaciones seguras y estables entre aplicaciones que corren en las computadoras ligadas en la red.

*IP:* Internet Protocol. El protocolo IP ofrece el manejo de redes individuales para establecer una sola gran red entre todas las redes.

Los servicios que soporta TCP/IP son transferencia de archivos, login remoto y correo electrónico.

Correo electrónico. Este permite que un usuario envíe y reciba información. Aunque existen muchos sistemas de correo electrónico el utilizar el TCP/IP logra que la entrega sea más confiable debido a que no se basa en computadoras intermedias para distribuir los mensajes de correo. Un sistema de entrega de correo TCP/IP opera al hacer que la máquina del transmisor contacte directamente la máquina del receptor. Por lo tanto, el transmisor sabe que una vez que el mensaje salga de su máquina local se habrá recibido exitosamente en el sitio del destino.

Transferencia de archivos. Los protocolos TCP/IP incluyen un programa de aplicación para transferencia de archivos el cual permite que los usuarios envíen o reciban archivos sin importar su tamaño de programas o datos. Como en el caso del correo electrónico la transferencia de archivos a través de una red de redes TCP/IP es confiable.

**Acceso remoto.** Este permite que un usuario que este frente a una computadora se conecte a una máquina remota y establezca una sesión interactiva. El acceso remoto hace aparecer una ventana en la pantalla del usuario, la cual se conecta directamente con la máquina remota. Cuando termina la sesión de acceso remoto, la aplicación regresa al usuario a su sistema local.

**Internet como herramienta para el comercio internacional**

Para poder tomar beneficio del Internet, es preciso atender dos cuestiones principalmente: Presencia y Promoción

Para empezar debemos contar con presencia a través de un Sitio Web. Para una buena presencia son muchos los factores a considerar, ya que un Sitio Web, debe ser, en medida de lo posible, una representación fiel de la empresa, de la gente y de su imagen corporativa. Su oficina virtual debe ser un medio "adicional" para atender posibles nuevos clientes, distribuidores o proveedores o a los que ya tiene. Y como toda empresa, el Sitio esta en cambio permanente, necesita de mantenimiento y actualización.

Ya con presencia ahora debemos procurar la promoción, no basta con estar registrado en los buscadores (search engines) se tiene que aprovechar todos los recursos posibles para los visitantes deseados, tomando en cuentas las reglas que rigen la Red, para mayor información al respecto se puede consultar la página <http://www.etiquette.net/index.html> en donde se definen a detalle dichas reglas.

### 1.3.6 WORLD WIDE WEB

En los primeros años, la interfaz de los usuarios con Internet era mediante comandos de texto desde máquinas de UNIX, en 1989 Timothy Berners-Lee, un físico del European Organization for Nuclear Research, CERN, en Ginebra, planeó el Web como una herramienta de navegación en Internet. Berners-Lee necesitaba un método más sencillo para colaborar en proyectos con físicos de diferentes localidades, para lo que propuso un sistema de hiperenlaces o hipervínculos, una red de enlaces que permitiera a los usuarios moverse fácilmente de una computadora a otra en Internet, en busca de información relacionada. Los documentos se escribían en un lenguaje abierto, el lenguaje de marcas de hipertexto (Hiper Text Markup Language, HTML), que pudiera interpretar cualquier tipo de computadora, independientemente de su sistema operativo.

El hablar de un lenguaje hipertexto se refiere a que en lugar de leer un texto rígido, con una estructura lineal como la de un libro, se puede saltar de un punto a otro

obteniendo más información, ir hacia atrás, saltar hacia otros tópicos, y navegar a través del texto con base en los intereses que el usuario tenga en ese momento.

En 1990 se unieron los lugares que usaban HTML para formar lo que se denominó el *World Wide Web*. A los documentos escritos en HTML se les llamó *páginas Web*, al servidor se le llamó *Web Server*, y al cliente *browser*.

Inicialmente las páginas Web incluían solamente texto hasta que Marc Andreesson, estudiante de la Universidad de Illinois, desarrolló un browser al que llamó Mosaic, que se ejecutaba en una PC y permitía leer páginas en HTML mediante una interfaz gráfica de usuario (GUI), parecida a la de Microsoft Windows. Netscape Corporation convirtió el Mosaic de Andreesson en el diseño de la primera versión del browser Netscape Navigator, disponible libremente en Internet en 1994, lo que permitió su distribución en todo el mundo. A sí mismo las empresas comenzaron a desarrollar lugares interesantes en el Web, y cada vez más usuarios empezaron a navegar por curiosidad, lo que condujo a un incremento de la presencia corporativa para captar más clientes.

Así pues, podemos concluir que el World Wide Web, (WWW), es un sistema de información con hipertexto o hipervínculos, es decir en HTML y multimedia creado para el intercambio de información a nivel global.

La tecnología World Wide Web incluye un Web Server, un browser, el lenguaje HTML, el localizador universal de recursos URL, y las conexiones en Internet.

## HIPERVÍNCULOS O HIPERTEXTO

Cuando hablamos del World Wide Web, es fundamental el concepto de hipervínculo. Las páginas Web, creadas bajo el lenguaje HTML, contienen palabras, imágenes y otros elementos remarcados que permiten al usuario desplazarse de una página a otra con un simple clic del ratón. La razón es que el elemento marcado ha sido relacionado a la otra página, de tal manera que, cuando hacemos clic en el ratón, el Servidor Web busca la página y la dirección indicada en el vínculo, y la pone a disposición del usuario en su browser.

## BROWSER

Un *browser* es la aplicación que permite a los usuarios navegar por el *World Wide Web*. Los usuarios tan sólo describen la dirección de una página *Web* (URL), y el browser es el encargado de buscar dicha página y enviar la información solicitada desde el Web Server, es en este momento cuando el usuario puede ver la información en el browser.

El *browser* se comunica con el servidor de Web sobre la red y trae el documento de ese servidor. Si el documento es un archivo HTML, interpreta el código HTML

contenido en ese documento, formateando y desplegando el documento. Si el documento contiene imágenes o ligas hacia otros documentos también los maneja. Algunos de los browsers más populares es el Netscape Navigator, desarrollado por Netscape Communications Corporation. Existen *browsers* disponibles casi para cada plataforma, incluyendo interfaces de usuario gráficas basadas en sistemas como Mac, Windows, X11 o de texto solamente para conexiones UNIX. Muchos de los *browsers* son *freeware* o *shareware* (es decir software gratis y compartido), algunos de ellos como el Netscape le permite usar su *browser* por un periodo de tiempo que se considera de evaluación, después del cual se espera que el usuario lo compre. Usualmente lo único que se tiene que hacer es bajarlo de la red.

## VENTAJAS DEL WORLD WIDE WEB (WWW)

El World Wide Web, se ha vuelto popular, por la facilidad de desplegar texto y gráficos a todo color en la misma página. El WWW provee capacidades para video, sonido y gráficos para que puedan ser incorporados con el texto.

Así mismo el WWW cuenta con una plataforma cruzada (cross-platform), esto quiere decir que no importa que sistema este corriendo, así que uno puede tener acceso al World Wide Web por medio de una aplicación llamada *browser*, como el Netscape Navigator, el NCSA's Mosaic, o Explorer por ejemplo.

La característica del WWW de ser distribuido implica que la información que ahí encontremos de un tamaño considerable en bytes, por lo tanto se hablaría de mucho espacio ocupado en disco, y más si nos ponemos a pensar en el sonido y la imagen que complementan la información. Para almacenar toda la información que el WWW provee, se necesitaría una gran cantidad de espacio en disco, además el manejar dicha información sería casi imposible. El WWW provee exitosamente toda esa gran cantidad de información debido a que esa información está distribuida globalmente a través de miles de *sitios* corriendo en Internet, cada uno de los cuales contribuye con el espacio para la información que publica.

Se dice que el WWW es dinámico ya que la información que está contenida en el *sitio* que la publica está en constante cambio, es decir la gente que la publica puede actualizarla en cualquier momento.

El WWW es interactivo por naturaleza, el hecho de seleccionar una liga y llegar a otra página de información es una forma de interactividad, además de que el Web nos permite diseñar pantallas que lucen como formas. La gente que navega en estas pantallas puede hacer varias selecciones o puede llenar información en campos, así como seleccionar un botón para realizar una operación en particular. Otra forma de interacción del WWW son los *imagemap* (mapas de imágenes). Un *imagemap* es como una fotografía en la página que tiene diferentes áreas

sensibles, esto último quiere decir que donde se haga clic en la imagen determinará lo que sigue a continuación. Los *imagemap* son comúnmente usados en los menús.

Además el WWW con la finalidad de brindar un nuevo sistema para publicar y distribuir información, es compatible con las viejas formas de distribución de información como lo son el ftp (File Transfer Protocol), Gopher, Usenet news, Telnet, y el e-mail. Usando solo el *browser* del WWW se puede leer la información del WWW en hipertexto, así como también obtener los archivos y la información contenida en otros sistemas de información. Hay que mencionar que se pueden crear ligas hacia la información en esos sistemas, como si se estuvieran creando ligas hacia cualquier información del WWW.

### ¿A QUIÉN LE PERTENECE EL WORLD WIDE WEB?

No hay algo o alguien en particular al que le pertenezca el WWW. Dada la enorme cantidad de *sitios* que brindan tanta información, en el WWW sería imposible para cualquier organización establecer reglas o guías. Sin embargo existe el World Wide Web Consortium, basado en el MIT (Massachusetts Institute Technologic) en los Estados Unidos; y el INRIA en Europa. El World Wide Web Consortium es una organización de particulares y organismos interesados en establecer y definir los lenguajes y protocolos que hacen el WWW (HTTP, HTML, etc.). Además provee productos, como *browsers*, servidores, etc. mismos que están disponibles sin costo alguno para los que los quieran usar.

### SITIOS WEB

Cualquier información almacenada en una red a la que se puede acceder utilizando un Web browser se llama sitio Web ó Web Site. Los propietarios y operadores de Web Site son principalmente:

- Empresas
- Gobiernos
- Universidades
- Individuos



En lo que al Comercio Internacional se refiere, un Sitio Web es la presentación interactiva de una empresa en Internet y según nuestros intereses se compone de tres partes:

- El Dominio (domain) ([www.mextrade.com](http://www.mextrade.com))
- El Servidor (host) (207.5.46.43)
- Las páginas Web. (html)

Se puede decir que el dominio es el nombre comercial de su sitio, el cual direccional al Servidor, para encontrarse con los textos e imágenes que incluyen sus páginas Web. El Servidor debe estar siempre conectado a Internet dando acceso a todo navegante a través de su Navegador (Browser).

El Sitio interactúa con sus visitantes principalmente a través del Correo Electrónico (e-mail), el cual nos permite enviar y recibir mensajes.

Este mismo sistema se puede reproducir de forma que una computadora personal funcione como servidor en una red local donde las computadoras integradas a esa red pueden acceder con un Navegador (Browser) al Sitio "Web" con la diferencia que el acceso esta limitado por la misma red. Este sistema se le llama Intranet.

Ahora bien, la posibilidad de conectar ese mismo servidor "privado" a Internet permite compartir la información de la empresa con gente de fuera, como clientes o proveedores, a esto se le conoce como Extranet.

En el World Wide Web, encontramos documentos integrados por páginas de información interactiva, también llamadas páginas Web, sitios Web, o home pages las cuales emplean hipertexto, imágenes, animaciones y sonido para el intercambio de información, entre computadoras, en la figura 1.16 se muestra un ejemplo del sitio Web de Netscape.

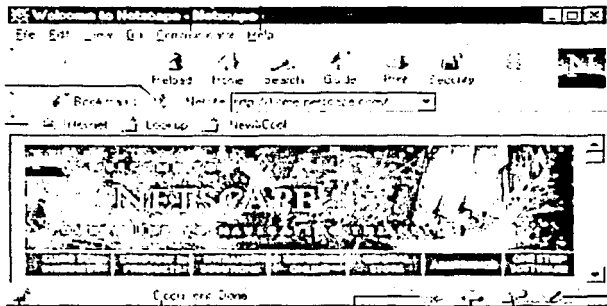


Figura 1.16 Browser Netscape

## WEB SERVER

Un Web Server es un servidor de archivos UNIX, LINUX o NT configurado con el hardware y software apropiado para responder a las peticiones de los clientes. Así pues, los Servidores Web son aquellos que permiten a los clientes compartir datos, documentos y multimedia en formato Web.

Un Web Server se usa principalmente para mantener un directorio de páginas y sitios Web. Un Servidor Web utiliza el protocolo HTTP para escuchar los requerimientos de los archivos del *browser*.

Los servidores que proveen el servicio WWW, almacenan las páginas que uno ve con un programa de navegación. Cuando se hace "clic" sobre un hipervínculo ó hipertexto en cualquier página Web, en realidad, se está solicitando un documento Web a un servidor WWW. En la figura 1.17 se muestra un esquema del entorno de un servidor WWW.

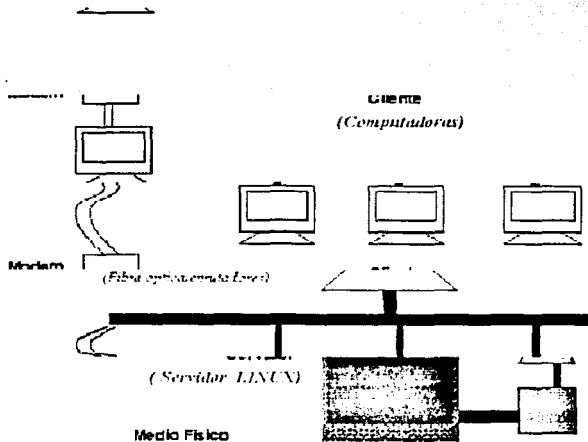


Figura 1.17 Entorno de WWW

Aunque parte de la tecnología Cliente-Servidor, el servidor Web aporta unas ventajas adicionales en aspectos muy importantes como son:

El Web se crea normalmente como un sistema abierto al que cualquiera puede contribuir y acceder desde cualquier punto de la red de la empresa: no requieren logins ni password como los sistemas tradicionales Cliente-Servidor.

Los Servidores Web despliegan información con un simple clic del ratón a través de un proceso de hipervínculo.

La información servida puede ser de cualquier tipo (datos, documentos, multimedia, etc.), gracias a la utilización de los estándares Internet. Esa información es de sólo lectura pues, a diferencia de los sistemas normales Cliente-Servidor, el usuario no puede hacer cambios en el dispositivo original de los datos.

Dado que el Servidor de Web es de tan fácil acceso, ello hace posible publicar información de forma instantánea en toda la compañía mediante un simple almacenamiento de la misma en el servidor.

Un Servidor de Web en la Intranet puede servir la copia de un archivo a toda la organización, de la misma forma que un servidor Web en Internet puede servir a todo el mundo.

La amplitud de la red suele ser mayor que otros sistemas Cliente-Servidor, pues la mayor parte de Intranet se construye sobre Redes de Área Extensa (WAN), más que sobre redes LAN.

## URL

El Localizador Universal de Recursos URL, es la dirección de una página Web:  
<http://www.cinstrum.com/indice.htm>

El URL, es la dirección de un servidor ó de una página Web. El URL aparece en la barra de dirección del browser.

El URL tiene tres partes:

1. El <http://> hace alusión al protocolo de comunicación, "Hyper Text Transfer Protocol". Con [http:](http://) se indica que el browser envía una petición usando el protocolo de transferencia de hipertexto (HTTP), este protocolo permite realizar la petición al Web Server que interpretará el resto de cadena URL.
2. La segunda parte identifica al servidor de WWW.
3. La tercera parte indica la página actual e identifica su localización en el servidor WWW.

Resumiendo, el formato es:

`http://www.servidor.tipo.pais/ página.html`  
(1)                      (2)                      (3)

## HTML

Una de las bases del Web es el lenguaje de marcado de hipertexto (HTML). HTML es el formato primario en el cual se distribuye los documentos en el Web. HTML fue ideado como un formato para que los científicos pudieran utilizar diversas computadoras y compartir información a su conveniencia en red; para ello se requirieron varias características como: la independencia de plataformas, en la cual un documento se puede desplegar de igual modo en computadoras con diferentes capacidades (es decir, fuentes, gráficos y color). El Hipertexto, que significa que cualquier palabra o frase de un documento puede hacer referencia a otro documento, permitiría una navegación más fácil entre y en los muchos y extensos documentos que hay en el sistema. Los documentos estructurados rigurosamente permitirían aplicaciones avanzadas como la conversión de documentos y desde otros formatos y la búsqueda en bases de datos de texto. La manera más fácil de ver un documento HTML es dejar que un navegador interprete el archivo por nosotros. Un documento HTML está en formato de texto ASCII y puede ser creado utilizando la mayoría de los editores de texto aunque existen editores específicamente para edición en HTML.

## 1.4 SOFTWARE LIBRE EN INTERNET

En 1971, Richard Stallman comenzó su carrera en el MIT (Instituto de Tecnología de Massachussets), trabajando con un grupo que usaba exclusivamente software libre, para los 80's casi todo el software era propietario, esto hizo que en 1984 Richard Stallman decidiera reunir suficiente software libre para lanzar el proyecto GNU desarrollando un sistema operativo tipo UNIX compartido con otras personas sin firmar un acuerdo de confidencialidad o un acuerdo de licencia de software.

Es importante aclarar que el término «*free software*» (software libre) no tiene nada que ver con el precio sino con la libertad de uso, así que un software libre tiene las siguientes características:

Tiene libertad para ejecutar el programa, con cualquier propósito.

Tiene la libertad para modificar el programa para adaptarlo a sus necesidades. (Para que esta libertad sea efectiva en la práctica, se debe tener acceso al código fuente, porque modificar un programa sin disponer del código fuente es extraordinariamente dificultoso.)

Tiene la libertad para redistribuir copias.

Tiene la libertad para distribuir versiones modificadas del programa, de tal manera que la comunidad pueda beneficiarse con sus mejoras.

Ya que el proyecto GNU requería de un sistema operativo no propietario, en 1990 ya había escrito la mayoría de los componentes excepto el kernel, entonces Linus Torvalds desarrollo uno. Combinando LINUX con el casi completo sistema GNU resulto el sistema operativo completo GNU basado en LINUX.

Sin embargo, el proyecto GNU no esta limitado a sistemas operativos también provee un amplio espectro de software.

La meta de GNU era dar libertad a los usuarios, no sólo ser popular. Por lo tanto, se usan términos de distribución que impidan que el software GNU se transforme en software propietario. El método utilizado se denomina ***copyleft***.

El ***copyleft*** usa la ley de copyright, pero para servir a lo opuesto de su propósito usual: en lugar de ser un medio de privatizar el software, se transforma en un medio de mantener libre al software.

La idea central del copyleft es que da cualquiera el permiso para correr el programa, copiar el programa, modificar el programa y redistribuir versiones modificadas--pero no le da permiso para agregar restricciones propias. De esta manera, las libertades cruciales que definen al «software libre» quedan garantizadas para cualquiera que tenga una copia; se transforman en derechos inalienables.

La implementación específica de copyleft para la mayoría del software GNU es la Licencia Pública General de GNU GPL (GNU General Public License).

Como se requerían fondos para el desarrollo de software libre en 1985 se creó la Fundación para el Software Libre (FSF) para recaudar dinero libre de impuestos además de estar dedicada a eliminar las restricciones sobre el copiado, redistribución, entendimiento, y modificación de programas de computadoras. Muchas organizaciones distribuyen cualquier software libre que este disponible. En cambio, la Fundación para el Software Libre se concentra en desarrollar nuevo software libre, y en hacer de este software un sistema coherente, el cual puede

#### 1.4.1 SISTEMAS OPERATIVOS GNU.

##### LINUX

LINUX es un sistema operativo bajo GNU GPL que se creó inicialmente como un hobby por un joven estudiante, Linus Torvalds, de la Universidad de Helsinki en Finlandia. Linus tenía interés en desarrollar un sistema que excediera las capacidades del sistema Minix, un sistema pequeño de UNIX desarrollado por Tanenbaum. En 1991 Linus liberó la primera versión del kernel de LINUX, fue hasta 1994 cuando libero la versión 1.0 del kernel de LINUX con la ayuda de desarrolladores de todo el mundo.

Debido a la misma naturaleza de la funcionalidad y disponibilidad de LINUX, un gran número de programadores de software ha tomado el código de la fuente y lo ha adaptado para satisfacer sus necesidades individuales.

Existen varias razones por las que LINUX figura como un sistema operativo funcional:

- Requerimientos de hardware muy accesibles.
- Código fuente disponible en Internet.
- Su distribución no tiene costo (o un par de dólares por CD)
- Estable, la caída de una aplicación muy probablemente implica derrumbar el sistema operativo.

- Altamente fiable comparado con el reboot regular requerido en sistema de Windows.
- Ambiente de desarrollo completo, incluso C, C++, compiladores de Fortran, el toolkits como Qt e idiomas del scripting como Perl, Awk y sed.
- Medios de la gestión de redes excelentes, permitiéndole compartir CPUs, modems, impresoras etc.
- Una gran variedad software comercial está disponible si sus necesidades no se satisfacen por el software libre.
- Un sistema operativo que es fácilmente actualizable
- Multitareas.
- Multiprocesos.
- Cuenta con un sistema de ventanas llamado X.
- Disponible para diferentes arquitecturas, incluyendo Intel, SPARC, Alpha y PowerPC.
- Compatible con diferentes aplicaciones, incluyendo servidores de Web como APACHE, servidores de mail como sendmail, y servidores de bases de datos como Oracle, Informix, MySQL y Postgres.

#### 1.4.2 MANEJADOR DE BASES DE DATOS RELACIONALES GNU.

##### MYSQL

MySQL es un manejador de base de datos GNU GPL, surgió como una variación de msql por la compañía MySQLAB con modificaciones para soporte a ambientes multi-usuario con aplicaciones OLTP (online transaction process), de alta concurrencia y mayor flexibilidad para consultas y migraciones. MySQL es altamente confiable y fácil de usar, actualmente ofrece una gran variedad de funciones útiles.

MySQL además de soporte al estándar SQL (ANSI), compila en varias plataformas y tiene habilidades del multithreading en servidores UNIX lo que le da un gran rendimiento. Para ambiente Windows MySQL puede correr como un servicio en Windows NT y como un proceso normal en Windows 95/98.

MySQL es un sistema cliente/servidor que consiste de un servidor SQL multi-threaded que soporta diferentes backends, como programas clientes, herramientas administrativas y diferentes interfaces programadas.

### 1.4.3 LENGUAJES DE PROGRAMACIÓN GNU.

#### PHP

PHP proviene de un producto llamado PHP/FI Personal Home Page / Forms Interpreter. PHP/FI fue creado por Rasmus Lerdorf en 1995 inicialmente como un simple conjunto de scripts de Perl. Cuando más funcionalidad fue requerida, Rasmus escribió y libero al público una aplicación de C más grande que pudiera comunicarse con bases de datos permitiendo a los usuarios que desarrollaran aplicaciones de Web dinámicas.

El script de PHP se integra dentro del código HTML y en el servidor de Web un intérprete se encarga de leer y ejecutar el código antes de ser enviado como resultado la pagina HTML al navegador del usuario. El trabajar con PHP incluye tratar con formularios, distinguir navegadores, hacer funciones para evitar la duplicidad de código y consultar bases de datos de todo tipo por mencionar de las capacidades de PHP.

La diferencia entre un script de PHP y un script CGI escrito en otros lenguajes de programación como Perl o C es que en un script en PHP en vez de tener lotes de comandos como salida para HTML, tiene código PHP incrustado en HTML.

Los programas de PHP al igual que los programas hechos en CGI pueden coleccionar datos, generar páginas dinámicas, o enviar y recibir cookies, además de ofrecer la ventaja de tener un amplio soporte de bases de datos con código muy simple.

Entre los RDBMS que soporta PHP se encuentran:

- Adabas
- DIngresOracle (OCI7 and OCI8)
- Dbase
- PostgreSQL
- MSQSLSolid
- MS-SQL
- Sybase



- DB2
- MySQL

#### 1.4.4 SERVIDORES WEB

##### APACHE

El proyecto APACHE es desarrollado por diferentes colaboradores del mundo como un esfuerzo para lograr un servidor de Web robusto disponible al público. A esta organización se le conoce como APACHE Group.

En febrero de 1995, el servidor de software de dominio público más popular en la Web era el demonio HTTP desarrollado por Rob McCool del National Center for Supercomputing Applications de la Universidad de Illinois. Sin embargo el desarrollo de httpd quedo detenido después de que Rob McCool dejo la NCSA, para entonces algunos desarrolladores de Web tenían sus propias versiones de httpd con correcciones al código original. Un pequeño grupo de desarrolladores de Web se contactaron via e-mail para compartir las mejoras hechas individualmente en un espacio compartido de información en Internet, fue así como ocho contribuyentes formaron el grupo original APACHE.

Actualmente el servidor de Web APACHE ocupa el número uno de los servidores de Web en Internet.

## CAPITULO 2. DISEÑO DEL SISTEMA

El desarrollo del Sistema de Cotización e Inventarios para el Laboratorio de Metrología se basó en el modelo en espiral de ingeniería de software por ser el que nos permitirá evaluar riesgos, hacer entrevistas continuas con los investigadores, además de facilitar el desarrollo en diferentes etapas de pruebas y entregas parciales de aprobación hasta liberar el sistema completo.

### 2.1 ANÁLISIS DE LA PROBLEMÁTICA ACTUAL.

El Laboratorio de Metrología del Centro de Instrumentos de la Universidad Nacional Autónoma de México ofrece servicios de calibración y medición de instrumentos a diferentes empresas del interior y exterior de la República Mexicana, las cuales tienen conocimiento de ello a través del diario oficial, la lista de proveedores de SECOFI, o congresos organizados por la Universidad Nacional Autónoma de México.

Para entender la operación del Laboratorio de Metrología, se realizaron cuestionarios y entrevistas al personal del mismo con el objeto de:

- Estudiar los procedimientos de administración existentes.
- Identificar funciones, procesos y datos.
- Detectar problemáticas.
- Identificar requerimientos de la organización de acuerdo a sus reglas de negocio.
- Conocer la infraestructura tecnológica, su uso, disponibilidad y aprovechamiento.
- Diseñar un esquema conceptual acorde a las necesidades del Laboratorio de Metrología.

Todo lo anterior con el fin de presentar la situación actual encontrada y utilizarla como marco de referencia para el diseño e implementación del sistema de cotización e inventario.

El procedimiento **solicitud-cotización-inventario** engloba la operación en cuanto se refiere a prestación de servicios del Laboratorio de Metrología por lo que se analizará en el siguiente punto.

### 2.1.1 PROCEDIMIENTO DE SOLICITUD-COTIZACIÓN-INVENTARIO.

El procedimiento de **solicitud-cotización-inventario** se refiere al proceso que se lleva a cabo por parte del cliente y del centro de instrumentos para realizar un servicio.

Inicialmente el cliente contacta al personal del Laboratorio de Metrología quien puede ser la secretaria en turno o bien alguno de los investigadores (metrólogos), para que sea atendida su **solicitud** de servicio enviándola por fax, correo, o bien presentándose en las instalaciones del centro de instrumentos.

Una vez recibida la **solicitud** de servicio el investigador responsable se encarga de realizar una **cotización** la cual será archivada y devuelta al cliente en un periodo de 48 horas.

Cuando el cliente recibe la **cotización**, tiene un limite de 30 días para aceptar el servicio, si este es el caso enviará el instrumento al Laboratorio de Metrología.

El investigador responsable recibirá el instrumento registrando en una libreta los siguientes datos:

- **OrdenSM:** Número consecutivo de control formado por los caracteres SM más los dos últimos dígitos del año en curso, más tres dígitos iniciando en 001.
- **Instrumento:** Nombre del instrumento.
- **Empresa:** Nombre de la empresa.
- **Fecha de recepción:** Fecha en la que ingresa el instrumento al Laboratorio de Metrología.
- **Fecha de entrega:** Fecha tentativa de entrega del instrumento.
- **Estatus del servicio:** Este campo indica si el servicio fue concluido.
- **Responsable:** Nombre del investigador responsable del servicio al instrumento.

En la tabla 2.1 se muestra un ejemplo de la información contenida en la libreta.

Osm	Instrumento	Empresa	Fecha de recepción	Fecha de entrega	Estatus	Responsable
SM01001	BLOQUE PATRÓN	PIASA	08/01/01	22/01/01	✓	ING. BENJAMÍN VALERA OROZCO

Tabla 2.1 Control de ordenes SM

El investigador responsable deberá llenar el formato CISM-01 figura 2.1, y entregará una parte de este como comprobante al cliente, así como deberá pegar otra parte del formato al instrumento para llevar un control de los instrumentos ingresados al Laboratorio de Metrología.



Es importante hacer notar que sólo se dará servicio a los instrumentos a los que se les haya registrado en la libreta con un número de ORDEN SM, y esto se hace cuando físicamente el instrumento se encuentra en el Centro de Instrumentos.

Cuando llega un conjunto de instrumentos iguales como es el caso de los bloques patrón y se les va a realizar el mismo servicio, entonces solo se llena una vez el formato CISM-01 y genera una sola ORDEN SM.

Ya que no todos los instrumentos cuentan con un número de serie el campo Serie del formato CISM-01 algunas veces queda vacío.

El Centro de Instrumentos ofrece servicios estándares sin embargo se puede contemplar un servicio especial.

Para llevar un control de los servicios realizados en el Laboratorio se hace un inventario en un archivero en donde se guardan por servicio a un instrumento, la solicitud de servicio, la cotización del servicio y la parte que corresponde al Centro de Instrumentos del formato CISM-01.

Cuando el servicio al instrumento es concluido el investigador lo marca con estatus de terminado en la libreta, y se le comunica al cliente que puede recogerlo.

En la figura 2.2 se muestra un diagrama que esquematiza el procedimiento de *solicitud-cotización-inventario* del Centro de Instrumentos.

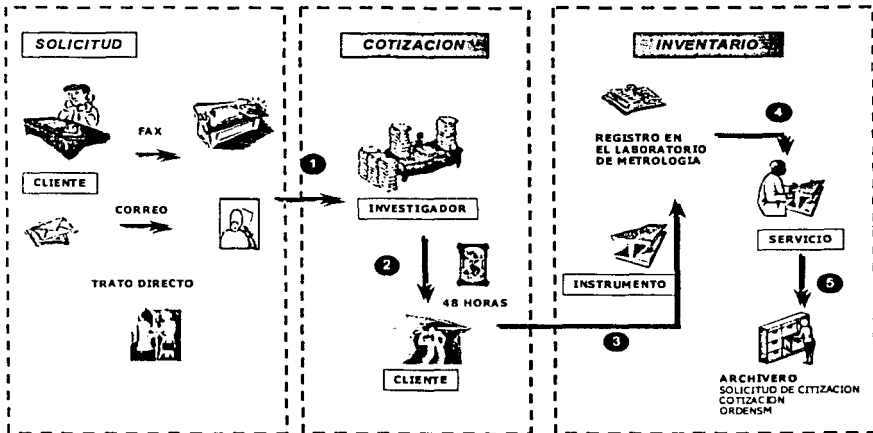


Figura. 2.2 FLUJO DEL PROCEDIMIENTO SOLICITUD-COTIZACIÓN-INVENTARIO

En el proceso administrativo de atención a clientes e identificó la siguiente problemática:

Durante la recepción de solicitudes de cotización, el tiempo de respuesta, al usar los medios convencionales como mensajería o fax, es ineficiente, poniendo en riesgo la confiabilidad del proceso de atención al cliente.

El tiempo de cotización en respuesta a la solicitud es aproximadamente de 48 horas, lo que implica una pérdida de tiempo tanto para el cliente como para el laboratorio.

El proceso manual de inventario de órdenes de servicio se convierte en un proceso redundante de duplicación de la información de solicitudes de cotización.

Se invierte una cantidad considerable de recursos humanos en el proceso de *solicitud-cotización-inventario*, tanto de personal administrativo como técnico.

El seguimiento del proceso *solicitud-cotización-inventario* se realiza con el apoyo de bitácoras escritas, entorpeciendo búsquedas eficientes. La administración y manejo de documentos escritos presenta serias carencias frente a la administración electrónica.

La información del servicio para el cliente no esta disponible en cualquier momento

La descentralización de la información genera retrasos y duplicidades que hacen ineficiente la respuesta esperada por parte del cliente.

Se tuvo una entrevista con tres de los investigadores del Laboratorio de Metrología en la cual se contestó el siguiente cuestionario para determinar necesidades y alcances del sistema.

1. ¿A qué se dedica el Laboratorio de Metrología?

En el Laboratorio de Metrología se encarga de medición y calibración de instrumentos, de la Universidad o a empresas independientes del interior y exterior de la República.

2. ¿Cómo es el proceso de solicitud de servicio?

El cliente se pone en contacto con el laboratorio vía telefónica o por mensajería solicitando el servicio.

3. ¿Quién atiende la solicitud de servicio?

La secretaria del Centro de Instrumentos, posteriormente la canaliza a alguno de los investigadores del laboratorio.

4. ¿Cuánto tiempo pasa desde que se recibe la solicitud y se entrega al investigador?

Esto es variable, máximo un día.

5. ¿Cómo se hace la evaluación del servicio?

Se evalúa la disponibilidad del personal para realizar el servicio, y se envía una cotización por fax o mensajería al cliente, en un tiempo no mayor a 48 horas desde que la solicitud fue recibida por la secretaría del Centro de Instrumentos.

6. ¿Cuánto tiempo es vigente la cotización?

La cotización tiene una vigencia de 30 días desde que es enviada al cliente.

7. ¿Se cuenta con un formato de cotización?

Cada investigador tiene su propio formato.

8. ¿Qué pasa cuando el cliente acepta la cotización?

Debe enviar o traer personalmente el instrumento al Laboratorio de Metrología y se corrobora la información con la cotización que se elaboró, después se anota en una libreta el número de orden, instrumento, empresa, fecha de recepción, fecha estimada de entrega y responsable.

Además se llena el formato CISM-01 y se pega la sección correspondiente al instrumento y la correspondiente se le entrega al cliente o en caso de que haya sido enviado el instrumento por mensajería solo se pega la sección al instrumento.

9. ¿Cuándo se termina el servicio como se guarda su histórico?

Existe un archivero en donde se guarda la cotización, y el formato CISM-01.

10. ¿Cómo sabe el cliente cuando esta terminado el servicio?

El cliente tiene que comunicarse al laboratorio y el investigador que atendió la llamada verifica en la libreta el estado del servicio.



Para resolver la problemática detectada en el Centro de Instrumentos, se eligió desarrollar un sistema que logre:

- Centralizar la información, para contar con alta disponibilidad.
- Reducir los tiempos de respuesta automatizando la atención al cliente, evitando el seguimiento del personal administrativo del Centro de Instrumentos en solicitudes de servicio y estado del mismo.
- Automatizar el proceso de cotización al contar con una plantilla en el sistema que se actualice para cada cliente sin intervención del responsable del laboratorio, excepto en servicios especiales.
- Eliminar redundancia de información.
- Llevar un control de cotizaciones.
- Controlar la secuencia del número de orden sm.
- Tener un inventario actualizado y disponible de las historia de las ordenes sm.
- Consultas eficientes de cotizaciones y de ordendes sm.
- Seguridad de la información con lo que respecta al proceso *solicitud-cotización-inventario*.
- Difusión de los servicios que ofrece el Laboratorio de Metrología en cualquier parte del mundo y en cualquier momento.
- Aprovechar la infraestructura tecnológica con la que cuenta el Laboratorio de Metrología.
- Licencias públicas de software que no generen gastos al Laboratorio de Metrología.

#### Análisis de riesgos del sistema

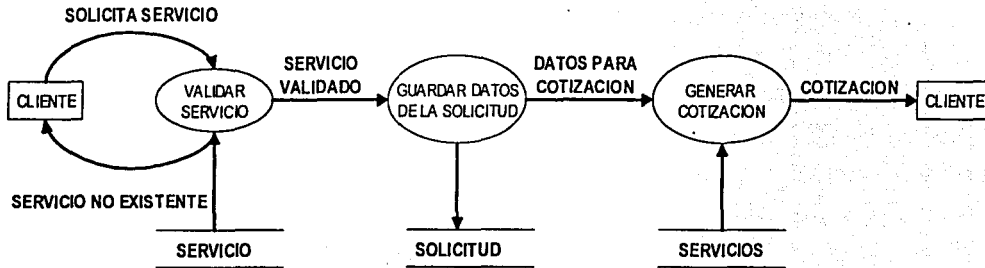
Disponibilidad de tiempo de personal: Las revisiones deben en ser en conjunto con todos los investigadores y los tesisistas.

Modificaciones al prototipo: Las observaciones hechas por los investigadores deben ser puntuales y finitas.

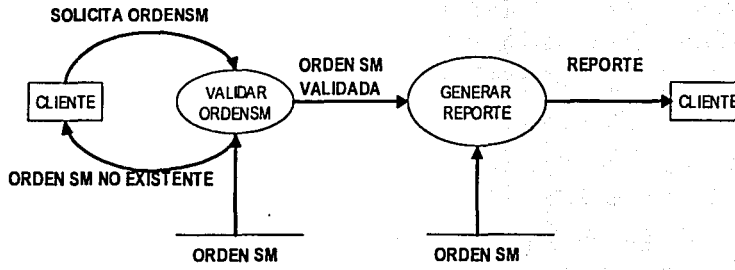
## 2.2 DIAGRAMAS DE FLUJO DE DATOS

A continuación se muestran los diagramas que representan el flujo de los datos y los procesos participantes en las actividades de la problemática planteada.

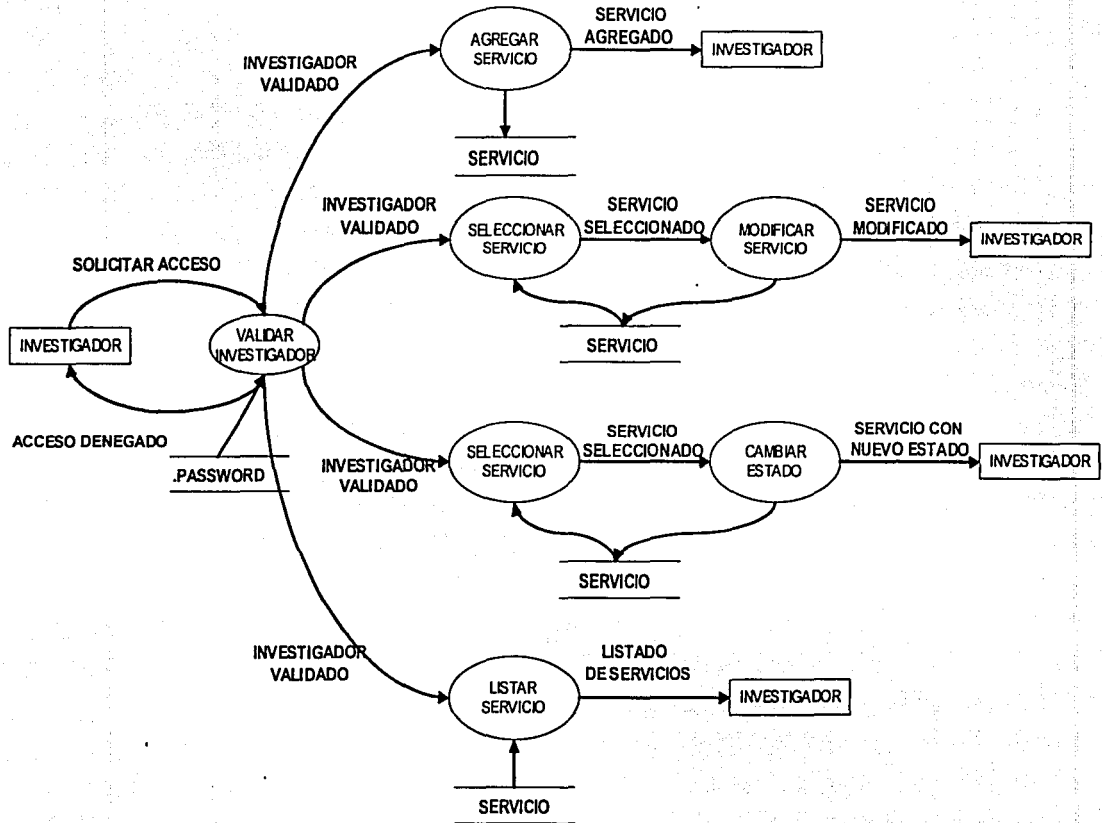
### DFD1 SOLICITUD DE COTIZACION



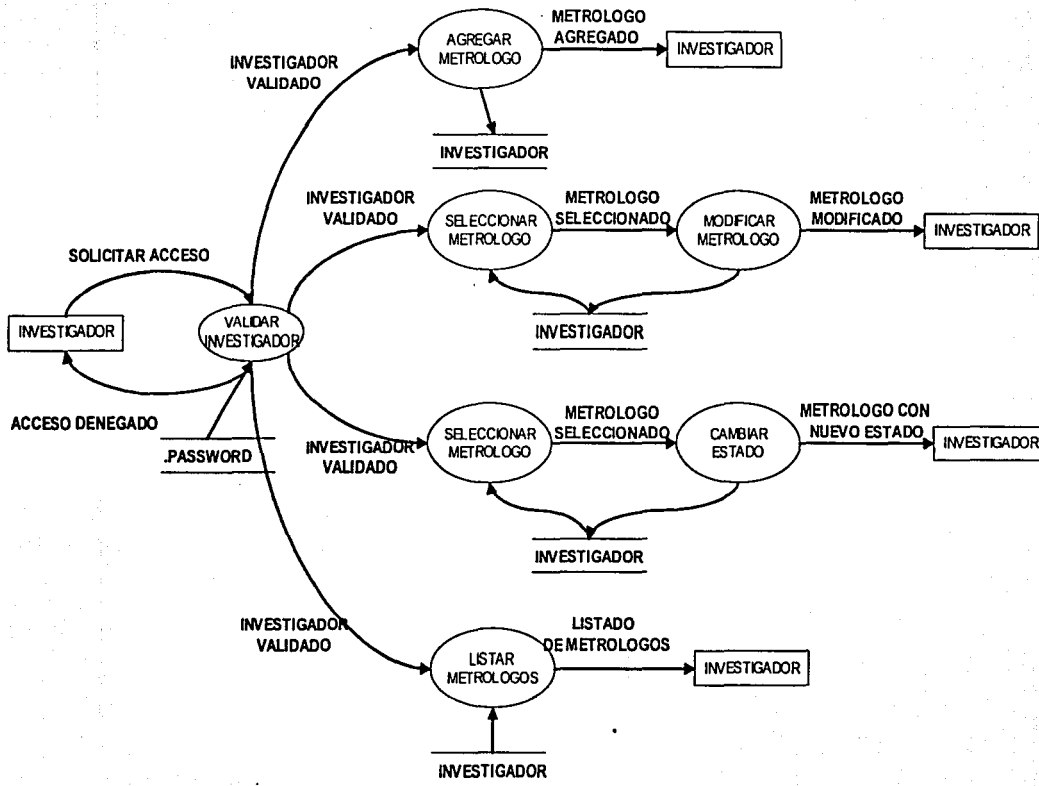
### DFD2 CONSULTA DE ORDEN SM



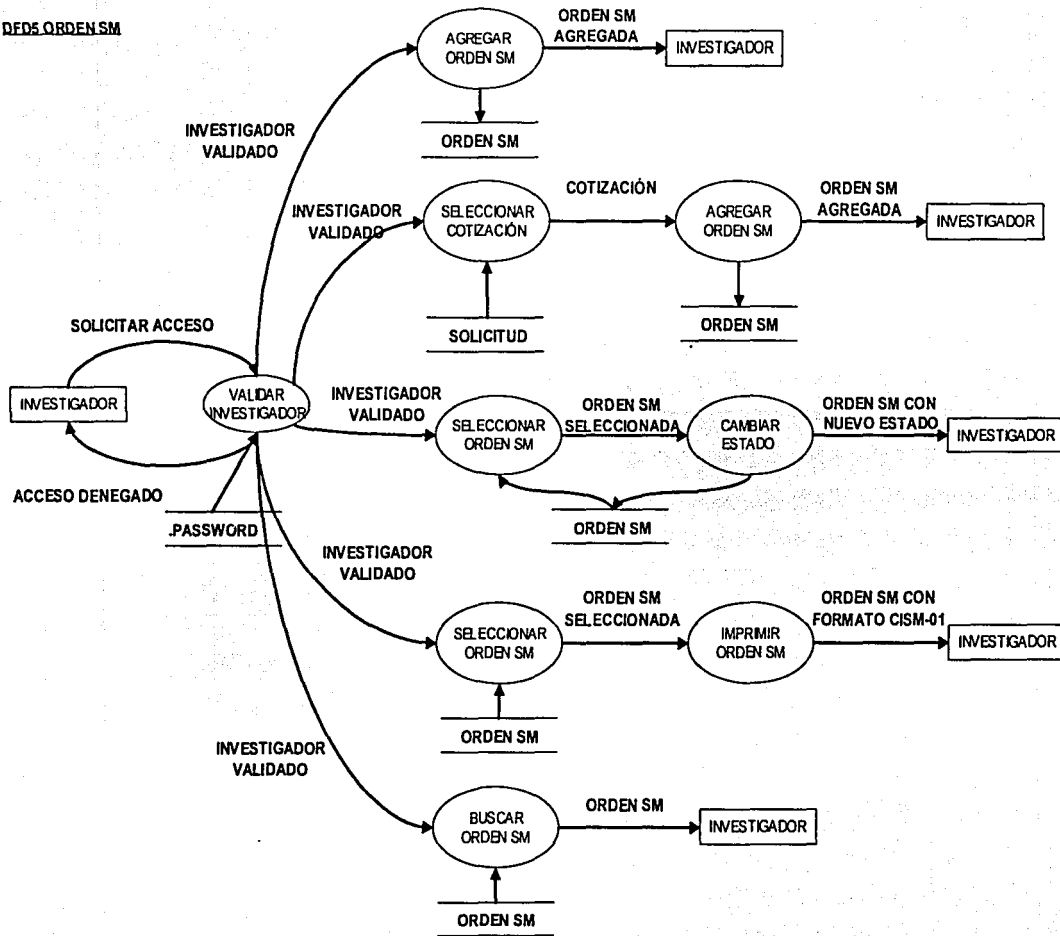
## DFD3 SERVICIOS

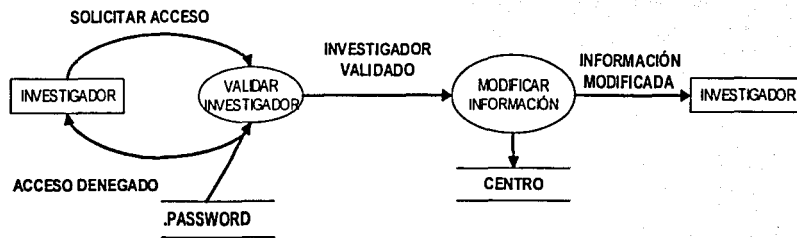
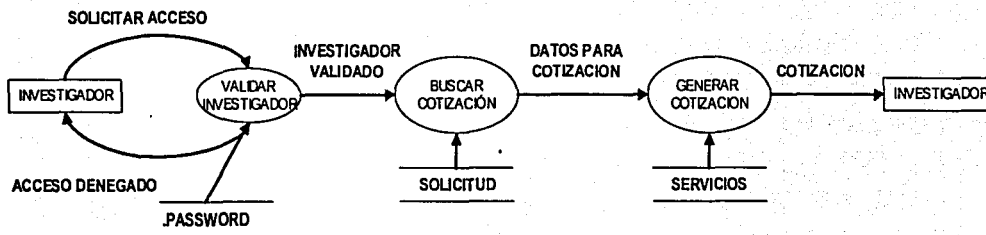


## DEFINICIÓN DE METROLOGOS



DFDS.ORDEN SM



**DFD6 ACTUALIZACIÓN DE LA INFORMACIÓN DEL LABORATORIO DEL CENTRO DE INSTRUMENTOS****DFD7 BÚSQUEDA DE COTIZACIÓN**

## 2.3 DISEÑO Y MODELADO DE LA BASE DE DATOS.

De acuerdo a un análisis del procedimiento de *solicitud-cotización-inventario* mostrado en la Figura 2.3 se define en este apartado un esquema conceptual respaldado en la teoría de bases de datos tratada en el capítulo anterior.

Asimismo se propone el diagrama Entidad/Relación de la figura 2.3 con la siguiente descripción de entidades y relaciones.

### 2.3.1 ENTIDADES Y ATRIBUTOS

De la definición de entidad del capítulo de conceptos básicos se proponen las siguientes entidades con sus atributos correspondientes:

Entidad	Atributo
INVESTIGADOR	Clave de investigador Nombre Título Cargo Descripción del cargo Email
EMPRESA	Clave de la empresa Nombre de la empresa Dirección de la empresa Fax de la empresa Teléfono de conmutador Clave del contacto Nombre del contacto Teléfono del contacto Email del contacto
INSTRUMENTO	Clave de instrumento Nombre Número de serie Marca Modelo Material Otras especificaciones
SERVICIO	Clave del servicio Nombre Incertidumbre Precio Vigencia
LABORATORIO	Clave del laboratorio. Nombre de la institución. Nombre de la sección. Cuenta de pagos por depósito. Nombre de la cuenta de depósito. Teléfono Fax

**Las asociaciones identificadas:**

- Un empresa involucra una o mas solicitudes
- Una solicitud involucra un servicio.
- Una solicitud involucra un instrumento
- Una solicitud involucra una empresa
- Una solicitud involucra una ordensm
- Una ordensm involucra una solicitud
- Una ordensm involucra un investigador
- Un investigador involucra una o más ordenesm
- El laboratorio tiene un responsable.
- Solo un investigador es responsable del laboratorio.

De las asociaciones identificadas se proponen las siguientes entidades.

Asociacion	Atributo
ORDENSM	Ordensm Estado de la orden Fecha de Recepción Fecha de entrega
SOLICITUD	Clave de solicitud Fecha de solicitud Cantidad de instrumentos. Precio de solicitud



DIAGRAMA ENTIDAD/RELACION

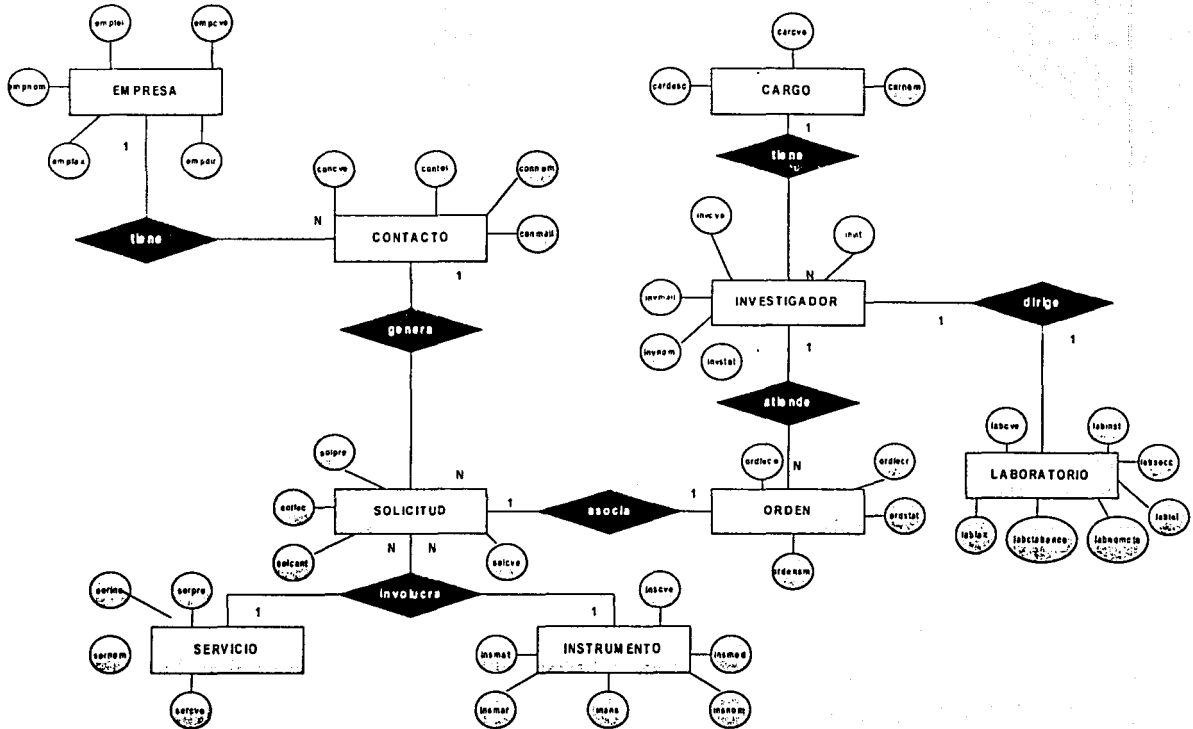


Figura 2.3 DIAGRAMA ENTIDAD/RELACION

### 2.3.2 PROCESO DE NORMALIZACIÓN.

*1FN: Una relación esta en 1FN si todos los atributos de la relación contienen valores atómicos.*

Las entidades propuestas cumplen con la 1FN ya que en todas ellas sus atributos no se pueden decomponer porque perderían su significado, es decir todos son valores atómicos.

*2FN: Una relación está en 2FN si todos los atributos no llave dependen totalmente de la llave primaria.*

Para cumplir con la 2FN es necesario proponer llaves primarias para cada entidad:

En la entidad INVESTIGADOR se propone la clave investigador como llave primaria:

clave de investigador → investigador(nombre,título, cargo, descripción del cargo, email)

De lo propuesto se deduce que todos los atributos no llave no dependen totalmente de la llave primaria clave del investigador, este es el caso del cargo del investigador ya que si el investigador ya no trabaja más en el Centro de Instrumentos el cargo desaparecerá, siendo así se propone la entidad un catálogo de cargos, generando dos entidades.

clave del investigador → investigador(nombre,título, email)

clave de cargo → cargo(nombre del cargo, descripción del cargo)

Esta división implica crear dos nuevas asociaciones.

- Un investigador tiene un cargo
- Un cargo lo tienen varios investigadores

Con esta división la dos entidades se encuentran en segunda forma normal

En la entidad EMPRESA solicitante del servicio se propone la clave de la empresa como llave primaria.

clave de la empresa → empresa(nombre de la empresa, dirección de la empresa, telefono del conmutador, fax, nombre del contacto, telefono del contacto, email del contacto)

En la propuesta de usar la clave de la empresa como llave primase se detectó que que el teléfono del contacto y e-mail del contacto dependen directamente del nombre del contacto por lo que se sugiere una entidad llamada CONTACTO quedando las relaciones como:

clave de la empresa → empresa(nombre de la empresa,dirección de la empresa, telefono del conmutador, fax)

clave del contacto → contacto(nombre del contacto, telefono del contacto, e-mail del contacto)

Generando la división de la entidad EMPRESA dos nuevas asociaciones:

- Una empresa involucra una o más contactos.
- Un contacto involucra una empresa.

Con esta división las dos entidades cumplen con que todos los atributos no llave dependen totalmente de la llave primaria, y por lo tanto se encuentran en segunda forma normal.

En la entidad INSTRUMENTO se propone la clave del instrumento como llave primaria, cabe destacar que no se utilizo el número de serie comollave primaria porque en contadas ocasiones los instrumentos no tienen una número de serie por su antigüedad o porque son piezas que no lo llevan, por este motivo se sugiere el campo clave de instrumento para control interno del Centro de Instrumentos.

clave del instrumento → instrumento(nombre del instrumento, número de serie, marca, modelo, material, otras especificaciones)

De lo propuesto se deduce que todos los atributos no llave dependen totalmente de la llave primaria clave de instrumento, lo que significa que la entidad INSTRUMENTO se encuentra en segunda forma normal.

En la entidad SERVICIO se propone la clave del servicio como llave primaria.

clave del servicio → servicio( nombre del servicio, incertidumbre del servicio, precio del servicio, vigencia del servicio)

La propuesta resulta correcta porque todos los atributos no llave dependen totalmente de la llave primaria clave del servicio y cumple con la 2FN.

De la entidad ORDENSM se propone un el campo ordenism que es un número consecutivo de ordenism como la llave primaria .

ordenism→ ordenism(fecha de recepción, fecha de entrega, estado de la ordenism)

Ya que todos los atributos no llave dependen totalmente de la llave primaria la entidad ORDENSM esta en segunda forma normal.

La entidad SOLICITUD la llave primaria clave de la solicitud.

clave de solicitud → solicitud(fecha de la solicitud, cantidad de instrumentos)

Como todos los atributos no llave dependen totalmente de la llave primaria se dice que la entidad SOLICITUD se encuentra en segunda forma normal.

Finalmente la entidad LABORATORIO se propone la clave del laboratorio como llave primaria.

clave del laboratorio → servicio( nombre de la institución, nombre de la sección, cuenta bancaria de depósito, nombre de la cuenta bancaria, telefono del laboratorio, fax del laboratorio)

La propuesta resulta correcta porque todos los atributos no llave dependen totalmente de la llave primaria clave del servicio y cumple con la 2FN.

*3FN: Una relación esta en 3FN si los atributos no llave son mutuamente independientes y dependientes por completo de la llave primaria.*

La tercera forma normal implica que todos los atributos no llave sean dependientes por completo de la llave primaria lo que se corroboró en la segunda forma normal.

En cuanto a lo que respecta que sean mutuamente independientes es decir que ninguno de ellos pueda actualizarse sin tomar en cuenta cualquier ningún otro atributo.

En la entidad INVESTIGADOR los atributos nombre, título y e-mail pueden ser actualizados aisladamente sin impactar a ningún otro atributo cumpliendo así la tercera forma normal.

clave del investigador → investigador(nombre, título, email)

En la entidad CARGO los atributos nombre del cargo y descripción del cargo pueden ser modificados independientemente de cualquier otro campo cumpliéndose así la tercer forma normal.

cve de cargo → cargo(nombre del cargo, descripción del cargo)

En la entidad EMPRESA los atributos nombre de la empresa, dirección de la empresa, telefono del conmutador y fax pueden ser modificados independientemente cumpliendo con la tercera forma normal

clave de la empresa → empresa(nombre de la empresa, dirección de la empresa, telefono del conmutador, fax)

En la entidad CONTACTO los atributos nombre del contacto, telefono del contacto y email del contacto pueden ser modificados independientemente cumpliendo con la tercera forma normal.

clave del contacto → contacto(nombre del contacto, telefono del contacto, e-mail del contacto)

Los campos nombre del instrumento, número de serie, marca, modelo, material y otras especificaciones pueden ser modificados independientemente cumpliendo así la tercera forma normal.

clave del instrumento → instrumento(nombre del instrumento, número de serie, marca, modelo, material, otras especificaciones)

En el caso de la entidad SERVICIO los atributos nombre del servicio, incertidumbre del servicio, precio del servicio y vigencia del servicio pueden ser actualizados independientemente cumpliendo con la tercera forma normal.

clave del servicio → servicio( nombre del servicio, incertidumbre del servicio, precio del servicio, vigencia del servicio)

Acérca de la entidad ORDENSM los atributos fecha de recepción, fecha de entrega y estado de la ordensm se pueden actualizar aisladamente cumpliendo con la tercera forma normal.

ordensm → ordensm(fecha de recepción, fecha de entrega, estado de la ordensm)

La entidad SOLICITUD los atributos fecha de solicitud, cantidad de instrumentos en la solicitud se pueden actualizar por separado cumpliendo con la tercer forma normal.

clave de solicitud → solicitud(fecha de la solicitud, cantidad de instrumentos)

Finalmente en la entidad LABORATORIO todos los atributos pueden ser actualizados independientemente cumpliendo con la tercera forma normal.

clave del laboratorio → laboratorio( nombre de la institución, nombre de la sección, cuenta bancaria de depósito, nombre de la cuenta bancaria, telefono del laboratorio, fax del laboratorio)

*4FN: Una relación esta en 4FN si esta en tercera forma normal y todas las dependencias multivaluadas en la relación son de hecho dependencias multivaluadas.*

Ya que en ninguna de las entidades propuestas se tienen llaves compuestas, no es necesario eliminar las dependencias multivaluadas.

Debido a la complejidad del sistema no requiere mayor grado de normalización hasta aquí se detiene el proceso de normalización quedando el modelo Entidad/Relación como se muestra en la figura 2.4

## 2.3.3 MODELO ENTIDAD/RELACIÓN

Del diagrama Entidad/Relación del punto anterior se establece el modelo Entidad/Relación del sistema como se muestra en la siguiente figura:

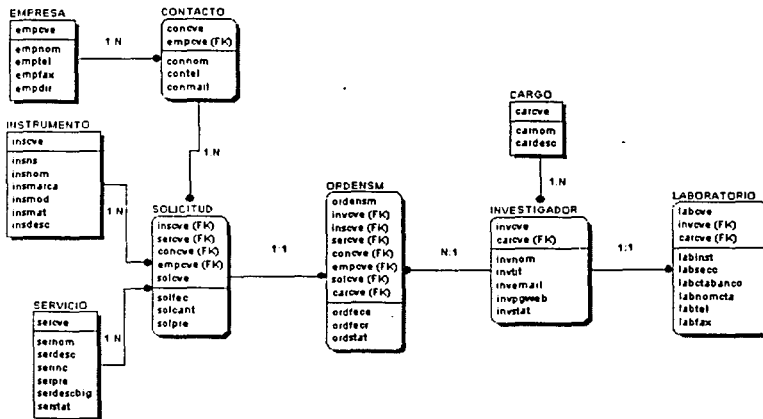


Figura. 2.4 MODELO ENTIDAD/RELACION

TESIS CON  
FALLA DE ORIGEN



## 2.3.4 DICCIONARIO DE DATOS

Para concluir con el modelado de la base de datos se formaliza la descripción de tablas y campos en las siguientes tablas del diccionario de datos.

Tabla	Descripción
CARGO	Esta tabla contiene información de los cargos y su descripción en el laboratorio de Metrología
CONTACTO	Esta tabla contiene información del representante de la empresa que solicita un servicio.
EMPRESA	Esta tabla contiene información de la empresa que solicita un servicio al laboratorio de metrología.
INSTRUMENTO	Esta tabla contiene información del instrumento al que se le va a realizar un servicio en el laboratorio de metrología.
INVESTIGADOR	Esta tabla contiene datos de los investigadores del laboratorio de metrología que realizan los servicios.
ORDENSM	Esta tabla contiene información de la orden sm.
SERVICIO	Esta tabla contiene información de los servicios que ofrece el laboratorio de metrología.
SOLICITUD	Esta tabla contiene información de las cotizaciones hechas desde internet.
LABORATORIO	Esta tabla contiene información acerca del laboratorio.

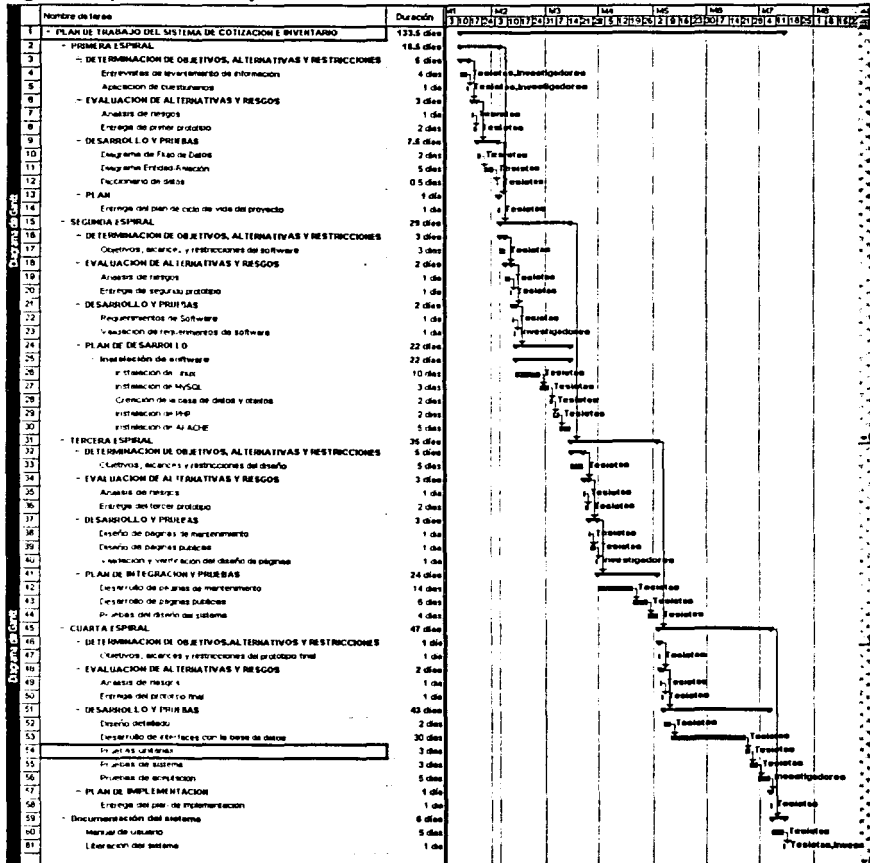
Tabla	Columna	Tipo de dato	Pk	Fk	Null	Descripción
CARGO	carcve	int	SI		NOT NULL	Clave del cargo
	carnom	varchar(60)				Nombre del cargo
	cargdesc	varchar(200)				Descripción del cargo
CONTACTO	Concve	int	SI		NOT NULL	Clave del contacto
	empcve	int		SI	NOT NULL	Clave de la empresa
	connom	varchar(60)				Nombre del contacto
	Coniel	varchar(20)				Teléfono del contacto
	conmail	varchar(50)				Email del contacto
EMPRESA	empcve	int	SI		NOT NULL	Clave de la empresa
	empnom	varchar(60)			NOT NULL	Nombre de la empresa
	Empitel	varchar(20)				Teléfono de la empresa
	empfax	varchar(20)				Fax de la empresa
	Empdir	varchar(200)			NOT NULL	Dirección de la empresa

TESIS CON  
FALLA DE ORIGEN

Tabla	Columna	Tipo de dato	Pk	Fk	Null	Descripción
INSTRUMENTO	inscve	Int	SI		NOT NULL	Clave del instrumento
	insns	int				Número de serie del instrumento
	insnom	Varchar(100)				Nombre del instrumento
	insmarca	Varchar(50)				Marca del instrumento
	insmod	Varchar(30)				Modelo del instrumento
	insmat	Varchar(100)				Material del instrumento
	insdesc	Varchar(200)				Descripción del instrumento con otras especificaciones
SOLICITUD	solcve	Int	SI		NOT NULL	Clave de la solicitud (llave primaria)
	inscve	Int		SI	NOT NULL	Clave del instrumento
	sercve	Int		SI	NOT NULL	Clave del servicio
	concve	Int		SI	NOT NULL	Clave del contacto
	solfec	Date				Fecha en que se realiza la solicitud
	solcant	Int				Cantidad de instrumentos de la solicitud
ORDENSM	ordensm	Int	SI		NOT NULL	Número de orden SM (llave primaria)
	solcve	Int		SI	NOT NULL	Clave de la solicitud
	ordfece	Date				Fecha de entrega del instrumento o conjunto de instrumentos relacionado con la orden SM
	ordfrecr	Date				Fecha de recepción del instrumento o conjunto de instrumentos relacionado con la orden SM
	ordstat	char(1)				Estado de la orden SM (terminado o no terminado)
SERVICIO	sercve	int	SI		NOT NULL	Clave del servicio
	sernom	varchar(60)			NOT NULL	Nombre del servicio
	serdesc	varchar(100)				Descripción corta del servicio
	serinc	float			NOT NULL	Incertidumbre del servicio
	serpre	float			NOT NULL	Precio del servicio
	serdescbig	varchar(300)				Descripción detallada del servicio
	serstat	char (1)				Estado del servicio (vigente o no vigente)
INVESTIGADOR	invce	int	SI		NOT NULL	Clave del investigador (llave primaria)
	carcve	int		SI	NOT NULL	Clave del cargo del investigador.
	invnom	varchar(60)				Nombre del investigador
	invlt	varchar(60)				Título del investigador
	invemail	varchar(50)				Email del investigador
	invpgweb	varchar(50)				Página web del investigador
	invstat	char(1)				Estado del investigador (vigente o no vigente)

### 2.3.5 PLANIFICACIÓN

Establecidos los requerimientos del Laboratorio de Metrología se elaboró el siguiente plan de trabajo.



## CAPITULO 3. IMPLEMENTACIÓN DEL SISTEMA

Actualmente el Laboratorio de Metrología cuenta con un computadora personal con las siguientes características:

- Procesador pentium II a 233 Mhz.
- 32 Mb de memoria RAM.
- Disco duro de 4 Gb.
- Tarjeta de red DAVICOM 9102 PCI FAST ETHERNET
- Monitor SuperVGA.
- Tarjeta de Sonido Sound Blaster 16.

En la propuesta de software a utilizar en el desarrollo del sistema se debió considerar:

- Software confiable y gratuito.
- Sistema operativo multiprocesos y multitareas.
- Consulta y actualización remota de información del sistema en tiempo real.
- Servidor de Web, y de bases de datos con acceso concurrente de 50 usuarios mínimo.
- Software de administración de base de datos relacional.
- Lenguaje de programación compatible con el servidor de Web y con el software de base de datos.
- Actualización y mantenimiento de software sencillo.

De acuerdo a las características del hardware disponible, y de los requerimientos se instaló el siguiente software:

- Sistema Operativo: LINUX
- Manejador de Base de Datos: MySql
- Lenguaje de Programación: PHP
- Servidor de Web: Apache

El software propuesto necesito un período de aprendizaje para el desarrollo del sistema ya que el sistema fue hecho a la medida y no se cuenta con código reutilizable, consecuencia de ello se considera un riesgo en las fechas de entrega.

En la figura 3.1 se muestra el entorno del sistema de cotización e inventario en un ambiente Web.

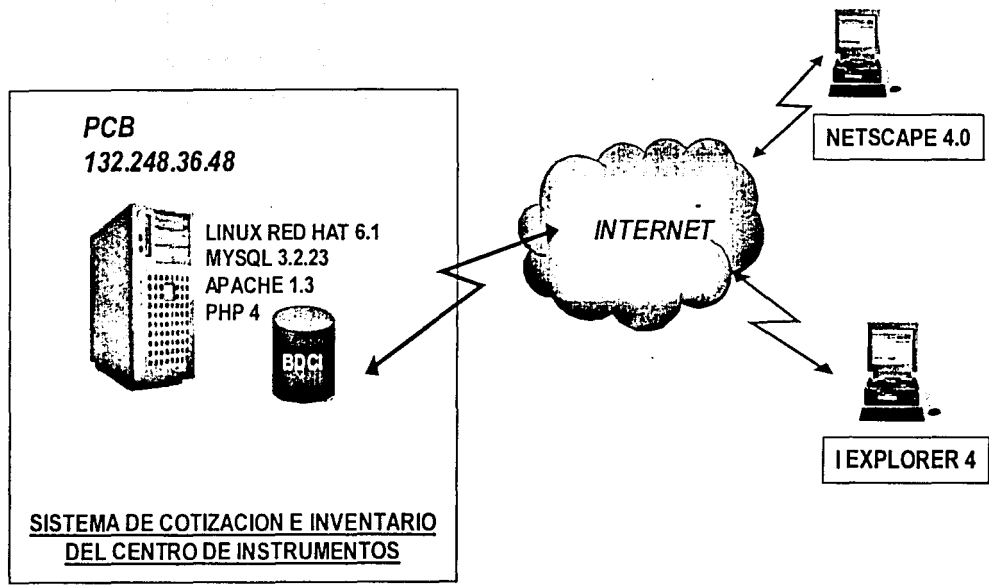


Figura 3.1 ENTORNO DEL SISTEMA DE COTIZACIÓN E INVENTARIO DEL CENTRO DE INSTRUMENTOS

### 3.1 IMPLANTACIÓN Y ADECUACIÓN DE LAS HERRAMIENTAS DE SOFTWARE

La implementación del sistema requiere de la instalación y adecuación del sistema operativo Linux, el servidor de Web Apache, el lenguaje de programación PHP y manejador de base de datos MySQL cumpliendo con los requerimientos del laboratorio.

#### 3.1.1 INSTALACIÓN DE LINUX.

Para realizar la instalación de linux, primero se tiene que conocer los requerimientos en hardware y hacer la evaluación correspondiente de la tabla 3.1:

<i>Disco(s) duro(s)</i> : Número, tamaño, tipo; p. ej: IDE hda=1.2G, IDE hdb=540MB	1) IDE hda=4 Gb
<i>Particiones</i> : correspondencia entre particiones y puntos de montaje; p. ej: /dev/hda1=/home, /dev/hda2=/	2) /dev/hda1=/mnt/winn /dev/hda2= SWAP /dev/hda3=/
<i>Memoria</i> : Cantidad de RAM instalada en su computadora; p. ej: 32MB, 64MB, 128MB	3) 32 Mb Compartida 2Mb
<i>CD-ROM</i> : Tipo de interfaz, por ejemplo: SCSI, IDE (ATAPI) u otra	4) IDE (ATAPI)
<i>Adaptador SCSI</i> : Si hay uno, marca y número de modelo; p. ej: BusLogic SCSI Adapter, Adaptec 2940UW	5) NO HAY
<i>Tarjeta de red</i> : Si hay una, marca y número de modelo; p. ej: Tulip, 3COM 3C590	6) DAVICOM 9102 PCI FAST ETHERNET ADAPTER
<i>Ratón</i> : Tipo, protocolo y número de botones; p. ej: ratón PS/2 genérico de 3 botones, ratón serie MouseMan de 2 botones	7) RATÓN GENÉRICO PS/2 3 BOTONES
<i>Monitor</i> : Marca, modelo y especificaciones del fabricante; p. ej: Optiquest Q53, ViewSonic G773	8) VGA GENÉRICO, 640 X 480 - 800 X 600 ppp, 60 hz.
<i>Tarjeta de video</i> : Marca, número de modelo y VRAM; p. ej: Creative Labs Graphics Blaster 3D, 8MB	9) SIS 620 REV 2ª, 2Mb
<i>Tarjeta de sonido</i> : Marca, conjunto de chips [chipset] y número de modelo; p. ej: S3 SonicVibes, Sound Blaster 32/64 AWE	10) Sound Blaster 16
<i>Dirección IP</i> : Cuatro números separados por puntos; p. ej: 10.0.2.15 (póngase en contacto con su administrador de red si necesita ayuda)	11) 132.248.36.48
<i>Máscara de red</i> : Normalmente cuatro números separados por puntos; p. ej: 255.255.248.0 (póngase en contacto con su administrador de red si necesita ayuda)	12) 255.255.255.0
<i>Dirección IP de la puerta de enlace (gateway)</i> : Cuatro números separados por puntos; p. ej: 10.0.2.245 (póngase en contacto con su administrador de red si necesita ayuda)	13) 132.248.36.254
<i>Una o más direcciones IP de servidores de nombres</i> : Normalmente uno o más conjuntos de números separados por puntos; p. ej: 10.0.2.1 (póngase en contacto con su administrador de red si necesita ayuda)	14) 132.248.10.2 132.248.204.1
<i>Nombre de dominio</i> : el nombre asignado a su organización; p. ej: El de RED HAT sería redhat.com (póngase en contacto con su administrador de red si necesita ayuda)	15) cinstrum.unam.mx
<i>Nombre de máquina</i> : el nombre de su ordenador; su elección personal de nombres, p. ej: galleta, parquesur.	16) pcb

Tabla 3.1 Requerimientos de Linux

La tabla anterior es de gran utilidad porque permitirá tener la información disponible durante el proceso de instalación, así como mantener un registro del sistema.

En base a los requerimientos de hardware y software con los que dispone el centro de instrumentos se eligió la distribución RED HAT Linux versión 6.1 ya que aunque existe en el mercado la versión 7.0 esta aún no se considera estable. RED HAT GNU GPL que se puede obtener de manera gratuita en la red, pero no se posee un respaldo ni soporte técnico si este se desea se puede comprar una versión que si lo brinda además de muchas utilerías provistas por ellos.

### Particiones de disco

Prácticamente todos los sistemas operativos modernos utilizan particiones de disco y Red Hat Linux no es una excepción. Para instalar Red Hat Linux será necesario trabajar con particiones de disco.

Al iniciar el proceso de instalación, se deben cumplir las siguientes condiciones.

La computadora debe tener suficiente espacio disponible en disco sin particionar para instalar Red Hat Linux.

La computadora debe tener una o más particiones que puedan ser borradas, liberando por tanto suficiente espacio en disco para instalar Red Hat Linux. Para realizar esto de una forma más sencilla se utilizó un programa llamado Partition Magic versión 5 que también esta disponible en Internet el cual permite reparticionar un disco sin la necesidad de darle formato o bien de perder la información que este tenga, de tal forma que el disco duro queda particionado como muestra la tabla 3.2:

Sistema Operativo	Partición (tamaño)	Sistema de Archivos	Asignación para Linux
Windows 98	2.6 Gb	Fat32	hda1
Linux native (reservado)	1.4 Gb	LINUXEXT2	hda3
Linux Swap (reservado)	100 Mb	Swap	hda2

Tabla 3.2 Partición de disco

Se necesitan dos particiones para este sistema operativo una para los archivos del sistema y una más para el manejo de la memoria virtual (swap) partición de uso exclusivo del sistema operativo, por lo que se tuvo que liberar espacio (ya que Windows 98 ocupaba todo el espacio en el disco duro) y crear las particiones.

## Inicio del programa de instalación

Para iniciar el proceso, en primer lugar es necesario arrancar el programa de instalación<sup>1</sup>. Este se encuentra en un CD-ROM (BOOTEABLE) apareciendo una pantalla que nos permitirá escoger el tipo de instalación entre el modo de texto o modo gráfico.

Se eligió la opción Modo Gráfico porque es más amigable y tiene ayuda en pantalla, el programa de instalación identificará el lector de CD-ROM mostrando la pantalla de selección de idioma como se muestra en la figura 3.1, el idioma seleccionado fue inglés ya que en muchos casos las traducciones no están completas.

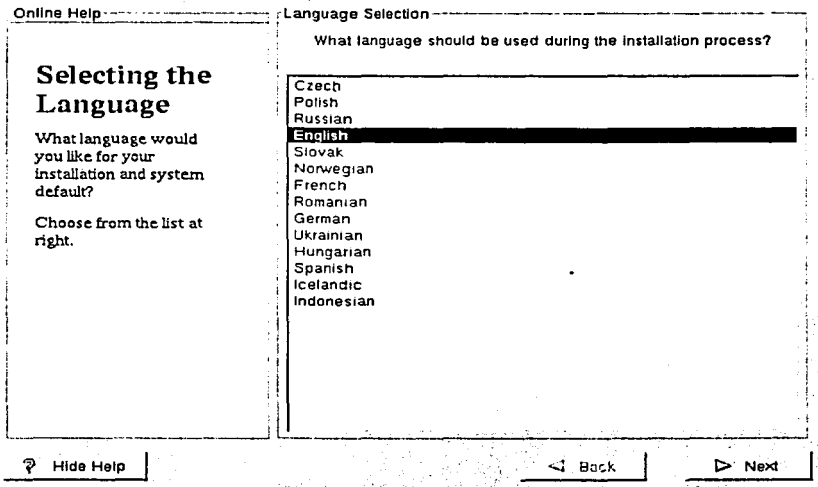


Figura 3.1 Selección de Idioma

<sup>1</sup> Previamente se verificó la compatibilidad de nuestro equipo con los requerimientos de esta distribución en la página <http://www.redhat.com/corp/support/hardware/index.html>



Después de seleccionar el idioma se debe elegir el tipo de teclado genérico de PC de 104 teclas [Generic 104-key PC]) como se muestra en la figura 3.2.

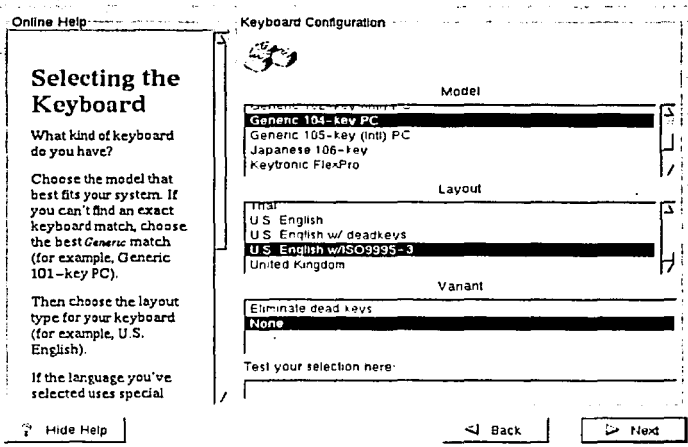


Figura 3.2 Selección del tipo de teclado

Seleccionado el teclado, se elige el tipo de ratón a una de las entradas genéricas que basado en el número de botones del ratón como se muestra en la figura 3.3.

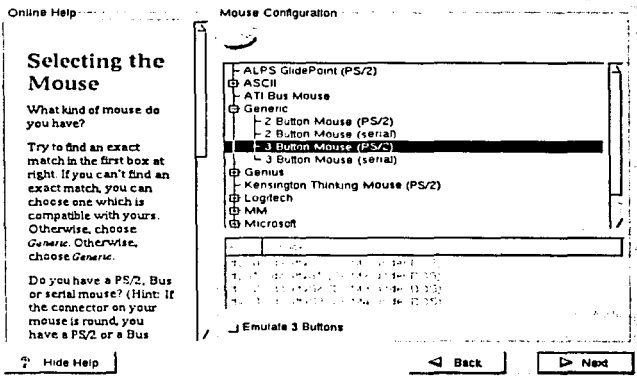


Figura 3.3. Selección del ratón

Después de elegir el ratón aparecerá la pantalla de «Bienvenida» que se muestra en la figura 3.4

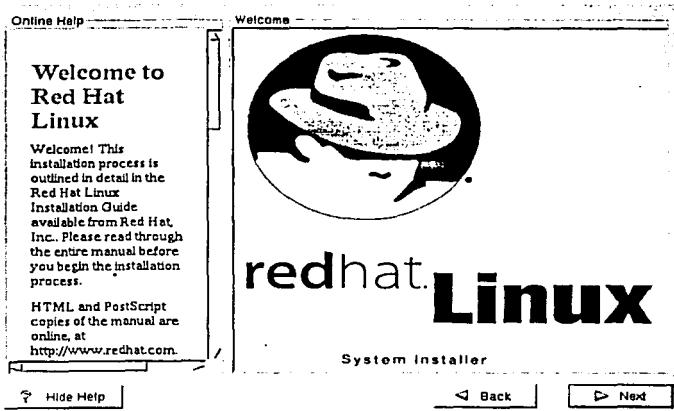


Figura 3.4 Bienvenido a Red Hat Linux

Después de la pantalla de bienvenida se debe elegir si se desea realizar una instalación completa o una actualización como muestra la figura 3.5

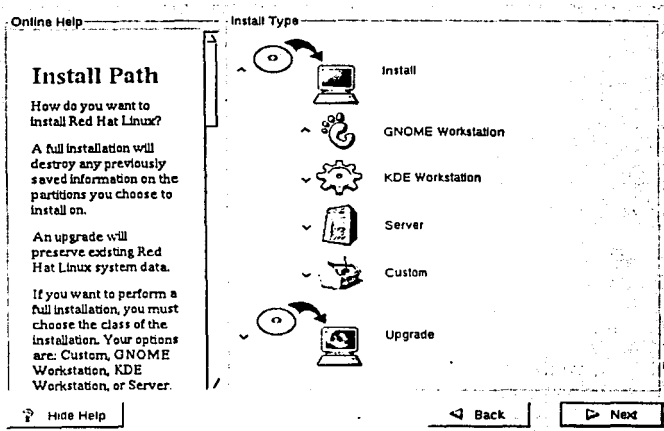


Figura. 3.5. Elección entre instalación y actualización

Red Hat Linux incluye tres tipos o clases diferentes de instalaciones:

- Estación de trabajo
- Servidor
- Personalizada

La instalación de tipo personalizada permite obtener la máxima flexibilidad. Los tipos estación de trabajo y servidor tienen ciertas configuraciones predeterminadas toman a medida que avanza la instalación.

Se decidió usar una instalación personalizada porque durante esta instalación, se puede decidir cómo se particionará el espacio en disco y se tendrá control completo sobre los paquetes que serán instalados en el sistema. También se puede determinar si se utilizará LILO para arrancar el sistema..

En este punto, es necesario informar al programa de instalación sobre el sitio donde se debe instalar Red Hat Linux. Esto se hace definiendo puntos de montaje para una o más particiones de disco en las cuales se instalará Red Hat Linux como se muestra en la figura 3.6.

The screenshot shows the Disk Druid interface. On the left is an 'Online Help' window titled 'Partitions' with instructions on how to define mount points and create/delete partitions. The main window displays a table of partitions and a 'Drive Summary' table.

**Partitions Table:**

Mount Point	Device	Requested	Actual	Type
<not set>	hda1	15M	15M	Linux native
<Swap>	hda5	86M	86,1	Linux swap
<not set>	hda6	4047M	4047M	Linux native
<not set>	hda7	2000M	2000M	Linux native

**Drive Summary Table:**

Drive	Geom [C/H/S]	Total (M)	Free (M)	Used (M)	Used (%)
hda	[ 784/255/83]	6149M	0M	6149M	100%

Buttons at the bottom include 'Add...', 'Edit...', 'Delete', 'Reset', 'Make RAW Device', 'Back', and 'Next'.

Figura 3.6 Particionado de su sistema

Cada línea de la sección Partitions (Particiones) de la figura 3.6 representa una partición de disco. Cada línea en esta sección tiene cinco campos diferentes:

**Punto de montaje:** Este campo indica el punto donde se montará la partición. Si la partición existe, pero «no está asignada» se tendrá que definir su punto de montaje. Esto se puede hacer seleccionando la línea correspondiente y usando el botón Editar.

Al crear las particiones deben de seguirse las siguientes recomendaciones:

**Una partición swap (al menos 16MB):** Las particiones swap se utilizan para implementar memoria virtual. Si el equipo tiene 16 Mb de RAM, o menos, *debe* crear la partición swap. Incluso si se tiene más memoria, la partición swap es recomendable. El tamaño mínimo de su partición swap debe ser igual o mayor a la RAM de su computadora (máximo 120 Mb) o 16 Mb (lo que resulte ser mayor).

**Una partición /boot (de 16MB como máximo):** La partición que se monte en /boot contiene el núcleo del sistema operativo, además de ficheros usados durante el proceso de arranque. Debido a limitaciones de la mayoría de las BIOS de los PCS, es una buena idea crear una pequeña partición para estos ficheros, que no debería ser mayor de 16 Mb.

**Una partición root (600MB-1.5GB):** Que es donde se almacena «/» (el directorio raíz). En esta configuración, todos los ficheros (excepto los almacenados en /boot) residen en la partición raíz. Una partición raíz de 600MB permitirá el equivalente a una instalación de tipo estación de trabajo (con *muy* poco espacio libre), mientras que una partición raíz de 1.5GB le permitirá instalar todos los paquetes.

**Dispositivo:** Este campo muestra el nombre de dispositivo de la partición en Linux.

**Solicitado:** El campo «Solicitado» muestra el tamaño mínimo solicitado cuando se definió la partición.

**Actual:** El campo «Actual» muestra el espacio actualmente asignado a la partición.

**Tipo:** Este campo muestra el tipo de la partición, por ejemplo Linux Nativo (Linux Native) o DOS.

Cada línea en la sección de Device Summary (Resumen de Discos) de la figura 3.6 representa un disco duro del sistema. Cada línea tiene los siguientes campos:

**Disco :** Este campo muestra el nombre de dispositivo del disco duro.

**Geom (C/H/S):** Este campo muestra la geometría del disco duro. La geometría consiste en tres números representando el número de cilindros, cabezas y sectores según la información dada por el disco duro.

**Total:** El campo «Total» muestra el espacio total disponible en el disco duro.

**Libre:** El campo «Libre» muestra la cantidad de espacio en el disco duro que está aún sin asignar.

**Usado:** Este campo muestra la cantidad de espacio del disco duro que está actualmente asignado a particiones en megabytes y porcentaje.

En el equipo del Laboratorio de Metrología se crearon con anterioridad dos particiones una para root (LINUX NATIVE DE 1.4 Gb) y otra para el swap (100 Mb), la partición para el boot no es necesaria ya que el boot estará en la partición donde está Windows 98, como las particiones ya están creadas solo hará falta asignar los puntos de montaje como se habían reservado y de acuerdo a la tabla de especificaciones.

A continuación todas las particiones recién creadas y asignadas deberán ser formateadas como se muestra figura 3.7, con excepción de la partición donde se encuentra Windows 98 ya que se perdería la información que en ella se encuentra.

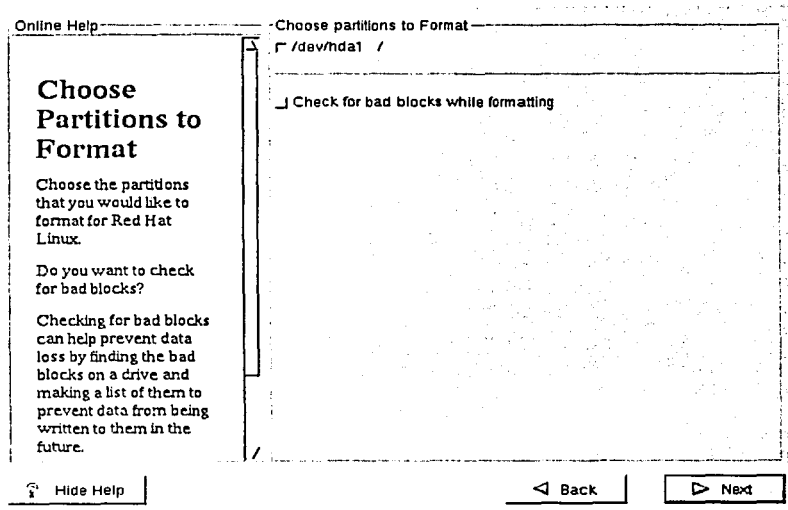


Figura 3.7 Elección de particiones a formatear

Si se desea comprobar si hay bloques dañados mientras formatea cada sistema de ficheros, habrá que seleccionar la opción Verificar bloques defectuosos.

La comprobación de bloques dañados puede prevenir la pérdida de datos localizando los bloques dañados de una unidad y haciendo una lista con ellos para impedir su uso en el futuro.

Para poder arrancar el sistema con Red Hat Linux normalmente se necesita instalar LILO (El cargador de Linux [the Linux Loader]). El LILO debe ser instalado en uno de los dos sitios:

#### El sector de arranque (MBR)

El sitio recomendado para instalar LILO, a menos que el MBR ya ejecute otro cargador de sistema operativo, como System Commander o el de OS/2. El sector de arranque es un área especial de su disco duro, que se carga automáticamente por la BIOS del equipo, y es el primer punto donde LILO puede tomar el control del proceso de arranque. Si instala LILO en el MBR, cuando su máquina arranque, LILO le mostrará un inductor de órdenes boot: . En ese momento se podrá arrancar Red Hat Linux o cualquier otro sistema operativo para el que LILO esté configurado para arrancar.

#### El primer sector de la partición raíz

Recomendado si ya está utilizando otro gestor de arranque en su sistema (como el de OS/2 o BOOTMANAGER). En este caso el otro cargador tomará el control primero. Entonces puede configurar el cargador de arranque para que ejecute LILO (que arrancará Red Hat Linux a continuación).

En sistemas con Win95/98, también se debe instalar LILO en el MBR para que LILO pueda arrancar ambos sistemas operativos como muestra la figura 3.8.

Si se tiene Windows NT (y se quiere instalar LILO) se debe elegir instalar LILO en el primer sector de la partición raíz, no en el MBR. Habrá de crearse un disco de arranque. En un caso como este, El cargador del sistema operativo Windows NT estará en el MBR y la única manera de elegir LILO (en el primer sector de su partición raíz) es utilizando el disco de arranque.

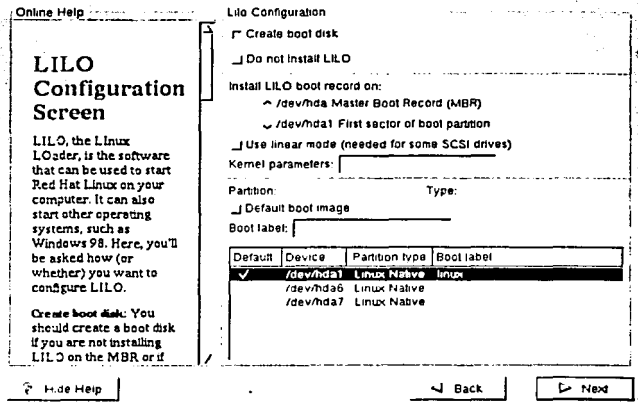


Figura 3.8 Pantalla de configuración de LILO

Si se desean añadir opciones predeterminadas al programa de arranque LILO, se pueden introducir en el campo de los parámetros del núcleo. Cualquier opción que se introduzca se le pasará al núcleo de Linux cada vez que arranque.

**Partición arrancable:** Todas las particiones arrancables son listadas, incluyendo las particiones utilizadas por otros sistemas operativos. La columna «etiqueta» será rellenada con la palabra linux en la partición que contiene el sistema de ficheros raíz de Red Hat Linux. Las otras particiones también pueden tener etiquetas de arranque. De esta forma se puede agregar una etiqueta de arranque para el otro sistema operativo instalado en nuestro equipo.

### Configuración de LILO

**Crear disquete de inicio:** La opción Crear disquete de inicio está siempre marcada al principio. Es muy conveniente disponer de él, por las siguientes razones:

Para utilizarlo en vez de LILO: Puede utilizar el disco de arranque en vez de LILO. Es muy conveniente si se prefiere no tocar el arranque del otro sistema operativo. Con un disco de arranque, volver al otro sistema operativo es tan fácil como quitar el disco de arranque y reiniciar la máquina.

<sup>2</sup> Si se olvida las etiquetas definidas en el sistema, se puede pulsar **tab**, en el inductor de órdenes **boot**: de LILO, para que se muestre en pantalla una lista de todas las etiquetas definidas.

Para utilizarlo en caso de emergencia: El disco de arranque puede ser utilizado en el modo de rescate (rescue mode), proporcionando las herramientas necesarias para volver a tener el sistema funcionando correctamente.

Para utilizarlo si otro sistema operativo sobrescribe LILO: Otro sistema operativo puede que no sea tan flexible en el arranque como Red Hat Linux. A menudo, al instalar o actualizar otro sistema operativo el sector de arranque (que en un principio contenía LILO) puede ser sobrescrito, haciendo imposible el arranque de la instalación de Red Hat Linux. El disco de arranque puede ser utilizado para arrancar Red Hat Linux y poder reinstalar LILO.

Cuando la configuración de LILO es terminada se debe configurar la red como se muestra en la figura 3.9.

Online Help

## Network Configuration

If you have a network card, you can set up your networking information. Otherwise, click Next to proceed.

Choose your device type and whether you would like to configure using DHCP. If you have multiple ethernet devices, each device screen will keep the information you have given to it. You can switch between device screens, for example eth0 and eth1: the

eth0

Configure using DHCP

Activate on boot

IP Address: \_\_\_\_\_

Netmask: \_\_\_\_\_

Network: \_\_\_\_\_

Broadcast: \_\_\_\_\_

Hostname: \_\_\_\_\_

Gateway: \_\_\_\_\_

Primary DNS: \_\_\_\_\_

Secondary DNS: \_\_\_\_\_

Tertiary DNS: \_\_\_\_\_

Hide Help

◀ Back

▶ Next

Figura 3.9 Pantalla de configuración de red

Habrà que elegir el dispositivo correcto y si se desea o no configurarlo utilizando DHCP. Si se tienen varios dispositivos ethernet, cada uno mantendrã la informaci3n que se le haya proporcionado. Se puede cambiar de dispositivo, por ejemplo eth0 y eth1, y la informaci3n que se le d3 serã especìfica para cada dispositivo. Si se selecciona *Activar al iniciar*, la interfaz de red se cargarã en el arranque.



Después se introduce, los números de Dirección IP, Máscara de red, Red, Broadcast y nombre del equipo.

Finalmente, agregar las direcciones de la Puerta de enlace (gateway), DNS primario, DNS secundario y DNS terciario

Se debe asignar la zona horaria seleccionando la ubicación física del equipo o especificando la diferencia en tiempo con la Hora Universal Coordinada (Universal Coordinated Time), también conocida como UTC como se muestra en la figura 3.10.

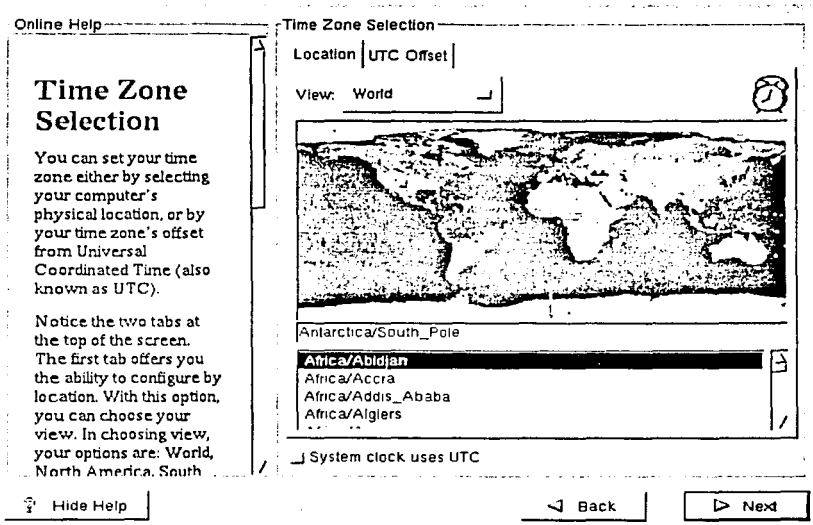


Figura 3.10. Configuración de la zona horaria

Se puede pulsar en una ciudad específica del mapa interactivo, como indican los puntos amarillos; aparecerá una X roja para indicar la selección. También puede desplazarse a lo largo de la lista y seleccionar la zona deseada.

La segunda pestaña ofrece la posibilidad de utilizar la diferencia respecto a la UTC. Se presentará una lista de desplazamientos respecto a la UTC entre los que se puede elegir, además de la opción de configurar el modo de ahorro diurno (Daylight Savings Time).

En la pantalla de configuración de cuentas de usuario permitirá asignar la contraseña del superusuario(root). Además también se pueden crear cuentas de usuario para entrar en el sistema una vez que acabe el proceso de instalación como lo muestra la figura 3.11.

Online Help

## Account Configuration

Enter a root password. The password must be at least six characters in length. Confirm the password. The "Next" button will become enabled once both entry fields match.

Now create a user account.

Enter a user account name. Then, create a password for that user account and confirm it. Finally, enter the full name of the account.

Account Configuration

Root Password:

Confirm:

Account Name:

Password:  Password (confirm):

Full Name:

Add Edit Delete New

Account Name	Full Name

? Hide Help

◀ Back

▶ Next

Figura 3.11 Creación de cuentas de usuario

El programa de instalación pedirá que se introduzca la contraseña del superusuario del sistema. Habrá que utilizar la palabra clave del superusuario para entrar por primera vez en el sistema Red Hat Linux.

La contraseña del superusuario debe tener al menos 6 caracteres de longitud y no es visible en la pantalla mientras se escribe. Habrá que introducirla dos veces y si las dos veces no coinciden, el programa de instalación pedirá que se introduzca de nuevo.

Como medida de seguridad hay que anotar la contraseña de root y guardarla en un sitio seguro.

Si se elige crear una cuenta de usuario en este momento, se dispondrá de una cuenta para entrar una vez que acabe la instalación. Esto permitirá entrar en el equipo de forma fácil y segura, sin tener que hacerlo como superusuario para crear otras cuentas.

Hay que introducir el nombre de la cuenta de usuario y a continuación introducir y confirmar la contraseña deseada.

También se pueden *Editar* o *Eliminar* cuentas de usuario que se hayan creado y ya no se quieran.

A menos que se vaya a configurar autenticación usando NIS, observará que ambas contraseñas MD5 y oculta (shadow) están seleccionadas como se muestra en la Figura 3.12. Se recomienda que se utilicen ambas opciones para hacer el equipo lo más seguro posible.

Para configurar la opción NIS, se debe estar conectado a una red NIS.

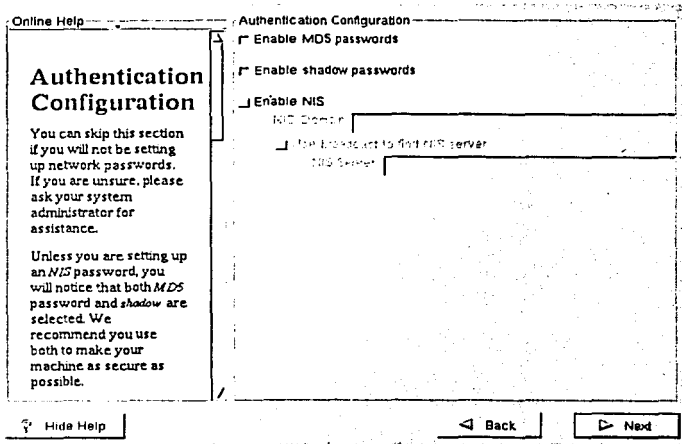


Figura 3.12. Configuración de autenticación

**Contraseña MD5:** Permite utilizar una palabra clave larga (hasta 256 caracteres) en vez de la estándar de 8 caracteres o menos.

**Contraseña oculta (shadow password):** Proporciona un método seguro para almacenar las contraseñas, que serán almacenadas en `/etc/shadow`, que sólo puede ser leído por el superusuario.

**Habilitar NIS:** Permite utilizar un grupo de computadoras en el mismo dominio del Servicio de Información en Red [Network Information Service] (NIS) con una contraseña y un grupo común. Hay dos opciones entre las que se puede elegir:

**Dominio NIS:** Esta opción le permite especificar a qué dominio o grupo de computadoras se pertenece.

**Servidor NIS:** Esta opción hace que el equipo utilice un servidor NIS específico, en vez de difundir un mensaje a la red de área local preguntando por algún servidor disponible para el sistema.

Después de haber seleccionado y configurado el formateo de las particiones, se está listo para seleccionar los paquetes que se desean instalar.

Se pueden seleccionar componentes que agrupan paquetes según su función, (por ejemplo, Desarrollo en C, Estación de trabajo en red, o Servidor Web), paquetes individuales, o una combinación de ambos.

Para seleccionar un componente, hay que pulsar en la casilla de verificación que está a su lado como se muestra en la figura 3.13.

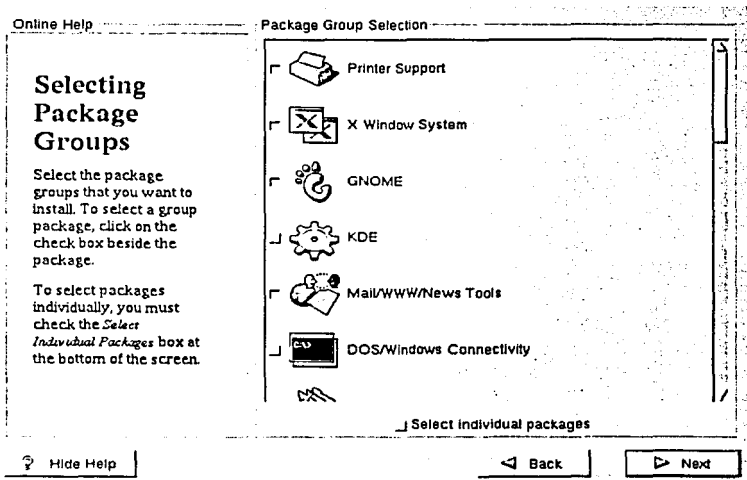


Figura 3.13 Selección de grupos de paquetes

Seleccionando Todo (esta al final de la lista de componentes) instala todos los paquetes incluidos en Red Hat Linux, lo que necesita cerca de 1.5GB de espacio libre.

Para seleccionar paquetes individualmente, hay que marcar la casilla **Seleccionar paquetes individualmente** en la parte inferior de la pantalla.

Después de seleccionar los componentes que desean instalar, se puede seleccionar o deseleccionar paquetes individuales. El programa de instalación le presentará una lista de los paquetes en ese grupo, que se puede seleccionar o deseleccionar utilizando el ratón como se muestra en la figura 3.14.

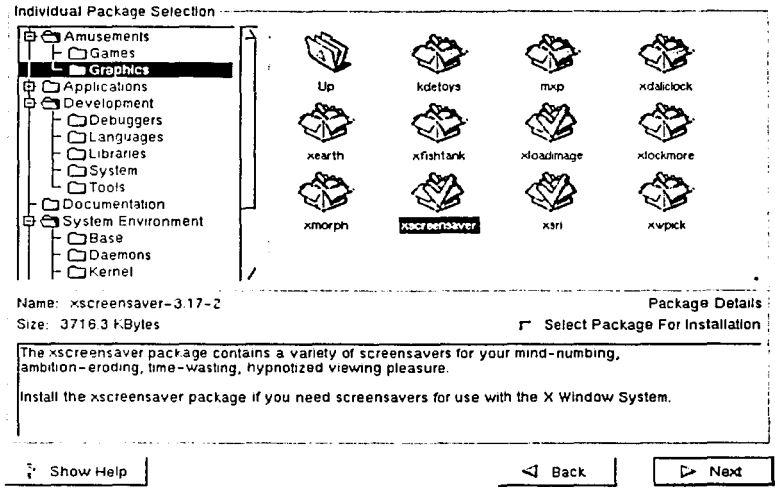


Figura 3.14 Selección de paquetes individuales

En la parte izquierda de la pantalla se ve un listado de los diferentes grupos de paquetes. Cuando se expande dicho listado (pulsando dos veces para seleccionarlo) y se pulse dos veces en cada directorio, el listado de los paquetes disponibles para su instalación aparecerá a la derecha.

Para seleccionar un paquete individual, hay que pulsar dos veces sobre él con el botón izquierdo del ratón, o pulsar una vez para resaltarlo, y pulsar en el botón **Seleccionar paquetes** de la parte inferior. Una marca verde de verificación aparecerá en todos los paquetes que se hayan seleccionado para la instalación.

Se puede leer información acerca de un determinado paquete antes de elegirlo para su instalación, pulsando una vez sobre él con el botón izquierdo y aparecerá la información sobre él en la parte inferior de la pantalla, con el nombre y el tamaño del mismo.

Algunos de los paquetes instalados en el sistema fueron:

- Kernel del sistema
- Servidor Web
- Servidor FTP
- Herramientas Gráficas
- Perl
- Xwindows

No se instalaron todos los paquetes que ofrece este sistema operativo, dadas las limitaciones de espacio en disco y de rendimiento en el equipo, además de que no todos son útiles para la implementación del sistema.

Muchos de los paquetes software dependen de otros para funcionar correctamente. Por ejemplo, muchas de las herramientas gráficas de administración del sistema de Red Hat requieren los paquetes *python* y *pythonlib*. Para asegurar que el sistema tiene todos los paquetes que necesita para ser completamente funcional, Red Hat Linux comprueba las dependencias de estos paquetes cada vez que instala o elimina paquetes software.

Si cualquier paquete necesita otro que no ha sido seleccionado para instalar, el programa presenta una lista de estas dependencias sin resolver y da la oportunidad de resolverlas como se muestra en la figura 3.15.

La pantalla de ***Dependencias no resueltas*** sólo aparecerá si falta algún paquete que necesitan los paquetes seleccionados. Debajo de la lista de paquetes que faltan, hay una casilla de verificación ***Instalar paquetes para resolver dependencias*** en la parte inferior de la pantalla y que está seleccionada en principio. Si se deja esa opción seleccionada, el programa de instalación resolverá las dependencias automáticamente añadiendo todos los paquetes necesarios a la lista de paquetes seleccionados.

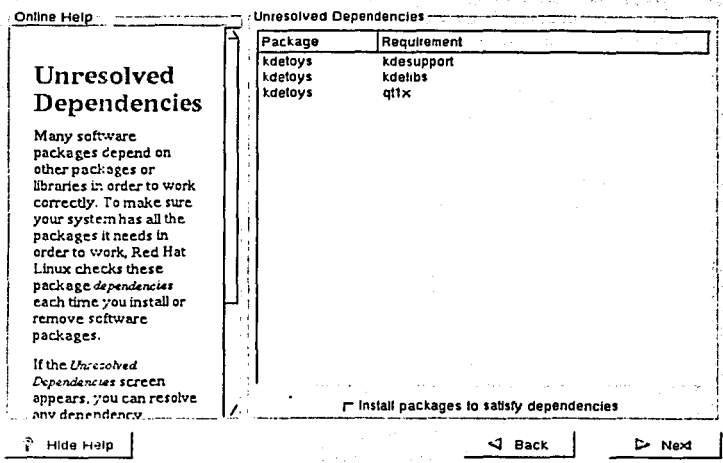


Figura 3.15 Dependencias sin resolver

A continuación se debe configurar las opciones del sistema Xwindows

*Xconfigurator* prueba primero el sistema e intenta determinar qué tipo de tarjeta de video se tiene. Si no es capaz de hacerlo, presentará una lista de tarjetas de video y monitores para que se haga la selección.

Después, *Xconfigurator* pregunta la cantidad de memoria de vídeo instalada en la tarjeta de video.

*Xconfigurator* muestra una lista de los disponibles chip de reloj para la tarjeta. Se recomienda utilizar la opción *No configurar chip de reloj [No Clockchip Setting]*, ya que Xfree86 puede detectar automáticamente el chip de reloj en la mayor parte de los casos.

Como el monitor que tenemos no aparece en la lista, seleccionamos *Personalizado (Custom)*. *Xconfigurator* pedirá que se seleccione el rango de sincronización horizontal y vertical del monitor<sup>3</sup>.

Una vez determinado el hardware, procedemos a comprobar la configuración. Hay que asegurarse de que la resolución y el color es el que se desea para trabajar.

<sup>3</sup> Es conveniente no sobrepasar los parámetros de funcionamiento del monitor, ya que es posible que se pueda sobrecargar y dañarlo o destruirlo.

Si se desea personalizar la configuración del sistema Xwindows, asegúrese de que el botón *Personalizar configuración de Xwindows* esté seleccionado. Si se elige personalizar, se mostrará otra pantalla que permite elegir cuantos colores se quieren usar (256, 65536 o 16 millones) y cual será la resolución deseada. De nuevo, se tendrá la opción de comprobar la configuración.

También se puede elegir *No configurar Xwindows* si se prefiere configurar el sistema Xwindows después de la instalación o nunca.

En este punto no hay mucho más que se pueda hacer hasta que se instalen todos los paquetes como se muestra en la figura 3.16. La velocidad a la que se realice depende del número de paquetes que se hayan seleccionado y de la velocidad del equipo.

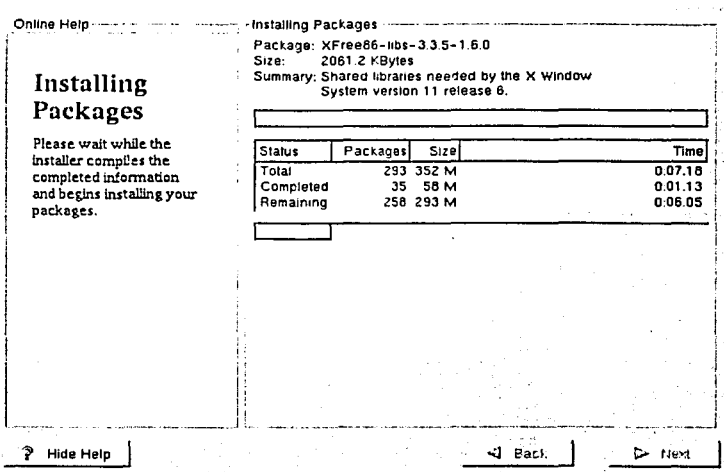


Figura 3.16 Instalación de paquetes

Si se elige crear un disco de arranque, en este momento hay que insertar un disco vacío en la unidad de disco de 3 1/2" como se muestra en la figura 3.17.

Tras un corto espacio de tiempo, se creará un disco de arranque. Después de retirarlo de la unidad de disco, debe ser etiquetado para evitar perderlo.



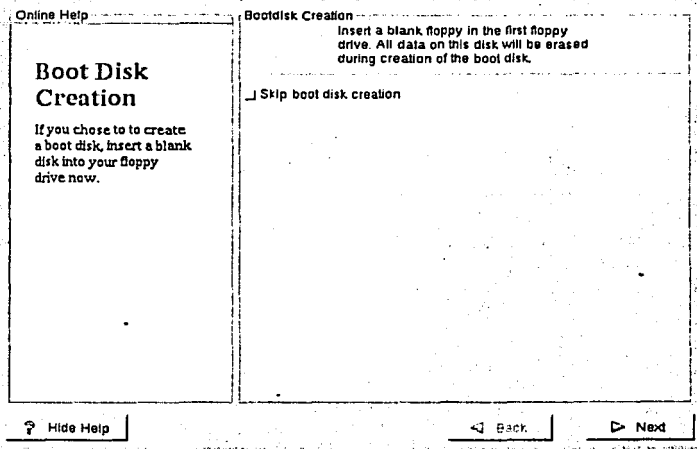


Figura 3.17 Creación de un disco de arranque

Después de crear el disco de arranque la instalación de Red Hat Linux 6.1 está completa.

El programa de instalación pedirá reiniciar el sistema, y habrá que quitar el CD-ROM de su unidad.

Después de la secuencia de arranque normal del equipo, se verá el inductor de órdenes estándar de LILO, que es *boot:*. En el inductor de órdenes *boot:*, se puede hacer una de las siguientes cosas:

Presionar *Intro*: Hace que se arranque la entrada predeterminada en LILO, para el sistema se predeterminó Windows 98.

Introducir una etiqueta seguida por *Intro*: Hace que LILO arranque el sistema operativo correspondiente a esa etiqueta, por ejemplo: *linux*, es la etiqueta para que nuestro equipo entre con este sistema operativo.

No hacer nada: Después del tiempo correspondiente, configurado en LILO (el valor predeterminado son 5 segundos), éste arrancará automáticamente la entrada predeterminada (Windows 98).

Después de hacer lo que sea necesario para arrancar Red Hat Linux, se verán una o más pantallas de mensajes. Finalmente, aparecerá la pantalla de login de la interfaz gráfica de usuario (GUI) .

### 3.1.2 INSTALACIÓN DE APACHE.

El software de Apache esta disponible en Internet en la página <http://www.apache.org/dist/httpd>.

Antes de iniciar la instalación de Apache es necesario validar que se cumplan los requerimientos de instalación de la tabla 3.3:

Requerimientos de hardware	Requerimientos de software
Aproximadamente 12 Mb de espacio disponible en el directorio temporal.  Como mínimo 3 Mb disponibles en disco duro para el software de apache el tamaño ocupado dependerá de los módulos instalados.	Compilador ANSI-C.  Sistema operativo: - LINUX - Solaris - AIX - IRIX - HP-UX - DGUX

Tabla 3.3 Requerimientos de Apache

Cuando se obtiene el software de apache, éste se encuentra compactado, por lo que debe descompactarse con las siguientes instrucciones:

```
$ gunzip apache_1.3.12.tar.gz
$ gunzip apache_1.3.12.tar
```

Una vez descompactado el software se debe ir al subdirectorio `apache_1.3.12`

```
$ cd apache_1.3.12
```

El siguiente paso es configurar la raíz del árbol de Apache, esto es muy importante porque es el prefijo en donde Apache será instalado posteriormente para trabajar correctamente:

```
$ ./configure --prefix=/www
```

A continuación debe compilarse el paquete de Apache:

\$ make

Finalmente se debe instalar Apache:

\$ make install

Para iniciar el servidor apache se utiliza el siguiente comando:

\$ www/bin/apachectl start

Iniciado el servidor Apache desde cualquier browser se podrá ver como muestra la figura 3.18:

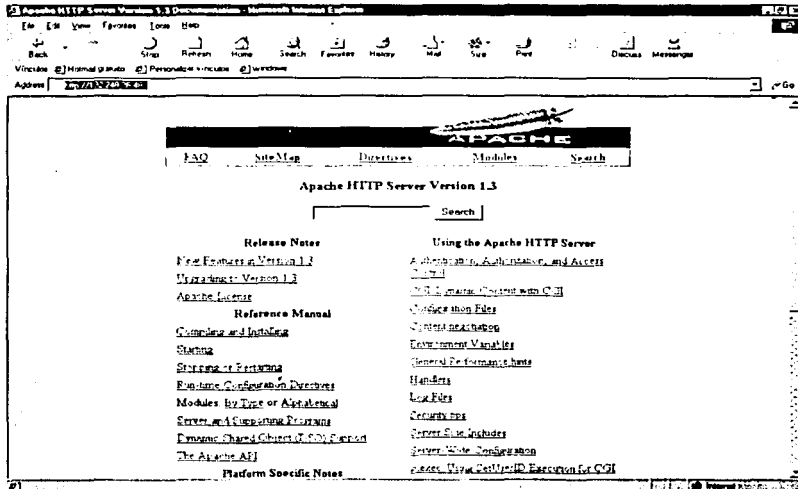


Figura 3.18 Servidor Apache desde un browser

### 3.1.3 INSTALACIÓN DE MYSQL.

En este apartado se dará una breve descripción de la instalación de *MySQL* en LINUX como parte de la implementación del sistema.

*MySQL* es un Relational Database Management System (RDBMS) gratis con licencia publica, la versión para el sistema operativo LINUX esta disponible en la pagina <http://www.mysql.com/Downloads/Linux>.

La mejor alternativa de instalación en LINUX es usar la distribución RPM (RED HAT Package Manager).

Los archivos incluidos en la distribución de RPM son:

MySQL-3.23.47-1.i386.rpm: Contiene el Server de *MySQL*.

MySQL-client-3.23.47-1.i386.rpm: Contiene los programas estándar de cliente de *MySQL*.

MySQL-bench-3.23.47-1.i386.rpm: Pruebas y comparativos.

MySQL-devel-3.23.47-1.i386.rpm: Contiene las librerías e incluye los archivos necesarios para compilar otro tipo de clientes de *MySQL*, como módulos de Perl.

MySQL-3.23.47-1.src.rpm: Este contiene el código fuente para todos los paquetes anteriores.

Desde la línea de comando de linux se debe ejecutar el siguiente comando para instalar el software del servidor y del cliente.

```
$ rpm -i MySQL-3.23.47-1.i386.rpm MySQL-client-3.23.47-1.i386.rpm
```

El comando anterior pondrá datos en `'/var/lib/mysql'`, también pondrá las líneas necesarias en el archivo `'/etc/rc.d'` de linux para que inicie el servidor de *MySQL* automáticamente al encender el servidor de LINUX.

Para comprobar que **MySQL** fue instalado correctamente desde la línea de comandos de Linux:

```
$ mysql/mysqlshow
+-----+
| Databases |
+-----+
| mysql    |
+-----+
```

El comando anterior muestra las bases de datos de **MySQL**.

El siguiente comando muestra las tablas que contiene la base de datos **MySQL**.

```
$ mysql/mysqlshow mysql
Database: mysql
+-----+
| Tables |
+-----+
| columns_priv |
| db           |
| func        |
| host        |
| tables_priv |
| user        |
+-----+
```

### 3.1.4 INSTALACIÓN DEL LENGUAJE DE PROGRAMACIÓN PHP4

Como ya se menciona en el capítulo uno PHP es un lenguaje interpretado, no compilado orientado a la creación de páginas Web dinámicas. Se integra dentro del código HTML y en él un intérprete se encarga de leer y ejecutar el código antes de ser enviado, originando como resultado la página HTML que se envía finalmente al navegador.

Trabajar con PHP incluye tratar con formularios, distinguir navegadores, hacer funciones para evitar la duplicidad de código, consultar bases de datos de todo tipo.

PHP se anexa como un módulo del servidor de APACHE y se puede obtener de <http://www.php.net/>

Cuando se obtiene el paquete de PHP esta compactado con gzip y se debe descompactar con la siguiente instrucción

```
$ gunzip php-4.1.1.tar.gz
```

El siguiente paso es descompactarlo del formato tar:

```
$ tar xvf php-4.1.1.tar
```

Desgraciadamente no hay forma de comprobar que PHP esta funcionando hasta que se integra como un módulo al servidor Apache como se verá en la siguiente sección.

### 3.1.5 INTEGRACIÓN DE APACHE-MYSQL-PHP4

Para lograr que *MySQL* y PHP sean reconocidos por el servidor de APACHE es necesario anexar los módulos y modificar el archivo de configuración:

Cambiarse al directorio de php:

```
$ cd ../php-4.1.1
```

Configurar el modulo de *MySQL* en Apache:

```
$ ./configure --with-mysql --with-apache=../apache_1.3.12 --enable-track-vars
```

```
$ make
```

```
$ make install
```

Configurar el modulo de PHP en Apache:

```
shell> cd ../apache_1.3.12
```

```
shell> ./configure --activate-module=src/modules/php4/libphp4.a
```

```
shell> make
```

```
shell> make install
```

```
shell> cd ../php-4.1.1
```

```
shell> cp php.ini-dist /usr/local/lib/php.ini
```

Editar el httpd.conf y agregar:

```
For PHP 4: AddType application/x-httpd-php.php
```

Una vez terminada la configuración se inicia el servidor, simplemente con el siguiente comando:

```
Shell> start httpd
```

En la sección anterior se preparo la infraestructura tecnológica en la que se va apoyar el sistema de cotización e inventario del Laboratorio de Metrología, en este apartado se va a explotar dicha infraestructura para implementar el sistema.

Inicialmente debe crearse la base de datos BDCI (Base de Datos del Centro de Instrumentos) con el siguiente comando:

```
$ mysqladmin create BDCI
```

Una vez creada la base de datos BDCI se deben crear las tablas del sistema, a modo de ejemplo se muestra la creación de la tabla EMPRESA.

Primero se debe establecer una conexión a la base de datos BDCI:

```
$mysql BDCI
```

Establecida la conexión se ejecuta una sentencia SQL para la creación de la tabla EMPRESA:

```
$mysql> CREATE TABLE EMPRESA (empcve int NOT NULL AUTO_INCREMENT, empnom varchar(60), emptel varchar(20), empfax varchar(20), empdir varchar(200), KEY (empcve) )g
```

La creación de las demás tablas se hizo en base al diccionario de datos del capítulo dos.

Creadas las tablas en la base de datos se desarrollan las páginas HTML del sistema con código PHP incrustado, haciendo consultas en línea de la base de datos, en el Anexo 1 se encuentra el código de algunas de las páginas del sistema.

### **3.2 DISEÑO Y DESARROLLO DE PÁGINAS DEL SISTEMA.**

En esta etapa se diseñara el flujo de páginas adecuado a la resolución de los requerimientos del Laboratorio de Metrología. El sistema tiene dos subsistemas uno orientado a los investigadores del Laboratorio y el otro al público en general.

**Páginas de mantenimiento:** Este es el subsistema de las páginas en las que los investigadores del Laboratorio de Metrología pueden consultar, modificar y agregar información de:

- El Laboratorio de Metrología (responsable, teléfono, dirección, etc...)
- Servicios que se realizan en el Laboratorio de Metrología.
- Metrólogos del Laboratorio de Metrología.
- Ordenes SM
- Solo consultas de cotizaciones.

**Páginas para el público:** Este es el subsistema de las páginas que son visualizadas desde Internet por el público en general para obtener información de:

- Servicios ofrecidos por el Laboratorio de Metrología.
- Consulta de Ordenes SM
- Solicitud de cotizaciones.
- Información del Centro de Instrumentos.
- Información de la Universidad Nacional Autónoma de México.



### 3.2.1. PÁGINAS DE MANTENIMIENTO

El flujo de las páginas de mantenimiento está esquematizado en el diagrama de la figura 3.19.

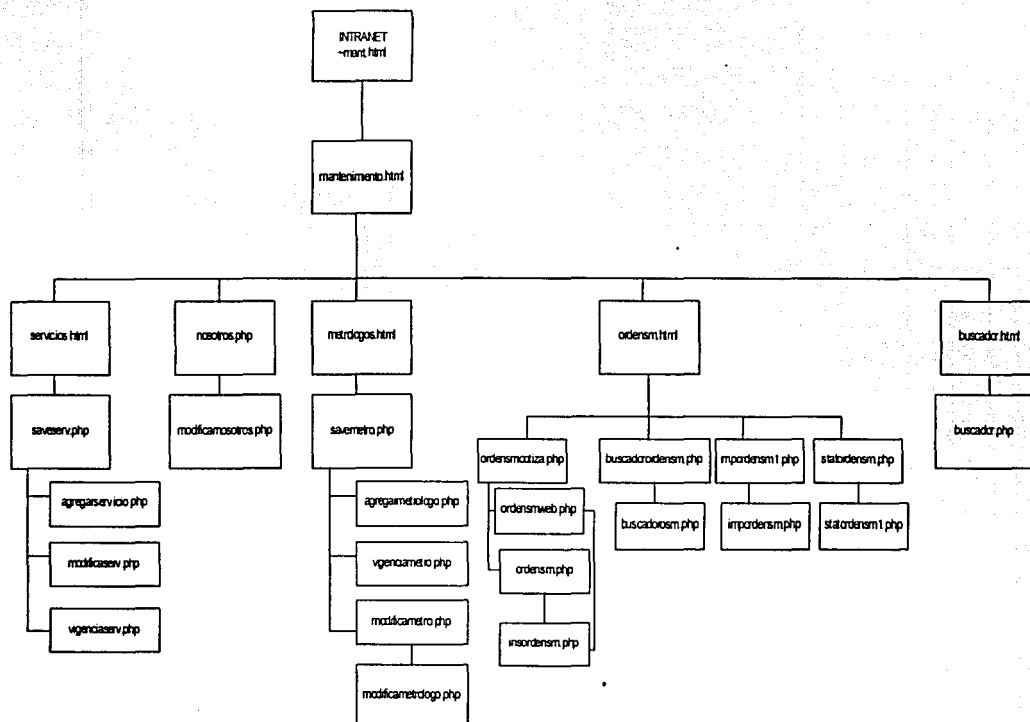


Figura 3.19 Esquema jerárquico de las páginas de mantenimiento.

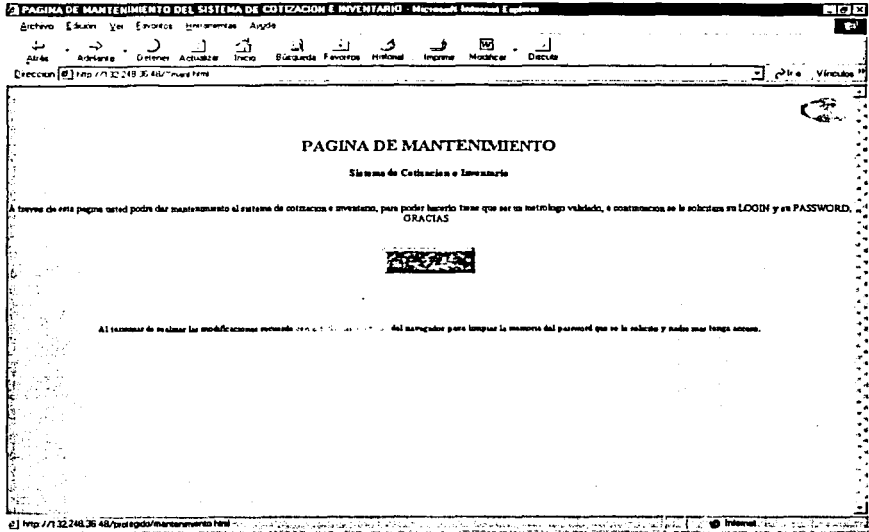
Se sintetiza la función e interrelación de cada una de las páginas de mantenimiento en la tabla 3.4.

Página	Descripción	Páginas asociadas
~mant.html	Página de bienvenida a la Intranet a través de la cual se autentican los investigadores	mantenimiento.html
mantenimiento.html	Menú principal de la Intranet	servicios.html nosotros.php metrólogos.html ordenSM.html buscador.html
servicios.html	Menú de los servicios que se ofrecen	save serv.php
save serv.php	Script que realiza el listado, solicita el número de servicio a modificar o cambiar vigencia y captura los datos para agregar el servicio.	vigenciaserv.php modificaserv.php agregarservicio.php
agregarservicio.php	Agrega el servicio a la tabla servicio .	servicios.html
Modificaserv.php	Script que envía los datos del servicio a ser modificados	modificarservicio.php
Modificarservicio.php	Modifica los datos del servicio	servicios.html
vigenciaserv.php	Cambia la vigencia del servicio	servicios.html
nosotros.php	Muestra los datos del centro de instrumentos	modificarnosotros.php mantenimiento.html
modificarnosotros.php	Actualiza los datos del centro de instrumentos	mantenimiento.html
metrologos.html	Menú de metrólogos	save metro.php mantenimiento.html
save metro.php	Script que realiza el listado, solicita el número de metrólogo a modificar o cambiar vigencia y captura los datos para agregar al metrólogo.	agregarmetrologo.php vigenciametro.php modificametro.php
Modificametro.php	Script que manda los datos a modificar del metrólogo	modificametrologo.php
vigenciametro.php	Cambia la vigencia del metrólogo	metrólogos.html
agregarmetrologo.php	Agrega el metrólogo a la tabla investigador	metrólogos.html
Modificametrologo.php	Script que actualiza los datos del metrólogo	metrólogos.html
Ordensm.html	Menú de ordenSM	ordensm cotiza.php buscadorordensm.php impordensm1.php statordensm.php mantenimiento.html
Ordensm cotiza.php	Menú para agregar una ordenSM desde una cotización desde Internet o sin ella	ordensmweb.php ordenSM.php ordenSM.html
OrdensmWeb.php	Script que trae los datos de la cotización de Internet a la ordenSM	insordensm.php
Ordensm.php	Script de captura de datos para la ordenSM	insordensm.php

Página	Descripción	páginas asociadas
Insordensm.php	Script que ingresa datos a la tabla ordenSM	ordensm.html
Buscadorordensm.php	Script que captura los datos para realizar la búsqueda de la ordenSM	buscadorosm.php
Buscadorosm.php	Script que muestra los resultados de la búsqueda de la ordenSM	ordensm.html
Impordensm1.php	Captura el número de ordenSM a imprimir	impordensm.php
Impordensm.php	Página de impresión de la ordenSM	ordensm.html
Statordensm.php	Script Captura el número de ordenSM a cambiar status (terminada o no terminada)	statordensm1.php
Statordensm1.php	Script que modifica el status en la tabla ordenSM	ordensm.html
Buscador.html	Script que captura los datos para realizar la búsqueda de la cotización	buscador.php mantenimiento.html
Buscador.php	Script que muestra los resultados de la búsqueda de la cotización para la impresión	buscador.html

Tabla 3.4 Inventario de páginas de mantenimiento

A continuación se muestra el prototipo operacional de las páginas de mantenimiento con su descripción y objetos involucrados, algunas de estas solo son pantallas de acceso o resultados por lo que no pueden presentar objetos.



Página ~mant.html

Objeto	Clase	Función
INGRESAR	Ref.	Invocar la página mantenimiento.html.

Para ingresar a las páginas de mantenimiento el servidor apache solicita un nombre de usuario y un password a validar como se muestra:

**Escribir contraseña de red** [?] [X]

Esciba su nombre de usuario y contraseña.

Sitio: 132.248.36.48

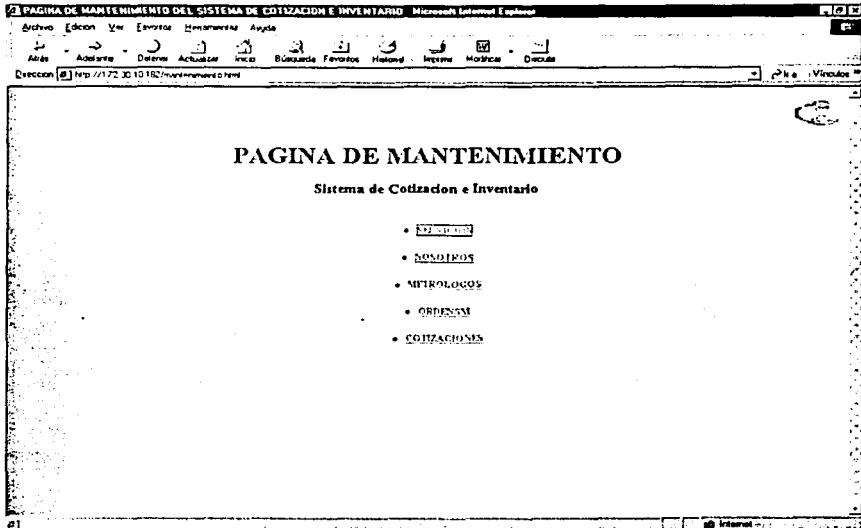
Dominio: Personal autorizado

Nombre de usuario:

Contraseña:

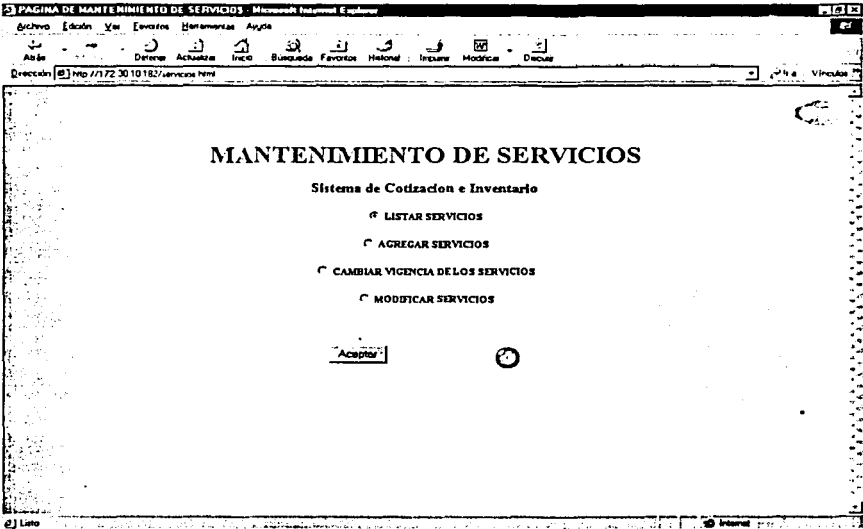
Guardar esta contraseña en la lista de contraseñas

Página de autorización de acceso.



Página de mantenimiento.html

Objeto	Clase	Función
SERVICIOS	href	Especifica un vínculo a la página de servicios.html
NOSOTROS	href	Es un vínculo al programa php/nosotros.php.
METROLOGOS	href	Es un vínculo a la página de metrólogos.
ORDENSM	href	Es una liga a la página ordenSM.html.
COTIZACIONES	href	Es un vínculo a la página buscador.html.



Página servicios.html

Objeto	Clase	Función
Rad	radio	Este objeto admite las siguientes opciones de los servicios: Listar Agregar Bajar Modificar
Aceptar	submit	Envía a php/saveserv.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152134_2	href	Es un vínculo a la página mantenimiento.html

**PAGINA DE SERVICIOS** - Microsoft Internet Explorer

Archivo Edición Ver Herramientas Ayuda

Inicio Búsqueda Favoritos Historial Imprimir Discard

Dirección http://132.249.36.48/psorp/psorp/saveserv.php

**SERVICIOS REGISTRADOS**

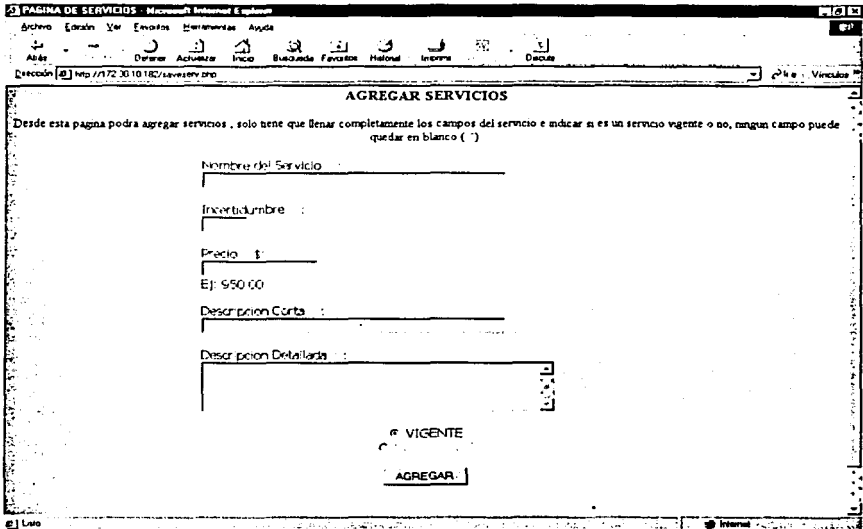
Los servicios se muestran con el formato siguiente:

Numero	Nombre del Servicio	Inscripciones	Precio	STATUS
1	BLOQUES PATRON PIEZA PIEZA AISLADA EN UNIDADES SI (mm)	0 0000	100.00	VIGENTE
2	BLOQUES PATRON CONJUNTO PIEZA EN CONJUNTO EN UNIDADES SI (mm)	0 0000	100.00	VIGENTE
3	BARRAS PATRON PIEZA INDIVIDUAL PIEZA AISLADA DE HASTA 1000 mm DE 1 A 40 PULGADAS	0 0000	200.00	VIGENTE
4	BARRAS PATRON CONJUNTO PIEZA EN CONJUNTO DE HASTA 1000 mm PIEZA EN CONJUNTO DE HASTA 1000 mm equivalente	0 0000	100.00	VIGENTE
5	PERNOS PATRON PIEZA PIEZA AISLADA DE HASTA 100 mm	0 0000	100.00	VIGENTE

Página saveserv.php

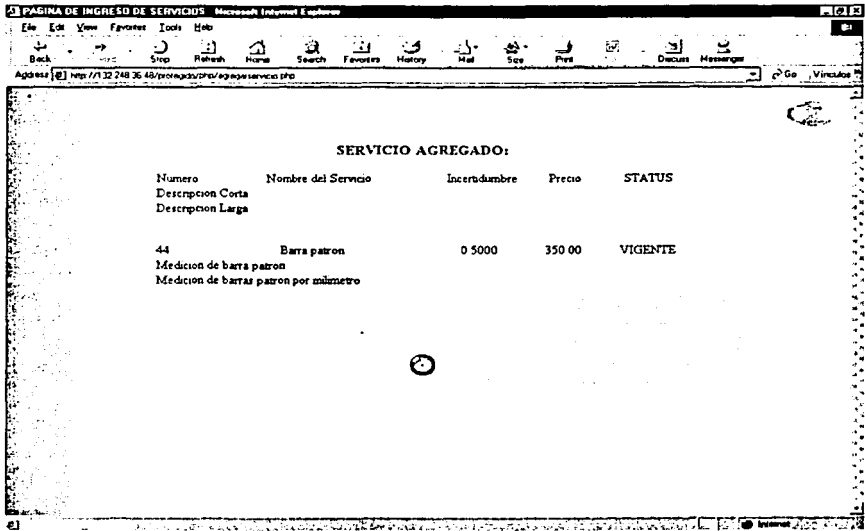
Objeto	Clase	Función
MouseOver_20000123152131_2	Ref.	Es un vínculo a la página servicios.html.





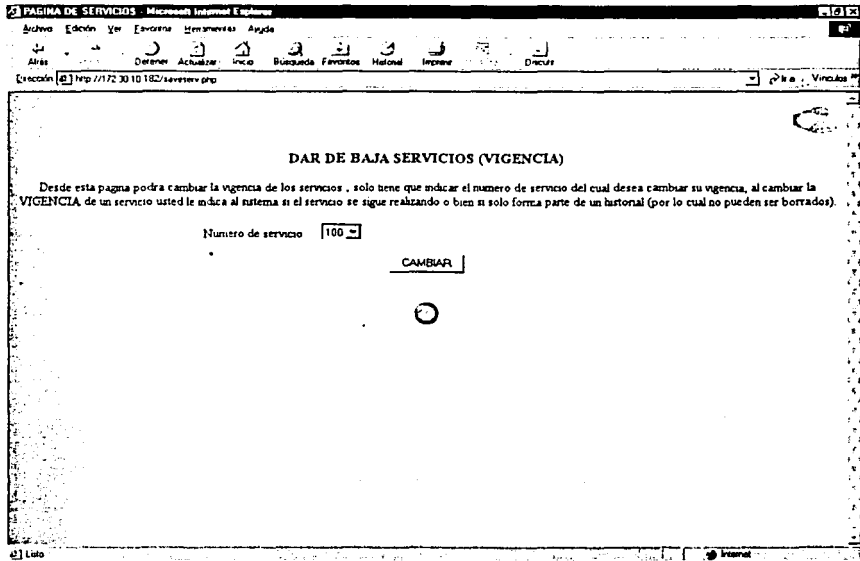
Página saverserv.php

Objeto	Clase	Función
sernom	input	Almacenar el nombre del servicio a agregar.
serinc	Input	Almacenar la incertidumbre del servicio.
servpre	input	Almacenar el precio del servicio.
serdesc	input	Almacenar una breve descripción del servicio.
serdescbig	textarea	Almacenar una descripción larga del servicio.
serstat	radio	Indica el estado del servicio, vigente o no vigente.
Aceptar	submit	Envía a agregar servicio.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152134_2	href	Es un vínculo a la página servicios.html.



Página agregarservicio.php

Objeto	Clase	Función
sercve	select	Seleccionar el número de servicio al que se le va a cambiar la vigencia.
CAMBIAR	submit	Envía a vigenciaserv.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	bottom	Es un vínculo a la página servicios.html.



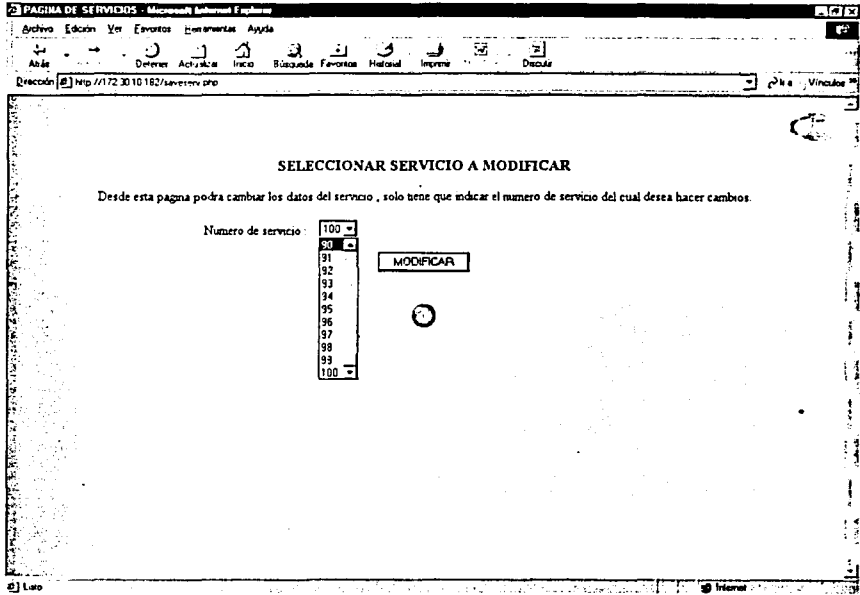
Página saverserv.php

Objeto	Clase	Función
servce	select	Seleccionar el número de servicio al que se le va a cambiar la vigencia.
CAMBIAR	submit	Envía a vigenciaserv.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	bottom	Es un vínculo a la página servicios.html.

¡Error! Vinculo no válido.

Página vigenciaserv.php

Objeto	Clase	Función
MouseOver_20000123152131_2	bottom	Es un vínculo a la página servicios.html.



Página saveserv.php

Objeto	Clase	Función
sercve	select	Seleccionar el número de servicio al que se va a modificar.
MODIFICAR	submit	Envía a modifcaserv.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	bottom	Es un vínculo a la página servicios.html.

**PÁGINA DE MODIFICACIÓN DE SERVICIOS** Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abre Detener Actualizar Inicio Búsqueda Favoritos Historial Imprimir Inicio

Dirección http://132.248.35.43/prof-ado/php/modificaserv.php

**MODIFICAR SERVICIOS**

Desde esta página podrá modificar los datos del servicio, solo los que tienen como elemento los campos del servicio o indicar si es un servicio vigente o no, siempre cuando puede quedar en blanco (\*)

Nombre del Servicio:  
BLOQUES PATRON PIEZA

Incertidumbre:  
0.0000

Precio \$:  
100.00  
Ej: 100.00

Descripción Corta:  
PIEZA AISLADA EN UNIDADES SI (mm)

Descripción Estilada:

VIGENTE

Página modificaserv.php

Objeto	Clase	Función
sernom	input	Almacenar el nombre del servicio a modificar.
serinc	input	Almacenar la incertidumbre del servicio.
servpre	input	Almacenar el precio del servicio.
serdesc	input	Almacenar una breve descripción del servicio.
serdescbig	textarea	Almacenar una descripción larga del servicio.
serstat	radio	Indica el estado del servicio, vigente o no vigente.
MODIFICAR	submit	Envía a modificaservicio.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	bottom	Es un vínculo a la página servicios.html.

PÁGINA DE ACTUALIZACIÓN SOBRE LA INFORMACIÓN REFERENTE A NOSOTROS (METROLOGÍA) - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio Detener Actualizar Inicio Búsqueda Favoritos Historial Imprimir Modificar Descarga

Dirección http://171.22.249.36/45/proteccion.php-nosotros.php

## INFORMACION REFERENTE A NOSOTROS

Sistema de Certificación e Inventario

A través de esta página usted podrá actualizar los datos referentes a Metrología y los contactos que existen, por lo cual algunos de estos pueden quedar en blanco ( \* )

Nombre de la Institución \* :

Nombre de la Sección \* :

Cuenta Pagos por depósito \* :

Deposito a nombre de :

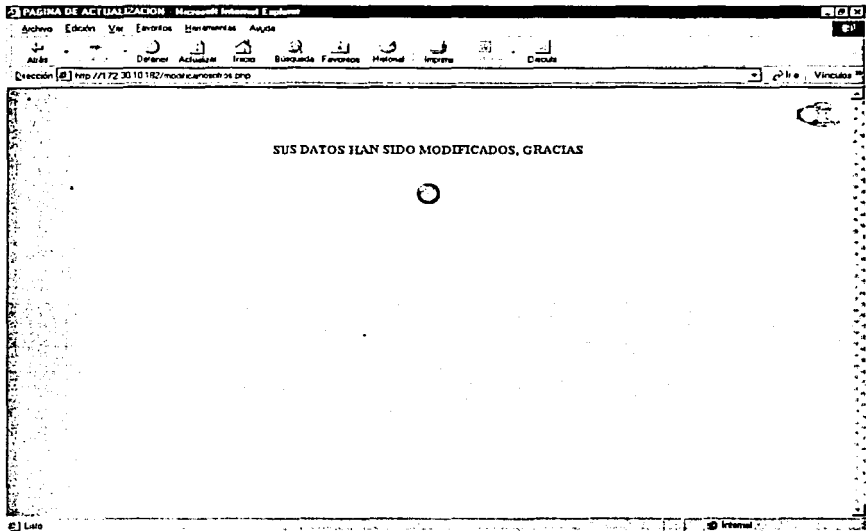
Teléfono :  Fax :

**RESPONSABLE**

2] Inicio Internet

## Página nosotros.php

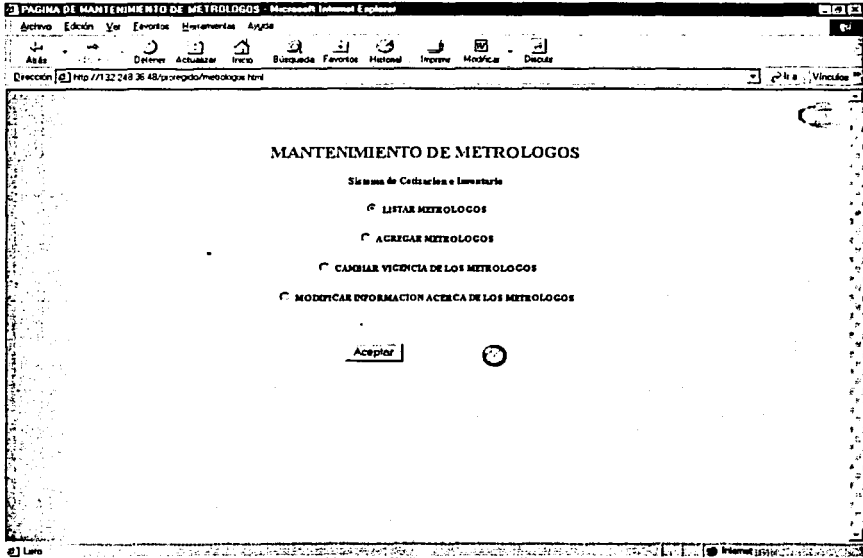
Objeto	Clase	Función
instnom	input	Almacenar el nombre de la institución.
instlab	Input	Almacenar el nombre de la sección.
instcuenta	Input	Almacenar el número de cuenta en donde se debe hacer el pago al centro de instrumentos.
instdep	Input	Almacenar el nombre de la cuenta en donde se debe hacer el pago al centro de instrumentos.
instel	Input	Almacenar el número telefónico de la institución.
instfax	input	Almacenar el número de fax de la institución.
instresp	input	Almacenar el nombre del responsable de la institución.
Aceptar	submit	Enviar a modificarnosotros.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	bottom	Es un vínculo a la página mantenimiento.html.



Página modificarnosotro.php

Objeto	Clase	Función
MouseOver_20000123152131_2	Bottom	Es un vinculo a la página mantenimiento.html.

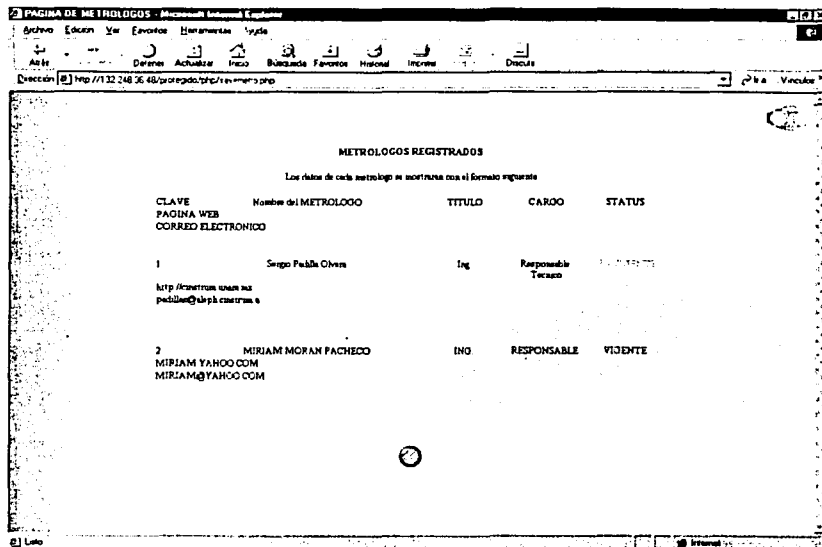




Página metrólogos.html

Objeto	Clase	Función
Rad	Radio	Este objeto admite las siguientes opciones de los metrólogos: Listar Agregar Bajar Modificar
Aceptar	Submit	Envía a php/savemetro.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152134_2	Ref.	Es un vinculo a la página mantenimiento.html

TESIS CON  
FALLA DE ORIGEN



Página savemetro.php

Objeto	Clase	Función
MouseOver_20000123152134_2	href	Es un vinculo a la página metrologos.html

PAGINA DE METROLOGOS - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Stop Refresh Home Search Favorites History Mail Size Print Discuss Messages

Address http://132.248.36.48/programa/zip/savemetro.php

### AGREGAR METROLOGOS

Desde esta página podrá agregar METROLOGOS , solo tiene que llenar completamente los campos e indicar si el METROLOGO aun trabaja en el centro (vigente o no), ningún campo puede quedar en blanco ( \* )

Nombre del Metrologo : \_\_\_\_\_

TITULO : \_\_\_\_\_

CARGO : \_\_\_\_\_

Página WEB : \_\_\_\_\_

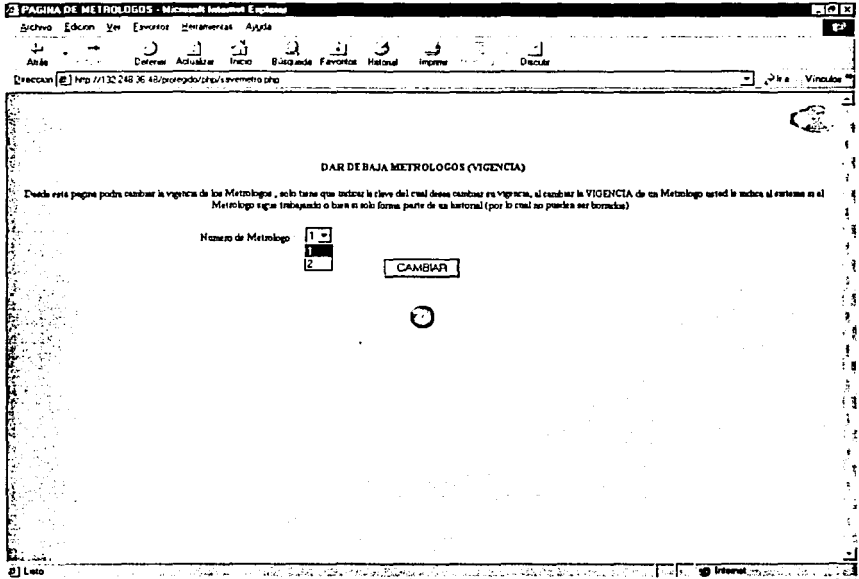
CORREO ELECTRONICO : \_\_\_\_\_

VIGENTE  
 NO VIGENTE

Done

Página savemetro.php

Objeto	Clase	Función
invnom	Input	Almacenar el nombre del metrólogo a agregar.
invtit	Input	Almacenar el título del metrólogo.
invcargo	Input	Almacenar el cargo del metrólogo.
invpgvWeb	Input	Almacenar la página Web del metrólogo.
invemail	Input	Almacenar el email del metrólogo.
invstat	radio	Indica el estado del metrólogo, vigente o no vigente.
Aceptar	submit	Envía a agregarmetrologo.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152134_2	href	Es un vínculo a la página metrólogos.html.



Página savemetro.php

Objeto	Clase	Función
invcve	select	Seleccionar el número de metrólogo al que se le va a cambiar la vigencia.
CAMBIAR	submit	Envía a vigenciametro.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	bottom	Es un vínculo a la página metrólogos.html.

PAGINA DE VIGENCIA DE METROLOGOS Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abre Detener Actualizar Inicio Búsqueda Favoritos Historial Impresora Discos

Dirección http://132.248.36.40/protected/metrologos.php

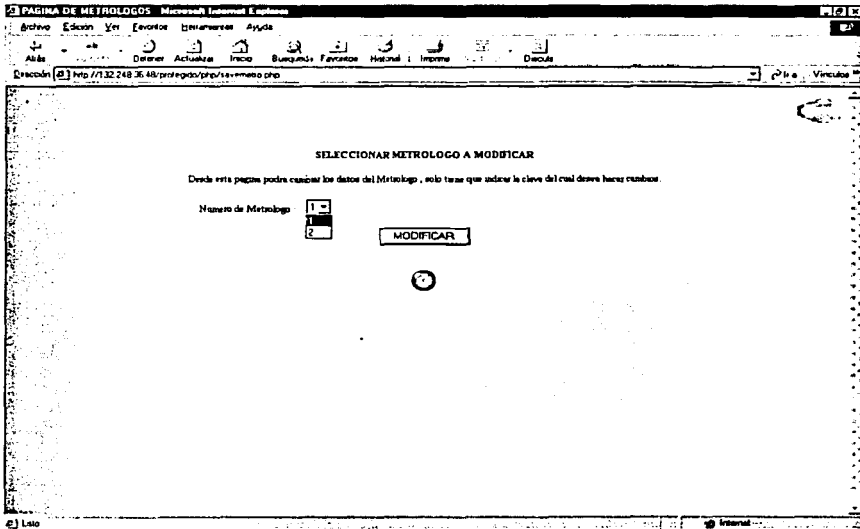
METROLOGO AL QUE SE LE CAMBIO LA VIGENCIA:

CLAVE	Nombre del METROLOGO	TITULO	CARGO	STATUS
Vigencia Anterior				
1	Sergio Peña Olvera	Ing.	Responsable Técnico	NO VIGENTE
<a href="mailto:Http://cuernavaca.com.mx/pedro@teleph.com.mx">Http://cuernavaca.com.mx/pedro@teleph.com.mx</a>				
Vigencia Actual				
1	Sergio Peña Olvera	Ing.	Responsable Técnico	VIGENTE
<a href="mailto:Http://cuernavaca.com.mx/pedro@teleph.com.mx">Http://cuernavaca.com.mx/pedro@teleph.com.mx</a>				

http://132.248.36.40/protected/metrologos.html

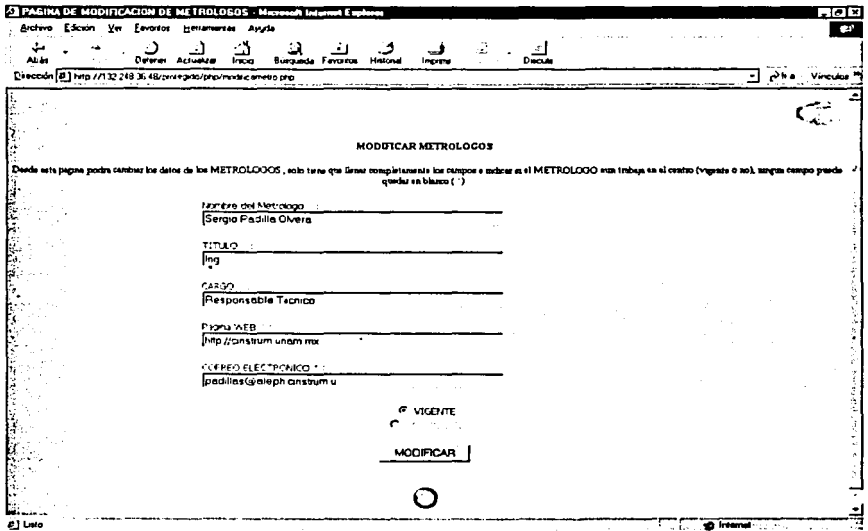
Página vigenciametro.php

Objeto	Clase	Función
MouseOver_20000123152131_2	bottom	Es un vínculo a la página metrólogos.html.



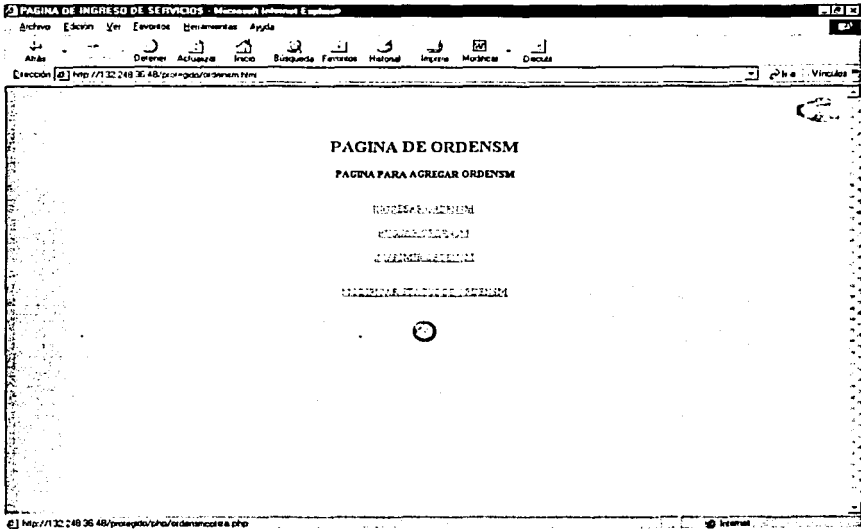
Página savemetro.php

Objeto	Clase	Función
invcve	Select	Seleccionar el número de metrólogo al que se le va a cambiar la vigencia.
MODIFICAR	Submit	Envía a modificametro.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	Bottom	Es un vínculo a la página metrólogos.html.



Página modificametro.php

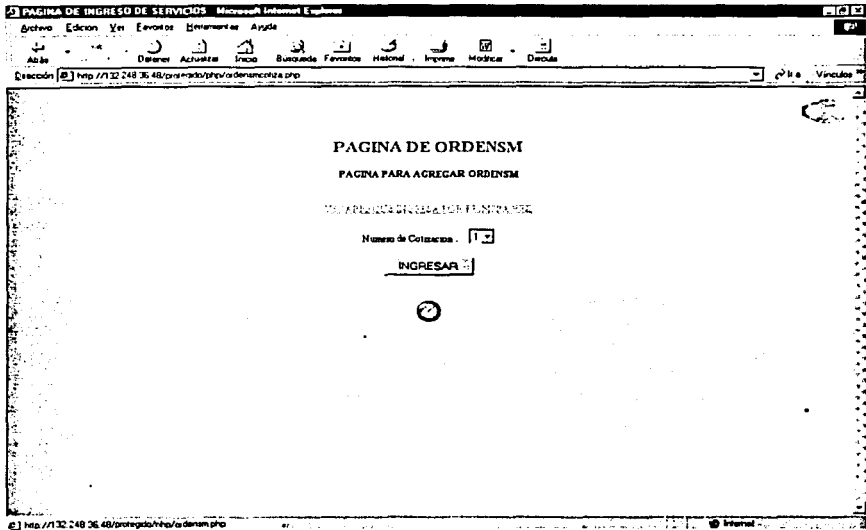
Objeto	Clase	Función
invnom	input	Almacenar el nombre del metrólogo a modificar.
invtit	Input	Almacenar el título del metrólogo.
invcargo	input	Almacenar el cargo del metrólogo.
invpgWeb	input	Almacenar la dirección de la página Web del metrólogo.
invemail	input	Almacenar la dirección del correo electrónico del metrólogo.
invstat	radio	Indica el estado del metrólogo, vigente o no vigente.
MODIFICAR	submit	Envía a modificarmetrologo.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	bottom	Es un vínculo a la página metrólogos.html.



Página ordenSM.html

Objeto	Clase	Función
INGRESAR ORDENSM	href	Especifica un vinculo a la página de php/ordensmcotiza.php
BUSCAR ORDENSM	href	Es un vinculo a la página php/buscadorordensm.php.
IMPRIMIR ORDENSM	href	Es un vinculo a la página de php/impordensm1.php.
MODIFICAR STATUS DE ORDENSM	href	Es un vinculo a la página php/statordensm.php.
MouseOver_20000123152131_2	bottom	Es un vinculo a la página mantenimiento.html.



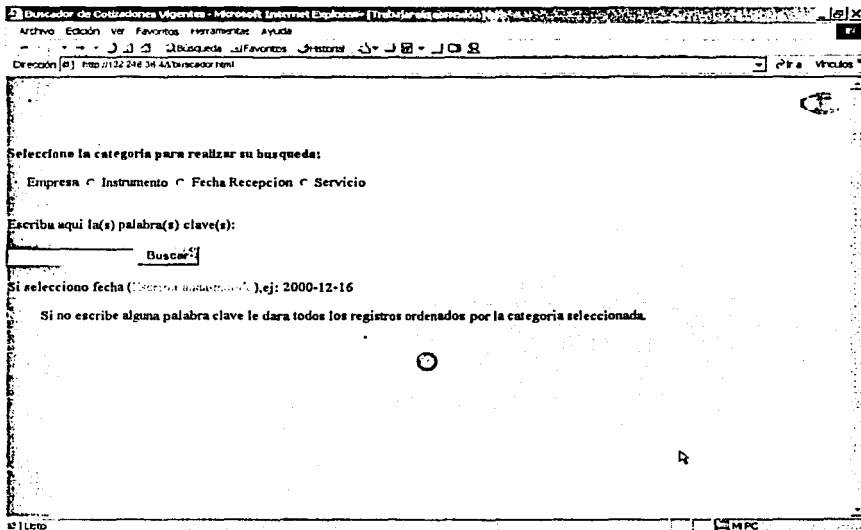


Página ordenmcoliza.php

Objeto	Clase	Función
USUARIO QUE INGRESA POR PRIMERA VEZ	href	Especifica un vínculo a la página ordenm.php
Idcotiza	Select	Seleccionar el número de cotización a la que se le va a crear la ordenSM.
INGRESAR	Submit	Envía a ordenmweb.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	Bottom	Es un vínculo a la página ordenSM.html.

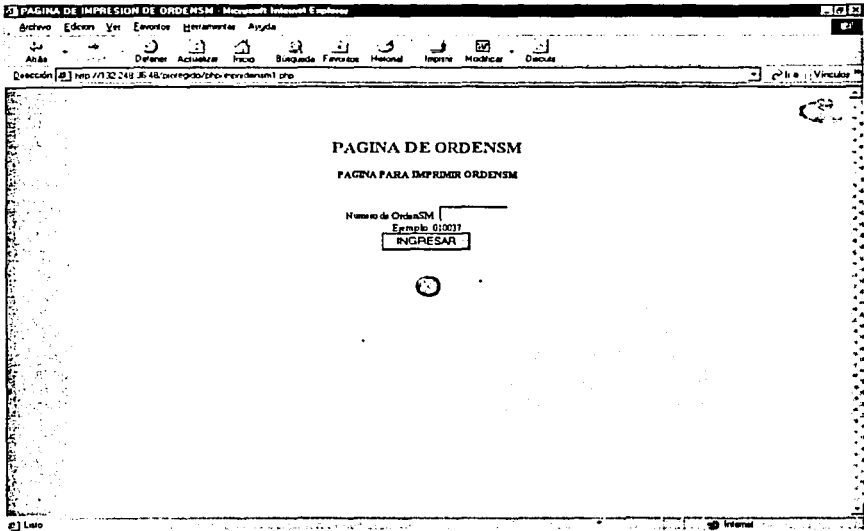






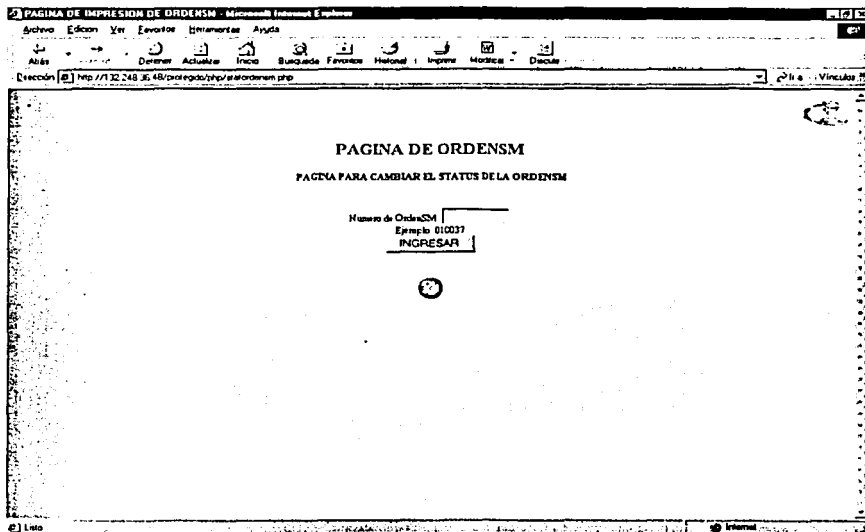
Página buscadorordensm.php

Objeto	Clase	Función
Rad	Radio	Este objeto admite las siguientes opciones de búsqueda de la ordenSM: Empresa Instrumento Fecha de recepción Servicio
BUSCAR	Input	Almacenar la palabra clave para hacer la búsqueda.
BUSCAR	Submit	Envía a buscadorosm.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	Bottom	Es un vínculo a la página protegido/ordensm.html.



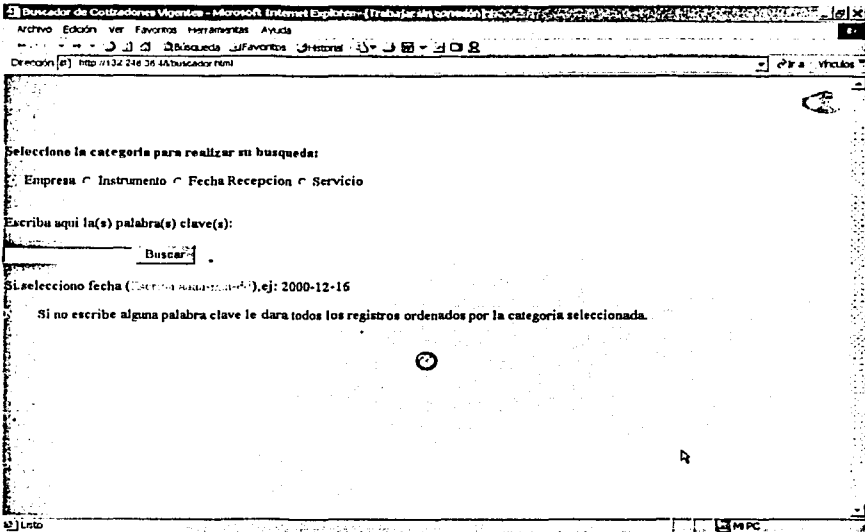
Página impordensm1.php

Objeto	Clase	Función
Ordensm	Input	Almacenar la palabra clave para hacer la búsqueda.
INGRESAR	Submit	Envía a impordensm.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	Bottom	Es un vínculo a la página ordenSM.html.



Página statordensm.php

Objeto	Clase	Función
Ordensm	Input	Almacenar el número de la ordenSM a cambiar el estatus para hacer la búsqueda.
INGRESAR	Submit	Envía a statordensm1.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	Bottom	Es un vínculo a la página ordenSM.html.



Página buscador.html

Objeto	Clase	Función
Rad	Radio	Este objeto admite las siguientes opciones de búsqueda de la cotización: Empresa Instrumento Fecha de recepción Servicio
BUSCAR	Input	Almacenar la palabra clave para hacer la búsqueda.
BUSCAR	Submit	Envía a buscador.php los valores de los objetos que contiene el formulario.
MouseOver_20000123152131_2	Bottom	Es un vínculo a la página protegido/mantenimiento.html.

Microsoft Internet Explorer

Back Stop Refresh Home Search Favorites History Mail Site Print Edit Discos Messenger

Agent http://132.248.36.48/registro/pre/empocotizacion.php?h=3

Cd. Universidad, 2 de Octubre de 2001

**Laboratorio de Metrología**

LILIANA SALAZAR VAZQUEZ  
 LIZWIS A. DE C.V.  
 CALLE P #48-1,  
 TRAPR,  
 04600, MEXICO D.F.  
 FAX 56716320.  
 TELEFONO 56774791  
 E-Mail lizwiz@yahoo.com

Atendiendo a su solicitud de calibración de los instrumentos que a continuación se mencionan, me permito informarle que nuestro *Laboratorio de Metrología*, tiene la capacidad de realizar dicho servicio con las siguientes características:

El costo por el servicio incluyendo informe con especificaciones es de:

Cantidad: 5 Instrumentos Barras Patron Conjunto Marca Minayo Precio Total \$ 500.00

**NOTAS.**

1. La UNAM no cobra I.V.A.
2. Esta cotización tiene vigencia de 30 días
3. El pago se efectúa en nuestras instalaciones y puede hacerse en efectivo o mediante cheque certificado o de ventanilla, o bien, mediante depósito bancario en la cuenta BANCOMER No. 1355809-3 Sucursal Península a nombre del "Centro de Instrumentos, UNAM" y enviar o traer ficha de depósito original.
4. El tiempo de entrega será de 5 días hábiles a partir de la recepción de los instrumentos en nuestro laboratorio para lo cual deberá haberse cubierto el pago correspondiente
5. La recepción de los mismos puede ser de inmediato
6. Si desea se le envíe el equipo y los informes de calibración o por mensajería, deberá enviarnos una copia pre-pagada o, si es el caso, una carta de autorización

Dado que la realización de una calibración de esta naturaleza requiere extremo cuidado y atención especial le pedimos confirmar a la brevedad posible la aceptación de la presente para programar el trabajo dentro de las actividades del Laboratorio de Metrología.

Atentamente  
 Sergio Padilla Civera  
 Responsable Técnico

Tel 56 22 86 03 al FAX 56 22 86 24 E-mail [registro@construm.unam.mx](mailto:registro@construm.unam.mx)  
 Página WEB <http://construm.unam.mx>

Página de cotización

Objeto	Clase	Función
MouseOver_20000123152131_2	bottom	Es un vínculo a la página mantenimiento.html.



### 3.2.2 PÁGINAS PÚBLICAS

El flujo de las páginas públicas esta esquematizado en el diagrama de la figura 3.20.

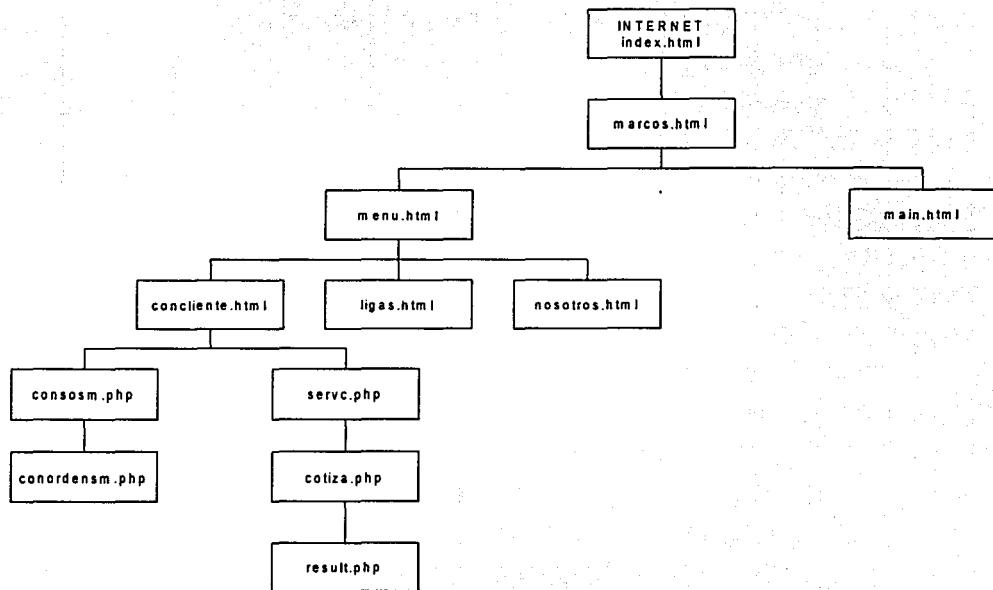


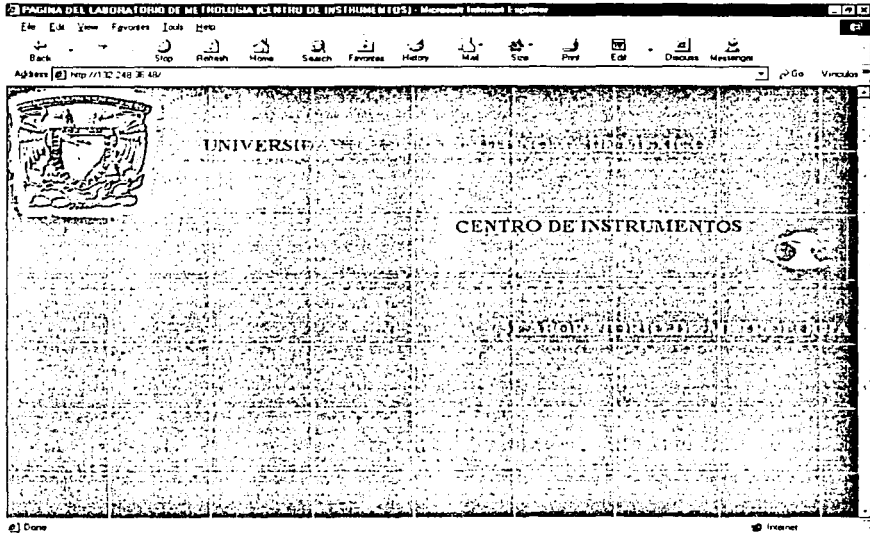
Figura 3.20 Esquema jerárquico de páginas públicas

En la tabla 3.5 se hace un inventario de las páginas públicas del sistema:

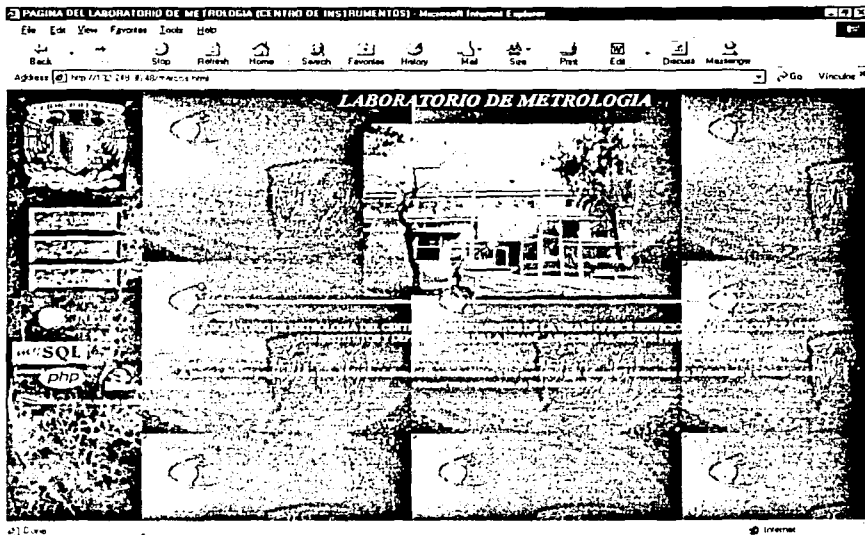
Página	Descripción	páginas asociadas
Index.html	Página de bienvenida	marcos.html
Marcos.html	División de la página principal en dos secciones.	menu.html main.html
Menu.html	Contiene las ligas para cada uno de los contenidos del sitio.	concliente.html ligas.html nosotros.html
Main.html	Página principal del sitio.	
Concliente.html	Página de menú de cotización de servicios o consulta de estado de la ordenSM.	consosm.php servc.php
Consosm.php	Script en donde se captura el número de ordenSM a consultar	conordensm.php
Conordesm.php	Muestra el estado de la ordenSM, terminado o no.	
Servc.php	Muestra los servicios que ofrece el laboratorio de metrología.	cotiza.php
Cotiza.php	Script que captura la información del cliente para procesar la cotización	result.php
Result.php	Script que ingresa datos a la tabla solicitud, y envía a pantalla la cotización para impresión	
Ligas.html	Página que incluye referencias a otras páginas de la UNAM	
Nosotros.html	Página que ofrece información sobre los proyectos y otras actividades del centro de instrumentos.	

Tabla 3.5 Inventario de páginas públicas

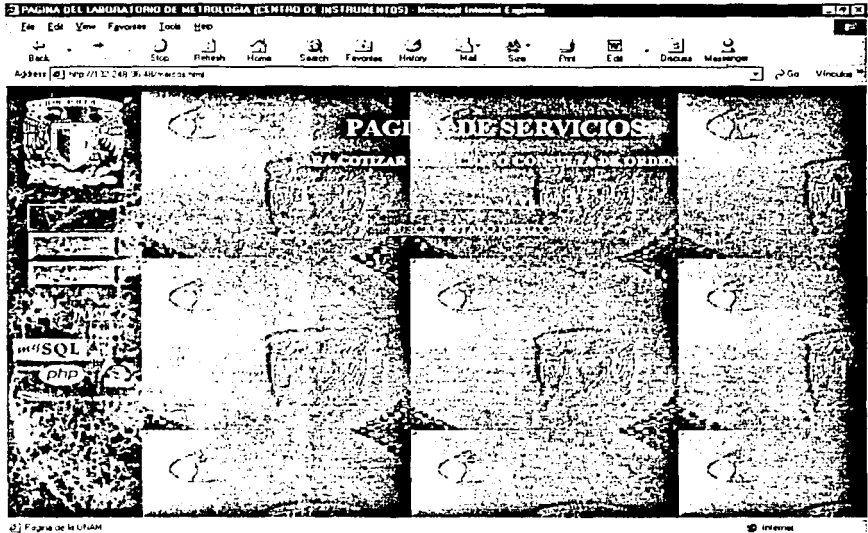
A continuación se muestra el prototipo operacional de las páginas públicas con su descripción y objetos involucrados, algunas de estas solo son pantallas de acceso o resultados por lo que no pueden presentar objetos.



Página index.html

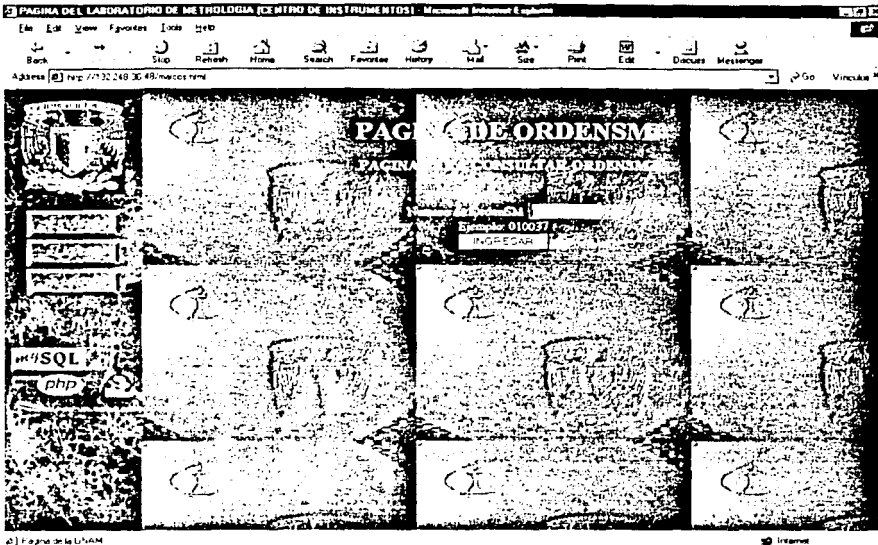


Página main.html



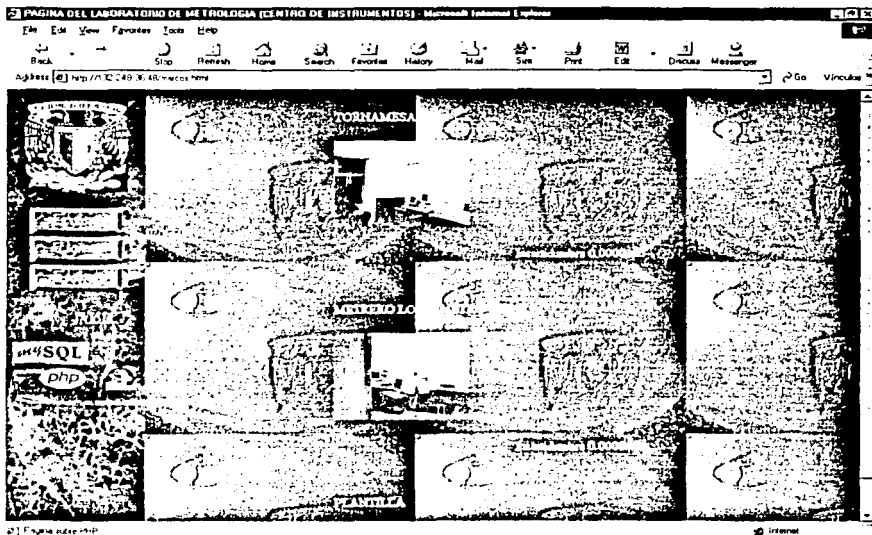
Página concliente.html

Objeto	Clase	Función
COTIZAR SERVICIO	Href.	Genera una cotización.
BUSCAR ESTADO DE ORDENSM	Href.	Busca el estado en que se encuentra una ordenSM para consulta del cliente.



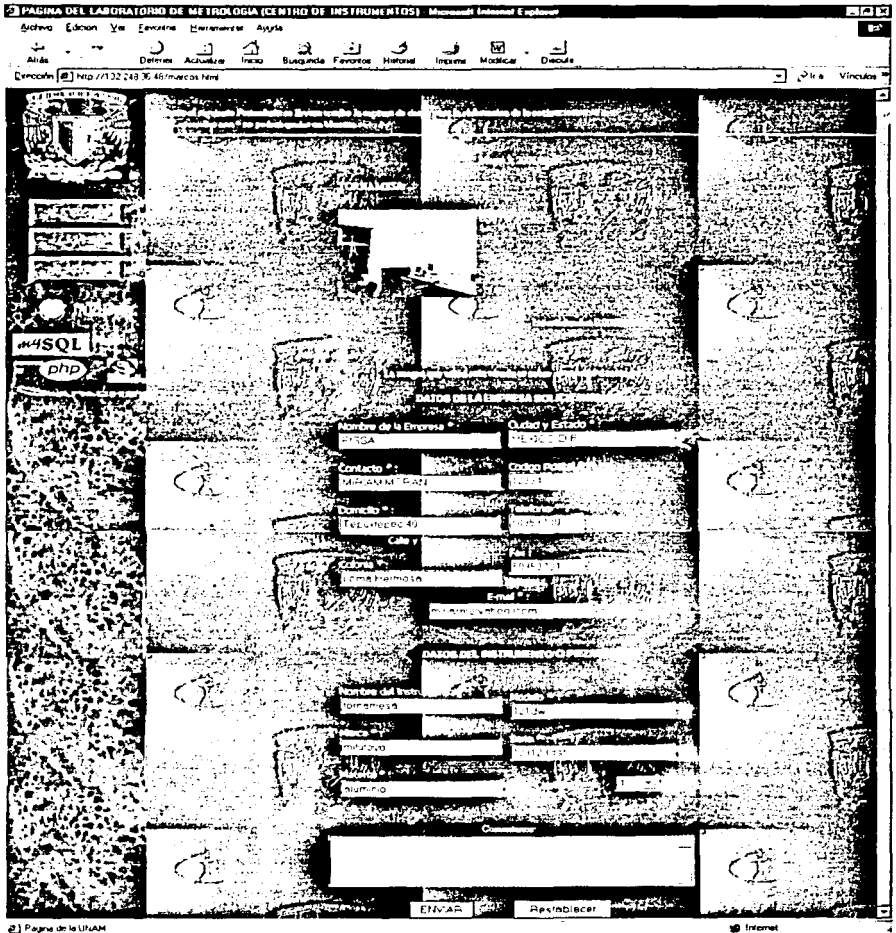
Página consosm.php

Objeto	Clase	Función
Ordensm	Input	Captura el número de la ordenSM.
INGRESAR	Submit	Envía los datos del formulario al script conordensm.php



Página servc.php

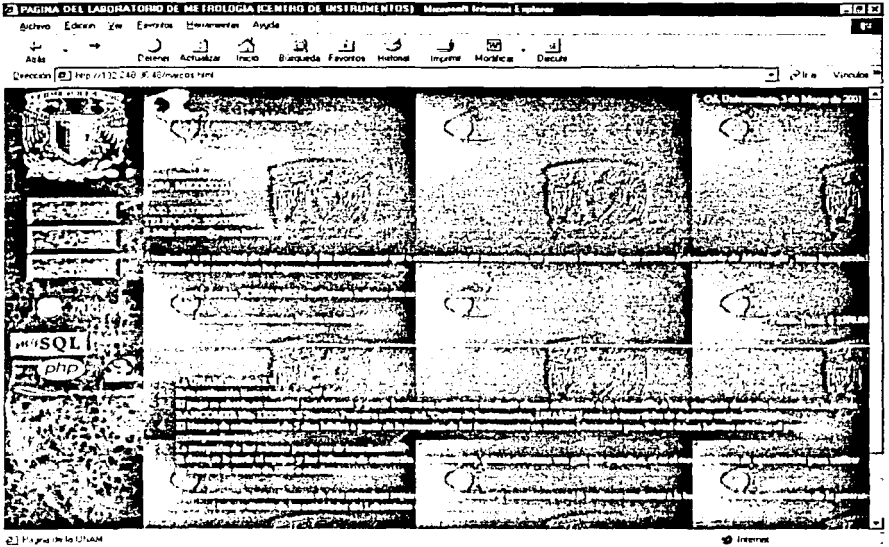
Objetivo	Clase	Función
Cotizar	Ref.	Busca la pagina de captura de datos correspondiente al servicio seleccionado
INICIO	Ref.	Lleva al inicio de la página
<a href="http://labmetrologiaunam.dhs.org">http://labmetrologiaunam.dhs.org</a>	Ref.	Lleva a la página de bienvenida



Página cotiza.php

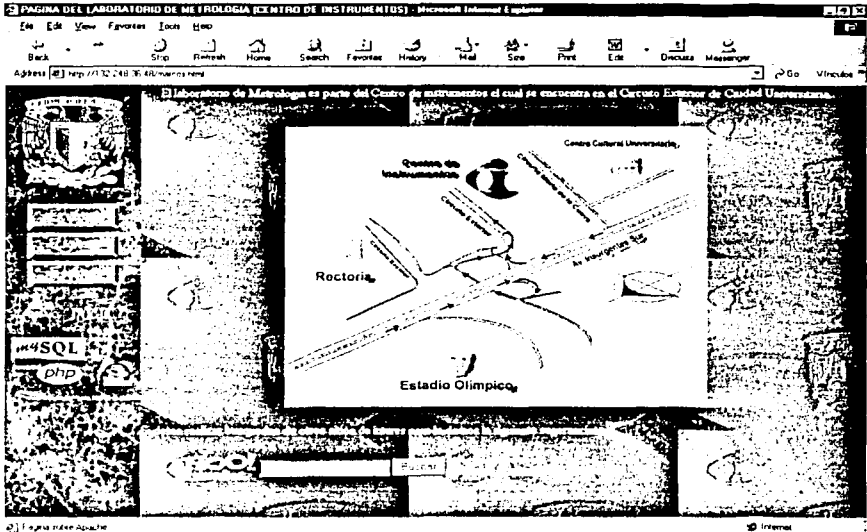


Objeto	Clase	Función
Empresa	Input	Captura el nombre de la empresa o persona
Contacto	Input	Captura el nombre de la persona que hace el contacto o responsable de la solicitud.
Domicilio	Input	Captura el domicilio de la empresa o persona, calle y número
Colonia	Input	Captura la colonia
Ciudad	Input	Captura la ciudad
Postal	Input	Captura el código postal
Tel	Input	Captura el número telefónico
Fax	Input	Captura el número de fax
Email	Input	Captura la dirección de la empresa o persona
Insnom	input	Captura el nombre del instrumento del cual se solicita el servicio
Marca	Input	Captura la marca del instrumento
Material	Input	Captura el nombre del material del instrumento
Modelo	Input	Captura el modelo del instrumento
Serie	Input	Captura el número de serie del instrumento
Cantidad	Select	Captura la cantidad de instrumentos que se desean se les haga el servicio.
Comentarios	Textarea	Captura información extra relativa al instrumento o servicio.
Restablecer	Reset	Borra los datos de todas las cajas anteriores, permitiendo una nueva captura de los mismos
Enviar	Submit	Envía los datos del formulario al script result.php



Página result.php

TESIS CON  
FALLA DE ORIGEN



Página ligas.html

### 3.3 PRUEBAS DEL SISTEMA

Para concluir con la última etapa del Modelo de Espiral se incluye la fase de pruebas que compruebe como se resolvió la problemática planteada con el Sistema de Cotización e Inventario del Laboratorio de Metrología.

FASE DE PRUEBAS Y VALIDACIÓN DE RESULTADOS DEL SISTEMA DE COTIZACIÓN E INVENTARIO	
Propósito:	Identificar todos los requerimientos del Sistema de cotización e inventario, que van a ser probados.
Antecedentes:	Debido a la automatización de operación del proceso de <i>solicitud-cotización-inventario</i> del Laboratorio de Metrología, se requieren hacer pruebas que arrojen los resultados esperados.
Alcance:	<p>La fase de pruebas contempla los siguientes tipos:</p> <p><b>Pruebas Unitarias:</b> Realizadas por el Ingeniero de Pruebas de Ejecución, su responsabilidad es asegurar la funcionalidad del requerimiento establecido y la integridad de la información de las pantallas para el subsistema que esta creando o modificando.</p> <p><b>Pruebas Integrales:</b> Realizadas por el Ingeniero de Pruebas de Diseño, su responsabilidad es garantizar que las pantallas o reportes cumplen con el requerimiento establecido e integrar cada una de ellas a la aplicación y probar con datos reales su funcionalidad e integridad de la información.</p> <p><b>Pruebas de Aceptación:</b> Realizadas por el usuario final del Laboratorio de Metrología, quien deberá verifica que cada subsistema del sistema creado cumpla los requerimientos definidos al inicio del proyecto.</p>

PRUEBAS DE PAGINAS DE MANTENIMIENTO SUBSISTEMA SERVICIOS		
PRUEBAS UNITARIAS	Objetivo	Asegurar por parte del ingeniero de pruebas de ejecución la apropiada funcionalidad de la aplicación en la captura de los datos de un nuevo servicio.
	Técnica:	<p>Verificar los resultados esperados cuando se utilizan datos válidos.</p> <p>Validar que existan mensajes de error o advertencia que son desplegados cuando se utilizan datos inválidos.</p> <p>Asegurar que cada regla de negocio sea aplicada apropiadamente.</p>
	Resultados	<p>Se logro ingresar correctamente el nombre del servicio, incertidumbre, precio, descripción corta, descripción larga y vigencia del nuevo servicio, apareciendo la página en la que se muestra el servicio agregado.</p> <p>Al tratar de ingresar un servicio sin precio el sistema envía el mensaje de error "Por favor escriba el precio del servicio."</p> <p>Todos los servicios deben tener una incertidumbre y un precio validados por el laboratorio de metrología, lo cual cumple el sistema.</p>
PRUEBAS INTEGRALES	Objetivo	<p>El Ingeniero de pruebas de diseño debe:</p> <p>Validar el diseño de páginas.</p> <p>Validar el reporte de servicios.</p> <p>Verificar que la navegación desde el menú principal hasta el menú de servicios a través de los elementos que reflejan las funciones y requerimientos del negocio incluyendo la navegación ventana a ventana, campo a campo y el uso de los métodos de acceso (movimientos del mouse y teclas).</p>
	Técnica	<p>Navegar en las pantallas desde el menú principal hasta el menú de servicios haciendo uso del mouse y teclas para ingresar todos los datos del nuevo servicio.</p> <p>Realizar una consulta de servicios para generar el reporte solicitados por el Laboratorio de Metrología.</p>
	Resultados	<p>La navegación de pantallas es consistente en el sistema.</p> <p>Cada ventana cumple con el diseño establecido en la documentación técnica del sistema.</p> <p>El reporte generado de servicios contiene toda la información solicitada.</p>
PRUEBAS DE ACEPTACIÓN	Objetivo	El usuario final verificara que todos los elementos del subsistema de servicios funcionen adecuadamente.
	Técnica	El usuario realizara altas, cambios ,consultas e impresiones de un servicio, basándose en el flujo del esquema jerárquico de la figura 3.20 y en el manual de usuario del Anexo A.
	Resultados	Las pruebas de altas, cambios, consultas e impresiones de servicio por el usuario final se realizaron satisfactoriamente.

PRUEBAS DE PAGINAS DE MANTENIMIENTO SUBSISTEMA NOSOTROS		
PRUEBAS UNITARIAS	Objetivo	Asegurar por parte del ingeniero de pruebas de ejecución la apropiada funcionalidad de la aplicación en la actualización del teléfono del laboratorio de metrología con datos válidos e inválidos.  Verificar que siempre exista un responsable.
	Técnica:	Actualizar con datos válidos e inválidos el teléfono del laboratorio de metrología.  Dejar sin asignar en el sistema al responsable del Laboratorio de Metrología.
	Resultados	Se actualizo exitosamente el número telefónico del laboratorio de metrología, enviando como resultado la página con la leyenda "SUS DATOS HAN SIDO MODIFICADO, GRACIAS".  En la inserción de datos inválidos el resultado es un mensaje de error.  Al no especificar al sistema el responsable del Laboratorio de Metrología se mantiene el responsable actual.
PRUEBAS INTEGRALES	Objetivo	El Ingeniero de pruebas de diseño debe :  Validar el diseño de páginas.  Verificar que la navegación desde el menú principal hasta el menú de nosotros a través de los elementos reflejen las funciones y requerimientos del negocio incluyendo la navegación ventana a ventana, campo a campo y el uso de los métodos de acceso (movimientos del mouse y teclas).
	Técnica	Navegar en las pantallas desde el menú principal hasta el menú de nosotros haciendo uso del mouse y teclas para actualizar los datos.
	Resultados	Cada ventana cumple con el diseño establecido en la documentación técnica del sistema.
PRUEBAS DE ACEPTACIÓN	Objetivo	El usuario final debe verificar que todos los elementos del subsistema de nosotros funcionen adecuadamente
	Técnica	El usuario deberá realizar cambios y consultas de los datos del Laboratorio de Metrología basándose en el flujo del esquema jerárquico de la figura 3.20 y en el manual de usuario.
	Resultado	Las pruebas de cambios y consultas de los datos del Laboratorio de Metrología por el usuario final se realizaron satisfactoriamente.

PRUEBAS DE PAGINAS DE MANTENIMIENTO SUBSISTEMA METRÓLOGOS		
PRUEBAS UNITARIAS	Objetivo	Asegurar por parte del ingeniero de pruebas de la apropiada funcionalidad de la aplicación en la actualización del título de un metrólogo.
	Técnica:	Modificar en el subsistema de metrólogo el título de un investigador
	Resultados	La actualización del título del metrólogo se realizó exitosamente, enviando la página en la que aparece la información actual del metrólogo.
PRUEBAS INTEGRALES	Objetivo	El Ingeniero de pruebas de diseño debe :  Validar el diseño de páginas.  Verificar que la navegación desde el menú principal hasta el menú de metrólogo refleje las funciones y requerimientos del negocio incluyendo la navegación ventana a ventana, campo a campo y el uso de los métodos de acceso (movimientos del mouse y teclas).  Generar reporte de lista de metrólogos.
	Técnica	Navegar por el subsistema de metrólogos.  Generar el reporte de listado de metrólogos.
	Resultados	Cada ventana cumplió con el diseño y los requerimientos del laboratorio de metrología establecidos.  El reporte generado de metrólogos contiene la información solicitada.
PRUEBAS DE ACEPTACIÓN	Objetivo	El usuario final debe verificar que todos los elementos del subsistema de metrólogos funcionen adecuadamente
	Técnica	El usuario deberá realizar cambios y consultas de los metrólogos del Laboratorio de Metrología basándose en el flujo del esquema jerárquico de la figura 3.20 y en el manual de usuario.
	Resultado	Las pruebas de cambios y consultas de los metrólogos del Laboratorio de Metrología por el usuario final se realizaron satisfactoriamente.

PRUEBAS DE PAGINAS DE MANTENIMIENTO SUBSISTEMA ORDENSM		
PRUEBAS UNITARIAS	Objetivo	Asegurar por parte del ingeniero de pruebas de ejecución la apropiada funcionalidad de la aplicación en la inserción de una ORDENSM generada por una solicitud de cotización en Internet.
	Técnica:	Identificar el número de cotización solicitada desde Internet que va generar la ORDENSM
	Resultados	Se ingresó la ORDENSM exitosamente con su número consecutivo correcto y con todos los datos de la cotización, agregando la fecha de entrega y el responsable del servicio.
PRUEBAS INTEGRALES	Objetivo	El Ingeniero de pruebas de diseño debe :  Validar el diseño de páginas.  Validar la generación del reporte de búsqueda de ordenes SM.  Validar la creación del formato CISM-01 para impresión .  Verificar que la navegación desde el menú principal hasta el menú de ordenSM a través de los elementos reflejan las funciones y requerimientos del negocio incluyendo la navegación ventana a ventana, campo a campo y el uso de los métodos de acceso (movimientos del mouse y teclas).
	Técnica	Navegar en las pantallas desde el menú principal hasta el menú de ordenSM haciendo uso del mouse y teclas para actualizar los datos.  Realizar búsquedas de ordenes SM, bajo los diferentes criterios de consulta.  Solicitar hacer una impresión del formato CISM-01 para impresión .
	Resultados	Cada ventana cumple con el diseño establecido.  Las búsquedas de ordenes SM fueron exitosas.  La impresión del formato CISM-01 fue realizada correctamente.
PRUEBAS DE ACEPTACIÓN	Objetivo	El usuario final debe verificar que todos los elementos del subsistema de ordenSM funcionen adecuadamente
	Técnica	El usuario deberá realizar cambios, consultas e impresiones de las ordenes SM basándose en el flujo del esquema jerárquico de la figura 3.20 y en el manual de usuario.
	Resultado	Las pruebas de cambios, consultas e impresiones de las órdenes SM por el usuario final se realizaron satisfactoriamente sin que la aplicación falle.



PRUEBAS DE PAGINAS DE MANTENIMIENTO SUBSISTEMA COTIZACIONES		
PRUEBAS UNITARIAS	Objetivo	Asegurar por parte del ingeniero de pruebas de ejecución la apropiada funcionalidad de la aplicación en construcción del reporte de búsqueda de las cotizaciones por empresa.
	Técnica:	Generar el reporte de búsqueda de las cotizaciones por empresa.
	Resultados	Se generó exitosamente el reporte de cotizaciones por empresa.
PRUEBAS INTEGRALES	Objetivo	El Ingeniero de pruebas de diseño debe:  Validar el diseño de páginas.  Generar el reporte de las cotizaciones bajo los diferentes criterios de búsqueda.  Verificar que la navegación desde el menú principal hasta el menú de cotización a través de los elementos que reflejan las funciones y requerimientos del negocio incluyendo la navegación ventana a ventana, campo a campo y el uso de los métodos de acceso (movimientos del mouse y teclas).
	Técnica	Navegar en las pantallas desde el menú principal hasta el menú de nosotros haciendo uso del mouse y teclas para actualizar los datos y validar el diseño.  Realizar la búsqueda de una cotización bajo los diferentes criterios.
	Resultados	Cada ventana cumple con el diseño.  Los reportes fueron generados de acuerdo a lo solicitado.
PRUEBAS DE ACEPTACIÓN	Objetivo	El usuario final debe verificar que todos los elementos del subsistema de cotización funcionen adecuadamente
	Técnica	El usuario deberá realizar consultas de las cotizaciones basándose en el flujo del esquema jerárquico de la figura 3.20 y en el manual de usuario.
	Resultado	La pruebas de consultas de las cotizaciones por el usuario final se realizaron satisfactoriamente sin que la aplicación falle.

PRUEBAS DE PAGINAS DE PUBLICAS SUBSISTEMA DE SOLICITUD DE COTIZACIONES		
PRUEBAS UNITARIAS	Objetivo	Asegurar por parte del ingeniero de pruebas de ejecución la apropiada funcionalidad de la aplicación en la solicitud de cotización de servicio desde Internet.
	Técnica:	<p>Verificar los resultados esperados cuando se utilizan datos válidos en la solicitud de cotización desde Internet.</p> <p>Validar que existan mensajes de error o advertencia que son desplegados cuando se utilizan datos inválidos en la solicitud de cotización desde Internet.</p>
	Resultados	<p>Se logro ingresar correctamente la solicitud de cotización desde Internet obteniendo como resultado la cotización.</p> <p>Al tratar de ingresar una solicitud de cotización en Internet con datos inválidos se obtiene un mensaje de advertencia con la leyenda correspondiente.</p>
PRUEBAS INTEGRALES	Objetivo	<p>El Ingeniero de pruebas de diseño debe:</p> <p>Validar el diseño de páginas.</p> <p>Validar la cotización en Internet.</p> <p>Verificar que la navegación desde la página principal hasta la página de cotización a través de los elementos reflejan las funciones y requerimientos del negocio incluyendo la navegación ventana a ventana, campo a campo y el uso de los métodos de acceso (movimientos del mouse y teclas).</p>
	Técnica	<p>Navegar en las pantallas desde la página principal hasta la página de cotización haciendo uso del mouse y teclas para validar el diseño.</p> <p>Realizar la solicitud de cotización en Internet.</p>
	Resultados	<p>La navegación de pantallas cumplió con el flujo y el diseño establecido.</p> <p>Se realizó una solicitud de cotización exitosamente.</p>
PRUEBAS DE ACEPTACIÓN	Objetivo	El usuario final verificara que todos los elementos de la solicitud de cotización se cumpla.
	Técnica	El usuario realizara solicitud de cotización, basándose en el flujo del esquema jerárquico de la figura 3.20 y en el manual de usuario del Anexo A.
	Resultado	Las pruebas de solicitud de cotización por el usuario final se realizaron satisfactoriamente.

PRUEBAS DE PAGINAS PUBLICAS: SUBSISTEMA CONSULTA DE ORDENSM		
PRUEBAS UNITARIAS	Objetivo	Asegurar por parte del ingeniero de pruebas de ejecución la apropiada funcionalidad de la aplicación en la consulta de estado de la ordenSM.
	Técnica:	Verificar los resultados esperados cuando se utilizan datos válidos.  Validar que existan mensajes de error o advertencia que son desplegados cuando se utilizan datos inválidos.
	Resultados	Se realizó la consulta del estado de la ordenSM exitosamente con los datos válidos.  Se recibió el mensaje de error "ORDEN SM NO ENCONTRADA" al hacer la petición con datos inválidos.
PRUEBAS INTEGRALES	Objetivo	El Ingeniero de pruebas de diseño debe :  Validar el diseño de páginas.  Verificar que la navegación desde el menú principal hasta el estado de la ordenSM a través de los elementos que reflejan las funciones y requerimientos del negocio incluyendo la navegación ventana a ventana, campo a campo y el uso de los métodos de acceso (movimientos del mouse y teclas).
	Técnica	Navegar en las pantallas desde el menú principal hasta el estado de la ordenSM haciendo uso del mouse y teclas para validar el diseño y flujo de datos.  Realizar una consulta de estado de ordenSM..
	Resultados	La navegación de pantallas fue consistente en el sistema.  Cada ventana cumplió con el diseño solicitado.  La consulta se realizó exitosamente.
PRUEBAS DE ACEPTACIÓN	Objetivo	El usuario final verificara que todos los elementos de la consulta de estado de ordenSM.
	Técnica	El usuario realizara consultas de estado de ordenes SM, basándose en el flujo del esquema jerárquico de la figura 3.20 y en el manual de usuario del Anexo A.
	Resultado	La pruebas de consultas de ordenes SM por el usuario final se realizaron satisfactoriamente.

## CAPITULO 4. CONCLUSIONES

Para el desarrollo de la tesis fue necesario establecer metodologías de trabajo apoyándonos en las bases teóricas de ingeniería de software. El proyecto inicio con el análisis de la problemática del Laboratorio de Metrología con lo que respecta al proceso de *solicitud-cotización-inventario* determinando los requerimientos y necesidades del sistema a través de entrevistas y cuestionarios al personal involucrado en el proceso.

Una vez recopilada la información necesaria se definieron alcances del sistema y se evaluaron riesgos, para posteriormente reflejar el flujo de procesos y la transferencia de datos del Laboratorio de Metrología en diagramas de flujo de datos. Concluida la elaboración de los diagramas de flujo de datos se realizó el plan de trabajo en el que se consideraron las actividades, tiempos y recursos del proyecto.

Nuestra siguiente tarea fue crear entidades y relaciones auxiliándonos de los diagramas de flujo de datos, y del diccionario de datos. Aplicamos el proceso de normalización de la teoría de bases de datos relacionales con el fin de obtener un modelo consistente que evite redundancia de información.

En la siguiente etapa instalamos y configuramos en la computadora de escritorio asignada para el proyecto el sistema operativo LINUX, el servidor de Web APACHE, el manejador de base de datos relacional MySQL, y el lenguaje de programación PHP, para tener lista la infraestructura tecnológica necesaria para el proyecto.

Diseñamos las páginas públicas y de mantenimiento del prototipo presentado a los investigadores para su revisión. El prototipo de fue modificado con las observaciones hechas de diseño y construimos el resto de las páginas. En las siguientes revisiones se afino el prototipo hasta cumplir con todos las necesidades del Laboratorio de Metrología.

En la última etapa realizamos las diferentes pruebas establecidas por la metodología en las que se pudo comprobar la funcionalidad del sistema de cotización e inventario de acuerdo a los requerimientos originales.

De la metodología de ingeniería de software utiliza y resumida anteriormente concluimos que fue importante establecer una comunicación con los investigadores en la que logramos hacer una evaluación correcta de riesgos, y un plan de trabajo considerando todas las actividades, responsables y tiempos estimados.

Utilizar la herramienta de modelado de datos de DFD nos ayudo a visualizar y comprender fácilmente de un modo grafico la operación del Laboratorio de Metrología.

El modelo entidad/relación que desarrollamos para el Laboratorio de Metrología soporta nuevas funcionalidades del sistema como consultas condensadas por los clientes y reportes estadísticos en un futuro.

La creación de la base de datos logró centralizar la información además de optimizar búsquedas, actualizaciones e impresiones en tiempo real de cotizaciones, ordenes SM, servicios, colaboradores e información del Laboratorio de Metrología.

Con el desarrollo del sistema conseguimos reducir los tiempos del proceso solicitud de cotización del cliente de 48 horas a unos cuantos minutos, disminuyendo la intervención del personal administrativo del Centro de Instrumentos.

Anexar al sistema una plantilla de cotización al cliente estandarizada se redujo el margen de errores humanos en el cálculo de costos, además de prescindir de la presencia de los investigadores, para generarla permitiendo a Laboratorio de Metrología aumentar las posibilidades de negociación con nuevos clientes.

Otra de las funcionales del sistema es el manejo de un inventario, guardando la historia de servicios, ordenes SM y solicitudes.

El producto de software que desarrollamos tiene la característica de que un usuario con conocimientos básicos de navegación en Internet puede explotar fácilmente la información, en el caso de la sección de mantenimiento la información de la base de datos es actualizable a través de las páginas por los mismos investigadores, lo que le agrega autosuficiencia al Laboratorio de Metrología.

Es importante mencionar que el sistema da presencia en Internet al Laboratorio de Metrología en la difusión de los proyectos y servicios que se realizan en el Centro de Instrumentos en la Universidad Nacional Autónoma de México a través de un sitio Web.

Queremos destacar que usar software GNU en la tesis nos hizo partícipes de una sociedad que tiene como objetivo compartir el conocimiento, lo que nos dio la libertad de elegir la mejor opción en la construcción de un sistema robusto confiable con el mejor costo/beneficio para el Laboratorio de Metrología.

En nuestra búsqueda por encontrar la mejor alternativa de solución a la problemática planteada, adquirimos el conocimiento que nos permitió evaluar las alternativas de las metodologías existentes de ingeniería de software, y decidir utilizar el modelo en espiral que nos permitió hacer reuniones periódicas con los investigadores para hacer análisis de riesgo en entregas parciales del sistema.

Asimismo investigamos y conocimos las ventajas que ofrecían cada una de las opciones de software GNU en Internet, decidimos utilizar LINUX y APACHE por ser estables y robustos, elegimos MySQL como RDBMS por ser el más poderoso de entre los gratuitos existentes, en el caso del lenguaje de programación seleccionamos PHP por ser una interfaz sencilla y directa entre MySQL y APACHE.

---

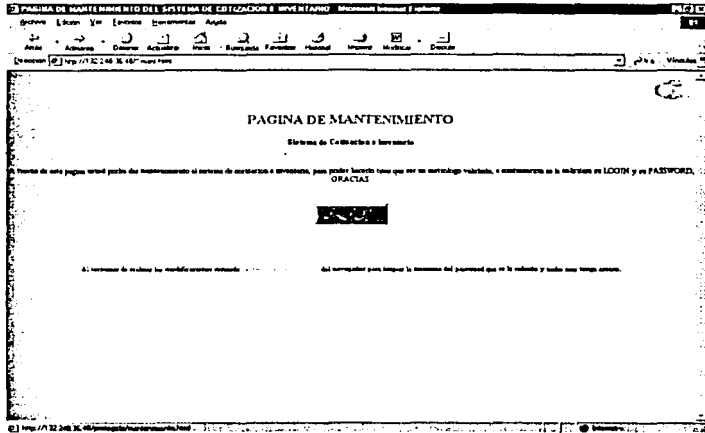
## BIBLIOGRAFÍA

- [1] C.J Date, 2001, Introducción a los sistemas de bases de datos, Prentice Hall, Séptima Edición, México.
- [2] Andrew S. Tanenbaum, Sistemas operativos distribuidos, Prentice Hall, Tercera Edición, México.
- [3] Andrew S. Tanenbaum, Redes de computadoras, Prentice Hall, Tercera Edición, México.
- [4] Luke Welling & Laura Thomson, PHP y Mysql Web Development, Sams, Third Edition, USA
- [5] Thomas A. Powell, HTML The complete reference, Osborne, Third Edition, USA
- [6] Gil, T., Principles of software Engineering Management, Adison Wesley, 1998, USA
- [7] Richard Peterson, Red Hat Linux The complete reference, Osborne, USA.
- [8] <http://www.mysql.com/>, Sitio de MySQL
- [9] <http://www.apache.org/>, The Apache Software Foundation
- [10] <http://www.php.net/>, Sitio de PHP
- [11] <http://hotwired.lycos.com/webmonkey/programming/php/tutorials/tutorial4.html>, Tutorial de integración de MySQL y PHP
- [12] <http://www.programacion.net/>, Sitio de tutoriales de creación de páginas.

## ANEXO. MANUAL DE USUARIO


### ACCESO AL SISTEMA

Desde cualquier navegador como Explorer o Netscape ingresar a la página <http://labmetrologiaunam.dhs.org/~mant.html> y presionar el botón **INGRESAR**:



A continuación debe ingresar su nombre de usuario y su contraseña para ser validado por el sistema:

**Escribe contraseña de red** [?] [X]

 Escribe su nombre de usuario y contraseña.

Sitio: 132.248.36.48

Dominio: Personal autorizado

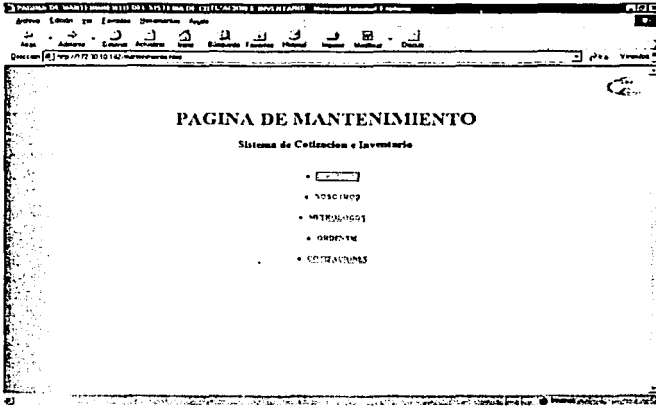
Nombre de usuario:

Contraseña:

Guardar esta contraseña en la lista de contraseñas

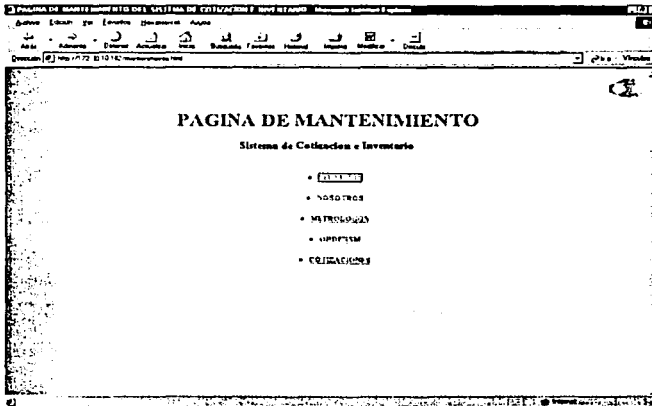


Una vez validado su nombre de usuario y su contraseña, aparecerá la **PAGINA DE MANTENIMIENTO**, en caso de denegar el acceso debe contactar a su administrador del sistema.

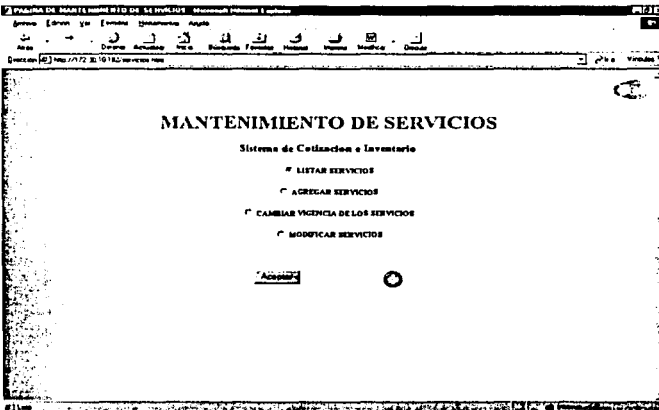


## CONSULTAR SERVICIOS

Desde la **PAGINA DE MANTENIMIENTO** seleccionar la liga **SERVICIOS**:



Después de seleccionar la liga *SERVICIOS* aparecerá la página de *MANTENIMIENTO DE SERVICIOS*:



Seleccionar la opción *LISTAR SERVICIOS* para obtener el listado de servicios:

Página 12 de 12 - 12/06/2015 - Mantenimiento Servicios Logotipo  
 Software [Inicio] [Salir] [Yol] [Cancelar] [Actualizar] [Ayuda]  
 Inicio [Actualizar] [Yol] [Cancelar] [Actualizar] [Ayuda]  
 Dirección [C:] \Programas\...

**SERVICIOS REGISTRADOS**

Los servicios se muestran con el Estado registrado

Numero	Nombre del Servicio	Inventariado	Precio	STATUS
1	BLOQUES PATRON PIEZA PIEZA ACILADA EN UNIDADES SI (mm)	0.0000	100.00	VIGENTE
2	BLOQUES PATRON CONJUNTO PIEZA EN CONJUNTO EN UNIDADES SI (mm)	0.0000	100.00	VIGENTE
3	BARRAS PATRON PIEZA INDIVIDUAL PIEZA ACILADA DE RAJTA 1000 mm DE 1 A 40 PULGADAS	0.0000	300.00	VIGENTE
4	BARRAS PATRON CONJUNTO PIEZA EN CONJUNTO DE RAJTA 1000 mm PIEZA EN CONJUNTO DE RAJTA 1000 mm solamente	0.0000	100.00	VIGENTE
5	PERNOS PATRON PIEZA PIEZA A TORNILLO DE RAJTA 1000 mm	0.0000	100.00	VIGENTE

## AGREGAR SERVICIO

Desde la página de **MANTENIMIENTO DE SERVICIOS** elegir la opción **AGREGAR SERVICIOS**:



A continuación aparecerá el formulario en donde se debe agregar la información del nuevo servicio, una vez terminada la captura presionar el botón **AGREGAR**:

The screenshot shows a window titled 'PROGRAMA DE SERVICIOS' with a menu bar containing 'Inicio', 'Edición', 'Ayuda', 'Salir', 'Nuevo', 'Actualizar', 'Eliminar', 'Actualizar', 'Eliminar', 'Actualizar', 'Eliminar'. The main content area is titled 'AGREGAR SERVICIOS'. Below the title is a note: 'Desde esta página puede agregar servicios, solo tener que llenar completamente los campos del servicio e indicar si es un servicio vigente o no, ningún campo puede quedar en blanco ( )'. The form contains the following fields: 'Nombre del Servicio', 'Descripción', 'Precio \$', 'Cantidad', 'Frecuencia de Pago', and 'Descripción Detallada'. At the bottom, there is a 'VICENTE' label and an 'AGREGAR' button.

Enviados los datos el sistema notificará que el servicio fue agregado exitosamente.

**SERVICIO AGREGADO:**

Numero	Nombre del Servicio	Intero-haberes	Precio	STATUS
44	Barra patron	0 5000	350 00	VIGENTE
Medición de barra patron				
Medición de barra patron por número				

## CAMBIAR VIGENCIA DE SERVICIO

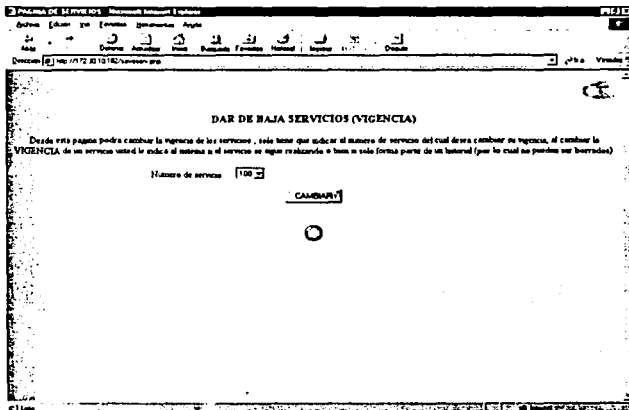
Desde la página de *MANTENIMIENTO DE SERVICIOS* elegir la opción *CAMBIAR VIGENCIA DE LOS SERVICIOS*:

**MANTENIMIENTO DE SERVICIOS**

Sistema de Contabilidad e Inventario

- LISTAR SERVICIOS
- AGREGAR SERVICIOS
- CAMBIAR VIGENCIA DE LOS SERVICIOS
- MODIFICAR SERVICIOS

En la lista de la página *DAR DE BAJA SERVICIOS* elegir de la lista desplegable el número de servicio al que se la va a cambiar la vigencia:

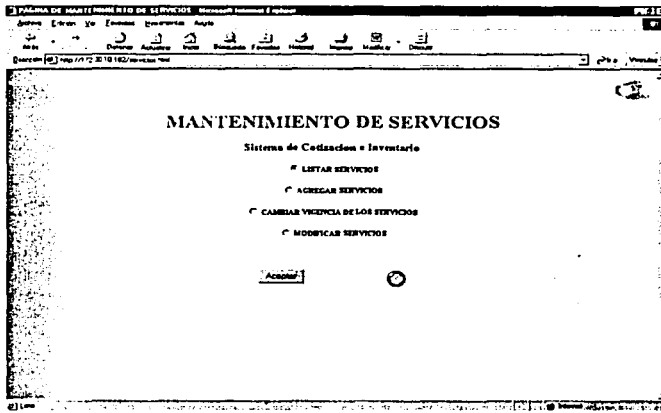


Posteriormente aparecerá la pantalla indicando el cambio realizado al servicio:

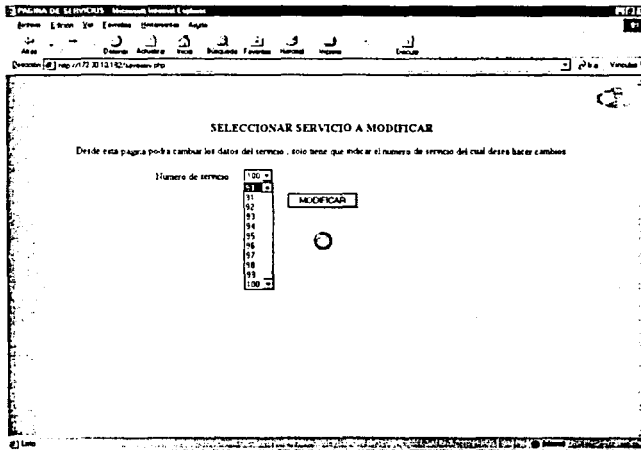
¡Error! Vínculo no válido.

## MODIFICAR SERVICIO

Desde la página de **MANTENIMIENTO DE SERVICIOS** elegir la opción **MODIFICAR SERVICIOS**:



Seleccionar de la lista desplegable el servicio a modificar.



Modificar la información necesaria en el formulario, y presionar el botón **MODIFICAR**:

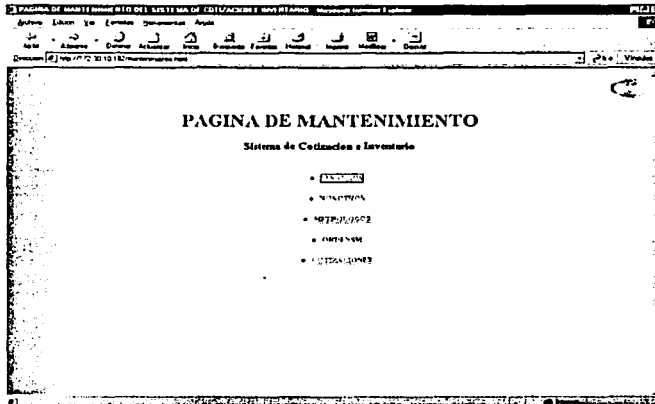




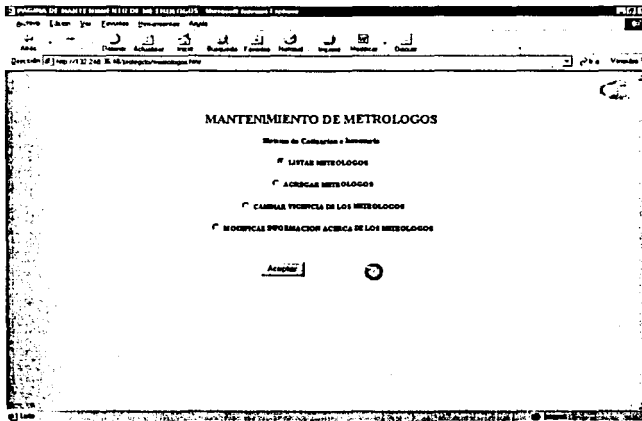


## LISTAR METRÓLOGOS

Desde la **PAGINA DE MANTENIMIENTO** seleccionar la liga **METRÓLOGOS**:



Una vez seleccionada la liga de **METRÓLOGOS** aparecerá la página de **MANTENIMIENTO DE METRÓLOGOS**, y elegir la opción **LISTAR METRÓLOGOS**:



A continuación aparecerá la página con la lista de metrólogos.

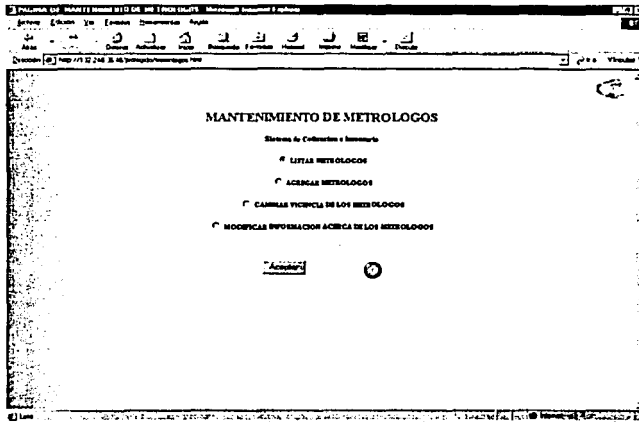
**METROLOGOS REGISTRADOS**

Los datos de cada metrólogo se muestran con el Estado siguiente:

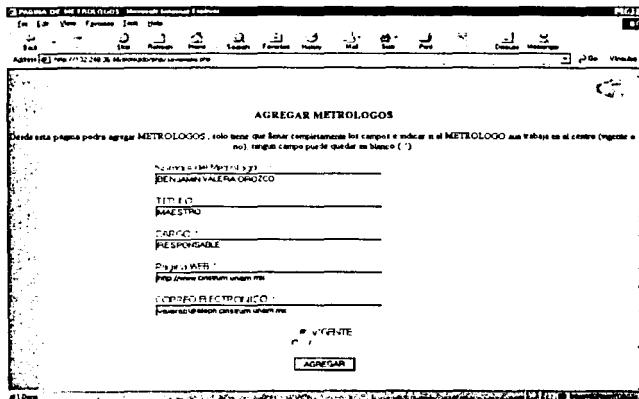
CLAVE	NOMBRE DEL METROLOGO	TITULO	CARGO	STATUS
1	Sergio Padilla Olvera	Ing.	Responsable Técnico	Activo
2	MIRIAM MORAN PACHECO MIRIAM YANCOOLAN MIRIAM@YANCOO.COM	ING	RESPONSABLE	INACTIVO

## AGREGAR METRÓLOGO

Desde la página de *MANTENIMIENTO DE METRÓLOGOS* elegir la opción de *AGREGAR METRÓLOGOS*:

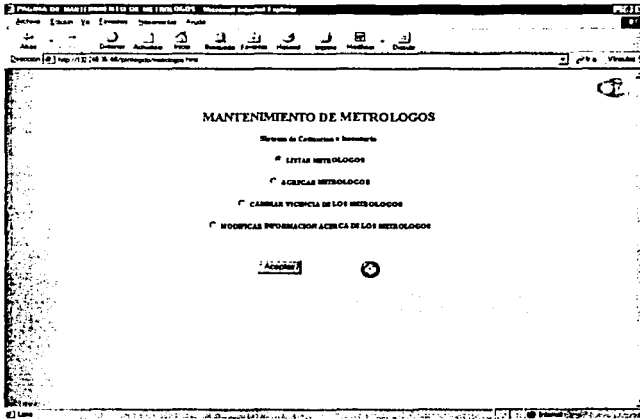


Ingresar los datos del metrólogo en el formulario de la página **AGREGAR METRÓLOGO**, y presionar el botón **AGREGAR**:

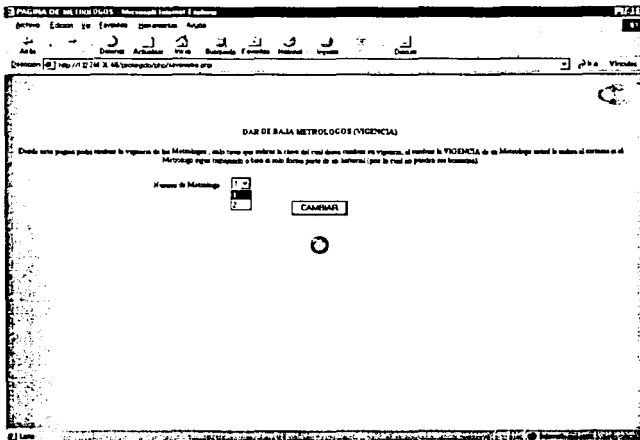


El sistema regresará la página de **METRÓLOGO AGREGADO** en donde muestra la información del metrólogo agregado.

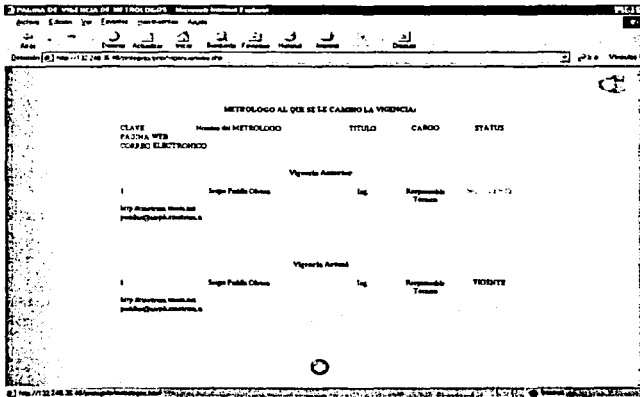




En la página **DAR DE BAJA METRÓLOGOS** seleccionar de la lista el número de metrólogo a quien se va a modificar la vigencia y presionar el botón **CAMBIAR**:

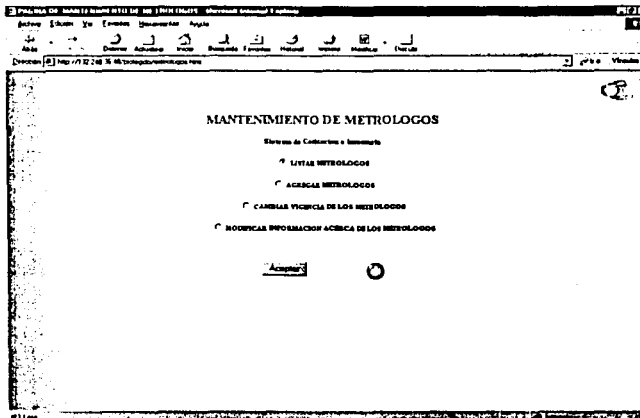


El sistema enviará la página en donde se mostrará los cambios realizados:



## MODIFICAR DATOS DEL METRÓLOGO

Desde la página de *MANTENIMIENTO DE METRÓLOGOS* elegir la opción de *CAMBIAR VIGENCIA DE LOS METRÓLOGOS*:



En el siguiente formulario cambiar la información que se requiera y presionar el botón **MODIFICAR**:

MODIFICAR METROLOGOS

Desde esta pagina puede cambiar los datos de los METROLOGOS, solo tiene que llenar completamente los campos e indicar si el METROLOGO esta trabajando en el centro, (ingrese o no) ningún campo puede quedar en blanco ( )

Nombre del Metrologo  
 Cargo  
 Titulo  
 Sexo  
 Cargo  
 Cedula  
 Cedula Profesional  
 Fecha de Emision

# VIGENTE

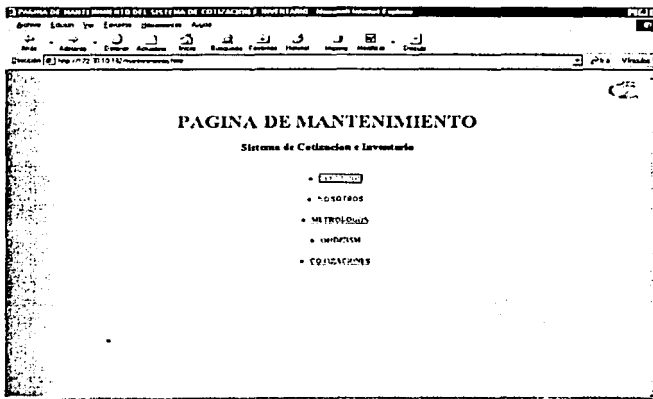
MODIFICAR

El sistema enviará una notificación de que los cambios se han realizado exitosamente.

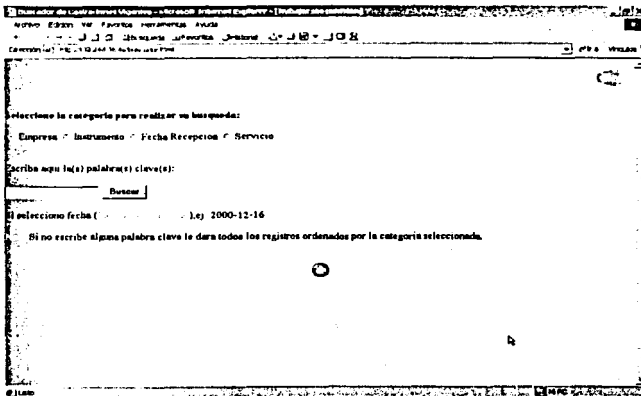
## CONSULTAR COTIZACIONES

Desde la **PAGINA DE MANTENIMIENTO** seleccionar la liga **COTIZACIONES**





En la siguiente página elegir el criterio de búsqueda, si el cuadro de texto esta vacío obtendrá todas las cotizaciones bajo ese criterio:



En la página **COTIZACIONES** desplegará un reporte, y presionar la liga de **ID** para obtener la cotización:

File Edit View Format Tools Help Window System | Internet Explorer

File Edit View Format Tools Help Window System | Internet Explorer

Address http://132.248.25.85/programa/imp/cotizaciones.php

## COTIZACIONES

Si usted desea acceder a cotizaciones por favor solista el campo ID

Empresa	Contacto	Domicilio	Colonia	Ciudad	Postal	Teléfono	Fax	E-mail	Nombre Del Contacto	Marcas
LENZ SA DE CV	LILIANA SALAZAR VAZQUEZ	CALLE P 848-1	UHAFR	MEXICO D.F	04800	56774791	56716320		Berta Patricia Compeán	Misocoro s
LENZ SA DE CV	LILIANA SALAZAR VAZQUEZ	CALLE P 848-1	UHAFR	MEXICO D.F	04800	56774791	56716320		Berta Patricia Compeán	Misocoro s
PISSA	MIRIAM MORAN	AV 4 103 Tepic	TEPUXTEPEC	MEXICO D.F	15010	50953700			plano	lacost s
PISSA	MIRIAM MORAN	Tepic	Loma Hermosa	MEXICO D.F	12354	50953700	50953701		torreanosa	misocoro s

© 2005

El sistema regresará la cotización enviada al cliente.

Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Stop Refresh Home Search Favorites History Mail See Print Edit Discuss Messenger

Address http://132.248.36.48/proyecto/cv/mecocapacion.php?id=

Go Virtual

Cd Unversitaria, 2 de Octubre de 2001

**Laboratorio de Metrología**

LILIANA SALAZAR VAZQUEZ  
 LIZWIZ S.A. DE C.V.  
 CALLE P #48-1,  
 PUJAR,  
 04800, MEXICO, D.F.

FAX 56716320  
 TELEFONO 56774791  
 E-Mail [lizwiz@yahoo.com](mailto:lizwiz@yahoo.com)

Atendiendo a su solicitud de calibración de los instrumentos que a continuación se mencionan, me permito informarle que nuestro *Laboratorio de Metrología*, tiene la capacidad de realizar dicho servicio con las siguientes características:

El costo por el servicio incluyendo informe con especificaciones es de

Cantidad: 5 Instrumento Barras Patron Conjunto Marca Minutoyo Precio Total \$ 500.00

**NOTAS:**

1. La UNAM no cobra I.V.A.
2. Esta cotización tiene vigencia de 30 días
3. El pago se efectúa en nuestras instalaciones y puede hacerse en efectivo o mediante cheque certificado o de ventanilla, o bien, mediante depósito bancario en la cuenta BANCOMER No. 1355809-3 Sucursal Pensar a nombre del "Centro de Instrumentos, UNAM" y enviar o traer ficha de depósito original.
4. El tiempo de entrega será de 5 días hábiles a partir de la recepción de los instrumentos en nuestro laboratorio para lo cual deberá haberse cubierto el pago correspondiente.
5. La recepción de los mismos puede ser de inmediato.
6. Si desea se le envíe el equipo y los uniformes de calibración por mensajería, deberá enviarnos una guía o-avada o, si es el caso, una carta de autorización.

Dado que la realización de una calibración de ésta naturaleza requiere extremo cuidado y atención especial le pedimos confirmar a la brevedad posible la aceptación de la presente para programar el trabajo dentro de las actividades del Laboratorio de Metrología

A t e n t a m e n t e

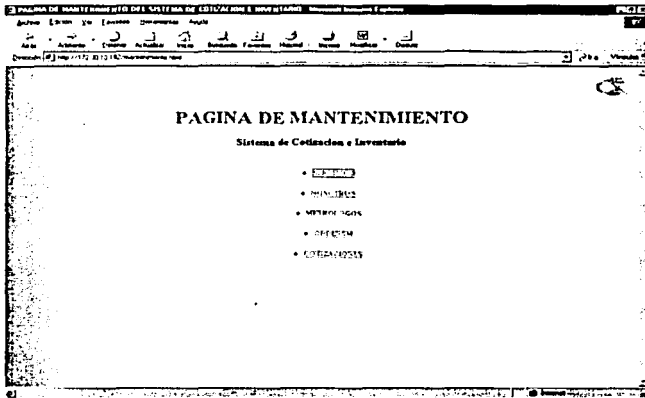
Sergio Padilla Olvera  
 Responsable Técnico

Tel 56 22 86 03 al FAX 56 22 86 24 E-mail: [lizwiz@yahoo.com](mailto:lizwiz@yahoo.com)  
 Pagina WEB <http://cnstrum.unam.mx>

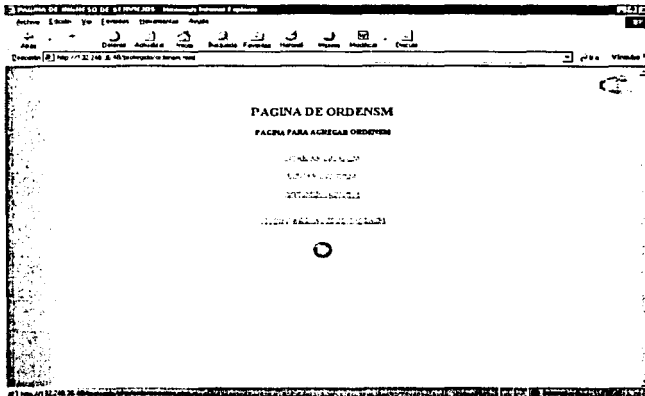
Done

## GENERAR ORDENSM MANUALMENTE

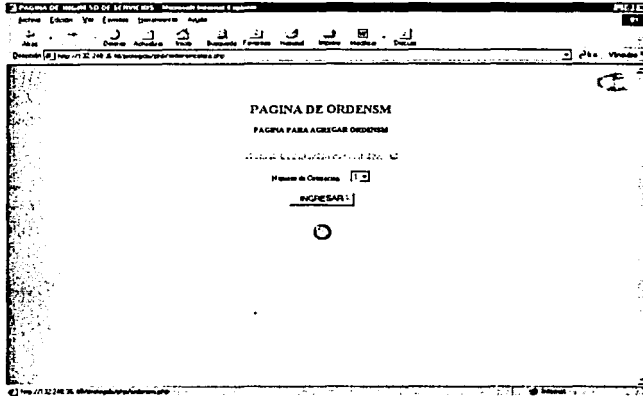
Desde la **PAGINA DE MANTENIMIENTO** seleccionar la liga **ORDENSM**:



Para ingresar Orden SM en la **PAGINA DE ORDENSM** seleccionar la liga **INGRESAR ORDENSM**:



Elegir la liga **USUARIO QUE INGRESA POR PRIMERA VEZ** y presionar el botón **INGRESAR**:



En la siguiente página se deben ingresar la información del cliente en la orden SM y presionar el botón enviar.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
CENTRO DE INSTRUMENTOS  
LABORATORIO DE METROLOGIA

ORDENSM #16971

CLIENTE

Nombre: \_\_\_\_\_

Apellido: \_\_\_\_\_

DIRECCIÓN

CALLE: \_\_\_\_\_ APO: \_\_\_\_\_ COLONIA: \_\_\_\_\_ CP: \_\_\_\_\_ ESTADO: \_\_\_\_\_

CELULAR: \_\_\_\_\_ FAX: \_\_\_\_\_ CUBA: \_\_\_\_\_ OTRO TELEFONO: \_\_\_\_\_

DATOS DEL INSTRUMENTO

Nombre del aparato: \_\_\_\_\_

Marca: \_\_\_\_\_ Modelo: \_\_\_\_\_ Año: \_\_\_\_\_

Características: \_\_\_\_\_

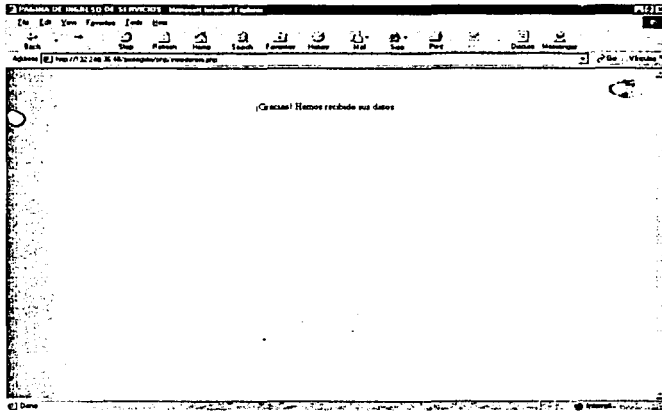
Forma de [v] uso: [v] Otro: \_\_\_\_\_

REF. LA ENTREGA DE ENTREGA

Sergio Padilla Urbina

**ENVIAR**

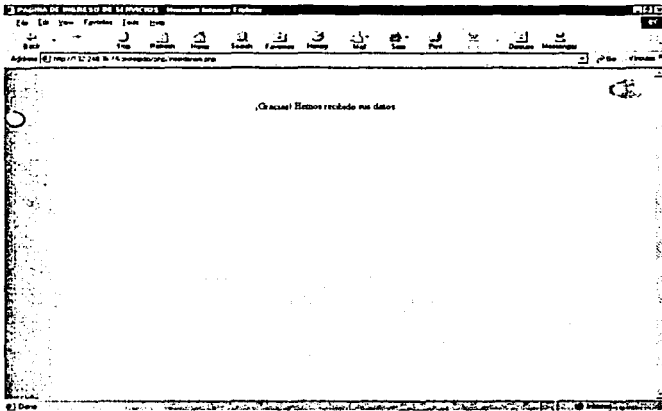
El sistema enviará la notificación de que los datos han sido guardados:



## GENERAR ORDENSM CON UNA COTIZACIÓN DE INTERNET

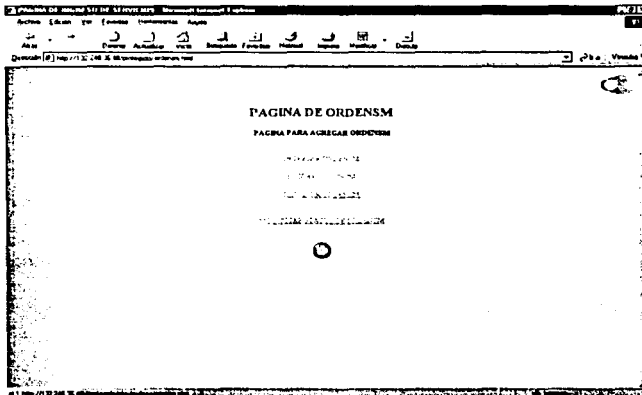
Si el cliente obtiene su cotización a través de Internet entonces se tiene la opción de generar la Orden SM automáticamente eligiendo en la página *PAGINA PARA AGREGAR ORDENSM* el número de cotización en la lista desplegable y presionar el botón *INGRESAR*:





## BUSCAR ORDENSM

Para ingresar Orden SM en la *PAGINA DE ORDENSM* seleccionar la liga *BUSCAR ORDENSM*:

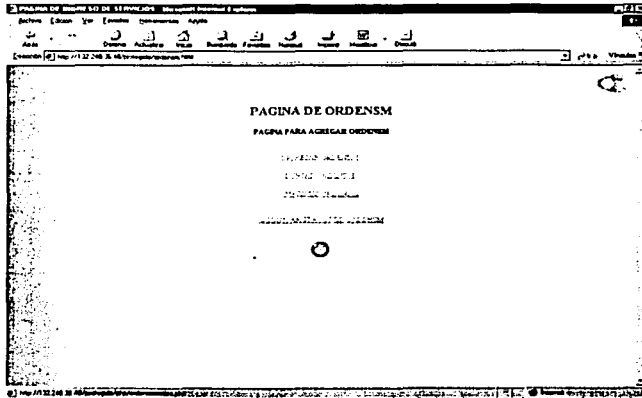




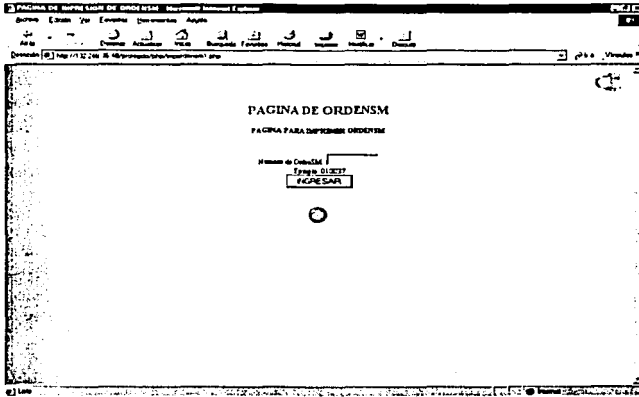


## IMPRIMIR ORDENSM

Para imprimir una Orden SM en la *PAGINA DE ORDENSM* seleccionar la liga *IMPRIMIR ORDENSM*:



En la caja de texto escribir el número de la orden SM a imprimir.



Aparecerá la página de la orden SM lista para imprimir:

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Stop Refresh Home Search Favorites History Mail Size Print Discos Messenger

Address http://132.248.36.48/intergeo/pnc/mpoordenm.php

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
CENTRO DE INSTRUMENTOS  
CIRCUITO EXTERIOR CIENBARRAS VEGETARIARIA, APDO POSTAL 19104, COTACACAN, CP 04310, MÉXICO D.F.

**LABORATORIO DE METROLOGÍA      ORDEN SM 010001**

EMPRESA/INSTITUCION SOLICITANTE: LINEA SA DE C/P  
RESPONSABLE: LILIANA SALAZAR PARQUEZ  
DOMICILIO: CALLE P #48-1 UMAPR 04800 MÉXICO D.F.  
CALLE COLONIA CP ESTADO  
TELEFONO: 56774791 FAX 56716320 EMAIL: lina@yaho.com FECHA RECEPCION: Octubre 2, 2001

**DATOS DEL INSTRUMENTO**

NOMBRE: Barras Patron Conjunto  
NOMBRE DEL SERVICIO: BARRAS PATRON CONJUNTO  
MARCA: Metuloyo MODELO: vsta SERIE: 0198716548459  
MATERIAL: plastico CONJUNTO DE SFZAS OTRAS ESPECIF

FECHA ESTIMADA DE ENTREGA: 2012-10-01  
ECLIDI DE CONFORMIDAD INSTRUMENTOS, EQUIPO: Sergio Padilla Olvera

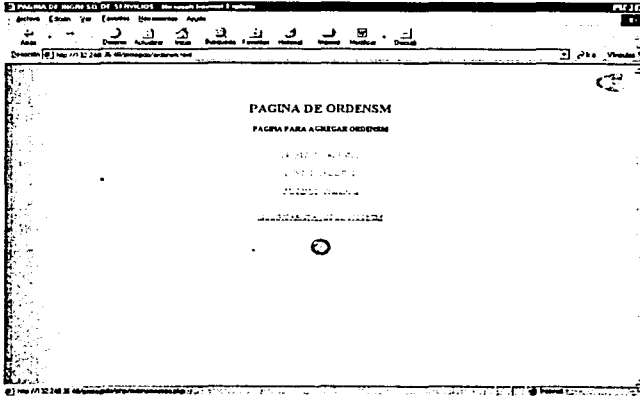
<b>CENTRO DE INSTRUMENTOS</b>	<b>CENTRO DE INSTRUMENTOS</b>
CIRCUITO EXTERIOR CD UNIVERSITARIO APDO POSTAL 49 104, COTACACAN, 04310, MÉX. D.F. TEL: 36 22 64 20 FAX: 36 22 64 03	<b>LABORATORIO DE METROLOGÍA</b>
<b>LABORATORIO DE METROLOGÍA</b>	<b>INSTRUMENTO</b>
COMPROBANTE PARA EL CLIENTE	ORDEN NUMERO
ORDEN: SM010001	SM010001

FECHA DE RECEPCION: \_\_\_\_\_  
FECHA ESTIMADA DE ENTREGA: 2012-10-01  
METROLOGO: Sergio Padilla Olvera      FOLIO: \_\_\_\_\_  
OBSERVACIONES: NO SE ESPERARÁ EL INSTRUMENTO SI PIERDE SIN LA RESERVA DE ESTE COMPROBANTE.  
FORMA NO SE RESPONSABILIZA DEL MAL USO QUE SE HAGA DE ESTE COMPROBANTE.      FOLIO: C12M-01

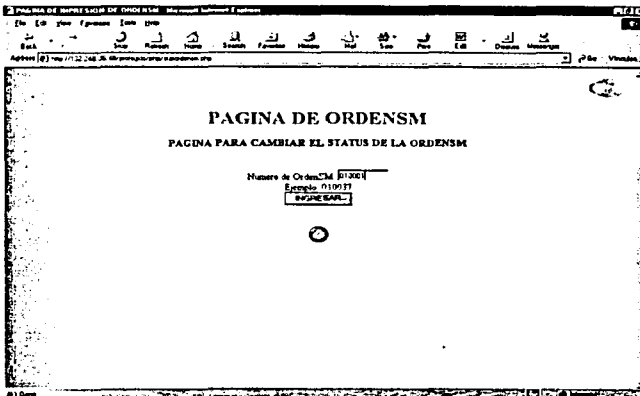
Done

## MODIFICAR STATUS DE LA ORDENSM

Para imprimir una Orden SM en la *PAGINA DE ORDENSM* seleccionar la liga *MODIFICAR STATUS DE LA ORDEN SM*:



Ingresar el número de orden SM a la que se va cambiar su estado



Aparecerá la página en donde se muestra el cambio hecho.

ORDENSM

Ordenes	Fecha de Recibo	Instrumento	Clave del Servicio
00010001	2001-10-02	Estado de México	4

TESIS CON  
FALLA DE ORIGEN