

41132  
29



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
ARAGÓN**

**AUTOMATIZACIÓN DEL CONTROL DE  
ENTRADA Y SALIDA DE EQUIPO DE  
CÓMPUTO DE LA FACULTAD DE  
INGENIERÍA**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:  
**INGENIERO EN COMPUTACIÓN**  
P R E S E N T A N :  
**GONZÁLEZ PÉREZ / DAVEY**  
**PÉREZ OLIVO ALBERTO JENNY**

ASESOR:  
ING. NORMA ELVA CHÁVEZ RODRÍGUEZ

TESIS CON  
FALLA DE ORIGEN

MÉXICO

2003

A



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**

## **AGRADECIMIENTOS**

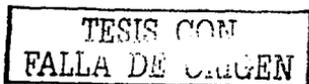
*Hoy en la celebración de mi titulación como ingeniero en computación quiero dar las gracias a todas aquellas personas que me han apoyado y formado durante mi crecimiento, ofreciéndome todo su amor y cariño.*

*Además agradecer a una persona que me ha dado su apoyo incondicional, esta persona es el Ing. Diego González R. mi Padre.*

*A mis hermanos Diego, Ricardo y Mariana, decirles que los quiero mucho y aunque no estemos juntos siempre están en mí, y comentarles que todo se puede realizar en esta vida con dedicación y esfuerzo, no importa que tan grande sea el obstáculo siempre hay una forma de poder seguir adelante.*

*A todos los seres queridos y amigos que me hacen falta les digo gracias.*

*A mi institución la UNAM, que me ha formado académicamente proporcionándome los elementos necesarios para poder desarrollarme profesionalmente.*



**Davey González Pérez**

B

## *AGRADECIMIENTOS*

*Gracias a Dios por darme la vida, por darme una familia, por guiarme siempre por un buen camino, por estar siempre a mi lado, cumpliendo todos mis sueños... Gracias padre....*

*Gracias a mis padres por haberme dado su apoyo incondicional en todo momento de mi vida, empujándome siempre hacia delante, dándome lo mejor de ellos, su amor, haciendo de mí lo que ahora soy, mil gracias, los amo, que Dios los bendiga...*

*A mi novia Montserrat que durante todo este tiempo a estado conmigo en la buenas y en las malas, ayudándome a conseguir todas mis metas, muchas gracias por darme todo ese cariño y amor, también por estar conmigo. Te amo... Faltas tu y espero que sea pronto, cuentas conmigo...*

*A mi hermano y su familia, decirles que los quiero mucho, que sigan adelante y deseo que el próximo en esto seas tu... Ya falta poco...*

*A todos mi tios, familiares y amigos que han contribuido a lo largo de mi vida y al logro de este trabajo, gracias por todo...*

*A mi institución, la UNAM y a mis profesores, por darme la oportunidad de extraer sus conocimientos y de proporcionarme los elementos necesarios para la vida profesional...*

*Alberto Jenny Pérez Olivo*

TESIS CON  
FALLA DE ORIGEN

C

**Indice**

	<i>Página</i>
Introducción.	1
<b>CAPITULO 1</b>	
<b>CONCEPTOS DE PROGRAMACIÓN E INGENIERÍA DE PROGRAMACIÓN.</b>	
1.1 Introducción.	3
1.2 Evolución del software.	4
1.2.1 Características del software.	6
1.2.2 La problemática en el desarrollo del software.	7
1.3 Definición de ingeniería de programación.	8
1.4 Modelos de procesos.	8
1.4.1 Ciclo de vida.	10
1.4.2 Prototipo.	11
1.4.3 Herramientas CASE o de cuarta generación.	12
<b>CAPITULO 2</b>	
<b>ESTUDIO DE LAS NECESIDADES DEL SISTEMA.</b>	
2.1 Introducción.	15
2.2 Antecedentes.	15
2.3 Necesidades.	16
2.4 Modelo conceptual actual.	17
2.5 Propuesta.	18
2.6 Modelo conceptual propuesto.	19
2.7 Tecnología a utilizar.	20
2.7.1 Servidor web apache.	21

TESIS CON  
FALLA DE ORIGEN

2.7.2 PHP.	23
2.7.3 MySQL.	29
2.7.4 HTML.	31
2.7.5 Lector optico.	34
2.8 Conclusiones del capitulo.	38

**CAPITULO 3****ANÁLISIS DE REQUERIMIENTOS.**

3.1 Técnica de especificación de requerimientos.	40
3.2 Análisis estructurado.	41
3.2.1 Modelo ambiental.	41
3.2.2 Modelo de comportamiento.	43
3.2.3 Modelo Entidad-Relación.	44
3.3 Creación de los diagramas de flujo de datos.	49
3.3.1 Diagrama de Contexto.	49
3.3.2 Diagrama de flujo de datos nivel "0".	51
3.3.3 Diagrama de flujo de datos nivel "1".	53
3.3.1 DFD nivel "1" Asignar equipo de computo.	54
3.3.2 DFD nivel "2" Procesar solicitud de salida.	55
3.3.3 DFD nivel "3" Registrar salida de equipo de computo.	56
3.3.4 DFD nivel "4" Generar etiquetas de código de barras.	57
3.4 Diagrama jerárquico funcional.	58
3.5 Diccionario de datos.	59
3.6 Diagrama Entidad-Relación.	79

**CAPITULO 4****DISEÑO DEL SISTEMA.**

4.1 Introducción.	81
4.2 Fases del diseño del sistema.	82
4.3 Diseño físico.	84
4.3.1 Descripción del software empleado.	85
4.3.2 Cálculo del espacio físico de la base de datos.	89
4.3.3 Creación del script.	93
4.4 Interfase.	96

**CAPITULO 5****DESARROLLO DEL SISTEMA.**

5.1 Programación.	109
5.2 Pruebas.	114
5.3 Instalación.	117
5.4 Seguridad.	120
5.5 Documentación.	123
5.5.1 Pantallas.	123
Conclusiones.	161
Bibliografía.	163

**Definición del problema**

Actualmente en las instalaciones de la Facultad de Ingeniería de la UNAM el registro de la salida del equipo de cómputo a su cargo para el servicio de préstamo se realiza de manera manual, es decir, la persona encargada de la vigilancia del edificio debe llenar a mano unas hojas de control interno que se requieren para poder sacar el equipo de las instalaciones. El problema de este procedimiento reside principalmente en que en ocasiones el equipo se extravía, no es fácil localizar un equipo de cómputo cuando se tiene en préstamo, en reparación, etc., siendo a veces necesario que el personal tenga que verificar físicamente en bodega para corroborar la existencia de dicho equipo, lo que trae como consecuencia pérdidas materiales y económicas no sólo a la Facultad de Ingeniería sino también a la UNAM.

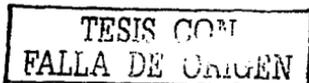
El sistema a desarrollar reducirá los posibles errores humanos al momento de registrar la entrada y/o salida del equipo mediante el uso de la tecnología de lectura de código de barras, así como el control del inventario y los movimientos del equipo con la implementación de una base de datos.

**Alcance**

Consideramos que se puede llegar a una solución satisfactoria del problema que implica el control de entrada y salida de equipo de cómputo en forma manual por medio de la utilización del sistema que hemos propuesto. El alcance del sistema a desarrollar se limita sólo al control de equipo de cómputo en los laboratorios de la Facultad de Ingeniería.

**Conexión con otros problemas**

Al automatizar la entrada y salida del equipo de cómputo de la Facultad de Ingeniería, creemos que puede ser de utilidad a las otras instituciones de la UNAM que aún no cuenten con un sistema que controle dicho proceso, es decir, que pueda ser de un uso generalizado dentro de la Universidad.



**Justificación**

El hecho de llenar las hojas de requerimientos a mano para permitir la entrada o salida del equipo provoca que el proceso sea un poco lento, además que cabe la posibilidad de registrar con error la clave del inventario en dichas hojas, perdiendo así la referencia real y el control que se busca obtener con el llenado de las formas. La pistola lectora de código de barras agilizará este proceso y además disminuirá los posibles errores en la captura del número de inventario del equipo, teniendo un registro más confiable y actualizado sobre los movimientos e incluso descripción detallada de dicho equipo. También se debe tomar en cuenta que en corto plazo el control adecuado de las entradas y salidas de equipo de cómputo puede redituarse en que se reduzcan las pérdidas de este equipo, dando como resultado un ahorro en el presupuesto de la Facultad de Ingeniería.

**Relevancia**

Cuando se cuenta con un buen registro y control del equipo del cual dispone una Institución, se puede apreciar con mayor exactitud las posibles acciones a tomar respecto a ciertos aspectos del mismo, como por ejemplo el tipo de equipo que existe (si se encuentra actualizado o es casi obsoleto), las razones por las que el equipo está saliendo, o el tiempo que se está tardando en regresar a la Facultad.

La información que proporcionará el sistema puede ser útil para lograr el óptimo control y funcionamiento del equipo de cómputo que tiene a su cargo la Facultad de Ingeniería. Además podría ayudar para que los recursos económicos que se emplean en la sustitución de un equipo de cómputo extraviado se emplearan para otro propósito útil,

**Objetivo específico**

Automatizar el proceso de entrada y salida del equipo de cómputo de la Facultad de Ingeniería para un mejor control del inventario y los movimientos del mismo, coadyuvando al mejor aprovechamiento de los recursos asignados a la Facultad de Ingeniería; tomando como principio la creación de una base de datos que tendrá todas las características del equipo, permitiendo además etiquetarlo con códigos de barras que agilizará su registro.

TESIS CON  
FALLA DE ORIGEN

CAPÍTULO 1

# CONCEPTOS DE PROGRAMACIÓN E INGENIERÍA DE PROGRAMACIÓN

## 1.1 Introducción

Hablar de ingeniería en computación o ingeniería del software en forma detallada, puede resultar muy extenso y difícil de desarrollar para los fines del presente trabajo. En la introducción de esta tesis no se intenta describir a detalle algunos de los aspectos principales de la ingeniería de software, sino por el contrario, explicar de manera general la evolución de la misma a lo largo de su historia con el propósito de resaltar la importancia que tiene en la solución de problemas de sistemas, tales como el presente tema a desarrollar: "Automatización del proceso de entrada y salida de equipo de computo de la Facultad de Ingeniería".

### Reseña histórica

Durante los inicios y algunas décadas posteriores de la informática, el principal desafío era el desarrollo de hardware, logrando así que se redujera el costo de procesamiento y almacenamiento de datos procesados. Con el avance de los años surgieron nuevas tecnologías, apareciendo la microelectrónica, la cual dio como resultado que se tuviera una mayor potencia de cálculo y se redujeran los costos de estos dispositivos de procesamiento y almacenamiento de los datos. Hoy esto es lo contrario y el principal desafío es mejorar la calidad y reducir el costo de las soluciones basadas en computadoras, soluciones que se implementan con los diversos tipos de software.

Actualmente el software ha superado al hardware como la clave del éxito de muchos sistemas basados en computadoras, sean estos para mantener un negocio,

3

TESIS CON  
FALLA DE URGEN

controlar el proceso de desarrollo de un producto, para que surjan nuevos sistemas y nos simplifique tareas repetitivas dentro de un proceso.

## 1.2 Evolución del Software

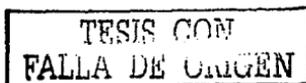
El contexto en el que se ha desarrollado el software está fuertemente ligado a las cinco décadas de evolución de los sistemas informáticos. A continuación se explica brevemente la evolución del software:

### Evolución en los 50's

Durante los primeros años de desarrollo de las computadoras, el hardware sufrió continuos cambios, mientras que el software se contemplaba simplemente como un agregado. La programación de computadoras era un arte para el que existían pocos métodos sistemáticos y el desarrollo del software se realizaba sin ninguna planificación (en donde los costos crecían y los planes eran un descalabro). La mayoría del software que se desarrollaba en ese entonces, era utilizado por la misma persona u organización: la misma persona lo escribía, lo ejecutaba y, si fallaba, depuraba el error que tenía. Dado lo anterior, el diseño de software era un proceso implícito, no tenía secuencia alguna ya que el usuario (desarrollador) solamente modificaba las líneas de código en donde posiblemente se encontraba el error, y la documentación normalmente no existía. Aquí el desarrollo de software era igual al desarrollo de un programa, debido a que todavía no existía la ingeniería de software.

### Evolución en los 60's

En esta época 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 complejidad del hardware y software. Esta época se caracterizó también por el establecimiento del software como producto y



la llegada de las denominadas "casas de software", donde el software ya se desarrollaba para tener una amplia distribución en un mercado.

### **Evolución en los 70's**

En esta década surgen las computadoras personales, el procesamiento distribuido (múltiples computadoras, cada una ejecutando funciones concurrentemente y comunicándose con alguna otra) incrementó notablemente la complejidad de los sistemas informáticos adoptando así un método de programación estructurado el cual permitió la creación de una variedad de sistemas.

Todos esos programas tenían que ser corregidos cuando se detectaban fallos, modificados por diferentes personas cuando cambiaban los requisitos de los usuarios o adaptados a nuevos dispositivos de hardware que se hubieran adquirido; a estas actividades se llamaron colectivamente mantenimiento del software. El esfuerzo gastado en el mantenimiento del software comenzó a absorber recursos en una medida alarmante, aún peor, la naturaleza personalizada de muchos programas los hacía virtualmente imposibles de mantener. Había comenzado una "crisis del software".

### **Evolución en los 80's**

La creación de las redes de área local y de área global, las comunicaciones digitales de gran ancho de banda y la creciente demanda de acceso "instantáneo" a los datos, hicieron una fuerte presión sobre los desarrolladores del software. Se produce la llegada y el amplio uso de los microprocesadores. Las computadoras personales han sido el catalizador del gran crecimiento de muchas compañías de software. Mientras que las compañías de software a finales de los 60's vendían cientos o miles de copias de sus programas, las compañías de software de los 70's venden decenas e incluso centenares de miles de copias, dado que cuentan con un método de programación orientado a objetos que hace más eficiente la creación de nuevo software. El desarrollo de infraestructura cliente-servidor hace que haya un aumento exponencial en el número

y tipos de usuarios, la cual genera que se demande aplicaciones en diversas disciplinas.

### **Evolución en los 90's hasta nuestros días**

Dada la masificación del uso de las computadoras personales, la invención de las redes y aplicaciones multimedia, hay un enorme crecimiento de Internet y el web introduciendo así nuevos tipos de aplicaciones como: comercio electrónico, educación a distancia, bibliotecas virtuales, etc.

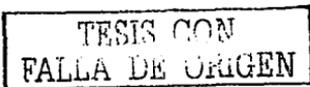
Las tecnologías orientadas a los objetos están desplazando rápidamente a enfoques de desarrollo de software más convencionales en muchas áreas de aplicación.

Pero aún en esta década, continúan intensificándose los problemas asociados con el desarrollo de software:

- La capacidad de construir nuevos programas no puede dar abasto a la demanda de nuevos programas
- La capacidad de mantener los programas existentes está amenazada por el mal diseño y el uso de recursos inadecuados
- Crecimiento de la demanda de nuevos productos
- Incumplimiento en los tiempos de entrega de los productos de software
- Incumplimiento de los presupuestos asignados a los proyectos
- Falta de métodos para la producción de software complejo
- Dificultad y altos costos para el mantenimiento del software existente
- Carencia de buenas especificaciones de requerimientos
- Dificultad para integrar nuevos participantes a los proyectos ya comenzados

#### **1.2.1 Características del Software**

Hace unas décadas, menos del uno por ciento de la gente podía describir de forma inteligente lo que significaba el "software de computadora". Hoy, la mayoría de las personas en general creen que entienden el software.



- El software se desarrolla, no se fabrica
- El software no se "estropea"
- La mayoría del software se construye a medida, en vez de ensamblar componentes existentes
- El software de computadora es información que existe en dos formas básicas: componentes no ejecutables en la máquina y componentes ejecutables en la máquina
- La "reusabilidad" es una característica importante para un componente de software de alta calidad
- Los componentes de software se construyen mediante un lenguaje de programación que tiene un vocabulario limitado, una gramática definida explícitamente y reglas bien definidas de sintaxis y semántica.

### 1.2.2 La problemática en el desarrollo del software

El desarrollo de software se ha caracterizado por estar fuera del plazo de entrega del producto, ser caro y contener errores.

El software no sólo implica programas computacionales asociados con alguna aplicación o producto, además, incluye la documentación necesaria para instalar, operar o usar, desarrollar y mantener dichos programas. Para sistemas grandes, a menudo el esfuerzo requerido para escribir esta documentación, es mayor que el requerido para el desarrollo de programas.

Suponiendo que el software entrega la funcionalidad requerida, existen cuatro atributos clave que un sistema de software de buena ingeniería debe poseer:

- El software debe ser mantenible. Dado que un software de larga vida útil está sujeto a cambios, debe ser escrito y documentado de modo que los cambios puedan ser efectuados sin incurrir en costos indebidos
- El software debe ser confiable. Esto significa que debe comportarse como lo esperan los usuarios y su frecuencia de fallas no debe ser mayor que lo indicado en su especificación

- El software debe ser eficiente. Esto no necesariamente significa que haya que lograr el grado más alto de rendimiento del sistema de hardware, ya que la maximización de rendimiento puede resultar en un software en que los cambios son laboriosos. Eficiencia significa que el sistema no debe malgastar el uso de los recursos del sistema tales como memoria y ciclos de procesador
- El software debe ofrecer una apropiada interfaz de usuario

### 1.3 Definición de ingeniería de programación

Todas las definiciones de ingeniería de software coinciden en que ésta trata de la construcción de sistemas de software mediante grupos o equipos de trabajo, más que individuos, utilizando principios de ingeniería en el desarrollo de estos sistemas, incluyendo tanto los aspectos técnicos como los no técnicos.

Algunas definiciones de autores contemporáneos son:

*Ingeniería de software (Bauer 1969)*

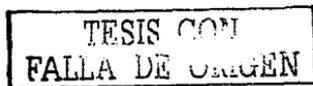
La ingeniería de software es el establecimiento y uso de asentados principios de ingeniería para la obtención económica de software fiable que funcione eficientemente en máquinas reales.

*Ingeniería de software (IEEE 1993)*

1. Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software
2. El estudio de los enfoques aplicables a la obtención de la definición 1

### 1.4 Modelos de procesos

La ingeniería de software emplea los modelos de procesos para lograr un enfoque organizado para el desarrollo de software. Para evitar cometer errores por requerimientos mal interpretados y por consecuencia una implementación incorrecta.



Hay numerosos modelos del proceso de desarrollo de software disponibles en la actualidad. No hay un modelo correcto para todos los propósitos y ningún modelo está completo como toda la abstracción en algún sentido desde los detalles de la realidad. Aún así, el uso de un modelo apropiado puede ayudar de manera considerable en el control de un proyecto de software. En la figura 1.1 se muestra un ejemplo de lo que tales modelos parecen.

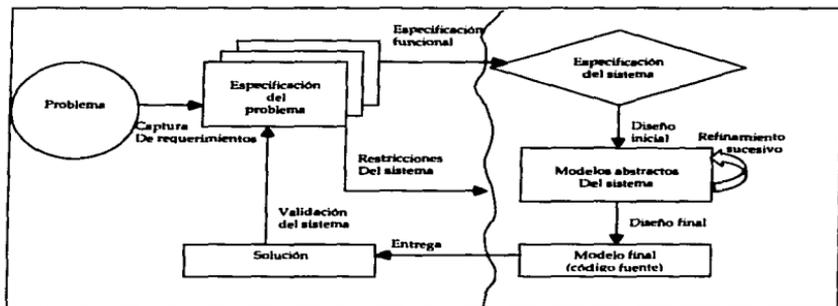


Figura 1.1 Comparación de los modelos con la realidad.

En este diagrama los aspectos orientados al problema del diseño (en los que el mundo real está involucrado, en el lado izquierdo) se distinguen de los aspectos orientados al diseño (en los que al modelar la abstracción es extrema, en el lado derecho). Esto es, sin embargo, sólo una de las muchas maneras de describir las varias actividades que involucran el desarrollo y el concepto de ciclo de vida del software.

Un modelo de ciclo de vida define el estado de las fases a través de las cuales se mueve un proyecto de desarrollo de software.

### 1.4.1 Ciclo de vida

Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas.

Un modelo de ciclo de vida del software:

- Describe las fases principales de desarrollo de software
- Define las fases primarias esperadas de ser ejecutadas durante esas fases
- Ayuda a administrar el progreso del desarrollo, y
- Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software

Así, los modelos por una parte suministran una guía para los ingenieros de software con el fin de ordenar las diversas actividades técnicas en el proyecto, por otra parte, suministran un marco para la administración del desarrollo y el mantenimiento, en el sentido en que permiten estimar recursos, definir puntos de control intermedios, monitorear el avance, etc. El ciclo de vida en cascada se muestra en la Fig. 1.2

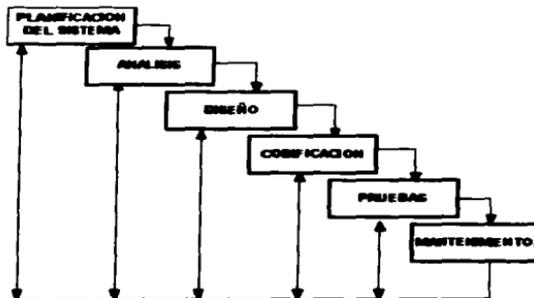


Figura 1.2 Modelo de ciclo de vida en cascada.

### 1.4.2 Prototipo

Como cualquier otra rama de la ingeniería, la ingeniería del software se apoya en la creación de modelos o prototipos para simplificar el estudio de un problema. Una forma de reducir los riesgos es construir sólo una parte del sistema, reservando otros aspectos para niveles posteriores.

El proceso de este modelo comienza con la recolección de requisitos:

- El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición
- Se realiza un diseño rápido, este se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente, por ejemplo enfoques de entrada y formatos de salida
- Luego el diseño rápido lleva a la construcción de un prototipo, el cual es evaluado por el cliente/usuario y utilizado para refinar los requisitos del software a desarrollo
- Finalmente, 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

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



Figura 1.3 Modelado por prototipo.

Uno de los factores que más influyen en el proceso de desarrollo de software y que prácticamente acompaña a toda aplicación es el hecho de que en su mayoría, no hay forma de tener todos los requerimientos corregidos antes del desarrollo del software. Muchas veces los requerimientos emergen a medida que la aplicación o partes de ella están en la etapa de experimentación práctica. En todos los casos, el trabajo comienza con la determinación de objetivos, alternativas y restricciones, paso que a veces se llama recolección preliminar de requisitos.

El prototipo es ampliamente recomendado para realizar la especificación de requerimientos, trata de minimizar los cambios en los requerimientos, mientras que el diseño modular minimiza el impacto de los cambios en los requerimientos.

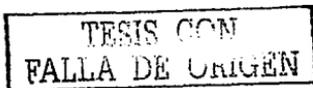
#### 1.4.3 Herramientas CASE o de cuarta generación.

En la etapa del análisis la ingeniería de software se apoya en herramientas CASE para aplicar la implementación del análisis, para esta tesis se utilizó una herramienta case del tipo Lower Case, concepto que se describe en esta sección, así como las características generales de este tipo de herramientas.

Las Herramientas de ayuda para el Desarrollo de Sistemas, surgieron para intentar dar solución a los problemas inherentes a los proyectos de generación de aplicaciones informáticas: plazos y presupuestos incumplidos, insatisfacción del usuario, escasa productividad y baja calidad de los desarrollos. Algunas de estas herramientas se dirigen principalmente a mejorar la calidad, como es el caso de las herramientas CASE (Computer Aided Software Engineering-Ingeniería de Software Asistida por Computadora). Otras van dirigidas a mejorar la productividad durante la fase de construcción, como es el caso de los lenguajes de cuarta generación (4GL-Fourth Generation Language).

Son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. El empleo de herramientas Case permiten integrar el proceso de ciclo de vida:

- Análisis de datos y procesos integrados mediante un repositorio



- Generación de interfases entre el análisis y el diseño
- Generación del código a partir del diseño
- Control de mantenimiento

La principal ventaja de la utilización de una herramienta CASE, es la mejora de la calidad de los desarrollos realizados y, en segundo término, el aumento de la productividad. Para conseguir estos dos objetivos es conveniente contar con una organización y una metodología de trabajo además de la propia herramienta.

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada. Podrían clasificarse de acuerdo a su funcionalidad y arquitectura de la siguiente forma:

- *Herramientas integradas, I-CASE (Integrated CASE, CASE integrado):* abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son llamadas también CASE workbench
- Herramienta(s) que comprende(n) alguna(s) fase(s) del ciclo de vida de desarrollo de software
- *Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior)* o front-end, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño
- *Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior)* o back-end, dirigidas a las últimas fases del desarrollo: construcción e implantación
- *Juegos de herramientas o toolkits*, son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento
- *Herramientas de soporte.* Se engloban en esta categoría las herramientas que recogen las actividades aplicables en todo el proceso de desarrollo, como las que se relacionan a continuación:
  - Herramientas de documentación
  - Herramientas para software de sistemas
  - Herramientas de control de calidad
  - Herramientas de bases de datos

- *Soporte de proyecto.* Este tipo de funciones hace referencia al soporte de actividades que se producen durante el desarrollo, derivadas fundamentalmente del trabajo en grupos, tales como facilidades de comunicación, soporte a la creación, modificación e intercambio de documentación, herramientas personales, controles de seguridad, etc. Los sistemas Case pueden conceder a estas cuestiones una importancia variable por lo cual el soporte de proyecto constituye un factor de diferenciación
- *Mejora continua de calidad.* Aunque frecuentemente se asocia a los sistemas Case con la mejora de la productividad en el desarrollo de aplicaciones, debe tenerse en cuenta que una de las principales ventajas estriba también, en la mejora de la calidad de los desarrollos realizados. Determinados sistemas Case enfatizan más sobre este punto que sobre el anterior, introduciendo herramientas que permiten ejercer un control intenso de garantía de calidad del software desarrollado desde las primeras fases de su ciclo de vida

CAPÍTULO **2**  
**ESTUDIO DE LAS NECESIDADES  
DEL SISTEMA**

### **2.1 Introducción**

En este capítulo se aplican los fundamentos de ingeniería de programación presentados en el capítulo anterior con el fin de modelar el sistema a desarrollar. Tal como se mencionó en el Capítulo 1, la combinación de los modelos de procesos nos permiten enriquecer el modelado del problema de tal forma que se pueda presentar un análisis lo más completo posible y que permita establecer los principales requerimientos y soluciones para el desarrollo del sistema en cuestión.

Este capítulo comienza con el establecimiento de los antecedentes que muestran la necesidad de automatizar el proceso actual de préstamo de equipo de cómputo en los Laboratorios de la Facultad de Ingeniería. Con el establecimiento de dichos antecedentes se da la pauta para la aplicación de los modelos de proceso expuestos en el capítulo anterior, que terminarán en el análisis del problema mismo; así como una propuesta de solución para finalmente comenzar con el desarrollo del sistema a través de un lenguaje de programación que se define en este mismo capítulo.

### **2.2 Antecedentes**

En los diversos departamentos de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México se cuenta con la infraestructura para poder cumplir con los objetivos de la institución que están orientados —entre otras cosas— a la enseñanza de los alumnos. Es por esto que uno de los principales elementos de apoyo para alcanzar estos objetivos se centra en el uso de los equipos de cómputo disponibles en

los departamentos y en los cuales, se desarrollan diversos proyectos escolares e incluso aquellos proyectos que se enfocan a un fin científico.

El equipo de cómputo se tiene disponible para el uso de personal académico de la institución y no es exclusivo de su utilización dentro de las instalaciones del departamento, ya que se tiene la posibilidad de préstamo fuera de las mismas con la elaboración de una solicitud de préstamo por parte del interesado.

El proceso de préstamo actualmente se realiza de forma manual y, como cualquier otro proceso de este tipo, es susceptible de errores, humanos principalmente. Como consecuencia se puede llegar a extravíar el equipo o perder el control de su localización. La solicitud involucra una orden de salida (normalmente un memorando) la cual es llenada por el solicitante interesado, después se procede a la recopilación de la firma o las firmas de la persona que autoriza dicha salida y finalmente el personal de vigilancia asignado en la entrada del departamento permite la salida, por concepto de préstamo, del equipo.

Todo este proceso aunque mantiene un control de las salidas de computadoras y periféricos, no es lo suficientemente robusto si se considera que el respaldo en el papel con las firmas autorizadoras corren el riesgo de extravío y al no contar con algún otro respaldo la situación se complica.

Un posible extravío de equipo provoca que los encargados de departamento queden señalados de tal forma que las siguientes asignaciones de equipo nuevo a su departamento no les sea autorizada debido a esos antecedentes de falta de control sobre los equipos a su cargo y que se prestan.

### 2.3 Necesidades

Dado que en los distintos departamentos de la Facultad de Ingeniería existe la situación de préstamos antes mencionada, se manifiestan a continuación los puntos que responden a las necesidades de automatización del proceso de préstamo de equipo de cómputo:

- Se presenta la necesidad de un sistema que controle el flujo de entrada / salida de equipo de cómputo del departamento
- Mantener registros del equipo existente
- La aplicación de los conocimientos de ingeniería en computación en el desarrollo de un sistema de este tipo en beneficio de los departamentos
- Un historial de órdenes de salida para consulta
- Un sistema que permita ejecutar la actividad de préstamo más rápida y eficientemente
- La automatización del proceso, que actualmente es manual

## 2.4 Modelo Conceptual Actual

El proceso que hoy en día se utiliza para poder prestar un equipo de cómputo del departamento mantiene el siguiente flujo:

1. Verificar la disponibilidad de un equipo determinado
2. Llenar la orden de salida
3. Recaudar las firmas correspondientes con las personas que autorizan la salida
4. Presentar el equipo y la orden firmada en la puerta de salida
5. Después del tiempo cumplido, devolver el equipo

La orden de salida se archiva para mantener el registro del préstamo del equipo de cómputo. Cabe mencionar que para el punto dos -el llenado de la orden- se requiere poner datos generales del solicitante, así como las características del equipo que se está prestando.

Los puntos descritos anteriormente se ilustran en la figura 2.1 en la cual se muestra el esquema actual del proceso de solicitud de préstamo de equipo de cómputo de un departamento específico de la Facultad de Ingeniería.

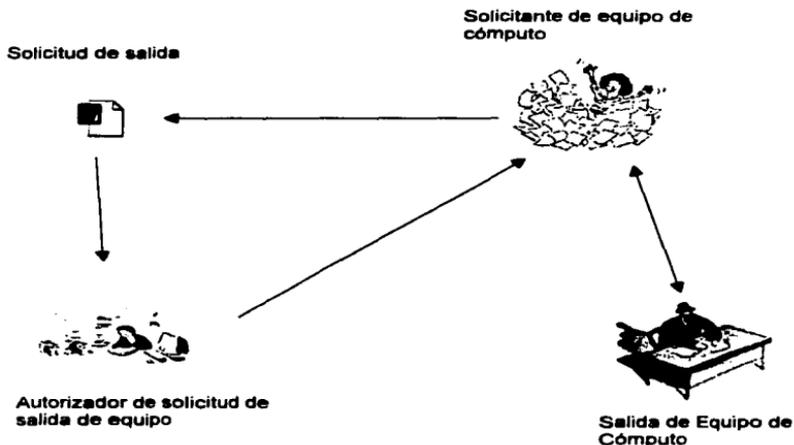


Figura 2.1 Modelo Conceptual Actual

## 2.5 Propuesta

Dada la situación en los préstamos de equipo de cómputo de los departamentos de la Facultad de Ingeniería, se propone a la Facultad desarrollar el sistema que seguirá el siguiente flujo:

1. Realizar la solicitud de salida utilizando el sistema desarrollado
2. Seleccionar el equipo de acuerdo al registro del mismo en el sistema
3. Enviar vía correo electrónico la solicitud al Jefe de Departamento para su autorización

4. El Jefe de Departamento autoriza la solicitud utilizando el sistema
5. Se registra fecha y hora de la salida del equipo mediante su número de inventario UNAM en el sistema
6. Se registra también lugar y responsable del lugar al que saldrá el equipo
7. Se regresa el equipo en la fecha acordada, registrando su disponibilidad nuevamente en el sistema
8. Toda esta información será histórica

Los puntos descritos anteriormente se ilustran en la Figura 2.2, en la cual se muestra el esquema propuesto del proceso de solicitud de préstamo de equipo de cómputo de un departamento específico de la Facultad de Ingeniería.

## 2.6 Modelo conceptual propuesto

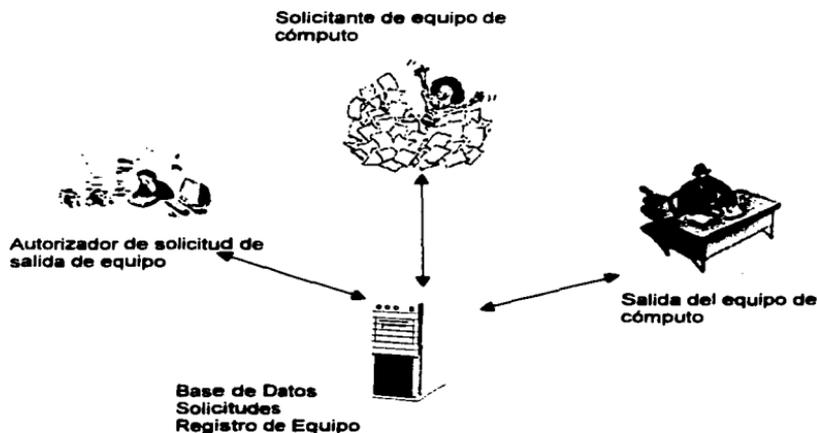


Figura 2.2 Modelo Conceptual Propuesto

## 2.7 Tecnología a utilizar

En este punto se plantean las características del software que se va a utilizar para el desarrollo del sistema propuesto.

Actualmente existen diferentes formas de desarrollar un sistema, para el caso de préstamo de equipo de cómputo de los departamentos de la Facultad de Ingeniería se ha elegido utilizar la tecnología Web por dos razones principales:

- a) El auge que ha mostrado en los últimos años y
- b) Se aprovecha la infraestructura de red interna o Intranet de los departamentos para un manejo versátil del sistema.

Muchas son las herramientas con las cuales se puede desarrollar e implementar un sistema en Web, sin embargo, para los fines que persigue el presente trabajo, se han elegido los siguientes productos por su facilidad de programación, flexibilidad y mejor manejo:

- |                              |                         |
|------------------------------|-------------------------|
| • Servidor web               | <b>Apache</b>           |
| • Scripts para servidor web  | <b>PHP</b>              |
| • Scripts para cliente web   | <b>html, JavaScript</b> |
| • Manejador de base de datos | <b>MySQL</b>            |

Antes de continuar con el proceso de análisis del sistema de préstamo de equipo de cómputo de los departamentos de la Facultad de Ingeniería, se procede a realizar una breve explicación de cada uno de los productos que se han elegido como plataforma para el desarrollo e implementación del sistema.

Es importante resaltar estos aspectos ya que con esto se tiene una panorámica global acerca de cada producto y que más adelante en los siguientes capítulos nos va a dar la razón de muchas cosas que se van a mencionar.

El uso de Apache como servidor web resulta favorable con el uso del lenguaje PHP pues este último se encuentra principalmente diseñado para ser ejecutado de una manera transparente por este servidor.

Por su parte, PHP ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales y utilizando ODBC a todas las bases de datos posibles en sistemas Microsoft. La interacción de PHP con datos se realiza por un lado a partir de las

funciones que PHP propone para cada tipo de base de datos y, por otro, estableciendo un diálogo a partir de un idioma universal: SQL (Structured Query Language).

La elección de MySQL como manejador de base de datos en nuestro sistema, responde a que éste complementa un kit de desarrollo para Web formado por Apache-PHP-MySQL.

Este kit al ser de software libre, evita la compra de licencias por uso de estos productos tanto a desarrolladores como a usuarios finales del sistema.

### 2.7.1 Servidor web Apache

El servidor web Apache es uno de los mayores triunfos del software libre. En diciembre de 1997 tenía una utilidad en el mercado cercana al 45% y en julio del 2000 ya estaba por encima del 60%, según los estudios de Netcraft, que ya se han establecido como la referencia dentro del mercado de servidores web. Esta es la primera cifra que hace que cualquier usuario que desee tener un servidor confiable tome Apache como el servidor de referencia<sup>1</sup>, tal y como puede apreciarse en la gráfica que se muestra en la Figura 2.3.

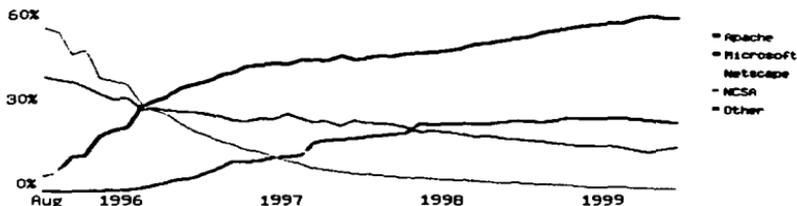


Figura 2.3 Crecimiento de Servidores Web Agosto de 1995 - Julio de 1999.

En julio de 1999 este estudio recibió respuestas de 6,598,697 sitios.

<sup>1</sup> Netcraft.com

Apache era inicialmente unos parches al servidor de web de NCSA (National Center for Supercomputing Applications) conocido como httpd (a principios de 1995). Al igual que GNU/Linux, fue un proyecto que atrajo a mucha gente por el gran interés de su objetivo: lograr el servidor web más rápido, más eficiente y con mayor funcionalidad desde el enfoque del software libre. Y ha sido un objetivo que se ha logrado poco a poco con la ayuda de muchos colaboradores alrededor del mundo.

Con un enorme equipo de voluntarios a lo largo y ancho de toda la red, se han logrado estos objetivos, logrando batir a compañías comerciales de la talla de Microsoft y Netscape. Y no sólo se ha logrado batir a los servidores web de grandes compañías: se ha logrado atraer incluso a IBM, que está apoyando Apache tanto a nivel de soporte como de desarrollo.

Apache es un servidor de red para el protocolo HTTP (Hyper Text Transfer Protocol), elegido para poder funcionar como un proceso standalone, sin que solicite el apoyo de otras aplicaciones o directamente del usuario. Para poder hacer esto, Apache, una vez que se ha iniciado, crea unos subprocesos (que normalmente son llamados procesos hijos, "children processes") para poder gestionar las solicitudes. Estos procesos, sin embargo, no podrán nunca interferir con el proceso mayor (llamado también proceso padre), aunque si puede suceder lo contrario, es decir, si se manda una señal de alto al proceso principal, todos los procesos hijos se detendrán también automáticamente.

Un típico árbol de procesos de Apache podría ser algo similar a:

USER	PID	%CPU	%MEM	SIZE	RSS	TTY	STAT	START	TIME	COMMAND
Root	203	0.01	1.01	4952	720	?	S	17.20	0.03	/usr/sbin/apache
user	212	0.00	2.03	5012	1456	?	S	17.20	0.00	\_ /usr/sbin/apache
user	213	0.00	2.02	5008	1424	?	S	17.20	0.00	\_ /usr/sbin/apache
user	214	0.00	0.00	4976	0	?	SW	17.20	0.00	\_ (apache)
user	216	0.00	0.00	4976	0	?	SW	17.20	0.00	\_ (apache)
user	473	0.00	1.06	4976	1072	?	S	18.05	0.00	\_ /usr/sbin/apache
user	477	0.00	1.07	4976	1076	?	S	18.05	0.00	\_ /usr/sbin/apache

### **Características**

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP/1.1). Entre sus características destacan:

- Multiplataforma.
- Es un servidor de web conforme al protocolo HTTP/1.1
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

Pero una de las características fundamentales de Apache es su flexibilidad a la hora de ser configurado. Dicha configuración está basada en módulos, que pueden ser añadidos o eliminados, según el interés del usuario a cierta funcionalidad. Hay muchos proyectos asociados a Apache cuyo objetivo es aumentar su funcionalidad. Se destacará uno de ellos muy útil para los desarrolladores, conocido como PHP.

#### **2.7.2 PHP (Hypertext Preprocessor)**

En la programación de aplicaciones en Internet, son importantes las herramientas de programación que se utilizan tanto en el lado del cliente, como del servidor. Se pueden utilizar lenguajes como Javascript, JScript, o VBScript del lado del cliente, para aumentar la funcionalidad de las páginas HTML. De forma similar, hay lenguajes de programación del lado del servidor, que se introducen dentro de las páginas HTML.

En el servidor de web Apache, existe un lenguaje de programación que se puede introducir dentro de las páginas HTML conocido como PHP, que es sencillo de utilizar. Es un lenguaje en constante desarrollo, cuya principal característica es proporcionar una librería de funciones que permita acceder a las principales bases de datos del

mercado (Adabas, ilustra de Informix, Oracle, MySQL, PostgreSQL entre muchas otras), lo que facilita mucho la integración del web con el mundo de las bases de datos.

PHP es un lenguaje muy potente en la programación del lado del servidor, y que sustituye de forma elegante a la programación de cgi-bin, ya que es muy atractivo para todo desarrollador de aplicaciones en Internet.

PHP (acrónimo de Hypertext Preprocessor) es, en resumen, un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en un servidor web.

### **Introducción a la programación en PHP**

PHP es uno de los lenguajes de lado servidor más extendidos en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente creciente que ha tenido una gran aceptación en la comunidad de webmasters debido sobre todo a la potencia y simplicidad que lo caracterizan.

PHP permite entender fácilmente sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas desarrollados íntegramente en un lenguaje distinto al HTML. Por otra parte, y es aquí donde reside su mayor interés con respecto a los lenguajes pensados para los CGI, PHP ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana, sin complicaciones.

PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones.

### **Breve historia de PHP**

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form

interpretar) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.

La siguiente gran contribución al lenguaje se realizó a mediados de 1997, cuando se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3. Actualmente PHP se encuentra en su versión 4, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de la versión anterior. Algunas mejoras de esta nueva versión son su rapidez (debido a que primero se compila y luego se ejecuta, mientras que antes se ejecutaba mientras se interpretaba el código), su mayor independencia del servidor web creando versiones de PHP nativas para más plataformas y un API más elaborado y con más funciones. A continuación, en la Figura 2.4, una gráfica que muestra el número de dominios y direcciones IP que utilizan PHP.



Figura 2.4 Uso de PHP en dominios y direcciones IP. Fuente: Netcraft.

En el último año, el número de servidores que utilizan PHP se ha disparado, logrando situarse cerca de los 5 millones de sitios y 800.000 direcciones IP, lo que ha convertido a PHP en una tecnología popular. Esto es debido, entre otras razones, a que PHP es el complemento ideal para que Linux-Apache sea compatible con la programación del lado del servidor de sitios web.

### **Características principales**

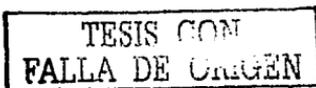
Una característica importante y que resulta muy fácil es la gestión de bases de datos; el lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales y por ODBC a todas las bases de datos posibles en sistemas Microsoft, a partir de las cuales es posible editar el contenido de un sitio web con absoluta sencillez.

También puede parecer útil el crear botones dinámicos, es decir, botones en los que se utilizan el mismo diseño y sólo se cambia el texto. Se puede, por ejemplo, crear un botón haciendo una única llamada a una función en la que se introduce el estilo del botón y el texto a introducir obteniendo automáticamente el botón deseado. Con su librería de funciones gráficas se pueden realizar muchas cosas.

Muchas otras funciones pensadas para Internet (tratamiento de cookies, accesos restringidos, comercio electrónico entre otras) o para propósito general (funciones matemáticas, explotación de cadenas, de fechas, corrección ortográfica, compresión de archivos) son realizadas por este lenguaje. A esta inmensa librería cabe ahora añadir todas las funciones personales que el desarrollador va creando por necesidades propias.

El modo de funcionamiento de una página PHP, a grandes rasgos, no difiere del clásico para una página dinámica de lado servidor: El servidor va a reconocer la extensión correspondiente a la página PHP (phtml, php, php4, por mencionar algunas) y antes de enviarla al navegador se encarga de interpretar y ejecutar todo aquello que se encuentre entre las etiquetas correspondientes al lenguaje PHP. El resto, lo enviará sin más, pues asumirá que se trata de código HTML absolutamente comprensible por el navegador.

Un programa en PHP no es lo mismo que un script escrito en otro lenguaje de programación como Perl o C, ya que en vez de escribir un programa con muchos comandos para crear una salida en HTML, se escribe el código HTML con cierto código PHP embebido (introducido) en el mismo, que producirá cierta salida. El código PHP se incluye entre etiquetas especiales de comienzo y final que permiten entrar y salir del modo PHP.



Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor y dicho servidor puede ser incluso configurado para que procese todos los ficheros HTML con PHP.

#### **Aplicaciones comunes de PHP**

PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies, entre otras. Existen tres campos principales en los que se utilizan scripts de PHP:

- Scripts en la parte del servidor.

Este es el más tradicional y el principal campo de trabajo. Se necesitan tres cosas para que esto funcione. El parser PHP (CGI ó módulo), un servidor web y un navegador. Se necesita correr el servidor web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectando con el servidor web. Es también el que se utilizará en el sistema.

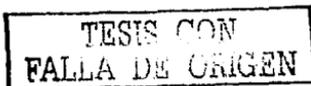
- Scripts en línea de comandos.

Se puede crear un script PHP y correrlo sin ningún servidor web ó navegador. Solamente se necesita el parser PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde (en Unix ó Linux) ó en el Planificador de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesamiento de texto.

- Escribir aplicaciones gráficas clientes.

PHP no es probablemente el mejor lenguaje para escribir aplicaciones gráficas, pero si se domina PHP y se desea usar algunas características avanzadas en programas clientes, se pueden utilizar PHP-GTK para escribir dichos programas. Es posible también escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, tales como Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X y RISC OS, principalmente. PHP soporta la



mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server (IIS), Personal Web Server, Netscape y iPlanet, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPd entre muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, incluso para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

Con PHP se tiene la libertad de escoger el sistema operativo y el servidor que mejor se adapte a las necesidades del sistema a desarrollar. También se tiene la posibilidad de usar programación orientada a objetos. Aunque no todas las características estándar de la programación orientada a objetos están implementadas en la versión actual de PHP, muchas librerías y aplicaciones grandes (incluyendo la librería PEAR) están escritas íntegramente usando programación orientada a objetos.

Con PHP no se está limitado a resultados en HTML. Entre las habilidades de PHP se incluyen, creación de imágenes, ficheros PDF y películas Flash (usando libswf y Ming) sobre la marcha. También se pueden presentar otros resultados, como XHTML y archivos XML. PHP puede auto generar estos archivos y grabarlos en el sistema de ficheros en vez de presentarlos en la pantalla.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP. Las siguientes bases de datos están soportadas actualmente:

Adabas D, Ingres, Oracle (OCI7 y OCI8), dBase, InterBase, Ovrimos, Empress, FrontBase, PostgreSQL, FilePro (read-only), mSQL, Solid, Hyperwave, Direct MS-SQL, Sybase, IBM DB2, MySQL, Velocis, Informix, ODBC, Unix dbm

Adicionalmente, PHP soporta ODBC (Open Database Connection Standard), por lo que se puede conectar a cualquier base de datos que soporte dicho estándar.

PHP tiene soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear raw sockets. PHP soporta WDDX para intercambio de datos entre lenguajes de programación en web. Y hablando de interconexión, PHP puede utilizar objetos Java de forma transparente como objetos PHP Y la extensión de CORBA puede ser utilizada para acceder a objetos remotos.

PHP tiene unas características muy útiles para el proceso de texto, desde expresiones regulares POSIX Extended ó Perl hasta parser de documentos XML. Para ejecutar el parser y acceder documentos XML, soporta los estándares SAX y DOM. Se puede utilizar la extensión XSLT para transformar documentos XML.

### 2.7.3 MySQL.

#### Conceptos básicos

*Definición de bases de datos:* Una Base de Datos es una colección estructurada de datos fiables y homogéneos, organizados independientemente en una computadora, accesibles en tiempo real. Una base de datos puede ser, desde una simple lista de artículos a las inmensas cantidades de información en una red corporativa.

*Manejador de bases de datos:* Un Manejador de Bases de Datos, es una colección de rutinas de software interrelacionadas, cada una de las cuales responde a una tarea específica. El objetivo principal de un manejador es brindar al programador un ambiente en el que se pueda crear, guardar y recuperar información en forma rápida y eficiente.

MySQL es un sistema de gestión (o manejador) de bases de datos, además es software de código abierto. Esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Se puede descargar MySQL desde Internet y usarlo sin pagar nada, de esta manera cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que se puede o no hacer con el software en diferentes situaciones.

#### Historia

MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños, aunque esta rapidez es a costa de no implementar ciertos aspectos del SQL. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones.

La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos en Internet.

IBM empezó a comercializar en 1981 el SQL y desde entonces este producto ha tenido un papel importante en el desarrollo de las bases de datos relacionales. IBM propuso (y fue aceptada) una versión de SQL al ANSI (American National Standard Institute) y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. En 1983 nació DB2, la más popular (por lo menos en los grandes ordenadores) de las bases de datos de este tipo hasta ese momento.

Alrededor de la década del 90, Michael Windenis comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Sin embargo, después de algunas pruebas llegó a la conclusión que mSQL no era lo suficientemente rápido ni flexible para sus necesidades. De todo esto surgió una nueva interfaz SQL (claro que con código más portables) con algunas apariencias similares en la API de C y en los nombres y funciones de muchos de sus programas. Esto había sido hecho para lograr con relativa facilidad portar aplicaciones y utilidades de MiniSQL a MySQL.

El Origen del nombre MySQL no está muy claro. Algunos lo atribuyen al hecho de que un gran número de sus librerías y herramientas se le asignan el prefijo "My", por costumbre. Otros dicen que se debe al nombre de la hija de Windeis. Así que cual de las dos razones da el nombre a MySQL es aún un misterio.

#### **Principales Características.**

- El principal objetivo de MySQL es velocidad y robustez.
- Escrito en C y C++, probado con GCC 2.7.2.1. Usa GNU autoconf para portabilidad.
- Clientes C, C++, JAVA, Perl, TCL.
- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y sistemas operativos distintos.
- Sistema de contraseñas y privilegios muy flexible y seguro.
- Todas las palabras viajan encriptadas en la red.
- Registros de longitud fija y variable.

- 16 índices por tabla, cada índice puede estar compuesto de 1 a 15 columnas o partes de ellas con una longitud máxima de 127 bytes.
- Todas las columnas pueden tener valores por defecto.
- Utilidad (isamchk) para revisar, optimizar y reparar tablas.
- Todos los datos están grabados en formato ISO8859\_1.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintos idiomas.
- Todos los comandos tienen -help o -? para las ayudas.
- Diversos tipos de columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc.
- ODBC para Windows (con fuentes), se puede utilizar ACCESS para conectar con el servidor.

#### 2.7.4 HTML (Hyper Text Markup Language).

##### Conceptos básicos

*World Wide Web (WWW):* Es un sistema de información; el sistema de información propio de Internet. Sus características son:

*Información por hipertexto:* Diversos elementos (texto o imágenes) de la información que se muestra en la pantalla están vinculados con otras informaciones que pueden ser de otras fuentes. Para mostrar en pantalla esta otra información bastará con hacer clic sobre ellos.

*Gráfico:* En la pantalla aparece simultáneamente texto, imágenes e incluso sonidos.

*Global:* Se puede acceder a él desde cualquier tipo de plataforma, usando cualquier navegador y desde cualquier parte del mundo.

*Pública:* Toda su información está distribuida en miles de computadoras que ofrecen su espacio para almacenarla. Toda esta información es pública y toda puede ser obtenida por el usuario.

*Dinámica:* La información, aunque está almacenada, puede ser actualizada por el que la publicó sin que el usuario deba actualizar su soporte técnico.

*Independiente:* Dada la inmensa cantidad de fuentes, es independiente y libre.

*Navegador:* Es el programa que ofrece acceso a Internet. Debe ser capaz de comunicarse con un servidor y comprender el lenguaje de todas las herramientas que manejan la información de Web. Puede decirse que cada casa de software podría tener su navegador propio, aunque los más populares son Netscape e Internet Explorer.

*Servidor:* Se encarga de proporcionar al navegador los documentos y medios que éste solicita. Utiliza un protocolo HTTP para atender las solicitudes de archivos por parte de un navegador.

*HTTP:* Hyper Text Transfer Protocol. Es el protocolo de transferencia de hipertexto, o sea, el protocolo que los servidores de World Wide Web utilizan para mandar documentos HTML a través de Internet.

*URL:* Es el Localizador Universal de Recursos, o dicho más claramente, es la dirección que localiza una información dentro de Internet. Los URL son las direcciones de la información que busca en Internet y constan de tres partes:

*Protocolo:* Es el programa que utilizará el navegador para obtener el archivo elegido

*Nombre del host:* Se trata del sistema donde se encuentra almacenada la información que se busca.

*Ruta del archivo:* Se trata de la ubicación del archivo dentro del host.

*Por ejemplo:*

<http://www.bienvenidos.mx/publico/saludos.html>

*http:* Es el protocolo

*www.bienvenidos.mx:* Nombre del host

*publico/saludos.html:* Ruta del archivo

Entre los principales tipos de URL destacan:

*HTTP:* Son los más populares ya que son los utilizados por los servidores de WWW para mandar documentos a través de Internet.

**FTP:** Se utilizan para apuntar hacia los archivos que estén en servidores que usan el protocolo FTP (File Transfer Protocol). Este protocolo es normalmente utilizado para enviar y recibir archivos. Es el protocolo que se usa para enviar las páginas al servidor de Internet. Es en estos servidores donde se almacenan los archivos que forman parte de una presentación web.

**File:** Apuntan hacia archivos contenidos en el mismo disco que se encuentra el navegador. No resulta muy interesante poner estos URL en las presentaciones puesto que otra persona que desde otro sistema apunte hacia este URL, generalmente fallará en su intento y no podrá tener acceso a él.

**Mailto:** Se usa para mandar correos electrónicos. Cuando se selecciona este tipo de URL se abre la aplicación de correo electrónico de la computadora para enviar un correo a la dirección hacia la que apunta el URL. La forma estándar es: `mailto:webmaster@bienvenidos.es`

**News:** Son URL de grupos de noticias. En estos servidores se almacenan mensajes en los que se discuten sobre diferentes temas.

### **HTML (Hyper Text Markup Language)**

HTML es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido, etc.). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Internet Explorer o Netscape).

Esto es en breve, los aspectos más generales de los productos que se utilizan para el desarrollo de la aplicación de préstamo de equipo de cómputo de los departamentos de la facultad de Ingeniería, otro aspecto que es necesario resaltar, es que el desarrollo del sistema contempla el uso de una pistola lectora de código de barras para obtener un acceso inmediato a las características generales del equipo que

se solicita para préstamo. Esto involucra conceptos de código de barras para la implementación del sistema. A continuación se muestra una descripción breve de los conceptos relacionados con el código de barras, una vez terminado este punto, se procede a continuar con el modelado del sistema mismo.

### 2.7.5 Lector óptico

Un aspecto importante para el desarrollo de la aplicación es la obtención de información a través de un lector óptico, el cual va a permitir la lectura de un código de barras (que tendrá el número de inventario asignado por la UNAM al componente) pegado al equipo de cómputo para poder así registrar cualquier movimiento del equipo.

Para comenzar, una introducción sobre la tecnología de códigos de barras.

Los códigos de barras son una técnica de entrada de datos (tal como la captura manual, el reconocimiento óptico y la cinta magnética) con imágenes formadas por combinaciones de barras y espacios paralelos, de anchos variables.

Representan números que a su vez pueden ser leídos y descifrados por lectores ópticos o scanners. Un ejemplo de código de barras se muestra en la Fig. 2.5



Figura 2.5 Ejemplo de código de barras

El código<sup>2</sup> sirve para identificar un producto de forma única pues cuenta con información detallada del artículo o del documento que los contiene, a través de la asociación con la base de datos.

Además de lograr una mejor eficiencia en la captura, almacenamiento, recuperación y manejo de datos, también se reducen costos de operación gracias a la capacidad de los sistemas informáticos para desarrollar estas tareas en forma rápida y sin errores.

---

<sup>2</sup> El primer sistema de código de barras fue patentado en 1940 por Norman Woodland y Bernard Silver, La primera lectura de un código de barras en un comercio se llevó a cabo en 1974 en Troy, Ohio, EE.UU.

### Tipos de código de barras

Así como en el mundo existen muchos idiomas y alfabetos, también hay varias simbologías de código de barras. Todos ellos fueron desarrollados con propósitos distintos; a primera vista se parecen, sin embargo, tienen sus diferencias, dependiendo de la aplicación para la que fueron creados.

En general, se pueden clasificar en dos grandes grupos:

- Los lineales (1-D) como los que se usan en productos y permiten incluir mensajes cortos. Se muestra un ejemplo en la Fig. 2.6



Figura 2.6 Código de barras lineal

- Los de dos dimensiones (2-D) que han empezado a usarse en documentos para controlar su envío o en seguros médicos y, en general, en documentos que requieren la inserción de mensajes más grandes (de hasta 2725 dígitos) como un expediente clínico completo. Ver Fig. 2.7

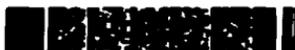


Figura. 2.7 Código de barras de dos dimensiones.

Existen diversas simbologías que pueden ser utilizadas para distintos fines, sin embargo, a nivel comercial, las más usadas en el mundo son el Universal Product Code "UPC" y el European Article Numbering Association "EAN".

### Estructura de un código de barras

Como muestra se analizará la estructura general de un código de barras lineal; sin embargo, los conceptos se aplican también a los códigos de dos dimensiones. Cabe hacer mención que el ancho de las barras y los espacios, así como el número de cada

uno de éstos varía para cada simbología. La Fig. 2.8 será el modelo del cual partirá toda la explicación.

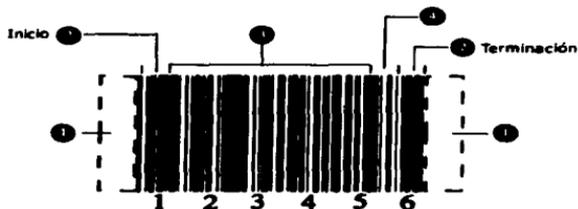


Fig. 2.8 Modelo de código de barras.

#### ❶ Quietzone

Se le llama así a la zona libre de impresión que rodea al código y permite al lector óptico distinguir entre el código y el resto de información contenida en el documento o en la etiqueta del producto.

#### ❷ Caracteres de inicio y terminación

Son marcas predefinidas de barras y espacios específicos para cada simbología. Como su nombre lo indica, marcan el inicio y terminación de un código. En el ejemplo que se muestra (Fig. 2.6.4.4) son iguales, pero en otras simbologías pueden diferir uno de otro.

#### ❸ Caracteres de datos

Contienen los números o letras particulares del símbolo.

#### ❹ Checksum

Es una referencia incluida en el símbolo, cuyo valor es calculado de forma matemática con información de otros caracteres del mismo código. Se utiliza para ejecutar un chequeo matemático que valida los datos del código de barras. Aunque puede ser importante en cualquier simbología, no son requeridos en todas ellas.

#### Tipos de lectores ópticos

Los lectores ópticos o scanners captan la información contenida en el código de barras. Son fáciles de instalar y usar. En general, emiten una línea de luz roja que se

refleja en los patrones de luz clara y oscura contenidos en las barras y los espacios. Dichos reflejos son tomados por un transductor del scanner que los convierte en una señal eléctrica, que a su vez es transformada por el decodificador del scanner en ceros y unos, o sea, en el dato binario de las computadoras.

Existen en el mercado lectores ópticos de diferentes formas y tamaños. En forma de pluma o rastrillo (que requieren hacer contacto con el código) o tipo pistola láser, que pueden hacer la lectura a distancia.

En cuanto a la facilidad de pasar la información a la computadora existen:

• **Lectores con emulación de teclado**

Se conecta a una computadora a través de un puerto de interfase de teclado. Cuando un código de barras es digitalizado, la información es transmitida a través de éste al tiempo que fue capturada en el teclado. Algunas veces se les refiere como lectores con emulación de teclado porque físicamente hay una emulación entre el teclado y la computadora que contiene un segundo teclado. Otra gran ventaja de estos lectores es que la lectura de código de barras puede ser agregada sin que haya cambios en el software; el software piensa que recibe la información como si lo hubiera hecho alguien que teclaea muy rápido. Con un lector con emulación cualquier programa que acepta datos teclados aceptará la información del código de barras sin ningún cambio.

• **Lectores seriales de código de barras**

Otra forma de transmitir datos de un lector a una computadora es conectándola a la computadora a través de un puerto serial RS-232. La información del código de barras será transmitida a la computadora en un formato ASCII para aparecer como datos teclados a la computadora. Usando una conexión de puerto serial es ideal para una computadora multiusuario. Con terminales seriales ASCII para cada usuario, el lector de código de barras puede conectarse entre la terminal y la computadora y transmitir datos ASCII justo como una terminal.

- **Lectores portátiles tipo batch**

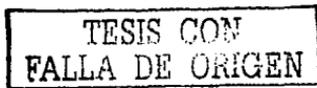
Los lectores portátiles y los lectores portátiles operados por una batería almacenan la información en memoria para actualizarla en la computadora varias veces. Un lector portátil tipo batch contiene un escáner, una pantalla LCD para agilizar al usuario a mejorar una tarea e incluso se pueden agregar variables de teclado, como por ejemplo, cantidades. Se debe contar con una cuna para actualizar la información a la computadora. Los escáners portátiles tipo batch son ideales cuando la movilidad es un factor a considerar y cuando la información recolectada no es inmediatamente necesaria. Estos escáners vienen en una variedad de estilos incluyendo los portátiles, con cables o montados en algún vehículo.

- **Lectores portátiles inalámbricos**

Cuando se requiere recolectar información en un lugar remoto y se necesita contar con la información inmediatamente, una solución inalámbrica es la ideal para este tipo de requerimiento. Un escáner inalámbrico está incluido dentro de una terminal, y actualiza la información hacia la computadora al mismo tiempo que es digitalizado, instantánea y precisamente. Los productos inalámbricos le permiten al usuario escanear la información en el punto de actividad lo cual es ideal en las grandes industrias.

## **2.6 Conclusión del Capítulo**

Se han expuesto hasta este momento las características de los productos Apache-PHP-MySQL, en la Figura 2.9 se ilustra el esquema propuesto de desarrollo con lo cual se concluye este capítulo.



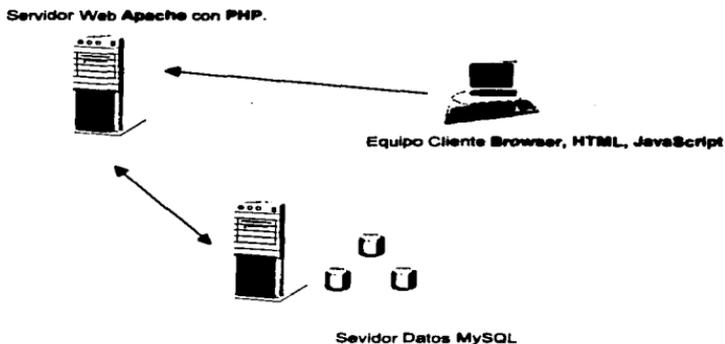


Fig. 2.9 Esquema de desarrollo

En base a la figura anterior, tanto el servidor Web como el servidor de datos con el manejador MySQL pueden residir en el mismo equipo.

### 3.1 Técnica de especificación de requerimientos

La técnica empleada para la especificación de requerimientos del problema a resolver se basa primordialmente en el análisis estructurado, utilizando la metodología tradicional de Yourdón.

El propósito primordial del análisis estructurado es convertir las entradas, insumos, factores y políticas de usuario del problema a resolver en una especificación estructurada, que defina el problema en una forma organizada y esquematizada entendible para quienes se encargan de realizar el proyecto.

Las características del análisis estructurado son:

- Participación directa del usuario final. El usuario, quien necesita solucionar un problema específico, participa directamente en el análisis del problema a resolver, ya que es él quien indica las reglas del negocio, características del problema a resolver, presupuesto disponible, tiempo de respuesta, etc.
- Independiente de la tecnología de programación
- Metódico, planeado y sensible al tiempo. Se dice que es metódico y planeado porque se puede cuantificar y estimar el tiempo como un factor en la planeación
- Diagramas descriptivos. Mediante el uso de diagramas de flujo de datos y el diagrama de contexto se podrán especificar todas las entradas y salidas que intervienen en el problema a resolver, así como las entidades internas y externas que estén relacionadas. Con la aplicación de estas técnicas el problema resulta más comprensible para todos los que participan en la solución del mismo
- Diccionario soporte. Se cuenta con un diccionario de datos que contiene información del proyecto que puede estar compartido por todas las personas que participan en el proyecto. Asegurando con ello la autenticación única de la información que se maneja

## 3.2 Análisis estructurado

El análisis estructurado es la definición de los requerimientos del sistema mediante la obtención de los tres modelos o dominios del modelo de análisis. El modelo ambiental, el funcional, y el de comportamiento.

El objetivo que persigue el análisis estructurado es organizar las tareas asociadas con la determinación de requerimientos para obtener la comprensión completa y exacta de una situación dada.

### 3.2.1 Modelo ambiental

El modelo ambiental define la frontera entre el sistema y el ambiente en el cual existe. Define qué está en el interior del sistema y en el exterior. Se necesita conocer la información que entra al sistema desde el ambiente exterior, y que información produce como salida el ambiente externo.

#### Diagrama de contexto

El diagrama de contexto es una introducción de manera general a los diagramas de flujo de datos para el usuario. Este diagrama puede ayudar a definir la estructura del área bajo revisión y asegurar con el usuario la estructura y comportamiento de dicho sistema. Principalmente se utiliza para describir la transformación de entradas a salidas.

Los componentes de un diagrama de flujo de datos son: Procesos, Flujos, Almacenes y Terminadores.

En la Figura 3.1, que se muestra a continuación, se puede observar como están relacionados los componentes de un diagrama de flujo.

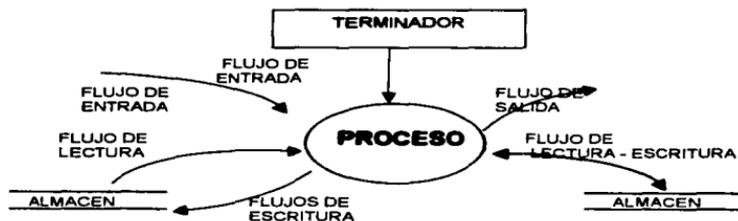


Figura 3.1 Diagrama de Contexto

Los flujos indican si los datos se están moviendo hacia adentro o hacia fuera de un proceso, almacén o terminador. Se identifican por un nombre, el cual representa los datos que se mueven durante el proceso.

Una cabeza de flecha en cualquier extremo del flujo indica si los datos se están moviendo hacia adentro o hacia fuera.

A diferencia del diagrama de flujo de datos, el diagrama de contexto enfatiza varias características importantes del sistema y se compone por:

- La frontera entre el sistema y mundo exterior.
- Las personas, organizaciones y sistemas (Terminadores) con los que se comunica el sistema; se representan por medio de un rectángulo .
- Del mundo exterior el sistema recibe los datos, los procesa y retorna nuevamente hacia él. Esto se conoce como Flujo de Datos. Se representa por medio de una flecha con punta en un sentido  $\longrightarrow$  o ambos  $\longleftrightarrow$ .
- Los almacenes de datos que el sistema comparte con los terminadores se pueden crear fuera o dentro del sistema para su uso interno o externo. En el presente trabajo se utilizará la representación de dos líneas paralelas  $\equiv$ . Si se requiere representarlo más de una vez será por medio de dos líneas paralelas punteadas  $\cdots\cdots$ .
- Para la representación de la concentración de almacenes o de información en general se utilizará un cilindro. .

### 3.2.2 Modelo de comportamiento

Describe el comportamiento que se requiere del sistema que va interactuar con el ambiente. El modelo de comportamiento se compone básicamente de:

- Diagrama Jerárquico Funcional
- Diagrama de Flujo de Datos
- Diagrama Entidad Relación

#### Diagrama Jerárquico Funcional

En este diagrama se estructuran jerárquicamente los módulos (procesos) que conforman el sistema de manera integral y se ordena para su posterior distribución en menús del sistema.

Para llevar a cabo este diagrama se parte de los Diagramas de Flujo de Datos, en donde cada proceso se convierte en un módulo.

Los módulos se clasifican de acuerdo a las funciones que se realizan, por ejemplo:

- Módulos coordinadores o principales: Son aquellos que controlan a otros módulos
- Módulos de entrada: Aquellos en los que sólo se ingresan datos
- Módulos de salida: Aquellos que a través de un proceso generan una salida a pantalla, impresora, Internet, disco, etc.
- Módulos de entrada - salida: Aquellos que cumplen con ambas funciones

Una manera de representar los módulos es por medio de rectángulos mientras que las relaciones entre ellos, se muestran a través de líneas. A continuación se presenta el diagrama jerárquico o de módulos para el sistema.

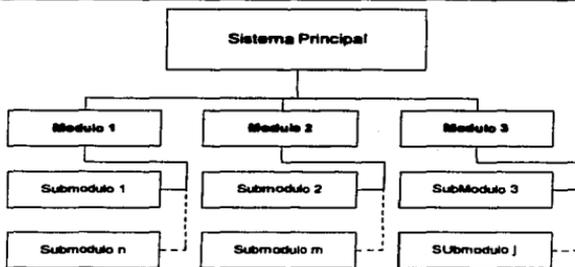


Figura 3.2 Diagrama Jerárquico

### 3.2.3 Modelo Entidad – Relación.

El objetivo del modelo Entidad – Relación es definir y entender todas las reglas de negocio que se necesitan conocer y la información que se necesita almacenar, así como la relación o asociación entre ellas.

El modelo Entidad - Relación es una técnica que se utiliza para diseñar la estructura de la base de datos en la que reside la información que se requiere, a la vez que provee de un firme fundamento para una alta calidad en los sistemas. Además, es una forma efectiva para integrar y documentar los requerimientos de información de una organización.

#### Acercamiento

Un modelo ER (Entidad - Relación) es una forma efectiva para integrar y documentar los requerimientos de información de una organización.

#### Sintaxis

Un modelo ER documenta los requerimientos de información de la organización en un formato preciso y claro.

### **Definición del alcance**

Un modelo ER provee una clara imagen del alcance de los requerimientos de información de las organizaciones.

Para definir las entidades se emplea el método conocido como Top-Down, donde se observan todas las cosas sobre las cuales se necesita conservar la información.

### **Componentes del Modelo Entidad – Relación**

- Entidad
- Relación
- Atributo
- Identificador único

#### **Entidad**

Una entidad es un aspecto importante del negocio sobre el cual se requiere tener o conocer información.

Existen algunas convenciones y reglas básicas que son utilizadas para simplificar el diagrama y mostrar información adicional.

#### **Relación**

Las entidades no existen aisladas, deben relacionarse a otras entidades.

- Una relación es una asociación entre dos entidades
- Una relación es "lo que una cosa tiene que hacer con otra cosa"
- Las relaciones son las reglas del negocio que unen las necesidades de información del negocio
- Una relación es una asociación bi-direccional entre dos entidades, o entre una instancia de una entidad y otra instancia de la misma entidad

Cuando se considera una relación, se debe pensar como fluye en dos direcciones.

*Relaciones posibles.* En la Figura 3.3 se pueden observar las representaciones de los más comunes tipos de relaciones.

- La primer Relación mostrada es la más común, conocida como "Maestro – Detalle", la cual es vista a menudo en transacciones

- Las Relaciones Muchos a Muchos son más comunes en versiones iniciales de un modelo porque muchas de ellos serán resueltas con entidades de intersección al final del análisis.

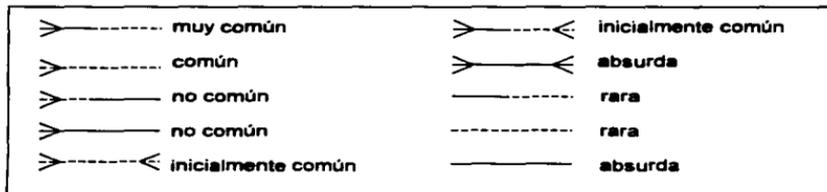


Figure 3.3 Tipos de relaciones

### Atributo

Los atributos describen una entidad y son piezas específicas de información; las cuales, necesitan ser conocidas. Cuando se identifica una entidad, también se encuentran algunos atributos para la entidad. Todas las entidades deben tener atributos, ya que de lo contrario, no son entidades. Los atributos son la forma en que se almacena la información acerca de las entidades.

### Reglas para definir un atributo

Un Atributo debe tener un nombre significativo en singular, el cual la mayoría de las veces no debe incluir el nombre de la entidad. El nombre debe ser único dentro de una entidad. Debe haber un valor por Atributo para cada ocurrencia de un tipo de entidad. Para cada Atributo es necesario definir: opcionalidad, formato, porcentaje de ocurrencias y opcionalmente definir, valor default (o por omisión), rango y/o valores permitidos y porcentaje de ocurrencias.

Los Atributos que tienen un formato común como dinero y capacidad, pueden describir una sola vez sus líneas de dirección líneas de dirección pueden como un Dominio. Todos los Atributos que tengan asignado ese Dominio utilizarán las mismas reglas de validación.

### Convenciones para mostrar los atributos en un diagrama

Se deben mostrar los Atributos en un diagrama Entidad-Relación sólo cuando sea útil para el entendimiento de la entidad. Si se conoce, se debe preceder el nombre del Atributo con un símbolo que indique su opcionalidad u obligatoriedad. En resumen, cuando es necesario guardar Atributos de un Atributo esto es una clara señal de que hace falta una entidad más.

Símbolo	Significado
*	Obligatorio $\Rightarrow$ El atributo debe siempre contener un valor
o	Opcional $\Leftrightarrow$ El atributo puede no tener un valor.

### Identificador único

Cada instancia de una entidad debe ser identificable de manera única, de tal forma que sea distinta de las otras. Un identificador único (UID) es cualquier combinación de atributos y relaciones que sirven para identificar una instancia dentro de una entidad de manera única. Se pueden especificar identificadores únicos en cualquier momento del análisis, pero cada entidad debe tener un UID para empezar el diseño.

Convenciones para mostrar los identificadores únicos en un diagrama ER. Un identificador único puede estar formado por:

- Uno o más Atributos
- Una o más Relaciones
- Una combinación de Atributos(s) y Relación(es)

Los símbolos en un diagrama para indicar los componentes de UID's son:

Símbolo	Descripción
#	Atributo el cual es(o es parte de) un identificador único
- -----	Relación la cual es parte de un identificador único.(Barra UID)

### **Normalización**

La normalización es el proceso que se lleva a cabo durante el diseño de la base de datos para eliminar los problemas de redundancia, así como identificar entidades y relaciones perdidas. Es una validación de atributos en el análisis y de columnas en el diseño.

Se puede definir a la normalización como el proceso donde se ajustan las tablas, claves, columnas y relaciones para la creación de un diseño de base de datos uniforme. En otras palabras es el proceso de agrupar columnas en varias tablas o entidades relacionadas. La normalización se consigue al aplicar una serie de pruebas o normas a las tablas. Usualmente se aplican los tres primeros niveles de normalización aunque existen dos niveles más.

Para ello se tomaron en cuenta las tres formas normales siguientes:

#### **Primera Forma Normal (1FN)**

En esta etapa se establece que cada columna debe tener un único valor indivisible, este valor también se le conoce como atómico. Las columnas deben ser únicas en la tabla, es decir, no pueden existir dos columnas que almacenen la misma información en una tabla. Partiendo del hecho que se realizó con satisfacción esta etapa, se puede establecer con mayor facilidad una clave primaria, la cual permite identificar a cada renglón de forma única dentro de la misma tabla.

#### **Segunda Forma Normal (2FN)**

Para poder aplicar la segunda forma normal a una tabla, es necesario que antes esté en la primera forma normal. Esta segunda forma se aplica en las tablas que tengan claves primarias compuestas, o sea claves formadas por dos o más columnas. En tablas con esta característica, se deben quitar las columnas que no dependan totalmente de la clave primaria y crear una o varias tablas con los datos suprimidos, dejando únicamente los datos dependientes.

#### **Tercera Forma Normal (3FN)**

Se cumple la tercera forma normal sólo si está en la segunda forma normal. Esta forma indica que no debe existir una relación estrecha entre las columnas o atributos de una tabla, es decir, no debe haber dependencia transitiva entre las columnas.

### 3.3 Creación de los diagramas de flujo de datos

Los componentes del diagrama de flujo de datos que se construyeron para diseñar el modelo de comportamiento son:

- Diagrama de Contexto: Representación del sistema en forma general
- Diagrama de flujo de datos nivel 0: Enumera los subsistemas
- Diagramas de flujo de datos nivel 1: Describe cada uno de los subsistemas del nivel 0

#### 3.3.1 Diagrama de Contexto

Para el diseño del diagrama de contexto se identifica el medio ambiente en que se desarrolla el problema, los grupos, personas o entidades relacionadas y los agentes externos, así como las entradas y salidas dentro del entorno. La Figura 3.4 muestra el diagrama de contexto del sistema a desarrollar en el presente trabajo.

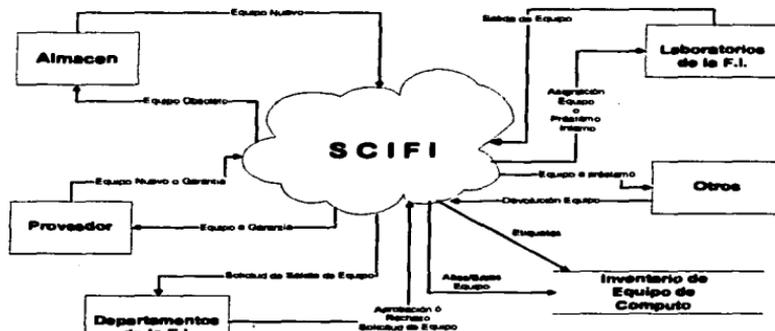


Figura 3.4 Diagrama de Contexto del Sistema de Control de Inventarios de la Facultad de Ingeniería

### Descripción

Una vez que el equipo de cómputo se recibe en el almacén se procede a registrar el equipo y entonces así, poder asignarlo a los diferentes laboratorios que pertenecen a los Departamentos de la Facultad de Ingeniería.

Las salidas de equipo de cómputo que se consideran son:

- Garantía (envío al proveedor)
- Préstamo interno o externo
- Baja del equipo (envío al almacén)

**Garantía (envío al proveedor):** El equipo de cómputo sólo se enviará al proveedor cuando éste tenga vigente la garantía; en tal caso el sistema registrará la salida y destino del equipo de cómputo.

**Préstamo interno o externo:** El laboratorio solicita la salida del equipo, ya sea préstamo interno (mejor conocido como préstamo a otro laboratorio de la Facultad de Ingeniería) o préstamo externo hacia algún otro destino.

**Baja del equipo (envío al almacén):** Cuando el equipo se necesita dar de baja por falla definitiva, obsolescencia, etc., el equipo se envía al almacén, donde se registra la baja.

Los Departamentos de la Facultad de Ingeniería tienen la función de autorizar o rechazar las solicitudes de salida de equipo de cómputo, así como consultar el inventario de equipo de cómputo de los laboratorios de la Facultad para revisar existencias y asignaciones de los mismos.

El almacén es el encargado de asignar números de inventario a cada equipo de cómputo y accesorios de equipo de cómputo. También tiene la tarea de imprimir etiquetas de código de barras con el número de inventario de cada equipo para su identificación y control.

### **Alcance**

Es importante señalar que para el presente trabajo, el alcance del sistema se limita solo al préstamo de equipo de cómputo entre los laboratorios de la Facultad de Ingeniería. Para los casos en que el equipo cause baja o salga a garantía se especificará en las características de la solicitud de préstamo, por lo que los diagramas de flujo de datos de los subsistemas nivel 0 y 1 se detallan sólo para el alcance mencionado en este último apartado.

#### **3.3.2 Diagrama de flujo de datos Nivel "0"**

Para el diseño del diagrama de flujo de datos (DFD) nivel 0 se toma en cuenta las reglas siguientes:

- Los procesos deben tener una actividad y un número de identificación
- Los procesos deben tener verbos en su texto
- Entidades externas y almacenes no se pueden unir entre sí
- No está permitido el uso de un if – then (si - entonces)

En el DFD nivel "0" se enumeran los diferentes subsistemas que integran al sistema, que de acuerdo al análisis de requerimientos se muestra en la Figura 3.5 mostrada a continuación:



3. Cualquier tipo de salida del equipo de cómputo de los laboratorios se verifica en el sistema mediante una consulta a las solicitudes de salida que han sido aprobadas y una vez que se ha verificado la aprobación se registra la salida del equipo del edificio.

4. Cuando en el proceso número 1 de esta descripción (altas y bajas) no se han impreso todas las etiquetas de código de barras, se seleccionan todos aquellos números de inventario que no tienen etiqueta impresa. Con el conjunto de números de inventarios seleccionados se imprimen las etiquetas correspondientes para cada número de inventario y se pegan a su correspondiente componente. La impresión de los códigos de barras se obtiene del software designado para este propósito, el cual se encuentra como un módulo independiente del mismo, es decir, es un programa comercial.

### 3.3.3 Diagrama de flujo de datos Nivel "1"

Para los DFD's nivel "1" se procedió a describir cada uno de los subsistemas descritos en el punto 3.3.1, a un mayor detalle de comprensión empezando en orden ascendente.

A continuación, se muestra el Diagrama de Flujo de Datos Nivel 1, denominado: "Asignar equipo de cómputo" en la Figura 3.6

## 3.3.3.1 DFD Nivel "1" Asignar Equipo de Cómputo

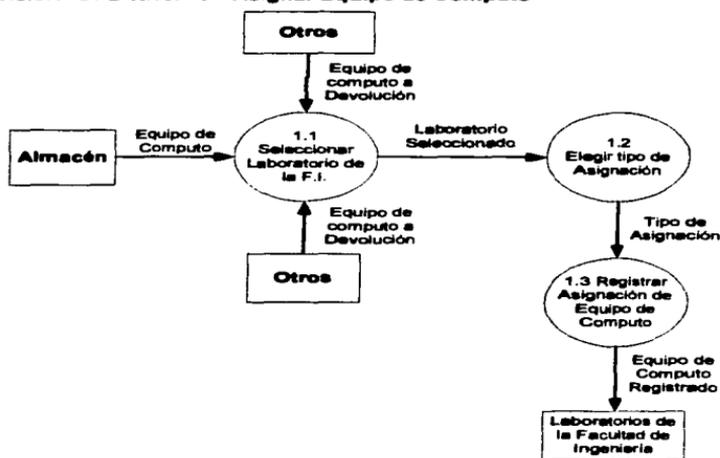


Figura 3.6 DFD Nivel 1 Asignación de equipo de cómputo.

**Descripción**

1.1 Seleccionar o localizar el laboratorio al que se le asignará el equipo de cómputo.

1.2 Con el laboratorio seleccionado se elige un tipo de asignación, pudiendo ser una de las siguientes:

- Asignación: Equipo de cómputo nuevo proveniente del almacén
- Préstamo: Equipo en calidad de préstamo
- Devolución: Equipo que es devuelto
- Otra: Se debe especificar

1.3 Con los parámetros Laboratorio y Tipo de asignación se procede a registrar la asignación del equipo de cómputo.

### 3.3.3.2 DFD Nivel "1" Procesar Solicitud de Salida

En la Figura 3.7 se muestra dicho DFD

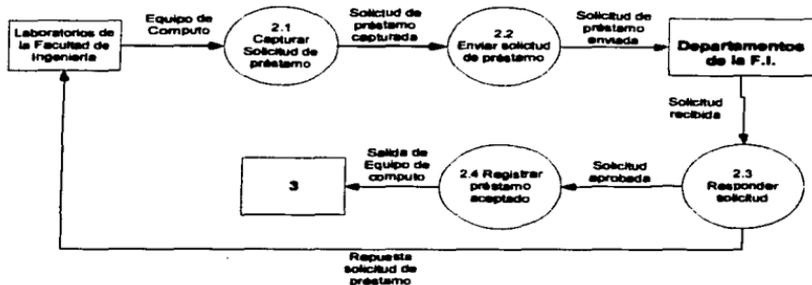


Figura 3.7 DFD Nivel 1 Procesar Solicitud de salida de equipo de cómputo.

#### Descripción

2.1 Cuando un laboratorio de la Facultad de Ingeniería va a prestar un equipo de cómputo asignado al mismo, procede a capturar los datos del equipo en una solicitud, incluyendo además destinatario, fecha de préstamo y devolución, así como cualquier observación que se considere necesaria

2.2 La solicitud capturada se envía al jefe del Departamento al que pertenece el laboratorio para su aprobación o rechazo. El jefe del Departamento es notificado, vía correo electrónico, de que se necesita aprobar una solicitud

2.3 Cuando el jefe del Departamento revisa la solicitud de préstamo procede a autorizarla o rechazarla; esta respuesta se registra en el sistema en el módulo de solicitudes, para que se tenga conocimiento si el equipo puede salir o no

2.4 Cuando la solicitud de préstamo se aprueba se procede a registrar la solicitud de préstamo como aceptada. Es entonces cuando el equipo de cómputo sale del laboratorio de la Facultad de Ingeniería

### 3.3.3.3 DFD Nivel "1" Registrar salida de equipo de cómputo

La ilustración de este diagrama de muestra en la Figura 3.8

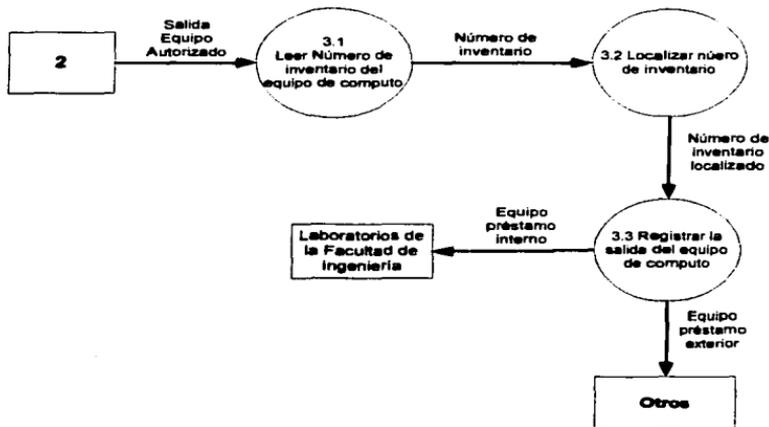


Figura 3.8 DFD Nivel 1 Salida del equipo de cómputo por préstamo interno o externo.

#### Descripción

3.1 Cuando un equipo de cómputo sale de alguno de los laboratorios de la Facultad de Ingeniería, se procede a capturar (por medio de una lector óptico) el código de barras que representa el número de inventario del equipo.

3.2 El número de inventario se busca en la lista de solicitudes aprobadas para salir. Cuando se localiza, se visualizan todos los datos del equipo de cómputo para verificar que efectivamente se trate del equipo que tiene permitido salir del laboratorio.

3.3 Si procede la salida del equipo de cómputo se registra en el sistema, indicando la fecha de salida.

### 3.3.3.4 DFD Nivel "1" Generar etiquetas de códigos de barra con el número de inventario UNAM.

Este diagrama se muestra en la Figura 3.9

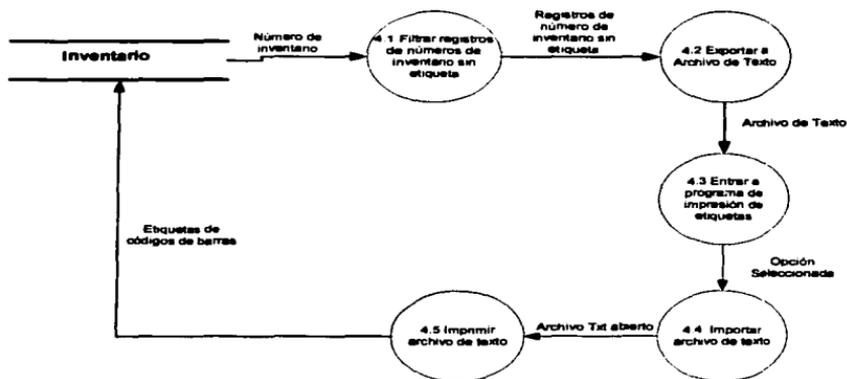


Figura 3.9 DFD Nivel 1 Generación de etiquetas de código de barras con el número de inventario UNAM para los equipos de cómputo que aún no la tienen.

#### Descripción

4.1 Con la información contenida en el inventario de equipo de cómputo, se filtran todos los equipos que no tienen aún impresa su etiqueta con código de barras.

4.2 La relación de equipos de cómputo sin etiqueta se exporta a un archivo de texto con formato de separación con comas.

4.3 Entrar a la opción del programa de impresión de etiquetas.

4.4 Desde el programa de impresión de etiquetas importar el archivo de texto que contiene la relación de números de inventario a imprimir.

4.5 Imprimir el archivo de texto directamente para obtener la etiqueta con códigos de barras.

### 3.4 Diagrama jerárquico funcional

Dentro del sistema se identifican como módulos de entrada y salida a los módulos de mantenimiento a catálogos, asignación de equipos, préstamos, salida de equipos, solicitudes, consultas generales y reportes del sistema.

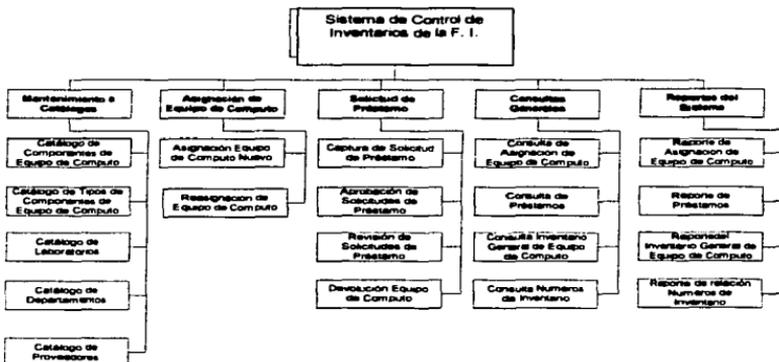


Figura 3.10 Diagrama jerárquico de los módulos del Sistema de Control de Inventarios de la Facultad de Ingeniería.

El diagrama se forma a partir Diagramas de Flujos de Datos del punto 3.3 de este CAPÍTULO, en donde cada proceso se transforma en un módulo.

### 3.5 Diccionario de datos

Un diccionario de datos es un depósito de información sobre la definición, estructura y uso de los datos. Esta información es utilizada para análisis, planificación, control y documentación general, a lo largo de la vida del sistema.

Un Diccionario de Datos es una forma de documentación para el diseñador de bases de datos. Su utilidad básica se describe en las siguientes funciones:

1. Describir el propósito de la base de datos y quiénes serán sus usuarios.
2. Documentar las especificaciones detrás de la base de datos misma: en qué dispositivo estará almacenada, cuál será el tamaño estándar de la base de datos junto con sus archivos lógicos (aquellos que almacenan información sobre operaciones en algunos sistemas de gestión de bases de datos).
3. Almacenar código fuente de SQL referente a la instalación y desinstalación de la base de datos, incluida también documentación respecto al uso de herramientas de importación / exportación.
4. Proveer una descripción detallada de cada tabla dentro de la base de datos y explicar su propósito en términos de procesos de negocios.
5. Documentar la estructura interna de cada tabla, incluyendo todos sus campos y sus tipos de datos con comentarios, todos los índices y todas las vistas.
6. Contener todo el código fuente SQL para todos los procedimientos y triggers.
7. Describir reglas, tales como el uso de valores no nulos, valores únicos, etc.

El diccionario de datos es también un listado organizado de todos los datos pertinentes al sistema con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios.

**Notación del diccionario de datos**

Existe cierta simbología utilizada en el diccionario de datos. Abajo, se muestra el significado de los símbolos usados:

- = quiere decir, está compuesto de
- + y
- { } iteración
- [ ] seleccionar una de varias alternativas
- | separa opciones alternativas en la construcción (OR)

Recomendaciones para la creación de un buen diccionario de datos:

- Hay que asegurar que el diccionario sea completo y consistente.
- Definir en el diccionario todas las componentes del Diagrama de Flujo de Datos (o del modelo).
- Definir todos los componentes de las estructuras en el diccionario.
- No tener más de una definición para el mismo dato.
- Utilizar la notación correcta.
- No tener definiciones que no se utilicen en el Diagrama de Flujo de Datos (o el modelo).

A continuación, el diccionario de datos del Sistema de Control de Inventarios de la Facultad de Ingeniería.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
comp	Componente de equipo de cómputo	Elemento del equipo de cómputo que pueda salir o entrar a algún laboratorio de la Facultad de Ingeniería	comp = comp_inv + comp_tipo + comp_clav_marc + comp_clav_prov + comp_desc + comp_serie + comp_estd + comp_data			Table
comp_inv	Inventario del componente	Número de inventario Asignado por la UNAM al componente	7(0-9)7	Integer(7)	NO NULO	Llave primaria
comp_tipo	Tipo de componente	Clave del tipo de componente	1(0-9)2	Integer(2)	NO NULO	Llave foránea
comp_clav_marc	Marca del componente	Clave de la marca a la que pertenece el componente	1(0-9)2	Integer(2)	NO NULO	Llave foránea

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMBIÓN	CARACTERÍSTICAS
comp_clav_prov	Proveedor del componente	Clave del proveedor del componente	1{0-9}2	Integer(2)	NO NULO	Llave foránea
comp_desc	Descripción del componente	Explica algunas características generales del componente	1{A-Z}{0-9}20	Varchar(20)	NULO	Elemento
comp_serie	Serie del componente	Número de serie de fábrica del componente	1{A-Z}{0-9}30	Varchar(30)	NO NULO	Elemento
comp_detla	Detalles del componente	Descripción detallada del componente.	1{A-Z}{0-9}250	Varchar(250)	NULO	Elemento

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
cont	Contiene	Indica que los componentes pueden estar contenidos en una solicitud de salida	cont = cont_clav_sol + cont_inv_comp			Table
cont_clav_sol	Clave de la solicitud	Clave de la solicitud de salida de equipo	1(0-9)4	Integer(4)	NO NULO	Llave primaria
cont_inv_comp	Clave del componente	Número de inventario asignado por la UNAM al componente	7(0-9)7	Integer(7)	NO NULO	Llave primaria

TESIS CONT  
FALTA DE CUBRIR

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
dep	Departamento de la Facultad de Ingeniería	Departamento responsable de los laboratorios de la Facultad de Ingeniería y del equipo de cómputo que se encuentra en los mismos	dep=dep_clav +dep_resp +dep_nomb +dep_tele +dep_ubic			Table
dep_clav	Clave del Departamento	Notación numérica única que identifica al Departamento de la Facultad de Ingeniería	1(0-9)2	integer(2)	NO NULO	Llave primaria
dep_resp	Clave del Responsable del Departamento	RFC del responsable	4(A-Z)+(AA+MM+DD)   4(A-Z)4+(AA+MM+DD)+ 3(A-Z)(0-9)3	varchar(13)	NO NULO	Llave foránea
dep_nomb	Nombre del Departamento	Nominación del Departamento de la Facultad de Ingeniería	1(A-Z)40	varchar(40)	NO NULO	Elemento
dep_tele	Teléfono del Departamento	Número telefónico del Departamento sin guiones ni espacios	8(0-9)10	varchar(10)	NULO	Elemento

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
dep_ubic	Ubicación del Departamento	Descripción de dónde se encuentra el Departamento, especificando edificio y piso	1(A-Z)(0-9)40	varchar(40)	NULL	Elemento

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
inv	Inventario	Inventario del equipo de cómputo existente en los laboratorios de la Facultad de Ingeniería	inv = inv_clav + inv_clav_dep + inv_clav_lab + inv_fech_reg + inv_fech_bej + inv_etiq + inv_estd			Table
inv_clav	Inventario del componente	Número de inventario asignado por la UNAM al componente	7(0-9)7	Integer(7)	NO NULO	Llave primaria
inv_clav_dep	Clave de departamento	Notación numérica única que identifica al Departamento de la Facultad de Ingeniería	1(0-9)2	Integer(2)	NO NULO	Llave primaria
inv_clav_lab	Clave de laboratorio	Notación numérica única que identifica a un Laboratorio de la Facultad de Ingeniería	1(0-9)3	Integer(3)	NO NULO	Llave primaria

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
inv_fech_reg	Fecha de registro	Fecha en la cual ingresa un componente a un Laboratorio de la Facultad de Ingeniería	MM-DD-YYYY HH:MM	Date	NO NULO	Elemento
inv_fech_baj	Fecha de baja	Fecha en la cual un componente es dado de baja de un Laboratorio de la Facultad de Ingeniería	MM-DD-YYYY HH:MM	Date	NULO	Elemento
inv_etiq	Etiqueta	Indicativo de que el componente cuenta con etiqueta de código de barras impresa 0 = No tiene etiqueta 1 = Si tiene	{0,1}	Smallint(1)	NO NULO	Elemento

TESIS CON FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
inv_estado	Estado	Indicativo del estado en que se encuentra el componente inventariado: 0 = Disponible 1 = En préstamo 2 = Garantía 3 = Baja 4 = En Proceso	{0 1 2 3}	Smallint(1)	NO NULO	Elemento

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
 FACULTAD DE INGENIERÍA  
 SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
lab	Laboratorio de la Facultad de Ingeniería	Laboratorio perteneciente a la Facultad de Ingeniería que tiene el equipo de cómputo	lab= lab_clav + lab_clav_dep + lab_resp + lab_nomb + lab_ubic + lab_tele			Tabla
lab_clav	Clave de Laboratorio	Notación numérica única que identifica a un Laboratorio de la Facultad de Ingeniería	1{0-9}3	Integer(3)	NO NULO	Llave primaria
lab_clav_dep	Clave de Departamento	Notación numérica única que identifica al Departamento de la Facultad de Ingeniería y del cual depende el Laboratorio	1{0-9}2	Integer(2)	NO NULO	Llave primaria
lab_resp	Clave del Responsable del Laboratorio	RFC del responsable	{4[A-Z]+(AA+MM+DD)}   4{A-Z}4+(AA+MM+DD)+ 3{A-Z}{0-9}3	Varchar(13)	NO NULO	Llave foránea

TESIS CON FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
lab_nomb	Nombre del Laboratorio	Nominación del Laboratorio de la Facultad de Ingeniería	1(A-Z)40	Varchar(40)	NO NULO	Elemento
lab_ubic	Ubicación del Laboratorio	Descripción de dónde se encuentra el Laboratorio, especificando edificio y piso	1(A-Z)(0-9)40	Varchar(40)	NULO	Elemento
lab_tele	Teléfono del Laboratorio	Número telefónico del Laboratorio sin guiones ni espacios	8(0-9)10	Varchar(10)	NULO	Elemento

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
prov	Proveedor	Empresa o persona que provee el equipo de cómputo	prov = prov_clav + prov_nomb + prov_contc + prov_dirac + prov_tele + prov_mail			Tabla
prov_clav	Clave de proveedor	Notación numérica única que identifica a un proveedor de equipo de cómputo	1(0-9)2	Integer(2)	NO NULO	Llave primaria
prov_nomb	Nombre del proveedor	Nominación del proveedor de equipo de cómputo	1(A-Z)(0-9)30	Varchar(30)	NO NULO	Elemento
prov_contc	Contacto	Nombre de la persona responsable por parte del proveedor	1(A-Z)30	Varchar(30)	NO NULO	Elemento
prov_dirac	Dirección	Domicilio del proveedor	1(A-Z)(0-9)40	Varchar(40)	NULO	Elemento

TESIS CON FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
prov_tele	Teléfono	Número telefónico del proveedor	8(0-9)10	Varchar(10)	NO NULO	Elemento
prov_mail	e-mail	Dirección de correo electrónico del proveedor	1(A-Z)(0-9)25	Varchar(25)	NULO	Elemento

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OPCIÓN	CARACTERÍSTICAS
sol	Solicitud de salida	Solicitud de salida de equipo de cómputo de un Laboratorio	sol = sol_clav + sol_resp + sol_fech + sol_tipo + sol_dest + sol_resp_dest + sol_aprb + sol_aprb_ok			Table
sol_clav	Clave de solicitud	Notación numérica única que identifica una solicitud de salida de equipo de Cómputo	1(0-9)4	Integer(4)	NO NULO	Llave primaria
sol_resp	Clave del responsable de la solicitud	RFC del responsable	[4(A-Z)]+(AA+MM+DD)   4(A-Z)4+(AA+MM+DD)+ 3(A-Z)0-9]3	Varchar(13)	NO NULO	Llave foránea
sol_fech	Fecha de solicitud	Fecha en la cual se realiza la solicitud de salida de equipo	MM-DD-YYYY HH.MM	Date	NO NULO	Elemento

TESIS COM  
FALTA DE CARGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
sol_tipo	Tipo de solicitud	Indicativo de la razón por la cual se solicita la salida de equipo: 0 = Préstamo 1 = Reparación 2 = Baja	{0}1{2}	Smallint(1)	NO NULO	Elemento
sol_dest	Destino del equipo	Lugar al que será llevado el equipo de cómputo	1{A-Z}{0-9}30	Varchar(30)	NO NULO	Elemento
sol_resp_dest	Responsable en el Destino	Nombre de la persona responsable del equipo al lugar donde va	1{A-Z}40	Varchar(40)	NO NULO	Elemento
sol_aprb	Aprobación de la solicitud de salida	Indicativo de si la solicitud de salida de equipo fue aprobada o no: 0 = No fue aprobada 1 = Aprobada	{0}1	Smallint(1)	NO NULO	Elemento

TESIS CON FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
 FACULTAD DE INGENIERÍA  
 SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
sol_aprb_ok	Confirmación salida equipo	Indicativo de si el componente salió de las instalaciones o no: 0 = Si salió 1 = No salió	{0,1}	Smallint (1)	NO NULO	Elemento

TESIS CON  
 FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
tipo	Tipo de componente	Descripción de los diferentes tipos de equipo de cómputo	tipo = tipo_clav + tipo_desc			Tabla
tipo_clav	Clave de tipo de componente	Notación numérica única que identifica el tipo de componente	1(0-9)2	Integer(2)	NO NULO	Llave primaria
tipo_desc	Descripción	Nominación del tipo de componente	1(A-Z)25	Varchar(25)	NO NULO	Elemento

TESIS CON  
FALLA DE ORIGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMBIÓN	CARACTERÍSTICAS
usr	Usuarios	Información de los usuarios autorizados para el manejo del sistema	usr = usr_clav + usr_nomb + usr_tipo + usr_tele + usr_mail + usr_login + usr_pass			Tabla
usr_clav	Clave de usuario	Notación alfanumérica que identifica a un usuario (RFC)	{4(A-Z)+(AA+MM+DD)   4(A-Z)+(AA+MM+DD)+ 3(A-Z)(0-9)3}	Varchar(13)	NO NULO	Llave primaria
usr_nomb	Nombre del usuario	Nombre completo del usuario	1(A-Z)50	Varchar(50)	NO NULO	Elemento
usr_tipo	Tipo de usuario	Indicativo del grado que ocupa el usuario: 0 = Jefe de Departamento 1 = Jefe de Laboratorio 2 = Ayudante de Laboratorio 3 = Vigilante 4 = Administrador del Sistema	{0 1 2 3}	Smallint(1)	NO NULO	Elemento

TESIS CON  
FALLA DE CINGEN

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERÍA

DICCIONARIO DE DATOS

CLAVE DE ELEMENTO	NOMBRE COMPLETO	SIGNIFICADO	DEFINICIÓN	TIPO EN MySQL	VALOR POR OMISIÓN	CARACTERÍSTICAS
usr_tele	Teléfono	Número telefónico en el que se puede localizar al Usuario	8(0-9)13	Varchar(13)	NULLO	Elemento
usr_mail	e-mail	Dirección de correo electrónico del usuario	1(A-Z)(0-9)25	Varchar(25)	NULLO	Elemento
usr_login	Login	Login con el cual el usuario será identificado por el sistema	1(A-Z)(0-9)10	Varchar(10)	NO NULLO	Elemento
usr_passw	Password	Clave con la que se le permitirá al usuario el acceso al sistema	1(A-Z)(0-9)10	Varchar(10)	NO NULLO	Elemento

TESIS CON FALLA DE ORIGEN

### 3.6 Diagrama Entidad Relación

El diagrama Entidad – Relación del Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI) se construyó en la etapa final del análisis de los requerimientos mediante las técnicas descritas a lo largo del presente capítulo. Las entidades que se identifican para elaborar el diagrama son:

- Departamentos
- Laboratorios
- Usuarios
- Componentes
- Proveedores
- Inventario
- Solicitudes

En la Fig. 3.11 se muestra el Diagrama Entidad – Relación del Sistema:

[1.1]

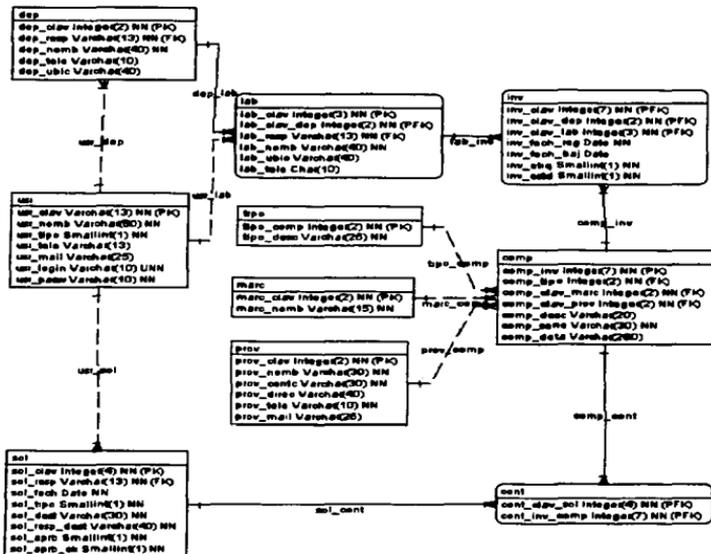


Fig. 3.11 Diagrama Entidad – Relación del Sistema de Control de inventarios de la Facultad de Ingeniería

#### 4.1 Introducción

En el capítulo anterior se presentó la información concerniente a los conceptos teóricos sobre el análisis de requerimientos.

En este capítulo se aplican esos conceptos al problema planteado en este trabajo de tesis, lo cual nos permite modelarlo de una manera sencilla a través del análisis estructurado bajo la metodología tradicional de Yourdón.

El modelado del sistema a través del diagrama de contexto, diagrama jerárquico funcional, diagrama de flujo de datos y diagrama Entidad - Relación se presentan en este capítulo y encierran toda la base del análisis que conforman la frontera a la siguiente etapa del desarrollo del sistema, esto es, su implementación en un lenguaje de programación.

El diagrama jerárquico funcional muestra la jerarquía de los procesos que conforman el sistema a través de una relación de bloques que los identifican. El enfoque de dichos bloques se realiza de acuerdo a las necesidades que se presentan en el desarrollo del sistema en cuestión y es en este diagrama en donde se muestran los módulos principales del sistema a través de prototipos de pantallas que identifican la captura de una solicitud de salida, las pantallas de catálogos y demás pantallas.

El diagrama Entidad - Relación, es donde se identifican las entidades que conforman la base de datos y la relación existente entre dichas entidades, así como los atributos correspondientes a cada una de ellas. Para el caso del sistema que se desarrolla en este trabajo de tesis, se identifican las entidades que permitirán el almacenamiento posterior de la información a través del diseño de la base de datos por medio de las tablas que la conforman.

## 4.2 Fases del Diseño del sistema.

El diseño del sistema esta compuesto de varias actividades, teniendo como principal producto el sistema automatizado de información propiamente, pero también de ellas se derivan las especificaciones y finalmente la compra, integración o contratación de hardware, software, datos, personal y procedimientos.

### El diseño físico

Señala el entorno, plataforma e interfases humano - computadora, específicas para los propósitos del sistema de información. De la misma forma, señala los elementos de los cinco componentes de un sistema automatizado de información:

*Hardware:* Señala el tipo de computadoras personales, servidores, equipos de telecomunicaciones y redes, memoria, capacidad de almacenamiento, velocidades, etcétera.

*Software:* Especificaciones relativas al sistema operativo, software de red y telecomunicaciones, lenguajes de programación, software de respaldo y recuperación, software de apoyo a la productividad del sistema, tales como procesadores de texto, hojas de calculo, gestores de correo electrónico y agendas, entre otros.

*Datos:* Incluye aspectos como la seguridad, integridad, redundancia, estructuras físicas, manejadores de bases de datos.

*Personal:* Capacitación del personal, reasignación, cultura organizacional, etc.

*Procedimientos:* Material de capacitación, documentación del usuario, documentación operacional para el área de sistemas y procedimientos de respaldo y recuperación.

### Construcción del software

Es la actividad que esta concentrada en la creación, generación y compra del software necesario para el sistema de información. La actividad de creación es propiamente la programación. La generación incluye el uso de herramientas capaces de

generar de manera automática código de algunas partes de la aplicación, y la compra de componentes de software desarrollados por terceros que se puedan incorporar al sistema.

#### **Documentación del usuario**

Es una fase a la que se debe dedicar especial atención, ya que en esta se desarrolla la documentación para capacitar a todo tipo de usuarios del sistema. Dicha documentación es conocida como manuales de usuario.

#### **Migración de datos**

Se da cuando existe un sistema que es reemplazado por uno nuevo. Esta actividad se concentra en mover los datos de esa aplicación antigua a la nueva. Requiere de un análisis muy detallado, ya que en algunos casos consume una cantidad enorme de recursos, tanto de tiempo, como materiales y de personal al tratar de adecuar los datos a las nuevas estructuras, e inclusive en algunos casos tiene que recurrirse a procesos manuales o bien que algunas partes del viejo sistema se mantengan en funcionamiento hasta que se logre la conversión de los datos.

#### **Pruebas**

Se realizan diferentes exámenes de desempeño antes de la implantación del sistema de información. Estos se refieren a diferentes unidades del sistema y se desarrollan en entornos variados e incluyen pruebas de módulos, funciones, subsistemas, integración de componentes y pruebas donde se mide la aceptación del usuario. Esta fase concluye con la prueba del sistema funcionando completamente dentro del entorno para el que fue diseñado, con la aprobación de los usuarios.

#### **Implementación**

Una vez concluido el sistema, este se inserta en la institución, que a grandes rasgos se encuentra formada por tres sub-actividades: instalación, activación e institucionalización. La implantación pone en servicio el nuevo sistema de información. La instalación básicamente consiste en colocar el sistema en producción, lo que

involucra la instalación y prueba del hardware y software, la disponibilidad de los procedimientos y datos, así como tener listo al personal. La activación es el aspecto donde se le da a conocer a las personas para que usen el nuevo sistema y los mandos de coordinación confirmen su uso y, finalmente, la institucionalización, que es cuando el software adquiere un estatus en la organización y puede procederse a retirar el anterior.

### 4.3 Diseño físico

De manera inicial, después de realizar un minucioso estudio de diversos equipos, sistemas operativos y manejadores de bases de datos, se encuentra que la opción más adecuada a las necesidades del sistema es la que se plantea en capítulos anteriores respecto al software. En cuanto al hardware se requiere de un servidor con una buena disponibilidad sin que esta sea crítica, es decir, que no se requiere de un servidor que preste un servicio ininterrumpido; para la cantidad de información a manejar en este sistema, inclusive se le puede dar mantenimiento una o dos veces al año por la noche de fin de semana.

No se establece en esta etapa el equipo de telecomunicaciones y redes requerido, debido a que ya se cuenta con la infraestructura necesaria para ello en este momento, por lo que no será necesaria la adquisición de este tipo de implementos.

A continuación se presenta una ficha técnica del equipo requerido.

**Características del hardware y software para el desarrollo del sistema**

Procesador	Intel Pentium III Processor 1.13Ghz
Memoria	256Mb total SDRAM 133 Mhz
Controladora	Integrated Dual Channel Wide-Ultra3 SCSI Adapted
Disco duro	40 Gb Ultra3 SCSI 10,000 rpm
Unidad de respaldo	12/24Gb dds-3 DAT Drive interno
Monitor	SVGA
CD-RW	High speed IDE
Tarjeta de red	Fast Ethernet 10/100 TX UTP

<b>Características del hardware y software para el desarrollo del sistema</b>	
MODEM	Lucent 56k v.90 PCI MODEM
Sistema Operativo	Linux Red Hat
Servidor Web	Apache
Manejador de bases de datos	Mysql
Creación de scripts	PHP

Gracias a que todo el software que se utiliza para la creación de dicho proyecto es de distribución libre, el costo del servidor baja considerablemente en comparación con el software comercial.

#### 4.3.1 Descripción del software empleado

La herramienta a utilizar para la construcción de la base de datos en un diagrama Entidad – Relación se conoce como CASE Studio Ver. 2.9. Las características principales de esta herramienta CASE son:

- Construcción de diagramas Entidad – Relación
- Construcción de diccionario de datos
- Compatibilidad entre diferentes manejadores de base de datos
- Generación de procedimientos de Ingeniería en Reversa
- Generación de código SQL
- Generación de documentos HTML y RTF
- Modelado de procesos DFD's

A continuación se presenta la pantalla principal del CASE Studio, Ver. 2.9 en la Figura 4.1. Con la opción de Entity Relationship Diagrammer (Diagramador de Entidades y Relaciones) se crean las entidades requeridas seleccionando de la barra de herramientas el botón con el icono del rectángulo. Se da un clic sobre el diagrama de trabajo para activar la selección y posteriormente otro clic para ingresar los datos de la entidad.

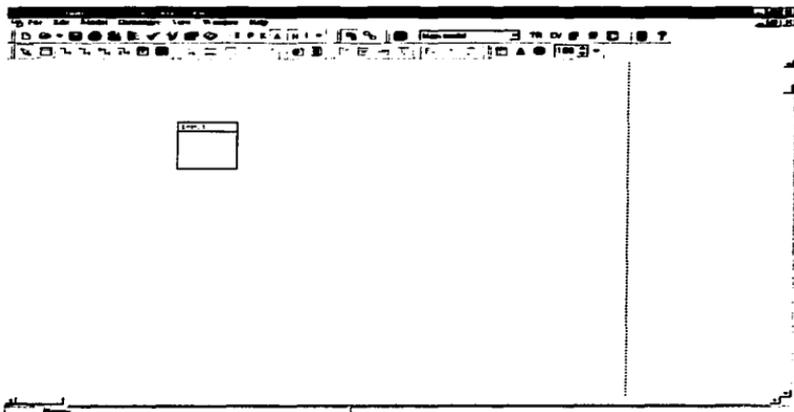


Figura 4.1 Interfaz de la herramienta CASE STUDIO para definición de entidades y relaciones

Para agregar los atributos correspondientes sólo basta con posicionarse en la entidad y presionar el clic derecho del mouse, luego la opción de editar entidades. Posteriormente se pueden dar de alta los atributos de la entidad, así como las llaves primarias de cada una de ellas.

La figura 4.2 muestra cómo se pueden editar y modificar los atributos de una entidad.

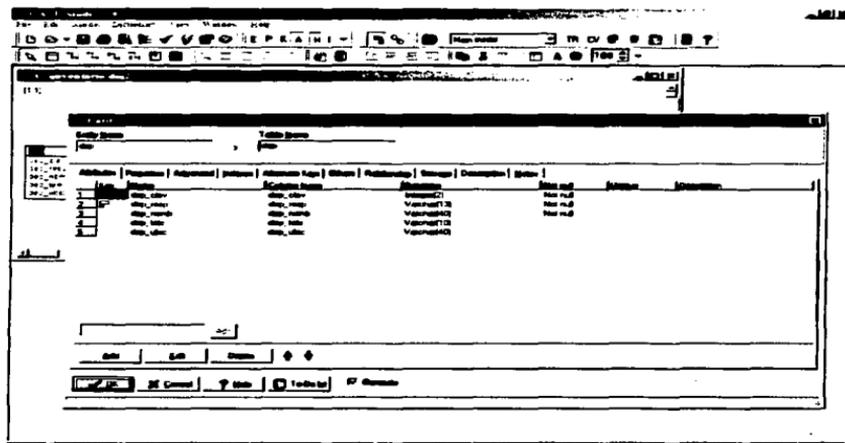


Figura 4.2 Edición y modificación de atributos de una entidad

En la parte superior izquierda se selecciona el tipo de relación que existe entre las entidades. Para relacionar las entidades se debe seleccionar la entidad origen y recorrer la relación hacia la entidad destino. En la Figura 4.3 se muestran los tipos de relaciones disponibles.



Figura 4.3 Tipos de relaciones entre entidades

Finalmente el Diagrama Entidad Relación construido permite identificar las relaciones que existen entre entidades, así como los atributos que corresponden a cada una de ellas, siendo también útil en la comunicación entre analistas y usuarios y entre analistas y diseñadores. En la Figura 4.4 se muestra el Diagrama Entidad – Relación del Sistema de Control de Inventarios de la Facultad de Ingeniería.

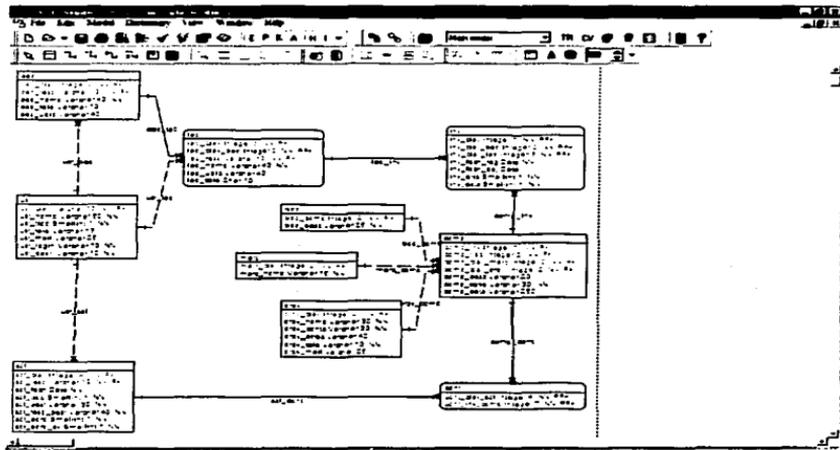


Figura 4.4 DER del Sistema de Control de Inventarios de la Facultad de Ingeniería

Para la generación del script con que se construyen las tablas de la base de datos, la herramienta CASE cuenta con una opción denominada "Reverse Engineering" o Ingeniería en Reversa cuyo propósito primordial es generar un script en lenguaje SQL y ejecutarlo en el manejador de base de datos Mysql.

## 4.3.2 Cálculo del espacio físico de la base de datos

A continuación se presenta un desglose del tamaño inicial que van a tener cada una de las tablas de la base de datos, el cual crecerá conforme pase el tiempo y se introduzca nueva información.

Tabla Departamento	Long. max de los campos		Total inicial	captura de 4 departamentos
dep_clav	Integer(2)	2+1	3	12
dep_resp	Varchar (13)	13+1	14	56
dep_nomb	Varchar (40)	40+1	41	164
dep_tele	Varchar (10)	10+1	11	44
dep_ubic	Varchar (40)	40+1	41	164
		<b>120 Bytes</b>	<b>110 bytes</b>	<b>440 bytes (1)</b>

Tabla Laboratorios	Long. max de los campos		Total inicial	captura de 6 laboratorios
lab_clav	Integer (3)	3+1	4	24
lab_clav_dep	Integer (2)	2+1	3	18
lab_resp	Varchar (13)	13+1	14	84
lab_nomb	Varchar (40)	40+1	41	246
lab_ubic	Varchar (40)	40+1	41	246
lab_tele	Char (10)	10+1	11	66
		<b>124 Bytes</b>	<b>124 bytes</b>	<b>684 bytes (2)</b>

Tabla Inventario	Long. max de los campos		Total inicial	captura de 40 inventarios
inv_clav	Integer (7)	7+1	8	320
inv_clav_dep	Integer (2)	2+1	3	120
inv_clav_lab	Integer (3)	3+1	3	120
inv_fech_reg	Date			
inv_fech_baj	Date			
inv_etiq	Smallint (1)	1+1	2	80
inv_estd	Smallint (1)	1+1	2	80
		<b>20 Bytes</b>	<b>20 bytes</b>	<b>720 bytes (3)</b>

TABLA USUARIOS	Long. max de los campos		Total inicial	Captura de 10 usuarios
usr_clav	Varchar (13)	13+1	14	140
usr_nom	Varchar (50)	50+1	51	510
usr_tipo	Smallint (1)	1+1	2	20
usr_tele	Varchar (13)	13+1	14	140
usr_mail	Varchar (25)	25+1	26	260
usr_login	Varchar (10)	10+1	11	110
usr_pasw	Varchar (10)	15+1	16	160
		<b>133 Bytes</b>	<b>133 bytes</b>	<b>1340 bytes (4)</b>

TABLA componente	Long. max de los campos		Total inicial	Captura de 40 componentes
comp_inv	Integer (7)	7+1	8	320
comp_tipo	Integer (2)	2+1	3	120
comp_clav_marc	Integer (2)	2+1	3	120
comp_clav_prov	Integer (2)	2+1	3	120
comp_desc	Varchar (20)	20+1	21	840
comp_serie	Varchar (30)	30+1	31	1240
comp_deta	Varchar (250)	250+1	251	10040
		<b>322 Bytes</b>	<b>322 bytes</b>	<b>12800 bytes (5)</b>

TABLA tipo	Long. max de los campos		Total inicial	captura de 40 tipos
Tipo_comp	Integer (2)	2+1	3	120
Tipo_desc	Varchar (25)	25+1	26	1040
		<b>29 Bytes</b>	<b>29 bytes</b>	<b>1160 bytes (6)</b>

TABLA marc	Long. max de los campos		Total inicial	Captura de 40 marcas
Marc_clav	Integer (2)	9+1	10	400
Marc_nomb	Varchar (15)	20+1	21	840
		<b>31 Bytes</b>	<b>31 bytes</b>	<b>1240 bytes (7)</b>

TABLA proveedores	Long. max de los campos		Total inicial	Captura de 20 proveedores
Prov clav	Integer (2)	2+1	3	60
Prov nom	Varchar (30)	30+1	31	620
Prov contc	Varchar (30)	30+1	31	620
Prov direc	Varchar (40)	40+1	41	820
Prov tele	Varchar (10)	10+1	11	220
Prov mail	Varchar (25)	25+1	26	520
			<b>143 Bytes</b>	<b>2860 bytes (8)</b>

TABLA solicitud	Long. max de los campos		Total inicial	captura de una solicitud
sol clav	Integer (4)	4+1	5	5
Sol resp	Varchar (13)	13+1	14	14
Sol fech	Date			
Sol tipo	Smallint (1)	1+1	2	2
Sol dest	Varchar (30)	30+1	31	31
sol resp_dest	Varchar (40)	40+1	41	41
sol aprob	Smallint (1)	1+1	2	2
sol aprob_ok	Smallint (1)	1+1	2	2
			<b>97 Bytes</b>	<b>97 bytes (9)</b>

TABLA cont	Long. max de los campos		Total inicial	Contador
Cont clav_sol	Integer (4)	4+1	5	5
Cont inv_comp	Integer (7)	7+1	8	8
			<b>13 Bytes</b>	<b>13 bytes (10)</b>

Con todo lo anterior se concluye que la base de datos tiene un tamaño en bytes de:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	Total inicial
440	684	720	1340	12800	1160	1240	2860	97	13	20,014

Lo que significa que será inicialmente una base de datos de 19.5 KB.

TESIS CON  
FALLA DE ORIGEN

Cabe mencionar que toda la información referente a la base de datos se almacena en un directorio del servidor. Al crear la base de datos con el script se crean tres archivos con el mismo nombre, pero diferentes extensiones (.FRM, .MYD, .MYI) de cada una de las entidades del diagrama Entidad - Relación.

Los archivos con extensión .FRM contienen la tabla de definiciones; los de extensión .MYD contienen almacenado los datos y los de extensión .MYI son archivos con las relaciones que existen en esta entidad. Dichos directorios están ubicados en:

Apache/Mysql/data/Nombre_base_datos/cont	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/sol	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/prov	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/inv	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/marc	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/tipo	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/comp	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/usr	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/lab	.FRM	.MID	.MYI
Apache/Mysql/data/Nombre_base_datos/dep	.FRM	.MID	.MYI

Las páginas web y demás archivos que hacen referencia al acceso de la base de datos están almacenadas en los siguientes directorios:

Apache/htdocs/Nombre_del_sistema/archivos.html	(páginas web)
Apache/htdocs/Nombre_del_sistema/archivos.php	(archivos de programación)
Apache/htdocs/Nombre_del_sistema/archivos.css	(hojas de estilo)
Apache/htdocs/Nombre_del_sistema/archivos.gif	(imágenes)

## 4.3.3 Creación del script.

El script de la creación de las tablas de la base de datos es el siguiente....

```

/*
Created          26/07/2002
Modified        30/09/2002
Project         Tesis
Model          E-R Inventario
Company        UNAM
Author         PAT Grupo 10 de 2002
Version
Database       mySQL
*/

USE mysql;

# Creación de la base de datos
DROP DATABASE IF EXISTS bd_fac_ing;
CREATE DATABASE bd_fac_ing;
# Crea cuenta y asigna permisos
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP
ON bd_fac_ing.*
TO sys@localhost
IDENTIFIED BY 'tesis'
USE bd_fac_ing;

# Elimina entidades (tablas) si existen en la base de datos.
drop table cont;
drop table sol;
drop table prov;
drop table inv;
drop table marc;
drop table tipo;
drop table comp;
drop table usr;
drop table lab;
drop table dep;

# Creación de las tablas
Create table dep (
  dep_clav integer(2) NOT NULL AUTO_INCREMENT,
  dep_resp varchar(13) NOT NULL ,
  dep_nomb varchar(40) NOT NULL ,
  dep_tele varchar(10) ,
  dep_ubic varchar(40) ,
  Primary Key (dep_clav)
);

Create table lab (
  lab_clav integer(3) NOT NULL AUTO_INCREMENT,
  lab_clav_dep integer(2) NOT NULL ,
  lab_resp varchar(13) NOT NULL ,
  lab_nomb varchar(40) NOT NULL ,
  lab_ubic varchar(40) ,
  lab_tele char(10) ,
  Primary Key (lab_clav,lab_clav_dep)
);

Create table usr (
  usr_clav varchar(13) NOT NULL ,

```

```
usr_nomb Varchar(50) NOT NULL ,
usr_tipo Smallint(1) NOT NULL ,
usr_tele Varchar(13) ,
usr_mail Varchar(25) ,
usr_login Varchar(10) NOT NULL UNIQUE ,
usr_pass Varchar(10) NOT NULL ,
Primary Key (usr_clav)
);

Create table comp (
  comp_inv Integer(7) NOT NULL ,
  comp_tipo Integer(2) NOT NULL ,
  comp_clav_marc Integer(2) NOT NULL ,
  comp_clav_prov Integer(2) NOT NULL ,
  comp_desc Varchar(20) ,
  comp_serie Varchar(30) NOT NULL ,
  comp_data Varchar(250) ,
  Primary Key (comp_inv)
);

Create table tipo (
  tipo_comp Integer(2) NOT NULL AUTO_INCREMENT ,
  tipo_desc Varchar(25) NOT NULL ,
  Primary Key (tipo_comp)
);

Create table marc (
  marc_clav Integer(2) NOT NULL AUTO_INCREMENT ,
  marc_nomb Varchar(15) NOT NULL ,
  Primary Key (marc_clav)
);

Create table inv (
  inv_clav Integer(7) NOT NULL ,
  inv_clav_dep Integer(2) NOT NULL ,
  inv_clav_lab Integer(3) NOT NULL ,
  inv_fech_reg Date NOT NULL ,
  inv_fech_baj Date ,
  inv_etiq Smallint(1) NOT NULL ,
  inv_estado Smallint(1) NOT NULL ,
  Primary Key (inv_clav,inv_clav_dep,inv_clav_lab)
);

Create table prov (
  prov_clav Integer(2) NOT NULL AUTO_INCREMENT ,
  prov_nomb Varchar(30) NOT NULL ,
  prov_contc Varchar(30) NOT NULL ,
  prov_direc Varchar(40) ,
  prov_tele Varchar(10) NOT NULL ,
  prov_mail Varchar(25) ,
  Primary Key (prov_clav)
);

Create table sol (
  sol_clav Integer(4) NOT NULL AUTO_INCREMENT ,
  sol_resp Varchar(13) NOT NULL ,
  sol_fech Date NOT NULL ,
  sol_tipo Smallint(1) NOT NULL ,
  sol_dest Varchar(30) NOT NULL ,
  sol_resp_dest Varchar(40) NOT NULL ,
  sol_aprb Smallint(1) NOT NULL ,
```

```
sol_aprb_ok Smallint(1) NOT NULL ,
Primary Key (sol_clav)
};

Create table cont (
cont_clav_sol Integer(4) NOT NULL ,
cont_inv_comp Integer(7) NOT NULL ,
Primary Key (cont_clav_sol,cont_inv_comp)
);
```

Una vez obtenido el script, se ejecuta en Mysql para crear físicamente la base de datos y las tablas que almacenan la información generada por el sistema.

TESIS CON  
FALLA DE ORIGEN

## 4.4 Interfaz

Es la descripción del proceso que se necesita establecer o indicar, donde se organizan claramente todos los detalles pertinentes del proceso real, ya que será la base para el desarrollo del sistema en las herramientas de diseño y programación. Una vez definidos todos estos parámetros, se presentan al usuario para su aprobación.

Se deben identificar y delimitar los distintos aspectos que presenta el problema, con vistas a la especificación parcial de cada uno de ellos. En último término, la especificación total es la síntesis de todas las especificaciones parciales, siempre que sean consistentes unas con otras.

### Módulo de Catálogos

### Especificación de Programa

Información General		
Proyecto: SCIFI		Modulo: CATÁLOGOS
Nombre Conceptual: SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERIA		
Nombre Físicos de los Programas:		
Programa		
Descripción General:		
En este módulo se puede capturar, modificar, borrar y consultar datos de cualquiera de los catálogos.		
Ambiente del Programa		
Entidad:	Campos:	Tipo de Índice:
DEP	DEP_CLAV, DEP_RESP, DEP_NOMB, DEP_TELE, DEP_UBIC	DEP_CLAV PK
LAB	LAB_CLAV, LAB_CLAV_DEP, LAB_RESP, LAB_NOMB, LAB_UBIC, LAB_TELE	LAB_CLAV PK
USR	USR_CLAV, USR_NOMB, USR_TIPO, USR_TELE, USR_MAIL, USR_LOGIN, USR_PASW	USR_CLAV PK
COMP	COMP_INV, COMP_TIPO, COMP_DESC, COMP_SERIE, COMP_ESTD, COMP_DET	COMP_INV PK
PROV	PROV_CLAV, PROV_NOMB, PROV_CONTC, PROV_DIREC, PROV_TELE, PROV_MAIL	PROV_CLAV PK
INV	INV_CLAV, INV_CLAV_DEP, INV_INV_CLAV, LAB, INV_FECH_REG, INV_FECH_BAJ, INV_ETIG, INV_ESTD	INV_CLAV PK INV_CLAV_DEP PK INV_CLAV_LAB PK
SOL	SOL_CLAV, SOL_RESP, SOL_FECH, SOL_TIPO, SOL_DEST, SOL_RESP_DEST, SOL_APRB, SOL_APRB_OK	SOL_CLAV PK

TESIS CON  
 FALLA DE ORIGEN

**Criterios de Ejecución del Programa:** Dentro de la pantalla principal existe una opción de catálogos, en la cual se accede con un clic del mouse o con la tecla **Alt + C** en el caso de que no se cuente con mouse, seleccionando la opción de catálogo deseado.

Tablas:	Acceso:	Referencia:
DEP LAB USUARIO COMP PROCV INV SOL	Insert, Update, Delete, Select Insert, Update, Delete, Select	

**Parámetros:** No aplica

**Rutinas Aplicativas:**

### Algoritmo

Cuerpo del programa

Mediante una pantalla principal se puede acceder a las pantallas donde se selecciona el catálogo a elegir, debe existir un botón para agregar registros, otro para modificar un registro y finalmente uno para eliminar un registro

Deben existir validaciones en cuanto a duplicados de registros, si el registro ya existe, el mensaje que debe enviar el sistema es el siguiente: (Nombre de la Llave primaria) duplicada.

Ejemplos. Departamento duplicado  
Laboratorio duplicado

En cada catálogo existen datos obligatorios, si uno de ellos no es llenado aparece la leyenda **Faltan Datos por llenar en: Nombre del campo.**

Si se requiere eliminar un registro el sistema debe desplegar un mensaje con la opción de **Si** o **No** para confirmar la acción.

Por último todas las pantallas deben contener un botón de **Salida** el cual debe estar deshabilitado cuando se edite o se cree algún registro y se habilite cuando se libere el registro y poder salir del sistema o regresar a la página principal.

**Anexos:**

Prototipo de pantalla

### Prototipo de catálogos del sistema

#### Menú Principal

- Departamentos
- Laboratorios
- Usuarios
- Componentes
- Proveedores
- Inventario
- Solicitudes

Consultar

Agregar

Borrar

Modificar

Salir

TESIS CON  
FALLA DE ORIGEN

**Módulo de Departamentos.****Especificación de Programa**

Información General		
Proyecto:	SCIFI	Modulo: Departamentos
Nombre Conceptual: SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERIA		
Nombre Físicos de los Programas:		
Programa		
Descripción General: Este módulo permite registrar, borrar, consultar y modificar la información correspondiente a los departamentos existentes		
Ambiente del Programa		
Entidad: DEP	Criterios: DEP_CLAV, DEP_RESP, DEP_NOMB, DEP_TELE, DEP_UBIC	Tipo de Índice: DEP_CLAV PK
Criterios de Ejecución del Programa: Dentro del menú principal debe existir una opción de departamento la cual se puede acceder seleccionando la opción y un botón indicando el proceso deseado.		
Tablas: DEP	Acceso: Insert, Update, Delete, Select	Referencia:
Parámetros: No aplica.		
Algoritmo		
Cuerpo del programa		
Se puede agregar, modificar y borrar datos de cualquiera de los departamentos asignados mediante un botón para cada acción.		
Para el registro de departamentos deberán solicitarse los siguientes datos:		
Clave del departamento, Clave del responsable del departamento, Nombre del departamento, Teléfono del departamento y Ubicación del departamento		
En cuanto a validaciones, no puede existir un departamento con la misma clave y nombre, si intenta ingresar estos datos el sistema debe desplegar el siguiente mensaje según sea el caso: Existe una clave, nombre duplicado sea el caso.		
Si se requiere eliminar un registro el sistema debe desplegar un mensaje para confirmar la acción.		
Por último la pantalla debe contener un botón de Salida el cual debe estar deshabilitado cuando se edite o se cree algún registro y se habilite cuando se libere el registro y poder salir del sistema		

TESIS CON  
 FALLA DE ORIGEN

## Prototipo de captura de departamentos

DEPARTAMENTOS		
CLAVE		
RESPONSABLE		Agregar
NOMBRE		Modificar
TELÉFONO		Borrar
UBICACIÓN		
Aceptar	Cancelar	Salir

## Módulo de Laboratorios

## Especificación de Programa

Información General		
Proyecto:	SCIFI	Módulo: Laboratorio
Nombre Conceptual: SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERIA		
Nombre Físicos de los Programas:		
Programa		
Descripción General: Este módulo permite registrar, borrar, consultar y modificar la información correspondiente a los laboratorios existentes		
Ambiente del Programa		
Entidad: LAB	Campos: LAB_CLAV, LAB_CLAV_DEP, LAB_RESP, LAB_NOMB, LAB_UBIC, LAB_TELE	Tipo de Índice: LAB_CLAV PK
Criterios de Ejecución del Programa: Dentro del menú principal debe existir una opción de laboratorio la cual se puede acceder seleccionando la opción y un botón indicando el proceso deseado.		

TESIS CON  
FALLA DE ORIGEN

<b>Tablas:</b> LAB	<b>Acceso:</b> Insert, Update, Delete, Select	<b>Referencia:</b>
<b>Parámetros:</b> No aplica		
<b>Algoritmo</b>		
<b>Cuerpo del programa</b>		
Se puede agregar, modificar y borrar datos de cualquiera de los laboratorios asignados mediante un botón para cada acción.		
Para el registro de laboratorios deben solicitarse los siguientes datos:		
Clave del laboratorio, Nombre del departamento al cual pertenece, Nombre del responsable del departamento, Nombre del laboratorio, Teléfono del laboratorio y Ubicación del laboratorio		
En cuanto a validaciones, no puede existir un laboratorio con la misma clave y nombre, si intenta ingresar estos datos sistema debe desplegar el siguiente mensaje según sea el caso: Existe una clave, nombre duplicado sea el caso.		
Si se requiere eliminar un registro el sistema debe desplegar un mensaje para confirmar la acción.		
Por último la pantalla debe contener un botón de Salida el cual debe estar deshabilitado cuando se edite o se cree algún registro y se habilite cuando se libere el registro y poder salir del sistema.		

### PROTOTIPO DE CAPTURA DE LABORATORIO

<b>LABORATORIOS</b>		
<input type="text" value="CLAVE"/>		
<input type="text" value="DEPARTAMENTO"/>		<input type="button" value="Agregar"/>
<input type="text" value="RESPONSABLE"/>		<input type="button" value="Modificar"/>
<input type="text" value="NOMBRE"/>		<input type="button" value="Borrar"/>
<input type="text" value="TELÉFONO"/>		
<input type="text" value="UBICACIÓN"/>		
<input type="button" value="Aceptar"/>	<input type="button" value="Cancelar"/>	<input type="button" value="Salir"/>

TESIS CON  
 FALLA DE ORIGEN

## Módulo de Usuarios

## Especificación de Programa

Información General		
Proyecto: SCIFI		Modulo: Usuarios
Nombre Conceptual: SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERIA		
Nombre Físicos de los Programas:		
Programa		
Descripción General: Este módulo permite registrar, borrar, consultar y modificar la información correspondiente a los usuarios existentes		
Ambiente del Programa		
Entidad: USR	Campos: USR_CLAV, USR_NOMB, USR_TIPO, USR_TELE, USR_MAIL, USR_LOGIN, USR_PASW	Tipo de Índice: USR_CLAV PK
Criterios de Ejecución del Programa: Dentro del menú principal debe existir una opción de usuarios la cual se puede acceder seleccionando la opción y un botón indicando el proceso deseado.		
Tablas: USUARIO	Acceso: Insert, Update, Delete, Select	Referencia:
Parámetros: No aplica.		
Algoritmo		
Cuerpo del programa		
Se puede agregar, modificar y borrar datos de cualquiera de los usuarios asignados mediante un botón para cada acción.		
Para el registro de usuarios deberán solicitarse los siguientes datos:		
Clave del usuario, Nombre del usuario, cargo del usuario, Tipo de usuario, Teléfono del usuario, correo electrónico del usuario, login y password.		
El tipo de usuarios se registrá de la siguiente manera: 0= Jefe de departamento, 1= Jefe de laboratorio, 2= Ayudante, 3= Vigilante		
En cuanto a validaciones, no puede existir un usuario con la misma clave y nombre, si intenta ingresar estos datos sistema debe desplegar el siguiente mensaje según sea el caso: Existe una clave o nombre duplicado según sea el caso.		
Si se requiere eliminar un registro el sistema debe desplegar un mensaje para confirmar la acción.		
Por último la pantalla debe contener un botón de Salida el cual debe estar deshabilitado cuando se edita o se crea algún registro y se habilita cuando se libera el registro y poder salir del sistema		

## Prototipo de captura de usuario

USUARIOS		
CLAVE		
NOMBRE		
TIPO		
TELÉFONO		Agregar
E-MAIL		Modificar
LOGIN		Borrar
PASSWORD		
Aceptar	Cancelar	Salir

## Modulo de Componentes

## Especificación de Programa

Información General		
Proyecto:	SCIFI	Modulo: Componentes
Nombre Conceptual: SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERIA		
Nombre Físicos de los Programas:		
Programa		
Descripción General: Este modulo permite registrar, borrar, consultar y modificar la información correspondiente a los componentes existentes		
Ambiente del Programa		
Entidad: COMP	Campo: COMP_INV, COMP_TIPO, COMP_CLAV, MARC, COMP_CLAV_PROV, COMP_DESC, COMP_SERIE, COMP_DET	Tipo de Entidad: COMP_INV PK
TIPO	TIPO_COMP, TIPO_DESC	TIPO_COMP PK
MARC	MARC_CLAV, MARC_NOV	MARC_CLAV PK

TESIS CON  
FALLA DE ORIGEN

**Criterios de Ejecución del Programa:** Dentro del menú principal debe existir una opción de componentes la cuál se puede acceder seleccionando la opción y un botón indicando el proceso deseado.

Tablas:	Acceso:	Referencia:
COMP TIPO MARC	Insert, Update, Delete, Select Insert, Update, Delete, Select Insert, Update, Delete, Select	

**Parámetros:** No aplica.

### Algoritmo

#### Cuerpo del programa

Se puede agregar, modificar y borrar datos de cualquiera de los componentes mediante un botón para cada acción.

Para el registro de componentes deberán solicitarse los siguientes datos:

Numero de inventario, tipo de componente (CPU, TECLADO, MONITOR, MOUSE, ETC), clave de la marca del componente, clave del proveedor, una descripción del componente, serie del componente y un campo para mayores detalles.

En cuanto a validaciones, no puede existir un componente con el mismo numero de inventario, si intenta ingresar estos datos el sistema debe desplegar el siguiente mensaje según sea el caso: Ya existe un componente.

Si se requiere eliminar un registro el sistema debe desplegar un mensaje para confirmar la acción.

Por último la pantalla debe contener un botón de Salida el cuál debe estar deshabilitado cuando se edite o se cree algún registro y se habilite cuando se libere el registro y poder salir del sistema.

### Prototipo de captura de componentes

COMPONENTES		
NÚMERO DE INVENTARIO	DESCRIPCIÓN	
TIPO	SERIE	Agregar
MARCA	DETALLES	Modificar
PROVEEDOR		Borrar
Aceptar	Cancelar	Salir

## Módulo de Proveedores

## Especificación de Programa

Información General		
Proyecto: SCIFI		Modulo: Proveedores
Nombre Conceptual: SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERIA		
Nombre Físicos de los Programas:		
Programa		
Descripción General: Este módulo permite registrar, borrar, consultar y modificar la información correspondiente a los proveedores existentes		
Ambiente del Programa		
Entidad: PROV	Campos: PROV_CLAV, PROV_NOMB, PROV_CONTC, PROV_DIREC, PROV_TELE, PROV_MAIL	Tipo de Índice: PROV_CLAV PK
Criterios de Ejecución del Programa: Dentro del menú principal debe existir una opción de proveedores la cual se puede acceder seleccionando la opción y un botón indicando el proceso deseado.		
Tablas: PROV	Acceso: Insert Update Delete Select	Referencia:
Parámetros: No aplica.		
Algoritmo		
Cuerpo del programa		
Se puede agregar, modificar y borrar datos de cualquiera de los proveedores mediante un botón para cada acción.		
Para el registro de proveedores deberán solicitarse los siguientes datos:		
Clave de proveedor, nombre del proveedor, nombre de la persona para contactarnos, dirección, teléfono y e-mail.		
En cuanto a validaciones, no puede existir un proveedor con la misma clave y nombre, si se intenta ingresar estos datos al sistema debe desplegar el siguiente mensaje según sea el caso: Ya existe un proveedor.		
Si se requiere eliminar un registro el sistema debe desplegar un mensaje para confirmar la acción.		
Por último la pantalla debe contener un botón de Salida el cual debe estar deshabilitado cuando se edite o se cree algún registro y se habilite cuando se libere el registro y poder salir del sistema.		

TESIS CON  
FALLA DE ORIGEN

## Prototipo de captura de proveedores

PROVEEDORES	
CLAVE DE PROVEEDOR	
NOMBRE	
ASESOR	Agregar
DIRECCIÓN	Modificar
TELÉFONO	Borrar
E-MAIL	
Aceptar	Cancelar
	Salir

## Módulo de inventario

## Especificación de Programa

Información General		
Proyecto:	SCIFI	Modulo: inventario
Nombre Conceptual: SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERIA		
Nombre Físicos de los Programas:		
Programa		
Descripción General:		
Este módulo permite consultar la información correspondiente al inventario existente		
Ambiente del Programa		
Entidad: INV	Campo: INV_CLAV, INV_CLAV_DEP, INV_CLAV_LAB, INV_FECH_REG, INV_FECH_BAJ, INV_ESTD, INV_ESTD	Tipo de Índice: INV_CLAV.PFK

TESIS CON  
FALLA DE ORIGEN

**Criterios de Ejecución del Programa:** Dentro del menú principal debe existir una opción que muestre componentes que tendremos en el inventario la cual se puede acceder seleccionando la opción y un botón indicando el proceso deseado.

**Tablas:**

**Parámetros:** No aplica.

**Acceso:**

Insert, Update, Delete, Select

**Referencia:**

### Algoritmo

Cuerpo del programa

Se puede asignar, cambiar de asignación y borrar los datos de los componentes que se tienen inventariados a los diferentes departamentos y laboratorios.

Se debe desplegar en pantalla la clave del inventario que tenga o se haya asignado al componente, la clave del departamento y la clave del laboratorio se asignara de los departamentos y laboratorios previamente dados de alta, la fecha de asignación o ingreso debe tomarse del sistema, a su vez la baja de dicho componente la dará el usuario, una vez obtenido estos datos se genera una etiqueta de con código de barras, la cual contiene todos estos datos para colocársela al componente, y en este modulo muestra el status del componente (0=DISPONIBLE, 1=EN PRESTAMO, 2=GARANTIA Y 3=BAJA)

En cuanto a validaciones, no puede asignar un componente a varios departamentos ni laboratorios, si se intenta ingresar estos datos al sistema debe desplegar el siguiente mensaje según sea el caso: El componente ya a sido asignado

### Prototipo de captura para asignar componentes a los departamentos y laboratorios

#### ASIGNACIÓN DE COMPONENTES

NÚMERO DE INVENTARIO

CLAVE DEL DEPARTAMENTO

CLAVE DEL LABORATORIO

FECHA DE INGRESO

FECHA DE BAJA

ESTADO DEL COMPONENTE

Asignar

Modificar

Borrar

Aceptar

Cancelar

Salir

TESIS CON  
FALLA DE ORIGEN

## Módulo de Solicitud de Salida

## Especificación de Programa

Información General		
Proyecto: SCIFI		Modulo: Solicitud de salida de equipo
Nombre Conceptual: SISTEMA DE CONTROL DE INVENTARIOS DE LA FACULTAD DE INGENIERIA		
Nombre Físicos de los Programas:		
Programa		
Descripción General: Este módulo permite consultar la información correspondiente al inventario existente		
Ambiente del Programa		
Entidad: SOL	Campo: SOL_CLAV, SOL_RESP, SOL_FECH, SOL_TIPO, SOL_DEST, SOL_RESP_DEST, SOL_APRB, SOL_APRB_OK	Tipo de Índice: SOL_CLAV PK
CONT	CONT_CLAV_SOL, CONT_INV_COMP	CONT_INV_SOL_PFK CONT_INV_COMP_PFK
Criterio de Ejecución del Programa: Dentro del menú principal debe existir una opción de solicitud, que se utilizará para el control de salidas de los equipos o componentes.		
Tablas: SOL CONT	Acceso: Insert, Listate, Delete, Select Insert, Update, Delete, Select	Referencia:
Parámetros: No aplica		
Algoritmo		
Cuerpo del programa		
En este modulo sirve para poder solicitar la salida a garantía, préstamo o baja del cualquier componente, previamente autorizado por el jefe del departamento.		
Se debe desplegar en pantalla la clave de la solicitud, esta clave será única e identifica la salida del equipo misma que genera el sistema a través de un contador con incremento de 1, clave del responsable del departamento, fecha de solicitud, tipo de salida (0=préstamo, 1=reparación y 2=baja), a donde se llevará el equipo (destino), un indicador si fue autorizada o rechazada dicha solicitud y por último la confirmación de salida ( 0 = Si salió, 1= No salió).		

TESIS CON  
FALLA DE ORIGEN

## Prototipo de captura de componentes

ASIGNACIÓN DE LABORATORIO	
NÚMERO DE SOLICITUD	
RESPONSABLE	
FECHA DE LA SOLICITUD	Asignar
TIPO DE SOLICITUD	0 PRÉSTAMO 1 REPARACIÓN 2 BAJA
DESTINO	Modificar
SOLICITANTE	Borrar
Sol. Aceptada	Sol. Cancelada
Salió comp.	No Salió comp.
Aceptar	Cancelar
	Salir

### **5.1 Programación**

La programación del sistema, como se menciona en el capítulo 4, se realiza en lenguaje PHP versión 4.05. Para el diseño y creación de imágenes del sistema se utiliza un software conocido como Adobe PhotoShop

Cada programa del que se compone el sistema esta formado por código en lenguaje PHP y fragmentos de código escrito en HTML, quedando ambos integrados en un documento PHP.

El código HTML se utiliza principalmente para establecer las características generales de cada pantalla como el tipo, tamaño y color de letra, manejo de hipervínculos, presentación de imágenes, márgenes, controles visuales, tablas, botones, etc. y en general el aspecto físico de cada módulo. Estas características se conocen también con el nombre de "Frame". Un Frame es una plantilla con ciertas características específicas definidas en la construcción de la misma, en los lenguajes visuales como Visual Basic o Delphi al frame se le conoce como también como forma.

El código escrito en lenguaje HTML que se incrusta en cada documento de tipo PHP tiene como función principal la interacción del sistema con la base de datos, cuyas tareas primordiales son:

- Mantenimiento y acceso a los catálogos del sistema.
- Almacenar y recuperar la información capturada.
- Consulta de la información.
- Control de usuarios.

En total el sistema esta formado por 11 programas escritos en PHP que están integrados en 7 documentos tipo HTML. A continuación se describe brevemente la función de cada uno de ellos:

TESIS CON  
FALLA DE ORIGEN

inv_adm_rnu.php	Menú principal que contiene los vínculos para cada opción del sistema.
inv_adm_usr.php	Administración del catálogo de usuarios, altas, bajas y cambios.
inv_adm_sol.php	Captura de solicitudes de préstamo de equipo de cómputo.
inv_adm_pro.php	Mantenimiento del catálogo de proveedores
inv_adm_mst.php	Acceso al módulo de herramientas del sistema, entre las que se encuentra la impresión etiquetas.
inv_adm_lab.php	Control de laboratorios de la Facultad de Ingeniería
inv_adm_inv.php	Mantenimiento y administración del inventario de equipo de cómputo de la Facultad de Ingeniería
inv_adm_her.php	Impresión de reportes y códigos de barras faltantes
inv_adm_dev.php	Devolución de equipo de cómputo
inv_adm_dep.php	Control de Departamentos de la Facultad de Ingeniería
inv_adm_com.php	Mantenimiento al catálogo de componentes de equipo de cómputo

Programas que integran el Sistema de Control de Inventarios de la Facultad de Ingeniería SCIFI

Para escribir el código de cada uno de los programas fuentes del sistema (en PHP) sólo es necesario un editor de texto y en el mercado existen una gran variedad de ellos. Por un lado están los generadores de código HTML como el FrontPage, que generan código de manera automática al momento de ir diseñando gráficamente el Frame; para la edición de los programas fuentes de PHP se puede utilizar cualquier editor de texto (incluso el Notepad de Windows); pero por simplicidad de las tareas de edición de los dos tipos de código (HTML y PHP) se utiliza un software conocido como HomeSite. Las principales razones por las que se eligió este editor son las siguientes:

- Tiene una interfaz gráfica de muy fácil manejo.
- Permite la escritura de código HTML combinado con PHP, ASP, JAVA
- Tiene una herramienta que permite visualizar la construcción de cada documento HTML, Frame o pantalla al momento de ir escribiendo el código.
- Contiene un conjunto de herramientas que permiten detectar y corregir errores de forma eficiente.
- Presenta un inspector que localiza cada uno de los elementos del programa fuente como etiquetas, funciones, procedimientos, referencia a diversos objetos, etc.



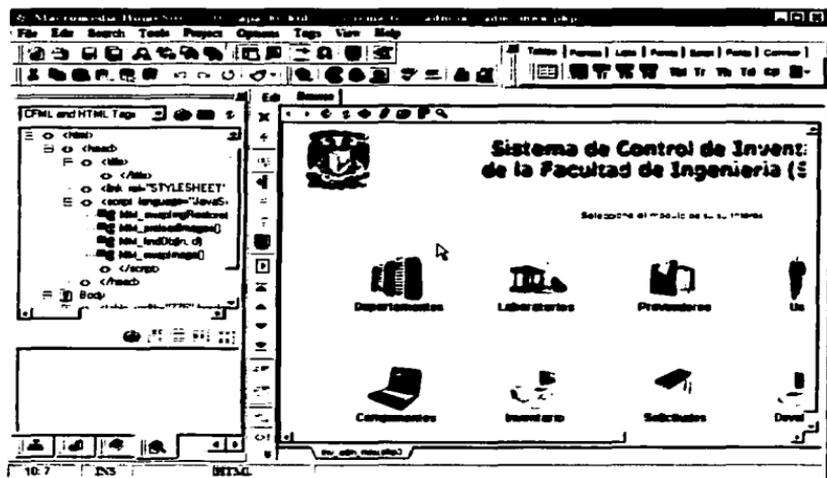


Figura 5.2 Interfaz gráfica del software HomeSite en modo de Previsualización

Existe un menú principal del sistema que contiene todas las llamadas a los demás módulos que lo integran. Para dar una idea general de cómo están elaborados los módulos en PHP se muestra a continuación el código fuente del programa principal.

## Código fuente del programa principal del SCIF1

```

<?PHP
function autentifica(){
global $logi,$pass;
if(!$logi||!$pass){ header ("Location: index.html"); exit;}
// conexion a la base
$db = mysql_connect ("localhost", "tesis", "tesis");
if(!$db){
readfile ("inv_adm_cab.html");
printf("<p><br><h3 align=center>Servicio temporalmente suspendido</h3>\n");
readfile ("inv_adm_pie.html");
exit;
}
mysql_select_db ("tesis",$db);
$result = mysql_query ("select * from usr where usr_login='$logi'",$db);
$rows = mysql_fetch_array ($result);
$rows = mysql_num_rows ($result);
$pass=$row["usr_pass"];
$tipo=$row["usr_tipo"];
if($pass==$pass){
setcookie ("cookie[aut]","1",time()+3600, "/", $HTTP_HOST);
setcookie ("cookie[user]",$logi,time()+3600, "/", $HTTP_HOST);
setcookie ("cookie[tpo]",$tipo,time()+3600, "/", $HTTP_HOST);
header ("Location: inv_adm_mnu.php3");
exit;
}
else{
readfile ("inv_adm_cab.html");
if($rows=0)
printf("<p><br><br><h3 align=center>Usuario inexistente </h3><br><br>\n");
if($rows=1)
printf("<p><br><br><h3 align=center>Password incorrecto</h3><br><br>\n");
readfile ("inv_adm_pie.html");
}
}
// modulo maestro del programa de inventarios
switch ($mod) {
case "aut": autentifica(); break; // autentificacion
case "dep": header ("Location: inv_adm_dep.php3"); break; // departamentos
case "lab": header ("Location: inv_adm_lab.php3"); break; // laboratorios
case "usr": header ("Location: inv_adm_usr.php3"); break; // usuarios
case "her": header ("Location: inv_adm_her.php3"); break; // herramientas
case "com": header ("Location: inv_adm_com.php3"); break; // componentes
case "pro": header ("Location: inv_adm_pro.php3"); break; // proveedores
case "inv": header ("Location: inv_adm_inv.php3"); break; // inventario
case "sol": header ("Location: inv_adm_sol.php3"); break; // solicitudes
case "dev": header ("Location: inv_adm_dev.php3"); break; // devoluciones
case "mnu": header ("Location: inv_adm_mnu.php3"); break; // menu principal
default: autentifica();
}
?>

```

TESIS CON  
FALLA DE ORIGEN

## 5.2 Pruebas del sistema

Un aspecto importante de la calidad del sistema es su confiabilidad. Un sistema es confiable si al usarse de manera razonable no produce fallas peligrosas o costosas. Aunque es prácticamente imposible desarrollar software que se pueda demostrar que este libre de errores, se debe hacer cuanto esté al alcance por prevenir su aparición, usando métodos y técnicas que incluyan la detección de errores, su corrección y tolerancia.

Las pruebas son procesos de ejecución del programa con la intención de hallar errores, es decir, hacer que el programa falle. Así, una prueba exitosa es la que encuentra un error.

Cabe mencionar que para el presente trabajo se emplearon cinco estrategias de pruebas, de las cuales a continuación se comentan los resultados

### Prueba alfa

Es lo que se conoce como verificación y tiene la intención de hallar errores al ejecutar el programa en un ambiente simulado. Por lo general, se solicita la intervención de un usuario para que ejecute la parte a verificar, mientras se observan y anotan los errores en los que se incurre durante su ejecución, así como las reacciones del usuario ante el programa.

Las expectativas de esta prueba son que el usuario aporte sus observaciones para evaluar las modificaciones con respecto a las ventanas que se le van presentando, si considera que su uso es complejo, su funcionalidad, entre otros comentarios.

En este caso se creó una versión alfa del sistema, y se solicitó la cooperación de un futuro usuario para que operara el mismo con datos ficticios. Se pudieron detectar, entre otros, los siguientes errores:

- Faltaban iconos de salida en algunas ventanas.
- No estaban deshabilitados algunos iconos que por seguridad debían estarlo

- En la solicitud de salida de equipo, el sistema no verificaba que el inventario concordara con el laboratorio al cual estaba asignado.
- Algunas ventanas no estaban centradas.

#### **Pruebas de recuperación**

La mayor parte de los sistemas deben tener la capacidad de recuperarse de los fallos ocasionados cuando las máquinas "se caen" o "se traban". Además esta operación de recuperación no debe perjudicar los datos almacenados en el sistema. Es por ello que las pruebas realizadas para este caso fueron en diferentes máquinas: unas con recursos de sobra y otras con menos capacidad de la requerida. Cabe aclarar que la parte evaluada fue solamente la del cliente.

Se observó que las computadoras con menos recursos eran más susceptibles a un bloqueo, teniendo esto como consecuencia el reiniciar el quipo y perder tiempo en dicho proceso. Por el contrario, las máquinas con más recursos no presentan problema alguno durante la ejecución del sistema.

Estas dos configuraciones no afectan directamente los archivos utilizados por sistema, ya que estos se encuentran localizados en el servidor, el cual demostró no tener problemas al recuperarse en un muy corto periodo de tiempo.

#### **Pruebas de seguridad**

Todos lo sistemas manejan información sensible, esto quiere decir importante, a la cual no todos los usuarios pueden tener acceso, principalmente por cuestiones de seguridad en cuanto a la integridad y fiabilidad de los datos almacenados. Es por ello que en esta prueba, se verifica que los mecanismos de protección incorporados al sistema funcionen adecuadamente, es decir, se verifica que valide el acceso al sistema de acuerdo a los diferentes privilegios asignados a cada tipo de usuario. Al respecto podemos comentar que no hubo problemas, ya que no se encontró error alguno.

### **Pruebas de resistencia**

En esta prueba se ejecuta el sistema de forma que demande recursos en cantidad, frecuencia, o volúmenes muy grandes de datos, por ejemplo:

- Incrementar la cantidad de datos de entrada con el fin de comprobar como responden las funciones de entrada ante tantas solicitudes al mismo tiempo.
- Realizar pruebas que requieran el máximo de memoria o de otros recursos.
- Realizar excesivas búsquedas de datos residentes en disco.
- Intentar "tirar" el sistema.

La respuesta del sistema a este tipo de pruebas fue un ligero incremento en el tiempo de espera de la respuesta del servidor, pero no fue lo suficientemente significativo como para considerarlo una falla mayor, sino que permanece en un grado aceptable.

### **Pruebas de desempeño**

Las pruebas de desempeño del hardware del servidor se tuvieron que realizar con un equipo diferente al propuesto, ya que por diversas cuestiones no fue posible conseguir el hardware propuesto, y se tuvo que adaptar a los recursos disponibles. Sin embargo el sistema a pesar de no estar en condiciones óptimas presentó un buen desempeño.

Por otra parte también se verificaron aspectos del desempeño de la red de comunicaciones, tales como:

- Comportamiento del servidor
- Tiempos de acceso al sistema
- Tiempos de actualización, borrado y consulta de los registros existentes
- Número de conexiones máximas.

### 5.3 Instalación

Considerando que se tiene una computadora con las características requeridas en el Capítulo 4 para el buen funcionamiento del sistema y además instalada una versión de Windows 95 o superior, se procede a la instalación del PHP Triad para Windows, que es un instalador gratuito del servidor de Web Apache, Servidor de base de datos MySQL y el módulo para la programación de PHP.

Una vez instalado el PHP Triad, es recomendable verificar si efectivamente funciona Apache y MySQL. Para hacer esta verificación, se puede realizar lo siguiente:

- En el botón Inicio (de la barra de tareas de Windows) ir a Programs -> PHP Triads, como se muestra a continuación en la Figura 5.3

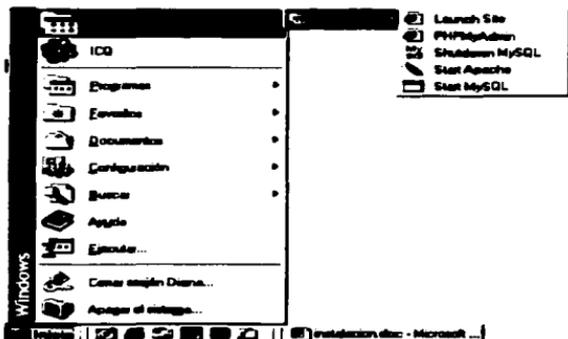


Fig. 5.3 Búsqueda del PHP Triad en Windows

- Dar clic en "Start Apache"
- Aparecerá una ventana de MS-DOS con el siguiente mensaje:

```
[Thu Nov 21 19:44:39 2002] [warn] pid file /apache/logs/httpd.pid overwritten --
Unclean shutdown of previous Apache run?
Apache/1.3.14 (Win32) running...
```

Indicando que el servidor Web está funcionando adecuadamente.

Posteriormente, se levanta el servidor de bases de datos MySQL:

- En el botón Inicio (de la barra de tareas de Windows) ir a Programs -> PHP Triads
- Dar clic en "Start MySQL"
- Aparecerá una ventana de MS-DOS llamada mysqld y luego desaparecerá, indicando que el servidor de bases de datos MySQL está corriendo.

La instalación de PHP triad crea una carpeta en la raíz del disco duro llamada "apache". Dentro de dicha carpeta es donde se encuentran los programas y archivos necesarios para la correcta funcionalidad del sistema, pero además, dentro de la carpeta apache existe una carpeta llamada "htdocs", que es la carpeta donde se depositan los documentos que se desea que el Servidor Apache publique en internet. Para fines prácticos y de fácil identificación, dentro de la carpeta htdocs se crea una carpeta llamada "sistema" y dentro de esta misma se crea otra carpeta llamada igual que la base de datos (en este caso, la base de datos se llama "tesis"), donde estarán guardados los programas PHP, los documentos html y las imágenes del sistema.

Una vez creada la carpeta "tesis" y teniendo toda la información en ella, se procede a crear la base de datos para luego subir las tablas del sistema. Esto se realiza así:

- Ir a c:\apache\mysql\bin
- Dar doble clic en mysql.exe

TESIS CON  
FALLA DE ORIGEN

- Aparece una ventana de comandos de MS-DOS con un prompt que dice `mysql>`
- Mediante el script "crea\_base.sql" se crea la base de datos y una cuenta de usuario llamado "tesis" al que se le otorgan los permisos necesarios para manipular la base de datos y que también es utilizada por todos los programas PHP para la explotación de la base de datos. El comando es:

```
mysql > source c:\apache\htdocs\sistema\tesis\crea_base.sql;
```

- Ahora, cargar las tablas que ya están en un script, contenido en un archivo llamado "tesis.sql", ubicado en c:\apache\htdocs\tesis\ con el comando siguiente:

```
mysql > use tesis;
```

```
mysql > source c:\apache\htdocs\sistema\tesis\tesis.sql;
```

La base de datos ha sido creada, así como las tablas que la conforman. Sólo falta verificar que el sistema funcione. Para esto, se pueden llevar a cabo los siguientes pasos:

- Abrir un navegador de Internet (por ejemplo, Internet Explorer).
- En el campo "Dirección" teclear:  
<http://localhost/sistema/tesis/>  
o bien, la dirección IP de la computadora, por ejemplo:  
<http://127.0.0.1/sistema/tesis/>
- Se muestra la pantalla que aparece en la Figura 5.4, indicando esto que se puede trabajar en el sistema.

TESIS CON  
FALLA DE ORIGEN

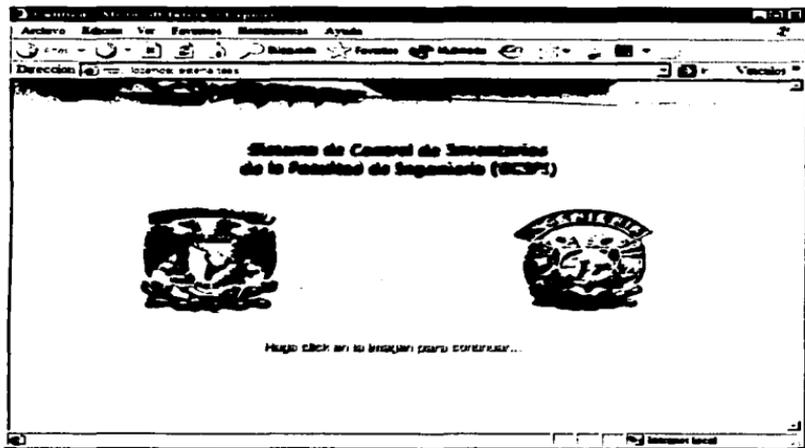


Fig. 5.4 Pantalla de presentación del SCIFI

## 5.4 Seguridad

Hablar de seguridad en un sistema es muy complejo, debido a que se deben definir las funciones de cada usuario dentro del sistema, o mejor aún, restringir el acceso al sistema en primera instancia y si al usuario se le permite entrar y/o utilizarlo, qué menús, sub-menús, pantallas o datos puede ver, modificar o en dado caso, eliminar.

Antes de poder tener un sistema de seguridad confiable, podríamos pensar en lo siguiente:

- Quién debe usar el sistema.
- A quién se le otorgará acceso.
- Para qué se otorga el acceso.
- Cuándo se requiere el acceso.
- Desde dónde se hará el acceso.
- Cómo se obtendrá la información.
- Cómo se usará la información.

La seguridad que emplearemos en este sistema se divide en cinco perfiles, niveles o funciones de usuarios, dependiendo de su actividad diaria y procedimientos que lleva a cabo dentro de la Facultad de Ingeniería. Los tipos de usuarios válidos se muestran a continuación:

Nivel	Nombre	Funciones dentro del sistema	Función Real
0	Jefe de Departamento	<p>Puede consultar todos los componentes inventariados, así como sus especificaciones o detalles, pero sin poder borrar o agregar alguno.</p> <p>Las solicitudes las puede consultar, crear y aprobar de acuerdo a quién las solicita, sin poder borrar alguna.</p> <p>Puede visualizar los equipos que hayan regresado de cualquier préstamo o salida por reparación.</p>	El jefe de Departamento es el encargado de aprobar las solicitudes de préstamo o salidas de cada uno de los componentes.
1	Jefe de Laboratorio	<p>Puede consultar todos los componentes inventariados así como sus especificaciones, sin poder borrar o agregar alguno.</p> <p>Las solicitudes las puede consultar y crear, sin poder borrar ni autorizar alguna.</p> <p>Puede visualizar los equipos que hayan regresado de cualquier préstamo o salida.</p>	Es el encargado de elaborar la solicitud de préstamo o salida de algún equipo.

TESIS CON  
 FALLA DE ORIGEN

Nivel	Nombre	Funciones dentro del sistema	Función Real
2	Ayudante de Laboratorio	<p>Puede consultar todos los componentes inventariados así como sus especificaciones, sin poder borrar o agregar alguno.</p> <p>Las solicitudes las puede consultar y crear, sin poder borrar ni autorizar alguna.</p> <p>Puede visualizar los equipos que hayan regresado de cualquier préstamo o salida.</p>	Es el encargado de elaborar la solicitud de préstamo o salida de algún equipo
3	Vigilante	Puede consultar y/o verificar si las solicitudes están o no aprobadas, para después corroborar que salga el equipo o componentes listados, poniendo una marca de salida.	El vigilante verifica si la solicitud fue aprobada por el jefe de Departamento, para después corroborar que todos los componentes listados en la misma son los que salen
4	Administrador	<p>Puede actualizar, agregar, borrar y consultar todas las bases de datos tales como departamentos, laboratorios, usuarios, componentes, proveedores, inventarios, solicitudes, devoluciones.</p> <p>Puede imprimir y dar un listado de los componentes en préstamo o garantía, según sea el estado de cada uno, además de poder consultar, agregar, actualizar y borrar dichos componentes.</p> <p>Puede imprimir las etiquetas con las que serán identificados los componentes.</p>	<p>El administrador es el encargado de dar mantenimiento al sistema.</p> <p>Es el único que puede agregar, consultar, actualizar y borrar, cada uno de los departamentos, laboratorios, usuarios, proveedores, componentes, etc.</p> <p>Se encarga de imprimir las etiquetas que se les asignarán a cada componente.</p> <p>Genera un informe de los componentes o equipo que se encuentre en préstamo o garantía</p>

**TESIS CON  
FALLA DE ORIGEN**

## 5.5 Documentación

En la creación de cualquier sistema informático existe un documento en el cual se especifican detalladamente las pantallas que va presentando el sistema, para que este pueda ser entendido por la persona que va a utilizarlo.

### 5.5.1 Pantallas

Esta sección muestra las pantallas que conforman al Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI).

Junto con la pantalla se da una breve explicación de la misma. La visualización de las mismas es en Internet Explorer de Microsoft y se asumen conocimientos básicos en el manejo de dicho navegador.

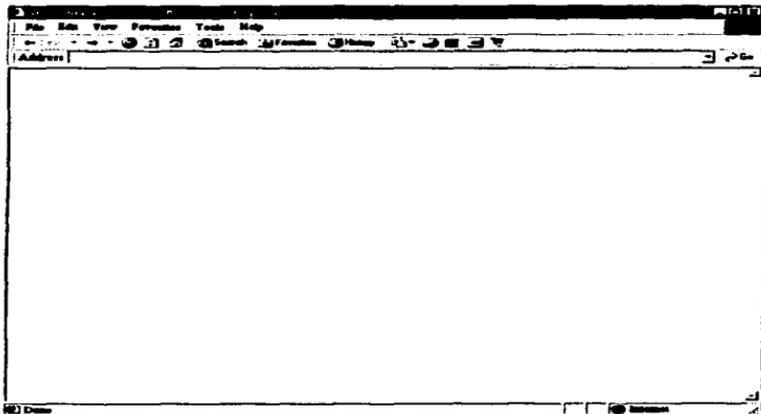


Figura 5.5 Pantalla de Internet Explorer

El Acceso al sistema comienza con la pantalla de la Figura 5.6. Al presionar el botón izquierdo del mouse sobre la imagen que se presenta en el browser se muestra la pantalla principal del sistema ilustrada en la Figura 5.7

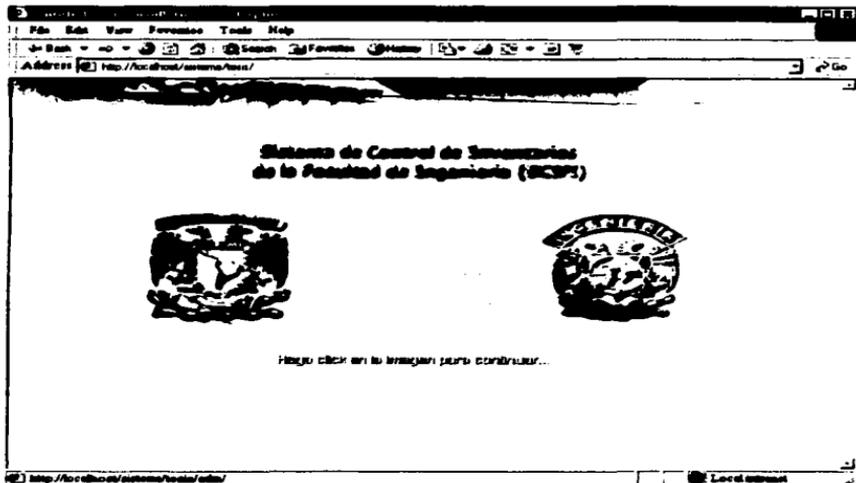


Figura 5.6 Pantalla de Entrada al SCIFI

Antes de continuar es necesario considerar las siguientes imágenes, las cuales se presentan en las pantallas restantes del sistema.

-  Icono de Actualizar Registro
-  Icono Estatus de Solicitud Aprobada
-  Icono de Borrar Registro
-  Icono de Solicitud Confirmada
-  Icono de Ir a Menú Principal
-  Icono de Insertar Nuevo Registro
-  Icono de Enviar Mail

Estas imágenes nos ayudarán a identificar los procesos que se pueden realizar en cada una de las pantallas que se presentan a continuación.

Además, es necesario comentar que en esta sección se muestran las pantallas que conforman el sistema, considerando que la presentación de las mismas depende del perfil del Usuario que entra al sistema, por lo que pueden variar.

La pantalla de ingreso al sistema se encuentra ilustrada en la Figura 5.7

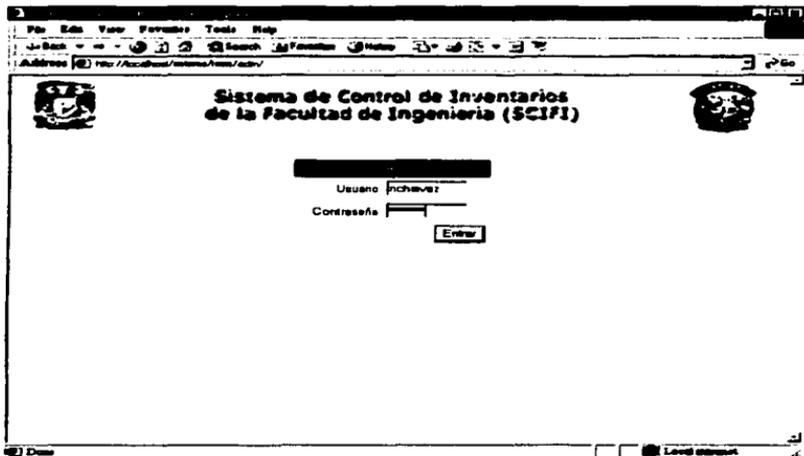


Figura 5.7 Pantalla de Ingreso al Sistema

Esta pantalla espera el ingreso de la clave de Usuario y la Contraseña asignada a este para tener acceso al sistema.

Un usuario puede estar registrado en la base de datos del sistema, pero no introducir correctamente el password, lo que provoca que no pueda ingresar al sistema. Esta situación se ilustra en la Figura 5.8

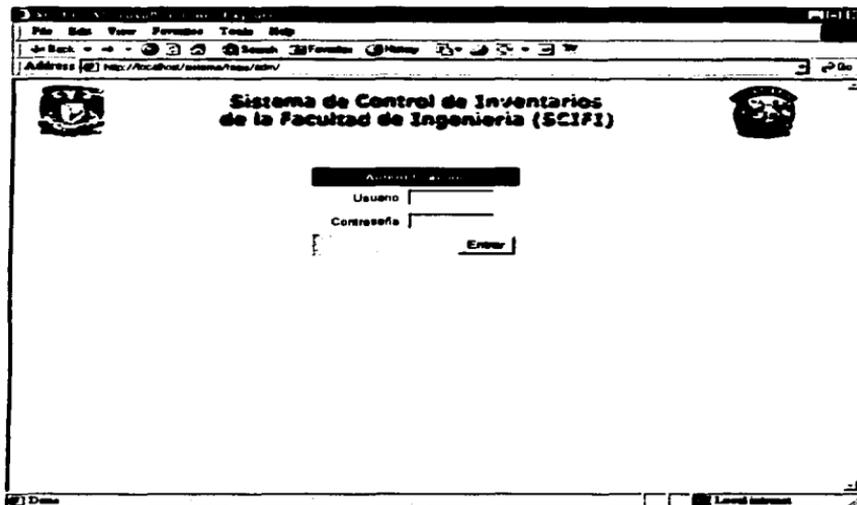


Figura 5.8 Usuario Válido – Password no válido

También un usuario que intente ingresar al sistema pero que no este dado de alta en la base de datos del propio sistema. Esta situación provoca que el usuario no pueda ingresar y se ilustra en la Figura 5.9

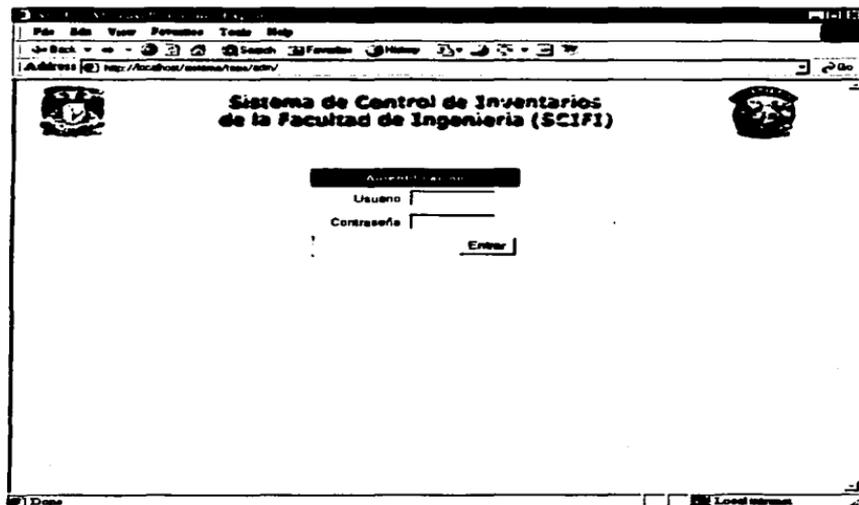


Figura 5.9 Usuario No Válido

Si por el contrario el Usuario que intenta ingresar al sistema es válido y proporciona su password correctamente, entonces ingresa al sistema sin ningún problema y aparece la pantalla ilustrada en la Figura 5.10

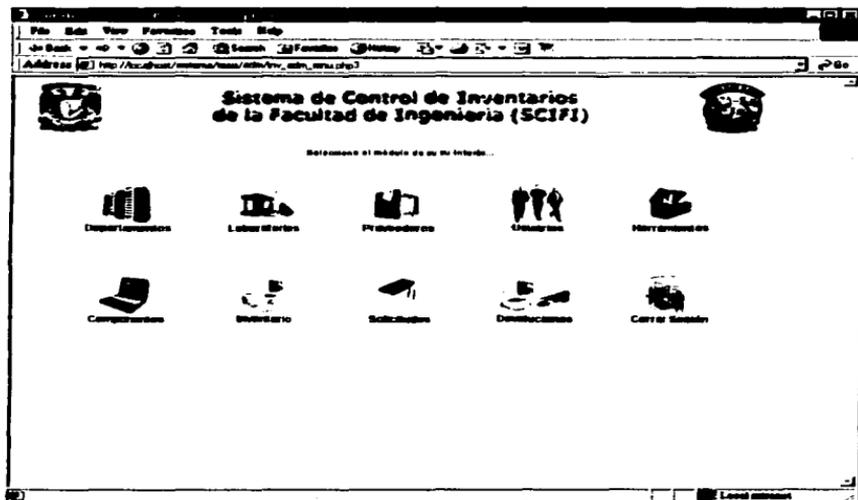


Figura 5.10 Pantalla Principal del SCIFI

En esta pantalla se despliega el menú principal del sistema.

Se puede seleccionar cualquiera de las opciones mostradas presionando el botón izquierdo del mouse sobre ellas para tener acceso a la pantalla correspondiente a la opción seleccionada.

La primer pantalla que se obtiene es la del **Módulo de Departamentos** y se ilustra en la Figura 5.11

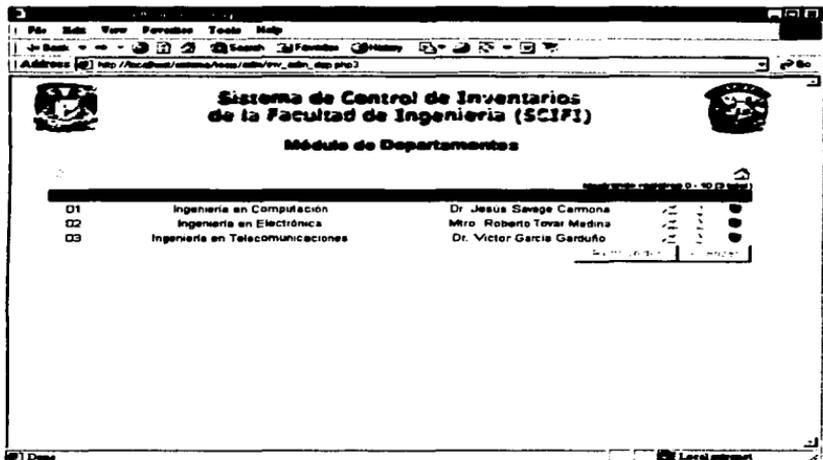


Figura 5.11 Pantalla del Módulo de Departamentos

**Avanzar**

Avanza al siguiente grupo de registros encontrados

**Retroceder**

Retrocede al grupo anterior de registros encontrados

En la pantalla de la Figura 5.11, al presionar el icono  se elimina de la base de datos el registro seleccionado

En la pantalla de la Figura 5.11, al presionar el icono  se presenta la pantalla de la Figura 5.12

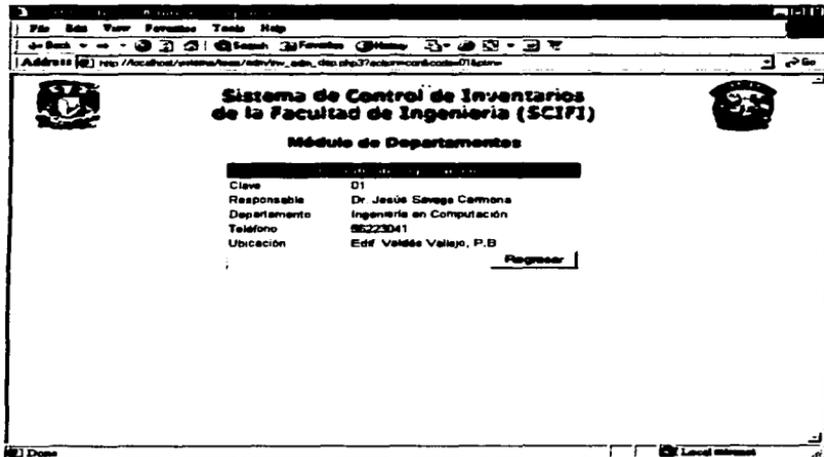
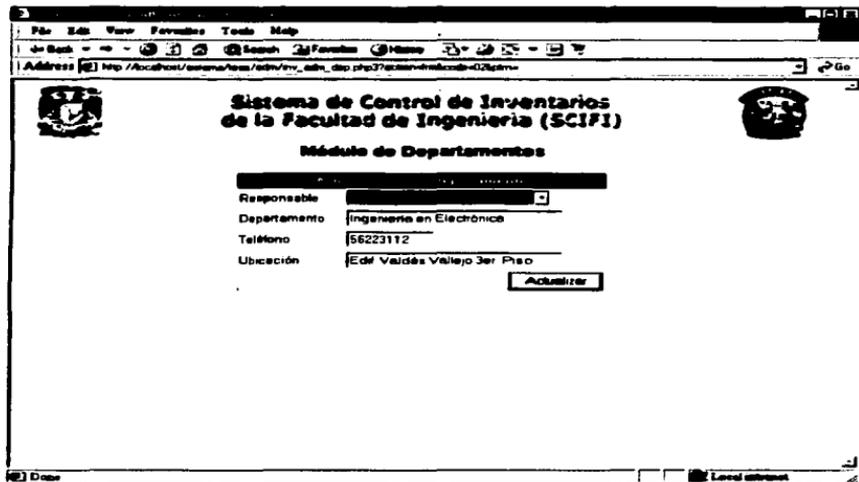


Figura 5.12 Pantalla de Consulta de Departamento

[Regresar](#)

Regresa a la pantalla anterior.

En la pantalla de la Figura 5.11, al presionar el icono  se presenta la pantalla de la Figura 5.13



**Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)**

**Módulo de Departamentos**

Responsable

Departamento

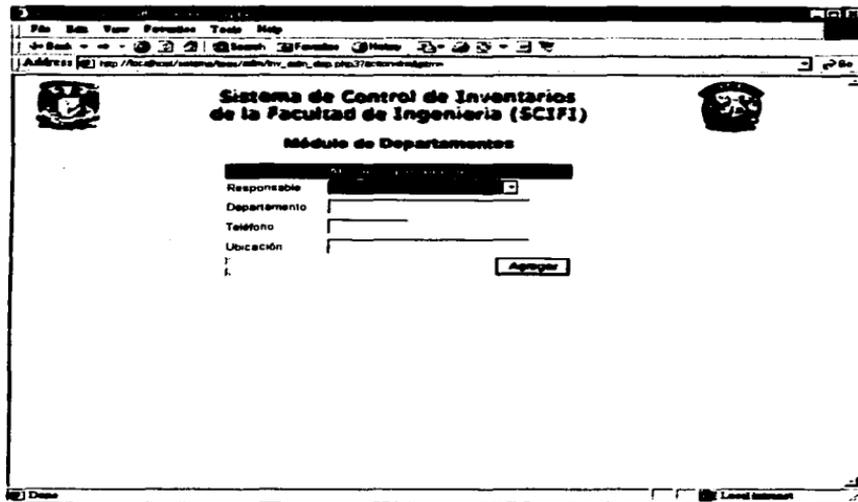
Teléfono

Ubicación

Figura 5.13 Pantalla de Actualización de Departamento

Actualiza el registro en pantalla en la base de datos.

En la pantalla de la Figura 5.11 al presionar el icono  se presenta una pantalla similar a la ilustrada en la Figura 5.13. Obsérvese la Figura 5.14



The screenshot shows a web browser window with the following content:

- Browser title: Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)
- Page title: Módulo de Departamentos
- Form fields:
  - Responsable:
  - Departamento:
  - Teléfono:
  - Ubicación:
- Buttons:

Figura 5.14 Pantalla de Alta de Departamento

Es necesario ingresar los datos requeridos y presionar el botón  para afectar el registro en la base de datos.

La pantalla del Módulo de Laboratorios se ilustra en la Figura 5.15

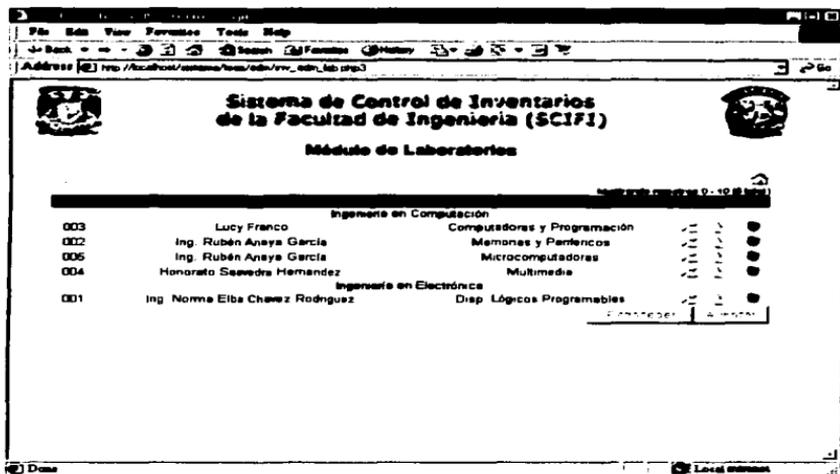


Figura 5.15 Pantalla del Módulo de Laboratorios

**Avanzar**

Avanza al siguiente grupo de diez registros encontrados

**Retroceder**

Retrocede al grupo anterior de diez registros encontrados

En la pantalla de la Figura 5.15, al presionar el icono ● se elimina de la base de datos el registro seleccionado

En la pantalla de la Figura 5.15, al presionar el icono  se presenta la pantalla de la Figura 5.16

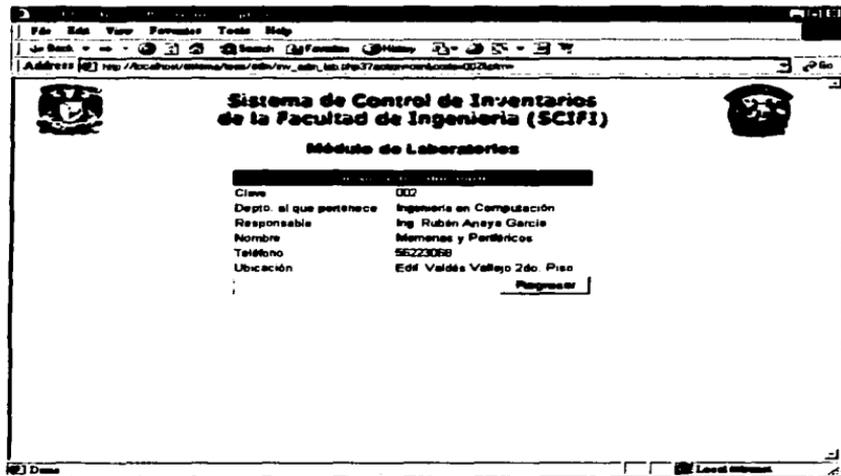


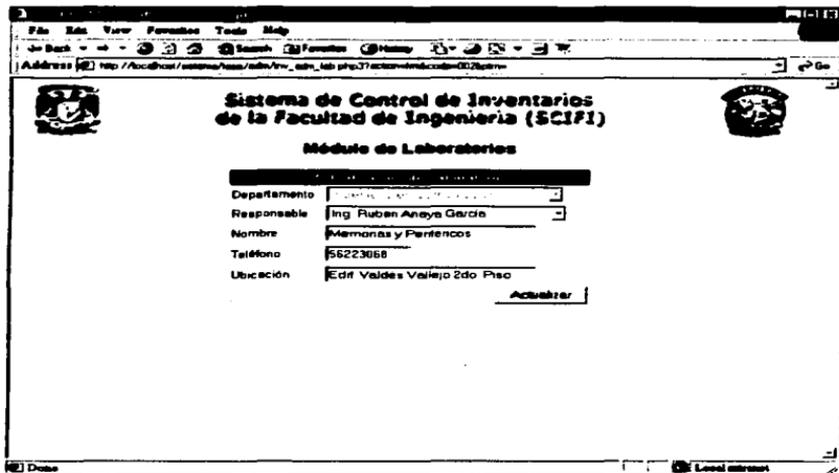
Figura 5.16 Pantalla de Consulta de Laboratorios

[Regresar](#)

Regresa a la pantalla anterior.

TESIS CON  
FALLA DE ORIGEN

En la pantalla de la Figura 5.15, al presionar el icono  se presenta la pantalla de la Figura 5.17



**Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)**

**Módulo de Laboratorios**

Departamento:

Responsable:

Nombre:

Teléfono:

Ubicación:

Figura 5.17 Pantalla de Actualización de Laboratorio

**Actualizar**

Actualiza el registro en pantalla en la base de datos.

En la pantalla de la Figura 5.15, al presionar el icono  se presenta una pantalla similar a la ilustrada en la Figura 5.17, como se puede observar en la Figura 5.18

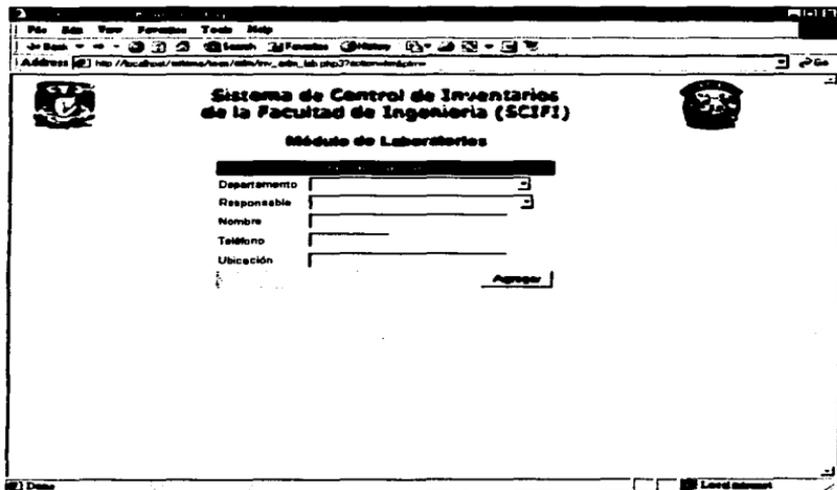


Figura 5.18 Pantalla de Alta de Laboratorio

Es necesario ingresar los datos requeridos y presionar el botón **Agregar** para afectar el registro en la base de datos.

La pantalla del Módulo de Usuarios se ilustra en la Figura 5.19

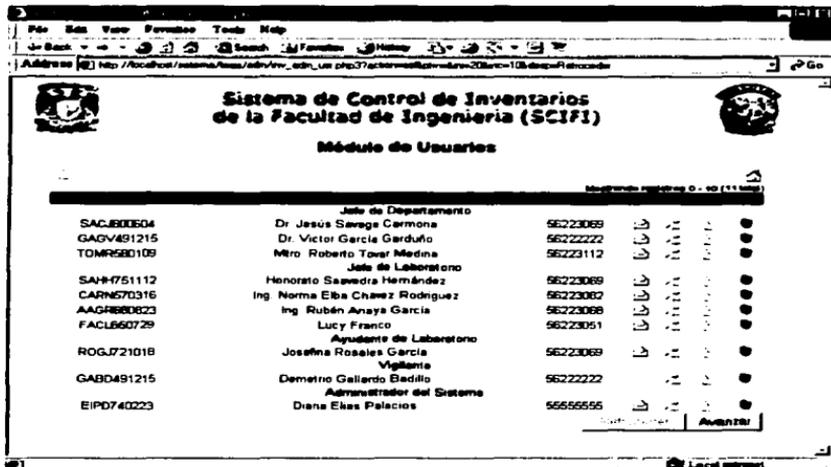


Figura 5.19 Pantalla de Módulo de Usuarios

**Avanzar**

Avanza al siguiente grupo de registros encontrados

**Retroceder**

Retrocede al grupo anterior de registros encontrados

En la pantalla de la Figura 5.19, al presionar el icono  se elimina de la base de datos el registro seleccionado.

En la pantalla de la Figura 5.19 al presionar el icono  se presenta la pantalla de la Figura 5.20. Esta pantalla corresponde al envío de correo electrónico y está sujeta al servicio de Mail administrado por el servidor Web.

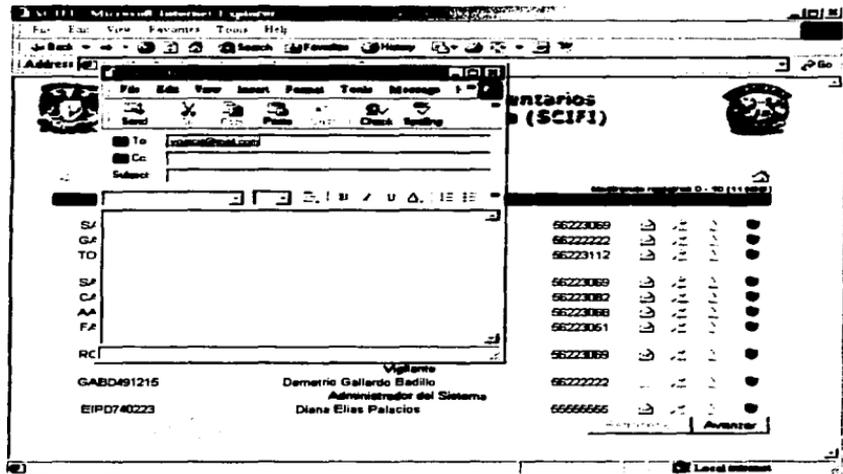


Figura 5.20 Pantalla de Envío de Correo Electrónico al usuario

TESIS CON  
FALLA DE ORIGEN

En la pantalla de la Figura 5.19, al presionar el icono  se presenta la pantalla de la Figura 5.21

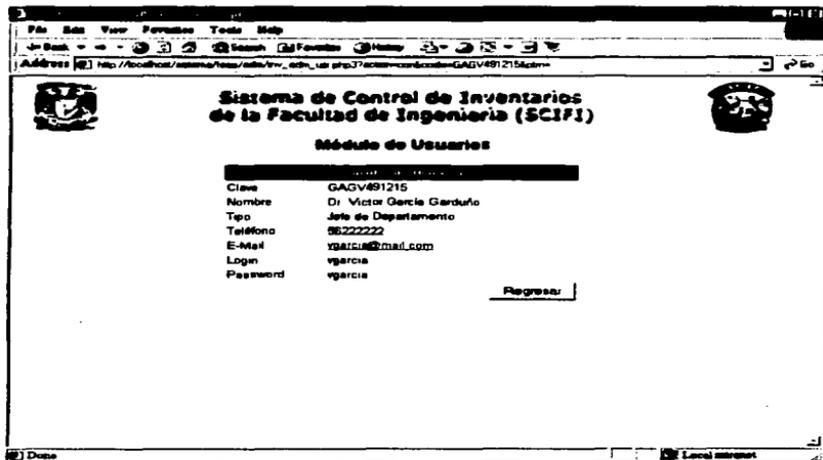
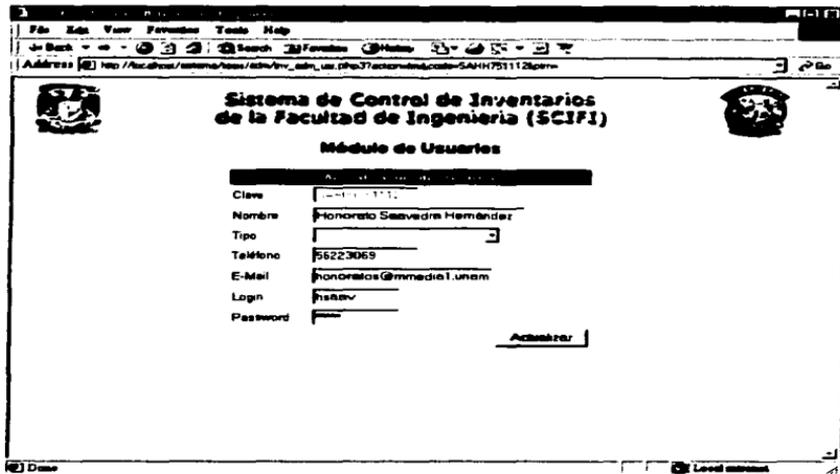


Figura 5.21 Pantalla de Consulta de Usuarios

[Regresar](#)

Regresa a la pantalla anterior.

En la pantalla de la Figura 5.19, al presionar el icono  se presenta la pantalla de la Figura 5.22



**Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)**

**Módulo de Usuarios**

Clave	<input type="text" value="123456789"/>
Nombre	<input type="text" value="Honoreto Saavedra Hernández"/>
Tipo	<input type="text" value=""/>
Teléfono	<input type="text" value="56223069"/>
E-Mail	<input type="text" value="honoretos@mede1.unam"/>
Login	<input type="text" value="hsav"/>
Password	<input type="password" value=""/>

Figura 5.22 Pantalla de Actualización de Usuarios

Actualiza el registro en pantalla en la base de datos.

TESIS CON  
FALLA DE ORIGEN



La pantalla del Módulo de Componente se ilustra en la Figura 5.24

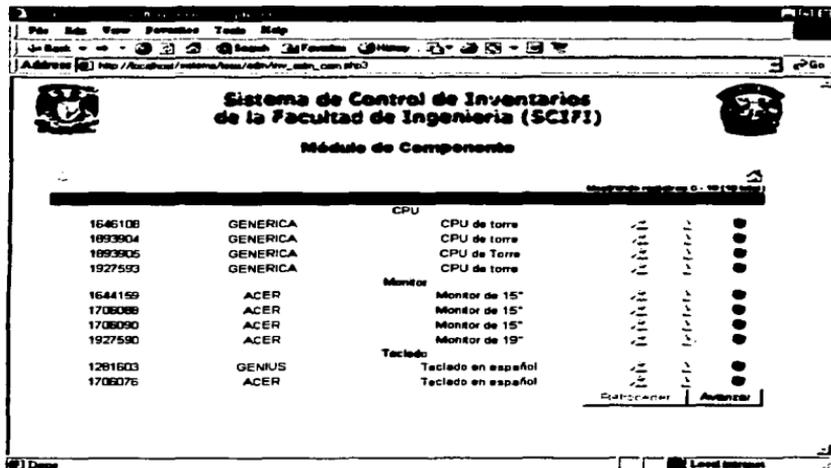


Figura 5.24 Pantalla de Módulo de Componente

**Avanzar**

Avanza al siguiente grupo de 10 registros encontrados

**Retroceder**

Retrocede al grupo anterior de 10 registros encontrados

En la pantalla de la Figura 5.24, al presionar el icono  se elimina de la base de datos el registro seleccionado

TESIS CON  
FALLA DE ORIGEN

En la pantalla de la Figura 5.24, al presionar el icono  se presenta la pantalla de la Figura 5.25

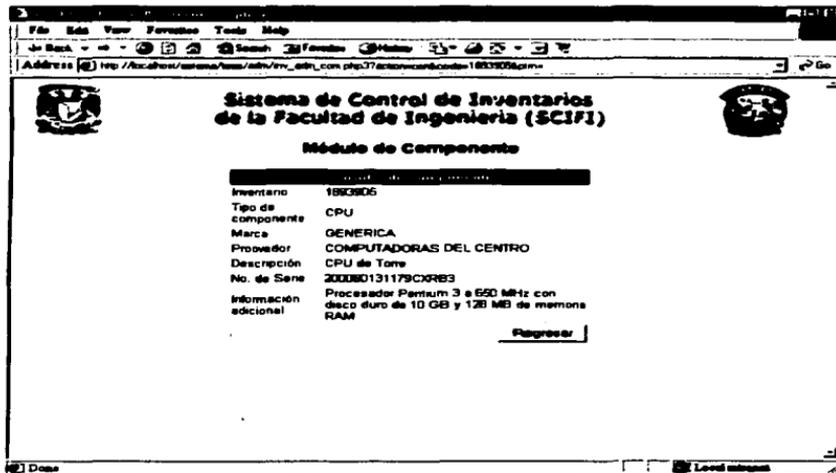
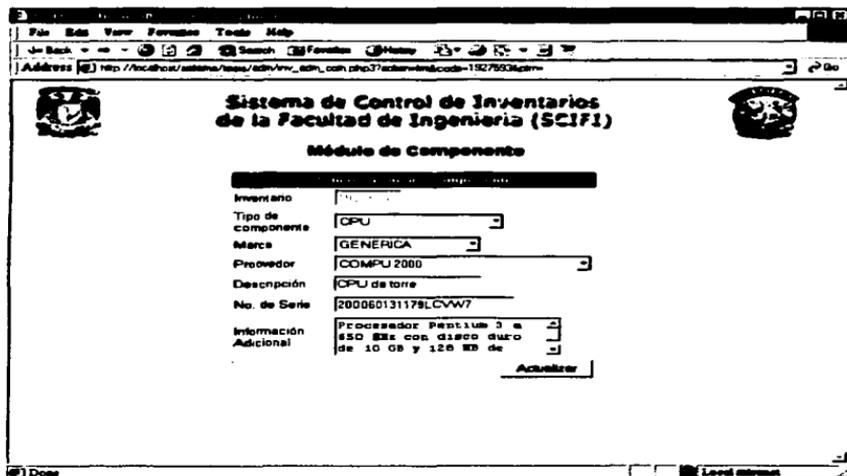


Figura 5.25 Pantalla de Consulta de Componente

[Regresar](#)

Regresa a la pantalla anterior.

En la pantalla de la Figura 5.24, al presionar el icono  se presenta la pantalla de la Figura 5.26



**Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)**

**Módulo de Componente**

Inventario	
Tipo de componente	CPU
Marca	GENÉRICA
Proveedor	COMPU 2000
Descripción	CPU de torre
No. de Serie	200060131179LCVW7
Información Adicional	Procesador Pentium 3 a 650 MHz con disco duro de 10 GB y 128 MB de RAM

**Actualizar**

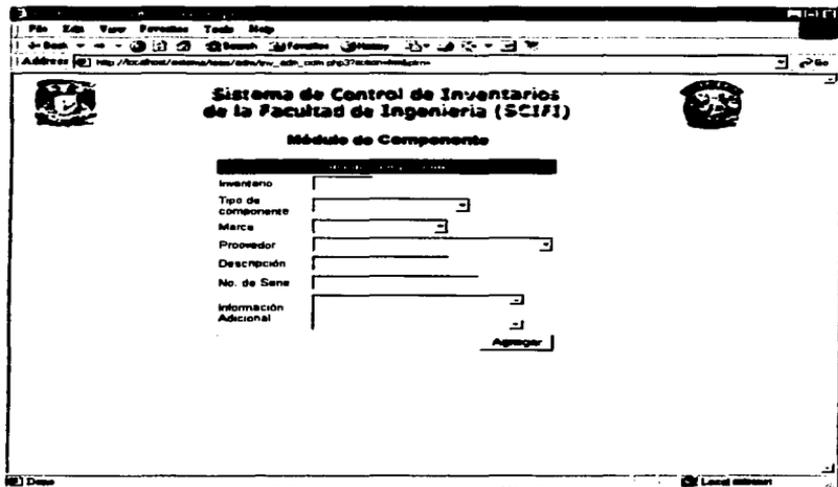
Figura 5.26 Pantalla de Actualización de Componente

**Actualizar**

Actualiza el registro en pantalla en la base de datos.

TESIS CON  
FALLA DE ORIGEN

En la pantalla de la Figura 5.24, al presionar el icono  se presenta una pantalla similar a la ilustrada en la Figura 5.26, como puede observarse en la Figura 5.27



The screenshot shows a web browser window with the following content:

- Browser title: Sistema de Control de Inventarios de la Facultad de Ingeniería (SCI/1)
- Page title: Módulo de Componente
- Form fields:
  - Inventario: [input type="text"]
  - Tipo de componente: [input type="text"]
  - Marca: [input type="text"]
  - Proveedor: [input type="text"]
  - Descripción: [input type="text"]
  - No. de Serie: [input type="text"]
  - Información Adicional: [input type="text"]
- Buttons: [Agregar]

Figura 5.27 Pantalla de Alta de Componente

Es necesario ingresar los datos requeridos y presionar el botón **Agregar** para afectar el registro en la base de datos.

La pantalla del Módulo de Proveedores se ilustra en la Figura 5.28

ID	Proveedor	Contacto	Dirección	Teléfono
01	ALBATROS	Lucrecia González	Panamericana 74 Col. Panamericana	55157210
02	ASESORIA EN COMPUTO SA DE CV	Cintya Martínez	San Simón 201 Col. San Rafael	53945352
03	CIASA COMERCIAL SA DE COMPU 2000	Enka León	Tejocotes 88 Col. Pasteros	53945678
04	COMPUTADORAS DEL CENTRO	Raúl Vázquez	Logroño 90 Col. Postal	56908451
05	COMPUTO 2000	Virginia Hernández	Pancho López 90 Col. Benito Juárez	57564301
06	DELL COMPUTER DE MEXICO	Alejandra Lara	Lago Chalco 200 Col. Agua Azul	57368888
07	HERGO COMPUTER SA DE CV	Lic. Mimam Mejía	San Mateo 8 Col. La Preciosa	56363412
08	INTERTEC DE MEXICO SA DE CV	Gustavo Mejía	República de Uruguay 10 Col. Centro	56297545
09	PC CHEAP SA DE CV	Ivan Reyes	Calle 7 No. 185 Col. Panislán	55599137
10		Ismail Montoya	Carlos J. Miness 98 Col. Guerrero	55590012

Figura 5.28 Pantalla de Módulo de Proveedores

**Avanzar**

Avanza al siguiente grupo de registros encontrados

**Retroceder**

Retrocede al grupo anterior de registros encontrados

En la pantalla de la Figura 5.28, al presionar el icono  se elimina de la base de datos el registro seleccionado.

En la pantalla de la Figura 5.28 al presionar el icono  se presenta la pantalla de la Figura 5.29. Esta pantalla corresponde al envío de correo electrónico y está sujeta al servicio de Mail administrado por el servidor Web

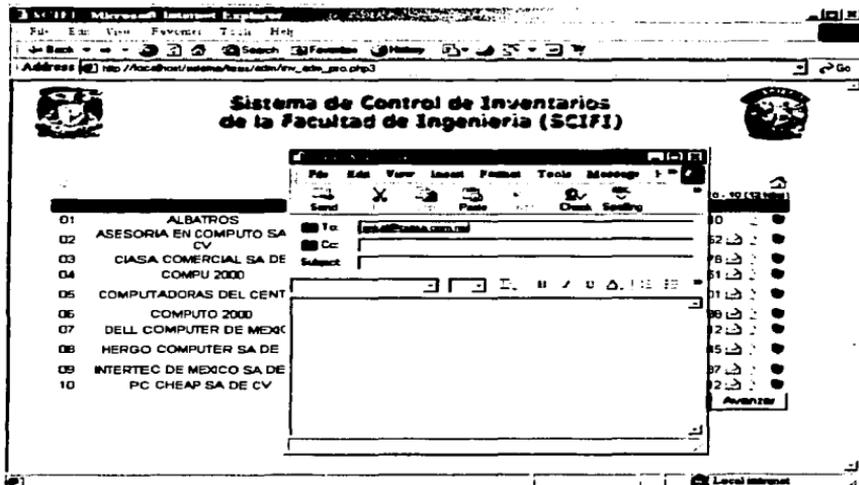


Figura 5.29 Pantalla de Envío de Correo Electrónico al Proveedor

TESIS CON  
FALLA DE ORIGEN

En la pantalla de la Figura 5.28, al presionar el icono  se presenta la pantalla de la Figura 5.30

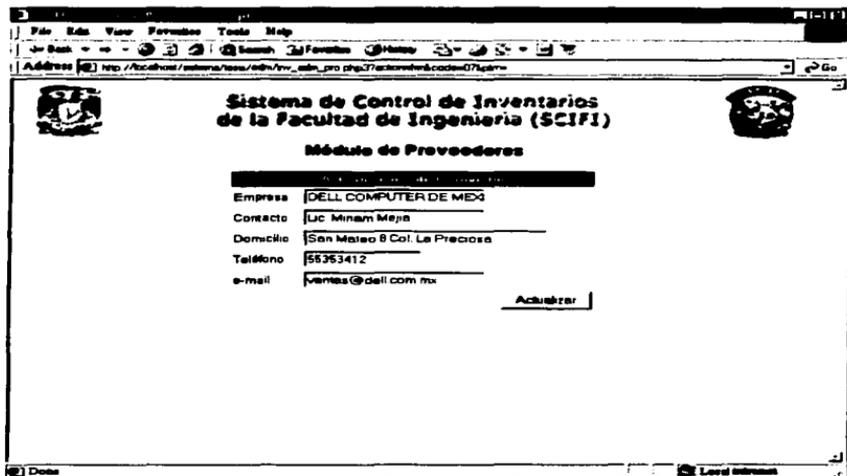


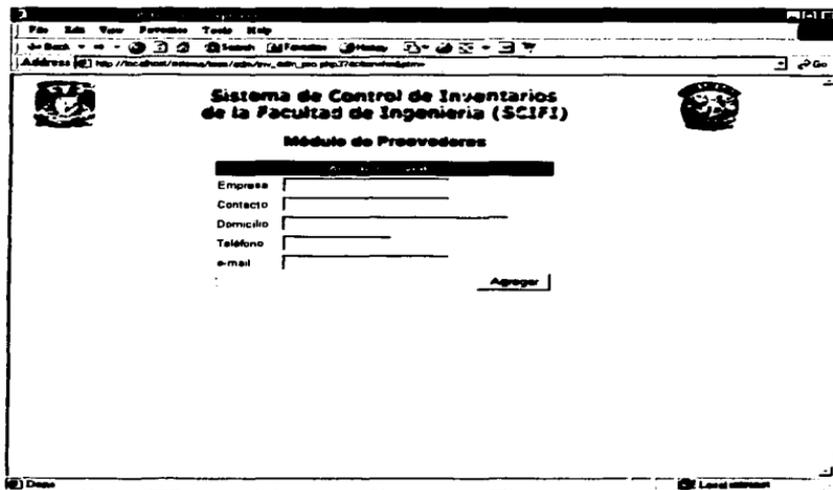
Figura 5.30 Pantalla de Actualización de Proveedor

Actualizar

Actualiza el registro en pantalla en la base de datos.

TESIS CON  
FALLA DE ORIGEN

En la pantalla de la Figura 5.28, al presionar el icono  se presenta una pantalla similar a la ilustrada en la Figura 5.31



The screenshot shows a web browser window with the following content:

- Browser title: Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)
- Address bar: http://localhost:8080/.../scifi/.../scifi\_jsp.php?funcion=altaProveedor
- Page title: Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)
- Page content: Módulo de Proveedores
- Form fields: Empresa, Contacto, Domicilio, Teléfono, e-mail
- Buttons: Agregar

Figura 5.31 Pantalla de Alta de Proveedor

Es necesario ingresar los datos requeridos y presionar el botón  para afectar el registro en la base de datos.

La pantalla del Módulo de Inventario se ilustra en la Figura 5.32

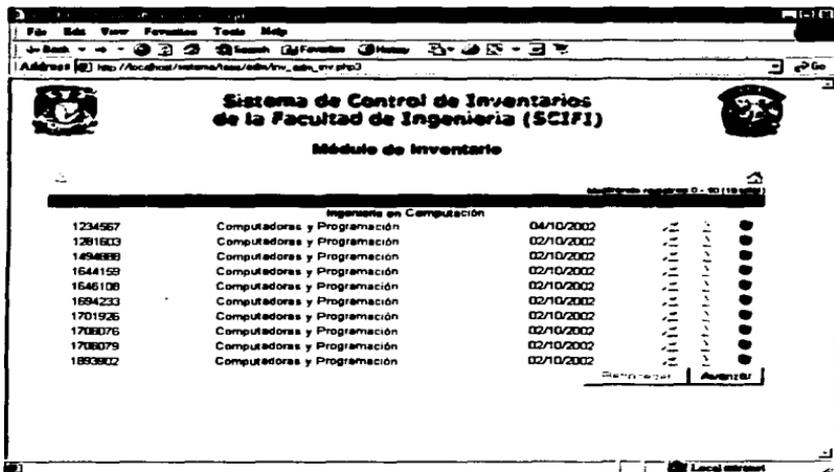


Figura 5.32 Pantalla de Módulo de Inventario

**Avanzar**

Avanza al siguiente grupo de 10 registros encontrados

**Retroceder**

Retrocede al grupo anterior de 10 registros encontrados

En la pantalla de la Figura 5.32 al presionar el icono  se elimina de la base de datos el registro seleccionado

En la pantalla de la Figura 5.32, al presionar el icono  se presenta la pantalla de la Figura 5.33

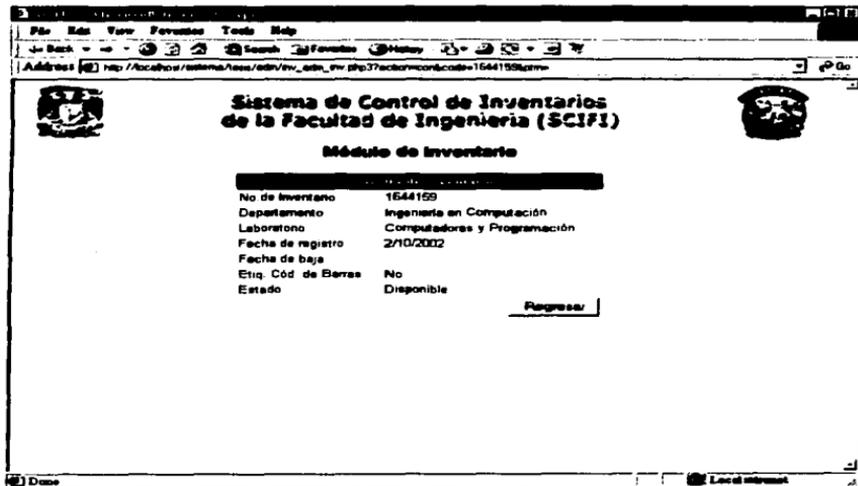
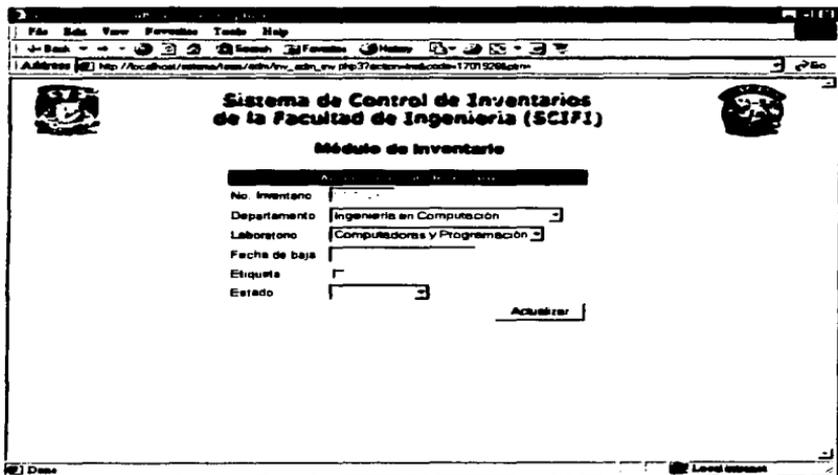


Figura 5.33 Pantalla de Módulo de Inventario

[Regresar](#)

Regresa a la pantalla anterior.

En la pantalla de la Figura 5.32, al presionar el icono  se presenta la pantalla de la Figura 5.34



**Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)**

**Módulo de inventario**

No. inventario:

Departamento:

Laboratorio:

Fecha de baja:

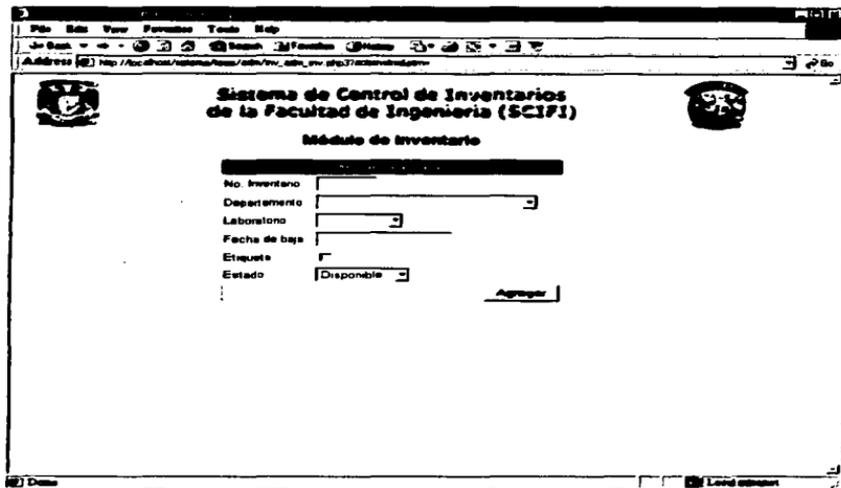
Etiqueta:

Estado:

Figura 5.34 Pantalla de Actualización de Inventario

Actualiza el registro en pantalla en la base de datos.

En la pantalla de la Figura 5.32, al presionar el icono  se presenta una pantalla similar a la ilustrada en la Figura 5.35



The screenshot shows a web browser window with the following content:

- Browser title: Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)
- Page title: Módulo de Inventario
- Form fields:
  - No. inventario:
  - Departamento:
  - Laboratorio:
  - Fecha de baja:
  - Etiqueta:
  - Estado:  Disponible
- Buttons:

Figura 5.35 Pantalla de Alta de Inventario

Es necesario ingresar los datos requeridos y presionar el botón  para afectar el registro en la base de datos.

TESIS CON  
FALLA DE ORIGEN

La pantalla del Módulo de Herramientas se ilustra en la Figura 5.36

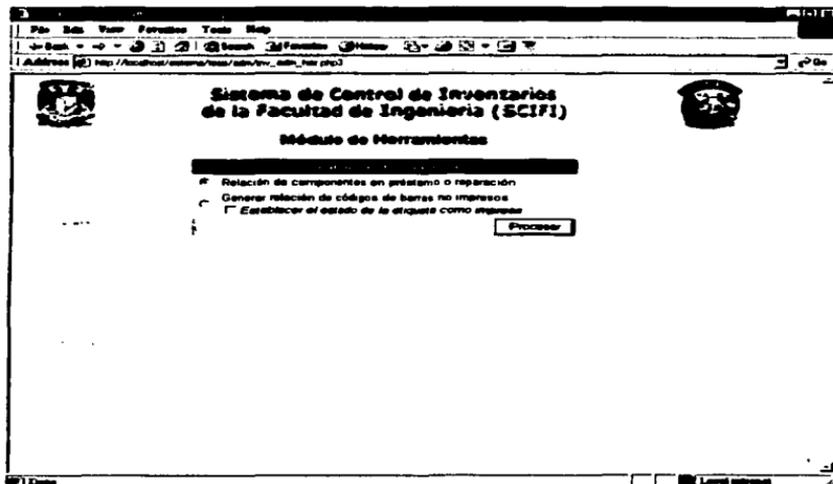


Figura 5.36 Pantalla de Módulo de Herramientas

**Procesar**

Ejecuta la acción que se seleccionó en la pantalla.

TESIS CON  
FALLA DE ORIGEN

La pantalla del Módulo de Solicitudes se ilustra en la Figura 5.37

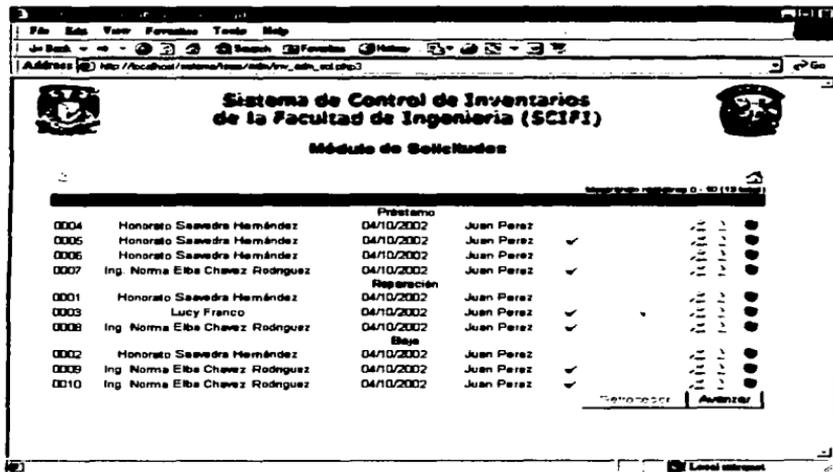


Figura 5.37 Pantalla de Módulo de Solicitudes

**Avanzar**

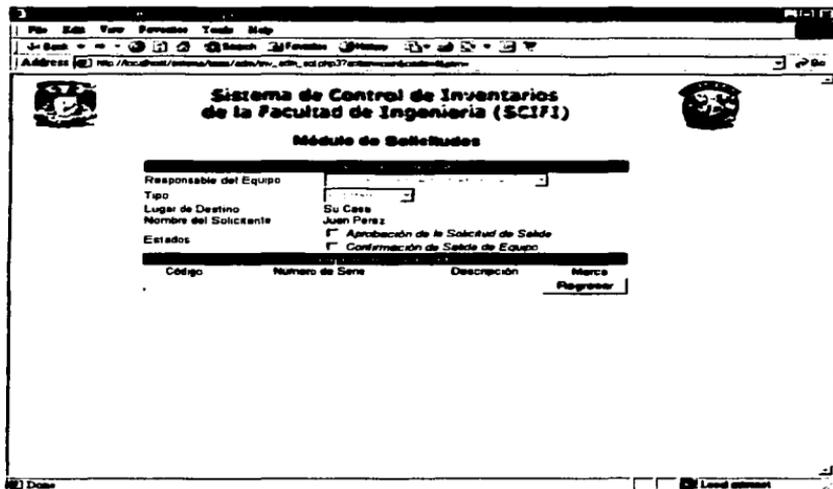
Avanza al siguiente grupo de 10 registros encontrados

**Retroceder**

Retrocede al grupo anterior de 10 registros encontrados

En la pantalla de la Figura 5.37 al presionar el icono ● se elimina de la base de datos el registro seleccionado

En la pantalla de la Figura 5.37, al presionar el icono  se presenta la pantalla de la Figura 5.38



**Sistema de Control de Inventarios de la Facultad de Ingeniería (SCI71)**  
Módulo de Solicitudes

Responsable del Equipo:

Tipo:

Lugar de Destino: Su Casa

Nombre del Solicitante: Juan Pérez

Estados:  Aprobación de la Solicitud de Saldo  
 Confirmación de Saldo de Equipo

Código	Número de Serie	Descripción	Marca

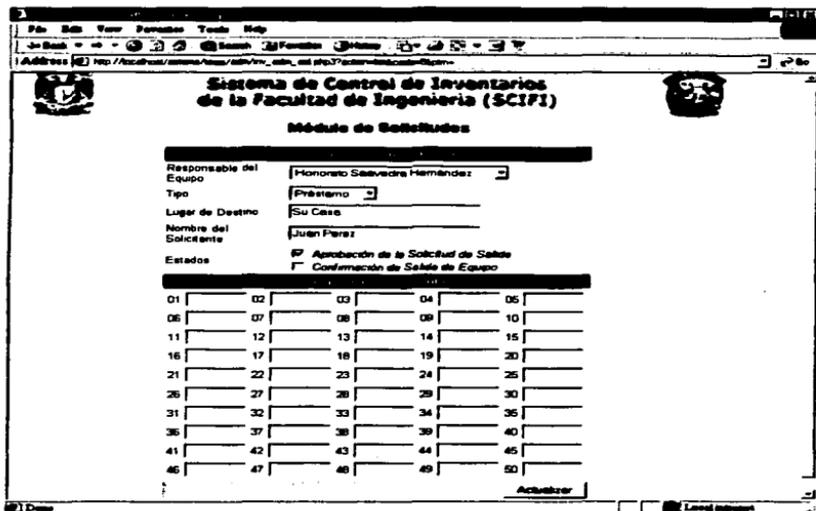
[Regresar](#)

Figura 5.38 Pantalla Consulta de Solicitudes

[Regresar](#)

Regresa a la pantalla anterior.

En la pantalla de la Figura 5.37, al presionar el icono  se presenta la pantalla de la Figura 5.39



**Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)**

**Módulo de Solicitudes**

Responsable del Equipo: [Honorable Sabvedra Hernández]

Tipo: [Préstamo]

Lugar de Destino: [Su Casa]

Nombre del Solicitante: [Juan Pérez]

Estados:  Aprobación de la Solicitud de Salida  
 Confirmación de Salida de Equipo

01	02	03	04	05
06	07	08	09	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50

Actualizar

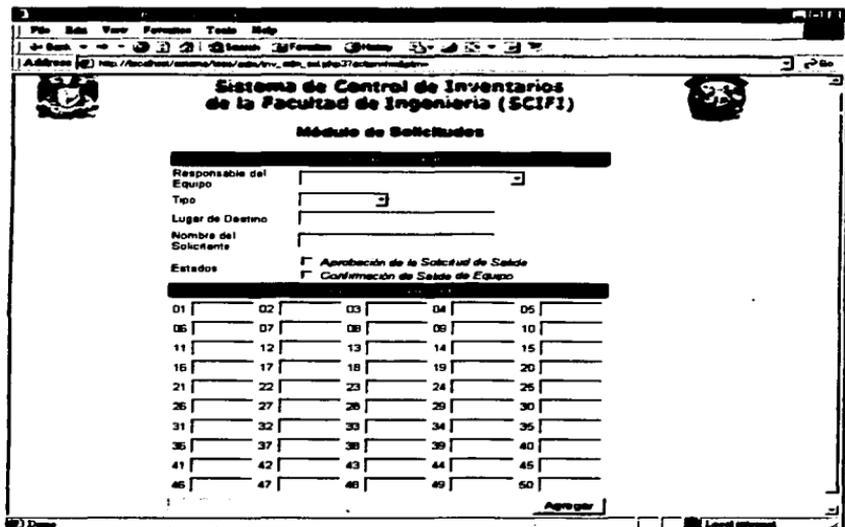
Figura 5.39 Pantalla Actualización de Solicitudes

Actualizar

Actualiza el registro en pantalla en la base de datos.

TESIS CON  
FALLA DE ORIGEN

En la pantalla de la Figura 5.37, al presionar el icono  se presenta una pantalla similar a la ilustrada en la Figura 5.40



**Sistema de Control de Inventarios de la Facultad de Ingeniería (SCIFI)**

**Módulo de Solicitudes**

Responsable del Equipo

Tipo

Lugar de Destino

Nombre del Solicitante

Estados  Aprobación de la Solicitud de Salida   
 Confirmación de Salida de Equipo

01	02	03	04	05
06	07	08	09	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50

Figura 5.40 Pantalla de Alta de Solicitud

Es necesario ingresar los datos requeridos y presionar el botón  para afectar el registro en la base de datos.

La primer pantalla que se obtiene es la del **Módulo de Devoluciones** y se ilustra en la Figura 5.41

**Sistema de Control de Inventarios  
de la Facultad de Ingeniería (SCIFI)**

**Módulo de Devoluciones**

01	02	03	04	05
06	07	08	09	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50

Actualizar

Figura 5.41 Pantalla del Módulo de Devoluciones

Actualizar

Actualiza el registro en pantalla en la base de datos.

TESIS CON  
FALLA DE ORIGEN

## **Conclusiones**

Al igual que en la mayoría de los casos, la elección del tema de tesis a desarrollar para presentar el Examen Profesional no fue fácil, pues todos los temas que se presentaron como propuesta sugerían aspectos interesantes de investigación y elaboración; aún así sobre todas las ideas prevaleció aquella que consistía en elaborar un sistema que controlara la entrada / salida de equipo de cómputo de algunos de los Laboratorios de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, ya que presentó características que lo hacían resaltar significativamente ante las otras propuestas planteadas, de las cuales la más importante fue sin duda la tecnología a utilizar, pues la idea de desarrollarlo para un entorno web, una solución que está presente en las nuevas aplicaciones, supuso un reto y la convicción de aprender muchas cosas útiles y, sobre todo, en auge en el ambiente de sistemas de bases de datos de hoy en día.

Otro de los aspectos importantes es el aprovechamiento de la infraestructura de red con la que se cuenta en las instalaciones de los laboratorios de la Facultad de Ingeniería, lo que permite la instalación de un servidor Web y de un manejador de base de datos para el manejo del sistema, así como la programación PHP que permite el empleo de un navegador para desplegar la información del lado del cliente.

Algo que llamó nuestra atención fue la gran importancia que tiene el aplicar una metodología de desarrollo para poder concluir satisfactoriamente un proyecto. En el presente trabajo utilizamos la metodología de Yourdón, publicada por primera vez hace ya más de dos décadas, pero que está tan bien planeada que sus principios son utilizados como fundamentos de muchas metodologías modernas.

Al concluir el desarrollo del sistema, motivo del presente trabajo de tesis, podemos decir que la solución que se esperaba encontrar para automatizar el control de entrada y salida de equipo de cómputo de los Laboratorios de la Facultad de Ingeniería fue satisfecha en sus aspectos principales: llevar un control del equipo que ha salido de las instalaciones, cuándo regresará, quién es responsable del mismo en el lugar a donde va, qué Departamento y Laboratorio es responsable del equipo, descripción detallada del mismo, entre otros aspectos relacionados, entre los que podemos mencionar desde la generación de los códigos de barras con número de inventario UNAM para pegarlos en el equipo, pasando por la facilidad que brinda el uso de la pistola lectora de dichos códigos, hasta un registro de los proveedores principales de equipo de cómputo de la Facultad de Ingeniería.

Además de satisfacer la necesidad presentada, podemos incluir que obtuvimos un mayor conocimiento y experiencia en cuanto al desarrollo y puesta en marcha de un sistema de base de datos se refiere. El empleo de servidores web y bases de datos, así como la programación PHP utilizados para la elaboración del sistema, todos de distribución libre, abre numerosas y prometedoras propuestas para la realización de sistemas vía web con la misma calidad e incluso seguridad de la que prometen los programas comerciales, lo cual es un gran aliciente para continuar con la investigación y desarrollo sin necesidad de contar con recursos económicos elevados.

Finalmente, la satisfacción de lograr que nuestra tesis se haya convertido en un sistema útil a la Facultad de Ingeniería, y retribuir en algo a lo mucho que nos brindó y enseñó nuestra Universidad.

**Bibliografía**

Michael Afergan, et. al.

Web Programming Desktop Reference

Que Corporation, Estados Unidos de América, 1996.

1105 pp.

Greenspan Jay y Brad Bulger

MySQL/PHP Database Applications

M & T Books, Estados Unidos de América, 2001.

586 pp.

Yourdón Edward

Análisis estructurado moderno

Pretince Hall Hispanoamérica, México, 1993.

735 pp

Date, C.J.

The Database Relational Model

Addison Wesley Publishing Company, Estados Unidos de América, 2000.

152 pp

Barker, Richard

Case Methodology

Addison Wesley Publishing Company, Estados Unidos de América, 2000.

321 pp

Burch, G Jhon y Gary Grandnitsky  
Diseño de sistemas de información. Teoría y práctica  
Megabyte, México, 1992.  
785 pp

García Pérez, Fernando y Félix Chamorro Atance  
Informática de gestión y sistemas de información  
Mc Graw Hill, España, 2000.  
239 pp

Stair, Ralph M.  
Principles of information System  
2da. ed.  
Boyd & Fraster Publisher Company, Estados Unidos de América, 1996.  
659 pp

Ligas consultadas en Internet:

<http://espanol.geocities.com/michelsoftpe/bdatos.htm>  
<http://www.htmlpoint.com/css/>  
<http://www.programacion.com/html/cursos/>  
<http://www.tutorial-web.com/asp/database/>  
<http://www.php.net>  
<http://www.mysql.com>  
<http://www.codigodebarras.com>