



**UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO**

39

**FACULTAD DE INGENIERIA**

**"DESARROLLO DE UN SISTEMA DE  
CONTROL DE PATRIMONIO".**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:

**INGENIERO EN COMPUTACION**

P R E S E N T A:

**ALEJANDRO GERMENOS RAMIREZ**

DIRECTOR DE TESIS: ING. ALBERTO TEMPLOS CARBAJAL



2824754



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.

## RECONOCIMIENTOS

Nadie nace sabiendo, y sólo la sabiduría puede dirigirnos para ser hombres de bien. Es la educación que nos brindan nuestros padres lo que nos hace hombres, y somos nosotros y nuestras decisiones las que nos perfilan para tomar el camino correcto.

Quiero agradecer muy especialmente a quienes me enseñaron que el mundo tiene de todo un poco; que sólo es cuestión de decidirse a tomar lo que uno quiere. No sólo por todo lo que me inculcaron desde pequeño, sino al increíble ejemplo que siempre tuve de ellos.

Con todo mi cariño: **A Mis Padres**

También es justo agradecer a quienes me apoyaron y me acompañaron a lo largo de mi carrera profesional. No es fácil encontrar amigos; pero cuando se encuentran, uno descubre lo invaluable que es una verdadera amistad.

Por que estuvieron siempre conmigo, en las buenas y sobre todo en las malas, quiero agradecer a tres grandes amigos que me apoyaron y me demostraron cuán cierta es la frase "querer es poder".

Hugo Lara  
Rafael Pavón  
Francisco Romero

Gracias

A mi director de tesis, el Ing. Alberto Templos, a mis compañeros de clase y a todas aquellas personas que de alguna manera me enseñaron algo.



TABLA DE CONTENIDO

<b>BREVE INTRODUCCIÓN A LA PROBLEMÁTICA DE PATRIMONIO .....</b>	<b>4</b>
INTRODUCCIÓN .....	4
PLANTEAMIENTO DEL PROBLEMA .....	5
NECESIDADES DE PATRIMONIO .....	6
<b>ARQUITECTURA DEL SISTEMA .....</b>	<b>8</b>
EN QUÉ CONSISTE UNA ARQUITECTURA .....	8
USO DE UML PARA EL DESARROLLO DE LA ARQUITECTURA .....	9
CASOS DE USO .....	10
ESTÁNDARES DE DESARROLLO .....	13
DISEÑO DE LA BASE DE DATOS .....	14
DISEÑO LA INTERFAZ GRÁFICA DE USUARIO.....	17
<b>MODELO DE LA BASE DE DATOS .....</b>	<b>20</b>
REQUERIMIENTOS DE INFORMACIÓN DE PATRIMONIO.....	20
NORMALIZACIÓN DE LA BASE DE DATOS .....	21
<b>DISEÑO LÓGICO DEL SISTEMA .....</b>	<b>25</b>
FINALIDAD DEL DISEÑO DE PATRIMONIO .....	25
DIAGRAMAS Y MÓDULOS DEL SISTEMA.....	26
<b>DISEÑO FÍSICO DEL SISTEMA.....</b>	<b>28</b>
ARQUITECTURA DEL DISEÑO FÍSICO .....	28
ESTÁNDARES DEL DISEÑO .....	29
PANTALLAS DEL SISTEMA DE PATRIMONIO .....	30
<b>PRUEBAS AL SISTEMA .....</b>	<b>40</b>
CALIDAD DE SOFTWARE .....	40
INSPECCIONES PERIÓDICAS .....	40
PROCEDIMIENTOS DE UNA INSPECCIÓN .....	41
PRUEBAS Y CALIDAD EN EL SISTEMA DE PATRIMONIO.....	42
<b>MANUAL DE USUARIO DEL SISTEMA DE PATRIMONIO.....</b>	<b>43</b>



Universidad Nacional Autónoma de México  
Facultad de Ingeniería

<b>DOCUMENTACIÓN DE LA BASE DE DATOS .....</b>	<b>52</b>
<b>CONCLUSIONES.....</b>	<b>53</b>
<b>BIBLIOGRAFÍA.....</b>	<b>54</b>



*Capítulo 1*

BREVE INTRODUCCIÓN A LA PROBLEMÁTICA DE PATRIMONIO

Introducción

La dependencia de Patrimonio, del H. Ayuntamiento Constitucional de Zapopan (HAZ), tiene la función de llevar control de los activos pertenecientes a toda persona que se considere resguardante de un bien. Esto quiere decir que cuando una persona tiene a su cargo un activo perteneciente al HAZ, se le considera resguardante de un activo.

Las tareas a realizar incluyen: dar de alta activos, llevar un control de los resguardantes y las dependencias a las que pertenecen, tener catalogados los activos, presentar reportes de los activos de cada resguardante y dependencia, etc. Dichas tareas son facilitadas por medio de un sistema de control de patrimonio. Sin embargo, se presentó la problemática del Y2K<sup>1</sup>. Tras un breve simulacro el sistema demostró no ser capaz de pasar dicho problema, por lo que se procedió al análisis de desarrollo de un sistema nuevo.

Lo que se pretende presentar a continuación, no es la codificación del sistema, sino cómo debe ser desarrollado el sistema de Patrimonio. La arquitectura, creo, es

---

<sup>1</sup> Year 2000: Consiste en la confusión de fechas ocasionado por el manejo de dos campos en lugar de cuatro para definir el año. Muchos sistemas pueden confundir el 00 con el año 1800, ocasionando problemas al manejar cálculos por tiempo.



la pieza fundamental para el desarrollo de todo sistema. Así como las personas requieren bases sólidas para su desarrollo, también un sistema en cómputo conviene de tener la estructura adecuada sobre la cual desarrollarse.

### Planteamiento del Problema

La dirección de Patrimonio del Municipio de Zapopan presenta los siguientes problemas con su sistema de control de patrimonio. Esta lista representa los problemas estructurales básicos del sistema:

- **El año 2000:** El sistema anterior de Patrimonio no supera el problema del año 2000. Este problema en sí es crítico para el buen funcionamiento de la dependencia, pues necesitan emitir reportes mensuales de los resguardos asignados al personal de cada una de las dependencias.
- **Integridad de la información:** Comenzaron a presentarse activos (bienes) repetidos en ciertas dependencias, así como ciertos errores en la información. Después de llevar a cabo un análisis del sistema anterior, se concluyó que la información que yacía en la base de datos del sistema había sido mal diseñada, teniendo carencias de integridad referencial. Esto se reflejó en datos faltantes, así como entidades “huérfanas”<sup>2</sup>
- **Carencia de red:** El sistema desarrollado anteriormente para la dependencia, no tenía un entorno de red; sin embargo, más de una

---

<sup>2</sup> RIORDAN M. Rebecca. “Designing Relational Database Systems”. “Tuples that contain foreign keys that don’t have a corresponding candidate key in the primary relation are called orphan entities”.



## Universidad Nacional Autónoma de México Facultad de Ingeniería

máquina trabajaba con el sistema en una base de datos independiente. Esto quiere decir que terminado el transcurso de cada día, se tenían que seleccionar los registros agregados y/o modificados, y migrarlos a las otras computadoras, haciendo la información vulnerable.

### Necesidades de Patrimonio

Las necesidades más destacadas del nuevo sistema de control de patrimonio son las siguientes:

- **Pasar el año 2000:** La necesidad primordial, y por la cual se solicitó el nuevo sistema de control de patrimonio.
- **Funcionar en un entorno de red:** Se propuso el desarrollo de un sistema que funcionara en modalidad cliente/servidor. Esto se hizo para evitar tener varias bases de datos independientes y darle una mayor integridad a la información. Además de esto, se prevé en una segunda etapa, el desarrollo de una base de datos centralizada por parte del Ayuntamiento de Zapopan.
- **Operar por medio de una Base de Datos Relacional:** La manera idónea de tener la información de manera íntegra es por medio de una base de datos relacional<sup>3</sup>. El modelo propuesto por Codd<sup>4</sup>, así como un

---

<sup>3</sup> Ibid. "The physical implementation of a relational model (the data model) is the best way to keep information integrity without many trade-offs".

<sup>4</sup> Dr. E. F. Codd. 1967, propone una serie de reglas para definir la implementación física de una base de datos basada en un modelo relacional.





Universidad Nacional Autónoma de México  
Facultad de Ingeniería

desarrollo en la tercera forma normal es propuesto para el desarrollo del sistema de Patrimonio.

Con base en estos planteamientos, se procedió a desarrollarse la arquitectura del sistema necesario para el control de patrimonio.



*Capítulo 2*

ARQUITECTURA DEL SISTEMA

En qué Consiste una Arquitectura

La arquitectura es una de las artes más viejas y más antiguas disciplinas. La historia de la arquitectura en la construcción puede ser trazada desde hace miles de años a las pirámides de Egipto, la Gran Muralla China, y los templos de México. La historia de la arquitectura del software comenzó poco tiempo después de que ENIAC comenzó a procesar bits en los cuarentas. Aunque muchas especializaciones de arquitectura (como la arquitectura de construcción, la arquitectura naval y la arquitectura de aeronaves) son disciplinas maduras, la arquitectura de software está en su infancia. Cabe notar que la arquitectura de software es mucho menos madura que su hermana, la arquitectura de hardware, que tiene a la Ley de Moore, para predecir mejoras regulares en el desempeño

La plasticidad del software ha permitido que se construya y renueven los sistemas de software en meses, en lugar de años; también nos permite ir haciendo una pelota con parches tela, mostrando arquitecturas que muestran suburbios en lugar de edificios. Como consecuencia, se pueden señalar muy pocas arquitecturas de referencia y sólo un pequeño número de patrones de arquitectura ameritan ser emulados. Actualmente la arquitectura de software es mucho más implícita que explícita, y rara vez es planeada.



Desafortunadamente, la casi invisibilidad del software ha facilitado la perpetración de este tipo de práctica irresponsable de arquitectura. Son sorprendentemente pocos los profesionales de software y casi ningún usuario, que pueden asegurar la calidad o robustez de una arquitectura de software.

Como la mayoría de las arquitecturas de software son implícitas (por ejemplo “ver el código fuente para los detalles”), y pocos son descritos por documentos detallados (por ejemplo, modelos de software y especificaciones relacionadas), muchas personas evalúan a los sistemas de software superficialmente con la base de que sólo se ve una parte, y es la interfaz de usuario. Aunque los usuarios típicamente saben que algo está mal cuando seleccionan un elemento del menú y el sistema presenta un error, no pueden identificar si el sistema es de fundación (como el sistema operativo), en el medio (red o middleware), o en la aplicación<sup>5</sup>.

El desarrollo de una buena arquitectura para el sistema de Patrimonio se considera entonces esencial, pues es un sistema desarrollado por un externo (es decir, que no pertenece al HAZ), por lo que su mantenimiento será muy forzado si carece de la documentación adecuada.

### Uso de UML para el Desarrollo de la Arquitectura

La arquitectura del sistema de control de patrimonio proporciona toda la información necesaria para el desarrollo del sistema. Contiene una descripción de cada uno de los elementos del sistema, así como de las acciones que relacionan a

---

<sup>5</sup> BOOCH, JACOBSON, RUBRAUGH, *Applied Software Architecture*. Ed. Addison Wesley.



éstos entre sí. También proporciona información pertinente al desarrollo propio del sistema.

Como podrá observarse, la arquitectura es la base del sistema. Aquí se deberá presentar toda la información requerida para desarrollar el sistema, además de facilitar el mantenimiento (de cualquier tipo que éste sea<sup>6</sup>) del mismo.

### Casos de Uso

Se utilizó para el desarrollo de la arquitectura, el estándar UML (Unified Modeling Language). La utilización básica de UML fue para el desarrollo de los procesos de negocio con casos de uso (Use Cases).

Los casos de uso se definen como los procesos que requieren de la interacción entre el usuario y la computadora. Por cada evento discreto, se le asigna un nombre y se da una descripción textual (de preferencia corta)<sup>7</sup>.

Los siguientes son los casos de uso asignados en Patrimonio:

- Los activos tienen mantenimiento
- Los resguardantes tienen mantenimiento
- Las dependencias tienen mantenimiento

---

<sup>6</sup> NOBLE Camargo, Sergio. Notas de Clase Ingeniería de Programación. "El mantenimiento de los sistemas puede ser de tres tipos: correctivo, preventivo o adaptativo.

<sup>7</sup> Fowler, Martin. "UML Distilled". Capítulo 3: Applying the Standard Object Modeling Language.



Universidad Nacional Autónoma de México  
Facultad de Ingeniería

- Los activos se reetiquetan
- Los activos son reasignados
- Los resguardantes son reasignados
- Los activos son desincorporados
- Las dependencias tienen inventario

Salen a relucir como actores en la lista: las dependencias, los resguardantes y los activos.

Salen a relucir como verbos(acciones) en la lista: el mantenimiento, la reetiquetación, la reasignación, la desincorporación y los inventarios. Haciendo un breve resumen de lo que se hace en cada acción, se obtiene lo siguiente:

1. Mantenimiento: Consiste en dar de alta, dar de baja, modificar. También se vislumbró la posibilidad de presentar un reporte de cada uno de los actores que presenten mantenimiento.
2. Reetiquetación: Consiste en cambiar la etiqueta, que es el medio de identificación para cada uno de los activos de Patrimonio. Esto es a causa de la pérdida de las mismas (etiquetas) por el uso que se le da a los activos.
3. Reasignación: Consiste en cambiar un activo de resguardante y/o dependencia.

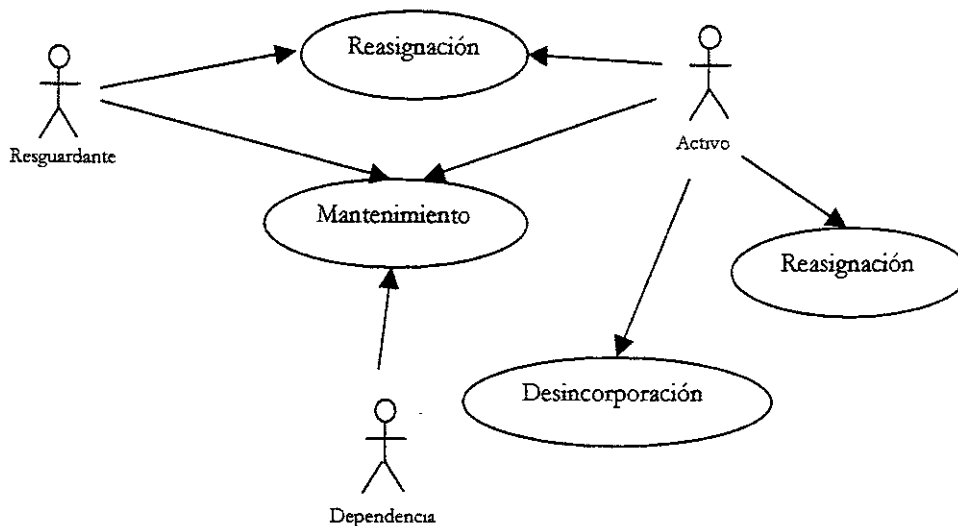


Universidad Nacional Autónoma de México  
Facultad de Ingeniería

4. Desincorporación: Consiste en dar de baja al activo. Cabe hacerse notar, que el activo debe permanecer almacenado, pues se tiene que presentar al Cabildo un reporte de los bienes desincorporados al final del año.



El diagrama de casos de uso es el siguiente:



### Estándares de Desarrollo

El proceso de estandarización del desarrollo de la aplicación, así como el modelo para el agregado de módulos posteriores, quedó establecido de la siguiente manera<sup>8</sup>:

La programación se procurará hacer de la manera más estandarizada posible. Es importante que el código tenga consistencia, para que al darle mantenimiento al código no se tenga que “adaptar” a la manera de programar de cada individuo. Esto se logrará documentando debidamente la codificación de cada

<sup>8</sup> Germenos Ramírez, Alejandro: Metodología de Desarrollo de Aplicaciones. Capítulo 2: Programación. Manual propiedad de Concepto Consultoría en Sistemas.



## Universidad Nacional Autónoma de México Facultad de Ingeniería

procedimiento o evento, así como manejando un estándar en el nombramiento de las formas y los controles, así como los módulos de cada sistema.

La documentación del código deberá ser abundante (mínimo un comentario explicativo por cada 10 líneas de código). Esto se debe a que el mantenimiento de Visual Basic se torna muy conflictivo cuando se intenta definir la función global que tiene la realización de determinado evento. Por esto se procurará documentar de la manera siguiente:

- Al inicio de cada evento, se indicará lo que realiza el botón y la forma (con su código) en la que se desarrolla la acción.
- En caso de que se maneje cierta complejidad en un algoritmo, se documentará lo que realiza el algoritmo, así como las variables locales, de entrada y salida que maneje.

Se manejará la declaración de todas las variables que se vayan a usar a lo largo de toda codificación. Para preveer esto, se declarará Option Explicit en todos los desarrollos.

### Diseño de la Base de Datos

El diseño de la base de datos se basa en lo que el sistema debe ser capaz de hacer y presentar al usuario. Estas son las funciones básicas del sistema de control de patrimonio:

- Control de Bienes (altas, bajas y modificaciones)
- Movimiento de Bienes





Universidad Nacional Autónoma de México  
Facultad de Ingeniería

- Manejo del Inventario de Patrimonio
- Manejo del Historial del Inventario

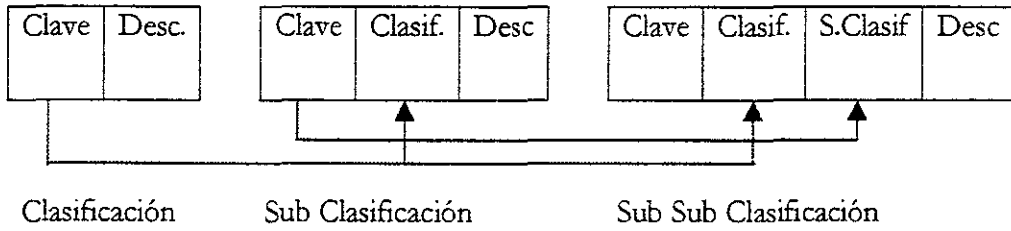
Así mismo deberá ser capaz de presentar reportes sobre los siguientes movimientos:

- Reporte de Historial de bienes
- Reporte de Rotación de Patrimonio
- Reporte de Bienes Comprados
- Reporte de Resguardantes por Dependencia
- Catálogo de Resguardantes
- Catálogo de Dependencias
- Catálogo de Tipos de Adjudicación
- Catálogo de Tipos de Desincorporación
- Catálogo de Clasificaciones

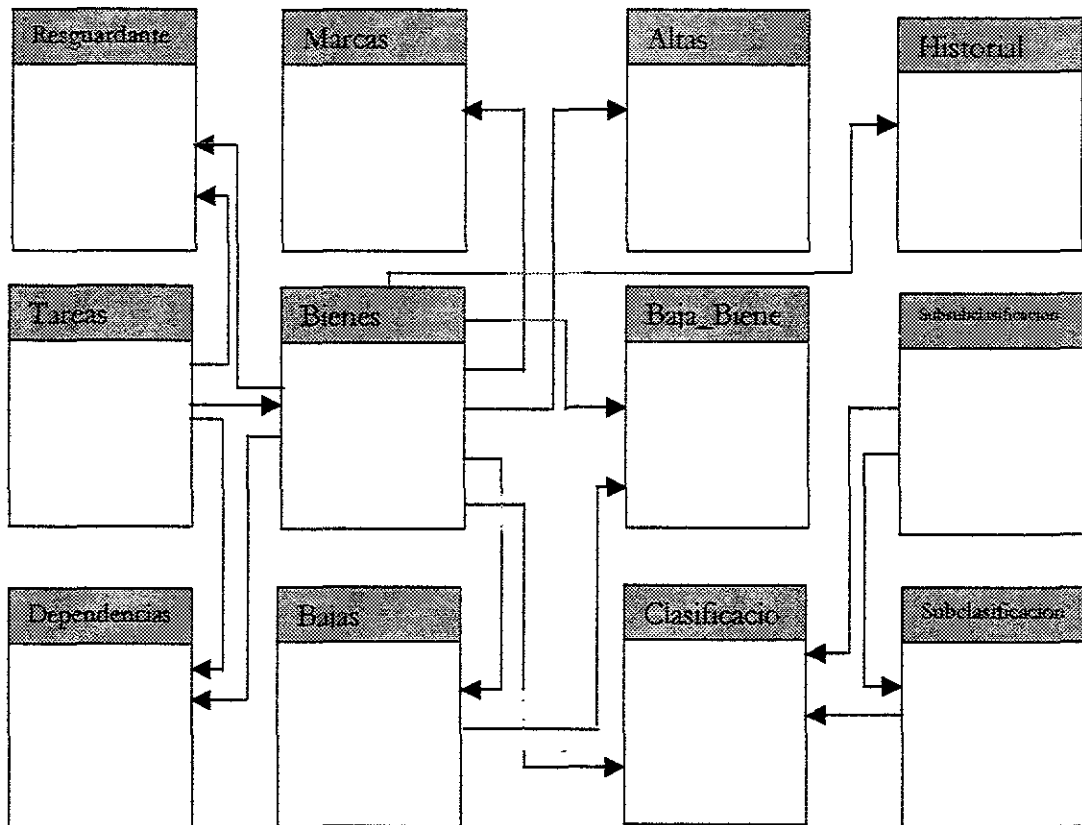
Con base en los actores presentados en los casos de uso, se manejan como tablas principales Bienes, Resguardantes y Dependencias. Así mismo, se agregan tablas de información para catálogos, como lo son Marcas, Clasificaciones, SubClasificaciones, etc.

La base de datos contiene doce tablas, con relaciones uno a uno (excepto en historial, que tiene una relación con bienes de muchos a uno).

La tabla de subclasificaciones, así como la de subsubclasificaciones presentan más de una llave primaria. La tabla de subclasificaciones se identifica por: un consecutivo de la subclasificación y el valor de la llave primaria de su clasificación. Así mismo, subsubclasificación tiene tres campos que conforman a la llave primaria: el consecutivo, el valor de la llave primaria de la clasificación y el valor del consecutivo de la subclasificación. Esto se puede ver más claro en la siguiente tabla:



El siguiente es el diseño de la base de datos, sin mostrar la información que deberá contener cada tabla (esto se presentará en el siguiente capítulo):





## Diseño la Interfaz Gráfica de Usuario

La Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés) es un tipo de formato de despliegue que permite a un usuario elegir comandos, ver listas de opciones y manejar menús de manera gráfica.

El sistema de Patrimonio pretende dividir el sistema en dos partes: un back-end, que contendrá la base de datos, reglas de negocio y algoritmos del sistema, que permitan el buen funcionamiento de Patrimonio, y un front-end, que constará de un GUI que permita al usuario desplazarse por las pantallas de una manera lógica y simple. La intención que se tiene con esta división tan clara es preveer, la posible preferencia que puede presentar un usuario con determinados componentes gráficos. Un ejemplo muy simple de esto se puede presentar con el siguiente caso que sucedió con el sistema en cuestión:

Se le presentó a la dirección de Patrimonio un prototipo con el front-end del sistema, para que decidiera los cambios posibles que pudiese tener. Se puso (por petición del usuario) una lista con los activos para ser manipulados, al seleccionar uno de los activos se presentaba otra pantalla, con toda la información perteneciente a éste.

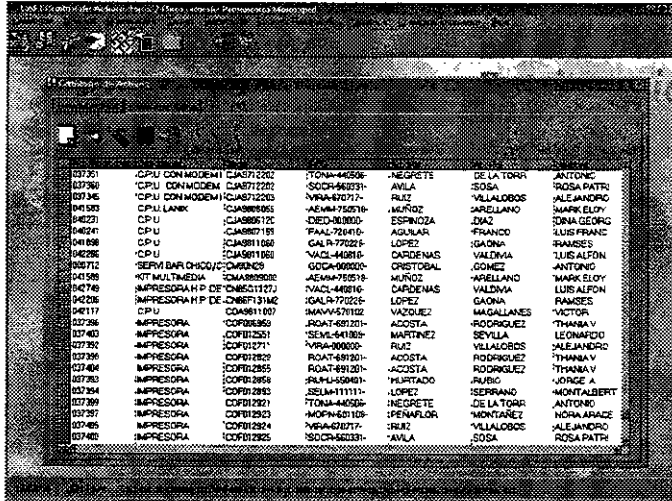
Sin embargo, cuando se presentó la versión beta del sistema, varios de los usuarios se quejaron, pues decían que ellos preferían manejar la selección de activos por medio de una matriz. La lista les pareció poco práctica, pues decían que ellos querían ver más información de los activos para saber cuál era el indicado a manejar.



# Universidad Nacional Autónoma de México

## Facultad de Ingeniería

La ventaja que se tuvo al manejar el GUI separado del back-end es aparente: el sistema no tuvo cambio alguno en su operación, pues la manipulación de la información no se hacía por medio de la lista. Dicho de otra manera, la lista no era la referencia de la información.



Diseño Final de la pantalla de activos.

Para el usuario el sistema es el front-end (el GUI). Es decir que si uno deja el mismo back-end del sistema, y rediseña un nuevo GUI, el usuario pensará que es un sistema completamente nuevo. Considero que una buena arquitectura se basa en el back-end, sin dejar de lado el GUI, pues es lo que el usuario tomará como el sistema en cuestión<sup>9</sup>.








No se puede dejar de lado la apariencia del sistema, pues es lo que el usuario manipulará a lo largo de la vida del software (aunque, si se maneja adecuadamente, es posible que se de mantenimiento al GUI si el back-end es lo bastante robusto y la arquitectura es la adecuada). Los iconos deben ser



Universidad Nacional Autónoma de México  
Facultad de Ingeniería

comprensibles para el usuario y deben ser constantes. Esto quiere decir que a lo largo del sistema, un mismo icono signifique una misma acción.

El sistema de control de patrimonio maneja los siguientes iconos, y con sus respectivas interpretaciones:

Icono	Nombre	Acción
	Agregar	Abre un nuevo registro en la tabla seleccionada
	Modificar	Permite modificar un registro seleccionado de la tabla.
	Grabar	Actualiza los cambios en la base de datos.
	Cancelar	Omite los cambios realizados.
	Salir	Cierra la forma (si es la forma principal, cierra el sistema).
	Reporte	Presenta el reporte de la pantalla (si es la forma principal, presenta un menú de reportes).
	Borrar	Permite borrar un registro de la tabla seleccionada.

<sup>9</sup> Osier, Batson and Grobman. "Teach yourself Delphi 3 in 14 Days". Capítulo 7: The Graphical User Interface Design.



### *Capítulo 3*

## MODELO DE LA BASE DE DATOS

### Requerimientos de Información de Patrimonio

La dependencia de Patrimonio presentó determinados requerimientos con respecto a su información. Con base en estos requerimientos se procedió a realizar una base de datos relacional para manipular su información de una manera eficiente.

En términos generales, una base de datos relacional tiene las siguientes características<sup>10</sup>:

- Toda información es representada conceptualmente por un arreglo ordenado de datos, dividido en renglones y columnas, denominado *relación*.
- Todos los valores son escalares. Es decir, dada una posición renglón/columna, existe una relación con un y sólo un valor.
- Todas las operaciones son realizádas en una relación y resultan en una relación; este concepto es conocido como *cerradura*.

---

<sup>10</sup> Designing Relational Database Systems. Op. Cit. pág 15.



## Normalización de la Base de Datos

La normalización es un proceso que consiste en aplicar una serie de reglas (seis) que van restringiendo la base de datos. Cada una de las reglas extiende un poco más la anterior, consolidando un modelo de datos bien definido y con pocas anomalías.

Cabe mencionarse que el hecho de que se normalice una base de datos, no quiere decir que la información esté organizada de manera óptima. Una base de datos puede cumplir con las seis formas normales, pero ser lenta e ineficiente para responder a las preguntas que se le hace.

En el caso del sistema de patrimonio, la base de datos no cumple con todas las formas normales, pues el cliente presentó requerimientos que eran poco eficientes de esta manera. Sin embargo se presentan las formas normales, así como un breve análisis de por qué cumple (o no) la base de datos realizada.

### **Primera Forma Normal**

Se dice que una relación está en su primera forma normal si los dominios de sus atributos definidos son escalares. Esto es un concepto simple, pero puede resultar complicado. Tomemos por ejemplo un campo que contiene una fecha. La fecha se podría considerar escalar, pero también podría decirse que tiene tres atributos (día, mes y año).

Considerando la fecha como un atributo escalar, podemos afirmar que el sistema de patrimonio *cumple con la primera forma normal*.



### Segunda Forma Normal

Se dice que una relación está en su segunda forma normal si, además de estar en su primera forma normal, todos sus atributos son dependientes de la llave primaria completa.

Un ejemplo sencillo de esto sería la siguiente tabla:

* Vendedor	* Producto	Precio	Teléfono
José	Refrescos	4	(013) 122 15 42
José	Carne	20	(013) 122 15 42
Pedro	Carne	21	(013) 122 26 45

En este caso Vendedor y Producto son la llave primaria, y se puede observar que el Teléfono depende de Vendedor, **mas no de Producto**. La tabla presentada no cumple con la segunda forma normal.

En el caso de la base de datos del sistema de control patrimonial, sólo SubClasificación y SubSubClasificación contienen más de un campo como llave primaria, y todos los campos dependen de toda la llave para definirse adecuadamente. Por ende, se puede decir que la base de datos del sistema de control de patrimonio *cumple con la segunda forma normal*.





### **Tercera Forma Normal**

Se dice que una relación cumple con la tercera forma normal si, además de cumplir con la segunda forma normal, todas las relaciones no pertenecientes a un atributo considerado parte de la llave primaria son mutuamente independientes.

En este caso la base de datos desarrollada para Patrimonio muestra ciertas anomalías que *no le permiten considerarla en la tercera forma normal*. Por ejemplo: la tabla de bienes tiene descripción de un activo, marca, modelo y costo. El costo del artículo depende de la marca y el modelo del activo.

El resto de las formas normales (4<sup>a</sup>, 5<sup>a</sup> y Bryce/Codd) no fueron consideradas para el desarrollo de la base de datos, pero ameritan ser mencionadas.

### **Forma de Bryce/Codd**

La forma normal de Boyce/Codd es considerada una variación de la tercera forma normal, y maneja un tipo de relación con múltiples llaves principales. De hecho, se necesitan dar las siguientes condiciones para aplicar la forma de Boyce/Codd:

- La relación debe tener al menos dos llaves primarias



## Universidad Nacional Autónoma de México Facultad de Ingeniería

- Al menos dos de las llaves deben ser compuestas
- Los atributos de las llaves se deben traslapar

La normalización Boyce/Codd funciona similarmente a la tercera forma normal. No debe existir dependencia funcional entre los campos (que en este caso son llaves).

### **Cuarta Forma Normal**

La cuarta forma normal provee de los fundamentos para un principio que es intuitivamente obvio: los grupos independientes que se repitan no se deben combinar en una misma relación.

### **Quinta Forma Normal**

La quinta forma normal trata del caso extremo de las dependencias unidas (*join dependencies*). Una dependencia unida expresa la restricción cíclica “Si Entidad1 está ligada a Entidad2, y Entidad2 está ligada a Entidad3, y Entidad3 está ligada a Entidad1, entonces las tres entidades deberán coexistir en la misma tupla **necesariamente**.”



*Capítulo 4*

DISEÑO LÓGICO DEL SISTEMA

Finalidad del Diseño de Patrimonio

El sistema de control de patrimonio tiene como fin último la seguridad de la información de los activos, así como de los resguardantes y dependencias a los que éstos pertenecen. Sin embargo, es también importante la facilidad del manejo del sistema, así como la velocidad de proceso de éste para el buen funcionamiento de la dependencia en su conjunto.

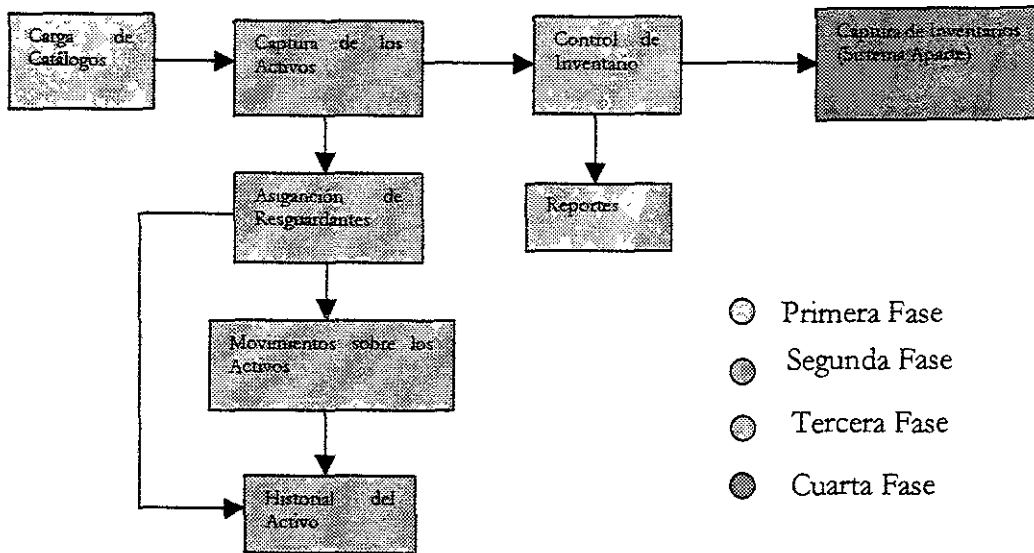
El diseño del sistema se orientó a las tareas, y se agruparon en menús aquellas tareas realizadas en grupo. Por ejemplo: la captura de los catálogos (que se deben generar periódicamente) se agrupan en el menú catálogos, mientras que la asignación de tareas (asignación de resguardante y dependencia a los activos recién adquiridos), movimientos del activo y mantenimiento del activo se mantienen por separado, pues estas tareas se realizan de manera continua.

Cabe mencionarse que la generación de dos reportes en particular (impresión de resguardos por bienes y por dependencia) se mantuvieron separados del conjunto de reportes, debido al uso continuo que tienen.



### Diagramas y Módulos del Sistema

El flujo de trabajo del sistema de patrimonio es el siguiente:



La carga de catálogos consiste en dar de alta toda la información pertinente para poder ser manipulada posteriormente. Esto incluye la captura de dependencias, resguardantes, marcas, tipos de alta, tipos de baja, etc.

La captura de activos se considera un proceso aparte, pues se desarrolla de manera continua, mientras que la carga de los catálogos se debe realizar esporádicamente.

La asignación de resguardantes pertenece a la fase de Movimientos de los activos, y debe manipular las fechas tanto del movimiento, así como de la expedición del documento que pide la transferencia del activo.



## Universidad Nacional Autónoma de México Facultad de Ingeniería

El módulo de inventario y de reportes se consideran de otra fase, pues pertenecen a consultas, mientras que el historial pertenece a la fase de manejo de activos por su manejo implícito con los movimientos de los activos.

Los módulos principales que maneja el sistema de control de patrimonio son de mantenimiento de registros, así como los de asignación de propiedad.

### *Mantenimiento de Registros*

Consiste en dar de alta, dar de baja, modificar y/o eliminar cualquiera de los campos. Todos los catálogos, así como los activos, pueden recibir mantenimiento. El historial no tiene acceso a este módulo, pues provee el histórico del bien.

### *Asignación de Propiedad*

Consiste en asignar un resguardante y una dependencia a un activo. Esto se realiza en dos situaciones: cuando se da de alta el activo, que se tiene que realizar en la pantalla de Tareas, y cuando se registra un movimiento del activo, que se realiza en la pantalla de Movimientos.



*Capítulo 5*

DISEÑO FÍSICO DEL SISTEMA

Arquitectura del Diseño Físico

El diseño físico, como se mencionó anteriormente, es la parte que el usuario comprende por el “sistema”. El diseño físico se procuró hacer de una manera simple, que permitiese al usuario ver de manera simple y clara su información, mientras que el “motor” del sistema hace los movimientos sobre la base de datos.

Para poder hacer que el usuario tenga acceso rápido a su información se utilizaron “vistas”. Estas son queries de SQL que son almacenados y compilados en el servidor. Esto permite al sistema operar sobre la base de datos en cuestión, mientras que el usuario siempre tiene una vista rápida de su información.

También se utilizaron controles Data Control de Visual Basic, que en este caso fueron Data Grids. Estos permitieron manejar la información de una manera amigable, que le permitiese al usuario ordenar y arreglar la presentación de su información a su gusto, mientras que el sistema operaba con librerías de ADO, para tener un tiempo de respuesta eficiente sobre las operaciones con la base de datos. Cabe mencionarse que no se utilizan los controles de ADO prediseñados por Visual Basic, pues restan desempeño al sistema.



### Estándares del Diseño

El diseño físico es de suma importancia, pues es el que le permitirá al usuario tener un acceso fácil a su información. Los controles deben ser visibles y el conjunto de los elementos que conforman a la forma deben tener un sentido lógico.

Lo más común en el diseño de formas de Visual Basic, es la realización de un diseño pobre. Suelen hacerse formas de tamaño estándar, con casillas colocadas en orden. El diseño de la forma debe permitir al usuario sentirse cómodo con la estructura. Su tamaño no debe ser excesivo, y la información debe estar organizada.

Forma de Diseño Pobre

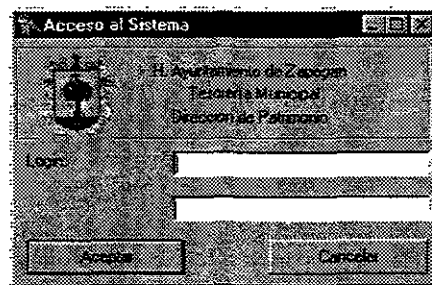
Forma mas Eficiente de Diseño



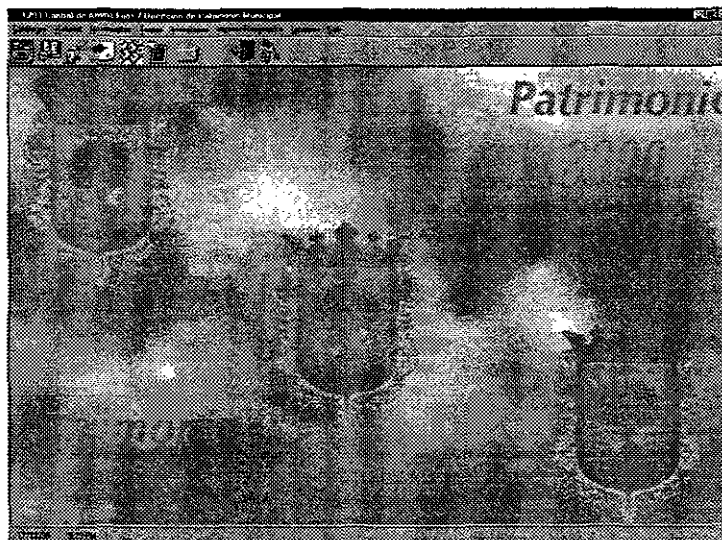
### Pantallas del Sistema de Patrimonio

El sistema desarrollado a Patrimonio consta de 14 pantallas. Esta es una breve explicación de la función de cada una de ellas:

**PANTALLA DE ACCESO:** Permite al usuario ingresar al sistema. Se tienen dos niveles de seguridad. El primero es el de usuario, que puede efectuar cualquier operación en el sistema. El segundo es el de administrador, que además puede dar de alta o eliminar a los usuarios.



**PANTALLA PRINCIPAL:** Es la pantalla contenedora de todas las demás pantallas o formas.

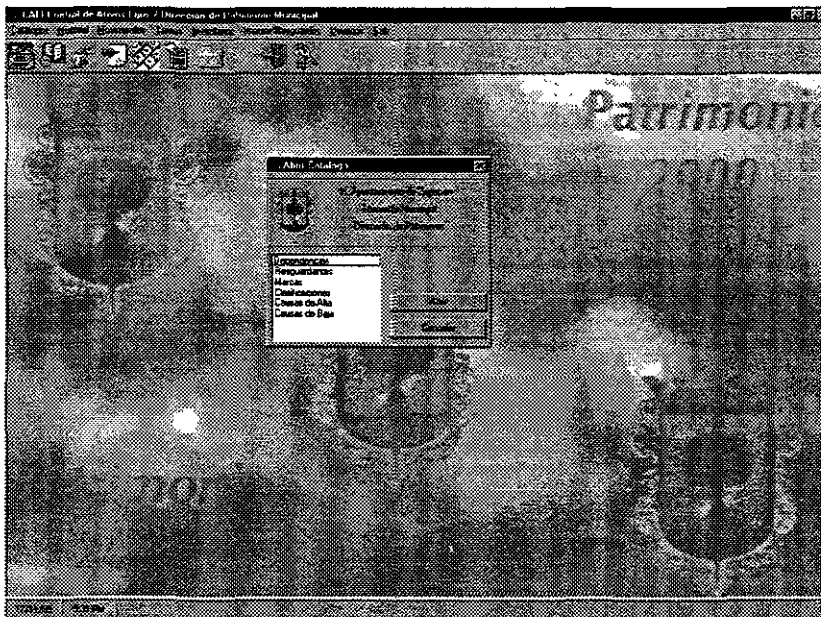






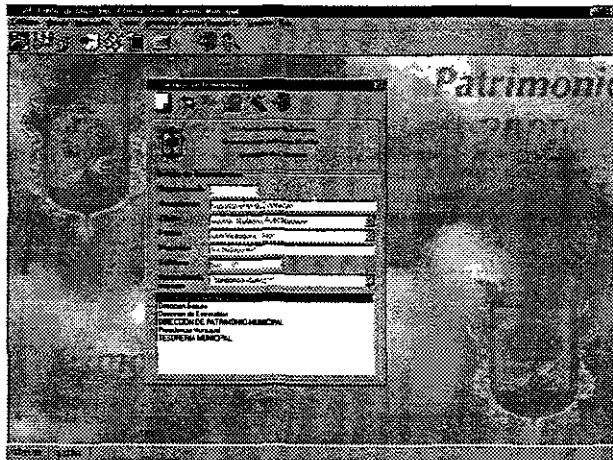
**PANTALLA DE CATÁLOGOS:** Esta pantalla da acceso a cada uno de los siguientes catálogos:

- Dependencias
- Resguardantes
- Modelos
- Clasificaciones
- Causas de Alta
- Causas de Baja

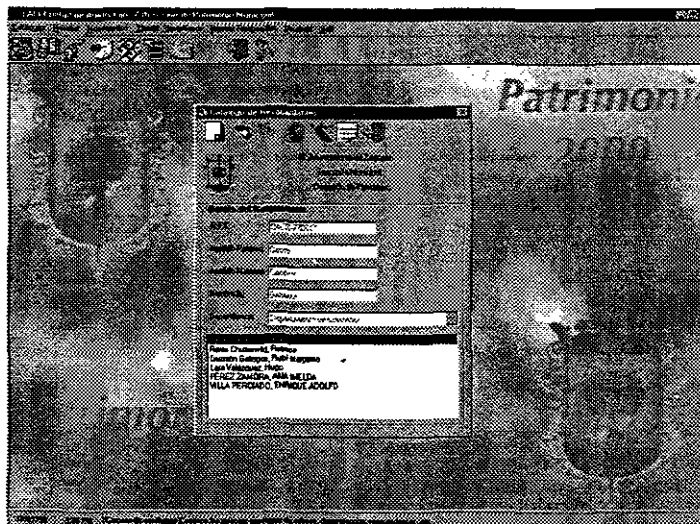




**CATÁLOGO DE DEPENDENCIAS:** Permite al usuario dar de alta, modificar o eliminar dependencias, así como especificar el enlace y el titular de éstas.

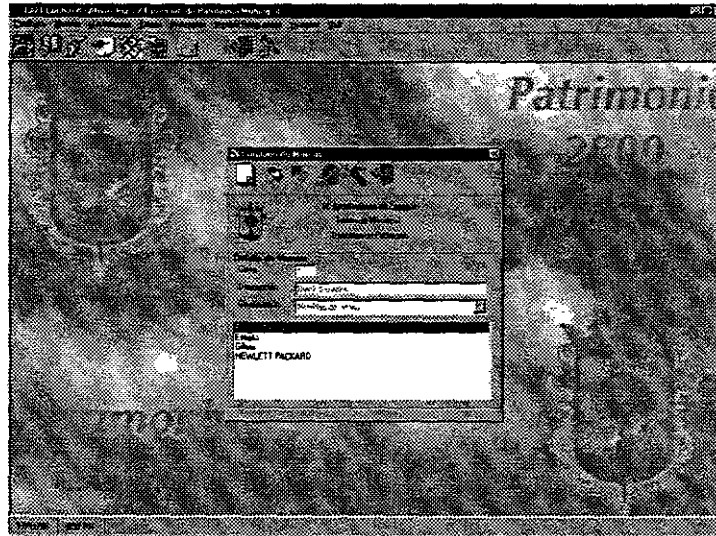


**CATÁLOGO DE RESGUARDANTES:** Permite agregar, modificar o eliminar usuarios. Se agrega a éste una lista de los bienes que tiene actualmente el usuario.

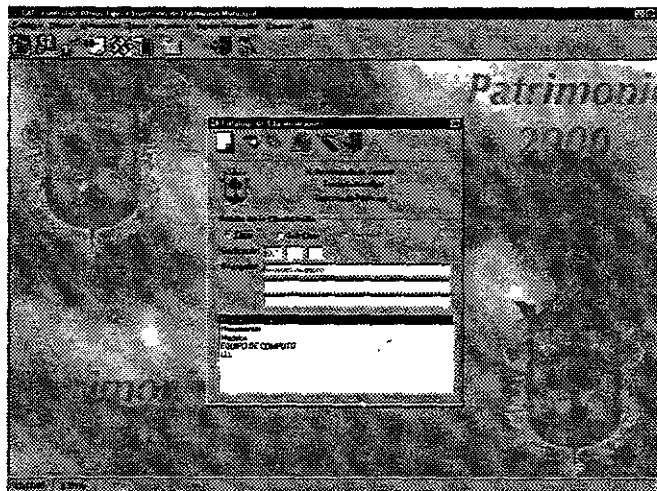




**CATÁLOGO DE MODELOS:** Permite al usuario asignar un nuevo modelo, así como a qué clasificación pertenece.

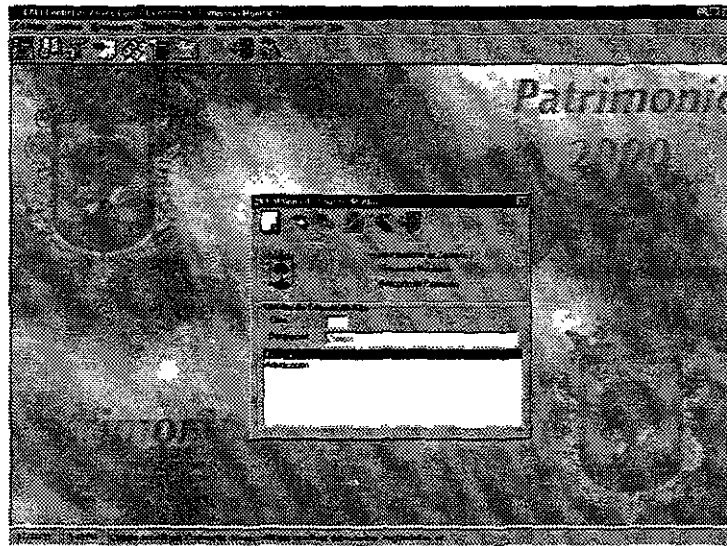


**CATÁLOGO DE CLASIFICACIONES:** Permite al usuario generar, modificar y/o eliminar las diversas clasificaciones, así como sus sub clasificaciones y sus sub sub clasificaciones.

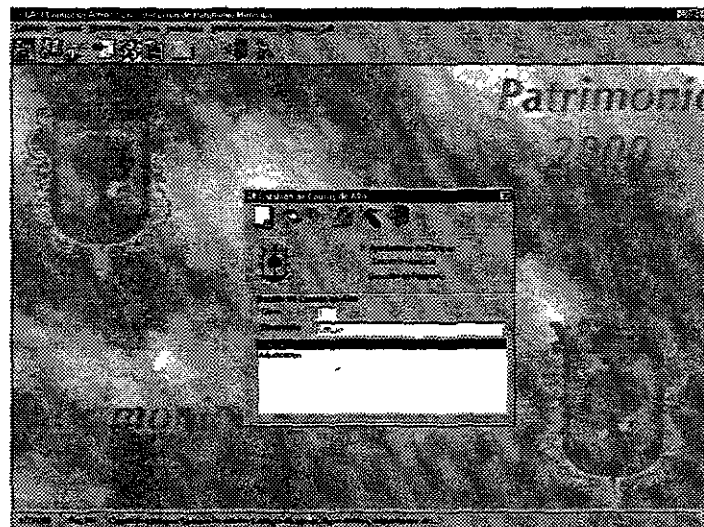




**CATÁLOGO DE CAUSAS DE ALTA:** Permite agregar, modificar o eliminar las causas de alta de un bien.



**CATÁLOGO DE CAUSAS DE BAJA:** Permite agregar, modificar o eliminar las causas de baja de un activo.





Universidad Nacional Autónoma de México  
Facultad de Ingeniería

CATÁLOGO DE ACTIVOS: Permite registrar un activo, así como modificarlo y/o darlo de baja. Este catálogo NO asigna resguardante ni dependencia, pero sí agrega éste a la lista de tareas, que se describirá más adelante.

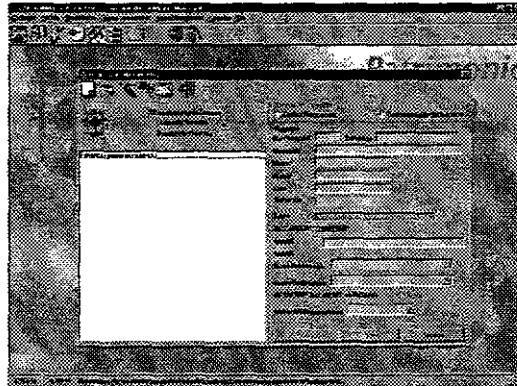
CAPI Control de Activos Fijos / Dirección de Patrimonio Material

Calendario de Asesorías

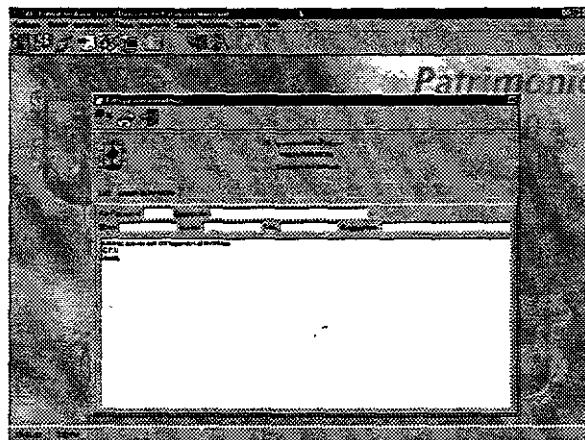
037351	PCPU CON MODEM	CJAS712202	TONA-440506-	NEGRETE	DE LA TORR	ANTONIO
037360	PCPU CON MODEM	CJAS712202	SOCR-560331-	AVILA	SOSA	ROSA PATRI
037345	PCPU CON MODEM	CJAS712203	VIRA-670717-	RUIZ	VILLALOBOS	ALEJANDRO
041563	PCPU LANX	CJAS806055	AEMM-758510-	MUÑOZ	ARELLANO	MARK ELOY
040231	PCPU	CJAS806120	DIED-003000-	ESPINOZA	DIAZ	DINA GEORG
040241	PCPU	CJAS807159	FAAL-720410-	AGUILAR	FRANCO	LUIS FRANC
041898	PCPU	CJAS811060	GALR-770226-	LOPEZ	GACNA	RAMSES
042286	PCPU	CJAS811060	YACL-440810-	CARDENAS	VALDIVIA	LUIS ALFON
080712	SERVIBAR CHICO/C	CM90N28	GCCA-000000-	CRISTOBAL	GOMEZ	ANTONIO
041589	KIT MULTIMEDIA	CMAS809000	AEMM-758510-	MUÑOZ	ARELLANO	MARK ELOY
042749	IMPRESORA H.P. DE	CN85G1127J	YACL-440810-	CARDENAS	VALDIVIA	LUIS ALFON
042206	IMPRESORA H.P. DE	CN86F131M2	GALR-770226-	LOPEZ	GACNA	RAMSES
042117	PCPU	COA8811007	MAVV-670102-	VAZQUEZ	MAGALLANES	VICTOR
037396	IMPRESORA	COF036959	ROAT-691201-	ACOSTA	RODRIGUEZ	ETHANIA V
037403	IMPRESORA	COF012651	SEML-841009-	MARTINEZ	SEVILLA	LEONARDO
037382	IMPRESORA	COF012711	VIRA-000000-	RUIZ	VILLALOBOS	ALEJANDRO
037386	IMPRESORA	COF012829	ROAT-691201-	ACOSTA	RODRIGUEZ	ETHANIA V
037404	IMPRESORA	COF012855	ROAT-691201-	ACOSTA	RODRIGUEZ	ETHANIA V
037363	IMPRESORA	COF012859	RUNJ-550401-	HURTADO	RUBIO	JORGE A
037364	IMPRESORA	COF012893	SELM-111111-	LOPEZ	SERRANO	MONTALBERT
037396	IMPRESORA	COF012821	TONA-440506-	NEGRETE	DE LA TORR	ANTONIO
037397	IMPRESORA	COF012923	MOPN-601109-	PERAFLOR	MONTAÑEZ	NORA ARACE
037406	IMPRESORA	COF012924	VIRA-670717-	RUIZ	VILLALOBOS	ALEJANDRO
037400	IMPRESORA	COF012925	SOCR-560331-	AVILA	SOSA	ROSA PATRI



**PANTALLA DE MOVIMIENTOS:** Abre la base de datos generada por el sistema de Levantamiento de Inventario, y lo carga en la lista. En caso de que el usuario lo decida, registra los movimientos capturados individualmente. También puede cambiar de resguardante y dependencia a un bien, también puede desincorporar un activo del inventario. Cabe mencionarse que estos movimientos los realiza automáticamente.



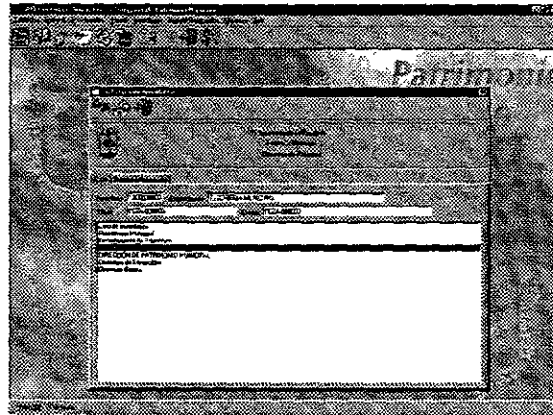
**PANTALLA DE INVENTARIO:** Permite al usuario generar los inventarios que serán registrados por la laptop. Presenta una lista de las dependencias.





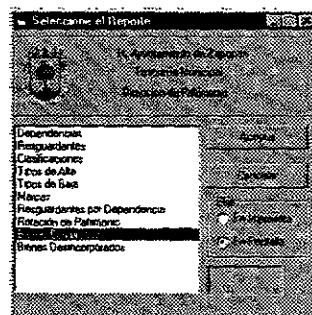
## Universidad Nacional Autónoma de México Facultad de Ingeniería

Si el usuario hace doble click sobre una de éstas, se presentan los activos que la dependencia contenga :



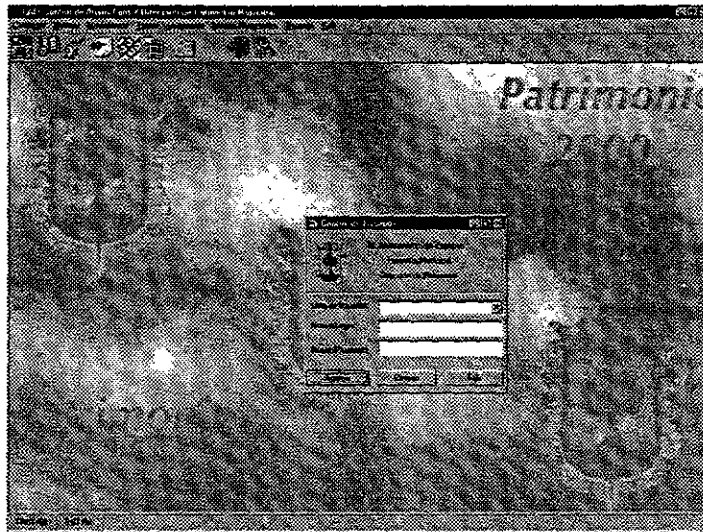
**PANTALLA DE SELECCIÓN DE REPORTES:** Permite al usuario seleccionar los siguientes reportes:

- Reporte de Dependencias
- Reporte de Resguardantes
- Reporte de Clasificaciones
- Reporte de Tipos de Alta
- Reporte de Tipos de Baja
- Reporte de Marcas
- Reporte de Resguardantes por Dependencia
- Reporte de Rotación de Patrimonio (selección por rango de fecha)
- Reporte de Bienes Comprados (selección por rango de fecha)
- Reporte de Bienes Desincorporados (selección por rango de fecha)

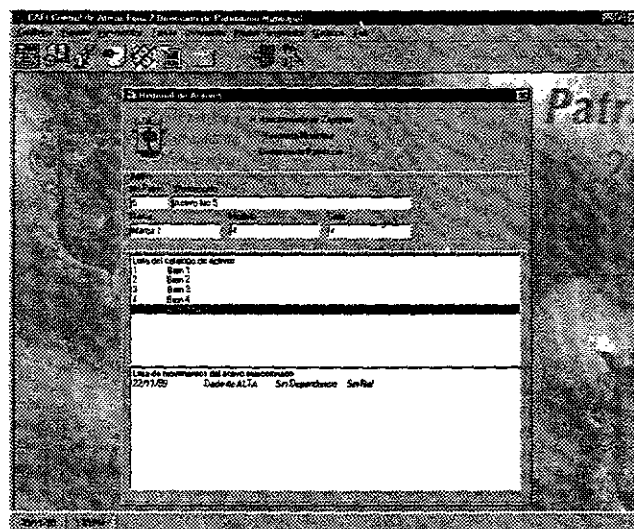




**PANTALLA DE CONTROL DE USUARIOS:** Exclusiva para los administradores, permite agregar o eliminar usuarios.



**PANTALLA DE HISTORIAL:** Presenta al usuario una lista con todos los bienes existentes para Patrimonio. Al seleccionar a uno de éstos, se presenta el historial de activo, con todos los movimientos que se han efectuado (así como el oficio con el que se realizó el movimiento) a lo largo de la vida del activo.







## *Capítulo 6*

### PRUEBAS AL SISTEMA

#### Calidad de Software

Como se ha mencionado a lo largo de esta tesis, el Sistema de Patrimonio es responsable de gran parte de las actividades de esta dependencia. Es, pues, necesario asegurarse que todas las partes se integren adecuadamente, así como que el sistema sea liberado con el menor número posible de errores.

Asegurar el 100% de la calidad en un sistema es prácticamente imposible, pero existen medidas para acercarse. Estos métodos permiten al software desarrollarse de tal manera que los errores sean localizados prontamente.

La calidad de software no sólo se debe revisar durante el desarrollo del sistema, sino también al terminarlo. Esto se hace desarrollando pruebas finales, que se comentarán más adelante en este capítulo.

#### Inspecciones Periódicas

La finalidad de éstas es permitir a los desarrolladores dedicarse completamente a la construcción del sistema, mientras que otro equipo se concentre en asegurar la calidad de éste. Las inspecciones presentan las ventajas de que el personal no está viciado con el sistema, además de que le pueden dedicar más tiempo a las opciones que presenta el sistema.



### Procedimientos de una Inspección

1. **Planeación:** El equipo de desarrollo entrega el código y las especificaciones al equipo de inspección. Debe recordarse que el equipo de inspección no está tan familiarizado con el producto, por lo que debe ilustrarse, antes de comenzar a revisarlo.
2. **Vista General:** El equipo de inspección debe tener un panorama del sistema en conjunto antes de empezar. Esto es peligroso, pues si el inspector no comprende bien el diseño, se puede convertir en una definición mal elaborada de lo que hace el sistema.
3. **Inspección:** Los inspectores deberán tener definidos los módulos que revisarán. La inspección puede abarcar incluso la revisión de código del desarrollo. Se debe procurar no discutir las soluciones a los problemas encontrados, pues esto corresponde a otra fase.
4. **Reporte de Inspección:** El reporte entregado por el equipo de inspección contiene todos los errores encontrados en el sistema. Se indica tanto el error como la severidad de éste.

Es responsabilidad del equipo de desarrollo corregir estas fallas, para así dar paso a una segunda (y tal vez una tercera) inspección. El objetivo final es obtener un código eficiente y libre de fallos.



Universidad Nacional Autónoma de México  
Facultad de Ingeniería

Pruebas y Calidad en el Sistema de Patrimonio

El aseguramiento de calidad en el sistema de Patrimonio se dividió en las siguientes tareas:

- Revisión periódica de código (quincenal)
- Pruebas periódicas a cada módulo del sistema (semanal)
- Una bitácora de cambios que se actualizó diariamente
- Uso de la aplicación Source Safe

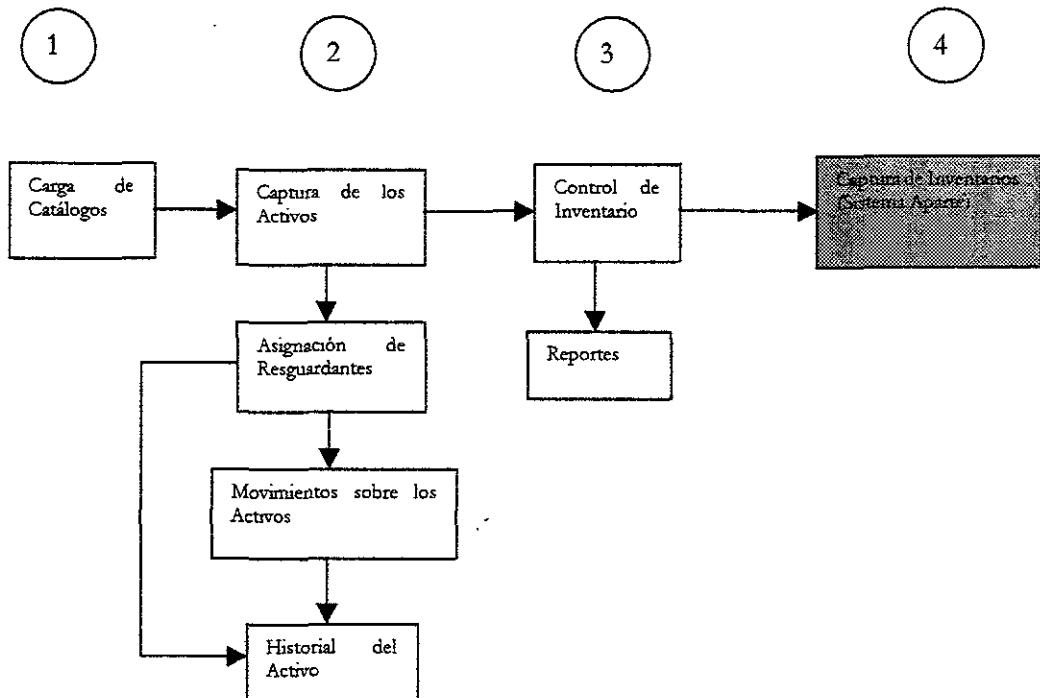


APÉNDICE 1

MANUAL DE USUARIO DEL SISTEMA DE PATRIMONIO

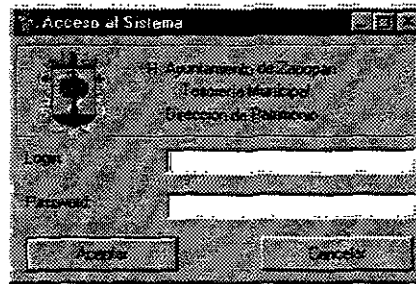
**Introducción al Sistema de Patrimonio de Bienes Muebles**

El sistema de Patrimonio de Bienes Muebles está diseñado para llevar el control de inventarios de una manera eficiente y sencilla para el usuario. El proceso que lleva a cabo el usuario debe ser el siguiente:





## Inicio del Sistema

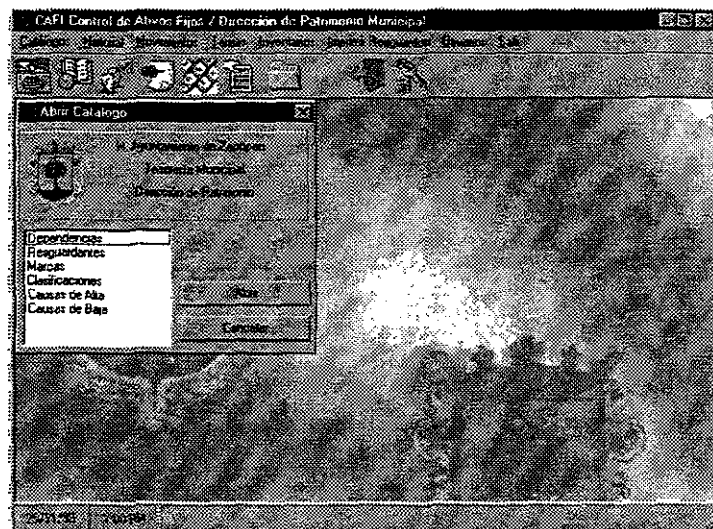


El sistema de Patrimonio de Bienes Muebles permite la entrada con un login y un password. El sistema tiene dos niveles de seguridad. Esto es, permite ingresar a un Administrador o a un Usuario. El administrador tiene permiso para dar de alta o baja a los usuarios del sistema, incluyendo a otros administradores.

## Carga de Catálogos

Los catálogos son todos aquellos elementos que, aunque no pertenecen al activo, tienen cierta relación con él. Los catálogos que tiene el sistema son:

- Dependencias
- Resguardantes
- Marcas
- Clasificaciones
- Causas de Alta
- Causas de Baja



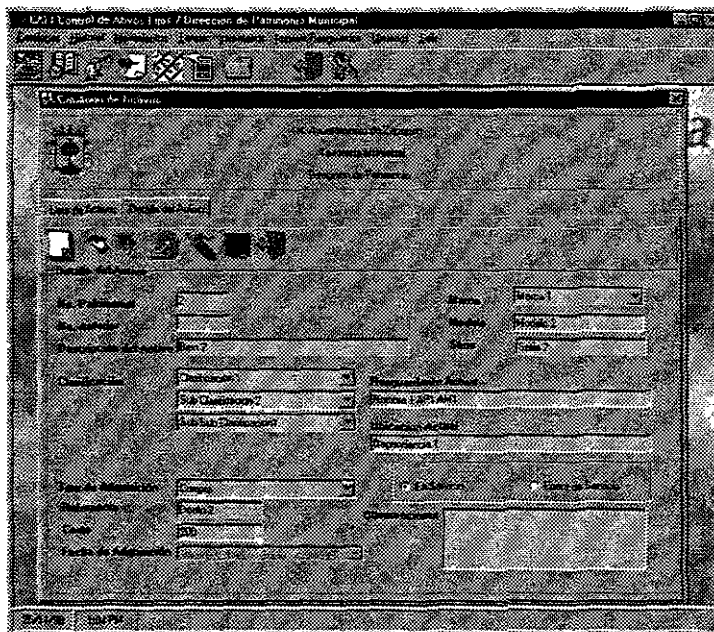


Una vez seleccionado el catálogo, se pueden realizar las siguientes operaciones:

- **Agregar** : Crea un nuevo elemento del catálogo
- **Modificar**: Altera la información de un elemento del catálogo
- **Eliminar**: Elimina un elemento del catálogo
- **Grabar**: Permite almacenar la información que está siendo agregada o modificada
- **Cancelar**: Permite deshacer el cambio **QUE SE ESTÁ REALIZANDO**, una vez que se graba, solo se puede cancelar eliminando.

### Carga de Activos

Aquí se selecciona al activo a consultar o a modificar. También es aquí donde se agregan Activos a la Base de Datos. Además de las opciones que se pueden realizar con los catálogos, en esta pantalla también se puede reetiquetar un activo. Al reetiquetarlo, aparecerá el número que tuvo anteriormente, y el historial cambiará sus movimientos a la nueva etiqueta.

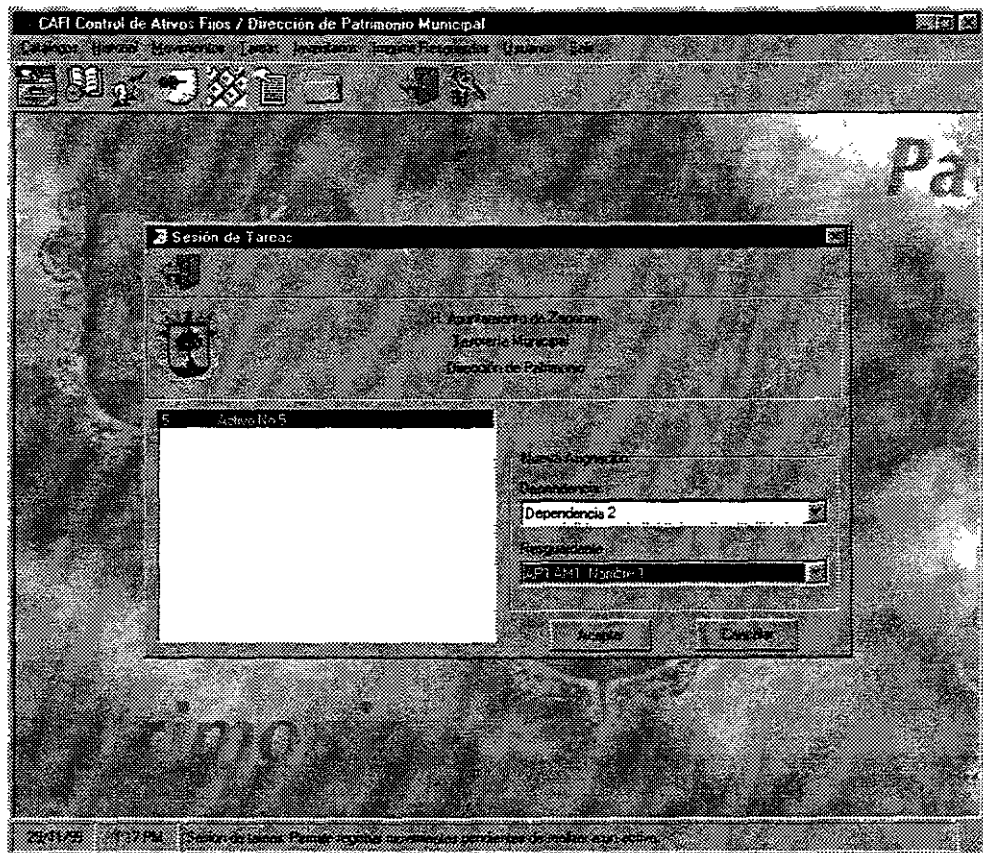


Una vez agregado el activo, se deberá ir a la pantalla de Tareas, para asignarle un Resguardante y Dependencia.



### Asignación de Resguardante (Tareas)

La pantalla de Tareas permite al usuario asignar un resguardante y una dependencia a los activos nuevos. EN CASO DE QUE EL ACTIVO YA EXISTA, se le deberá reasignar por medio de la pantalla de Movimientos.





## Movimientos Sobre los Activos

La pantalla de Movimientos permite realizar las siguientes operaciones:

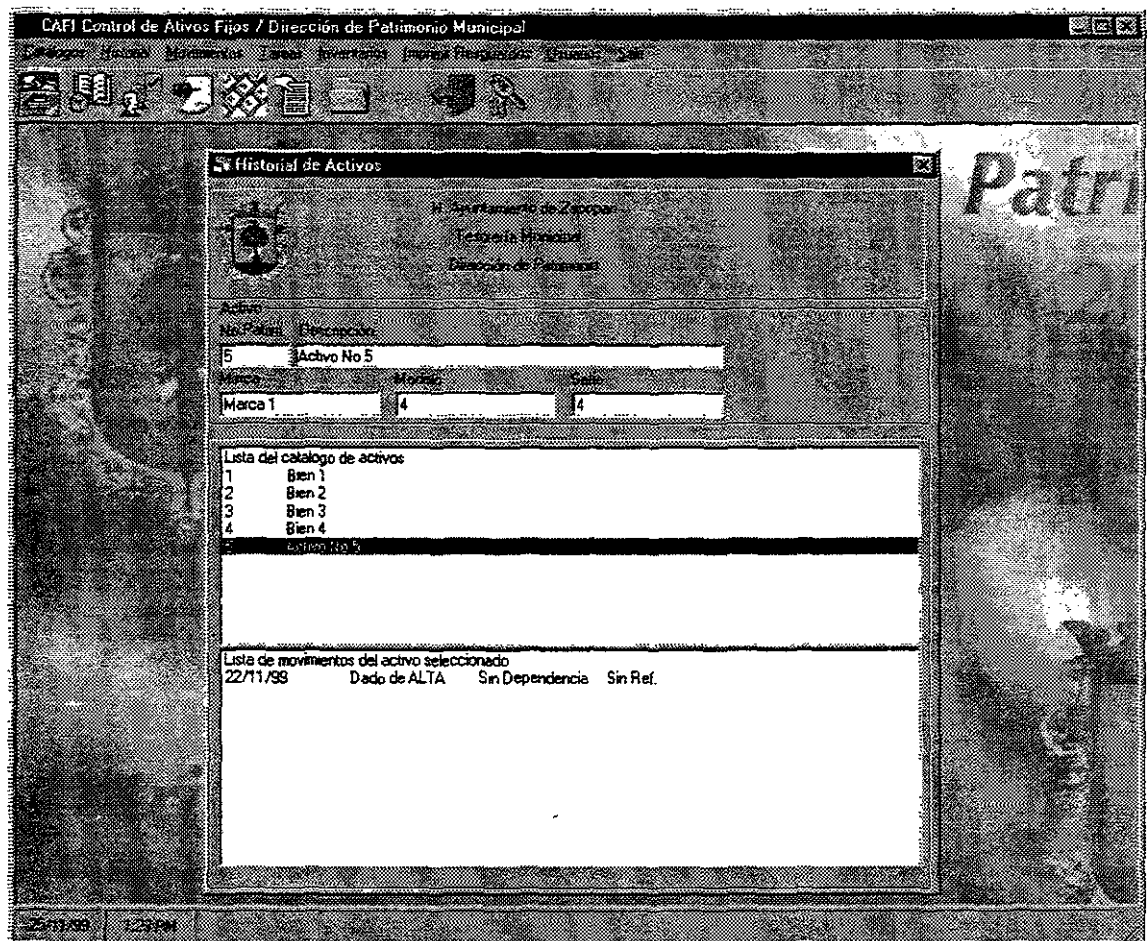
- Carga los posibles movimientos capturados por el sistema de Levantamiento de Inventario
- Realiza cambios de resguardante y dependencia sobre los activos existentes (en caso de que el activo sea nuevo, esto se deberá realizar en la pantalla de Tareas).
- Desincorpora un Activo del Inventario





### Historial de Activos

En esta pantalla, se visualizan los activos existentes, y una vez que se selecciona alguno, se presenta la información de todos sus movimientos. Básicamente es una pantalla de consulta, donde no se puede realizar ninguna operación sobre los activos.



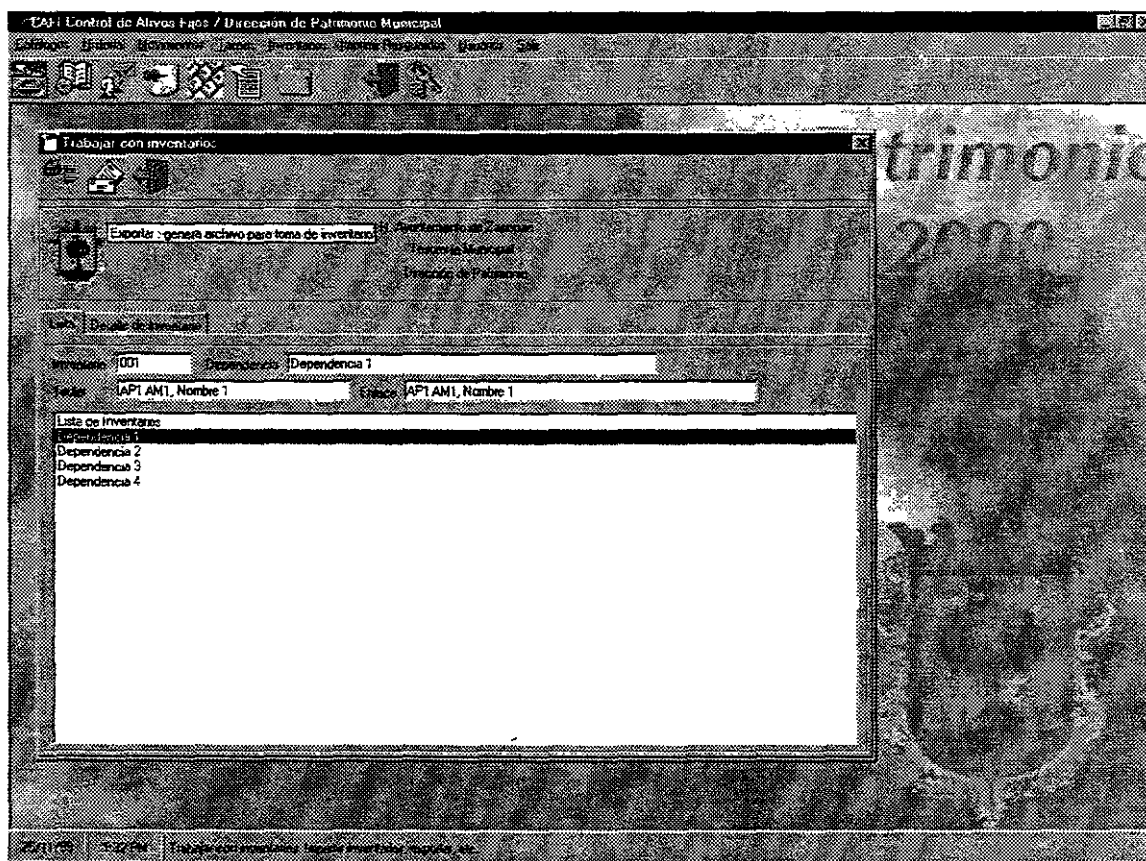


ESTA  
SALIR DE

DEBE  
BIBLIOTECA

### Control del Inventario

Esta pantalla permite dos cosas, básicamente. La primera es consultar los bienes que contiene cada dependencia, esto se hace haciendo doble click sobre la dependencia deseada. La segunda es la generación del inventario que irá al sistema de Levantamiento de Inventario. Al exportar el inventario deseado (seleccionando una dependencia), Se cargará en la base de datos Inventario.mdb la base de datos que deberá ser cargada en el sistema portátil.



trimonio  
2000

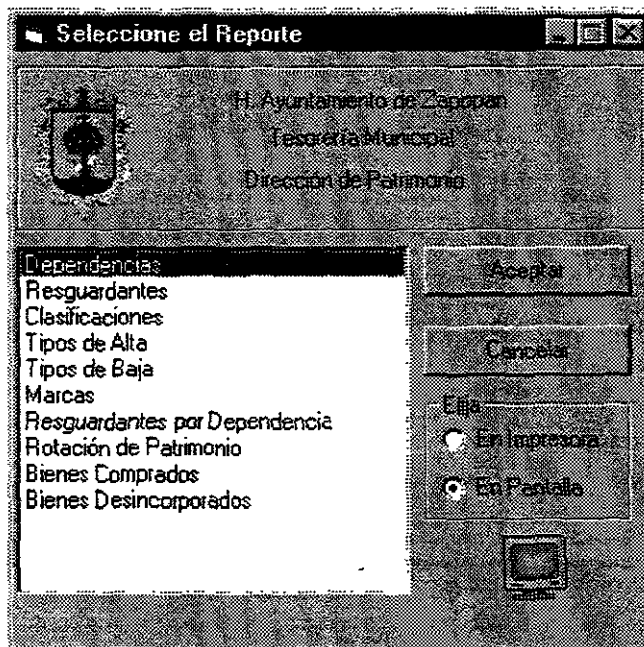
ESTA  
SALIR DE  
TESIS NO DEBE  
BIBLIOTECA



### Impresión de Reportes

La pantalla de reportes permite imprimir los siguientes reportes

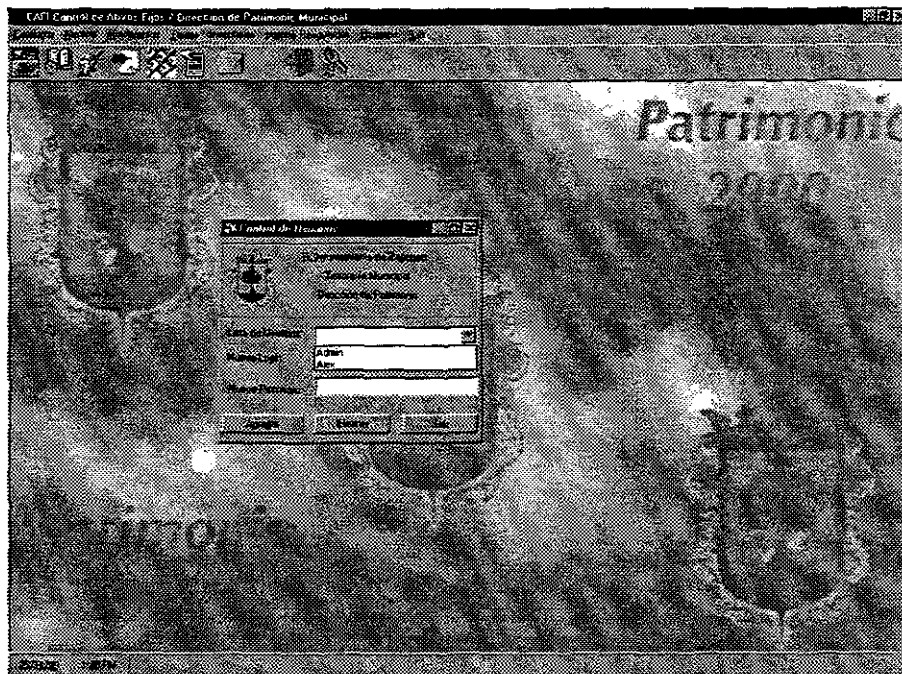
<i>Reporte</i>	<i>Selección por:</i>
Dependencias	Ninguno
Resguardantes	Ninguno
Clasificaciones	Ninguno
Tipos de Alta	Ninguno
Tipos de Baja	Ninguno
Marcas	Ninguno
Resguardantes por Dependencia	Dependencia
Rotación de Patrimonio	Rango de bienes
Bienes Comprados	Rango de fechas
Bienes Desincorporados	Rango de fechas





## Control de Usuarios

El control de usuarios es exclusivo para los administradores, y permite agregar o eliminar a los usuarios que accesan al sistema. Cabe mencionarse que los administradores pueden dar de alta y/o baja a otros administradores, por lo que se recomienda tener únicamente dos administradores.

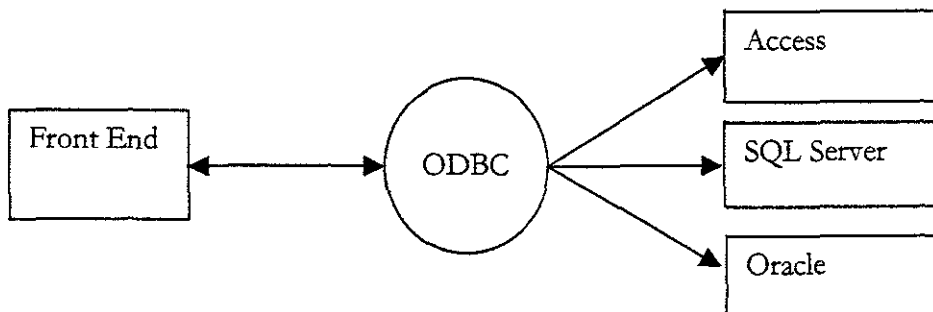




APÉNDICE 2

DOCUMENTACIÓN DE LA BASE DE DATOS

La base de datos diseñada para patrimonio fue desarrollada con SQL Server en mente. Esto se hizo por su excelente acoplamiento con Visual Basic, además de ser una de las más eficientes bases de datos en entorno cliente/servidor. Sin embargo, en caso de que la dependencia lo necesite, puede cambiar su base de datos al RDBMS que más le convenga, pues los drivers del sistema se basan en ODBC. Esto quiere decir que la base de datos se relaciona por un driver intermediario, lo que permite al sistema operar sin importarle el origen real de la base de datos.





## CONCLUSIONES

El desarrollo de aplicaciones para empresas particulares se ha convertido en una panacea para los ingenieros de cómputo. Esto no es un hecho censurable ni reprochable. Gran cantidad de los ingenieros en Computación de Nuestra Máxima Casa de Estudios (así como de muchas otras instituciones educativas orientadas a la computación), entran con la intención de generar software.

Lo que se debe considerar un hecho alarmante, es que en la actualidad gran cantidad de los ingenieros que se dedican a esto no generan software profesional. Con esto me refiero a que no se lleva a cabo ni un análisis previo, ni un diseño apropiado.

Considero una obligación de los ingenieros mexicanos el desarrollar la cultura de realizar ingeniería de software de calidad. La razón es que, mientras estamos sumamente aventajados en el desarrollo industrial, comercial e incluso el civil con respecto a muchos otros países, nuestras oportunidades de desarrollar software a nivel competitivo tienen como única desventaja el dominio del idioma inglés.

Como se mencionó en esta tesis, el diseño de arquitecturas de software está apenas comenzando en todo el mundo. Es decir que los estándares de calidad en el desarrollo de aplicaciones pronto comenzarán a tener un nivel muy superior al actual.

Los ingenieros que están saliendo hoy para comenzar a desarrollar las aplicaciones del mañana deberán adoptar esta nueva cultura de diseño/desarrollo. Solo así México podrá posicionarse como un país con diseño de software con nivel internacional.

La Ingeniería en Computación deberá esmerarse en educar a los próximos ingenieros para que logren superar la inmensa barrera que existe entre un "programador" y un "Constructor de Software". De esta manera podremos hacer de nuestros egresados gente digna de trabajar en el competitivo ámbito internacional como lo que son: Profesionistas mexicanos, gente responsable y capaz que busca el bienestar de la sociedad.



BIBLIOGRAFÍA

- Riordan, M. Rebecca  
*Relational Database Systems*.  
Microsoft Press, 1999.
- Osier, Batson and Grobman  
*Teach Yourself Delphi 3 in 14 Days*  
EUA.  
Sams Publishing, 1997.
- McConnell, Steven C  
*Code Complete, A Practical*  
*Handbook of Software Construction*  
Sams Publishin. 1993.
- Noble Camargo, Sergio. *Notas de Clase*  
*"Ingeniería de Programación"*.  
U.N.A.M, 1997.
- Hofmeister, Nord, Soni  
*Applied Software Architecture*  
Addison Wesley, 1999.
- Germenos Ramírez, Alejandro  
*Metodología de Desarrollo*  
Propiedad de CCS. 1999.
- Fowler, Martin H.  
*UML Distilled*  
Addison, Wesley, 1997.
- Kendall E. Keneth  
*Análisis y Diseño de Sistemas*  
Prentice-Hall. 1991.
- McConnell, Steven C.  
*Desarrollo y Gestión de Proyectos*  
*Informáticos*  
Microsoft Press, 1997.