



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

**ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES "ACATLAN"**

**"PROCESO UNIFICADO DE DESARROLLO
DEL SISTEMA INTEGRAL DE CONTROL
DE SOPORTE TECNICO DE LA
DELEGACION GUSTAVO A. MADERO"**

MEMORIA DE DESEMPEÑO PROFESIONAL

**QUE PARA OBTENER EL TITULO DE
LICENCIADO EN MATEMATICAS APLICADAS
Y COMPUTACION**

**P R E S E N T A
LUIS ARTURO GUTIERREZ DE LEON**

**A S E S O R :
GABRIEL DIAZ MIRON MAC DONOUGH**



ENERO 2004



Universidad Nacional
Autónoma de México



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.

ÍNDICE

PROCESO UNIFICADO DE DESARROLLO DEL SISTEMA INTEGRAL DE CONTROL DE SOPORTE TÉCNICO DE LA DELEGACIÓN GUSTAVO A. MADERO.

INTRODUCCIÓN

CAPÍTULO I. FUNCIONAMIENTO ACTUAL DEL SISTEMA DE SOPORTE TÉCNICO EN LA DELEGACIÓN GUSTAVO A. MADERO.	1
1.1. Antecedentes del Sistema Integral de Soporte Técnico	2
1.2. Descripción general del Sistema de Soporte Técnico que opera en la Delegación GAM.	4
 CAPÍTULO II. EL PROCESO UNIFICADO	 12
2.1. Antecedentes del Proceso Unificado	13
2.2. El Proceso Unificado	19
2.3. El Proceso de Desarrollo de Software	24
 CAPÍTULO III. MODELO DE CASOS DE USO Y MODELO DE ANÁLISIS DEL NUEVO SISTEMA INTEGRAL DE SOPORTE TÉCNICO DE LA DELEGACIÓN GUSTAVO A. MADERO.	 26
3.1. El Modelo de Casos de Uso	27
3.1.1. Diagramas de Casos de Uso	34
3.1.2. Descripción de los Casos de Uso	38
3.2. El Modelo de Análisis	52
3.2.1. Clases de Análisis	54
 CAPÍTULO IV. MODELO DE DISEÑO DEL SISTEMA INTEGRAL DE SOPORTE TÉCNICO DE LA DELEGACIÓN GUSTAVO A. MADERO.	 57
4.1. El Modelo de Diseño	58
4.2. Diagramas y Descripción de Clases	60
4.3. Diagramas de Secuencia	73
 CAPÍTULO V. MODELO DE IMPLEMENTACIÓN DEL NUEVO SISTEMA INTEGRAL DE SOPORTE TÉCNICO EN LA DELEGACIÓN GAM.	 101
5.1. El Modelo de Implementación	102
5.2. Políticas y Estándares de Programación en la Delegación GAM	104
5.3. Diseño de Componentes y Nodos	117
 Conclusiones	 121
Apéndice	123
Apéndice A El perfil profesional de Booch, Rumbaugh y Jacobson	124
Apéndice B Referencias de UML	126
Referencias Bibliográficas	131

INTRODUCCIÓN

El modelado es una parte básica de todas las etapas que conducen a un equipo de desarrollo a producir software de calidad. Se construyen modelos para encontrar la forma deseada del propio sistema. Se construyen modelos para comprender de mejor manera a los sistemas basados en software, y esto a su vez controla los posibles errores dentro del proceso de desarrollo.

La visión actual del modelado de sistemas con gran cantidad de software toma una perspectiva orientada a objetos. Esta forma vanguardista de visualizar el modelado se basa en el objeto clase, actualmente el enfoque orientado a objetos cubre perfectamente cualquier tamaño y complejidad de un sistema. Es por ello que en la Delegación Gustavo A. Madero se decide iniciar con el objetivo de **diseñar soluciones orientadas a objetos que sean viables para el proceso de implementación** en todo sistema que involucre grandes bloques de software.

En el caso particular del Sistema Integral de Control de Soporte Técnico, se decide retomar el Proceso Unificado de Rational como marco de trabajo para el proceso de desarrollo.

Inicialmente se efectúa una investigación acerca de los antecedentes más recientes de la Coordinación de Informática como unidad de desarrollo y soporte tecnológico dentro de la Delegación Gustavo A. Madero.

Partiendo de las necesidades de los usuarios finales del sistema se analizan y capturan los requerimientos más específicos de la Unidad Departamental de Soporte Técnico, para tomarlos como base para la producción del nuevo sistema software.

Utilizando las bases de la sintaxis y semántica del Lenguaje Unificado de Modelado (UML) se desarrolla el modelado de cada uno de los objetos que intervienen en el sistema.

Inicia el modelado desde la vista arquitectónica del Modelo de Casos de Uso desarrollando las especificaciones de estos y sus especializaciones correspondientes.

Proseguimos a refinar los requisitos funcionales capturados en el Modelo de Casos de Uso a través de las clases de análisis en las que se basa el Modelo de Análisis.

Posteriormente se modela el sistema para que soporte los requisitos no funcionales implementando físicamente los Casos de Uso, así como definiendo los atributos y operaciones de las Clases Estáticas y Dinámica del sistema.

Se describen las políticas y estándares de programación que deben seguir cada uno de los grupos de desarrollo de la Unidad Departamental de Desarrollo y Mantenimiento de Sistemas, partiendo de estos estándares se modela el sistema en términos de Componentes destacando los archivos de ejecución y su ubicación física.

El desarrollo concluye con la evaluación de los resultados evaluando al Proceso Unificado de Rational y UML como herramientas de desarrollo.

CAPÍTULO I

**FUNCIONAMIENTO ACTUAL DEL
SISTEMA DE SOPORTE TÉCNICO
EN LA DELEGACIÓN GUSTAVO A.
MADERO**

1.1 ANTECEDENTES DEL SISTEMA INTEGRAL DE SOPORTE TECNICO

Debido a la creciente demanda de crear modelos de Tecnología de Información de punta en todas las entidades gubernamentales del Distrito Federal, la Delegación Gustavo A. Madero (GAM), tiene la iniciativa de implementar proyectos tecnológicos de vanguardia para el óptimo funcionamiento como entidad de Gobierno de la Ciudad de México.

La Delegación Gustavo A. Madero, se ha dado a la labor de crear proyectos informáticos que sirvan de base para el desempeño de sus funciones de gobierno dentro de su jurisdicción geográfica en la ciudad de México.

En la Delegación GAM, la Coordinación de Informática es la encargada de implementar proyectos de Tecnología de Información que satisfagan las necesidades de dicha entidad del Gobierno de la Ciudad de México

El desarrollo de software que cubre con las necesidades de la Delegación GAM, se ha convertido una prioridad para el Jefe Delegacional y por consiguiente para la Coordinación de Informática.

Debido al óptimo desempeño que demanda el Jefe Delegacional, los funcionarios encargados de las diversas Unidades Departamentales adscritas a la Delegación GAM, envían sus peticiones de desarrollo de Tecnologías de Información a la Coordinación de Informática quien es la encargada, a través de la Unidad Departamental de Desarrollo y Mantenimiento de Sistemas, Unidad Departamental de Operación y Unidad Departamental de Soporte Técnico, de implementar proyectos de desarrollo de tecnología para darle solución a las demandas de las diversas Jefaturas dentro de la Delegación.

La Coordinación de Informática, la Jefatura de Soporte Técnico, la Jefatura de Desarrollo y Mantenimiento de Sistemas y la Jefatura de Operación, deciden iniciar el proyecto de control del inventario informático asignado a las diferentes Unidades Departamentales de la Delegación GAM, así como de los servicios de asistencia técnica prestados por la Unidad Departamental de Soporte Técnico.

El Sistema Integral de Control de Soporte Técnico, surge a partir de la necesidad de la Unidad Departamental de Soporte Técnico de llevar un control riguroso del inventario informático asignado a las diferentes Unidades Departamentales de la Delegación GAM.

La Unidad Departamental de Soporte Técnico es la encargada de prestar asistencia técnica en tecnología informática a las diferentes Unidades Departamentales adscritas a la Delegación GAM, así como de hacer válida la póliza de garantía de los equipos que soliciten servicio y aún tengan vigencia.

Las principales funciones de asistencia técnica de la Unidad Departamental de Soporte Técnico se basan en las necesidades de las Unidades de la Delegación GAM, pero en términos generales estas se dividen en dos: Los servicios de carácter correctivo y preventivo que tienen relación con la garantía de los equipos, software, hardware y telecomunicaciones, y los servicios de asistencia técnica prestados por el personal adscrito a la Unidad Departamental de Soporte Técnico, dentro de los cuales destacan debido a su alta demanda:

asistencia en respaldo de información, servicio de red, soporte en Internet, soporte en virus informáticos, etc.

Debido a la gran demanda de solicitudes de asistencia técnica recibidas por la Coordinación de Informática, nace el proyecto denominado **Sistema Integral de Control de Soporte Técnico**, el cual tiene como finalidad cubrir los siguientes puntos:

- Administrar el Inventario Informático de la Delegación GAM.
- Gestionar el Equipo Asignado por Unidad Departamental en la Delegación GAM.
- Administrar el Control de Asistencia Técnica brindada por la Unidad Departamental de Soporte Técnico.

A continuación se muestra el organigrama de la Delegación Gustavo A. Madero

ORGANIGRAMA DE LA DELEGACIÓN GUSTAVO A. MADERO

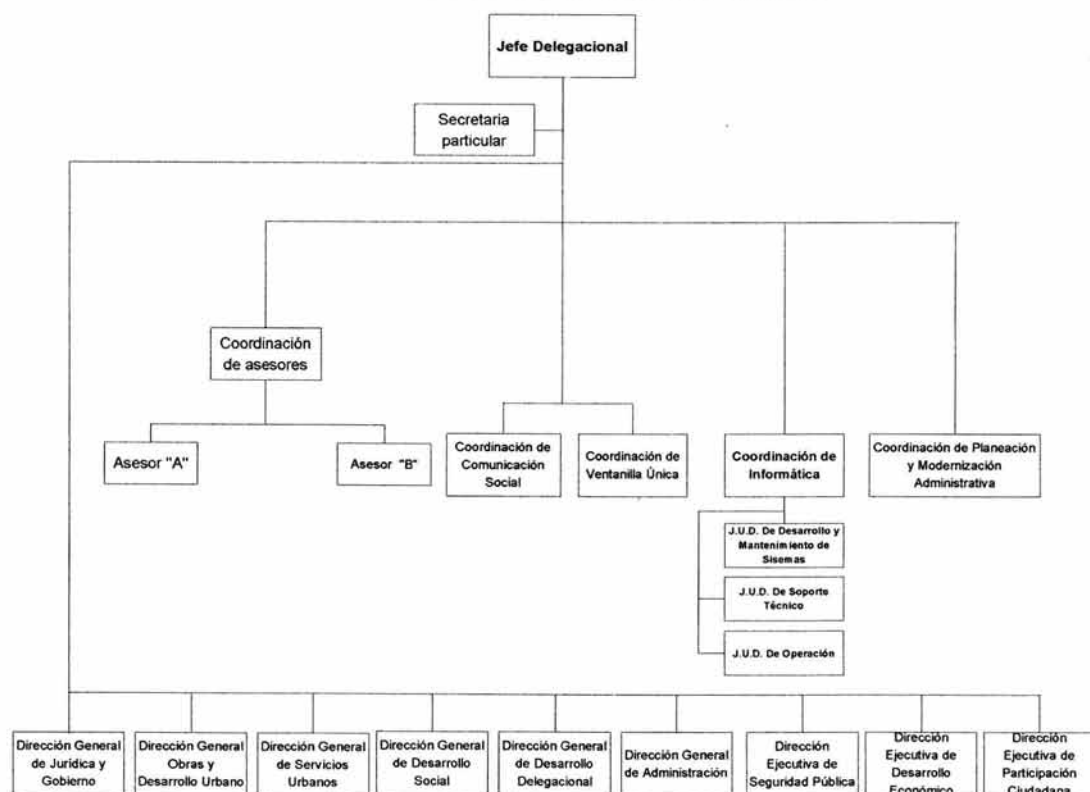


Figura 1.1
Organigrama Delegación GAM

1.2 DESCRIPCIÓN GENERAL DEL SISTEMA DE SOPORTE TÉCNICO QUE OPERA EN LA DELEGACIÓN GAM.

1. El Área solicitante, realiza su petición de soporte técnico, a la Coordinación de Informática en alguna de las siguientes modalidades:
 - Personalmente
 - Por escrito
 - Vía telefónica
2. La Coordinación de Informática canaliza la petición de servicio a la Jefatura de Soporte Técnico.
3. Para poder responder a la petición de soporte técnico, y localizar el equipo que solicita servicio, la Jefatura de Soporte Técnico requiere pedir los datos al área solicitante, para tener la descripción general del equipo (Serie, Tipo de Equipo, Marca, Garantía), al cual se le prestará asistencia técnica, así como de la Unidad Departamental que solicitó el servicio (Clave de Unidad Departamental, Descripción del Area, Teléfono, Extensión, e-mail, etc.).
4. Si el equipo que necesita servicio tiene póliza de garantía vigente, se hace válida la garantía.
5. El centro de servicio correspondiente corrige el desperfecto del equipo y entrega el equipo a la Jefatura de Soporte Técnico.
6. La Jefatura de Soporte Técnico se encarga de efectuar el reporte del tipo de servicio que recibió el equipo correspondiente.
7. Si el equipo que necesita servicio no tiene póliza de garantía vigente, la Jefatura de Soporte Técnico envía a su personal para prestar la asistencia técnica que corresponde.
8. El personal adscrito a la Jefatura de Soporte Técnico se encarga de brindar el apoyo de soporte técnico para el área solicitante.
9. Si el personal de la Jefatura de Soporte Técnico no puede corregir el desperfecto, el equipo es canalizado a un taller externo, que se encarga del servicio.
10. Si el personal de la Jefatura de Soporte Técnico pudo efectuar correctamente la asistencia técnica correspondiente, prosigue a realizar el reporte de servicio. (Ver diagrama de actividades Solicitud de servicio, figura 1.2).

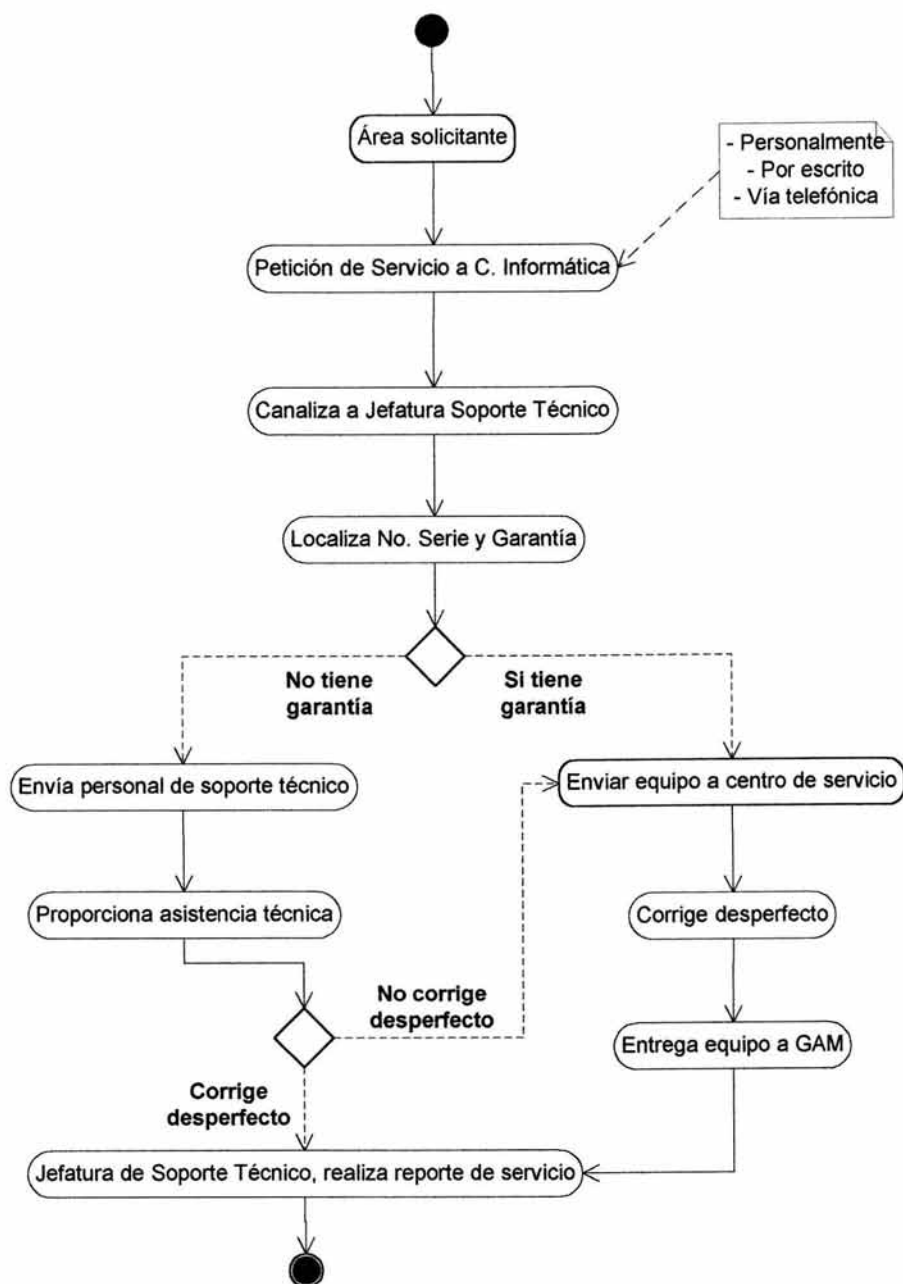


Figura 1.2
Diagrama de Actividades: Solicitud de servicio

PROCEDIMIENTO PARA LOCALIZAR UN REPORTE DE SERVICIO DE SOPORTE TÉCNICO

El usuario:

1. Proporciona el No. del reporte que desea buscar.
2. Efectúa una búsqueda en los archiveros y carpetas del reporte solicitado.
3. Si no localiza el No. de reporte de servicio vuelve al paso 1, de lo contrario el usuario si localizó el No. de reporte de servicio (Ver diagrama de actividades Buscar reporte de servicio, figura 1.3).

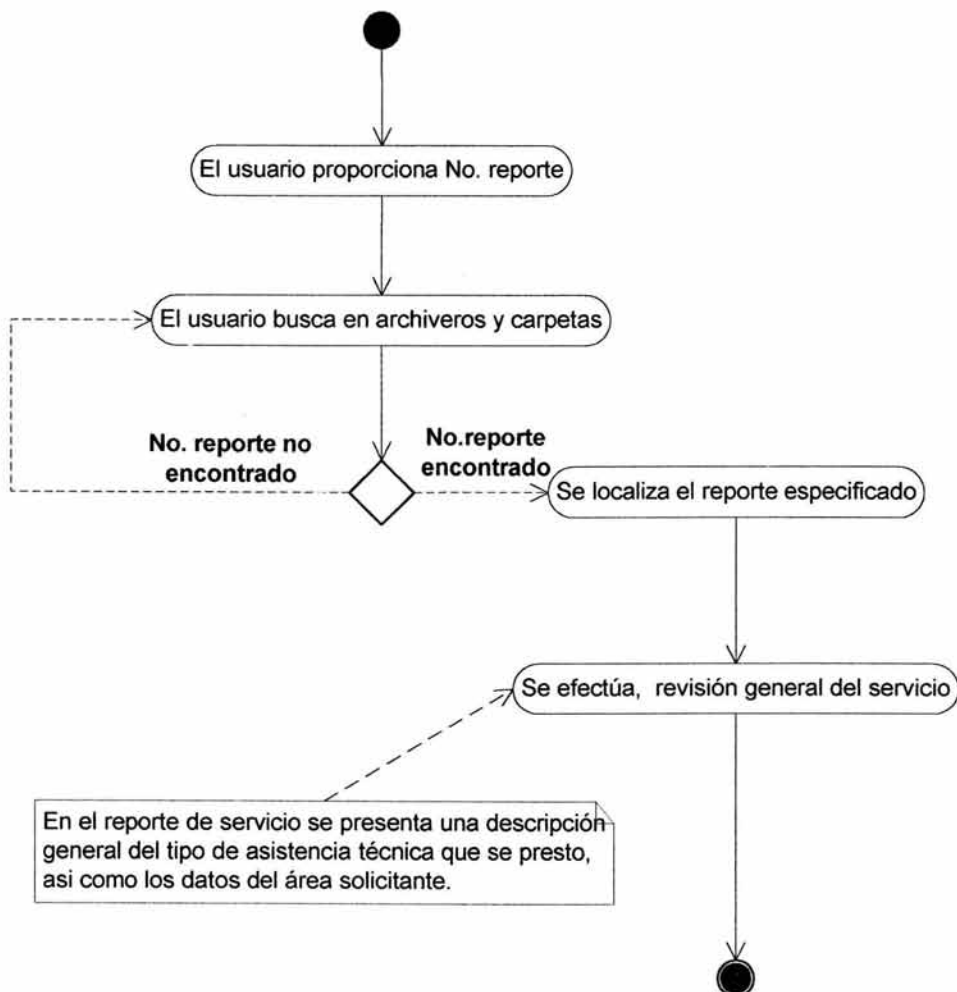


Figura 1.3
Diagrama de Actividades: Actualizar Área de Soporte Técnico
Caso: Buscar reporte de Servicio

PROCEDIMIENTO PARA INGRESAR UN NUEVO REPORTE DE SERVICIO

El usuario:

1. Proporciona la fecha en la que se efectuó el servicio de soporte.
2. Proporciona el No. de reporte asignado para este servicio de soporte.
3. Ingresar el nombre del solicitante del servicio.
4. Ingresar el puesto del solicitante del servicio.
5. Ingresar la Unidad Departamental solicitante del servicio.
6. Ingresar el número de extensión del área solicitante.
7. Ingresar el número de teléfono del área solicitante.
8. Ingresar el número de serie del equipo al que se le brindó el servicio.
9. Ingresar la descripción general del equipo (Tipo de equipo, marca del equipo, modelo del equipo, fecha de inicio y término de garantía) al que se le brindó el servicio.
10. Ingresar el tipo de asistencia técnica prestada al equipo.
11. Confirma que la información ingresada sea correcta.
12. Si la información no es correcta, el usuario modifica la información que es incorrecta y vuelve al paso 11, de lo contrario efectúa el paso 13.
13. Archiva en las carpeta del área de Soporte Técnico el nuevo reporte de servicio (Ver diagrama de actividades Ingresar nuevo reporte de servicio, figura 1.4).

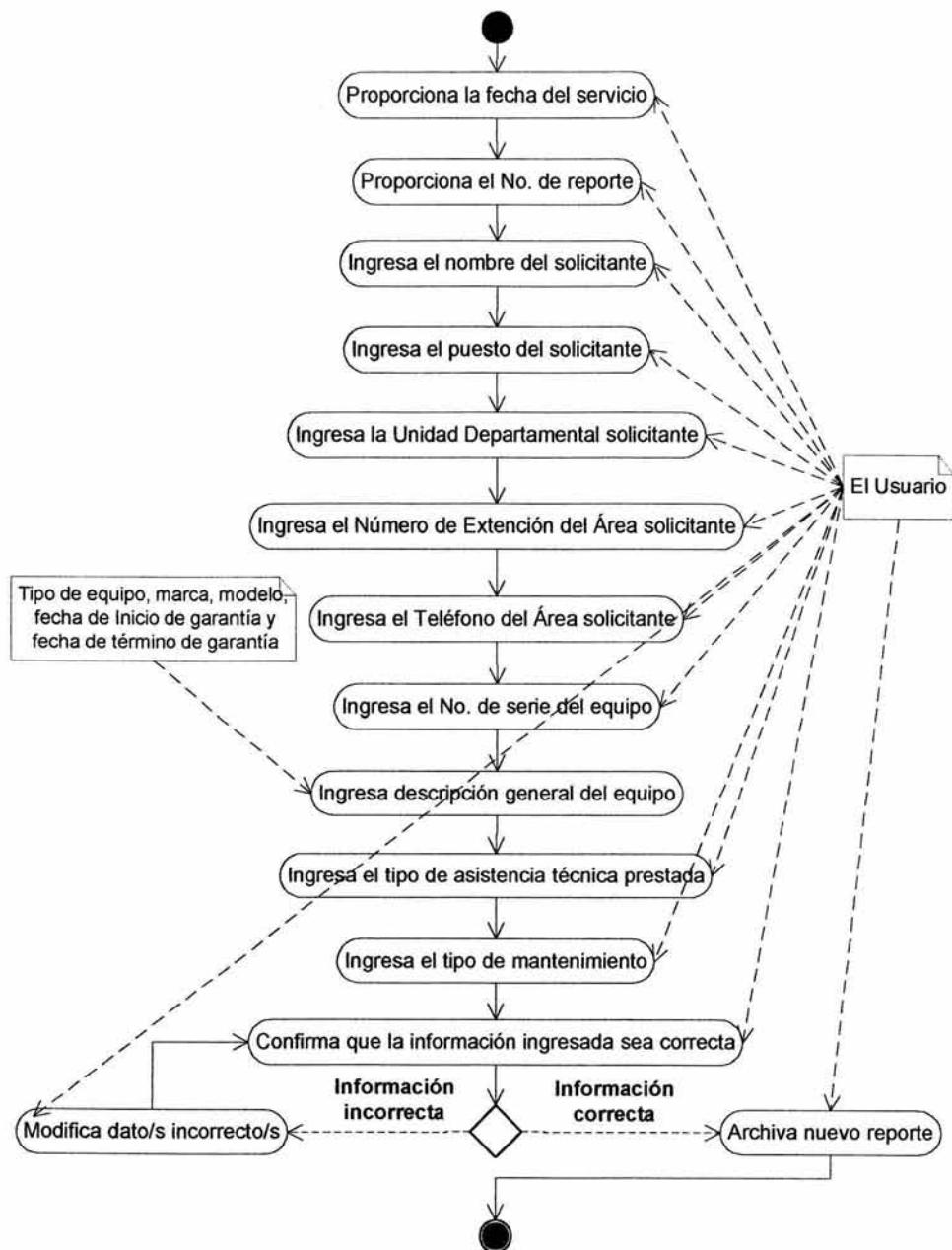


Figura 1.4
Diagrama de Actividades: Actualizar Área de Soporte Técnico
Caso: Ingresar nuevo reporte de servicio

Procedimiento para modificar los datos de un reporte de servicio

El usuario:

1. Busca en los archiveros y carpetas del área de Soporte Técnico el reporte que necesita modificar.
2. Si el usuario no encuentra el reporte de servicio, vuelve al paso 1, de lo contrario realiza el paso 3.
3. Modifica la información requerida del reporte de servicio.
4. Verifica que la información ingresada sea correcta.
5. Si la información ingresada no es correcta, regresa al paso 3 de lo contrario realiza paso 6.
6. Archiva el reporte de servicio (Ver diagrama de actividades Modificar datos del reporte de servicio, figura 1.5).

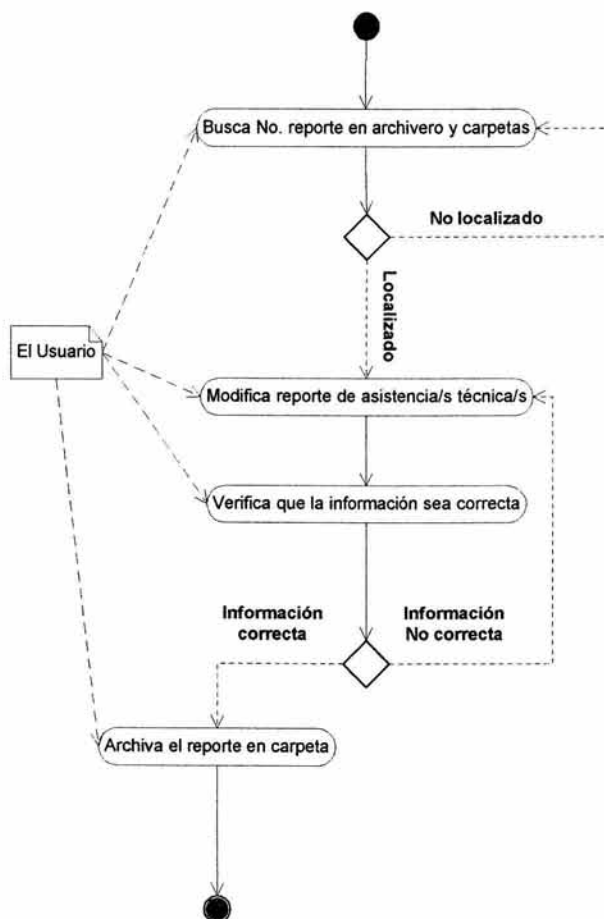


Figura 1.5
Diagrama de Actividades: Actualizar Área de Soporte Técnico
Caso: Modificar datos del reporte de servicio

PROCEDIMIENTO PARA REALIZAR BAJA DE REPORTE DE SERVICIO

El usuario:

1. Busca el número de reporte que desea dar de baja .
2. Si el número de reporte no se localiza regresa al paso 1, de lo contrario realiza paso 3.
3. Efectúa petición de baja al área de Soporte Técnico.
4. Si no es autorizado para realizar la baja, finaliza el proceso, de lo contrario realiza paso 5.
5. Realiza la baja del reporte de servicio (Ver diagrama de actividades Realizar baja de reporte de servicio, figura 1.6).

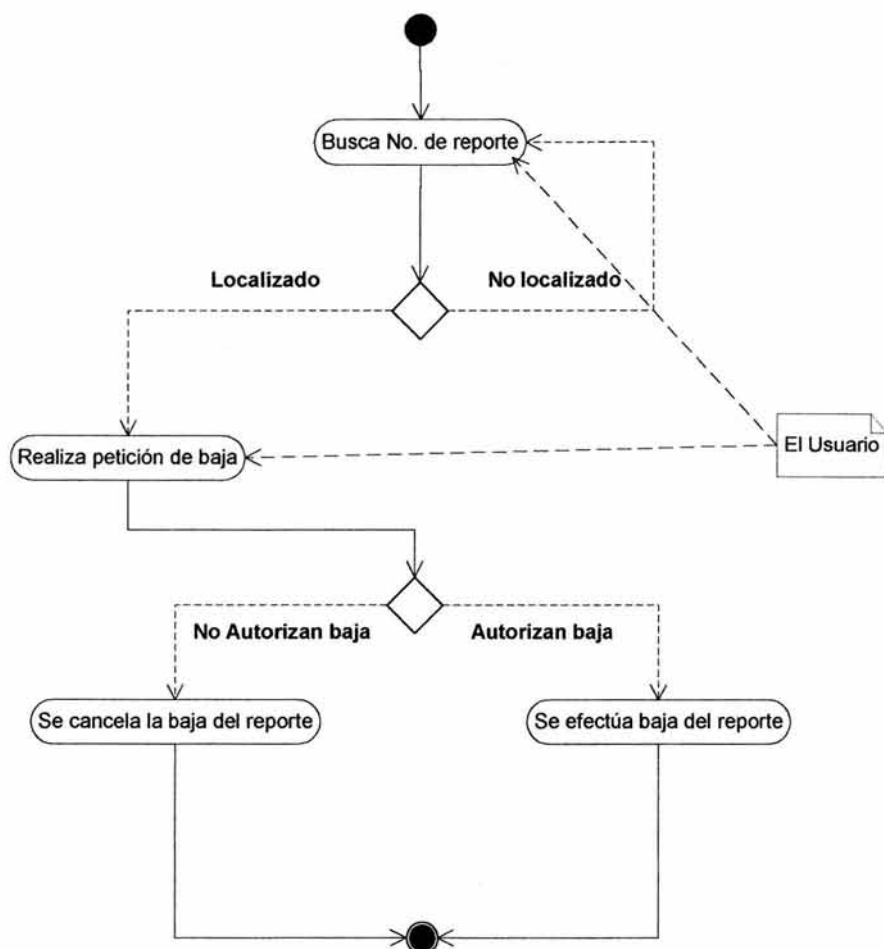


Figura 1.6
Diagrama de Actividades: Actualizar Área de Soporte Técnico
Caso: realizar baja de reporte de servicio

PROBLEMÁTICA ACTUAL DEL SISTEMA

1. La Coordinación de Informática, Jefatura de Unidad Departamental de Desarrollo y Mantenimiento de Sistemas, Jefatura de Unidad Departamental de Soporte Técnico y Jefatura de Unidad Departamental de Operación, coinciden en que es necesaria la Implementación del Sistema Integral de Control de Soporte Técnico para (el equipo asignado a) la Delegación Gustavo A. Madero.

La Jefatura de Unidad Departamental de Soporte Técnico:

2. Desearía tener mayor participación con el inventario del equipo informático dentro la Delegación Gustavo A. Madero.
3. Sugiere elaborar los catálogos necesarios para poder identificar las características básicas del equipo informático asignado a la Delegación (Tipos de equipo, Marcas y Modelos).
4. Requiere que la información del equipo asignado a la Delegación sea de carácter confidencial, a la que tengan acceso únicamente las áreas que así lo requieran.
5. Manifiesta que en el inventario del equipo informático, aparecen equipos duplicados con el mismo número de serie, lo cual entorpece su labor.
6. Manifiesta que en el inventario del equipo informático, aparecen equipos duplicados, asignados a diferentes Unidades Departamentales o Jefaturas de la Delegación, lo cual no permite que se lleve a tiempo, ni con la exactitud requerida, el soporte del hardware y software.
7. Necesita llevar un control riguroso del Soporte Técnico prestado, a las diferentes Direcciones, Subdirecciones y Jefaturas que forman parte de la Delegación, lo cual le permitirá llevar estadísticas para poder implementar programas de soporte correctivo y preventivo.
8. Recomendamos llevar un control riguroso, entre el inventario del equipo informático y el equipo informático asignado a cada Unidad Departamental de la Delegación.

CAPÍTULO II

EL PROCESO UNIFICADO

2.1 ANTECEDENTES DEL PROCESO UNIFICADO

El Proceso Unificado está equilibrado por ser el producto final de tres décadas de desarrollo y uso práctico. Su desarrollo como producto sigue un camino desde el Proceso Objectory, pasando por el Proceso Objectory de Rational hasta el Proceso Unificado de Rational. Su desarrollo ha recibido influencias de muchas fuentes. (Ver figura 2.1)

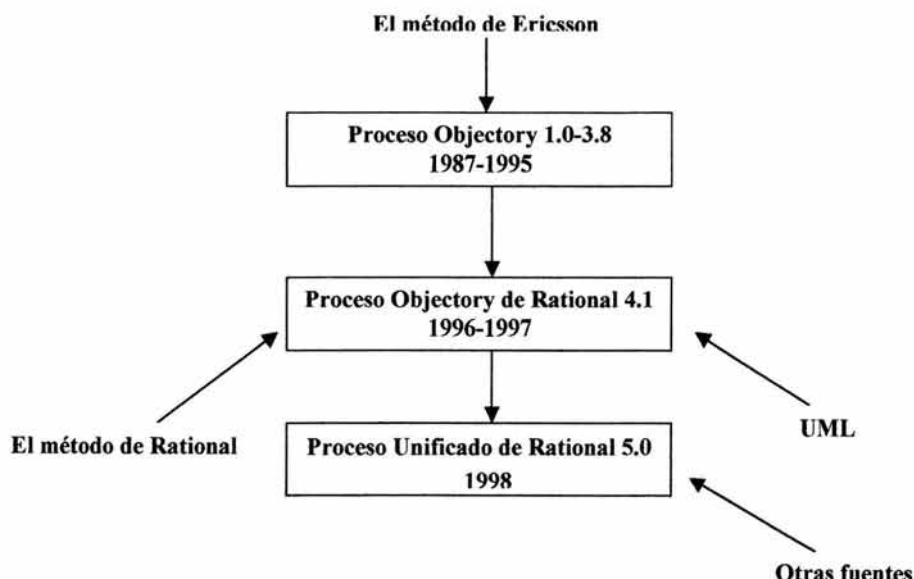


Figura 2.1
La historia del Proceso Unificado
[Booch Grady, etal, 2000 (a)]

EL METODO DE ERICSSON

En palabras de Peter F. Drucker, "El método Ericsson es una innovación basada en el conocimiento" [Booch Grady, et al, 2000 (a)]. Él nos informa de que hay un lapso de tiempo prolongado entre la emergencia del nuevo conocimiento y su destilación de tecnología utilizable. Entonces sucede otro largo periodo antes de que esta nueva tecnología aparezca en el mercado en productos o servicios.

Un motivo para este largo tiempo de aparición es que la innovación basada en el conocimiento se cimienta en la unión de muchos tipos de conocimiento, y esto lleva su tiempo. Otra razón es que las personas tienen que hacer efectiva la nueva idea y necesitan tiempo para comunicarla a todos los demás.

Como primer paso hacia el alumbramiento del Proceso Unificado, nos remontaremos a 1967 para esbozar los logros de Ericsson.

Ericsson modelaba el sistema entero como un conjunto de bloques interconectados. Después ensamblaba los bloques de más bajo nivel en subsistemas de más alto nivel y para hacer el sistema más manejable, identificaba los bloques estudiando los casos de negocios, hoy conocidos como **casos de uso** previamente especificados. Para cada caso de uso identificaba los bloques que debían cooperar para realizarlo. Con el conocimiento de las responsabilidades de cada bloque preparaba su especificación. Sus actividades de diseño producían un conjunto de diagramas de bloques estáticos con sus interfaces, agrupados en subsistemas. Estos diagramas de bloques corresponden directamente con una versión simplificada de los diagramas de clases o paquetes de UML simplificados en que sólo mostraban las asociaciones que utilizaban para comunicarse.

El primer producto de las actividades de diseño era una descripción de arquitectura software. Se basaba en la comprensión de los requisitos mas críticos y describía brevemente cada bloque y su agrupamiento en subsistemas. Un conjunto de diagramas de bloques describían a los bloques y a sus interconexiones. Sobre las interconexiones se comunicaban señales, es decir, un tipo de mensaje. Todos los mensajes quedaban descritos, uno por uno, en una biblioteca de mensaje. La descripción de la arquitectura de software y la biblioteca de mensajes eran los documentos fundamentales que guiaban el trabajo de desarrollo, pero también se utilizaban para presentar el sistema a los clientes. En aquellos momentos (1968) los clientes no estaban acostumbrados a que les presentasen los productos software por medios similares a los datos de proyectos de ingeniería.

Para cada caso de uso, los ingenieros preparaban bien un diagrama de secuencia o bien diagrama de colaboración (hoy desarrollados adicionalmente en UML). Estos diagramas mostraban como los bloques se comunicaban dinámicamente para llevar a cabo el caso de uso. Preparaban una especificación en forma de grafo de estados y un grafo de transición de estados (una versión simplificada de los diagramas de actividad de UML). Este método el diseñar a partir de bloques con interfaces bien definidas, fue la clave del éxito. De este modo, se podía crear una nueva configuración del sistema – por ejemplo, para un cliente nuevo – intercambiando un bloque por otro que proporcionase las mismas interfaces. Por tanto los bloques no eran tan sólo subsistemas y componentes de código fuente, se compilaban en bloques ejecutables, se instalaban en la máquina destino uno por uno, y se comprobaba que funcionaban con el resto de los bloques ejecutables. Es mas, debía ser posible instalar cada

bloque ejecutable, nuevo o modificado, sobre la marcha en un sistema en ejecución, mientras éste gestionaba llamadas de sistemas de telefonía en operación durante el 100 por ciento del tiempo. Sería como cambiar las ruedas de un coche que circula a 100 kilómetros por hora.

En esencia, el método que utilizaban era el que hoy conocemos como desarrollo basado en componentes. Ivar Jacobson fue el creador de este método de desarrollo. El dirigió su evolución hacia el proceso de desarrollo de software durante muchos años en el periodo anterior a Objectory [Booch Grady, et al, 2000 (a)].

EL LENGUAJE DE DESCRIPCIÓN Y ESPECIFICACIÓN

Un avance significativo durante este periodo fue la publicación en 1976 por parte del CCITT, el Organismo Internacional para la Estandarización en el área de las telecomunicaciones, del **Lenguaje de Especificación y Descripción** (Specification and Description Language, SDL) para el comportamiento funcional de los sistemas de telecomunicaciones. Este estándar, influenciado significativamente por el método de Ericsson, especificaba un sistema como un conjunto de bloques interconectados que se comunicaban unos con otros únicamente a través de mensajes (llamados señales en el estándar). Cada bloque poseía un conjunto de "procesos", que era el termino SDL para designar las clases activas. Un proceso poseía instancias de manera muy parecida a como lo hacen las clases en términos orientados a objetos. Las instancias de los procesos interactuaban mediante mensajes. SDL proponía diagramas que eran especializaciones de lo que hoy UML llama diagramas de clases, diagramas de actividades, diagramas de colaboración y diagramas de secuencia.

Por tanto, SDL era un estándar de modelado de objetos especializado. Se actualiza periódicamente, y todavía lo utilizan más de 10,000 desarrolladores y cuenta con el soporte de varios fabricantes de herramientas. Fue desarrollado inicialmente hace veinte años, y estaba muy por delante de su tiempo. Sin embargo, se desarrolló en un momento en el cual el modelado de objetos no había madurado. Probablemente, SDL será sustituido por UML, que se estandarizó en 1997 [Booch Grady, et al, 2000 (a)].

OBJECTORY

En 1987 Ivar Jacobson dejó Ericsson y fundó Objectory AB en Estocolmo. Durante los siguientes ocho años, él y sus colaboradores desarrollaron un proceso denominado Objectory ("Objectory Factory", fabrica de objetos). Extendieron su uso en otras industrias además de las telecomunicaciones, y en otros países aparte de Suecia.

Aunque el concepto de caso de uso había estado presente en el trabajo de Ericsson, ahora se le había dado un nombre, se había desarrollado una técnica de representación, y la idea se había ampliado para abarcar una variedad de aplicaciones. Es decir, los casos de uso que dirigían el desarrollo se hicieron más claros. Dicho de otro modo, la arquitectura que conduce a los desarrolladores e informa a los usuarios comenzó a destacar.

Los flujos de trabajo sucesivos se representaron en una serie de modelos: requisitos – casos de uso, análisis, diseño implementación y prueba. Un modelo es una perspectiva del sistema. Las relaciones entre los modelos de esta serie eran importantes para los desarrolladores como forma de hacer el seguimiento de una característica de un extremo a otro de la serie de

modelos. De hecho, la trazabilidad se convirtió en un prerequisite del desarrollo dirigido por casos de uso. Los desarrolladores podían seguir la traza de un caso de uso a través de la secuencia de modelos hasta el código fuente o bien, cuando surgían problemas, de volver hacia atrás.

El desarrollo del proceso Objectory continuó en una serie de versiones, desde Objectory 1.0 en 1988 a la primera versión interactiva, Objectory 3.8 en 1995.

Es importante hacer notar que el propio producto Objectory llegó a ser visto como un sistema. Esa forma de describir un proceso – como un producto de forma de sistema – proporcionaba una manera mejor de desarrollar una nueva versión de Objectory a partir de una anterior. Este modo de desarrollar Objectory hizo más fácil el ajustarlo para cubrir las necesidades específicas de diferentes organizaciones de desarrollo. El hecho de que el propio proceso de desarrollo de software Objectory era un producto de ingeniería era una característica única.

La experiencia en el desarrollo de Objectory también aportó ideas sobre cómo diseñar los procesos generales sobre los cuales opera un negocio.

EI METODO RATIONAL

Rational Software Corporation compró Objectory AB a finales de 1995 y la tarea de unificar los principios básicos subyacentes en los procesos de desarrollo. Rational había desarrollado algunas prácticas de desarrollo de software, la mayoría de ellas complementarias a las contenidas en Objectory.

Muchos libros, artículos y documentos internos detallan los desarrollos de Rational desde 1981, pero quizá las dos contribuciones más importantes al proceso fueron los énfasis en la arquitectura y en el desarrollo iterativo. Por ejemplo, en 1990, Mike Devlin, escribió un artículo introductorio sobre un proceso de desarrollo iterativo dirigido por la arquitectura. Philippe Kruchten, a cargo de la división de Prácticas de Arquitectura en Rational, firmó artículos sobre la iteración y la arquitectura [Rumbaugh James, et al, 2000 (a)].

Un artículo sobre una representación de la arquitectura con cuatro vistas: la vista lógica, la vista de procesos, la vista física y la vista de desarrollo, más una vista adicional que ilustraba las primeras cuatro vistas mediante casos de uso o escenarios. La idea de tener un conjunto de vistas, en lugar de tratar de meter todo en un único tipo de diagrama, nació de la experiencia de Kruchten en varios proyectos grandes. Las vistas múltiples permitieron encontrar, tanto a los usuarios como a los desarrolladores, lo que necesitaban para sus diferentes objetivos con la vista adecuada.

Hay gente que percibe el desarrollo iterativo como algo caótico o anárquico. El método de cuatro fases (comienzo, elaboración, construcción y transición) se diseñó para estructurar mejor y controlar el proceso durante las iteraciones. La planificación detallada de las fases y la ordenación de las iteraciones dentro de las fases fue un esfuerzo conjunto entre Walter Royce y Rich Reitman junto con la participación continua de Grady Booch y Philippe Kruchten [Rumbaugh James, et al, 2000 (a)].

Booch estaba presente desde los principios de Rational, y en 1996 en uno de sus libros mencionó dos principios fundamentales sobre la arquitectura y la iteración:

- Un estilo de desarrollo dirigido por la arquitectura es normalmente la mejor aproximación para la creación de la mayoría de los proyectos complejos muy basados en el software.

Para que un proyecto orientado a objetos tenga éxito, debe aplicarse un proceso iterativo e incremental.

EL PROCESO OBJECTORY DE RATIONAL: 1995 -1997

En el momento de la fusión, Objectory 3.8 había demostrado que se puede crear y modelar un proceso de desarrollo software como si fuese un producto. Había diseñado la arquitectura original de un proceso de desarrollo de software, un conjunto de modelos que documentaban el resultado del proyecto. Estaba desarrollado en áreas como el modelado de casos de uso, análisis y diseño, aunque en otras áreas – gestión de requisitos aparte de los casos de uso, implementación y pruebas – no estaba tan bien desarrollado. Además decía poco sobre gestión del proyecto, gestión de la configuración, distribución, y sobre la preparación del entorno de desarrollo.

Por ello se le añadieron la experiencia y prácticas de Rational para formar el Proceso Objectory de Rational 4.1. Se hizo explícita la arquitectura en forma de una descripción de la arquitectura.

Se desarrolló una definición precisa de la arquitectura, considerada como la parte más significativa de la organización del sistema. Representaba la arquitectura como vistas arquitectónicas de los modelos. Se amplió el desarrollo iterativo, pasando de ser un concepto relativamente general a ser un método dirigido por los riesgos que consideraba la arquitectura en primer lugar.

En estos momentos, UML estaba en fase de desarrollo y se incorporó como el lenguaje de modelado del Proceso Objectory de Rational. El equipo de desarrollo del proceso liderado por Philippe Kruchten, corrigió algunas de las debilidades del ROP reforzando, por ejemplo la gestión del proyecto, basada en aportaciones de Royce [Jacobson Ivar, et al, 2000 (a)].

EL LENGUAJE UNIFICADO DE MODELADO

Era evidente desde hace algún tiempo la necesidad de un lenguaje de modelado visual y consistente, en el cual expresar los resultados de las bastantes numerosas metodologías de orientación a objetos existentes a principios de los noventa.

Durante este periodo, Grady Booch, era el autor del método Booch, y James Rumbaugh era el desarrollador principal de OMT (Object Modelling Technique) creando en el Centro de Investigación y Desarrollo de General Electric. Cuando este último se incorporó a Rational en Octubre de 1994, los dos comenzaron el trabajo de unificar sus métodos, en coordinación con muchos de los clientes de Rational. Publicaron la versión 0.8 del Método Unificado en Octubre de 1995 casi en el mismo momento de la incorporación de Ivar Jacobson a Rational.

Los tres, trabajando juntos, publicaron la versión 0.9 del Lenguaje Unificado de Modelado. El esfuerzo se amplió para incluir a otros metodologistas, así como a diversas empresas, que incluían a IBM, HP, Microsoft, cada una de las cuales contribuyó al estándar en evolución.

EL PROCESO UNIFICADO DE RATIONAL

Durante este periodo, Rational compró o se fusionó a otras empresas fabricantes de herramientas. Cada una de ellas aportó a la mezcla su experiencia en áreas del proceso que ampliaron más aún el proceso Objectory de Rational:

- Requisite Inc: Aportó su experiencia en gestión de requisitos.
- SQA Inc: Había desarrollado un proceso de prueba para acompañar a su producto de pruebas, y lo añadió a la dilatada experiencia de Rational en este campo.
- Pure-Atria: Añadió su experiencia en gestión de configuración a la de Rational.
- Performance Awareness. Añadió las pruebas de rendimiento y las de carga.
- Vigortech: Añadió las pruebas de rendimiento y las de carga.

El proceso también se amplió con un nuevo flujo de trabajos para el modelado del negocio, basado en que se utiliza para obtener los requisitos a partir de los procesos de negocios que el software va a cubrir. También se extendió con diseño de interfaces de usuario dirigido por los casos de uso.

A mediados de 1998 el Proceso Objectory de Rational se había convertido en un proceso hecho y derecho, capaz de soportar el ciclo de vida del desarrollo en su totalidad. Para ello, integraba una amplia variedad de aportaciones, no solo de los tres autores originales, sino de las muchas fuentes sobre las cuales se basaron Rational y UML. En junio, Rational publicó una nueva versión del producto, el Proceso Unificado de Rational 5.0. En ese momento, por primera vez, se pusieron a disposición del público en general muchos elementos de este proceso.

El cambio de nombre refleja al hecho de que la unificación a tenido lugar en muchas dimensiones: unificación de técnicas de desarrollo, a través del Lenguaje Unificado de Modelado, y unificación del trabajo de muchos metodologistas; no solo de Rational, sino también en las oficinas de los cientos de clientes que llevaban utilizando el proceso muchos años.

2.2 EL PROCESO UNIFICADO

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software.

El Proceso Unificado es más que un simple proceso: es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

El Proceso Unificado esta basado en componentes, lo cual quiere decir que el sistema software en construcción esta formado por componentes software interconectados a través de interfaces bien definidas.

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado para todos los esquemas de un sistema software. De hecho UML es una parte esencial del Proceso Unificado. (Sus desarrollos fueron paralelos) [Booch Grady, et al, 2000 (a)].

No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres fases claves:

- Dirigido por Casos de Uso
- Centrado en la arquitectura
- Es iterativo e incremental

DIRIGIDO POR CASOS DE USO

Para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean.

El término usuario no solo hace referencia a usuarios humanos sino a otros sistemas. En este sentido, el termino usuarios representa alguien o algo que interactúa con el sistema que estamos desarrollando. Una interacción de este tipo es un **caso de uso**.

Un Caso de Uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los Casos de Uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el **modelo de casos de uso**.

Puede decirse que una especificación funcional contesta a la pregunta ¿Qué debe hacer el sistema?. La estrategia de los casos de uso puede describirse añadiendo tres palabras al final de esta pregunta ¿para cada usuario? [Booch Grady, et al, 2000 (a)].

Basándose en el modelo de casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. Los desarrolladores revisan cada uno de los sucesivos modelos para que sean conformes al modelo de casos de uso.

CENTRADO EN LA ARQUITECTURA

El papel de la arquitectura de software es parecido al que juega la arquitectura en la construcción de edificios. El edificio se contempla desde varios puntos de vista: estructura, servicios, conducción de la calefacción, fontanería, electricidad, etc. Esto permite a un constructor ver una imagen completa antes de que comience la construcción. Análogamente, la arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción.

El concepto de arquitectura software incluye los aspectos **estáticos** y **dinámicos** más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios y los inventores, y se refleja en los casos de uso. También se ve influida por muchos otros factores, como la plataforma en la que tiene que funcionar el software (arquitectura del hardware, sistema operativo, sistema de gestión de base de datos, protocolos para comunicaciones en red), los bloques de construcción reutilizables de que se dispone (marco de trabajo para interfaces gráficas de usuario), consideraciones de implantación, sistemas heredados, y requisitos no funcionales (rendimiento y fiabilidad).

La arquitectura es una vista del diseño completo con las características más importantes resaltadas dejando los detalles de lado. Debido a que lo que es significativo depende en parte de una valoración, que a su vez, se adquiere con la experiencia, el valor de una arquitectura depende de las personas que se hayan responsabilizado de su creación. Ayuda al arquitecto a centrarse en los objetivos adecuados, como la comprensibilidad, la capacidad de adaptación al cambio, y la reutilización.

Cada producto tiene tanto una función como una forma. Ninguna es suficiente por sí misma. Estas dos fuerzas deben equilibrarse para obtener un producto con éxito: la función corresponde a los casos de uso y la forma a la arquitectura. Los casos de uso deben encajar en la arquitectura cuando estos se llevan a cabo, por otro lado la arquitectura debe permitir el desarrollo de todos los casos de uso. Los casos de uso y la arquitectura deben evolucionar en paralelo.

Las principales funciones de la arquitectura software son:

- Crear un esquema de la arquitectura, comenzando por la parte de la arquitectura que no es específica de los casos de uso (por ejemplo la plataforma). Aunque esta parte de la arquitectura es independiente de los casos de uso, el arquitecto debe poseer una comprensión general de los casos de uso antes de comenzar la creación del esquema arquitectónico.
- Trabaja con un subconjunto de los casos de uso especificados, con aquellos que representen las funciones clave del sistema en desarrollo. Cada caso de uso seleccionado se especifica en detalle y se realiza en términos de **subsistemas y componentes**.
- A medida que los casos de uso se especifican y maduran, se descubre más de la arquitectura. Esto a su vez lleva a la maduración de más casos de uso.

ITERATIVO E INCREMENTAL

El desarrollo de un producto software supone un gran esfuerzo que puede durar entre varios meses o años. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas, esto es, deben seleccionarse y ejecutarse de una forma planificada. Es por esto por lo que son mini-proyectos.

Los desarrolladores basan la selección de lo que se implementará en una iteración en dos factores. En primer lugar, la iteración trata un grupo de casos de uso que juntos amplían la utilidad del producto desarrollado. En segundo lugar, la iteración trata los riesgos más importantes. Las iteraciones sucesivas se construyen sobre los artefactos de desarrollo tal como quedaron a través del trabajo de desarrollo subsiguiente: análisis, diseño, implementación y prueba, que termina convirtiendo en código ejecutable los casos de uso que se desarrollan en la iteración. Por supuesto, un incremento no necesariamente es aditivo. Especialmente en las primeras fases del ciclo de vida, los desarrolladores pueden tener que reemplazar un diseño superficial por uno más detallado. En fases posteriores, los incrementos son típicamente aditivos.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos, los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque.

Para alcanzar el mayor grado de economía en el desarrollo un equipo de proyecto intentará seleccionar sólo las iteraciones requeridas para lograr el objetivo del proyecto. Intentará secuenciar las iteraciones en un orden lógico. Un proyecto con éxito se ejecutará de una forma directa, sólo con pequeñas desviaciones del curso que los desarrolladores planificaron inicialmente. Por supuesto, en la medida en que se añadan iteraciones o se altere el orden de las mismas por problemas inesperados, el proceso consumirá más esfuerzo y tiempo. Uno de los objetivos de la reducción del riesgo es minimizar los problemas inesperados.

Son muchos los beneficios de un proceso iterativo o controlado:

La iteración controlada reduce el costo del riesgo a los costos de un solo incremento. Si los desarrolladores tienen que repetir la iteración, la organización sólo pierde el esfuerzo mal empleado de la iteración, no el valor del producto entero.

La iteración controlada reduce el riesgo de no sacar al mercado el producto en el calendario previsto. Mediante la identificación de riesgos en fases tempranas del desarrollo, el tiempo que se gasta en resolverlos se emplea al principio de la planificación, cuando la gente está menos presionada por cumplir los plazos. En el método tradicional, en el cual los problemas complicados se revelan por primera vez en prueba del sistema, el tiempo necesario para resolverlos normalmente es mayor que el tiempo que queda en la planificación, y casi siempre obliga a retrasar la entrega.

La iteración controlada acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo, en lugar de tener un calendario largo, que se prolonga eternamente.

La iteración controlada reconoce una realidad que a menudo se ignora, que las necesidades del usuario y sus correspondientes requisitos no pueden definirse completamente al principio. Típicamente, se refinan en iteraciones sucesivas. Esta forma de operar hace más fácil la adaptación a los requisitos cambiantes.

Los desarrollos dirigidos por los casos de uso, centrados en la arquitectura, iterativos e incrementales, son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. La eliminación de una de las tres etapas reduciría drásticamente el valor del Proceso Unificado [Jacobson Ivar, et al, 2000 (a)].

LA VIDA DEL PROCESO UNIFICADO

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo con una versión del producto para los clientes.

Cada ciclo consta de cuatro fases: Inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones (Ver figura 2.2).

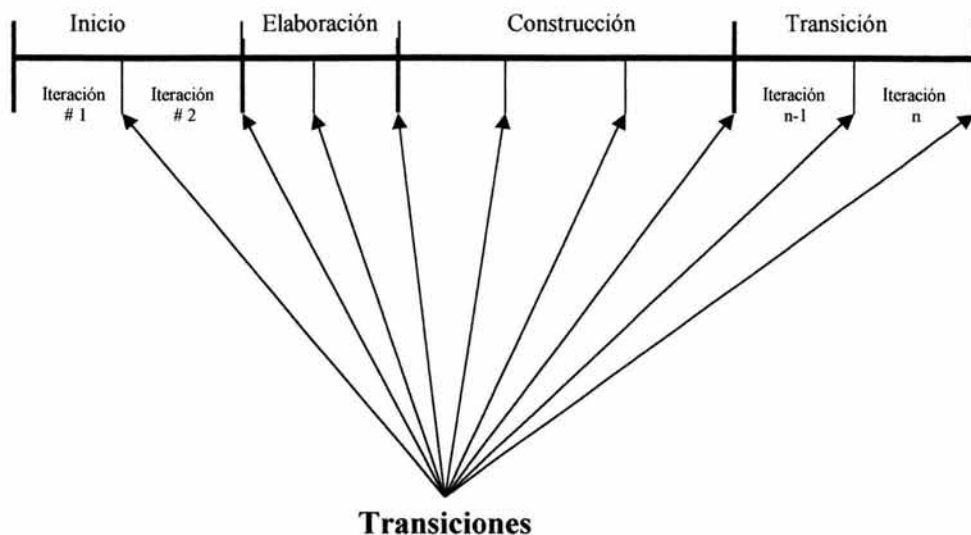


Figura 2.2
El ciclo de vida del Proceso Unificado
[Jacobson Ivar, etal, 2000 (a)]

2.3 EL PROCESO DE DESARROLLO DE SOFTWARE

Un proceso define quién está haciendo que, cuando y cómo alcanzar un determinado objetivo. En la ingeniería del software el objetivo es construir un producto software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Un proceso presenta las mejores prácticas que el estado actual de la tecnología permite. En consecuencia, reduce el riesgo y hace el proyecto más predecible. El efecto global es el fomento de una visión y una cultura comunes.

Es necesario un proceso que sirva como guía para todos los participantes clientes, usuarios, desarrolladores, y directores ejecutivos. No nos sirve ningún proceso antiguo, necesitamos uno que sea el mejor proceso que la industria pueda reunir en este punto de su historia. Por último, necesitamos un proceso que esté ampliamente disponible de forma que todos los interesados puedan comprender su papel en el desarrollo en el que se encuentra implicados.

Un proceso de desarrollo de software debería también ser capaz de evolucionar durante muchos años. Durante esta evolución debería limitar su alcance, en un momento del tiempo dado, a las realidades que permitan las tecnologías, herramientas, personas y patrones de organización.

- **Tecnologías.** El proceso debe construirse sobre las tecnologías, lenguajes de programación, sistemas operativos, computadores, estructuras de red, entornos de desarrollo, etc, disponibles en el momento en que se va a emplear el proceso. Por ejemplo hace veinte años el modelado visual no era realmente de uso general. En aquellos tiempos, un creador de un proceso prácticamente tenía que asumir que se usarían diagramas hechos a mano. Esa suposición limitaba mucho el grado en el cual el creador del proceso podía establecer el modelado dentro del proceso.
- **Herramientas.** Los procesos y las herramientas deben desarrollarse en paralelo. Las herramientas son esenciales en el proceso. Dicho de otra forma, un proceso ampliamente utilizado puede soportar la inversión necesaria para crear las herramientas que lo soporten.
- **Personas.** Un creador del proceso debe limitar el conjunto de habilidades necesarias para trabajar en el proceso a las habilidades que los desarrolladores puedan obtener rápidamente. Hoy es posible empotrar en herramientas de software técnicas que antes requerían amplios conocimientos, como la comprobación de la consistencia en los diagramas del modelo.
- **Patrones de organización.** Los desarrolladores de software no pueden ser expertos tan independientes. El creador del proceso debe adaptar el proceso a las realidades del momento hechos como las empresas virtuales, el trabajo a distancia a través de líneas de lata velocidad, la mezcla de socios de la empresa, empleados asalariados, trabajadores por obra y subcontratas de outsourcing, y la prolongada escasez de desarrolladores de software.

FORMAS EN LAS QUE EL PROCESO INTERACTUA CON LAS PERSONAS

El modo en que se organiza y gestiona un proyecto software afecta profundamente a las personas implicadas en él. Conceptos como la viabilidad, la gestión del riesgo, la organización de los equipos, la planificación del proyecto y la facilidad de comprensión del proyecto tienen un papel importante.

- **Viabilidad del proyecto.** La mayoría de la gente no disfruta trabajando en proyectos que parecen imposibles, nadie quiere hundirse con la nave. Una aproximación iterativa es el desarrollo que permite juzgar pronto la viabilidad del proyecto. Los proyectos que no son viables pueden detenerse en una fase temprana, aliviando así los problemas de moral.
- **Gestión del riesgo.** De igual forma, cuando la gente siente que los riesgos no han sido analizados y reducidos, se sienten incómodos. La exploración de los riesgos significativos en las primeras fases atenúa este problema.
- **Estructura de los equipos.** La gente trabaja de manera más eficaz en grupos pequeños de seis a ocho miembros. Un proceso que produce trabajo significativo para grupos pequeños como el análisis de un determinado riesgo, el desarrollo de un subsistema ó el llevar a cabo una iteración, proporciona esta oportunidad. Una buena arquitectura, con interfaces bien definidas entres subsistemas y componentes hace posible una división del refuerzo de este tipo.
- **Planificación del proyecto.** Cuando la gente cree que la planificación de un proyecto no es realista, la moral se hunde, la gente no quiere trabajar a sabiendas de que por mucho que lo intenten, nunca podrán obtener los resultados deseados. Las técnicas que se utilizan en las fases de inicio y de elaboración permiten a los desarrolladores tener una buena versión del producto. Debido a que hay una sensación adecuada de lo que debería hacer una definición del tiempo necesario para lograr los objetivos, atenuando el problema del "nunca acabaremos".
- **Facilidad de comprensión del proyecto.** A la gente le gusta saber lo que está haciendo, quieren tener una comprensión global. La descripción de la arquitectura proporciona una visión general para cualquiera que se encuentre implicado en el proyecto.
- **Sensación de cumplimiento.** En un ciclo de vida iterativo, la gente recibe retroalimentación frecuentemente, lo cual a su vez, hace llegar a conclusiones. La retroalimentación frecuente y las conclusiones obtenidas aceleran el ritmo del trabajo. Un ritmo de trabajo rápido combinado con conclusiones frecuentes aumenta la sensación de progreso de la gente.

CAPÍTULO III

MODELO DE CASOS DE USO Y
MODELO DE ANÁLISIS DEL
NUEVO SISTEMA INTEGRAL DE
SOPORTE TÉCNICO DE LA
DELEGACIÓN GUSTAVO A. MADERO

3.1 EL MODELO DE CASOS DE USO

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.

Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones.

El modelo de casos de uso puede hacerse bastante grande y difícil de digerir de un solo mordisco, de forma que es necesario algún medio de abordarlo en trozos más pequeños. UML nos permite presentar el modelo en diagramas que muestran los actores y los casos de uso desde diferentes puntos de vista y con diferentes propósitos. Si el modelo de casos de uso es grande, es decir, si contiene un gran número de casos de uso y actores, también puede ser útil introducir paquetes en el modelo para tratar su tamaño.

El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario. Cada uno de éstos se representa mediante uno o más actores. También se representa mediante uno o más actores cada sistema externo con el que interactúa el sistema, incluyendo dispositivos externos, que se consideran ajenos al sistema. Por tanto, los actores representan terceros fuera del sistema que colaboran con el sistema. Una vez que hemos identificado todos los actores del sistema tenemos identificado el entorno externo al sistema.

Un actor juega un papel por cada caso de uso con el que colabora. Cada vez que un actor interactúa con el sistema, la instancia correspondiente del actor está desarrollando ese papel. Una instancia de un actor es por tanto un usuario concreto que interactúa con el sistema. Cualquier entidad que se ajuste a un actor puede actuar como una instancia del actor.

Cada forma en que los actores usan el sistema se representa con un caso de uso. Los casos de uso son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia.

Por tanto, un caso de uso es una especificación. Especifica el comportamiento de cosas dinámicas, en este caso, de instancias de los casos de uso.

Dentro de UML, un caso de uso es un clasificador, lo cual quiere decir que tiene operaciones y atributos. Una descripción de un caso de uso puede por tanto incluir diagramas de estados, diagramas de actividad, colaboración, y diagramas de secuencia.

Una instancia de caso de uso es la realización de un caso de uso. Otra forma de decirlo es que una instancia de un caso de uso es lo que el sistema lleva a cabo cuando obedece a un caso de uso. Cuando se hace referencia a una instancia de un caso de uso, ésta interactúa con instancias de actores, y ejecuta una secuencia de acciones según se especifica en el caso de uso. Esta secuencia se especifica en un diagrama de estados o

diagramas de actividad, es un camino largo del caso de uso. Puede haber muchos caminos, y muchos de ellos pueden ser muy parecidos. Estas son alternativas de la secuencia de acciones para el caso de uso. Un camino como éste a través de n-casos de uso puede ser algo parecido a lo siguiente:

1. La instancia del caso de uso se inicia y pasa a estado de comienzo.
2. El caso de uso es invocado por un mensaje externo de un actor.
3. Transita a otro estado realizando una secuencia de acciones. Una secuencia de este tipo contiene cálculos internos, selección del camino y mensaje de salida.
4. Queda a la espera de otro mensaje externo de un actor.
5. Es invocado por un nuevo mensaje, etc. Esto puede continuar sobre muchos estados hasta que se termina la instancia del caso de uso.

La mayoría de las veces un actor invoca a la instancia del caso de uso, como se ha descrito, pero también puede ser un evento interno al sistema el que invoque a la instancia.

Los casos de uso como todos los clasificadores tienen atributos. Estos atributos representan los valores que una instancia de un caso de uso utiliza y manipula durante la ejecución de su caso de uso. Estos valores son locales a la instancia del caso de uso, es decir, no pueden ser utilizados por otras instancias del caso de uso.

La instancia de los casos de uso no interactúa con otras instancias de casos de uso. El único tipo de interacción en el modelo de uso tiene lugar entre instancias de actores e instancias de casos de uso. El motivo es que queremos que el modelo de casos de uso sea simple e intuitivo para permitir discusiones fructíferas con los usuarios finales y otros interesados sin quedar atrapados en los detalles. En el modelo de casos de uso no queremos tratar con interfaces entre casos de uso, concurrencia y otros conflictos entre instancias de uso, distintas. Consideramos atómicas las instancias de casos de uso, es decir, cada una de ellas se ejecuta por completo o no se ejecuta, sin interferencias por parte de otras instancias de casos de uso. Por lo tanto, el comportamiento de cada caso de uso puede interpretarse independientemente del los otros casos de uso, lo cual hace más sencillo el modelado de casos de uso.

El flujo de sucesos para cada caso de uso puede plasmarse como una descripción textual de la secuencia de acciones del caso de uso. Por tanto, el flujo de sucesos especifica lo que el sistema hace cuando se lleva a cabo el caso de uso especificado. El flujo de sucesos también especifica cómo interactúa el sistema con los actores cuando se lleva a cabo el caso de uso.

Desde la perspectiva de la gestión, una descripción de un flujo de sucesos incluye un conjunto de secuencias de acciones que pueden ser modificadas, revisadas, diseñadas, implementadas y probadas junto, que pueden ser descritas como una sección o subsección del manual del usuario.

Para poder describir en su totalidad el modelo de casos de uso, preparamos diagramas y descripciones, especialmente para observar cómo se relacionan los casos de uso entre sí y con los actores.

No hay una regla estricta sobre lo que se debe incluir en el diagrama. De hecho, elegimos el conjunto de diagramas que describan más claramente el sistema.

Para asegurar la consistencia cuando se describen varios casos de uso concurrentemente, resulta práctico desarrollar un glosario de términos [Jacobson Ivar, et al, 2000 (a)].

DESCRIPCIONES QUE DEBE INCLUIR UN CASO DE USO

Una descripción de un caso de uso debe incluir lo siguiente:

- La descripción de un caso de uso debe definir el estado inicial como precondition.
- Como y cuándo comienza el caso de uso.
- El orden requerido en el que las acciones se deben ejecutar. En este punto el orden se define por una secuencia numerada.
- Cómo y cuándo terminan los casos de uso.
- Una descripción de caso de uso debe definir los posibles estados finales como postcondiciones.
- Los caminos de ejecución que no están permitidos (el camino que el usuario no puede tomar).
- Las descripciones de caminos alternativos que están incluidos junto con la descripción del camino básico.
- La descripción de los caminos alternativos que han sido extraídas de la descripción del camino básico.
- La interacción del sistema con los actores y qué cambios producen.
- La utilización de objetos, valores y recursos de sistema.
- Describir explícitamente qué hace el sistema.

Los atributos de los casos de uso pueden utilizarse como inspiración para encontrar más tarde clases y atributos en el análisis y diseño.

Pueden utilizarse los diagramas de estado de UML para describir los estados de los casos de uso y sus transiciones entre esos estados.

Pueden utilizarse los diagramas de actividad para describir las transiciones entre estados con más detalle como secuencia de acciones. Los diagramas de actividad pueden describirse como una generalización de los diagramas de transición de estados.

Se pueden utilizar los diagramas de interacción para describir cómo interactúa una instancia de caso de uso con la instancia de un actor. Los diagramas de interacción, entonces muestran el caso de uso y el actor participante [Jacobson Ivar, et al, 2000 (a)].

En resumen el modelo de casos de uso se estructura para:

- Extraer descripciones de funcionalidad generales y compartidas que pueden ser utilizadas por descripciones más específicas.
- Extraer descripciones de funcionalidad adicionales u opcionales que pueden extender descripciones más específicas.

A continuación se especificarán los requisitos funcionales.

REQUERIMIENTOS DE LA UNIDAD DEPARTAMENTAL DE SOPORTE TÉCNICO

1. Para el mejor desempeño de sus labores, la Jefatura de Soporte Técnico tiene la necesidad de la creación del catálogo de Unidades Departamentales, en el que se requiere contenga las siguientes descripciones:
 - Clave Departamental
 - Descripción del Área
 - Nombre del Funcionario
 - Título del Funcionario
 - Correo Electrónico
 - Teléfono
 - Número de Extensión
 - Fax
 - Domicilio de la Unidad Departamental
 - Colonia
 - Código Postal
2. Para el manejo del equipo informático, es necesaria la creación de los siguientes catálogos:
 - Catálogo de Equipos
 - Catálogo de Marcas
 - Catálogo de Modelos
3. Dentro del catálogo de tipos o equipos informáticos se requieren los siguientes datos
 - Identificador único del equipo
 - Descripción del equipo
4. Dentro del catálogo de marcas de equipo informático se requieren los siguientes datos
 - Identificador único de la marca
 - Descripción de la marca
5. Dentro del catálogo de modelos de equipo informático se requieren los siguientes datos
 - Identificador único del modelo
 - Descripción del modelo
6. Para poder llevar un control estricto sobre el equipo asignado a las diferentes Unidades Departamentales de la Delegación Gustavo A. Madero, se necesita la formación del catálogo de equipo asignado por área, requiriendo los siguientes datos:
 - Clave Departamental
 - Teléfono
 - Extensión
 - Tipo de equipo

- Marca
- Modelo
- Número de Serie
- Inicio de Garantía
- Término de Garantía

REQUERIMIENTOS PARA ELABORAR Y TRABAJAR LOS REPORTES DE SERVICIO DE SOPORTE TÉCNICO

1. La Jefatura de Soporte Técnico elabora reportes de control de servicios efectuados (Hardware o Software) a cualquier Unidad Departamental adscrita a la Delegación Gustavo A. Madero.
2. Para elaborar los reportes de servicio, la Jefatura de Soporte Técnico requiere los siguientes datos de la Unidad Departamental solicitante:
 - Nombre del Funcionario que Solicita el Servicio.
 - Puesto del Funcionario que Solicita el Servicio.
 - Unidad Departamental a la que esta Adscrito el Solicitante.
 - Teléfono de la Unidad Departamental.
 - Extensión de la Unidad Departamental.
3. Para poder elaborar los reportes, la Jefatura de Soporte Técnico requiere los siguientes datos del equipo al que se le brindó servicio:
 - Número de Serie del Equipo.
 - Tipo de Equipo.
 - Marca del Equipo.
 - Modelo del Equipo.
 - Fecha de Inicio y Fin de Garantía del Equipo.
4. La Jefatura de Soporte Técnico agrupa el tipo de servicio en dos grupos:
 - Asistencia Técnica
 - Mantenimiento del Equipo
5. Los tipos de servicio prestado en la Asistencia Técnica, se dividen básicamente en 4 categorías:
 - Servicio de Soporte Técnico
 - Servicio de Asistencia a la Red
 - Servicio de Antivirus
 - Servicio de Respaldo
 - Servicio de Internet
6. El Mantenimiento del Equipo, se refiere a hacer válida la póliza de garantía en los centros de servicio correspondientes, y estos se dividen básicamente en 3 categorías de servicio:
 - Servicio de Hardware
 - Servicio de Software
 - Servicio de Telecomunicaciones

3.1.1 DIAGRAMAS DE CASOS DE USO

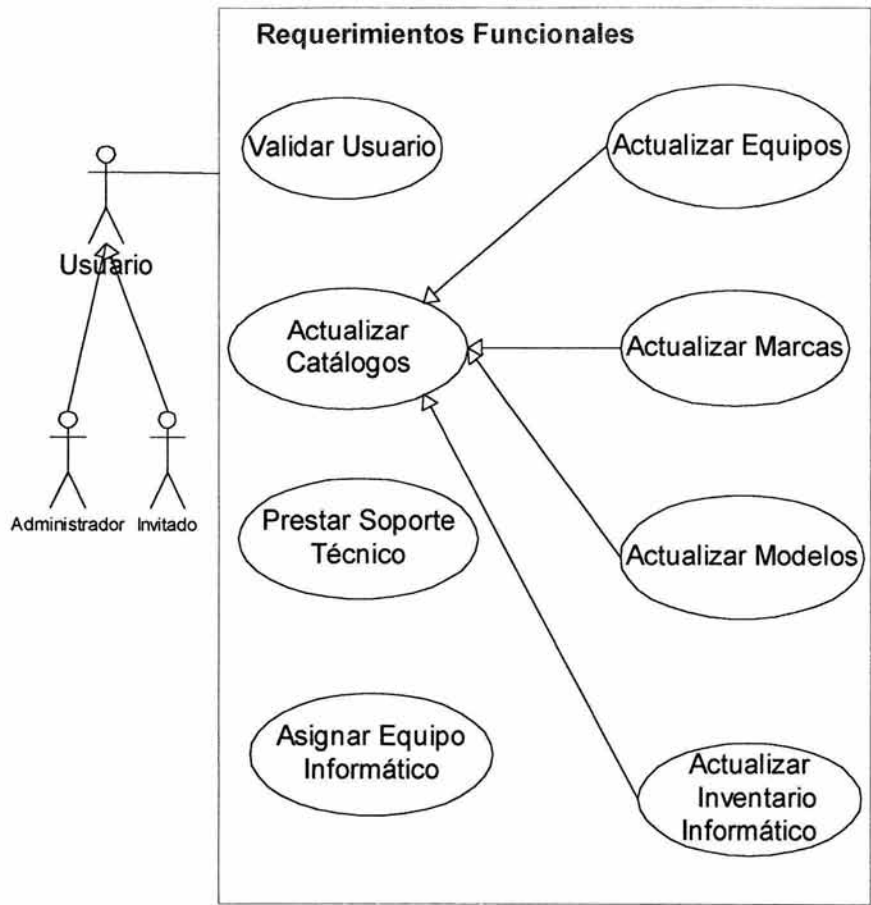
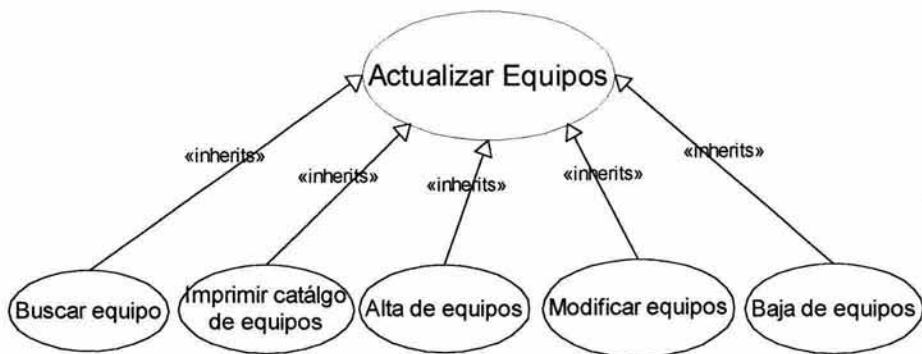


Figura 3.1.1
Contexto del Sistema

La descripción y especificaciones del modelado de los casos de uso se encuentran descritos en el capítulo 3.1.2 (página 38)

Especialización del caso de uso: Actualizar Equipos



Especialización del caso de uso: Actualizar Marcas

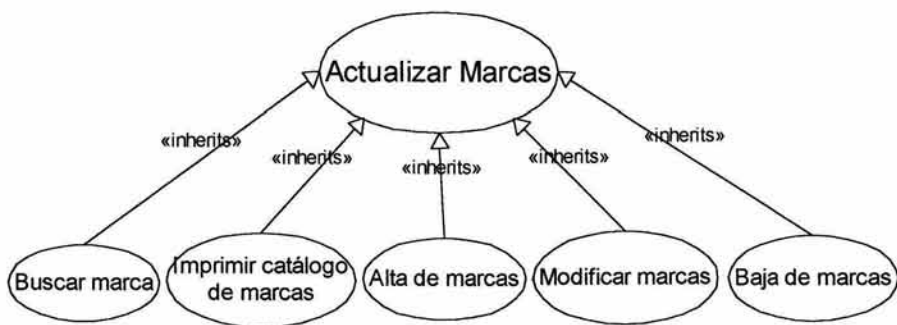
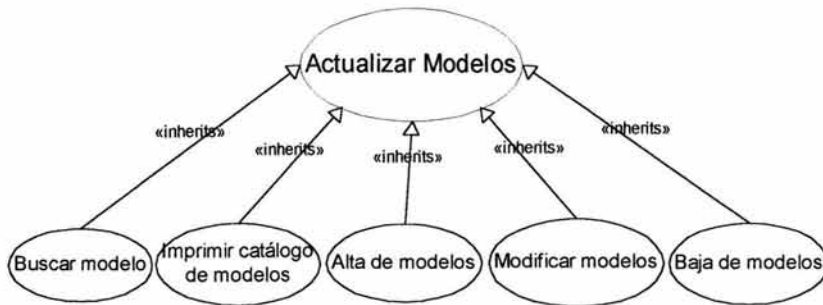


Figura 3.1.2

Los casos de uso Actualizar Equipos y Actualizar Marcas así como cada una de sus especializaciones se encuentran descritos en el capítulo 3.1.2 (páginas 38-40 y 41-42 respectivamente)

Especialización del caso de uso: Actualizar Modelos



ESPECIALIZACION DEL CASO DE USO: ACTUALIZAR INVENTARIO INFORMÁTICO

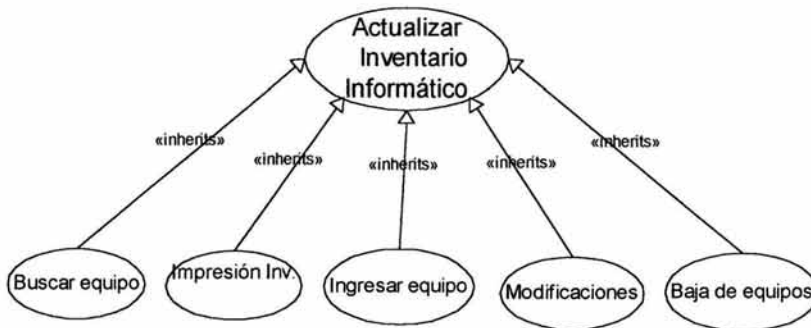


Figura 3.1.3

Los casos de uso Actualizar Modelos y Actualizar Inventario Informático así como cada una de sus especializaciones se encuentran descritos en el capítulo 3.1.2 (páginas 43-44 y 45-46 respectivamente)

Especialización del caso de uso: Asignar Equipo



Especialización del caso de uso: Prestar Soporte Técnico



Figura 3.1.4

Los casos de uso Asignar Equipo Informático y Prestar Soporte técnico así como cada una de sus especializaciones se encuentran descritos en el capítulo 3.1.2 (páginas 47-48 y 49-51 respectivamente)

3.1.2 DESCRIPCIÓN DE LOS CASOS DE USO

Tabla 3.1.2 CASO DE USO: Validar Usuario

Paso	Acción del usuario	Respuesta del sistema
1	Proporciona su identificador personal de acceso al Sistema de Control de Soporte Técnico.	Valida el identificador personal proporcionado por el usuario.
2		Permite iniciar sesión al usuario dentro del Sistema de Control de Soporte Técnico, configurando los permisos en el del panel de control principal.

Caso alternativo:

1. Si el identificador personal no es válido, el sistema envía un mensaje de error.
2. Si el identificador personal es inválido en tres intentos, el sistema es bloqueado.

Restricciones:

1. El identificador personal del usuario del sistema no puede exceder la longitud de 20 caracteres.

Tabla 3.1.3 CASO DE USO: Actualización de Catálogos

Paso	Acción del usuario	Respuesta del sistema
1	Realiza la petición de trabajo en el menú de catálogos informáticos disponibles en la Delegación.	Proporciona el acceso al panel de catálogos disponibles en la GAM.

Tabla 3.1.4 CASO DE USO: Actualizar Equipos

Paso	Acción del usuario	Respuesta del sistema
1	Realiza la petición de ingresar al catálogo de Equipos informáticos.	Permite el acceso al catálogo de Equipos.
2		Configura los permisos asignados a este usuario dentro de este catálogo

Tabla 3.1.5 Casos especializados del Caso de Uso: Actualización de Equipos

Caso de uso: Buscar Equipo

Paso	Acción del usuario	Respuesta del sistema
1	Ingresa el identificador del equipo informático que desea buscar.	Realiza la búsqueda del equipo dentro del catálogo, de acuerdo con el identificador proporcionado por el usuario.
2		Localiza el equipo especificado, y proporciona su descripción general.

Caso alternativo:

1. Si el identificador del equipo no es válido, el sistema notifica al usuario que el equipo no existe en el catálogo.

Restricciones:

1. El identificador del equipo no puede exceder los 5 caracteres.
2. El identificador es un valor numérico entero que se encuentre en el intervalo [0-99999]

Tabla 3.1.6 Caso de uso: Imprimir catalogo de equipos

Paso	Acción del usuario	Respuesta del sistema
1	Efectúa la petición de imprimir el catalogo de equipos informáticos.	Muestra en pantalla en vista preliminar el reporte del catálogo de equipos informáticos.
2	Manda imprimir el catalogo.	Envía a impresión en forma de reporte el catálogo de equipos.

Caso alterno:

1. El usuario puede cancelar el envío de impresión del catalogo.

Restricciones:

1. La impresión del catálogo en forma de reporte se efectúa en papel tamaño carta.

Tabla 3.1.7 Caso de uso: Alta de Equipos

Paso	Acción del usuario	Respuesta del sistema
1	Realizar la petición de alta en el catálogo de Equipos.	Agrega un nuevo registro al catalogo de Equipos Informáticos, y pide la confirmación del usuario.
2	Proporciona el identificador para el nuevo equipo.	
3	Proporciona la descripción general del nuevo equipo.	
4	Confirma el alta del nuevo equipo.	Verifica que los datos proporcionados por el usuario sean correctos y actualiza el catálogo de Equipos Informáticos.

Caso alterno:

1. El usuario puede cancelar la petición de alta.
2. Si los datos proporcionados por el usuario no son correctos, el sistema cancela la petición de alta del usuario.

Restricciones:

1. El identificador del nuevo equipo debe ser de tipo numérico entero con un máximo de 5 dígitos ([0-99999]).
2. El identificador y la descripción general del nuevo equipo son datos que se requieren de forma obligatoria.

Tabla 3.1.8 Caso de uso: Modificar Descripción de Equipo

Paso	Acción del usuario	Respuesta del sistema
1	Cambia la información general de un equipo en particular.	Modifica los datos proporcionados por el usuario y pide la confirmación de este.
2	Confirma las modificaciones realizadas al equipo.	Verifica que los datos sean correctos y actualiza el catálogo de Equipos

Caso alterno:

1. El usuario puede cancelar las modificaciones realizadas al equipo seleccionado.

Restricciones:

1. Si los datos modificados por el usuario no son correctos, el sistema cancela los cambios realizados por el usuario.

Tabla 3.1.9 Caso de uso: Baja de Equipos

Paso	Acción del usuario	Respuesta del sistema
1	Hace la petición para dar de baja un equipo del catálogo de Equipos Informáticos.	Pide la confirmación del usuario para dar de baja el equipo seleccionado.
2	Confirma que se lleve acabo la baja del equipo.	Realiza la baja y actualiza el catálogo de Equipos Informáticos

Caso alterno:

1. El usuario puede cancelar la petición de baja del equipo antes de que sea actualizado el sistema.

Tabla 3.1.10 CASO DE USO: Actualizar Marcas

Paso	Acción del usuario	Respuesta del sistema
1	Solicita trabajar dentro del catálogo de Marcas.	Permite el acceso al catálogo de Marcas y configura los permisos para el usuario dentro de este catálogo.

Tabla 3.1.11 Casos especializados del Caso de Uso: Actualizar Marcas**Caso de uso: Buscar Marca**

Paso	Acción del usuario	Respuesta del sistema
1	Ingresa el identificador de la marca que desea buscar	
2	Confirma que desea realizar la búsqueda del equipo.	Realiza la búsqueda de la marca, de acuerdo con el identificador proporcionado por el usuario.
3		Localiza la marca proporcionada por el usuario, obteniendo la clave del equipo, así como su descripción general.

Caso alternativo:

1. Si el identificador de la marca no es válido, el sistema notifica al usuario que la marca no existe dentro de este catálogo.
2. El usuario puede cancelar la búsqueda de la marca.

Restricciones:

1. El identificador de la marca no puede exceder los 5 caracteres.
2. El identificador es un valor numérico entero que se encuentre en el intervalo [0-99999]

Tabla 3.1.12 Caso de uso: Imprimir catálogo de marcas

Paso	Acción del usuario	Respuesta del sistema
1	Efectúa la petición de imprimir el catálogo de marcas.	Muestra en pantalla en vista preliminar el reporte del catálogo de marcas.
2	Manda imprimir el catálogo.	Envía a impresión en forma de reporte el catálogo de marcas.

Caso alternativo:

1. El usuario puede cancelar la impresión en forma de reporte.

Restricciones:

1. La impresión del catálogo en forma de reporte se efectúa en papel tamaño carta.

Tabla 3.1.13 Caso de uso: Alta de Marcas

Paso	Acción del usuario	Respuesta del sistema
1	Realizar petición de alta en el catálogo de Marcas.	Agrega un nuevo registro en el catálogo de Marcas, y pide la confirmación del usuario.
2	Proporciona el identificador del equipo relacionado con la nueva marca.	
3	Proporciona el identificador de la nueva	

	marca.	
4	Proporciona la descripción general de la marca.	
5	Confirma el alta de la nueva marca.	Verifica que los datos proporcionados por el usuario sean correctos, y actualiza el catalogo de Marcas.

Caso alterno:

1. El usuario puede cancelar la petición de alta.
2. Si los datos proporcionados por el usuario no son correctos el sistema cancela el alta.

Restricciones:

1. El identificador de la nueva marca, debe ser de tipo numérico entero, con un máximo de 5 dígitos, contenido en el intervalo [0-99999].
2. El identificador de la nueva marca, debe tener relacionado un equipo dentro del catálogo de equipos informáticos.
3. El identificador del equipo, identificador de la marca y la descripción general de la nueva marca son datos de carácter obligatorio.

Tabla 3.1.14 Caso de uso: Modificar Descripción de Marca

Paso	Acción del usuario	Respuesta del sistema
1	Cambia la información general de una Marca en particular.	Modifica, los datos proporcionados por el usuario y pide la confirmación de este.
2	Confirma que se efectúen las modificaciones realizadas a la marca.	Verifica que los datos modificados por el usuario sean correctos y actualiza el catálogo de Marcas.

Caso alterno:

1. El usuario puede cancelar las modificaciones realizadas a la marca seleccionada.

Restricciones:

1. Si los datos modificados por el usuario no son validos, el sistema cancela los cambios realizadas por el usuario.

Tabla 3.1.15 Caso de uso: Baja de Marcas

Paso	Acción del usuario	Respuesta del sistema
1	Hace la petición para dar de baja una marca del catálogo de Marcas.	Pide la confirmación del usuario para realizar la baja
2	Confirma que se efectúe la baja de la marca.	Realiza la baja y actualiza el catálogo de Marcas.

Caso alterno:

1. El usuario, puede cancelar la baja de la marca dentro del catálogo de Marcas.

Tabla 3.1.16 CASO DE USO: Actualizar Modelos

Paso	Acción del usuario	Respuesta del sistema
1	Solicita trabajar dentro del catálogo de Modelos.	Permite el acceso al catálogo de Modelos y configura los permisos para el usuario.

Tabla 3.1.17 Casos especializados del Caso de Uso: Actualizar Modelos
Caso de uso: Buscar Modelo

Paso	Acción del usuario	Respuesta del sistema
1	Ingresa el identificador del modelo que desea buscar.	
2	Confirma que se lleve a cabo la búsqueda.	Realiza búsqueda conforme al identificador de modelo proporcionado por el usuario.
3		Localiza el modelo mostrando una descripción general de este.

Caso alternativo:

1. Si el identificador del modelo no es válido, el sistema notifica al usuario que el modelo no existe dentro de este catálogo.

Restricciones:

1. El identificador del modelo no puede exceder los 5 caracteres.
2. El identificador del modelo es un valor numérico que se encuentre en el intervalo [0-99999]

Tabla 3.1.18 Caso de uso: Imprimir catálogo de modelos

Paso	Acción del usuario	Respuesta del sistema
1	Efectúa la petición de imprimir el catálogo de modelos informáticos.	Muestra en pantalla en vista preliminar el reporte del catálogo de modelos informáticos.
2	Manda imprimir el catálogo.	Envía a impresión en forma de reporte el catálogo de modelos.

Caso alternativo:

1. El usuario puede cancelar la impresión en forma de reporte.

Restricciones:

1. La impresión del catálogo se lleva a cabo en papel tamaño carta.

Tabla 3.1.19 Caso de uso: Alta de Modelos

Paso	Acción del usuario	Respuesta del sistema
1	Realizar la petición de alta dentro del catálogo de Modelos.	Agrega un nuevo registro en el catálogo de Modelos y pide la confirmación del usuario.
2	Proporciona identificador del nuevo modelo	
3	Proporciona la descripción general del	

	nuevo modelo.	
4	Confirma que se realice el alta.	Verifica que los datos proporcionados por el usuario sean correctos, y actualiza.

Caso alternativo:

1. El usuario puede cancelar la nueva alta realizada.

Restricciones:

1. El identificador de la nuevo modelo debe ser de tipo numérico, con un máximo de 5 dígitos dentro del intervalo [0-99999].
2. El identificador y la descripción del nuevo modelo son datos que el sistema requiere de forma obligatoria.
3. Si los datos proporcionados por el usuario no son correctos, el sistema cancela la petición de alta del usuario.

Tabla 3.1.20 Caso de uso: Modificar Descripción de Modelo

Paso	Acción del usuario	Respuesta del sistema
1	Cambia la información general de un Modelo en particular.	Modifica, los datos proporcionados por el usuario y pide la confirmación de éste.
2	Confirma las modificaciones realizadas al modelo.	Verifica que los datos ingresados por el usuarios sean correctos, y actualiza el catálogo de Modelos.

Caso alternativo:

1. El usuario puede cancelar las modificaciones realizadas al Modelo seleccionado.

Restricciones:

1. Si los datos modificados por el usuario no son correctos, el sistema cancela los cambios efectuados por el usuario.

Tabla 3.1.21 Caso de uso: Baja de Modelos

Paso	Acción del usuario	Respuesta del sistema
1	Hace la petición para dar de baja un modelo localizado dentro del catálogo.	Pide la confirmación del usuario para dar de baja el modelo.
2	Confirma que se lleve acabo la baja del modelo.	Realiza la baja y actualiza el catálogo de Modelos.

Caso alternativo:

1. El usuario puede cancelar la baja de la modelo dentro del catálogo de Modelos.

Tabla 3.1.22 CASO DE USO: Actualizar Inventario Informático

Paso	Acción del usuario	Respuesta del sistema
1	Realiza la petición de trabajo dentro del Inventario del equipo Informático de la GAM.	Permite el acceso al Inventario Informático de la Delegación y configura los permisos correspondientes.

Tabla 3.1.23 Casos especializados del Caso de Uso: Actualizar Inventario Informático
Caso de uso: Buscar Equipo

Paso	Acción del usuario	Respuesta del sistema
1	Ingresa el número de serie del equipo informático.	
2	Confirma que se realice la búsqueda dentro del catálogo.	Realiza una búsqueda del equipo, de acuerdo con el número de serie proporcionado por el usuario.
3		Localiza el número de serie y muestra una descripción general del equipo correspondiente.

Caso alterno:

1. Si el número de serie no es válido, el sistema notifica al usuario que el equipo no existe dentro del Inventario Informático de la Delegación Gustavo A. Madero.

Tabla 3.1.24 Caso de uso: Imprimir inventario informático

Paso	Acción del usuario	Respuesta del sistema
1	Efectúa la petición de imprimir el inventario informático.	Muestra en pantalla en vista preliminar el reporte del inventario informático de la GAM.
2	Manda imprimir el catálogo.	Envía a impresión en forma de reporte el inventario.

Caso alterno:

1. El usuario puede cancelar la impresión en forma de reporte.

Restricciones:

1. La impresión del inventario informático se efectúa en el formato correspondiente.

Tabla 3.1.25 Caso de uso: Ingresar Equipo informático

Paso	Acción del usuario	Respuesta del sistema
1	Ingresa un nuevo equipo al Inventario del equipo Informático de la GAM.	Agrega el nuevo registro en el Inventario del equipo Informático de la Delegación Gustavo A. Madero y pide la confirmación del usuario.
2	Selecciona el tipo de equipo que desea agregar al inventario informático.	
3	Selecciona la marca correspondiente el nuevo equipo.	

4	Selecciona el modelo correspondiente a la marca del equipo.	
5	Ingresa el número de serie.	
6	Ingresa la fecha de inicio de garantía y la fecha de término de garantía del nuevo equipo	
7	Confirma que se realice el alta correspondiente.	Verifica que los datos proporcionados por el usuario sean válidos y actualiza el Inventario Informático de la Delegación.

Caso alternativo:

1. El usuario puede cancelar la nueva alta realizada.
2. Si los datos proporcionados por el usuario no son correctos, el sistema cancela la petición de alta en el inventario.

Restricciones:

1. El nuevo número de serie proporcionado por el usuario no puede contener un valor duplicado, correspondiente a otro equipo dentro del inventario informático.

Tabla 3.1.26 Caso de uso: Modificar Descripción del Equipo

Paso	Acción del usuario	Respuesta del sistema
1	Cambia la información general de un número de serie correspondiente algún equipo en particular.	Modifica, los datos proporcionados por el usuario y pide la confirmación de este.
2	Confirma las modificaciones realizadas al número de serie.	Verifica que los datos modificados por el usuario sean correctos y actualiza el Inventario de Equipo Informático de la Delegación.

Caso alternativo:

1. El usuario puede cancelar las modificaciones realizadas al número de serie seleccionado.

Restricciones:

1. Si los datos modificados por el usuario no son válidos, el sistema cancela los cambios realizados por el usuario.

Tabla 3.1.27 Caso de uso: Baja de Equipos en Inventario

Paso	Acción del usuario	Respuesta del sistema
1	Hace la petición para dar de baja un número de serie en el inventario de equipo informático.	Pide la confirmación del usuario para dar de baja el número de serie.
2	Confirma que se lleve a cabo la baja del número de serie.	Realiza la baja y actualiza el Inventario de equipo informático.

Caso alternativo:

1. El usuario, puede cancelar la baja de la marca dentro del catálogo de Marcas.

Tabla 3.1.28 CASO DE USO: Asignar equipo informático

Paso	Acción del usuario	Respuesta del sistema
1	Solicita trabajar dentro de los equipos informáticos asignados a las diversas Unidades Departamentales de la GAM.	Permite el acceso al equipo asignado dentro de la Delegación.

Tabla 3.1.29 Casos especializados del Caso de Uso: Asignar equipo informático
Caso de uso: Buscar equipo asignado

Paso	Acción del usuario	Respuesta del sistema
1	Ingresa el número de serie del equipo informático.	
2	Confirma que se lleve a cabo la búsqueda del número de serie asignado por área.	Realiza una búsqueda del equipo, de acuerdo con el número de serie proporcionado por el usuario.
3		Localiza el número de serie, obteniendo la descripción general de dicho equipo, así como a la Unidad Departamental a la que este se encuentra asignado.

Caso alterno:

1. Si el número de serie no es válido, el sistema notifica al usuario que el equipo no existe dentro del Inventario Informático, o aún no a sido asignado a ninguna Unidad Departamental de la Delegación Gustavo A. Madero.

Tabla 3.1.30 Caso de uso: Imprimir Equipo Asignado

Paso	Acción del usuario	Respuesta del sistema
1	Selecciona la Unidad Departamental de la cual desea imprimir sus equipos asignados.	
2	Realiza la petición de impresión.	Muestra en pantalla en vista preliminar el reporte del equipo asignado a la Unidad Departamental seleccionada por el usuario.
3	Realiza la confirmación de la impresión.	Envía a impresión en forma de reporte el equipo asignado del área seleccionada.

Caso alterno:

1. El usuario puede cancelar la impresión en forma de reporte

Restricciones:

1. La impresión del reporte se efectúa en papel tamaño carta.

Tabla 3.1.31 Caso de uso: Asignar equipo

Paso	Acción del usuario	Respuesta del sistema
1	Asigna un equipo a una Unidad Departamental en particular.	Agrega un nuevo registro dentro de los equipos asignados por Unidad

		Departamental, y pide la confirmación del usuario.
2	Selecciona el Departamento al cual será asignado el equipo.	
3	Selecciona el número de serie del equipo que desea asignar a la Unidad Departamental.	
4		Muestra el tipo de equipo, marca y modelo, así como las fechas de inicio y fin de garantía relacionados con el número de serie seleccionado por el usuario.
5	Confirma que se asigne el equipo al Departamento seleccionado.	Verifica que los datos proporcionados por el usuario sean correctos y asigna el equipo al área correspondiente.

Caso alterno:

1. El usuario puede cancelar la asignación del equipo.

Restricciones:

1. Un equipo no puede ser asignado a más de una Unidad Departamental de la Delegación.
2. Si los datos proporcionados por el usuario no son correctos, el sistema cancela la petición de alta del usuario.

Tabla 3.1.32 Caso de uso: Modificar Equipo asignado

Paso	Acción del usuario	Respuesta del sistema
1	Cambia la información general de un número de serie correspondiente algún equipo que a sido asignado.	Modifica, los datos proporcionados por el usuario y pide la confirmación de este.
2	Confirma los cambios realizados al número de serie.	Actualiza el Inventario de Equipo Informático asignado por Unidad Departamental de la Delegación.

Caso alterno:

1. El usuario puede cancelar los cambios efectuados al número de serie seleccionado.

Restricciones:

1. Si los datos modificados por el usuario no son válidos, el sistema cancela los cambios efectuados.

Tabla 3.1.33 Caso de uso: Eliminar Equipo asignado

Paso	Acción del usuario	Respuesta del sistema
1	Hace la petición para dar de baja un número de serie asignado por Unidad Departamental.	Pide la confirmación del usuario para quitar el resguardo del número de serie.
2	Confirma que se lleve a cabo la baja del número de serie proporcionado.	Quita el resguardo al área, y actualiza el Inventario de equipo informático.

Caso alternativo:

1. El usuario puede cancelar la baja del resguardo del equipo a la Unidad Departamental.

Tabla 3.1.34 CASO DE USO: Prestar Soporte Técnico

Paso	Acción del usuario	Respuesta del sistema
1	Hace la petición de trabajo dentro del control de servicio del área de Soporte Técnico.	Permite el acceso al control del área de Soporte Técnico, configurando los permisos para el usuario.

Tabla 3.1.35 Especializaciones del Caso de Uso: Prestar Soporte Técnico**Caso de uso: Buscar reporte de servicio**

Paso	Acción del usuario	Respuesta del sistema
1	Proporciona el número de reporte que desea buscar.	
2	Confirma que se efectúe la búsqueda del número de reporte.	Busca el reporte de servicio de soporte técnico.
3		Localiza el reporte y visualiza los datos generales de la asistencia técnica proporcionada.

Caso alternativo:

1. Si el sistema no localiza el número de reporte solicitado por el usuario, notifica á este que el número de reporte no existe.

Tabla 3.1.36 Caso de uso: Imprimir reporte de servicio

Paso	Acción del usuario	Respuesta del sistema
1	Busca el número de reporte que desea imprimir	Encuentra el número de reporte proporcionado por el usuario.
2	Realiza la petición de imprimir el reporte.	Muestra en vista preliminar en pantalla el reporte.
3	Manda imprimir el reporte de servicio.	Manda a impresión el reporte.

Caso alternativo:

1. El usuario puede cancelar la impresión del reporte.

Restricciones:

1. La impresión del reporte de servicio técnico se efectúa en el formato correspondiente.

Tabla 3.1.37 Caso de uso: Ingresar reporte de servicio

Paso	Acción del usuario	Respuesta del sistema
1	Agrega un nuevo reporte de servicio de asistencia técnica, efectuado en la GAM.	Agrega un nuevo registro al control que gestiona el área de Soporte Técnico y pide la confirmación del usuario.
2	Proporciona la fecha en la cual se prestó la asistencia técnica.	
3	Proporciona el número de reporte	

	correspondiente al servicio.	
4	Proporciona el nombre del solicitante del servicio de soporte técnico.	
5	Proporciona el puesto del solicitante del servicio de soporte técnico.	
6	Selecciona el número de serie correspondiente al equipo al cual se le presta el servicio de soporte técnico.	
7		Muestra la descripción general del equipo relacionado al número de serie seleccionado por el usuario. (tipo de equipo, marca, modelo, inicio de garantía, y fin de garantía).
8	Ingresa la asistencia técnica realizada así como el tipo de mantenimiento proporcionado al equipo correspondiente.	
9	Confirma que se lleve a cabo el alta del nuevo reporte de servicio.	Verifica que los datos ingresados por el usuario sean correctos y actualiza el Control de Servicios de la Unidad Departamental de Soporte Técnico.

Caso alternativo:

1. El usuario puede cancelar el alta del nuevo reporte.

Restricciones:

1. El usuario necesita conocer la fecha y número de reporte asignado a el área que solicito el servicio.
2. Si los datos proporcionados por el usuario no son correctos, el sistema cancela la petición de alta del usuario.
3. La fecha en la cual se efectuó el servicio y el número de reporte son datos de carácter obligatorio.

Tabla 3.1.38 Caso de uso: Modificar reporte de servicio

Paso	Acción del usuario	Respuesta del sistema
1	Modifica el reporte de algún servicio realizado por la Unidad Departamental de Soporte Técnico.	Cambia los datos proporcionados por el usuario, y espera la confirmación de este.
2	Confirma que se lleven a cabo los cambios en el reporte de servicio.	Verifica que los cambios efectuados por el usuario sean correctos y prosigue a actualizar el control de servicios de la Unidad Departamental de Soporte Técnico.

Caso alternativo:

1. El usuario puede cancelar las modificaciones realizadas antes de que el sistema actualice el control de servicios.

Restricciones:

1. Si los datos modificados por el usuario no son válidos, el sistema cancela las modificaciones realizadas.

Tabla 3.1.39 Caso de uso: Eliminar reporte de servicio

Paso	Acción del usuario	Respuesta del sistema
1	Realiza la petición de la baja, de un reporte de servicio, dentro del control de asistencia técnica.	Pide la confirmación del usuario para realizar la baja del reporte.
2	Confirma la baja del reporte.	Efectúa la baja del reporte y actualiza el control de servicios proporcionados por la Unidad Departamental de Soporte Técnico.

Caso alterno:

1. El usuario puede cancelar la petición de baja del reporte de servicio.

3.2 EL MODELO DE ANÁLISIS

El lenguaje que utilizamos en el análisis se basa en un modelo de objetos conceptual, que llamamos modelo de análisis. El modelo de análisis nos ayuda a refinar los requisitos, y nos permite razonar sobre los aspectos internos del sistema, incluidos sus recursos compartidos internos. De hecho, un recurso interno puede representarse como un objeto en el modelo de análisis. El modelo de análisis nos ofrece un mayor poder expresivo y una mayor formalización, por ejemplo, lo que nos proporcionan los diagramas de interacción que se utilizan para describir aspectos dinámicos del sistema.

El modelo de análisis también nos ayuda a estructurar los requisitos como se ha explicado y nos proporciona una estructura centrada en el mantenimiento, en aspectos tales como la flexibilidad ante los cambios y la reutilización. Esta estructura no sólo es útil para el mantenimiento de los requisitos como tales, sino que también se utiliza como entrada en las actividades de diseño y de implementación. El modelo de análisis puede considerarse una primera aproximación al modelo de diseño, aunque es un modelo por sí mismo. Mediante la conservación del modelo de análisis durante el diseño, obtenemos un sistema que debería ser también mantenible como todo, será flexible a los cambios en los requisitos, e incluirá elementos que podrán ser reutilizados cuando se construyan sistemas parecidos.

Analizar los requisitos en la forma de un modelo de análisis es importante por varios motivos, a continuación enumeramos los fundamentales:

- Un modelo de análisis ofrece una especificación más precisa de los requisitos que la que tenemos como resultado de la captura de requisitos, incluyendo al modelo de casos de uso.
- Un modelo de análisis se describe utilizando el lenguaje de los desarrolladores, y puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.
- Un modelo de análisis estructura los requisitos de un modo que facilite su comprensión, su preparación, su modificación, y en general, su mantenimiento.
- Un modelo de análisis puede considerarse como una aproximación al modelo de diseño, y es por tanto una entrada fundamental cuando se da forma al sistema en el diseño y en la implementación. Esto se debe a que debería ser mantenible el sistemas en su conjunto, y no sólo la descripción de sus requisitos.

A continuación se muestran las diferencias básicas entre el modelo de casos de uso y el modelo de análisis (ver Tabla 3.2)

Tabla 3.2 Diferencias básicas entre el modelo de casos de uso y el modelo de análisis

MODELO DE CASOS DE USO	MODELO DE ANÁLISIS
Descrito con el lenguaje del cliente	Descrito con el lenguaje del desarrollador
Vista externa del sistema	Vista interna del sistema
Estructurado por los casos de uso, proporciona la estructura a la vista externa.	Estructurado por clases y paquetes estereotipados, proporciona la estructura a la vista interna.
Utilizado fundamentalmente como contrato entre el cliente y los desarrolladores sobre qué debería y que no debería hacer el sistema.	Utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado.
Puede contener redundancias, inconsistencias, etc, entre requisitos.	No debería contener redundancias, inconsistencias, etc, entre requisitos.
Captura la funcionalidad del sistema, incluida la funcionalidad significativa para la arquitectura.	Esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa par la arquitectura, sirve como una primera aproximación al diseño.
Define casos de uso que se analizarán con más profundidad en el modelo de análisis.	Define realizaciones de casos de uso, y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso.

Diferencias del Modelo de Casos de Uso y Modelo de Análisis

3.2.1 CLASES DE ANÁLISIS

Clase de Interfaz Validación usuario

La clase de interfaz **Validación Usuario**, permite al usuario editar su identificador personal, para que el sistema lo verifique, y el usuario tenga acceso al panel de control principal (Ver figura 3.2.1)

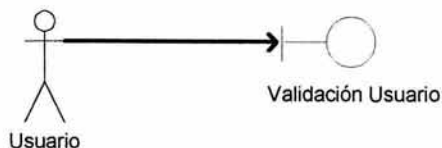


Figura 3.2.1

Clase de Interfaz Panel de control

La clase de interfaz **Panel de control** (Menú principal), permite al usuario navegar por todos los procesos disponibles en el sistema de Soporte Técnico. En dicha interfaz gráfica existen como atributos 5 botones de comando, que responden a sus eventos correspondientes mostrando en pantalla de las siguientes clases de Interfaz:

- Catálogos
- Catálogo equipo informático
- Asignación de equipo
- Control de Soporte Técnico
- Salir Windows

(Ver Figura 3.2.2)

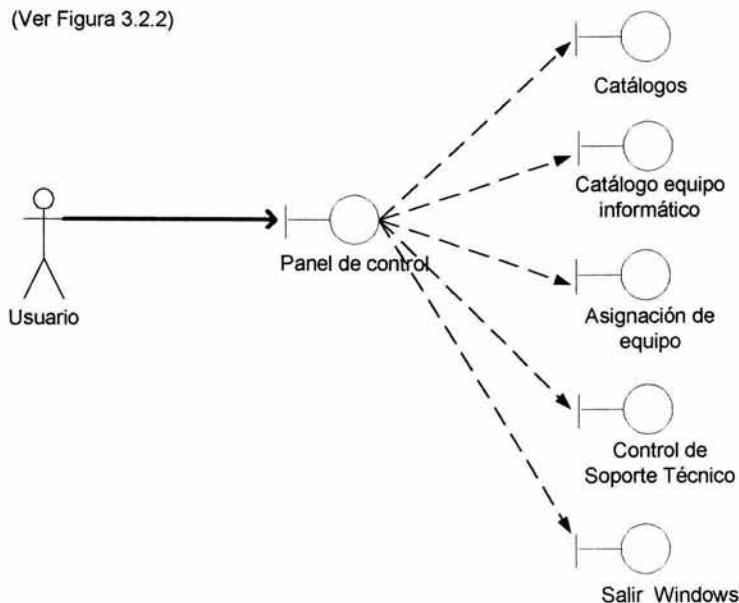


Figura 3.2.2

Clase de interfaz Catálogos y su interacción con otras clases de análisis

La clase de interfaz **Catálogos**, gestiona tres clases de interfaz **Catálogo Equipos**, **Catálogo Marcas** y **Catálogo Modelos**, a través de 3 botones de comando que forman parte de los atributos que destacan en el ambiente gráfico de dicha clase de interfaz.

Las clases de Interfaz **Catálogo Equipos**, **Catálogo Marcas** y **Catálogo Modelos** a su vez muestran un ambiente gráfico en el cual destaca como atributo principal una barra de herramientas personalizada, que permite al usuario del sistema las siguientes funciones:

- Búsqueda
- Impresión
- Altas
- Bajas
- Cambios

Las 3 clases de interfaz mencionadas anteriormente, realizan la actualización de sus clases de entidad **tblEquipos**, **tblMarcas** y **tblModelos** correspondientemente. Al actualizar las 3 clases de entidad mencionadas anteriormente, implícitamente se actualiza la clase de entidad **tblInventario** (Inventario informático) Ver figura 3.2.3

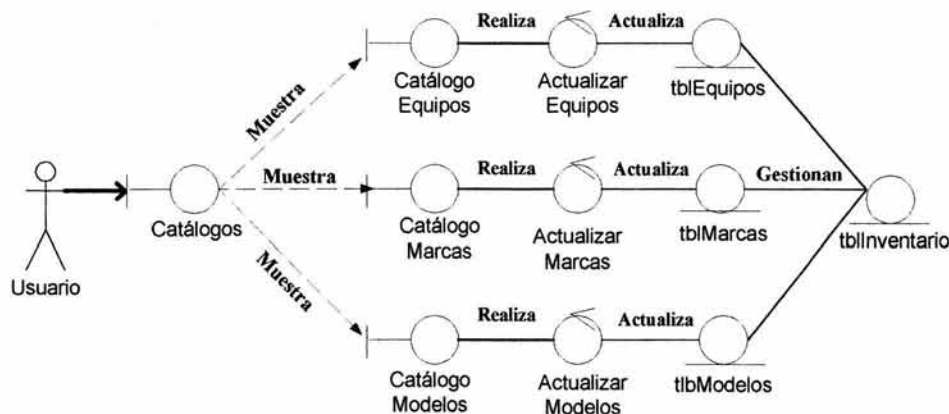


Figura 3.2.3

Clase de interfaz Catálogo Equipo Informático

La clase de interfaz **Catálogo equipo informático**, presenta el ambiente gráfico para que el usuario pueda ejecutar la clase de control **Actualizar Inventario informático** a través de una barra de herramientas personalizada que le permite efectuar las siguientes funciones:

- Buscar número de serie
- Imprimir inventario informático
- Agregar un nuevo equipo al inventario
- Modificaciones en los equipos
- Eliminar equipos del inventario

Las especializaciones anteriores actualizan la clase de entidad **tblInventario** (Inventario informático) Ver figura 3.2.4

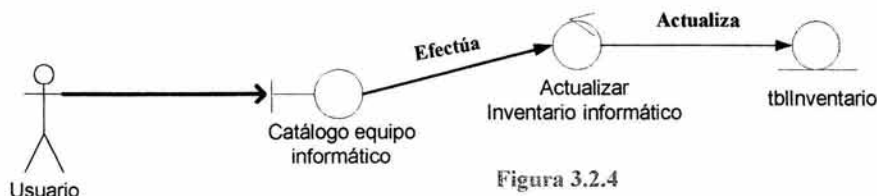


Figura 3.2.4

Clase de interfaz Asignación de equipo informático y su interacción con otras clases de análisis

La clase de interfaz **Asignación del equipo informático**, permite al usuario del sistema administrar la clase de control **Asignar equipo informático**. Dicha clase de control instancia el caso de uso **Asignar equipo informático**, que es el encargado de actualizar la clase de entidad **tblAsignacionEquipo** que representa almacenamiento persistente (Ver figura 3.2.5).

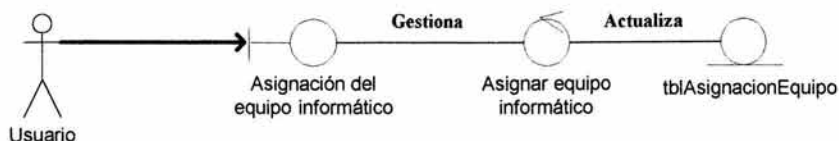


Figura 3.2.5

Clase de interfaz Soporte Técnico y su interacción con otras clases de análisis

La clase de interfaz **Soporte Técnico**, es la interfaz gráfica que se encarga de interactuar con el usuario para poder gestionar la clase de control **Prestar Soporte Técnico**, dicha interfaz permite al usuario las siguientes acciones:

- Búsqueda de reportes de servicio
- Impresión de reportes de servicio
- Ingreso de un nuevo reporte de servicio
- Cambios en los reportes de servicio
- Bajas de reportes de servicio

La clase de entidad **tblSoporte**, hace referencia de almacenamiento de información, y es actualizada al instanciar el caso de uso **Prestar Soporte Técnico** (Ver figura 3.2.6)

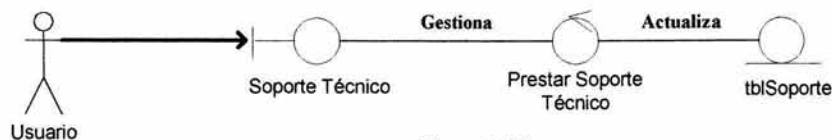


Figura 3.2.6

Clase de interfaz Salir Windows

La clase de interfaz **Salir Windows**, permite al usuario del sistema cerrar la sesión actual o en su caso alternar devolver el control al Panel de Control Principal (Menú principal) Ver figura 3.2.7

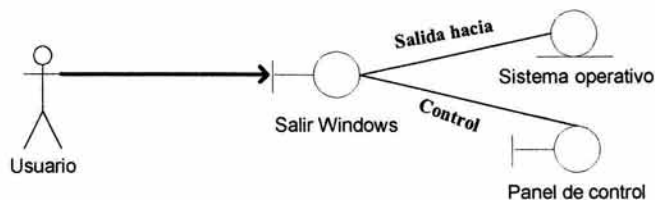


Figura 3.2.7

CAPÍTULO IV

**MODELO DE DISEÑO DE SISTEMA
INTEGRAL DE SOPORTE TÉCNICO
DE LA DELEGACIÓN GUSTAVO A.
MADERO**

4.1 EL MODELO DE DISEÑO

En el diseño modelamos el sistema y encontramos su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales y otras restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis; esto es, el modelo del análisis.

El modelo de análisis proporciona una comprensión detallada de los requisitos. Y lo que es más importante, impone una estructura del sistema que debemos esforzar por conservar lo más fielmente posible cuando demos forma al sistema. En concreto, los propósitos del diseño son:

- Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, etc.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia. Esto resulta útil en los casos en los que la descomposición no puede ser hecha basándose en los resultados de la captura de requisitos o análisis.
- Capturar las interfaces entre los subsistemas dentro del ciclo de vida del software. Esto ayuda cuando reflexionamos sobre la arquitectura y cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrollo.
- Ser capaces de visualizar y reflexionar sobre el diseño utilizando una notación común.
- Crear una abstracción sin costuras de la implementación del sistema, en el sentido de que la implementación es un refinamiento directo del diseño que rellena lo existente sin cambiar la estructura. Esto permite la utilización de tecnologías como la generación de código y la ingeniería de ida y vuelta entre el diseño y la implementación.

El modelo de diseño es un modelo de objetos que describe la realización física de los Casos de Uso centrándose en cómo los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de implementación tienen impacto en el sistema a considerar [Rumbaugh James, et al, 2000 (a)]. Además el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación.

Enseguida se efectuará una comparación entre el modelo de análisis y el modelo de diseño (Ver tabla 4.1)

Tabla 4.1 Diferencias básicas entre el modelo análisis y el modelo de diseño

Modelo de análisis	Modelo de diseño
Modelo conceptual, porque es una abstracción del sistema y permite aspectos de la implementación.	Modelo físico, porque es un plano de la implementación.
Genérico respecto al diseño.	No genérico, específico para la implementación.
Tres estereotipos conceptuales sobre las clases: Control, Entidad e Interfaz.	Cualquier número de estereotipos físicos sobre las clases, dependiendo del lenguaje de implementación.
Menos formal.	Más formal.
Menos caro de desarrollar.	Más caro de desarrollar.
Menos capas.	Más capas.
Dinámico (no muy centrado en la secuencia)	Dinámico. (muy centrado en la secuencia)
Bosquejo del diseño del sistema, incluyendo su arquitectura.	Manifiesto del diseño del sistema, incluyendo su arquitectura.
Creado principalmente como trabajo de a pie en talleres o similares.	Creado principalmente como programación visual en ingeniería de ida y vuelta, el modelo de diseño es realizado según la ingeniería de ida y vuelta con el modelo de implementación.
Puede no estar mantenido durante el ciclo de vida del software.	Debe ser mantenido durante todo el ciclo de vida del software.
Define una estructura que es una entrada esencial para modelar el sistema, incluyendo la creación del modelo de diseño.	Da forma al sistema mientras que intenta preservar la estructura definida por el modelo de análisis lo más posible.

4.2 DIAGRAMAS Y DESCRIPCIÓN DE CLASES

Tabla 4.2 Clase: tblCatalogoAreas

Atributo	Tipo	Longitud	Descripción	Objeto Índice
CveDepartamental	VB::Char	8	Es el identificador único de cada una de las Unidad Departamental de la Delegación Gustavo A. Madero.	Si
CvePresupuestal	VB::Char	10	Es el identificador de la clave presupuestal destinada a la Unidad Departamental.	*****
CveCorta	VB::Char	5	Es un identificador corto asignado a cada una de las Unidades Departamentales de la Delegación.	*****
Siglas	VB::Char	5	Son las siglas con las que se hacer referencia a la Unidad Departamental.	*****
GrupoDeTrabajo	VB::Char	14	Es el grupo de trabajo de la red al que se encuentra asignado la Unidad Departamental.	*****
Area	VB::Char	75	Es la descripción general del área adscrita a la Delegación Gustavo A. Madero.	*****
Funcionario	VB::Char	50	Es el nombre completo del funcionario.	*****
Titulo	VB::Char	10	Es el título que ostenta el funcionario.	*****
Nombre	VB::Char	25	Es el nombre el funcionario	*****
ApPaterno	VB::Char	20	Es el apellido paterno del funcionario.	*****
ApMaterno	VB::Char	20	Es el apellido materno del funcionario.	*****
EMail	VB::Char	50	Es la dirección de correo electrónico destinado para el funcionario titular de la Unidad Departamental.	*****
Ext1	VB::Char	4	Es el número de extensión de la Unidad Departamental.	*****
Ext2	VB::Char	4	Es la segunda extensión de la Unidad Departamental.	*****
Red	VB::Char	4	Descripción de la red a la que pertenece.	*****
Fax	VB::Char	8	Es el número de fax destinado a la Unidad Departamental.	*****
Telefono1	VB::Char	8	Es el teléfono de la Unidad Departamental.	*****
Telefono2	VB::Char	8	2do. teléfono de la Unidad Departamental.	*****

Telefono3	VB::Char	8	3er. Teléfono de la Unidad Dep.	*****
Celular	VB::Char	13	Es el número celular del Funcionario titular de la Unidad Departamental correspondiente.	*****
numUbicacion	VB::Long	*****	Es el número con el cual se identifica en que edificio y piso se localiza la Unidad Departamental correspondiente.	*****
Edificio	VB::Char	35	Es la descripción del edificio en el cual se encuentra la Unidad Departamental correspondiente.	*****
Calle	VB::Char	50	Es la calle de la dirección particular en donde se localiza la Unidad Departamental correspondiente.	*****
Interior	VB::Char	50	Es el número de interior que hace alusión a la dirección de la Unidad Departamental.	*****
Colonia	VB::Char	50	Es la colonia en la cual se localiza la Unidad Departamental.	*****
CodigoPostal	VB::Char	5	Es el código postal de la colonia en la cual se localiza la Unidad Departamental correspondiente.	*****
Foto	VB::Objet	*****	Es la fotografía del funcionario titular de la Unidad Departamental correspondiente.	*****

Tabla 4.2.1 Clase: tblEquipos:

Atributo	Tipo	Longitud	Descripción	Objeto Índice
Id_equipo	VB::Long	5	Es el número con el cual se identifica al equipo informático.	Si
Equipo	VB::Char	50	Es la descripción general del equipo.	*****

Tabla 4.2.2 Clase: tblMarcas:

Atributo	Tipo	Longitud	Descripción	Objeto Índice
Id_equipo	VB::Long	5	Es el número con el cual se identifica al equipo informático.	Si
Id_marca	VB::Char	15	Es el identificador del modelo del equipo informático.	Si
Marca	VB::Char	50	Es la descripción general de la marca del tipo de equipo.	*****

Tabla 4.2.3 Clase: tblModelos:

Atributo	Tipo	Longitud	Descripción	Objeto Índice
Id_modelo	VB::Long	5	Es el número con el cual se identifica el modelo del equipo informático.	Si
Modelo	VB::Char	50	Es la descripción general del modelo del equipo informático.	*****

Tabla 4.2.4 Clase: tblCatalogoEquipoInformatico:

Campo	Tipo	Longitud	Descripción	Objeto Índice
Equipo	VB::Char	50	Es la descripción general del equipo informático.	*****
Marca	VB::Char	50	Es la descripción general de la marca del equipo informático.	*****
Modelo	VB::Char	50	Es la descripción general del modelo del tipo de equipo informático.	*****
Serie	VB::Char	35	Es el número de serie que identifica al equipo informático	Si
Garantía	VB::Boolean	*****	Es la descripción de la garantía del equipo.	*****
InicioDeGarantia	VB::Date	8	Es la fecha en la cual empieza a correr la vigencia de la garantía.	*****
TerminoDeGarantia	VB::Date	8	Es la fecha de término de la vigencia de garantía del equipo.	*****
Observaciones	VB::Char	"n"	Hace referencia a cualquier tipo de descripción adicional al equipo informático.	*****

Tabla 4.2.5 Clase: tblAsignacionEquipoInformatico:

Atributo	Tipo	Longitud	Descripción	Objeto Índice
CveDepartamental	VB::Char	8	Es la clave departamental que identifica al Área correspondiente.	Si
Telefono	VB::Char	8	Es el teléfono de la Unidad Departamental	*****
Ext	VB::Char	4	Es el número de extensión de la Unidad Departamental	*****
Equipo	VB::Char	50	Es la descripción general del tipo de equipo informático.	*****
Marca	VB::Char	50	Es la descripción general de la marca que hace referencia al tipo equipo.	*****
Modelo	VB::Char	50	Descripción general del modelo.	*****
Serie	VB::Char	35	Es el número de serie que identifica	Si

			al equipo informático.	
Garantia	VB::Boolean	*****	Es la descripción de la garantía	*****
InicioDeGarantia	VB::Date	8	Es la fecha de inicio de la vigencia de garantía del equipo.	*****
TerminoDeGarantia	VB::Date	8	Es la fecha de término de la vigencia de garantía del equipo.	*****
Observaciones	VB::Char	"n"	Es cualquier tipo de observación adicional al equipo informático.	*****

Tabla 4.2.6 Clase: tblSoporte:

Atributo	Tipo	Longitud	Descripción	Indice
Fecha	VB::Date	8	Es la fecha en la cual se realizó la asistencia técnica.	*****
No_reporte	VB::Long	5	Es el número de reporte que es asignado al servicio de asistencia técnica realizada por la Jefatura de Soporte Técnico.	Si
Nombre	VB::Char	50	Es el nombre del solicitante del servicio de soporte técnico	*****
Puesto	VB::Char	50	Es la descripción del puesto del solicitante del servicio.	*****
Area	VB::Char	75	Es la descripción de la Unidad Departamental solicitante.	*****
Telefono	VB::Char	8	Es el número de teléfono de la Unidad Departamental solicitante	*****
Ext	VB::Char	4	Es el número de extensión de la Unidad Departamental del solicitante.	*****
Soporte	VB::Boolean	*****	Identifica si se prestó servicio de soporte técnico.	*****
Red	VB::Boolean	*****	Identifica si se prestó servicio a la red.	*****
Antivirus	VB::Boolean	*****	Identifica si se prestó servicio de instalación o actualización de antivirus.	*****
Respaldo	VB::Boolean	*****	Identifica si se prestó servicio de respaldo.	*****
Internet	VB::Boolean	*****	Identifica si se prestó asistencia para cualquier tema relacionado con la Internet.	*****
OtrosAsistencia	VB::Boolean	*****	Identifica si el equipo solicito algún tipo de asistencia técnica distinta a los tipos de asistencia técnica mencionados anteriormente.	*****
Desc_Asistencia	VB::Char	*****	Es la descripción general de la asistencia técnica brindada al equipo correspondiente.	*****
Hardware	VB::Boolean	*****	Identifica si al equipo se le realizó algún tipo de servicio relacionado con hardware.	*****
Software	VB::Boolean	*****	Identifica si al equipo se le realizó algún tipo de servicio relacionado con el software.	*****
Telecomunicaciones	VB::Boolean	*****	Identifica si el mantenimiento realizado fue relacionado con aspectos de telecomunicaciones.	*****
OtrosMantenimiento	VB::Boolean	*****	Identifica cualquier otro tipo de	*****

			servicio que no se encuentre enumerado en los anteriores.	
Desc_Mantenimiento	VB::Char	*****	Es la descripción adicional del tipo de mantenimiento que se brinda.	*****
Equipo	VB::Char	50	Es la descripción general del tipo de equipo informático.	*****
Marca	VB::Char	50	Es la descripción de la marca del que hace referencia al equipo informático.	*****
Modelo	VB::Char	50	Es la descripción general del modelo que hace referencia al equipo informático.	*****
Serie	VB::Char	35	Es el número de serie que identifica al equipo informático.	*****
Garantia	VB::Boolean	*****	Identifica si el equipo informático cuenta con póliza de garantía.	*****
InicioDeGarantia	VB::Date	8	Es la fecha de inicio de vigencia de la garantía.	*****
TerminoDeGarantia	VB::Date	8	Es la fecha de término de vigencia de la garantía.	*****

RELACIONES DE CLASES ESTATICAS

tblCatalogoAreas			tblAsignacionEquipolInformatico
CveDepartamental	1	0..*	CveDepartamental

Atributos: Forzado, Actualización en cascada, eliminación en cascada

Tipo de relación: Composición

tblCatalogoAreas			tblSoporte
CveDepartamental	1	0..*	Area

Atributos: Forzado, Actualización en cascada, eliminación en cascada

Tipo de relación: Composición

tblCatalogoEquipolInformatico			tblAsignacionEquipolInformatico
Serie	1	1	Serie

Atributos: Forzado, Actualización en cascada, eliminación en cascada

Tipo de relación: Composición

tblEquipos			tblAsignacionEquipolInformatico
Equipo	1	0..*	Equipo

Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición

tblEquipos			tblCatalogoEquipolInformatico
Equipo	1	0..*	Equipo

Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición

tblEquipos			tblMarcas
Id_equipo	1	0..1	Id_equipo

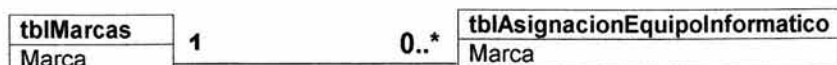
Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición

tblEquipos			tblSoporte
Equipo	1	0..*	Equipo

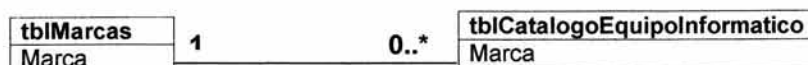
Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición



Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición



Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición



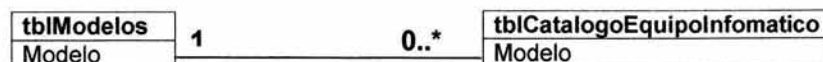
Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición



Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición



Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición



Atributos: Actualización en cascada, eliminación en cascada

Tipo de relación: Composición

OBJETOS ÍNDICES

Clase: tblCatalogoAreas

Objeto Índice	Atributo	Tipo
tblCatalogoAreas_CveDepartamental	CveDepartamental	Principal
tblCatalogoAreas_CveCorta	CveCorta	Único
tblCatalogoAreas_Siglas	Siglas	Único
tblCatalogoAreas_Areas	Areas	Único

Clase: tblAsignacionEquipolnformatico

Objeto Índice	Atributo	Tipo
tblAsignacionEquipolnformatico_CveDepartamental	CveDepartamental	Único
tblAsignacionEquipolnformatico_Equipo	Equipo	Único
tblAsignacionEquipolnformatico_Marca	Marca	Único
tblAsignacionEquipolnformatico_Modelo	Modelo	Único
tblAsignacionEquipolnformatico_Serie	Serie	Principal

Clase: tblCatalogoEquipolnformatico

Objeto Índice	Atributo	Tipo
tblCatalogoEquipolnformatico_Equipo	Equipo	Único
tblCatalogoEquipolnformatico_Marca	Marca	Único
tblCatalogoEquipolnformatico_Modelo	Modelo	Único
tblCatalogoEquipolnformatico_Serie	Serie	Principal

Clase: tblEquipos

Objeto Índice	Atributo	Tipo
tblEquipos_IdEquipo	Id_Equipo	Principal
tblEquipos_Equipo	Equipo	Único

Clase: tblMarcas

Objeto Índice	Atributo	Tipo
tblMarcas_IdMarca	Id_Marca	Principal
tblMarca_IdEquipo	Id_Equipo	Único
tblMarcas_Marca	Marca	Único

Clase: tblModelos

Objeto Índice	Atributo	Tipo
tblModelos_IdModelo	Id_Modelo	Principal
tblModelos_Modelo	Modelo	Único

Clase: tblSoporte

Objeto Índice	Atributo	Tipo
tblSoporte_NoReporte	No_Reporte	Principal

Clase: tblUsuarios

Objeto Índice	Atributo	Tipo
TblUsuarios_Password	Password	Único

Diagrama de clases: Presenta un conjunto de clases y las relaciones entre ellas. Los diagramas de clases describen la vista de diseño estático de un sistema (Figura 4.2).

DIAGRAMA DE CLASES ESTÁTICAS

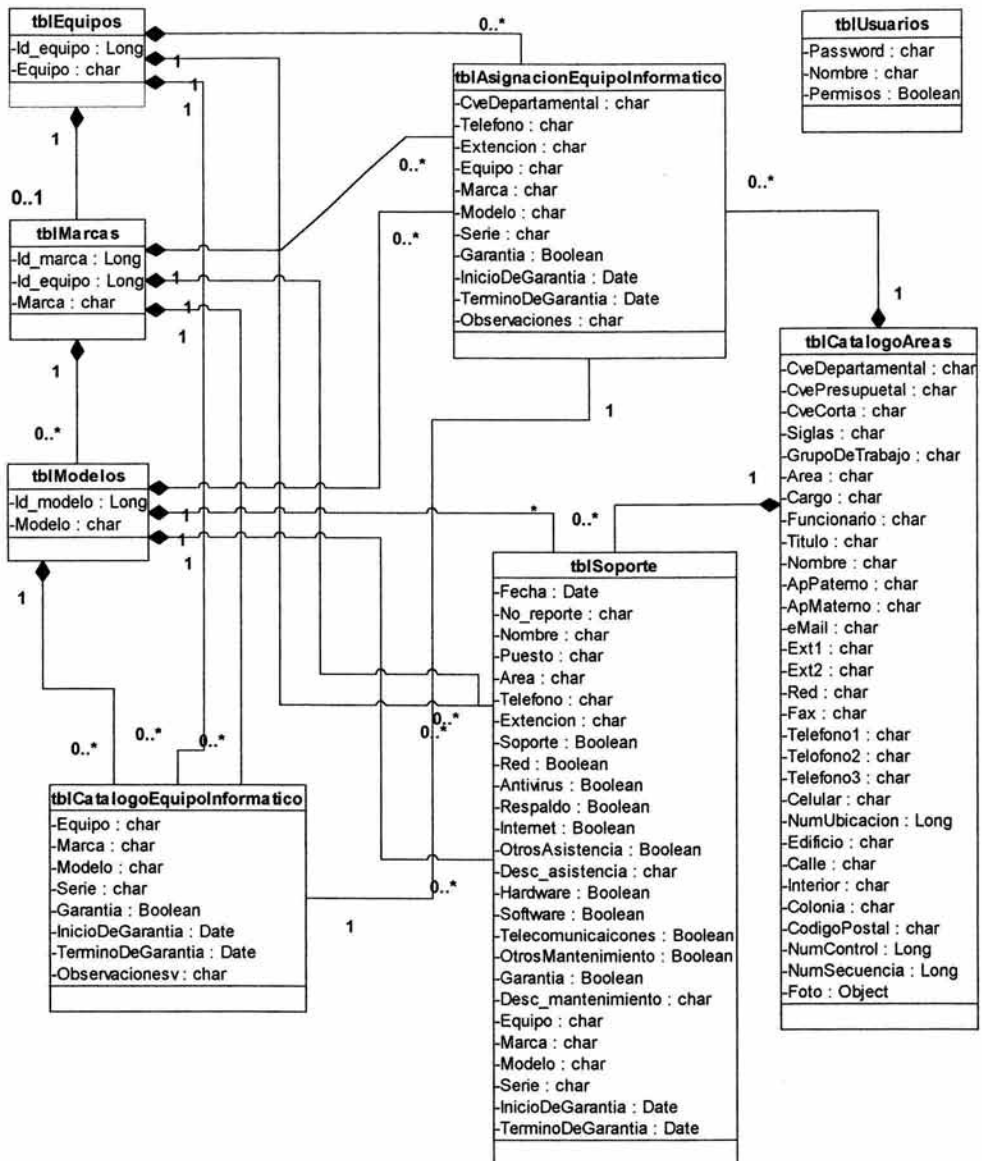


Figura 4.2
Modelado de Clases Estáticas

El modelo (figura 4.2 muestra cada una de las clases estáticas así como el tipo y relaciones existentes entre ellas (Ver Relación de clases estáticas, página 66 y 67)

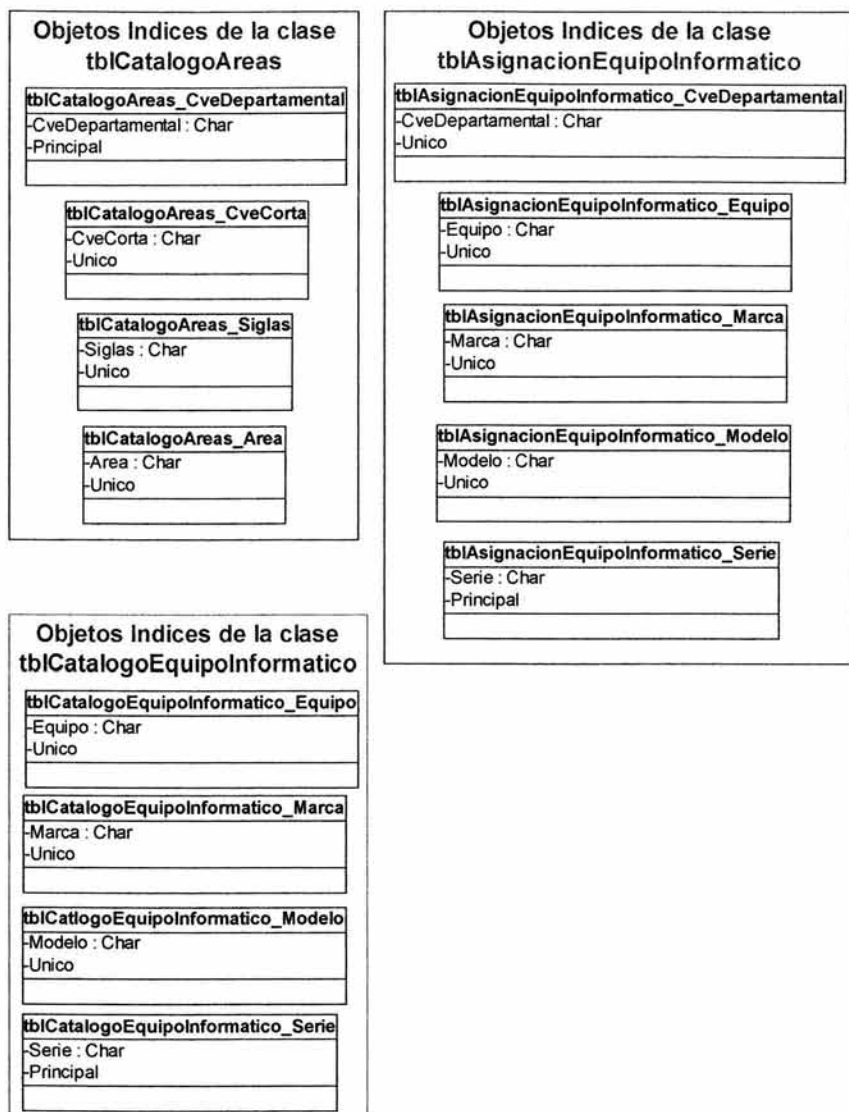


Figura 4.2.1

El modelo (figura 4.2.1) muestra cada una de las clases estáticas así como el tipo y relaciones existentes entre ellas (Ver Objetos índice, página 68)

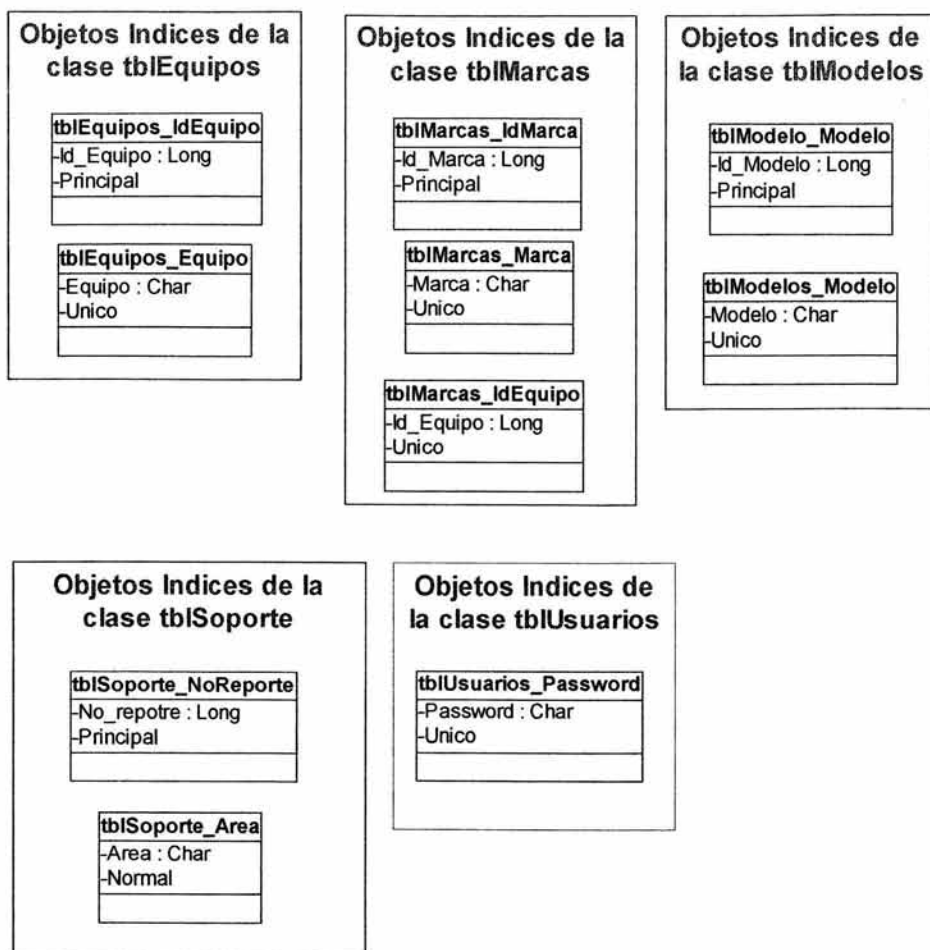


Figura 4.2.2

El modelo (figura 4.2.2) muestra cada una de las clases estáticas así como el tipo y relaciones existentes entre ellas (Ver Objetos índice, página 68)

Clases dinámicas: En la mayoría de los casos, un objeto tiene un tipo en tiempo de ejecución, este es el caso de las clases estáticas, Si un objeto puede cambiar su tipo en tiempo de ejecución, es el caso de las clases dinámicas

La clasificación dinámica es variación semántica en la que un objeto puede cambiar de tipo o de rol.

DIAGRAMA DE CLASES DINÁMICAS

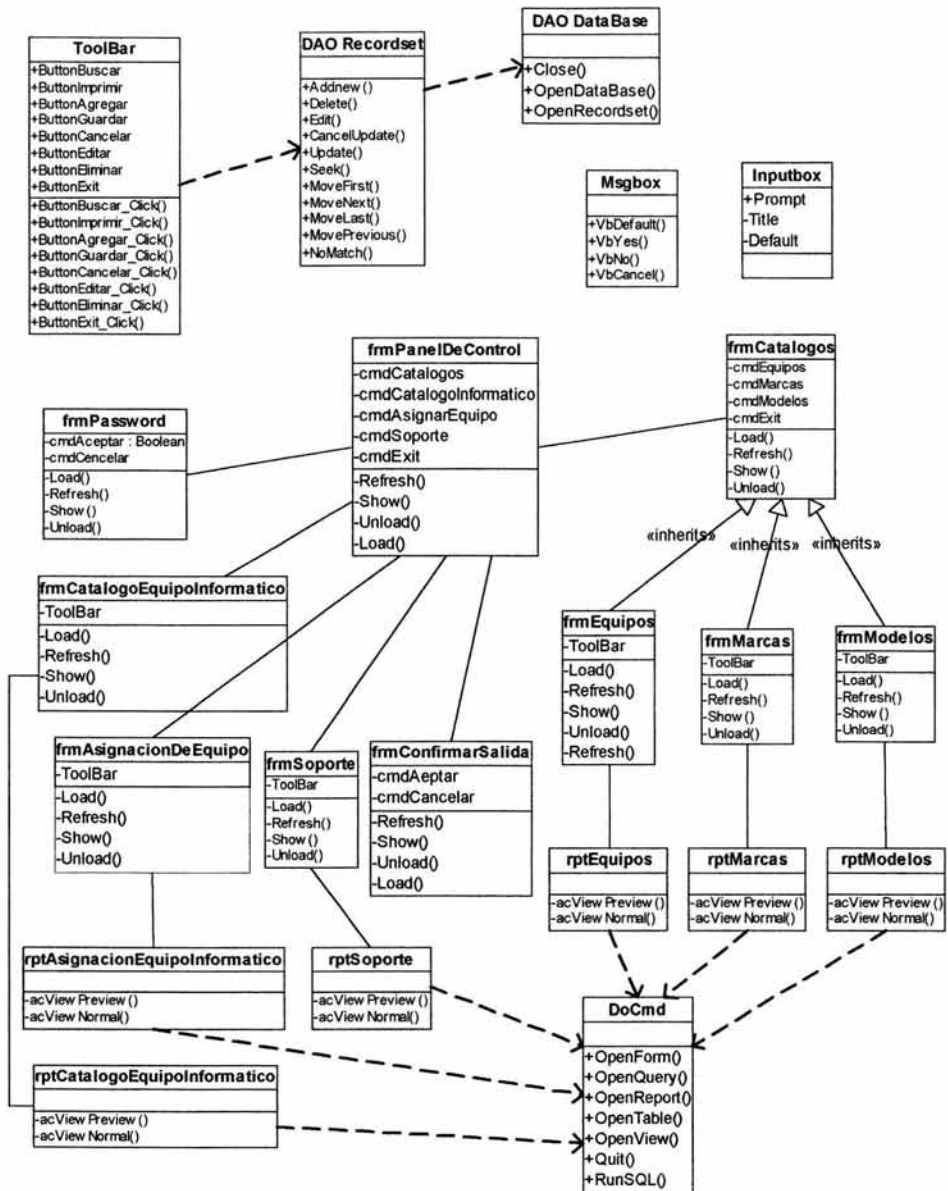
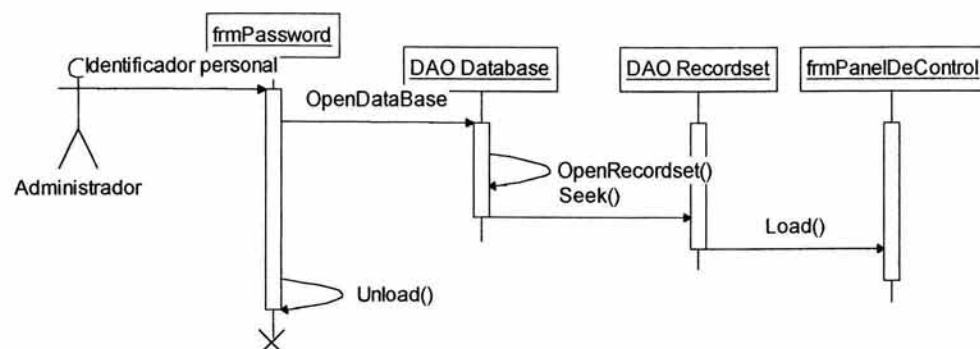


Figura 4.2.3
Modelado de Clases Dinámicas

En la figura 4.2.3 se muestra las clases dinámicas, su tipo y relaciones existentes.

4.3 DIAGRAMAS DE SECUENCIA DEL SISTEMA INTEGRAL DE CONTROL DE SOPORTE TÉCNICO EN LA GAM

Diagrama de Secuencia, Caso: Validar Usuario



Caso Alternativo

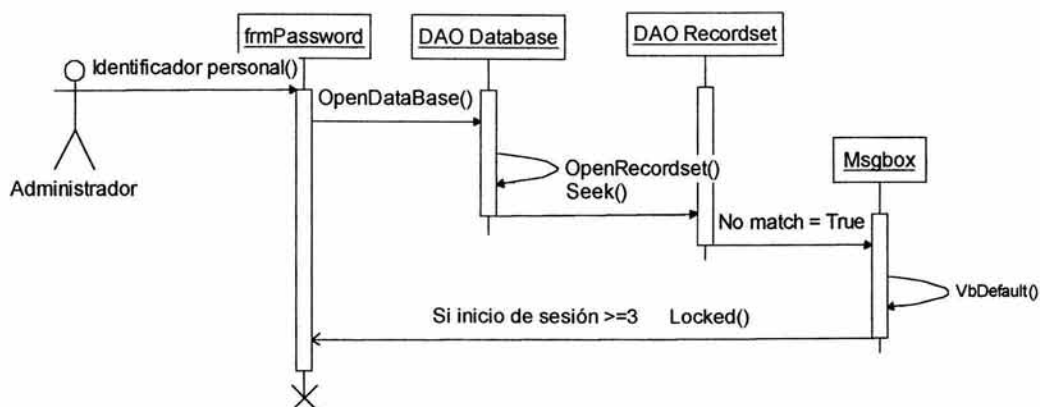


Figura 4.3.1

Los diagramas de interacción especifican la implementación física de los casos de uso. En los diagramas de secuencia: Validar usuario y su caso alternativo (figura 4.3.1) se instancia el caso de uso Validar usuario, cada uno de los pasos está descrito en el Capítulo 3.1.2 Descripción de los Casos de uso (Caso de uso: Validar usuario, tabla 3.1.2).

Diagrama de secuencia Panel de Control Caso: Catálogos

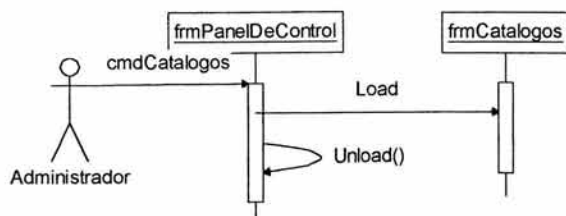


Diagrama de secuencia Panel de Control Caso: Catálogo Equipo Informático

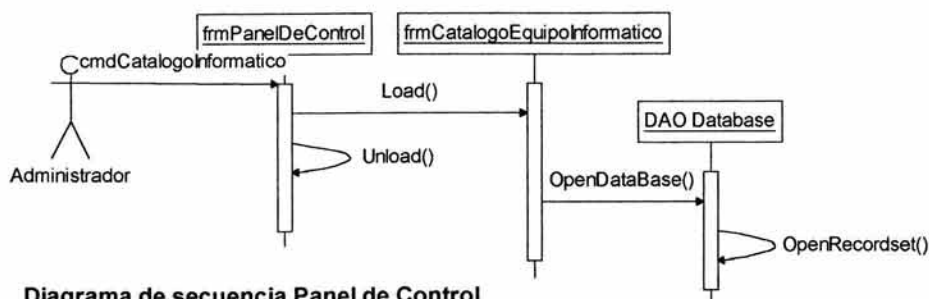


Diagrama de secuencia Panel de Control Caso: Asignación Equipo Informático

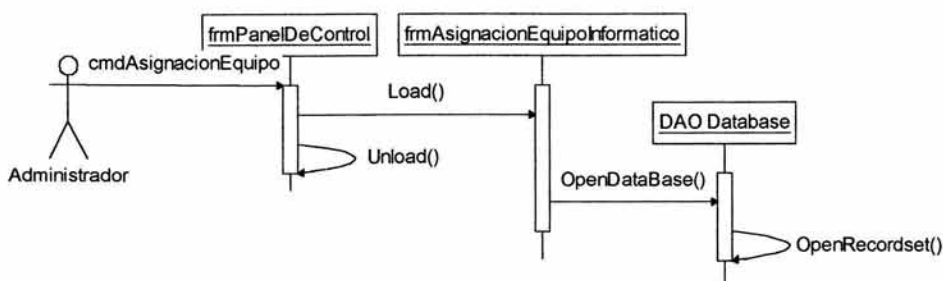


Figura 4.3.2

La figura 4.3.2 muestra tres diagramas de interacción Actualizar Catálogos, Catálogo de equipo informático y Asignación de equipo informático. El Caso de uso: Actualización de Catálogos describe los pasos de diagrama correspondiente (Ver página 38 tabla 3.1.3) El diagrama Catálogo de equipo informático no describe un caso de uso en realidad, pero es una especificación de los procesos del sistema. El diagrama de Asignación de equipo informático representa el caso de uso: Asignar equipo informático (Ver página 47 tabla 3.1.28).

Diagrama de secuencia Panel de Control Caso: Control de Soporte Técnico

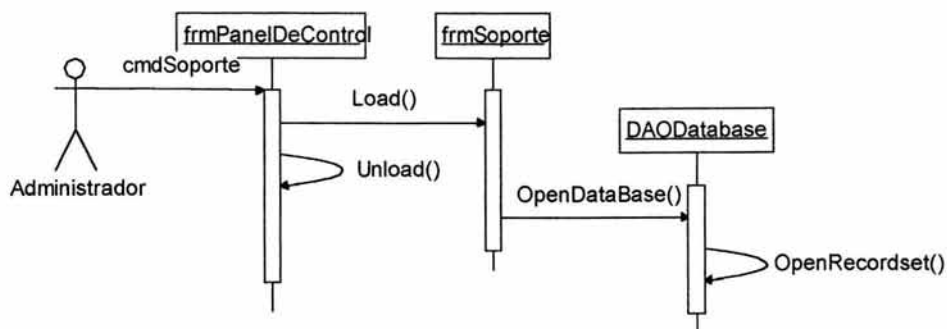


Diagrama de secuencia Panel de Control Caso: Salir a Windows

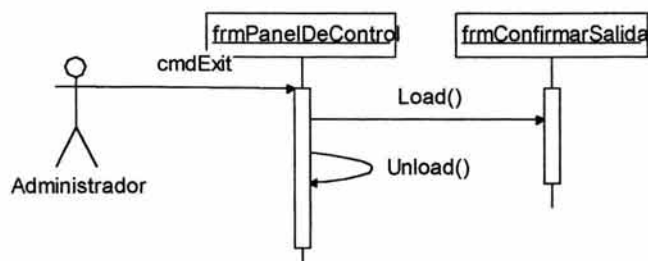


Figura 4.3.3

La figura 4.3.3 muestra los diagramas de secuencia Control de Soporte Técnico y Salir a Windows, básicamente el diagrama Control de Soporte Técnico se encuentra descrito en la página 49 Caso de uso: Prestar Soporte Técnico tabla 3.1.34. El diagrama de interacción Salir de Windows representa el proceso de salir de la aplicación cliente.

Diagrama de secuencia Panel de Control Caso: Catálogos

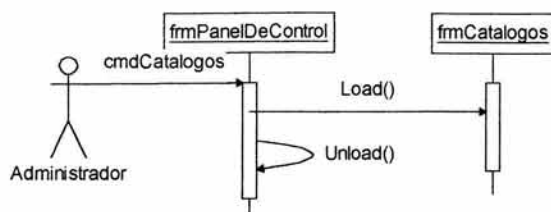


Diagrama de secuencia Panel de Control Caso: Catálogo Equipo Informático

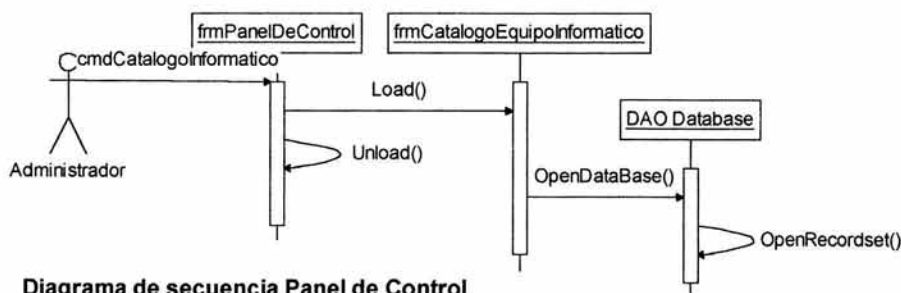


Diagrama de secuencia Panel de Control Caso: Asignación Equipo Informático

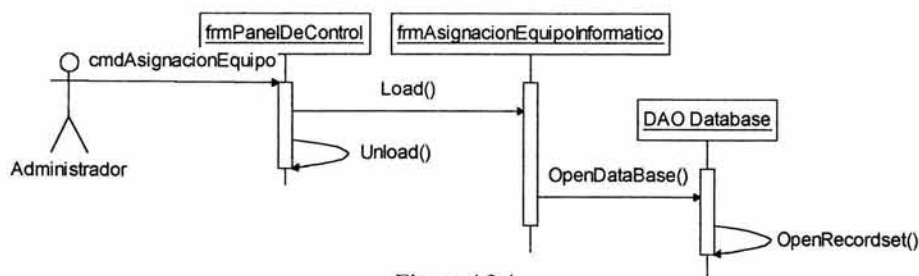


Figura 4.3.4

En la figura 4.3.4 se especifican 3 diagramas de secuencia Catálogos, Catálogo Equipo Informático y Asignación Equipo Informático. El diagrama Catálogos muestra la forma de poder administrar los catálogos disponibles en el sistema. El diagrama Catálogo Equipo Informático especifica el Caso de uso: Actualizar Inventario Informático (Ver página 45 tabla 3.1.22). El diagrama Asignación Equipo Informático especifica el Caso de uso: Asignar Equipo Informático (Ver página 47 tabla 3.1.28).

Diagrama de Secuencia Equipos Caso Especializado: Imprimir

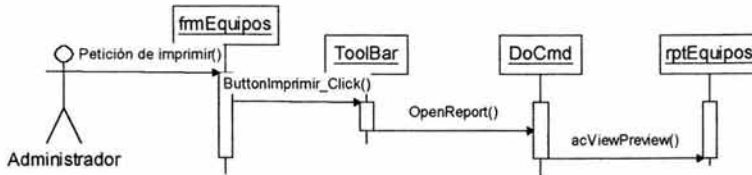
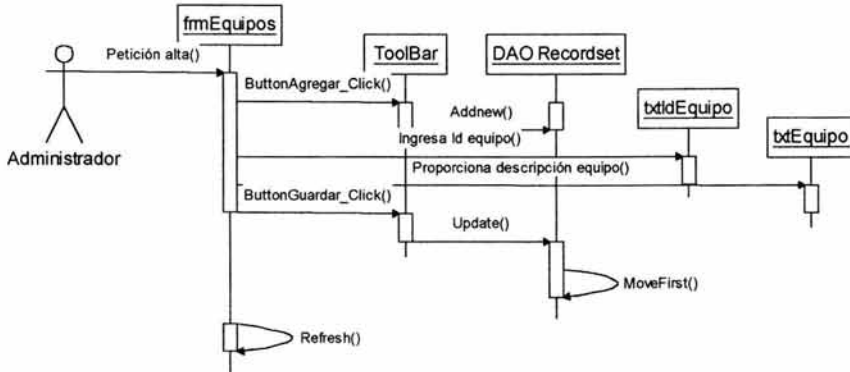


Diagrama de Secuencia Equipos Caso Especializado: Agregar



Caso alternativo Agregar

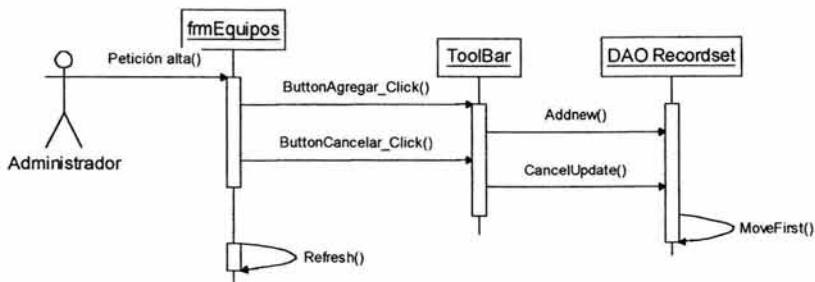
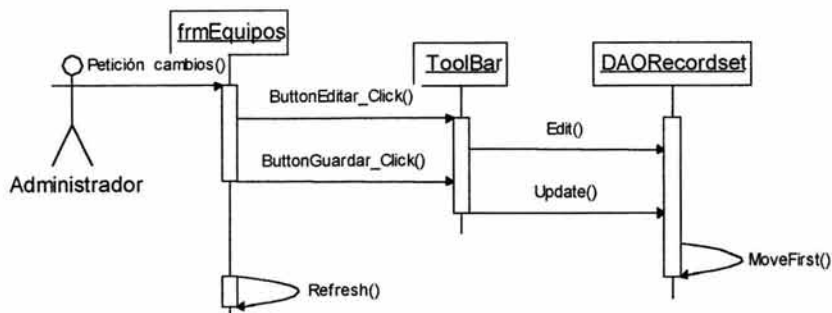


Figura 4.3.5

La figura 4.3.5 describe las especializaciones del Caso de uso Actualizar Equipos. Imprimir catálogo de equipos, Alta de equipos y su caso alternativo. Para mayor referencia ver página 39 tablas 3.1.6 y 3.1.7 respectivamente.

Diagrama de Secuencia Equipos Caso Especializado: Cambios



Caso alternativo Cambios

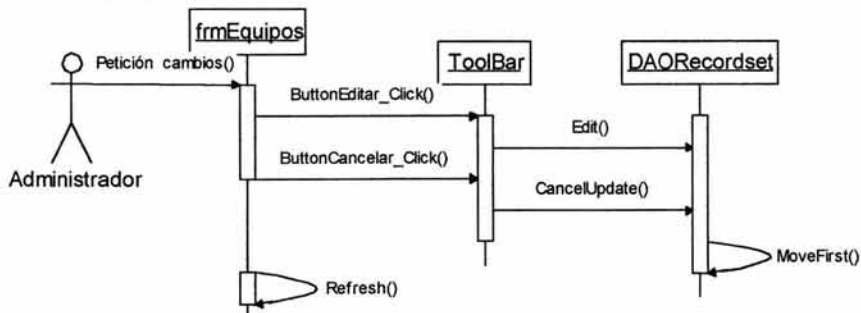


Diagrama de Secuencia Equipos Caso Especializado: Bajas

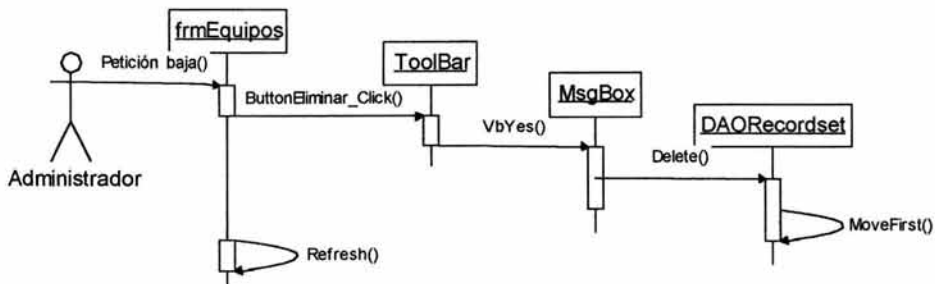


Figura 4.3.6

La figura 4.3.6 describe las especializaciones del Caso de uso: Actualizar Equipos. Modificar descripción de equipos, su caso alternativo y el Caso: Baja de equipos (Ver página 39 tabla 3.1.8, y página 40 tabla 3.1.9).

Diagrama de Secuencia Equipos

Caso Especializado: Bajas

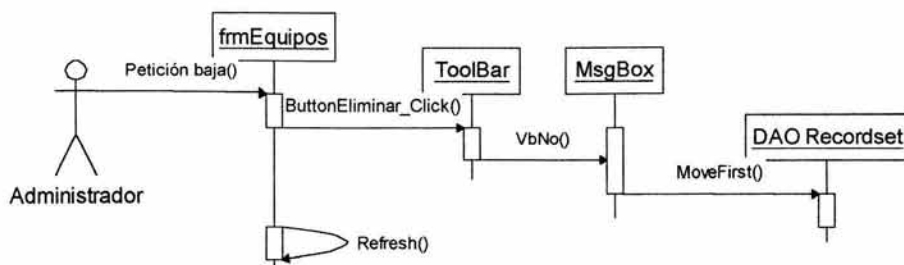


Diagrama de Secuencia Equipos

Caso Especializado: Salir catálogo Equipos

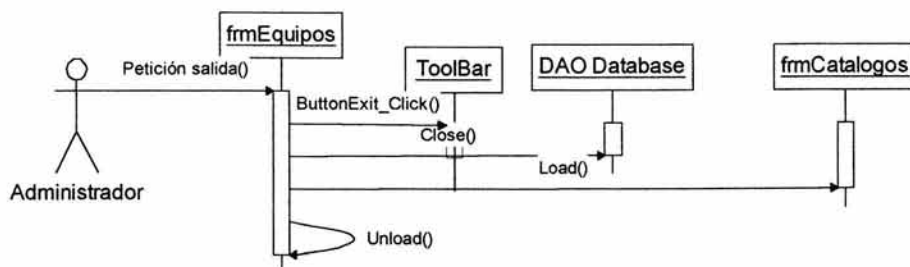
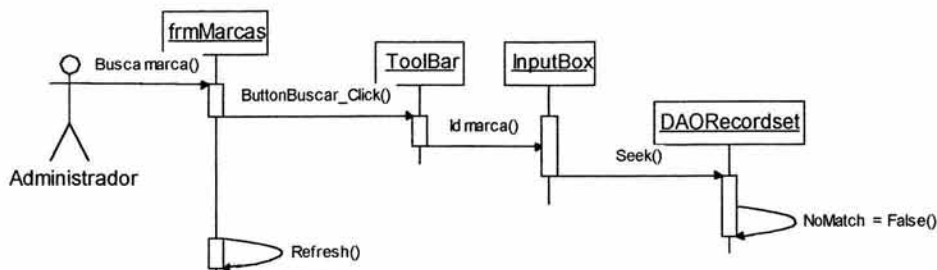


Figura 4.3.7

La figura 4.3.7 representa las especializaciones del Caso de uso: Actualizar Equipos iniciando con el caso alternativo del Caso de uso: Baja de equipos (Ver página 40 tabla 3.1.9).

Diagrama de Secuencia Marcas Caso Especializado: Buscar



Caso alternativo: Buscar marca

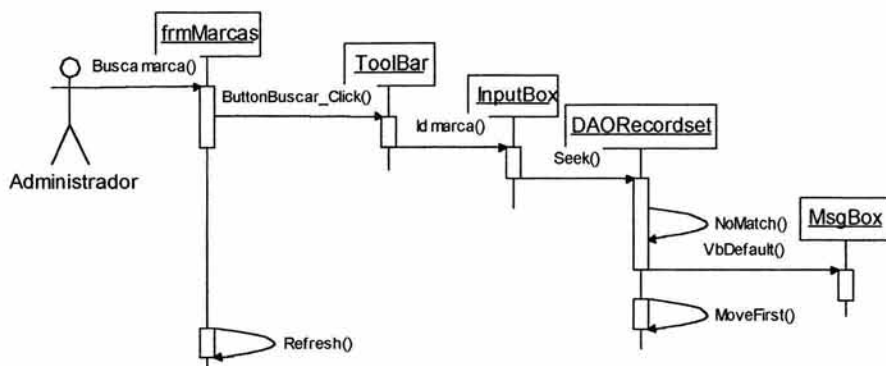


Diagrama de Secuencia Marcas Caso Especializado: Imprimir

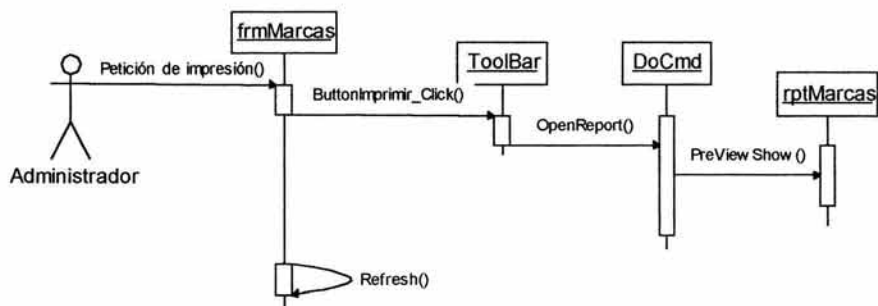
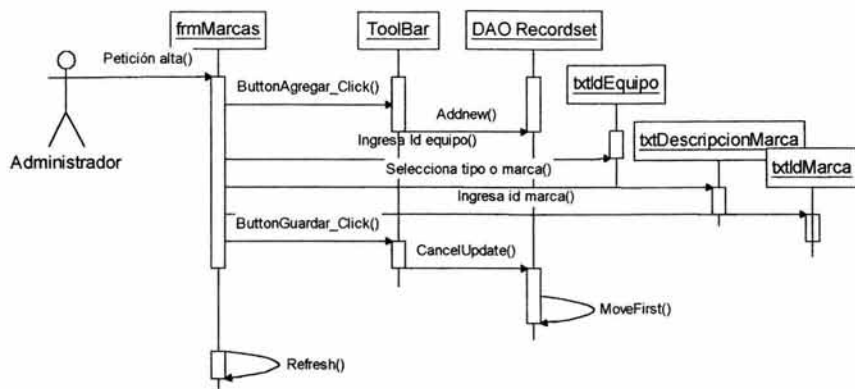


Figura 4.3.8

La figura 4.3.8 describe las especializaciones del Caso de uso: Actualizar Marcas. Buscar marca su caso alternativo e Imprimir catálogo de marcas (Ver página 41 tablas 3.1.11 y 3.1.12 respectivamente).

Diagrama de Secuencia Marcas Caso Especializado: Agregar marca



Caso alternativo: Agregar marcas

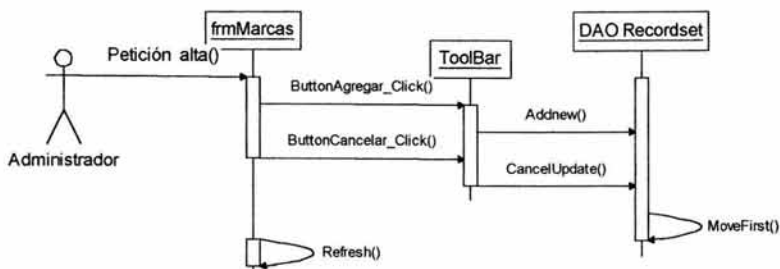


Diagrama de Secuencia Marcas Caso Especializado: Cambiar marca

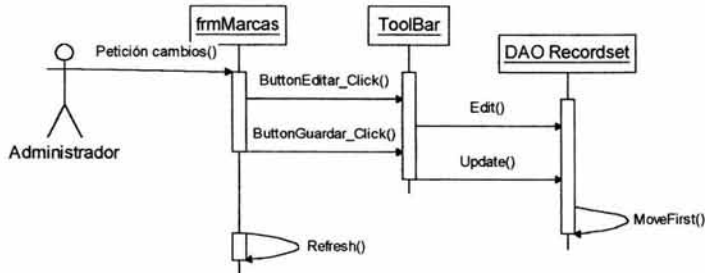


Figura 4.3.9

La figura 4.3.9 describe las especializaciones del Caso de uso: Actualizar Marcas. Alta de marcas su caso alternativo y Modificar descripción de marca (Ver página 41 tablas 3.1.13 y 3.1.14 respectivamente).

Caso Alterno: Cambiar marca

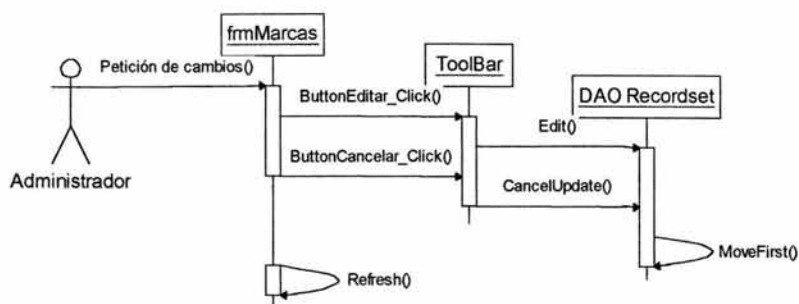
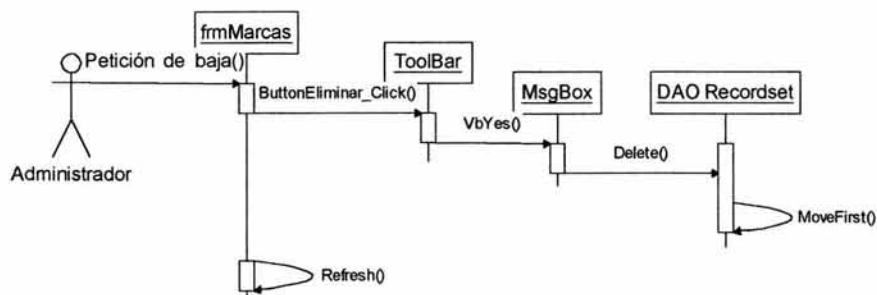


Diagrama de Secuencia Marcas Caso Especializado: Eliminar marca



Caso Alterno: Eliminar marca

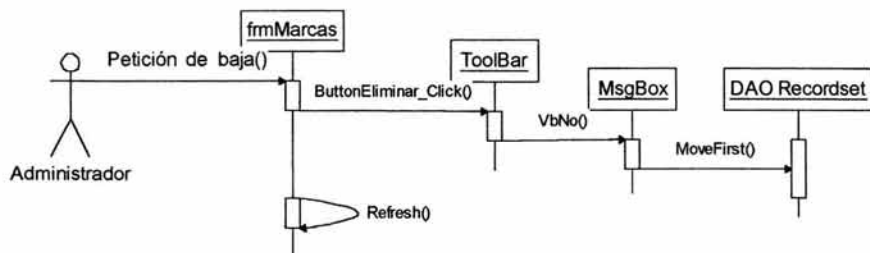


Figura 4.3.10

La figura 4.3.10 describe las especializaciones del Caso de uso: Actualizar Marcas. Caso alternativo de Modificar descripción de marcas, Baja de marcas y su caso alternativo. (Ver página 42 tabla 3.1.14 y 3.1.15 respectivamente).

Diagrama de Secuencia Marcas Caso Especializado: Salir catálogo de marcas

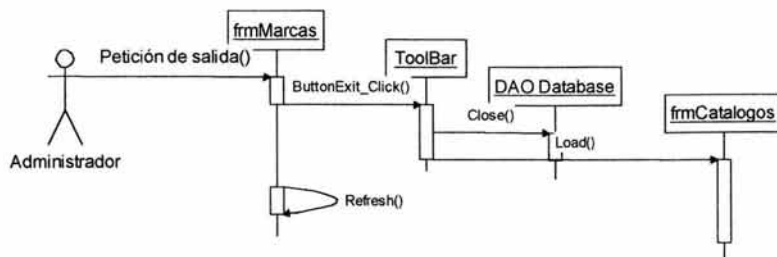
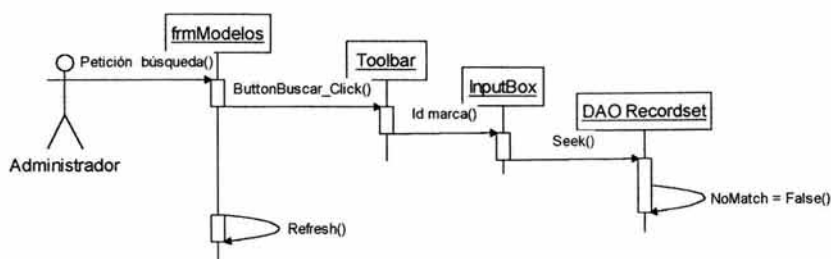


Diagrama de Secuencia Modelos Caso Especializado: Buscar Modelo



Caso Alternativo: Buscar Modelo

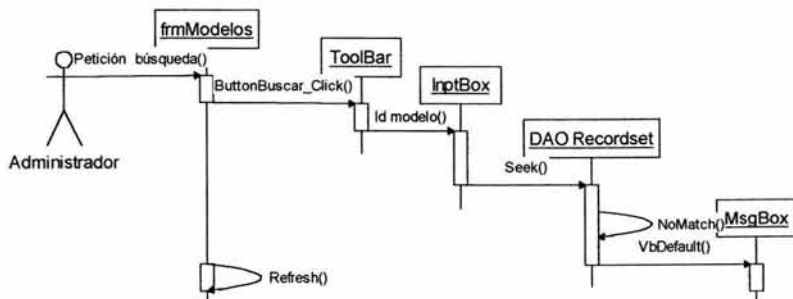


Figura 4.3.11

La figura 4.3.11 describe el proceso del sistema para salir del control del Catálogo de Marcas. También especifica la implementación física del caso de uso Buscar modelo así como de su caso alternativo (Ver páginas 43 tabla 3.1.17).

Diagrama de Secuencia Modelos
Caso Especializado: Imprimir Modelos

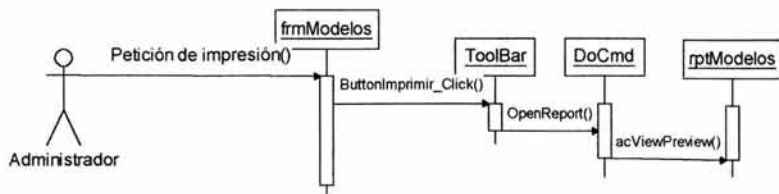
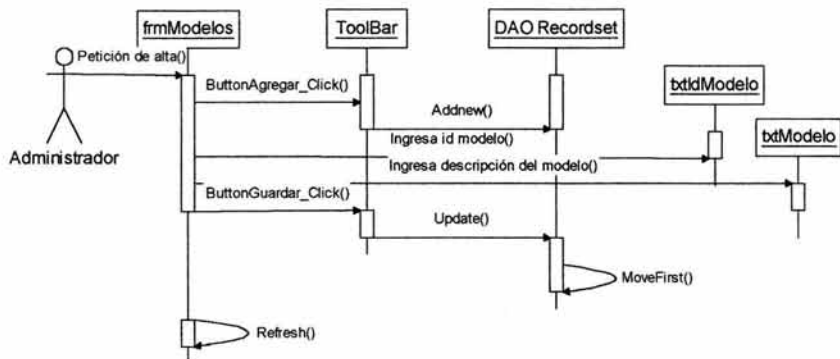


Diagrama de Secuencia Modelos
Caso Especializado: Altas de Modelos



Caso Alternativo: Altas de Modelos

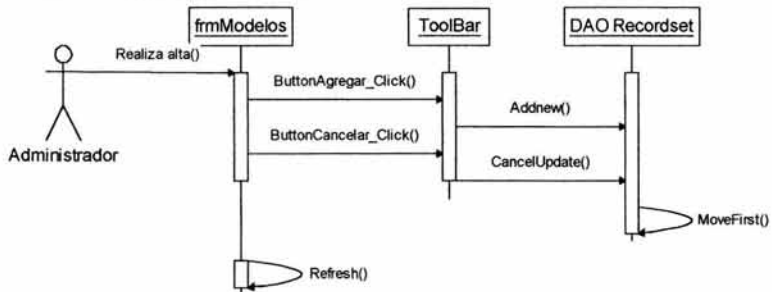
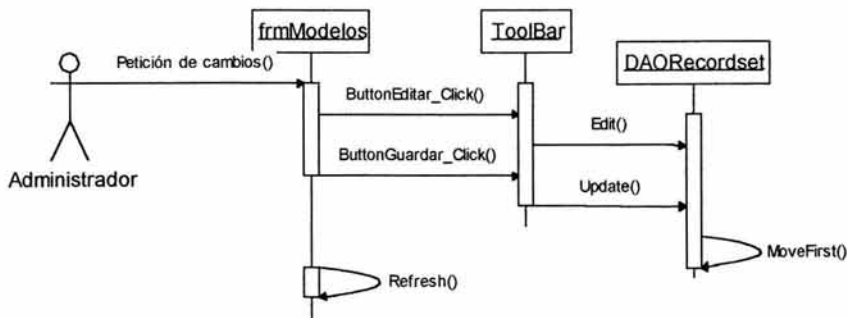


Figura 4.3.12

La figura 4.3.12 describe las especializaciones del Caso de uso: Actualizar Modelos. Imprimir catálogo de modelos, Alta de modelos y su caso alternativo (Ver página 43 tablas 3.1.18 y 3.1.19 respectivamente).

Diagrama de Secuencia Modelos

Caso Especializado: Cambios de Modelos



Caso alterno: Cambios de Modelos

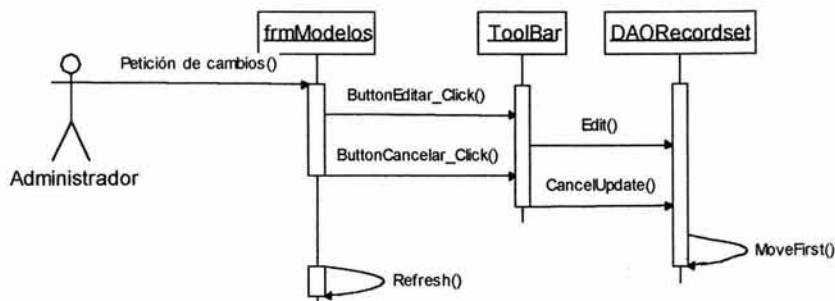


Diagrama de Secuencia Modelos

Caso Especializado: Bajas de Modelos

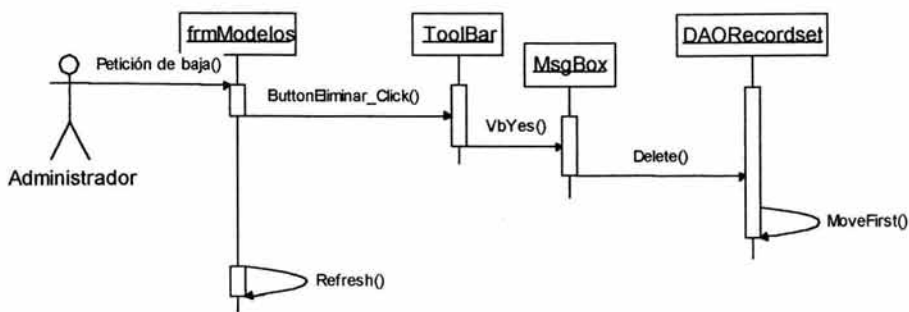


Figura 4.3.13

La figura 4.3.13 describe las especializaciones del Caso de uso: Actualizar Modelos. Modificar descripción de modelo su caso alterno y Baja de modelos (Ver página 44 tablas 3.1.20 y 3.1.21 respectivamente).

Caso Alterno: Bajas de Modelos

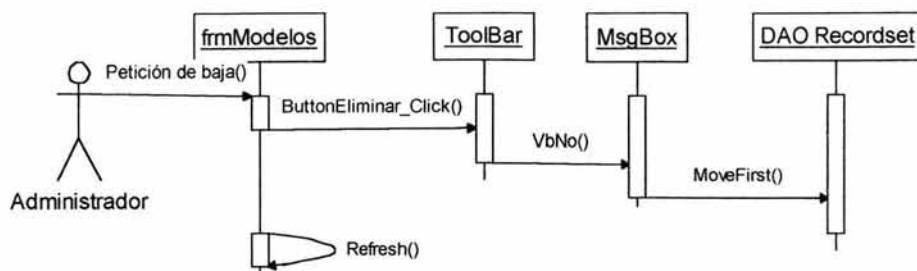


Diagrama de Secuencia Modelos

Caso Especializado: Salir del catálogo Modelos

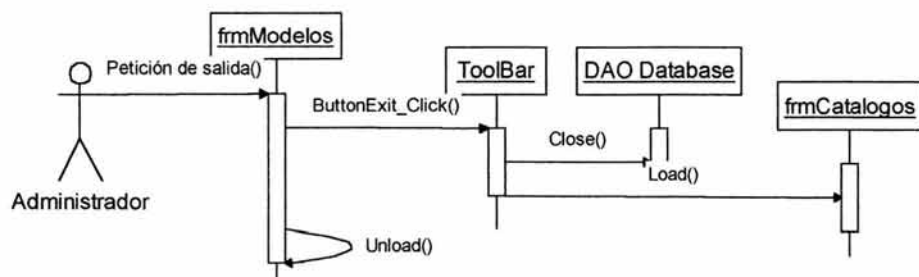
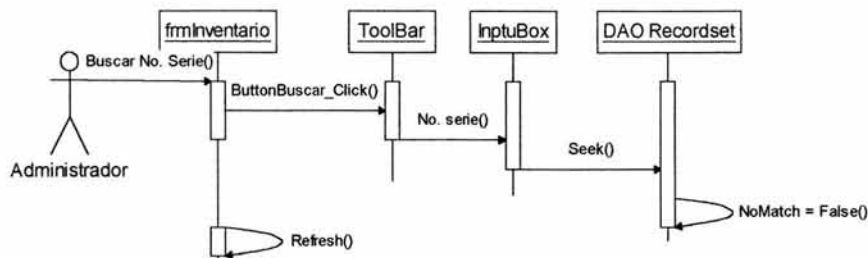


Figura 4.3.14

La figura 4.3.14 describe las especializaciones del Caso de uso: Actualizar Modelos. Baja de modelos (Ver página 44 tabla 3.1.21). El diagrama de secuencia Salir del catálogo de Modelos no representa propiamente un caso de uso, pero es un proceso disponible en la implementación del sistema.

Diagrama de Secuencia Actualizar Inventario Informático Caso Especializado: Buscar Equipo informático



Caso Alternativo: Buscar Equipo informático

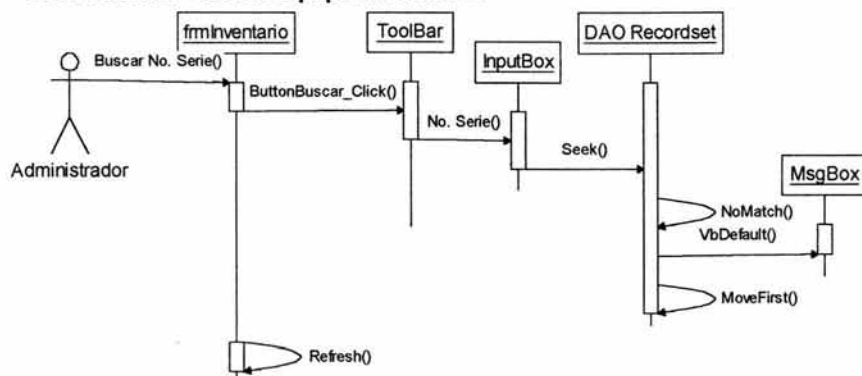


Diagrama de Secuencia Actualizar Inventario Informático Caso Especializado: Imprimir

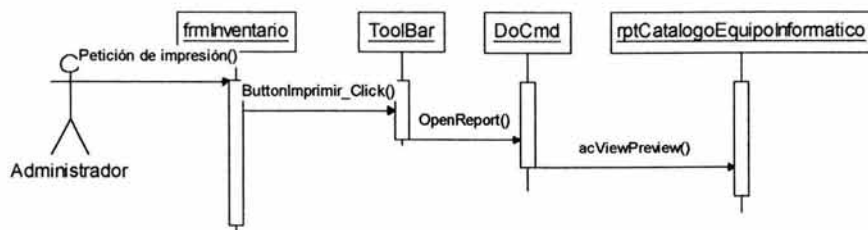
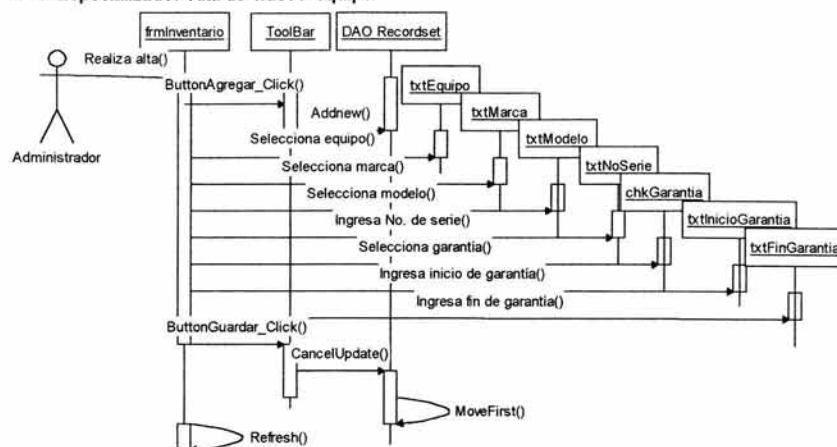


Figura 4.3.15

La figura 4.3.15 describe las especializaciones del Caso de uso: Actualizar Inventario Informático. Buscar equipo su caso alternativo e Imprimir inventario informático (Ver página 45 tablas 3.1.22, 3.1.23 y 3.1.24 respectivamente).

Diagrama de Secuencia Actualizar Inventario Informático
Caso Especializado: Alta de nuevo equipo



Caso alternativo

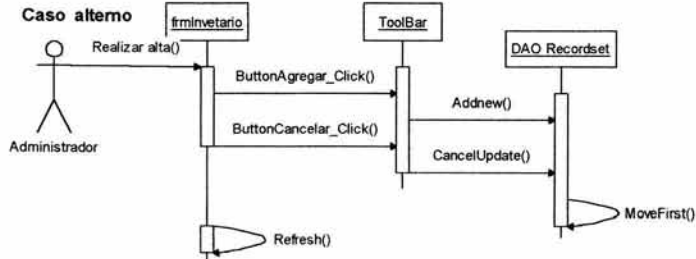


Diagrama de Secuencia Actualizar Inventario Informático
Caso Especializado: Cambios en Inventario

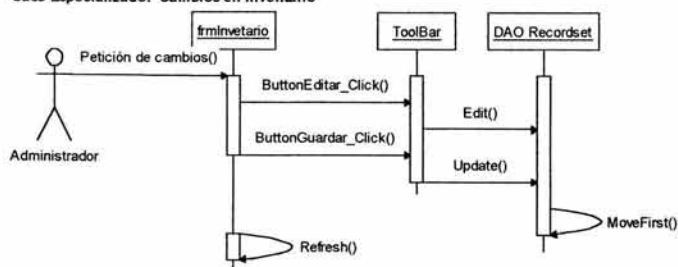


Figura 4.3.16

La figura 4.3.16 describe las especializaciones del Caso de uso: Actualizar Inventario Informático. Ingresar equipo informático su caso alternativo y Modificar descripción de equipo. (Ver páginas 45 y 46 tablas 3.1.25 y 3.1.26 respectivamente).

Caso Alterno: Cambios en Inventario

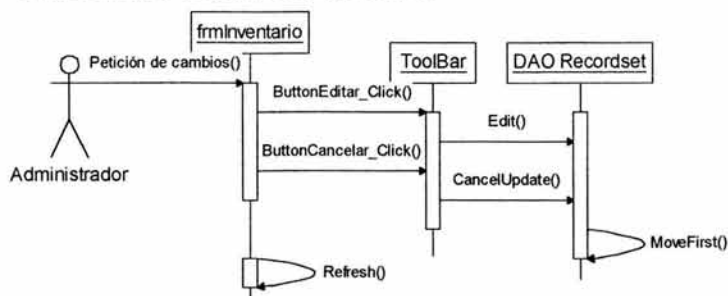
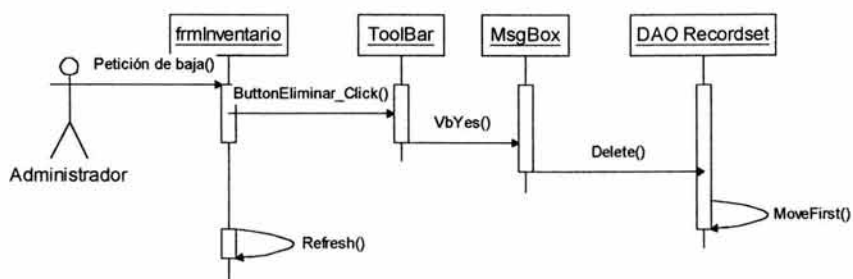


Diagrama de Secuencia Actualizar Inventario Informático Caso Especializado: Bajas en el Inventario



Caso Alterno: Bajas en Inventario

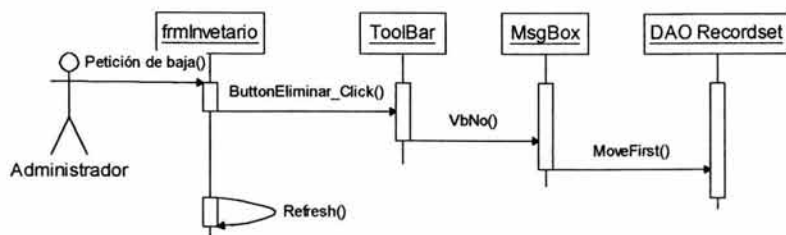


Figura 4.3.17

La figura 4.3.17 describe las especializaciones del Caso de uso: Actualizar Inventario Informático. Caso alternativo Modificar descripción de equipo y Baja de equipos, así como su caso alternativo. (Ver páginas 46 tablas 3.1.26 y 3.1.27 respectivamente).

Diagrama de Secuencia Actualizar Inventario Informático
Caso Especializado: Salir de Inventario

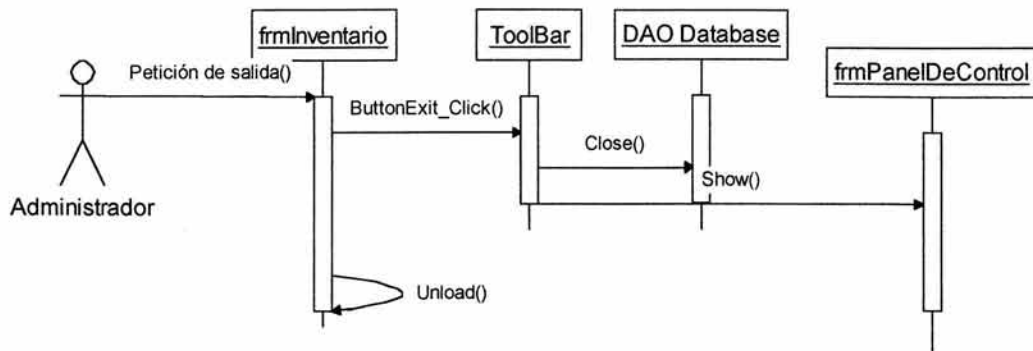
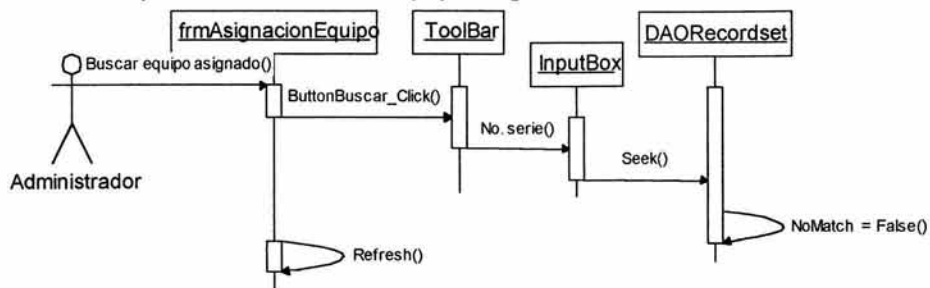


Figura 4.3.18

La figura 4.3.18 no representa ningún caso de uso en específico pero es un proceso dentro del sistema que permite al usuario del sistema, salir del control del inventario.

Diagrama de Secuencia Asignar equipo informático Caso Especializado: Buscar equipo asignado



Caso Alternativo: Buscar equipo asignado

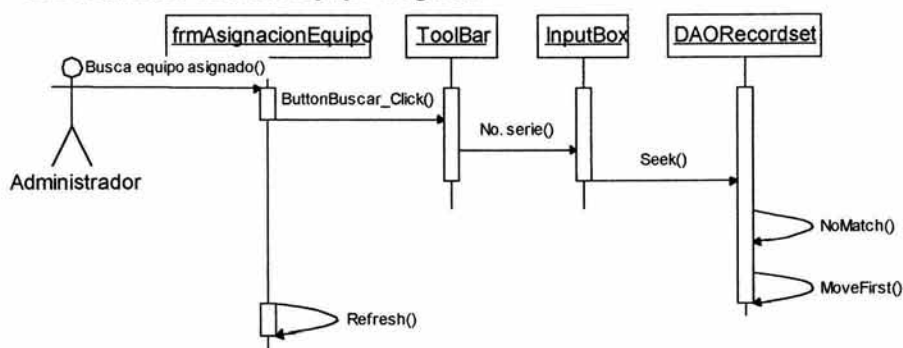


Diagrama de Secuencia Asignar equipo informático Caso Especializado: Imprimir

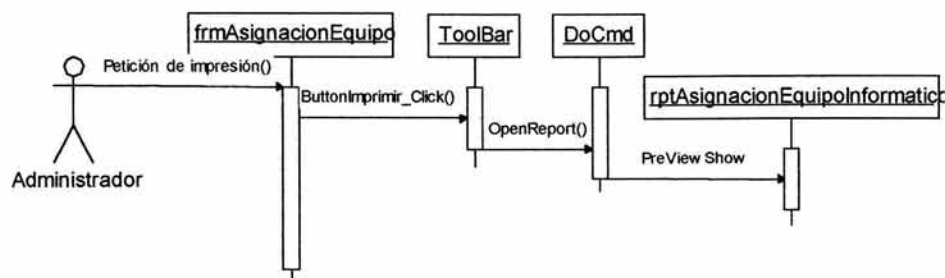
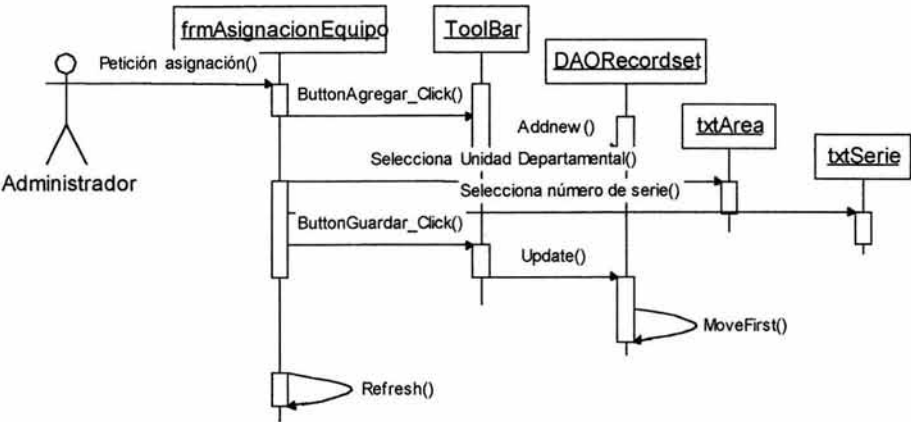


Figura 4.3.19

La figura 4.3.19 describe las especializaciones del Caso de uso: Asignar Equipo Informático. Buscar equipo asignado su caso alternativo e Imprimir equipo asignado. (página 47 tablas 3.1.29 y 3.1.30 respectivamente).

Diagrama de Secuencia Asignar equipo informático
Caso Especializado: Asignar equipo



Caso Alternativo: Asignar equipo

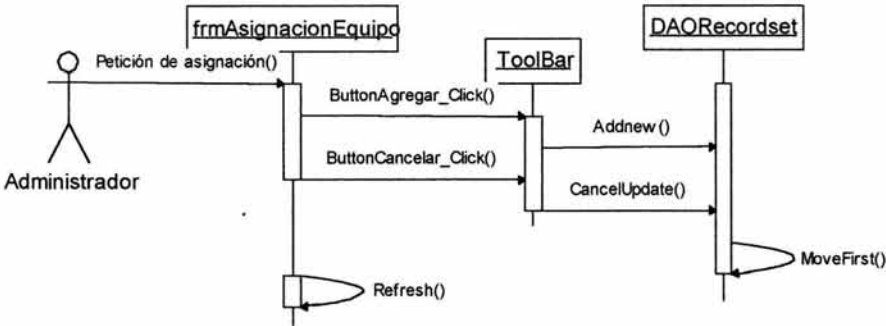
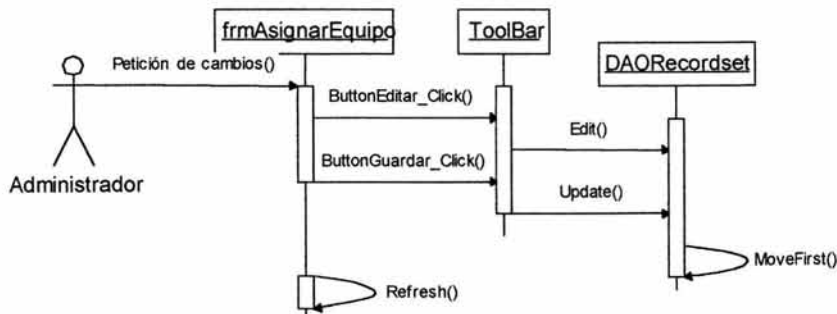


Figura 4.3.20

La figura 4.3.20 describe las especializaciones del Caso de uso: Asignar equipo informático y su caso alternativo (Ver página 48 tabla 3.1.31)

Diagrama de Secuencia Asignar equipo informático Caso Especializado: Modificar equipo asignado



Caso Alternativo: Modificar equipo asignado

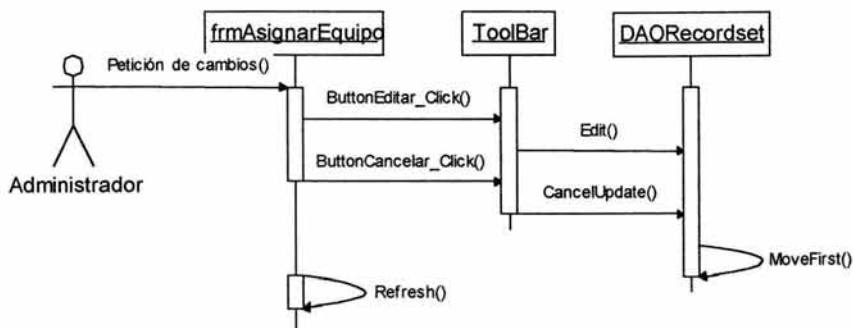


Diagrama de Secuencia Asignar equipo informático Caso Especializado: Bajas de equipo asignado

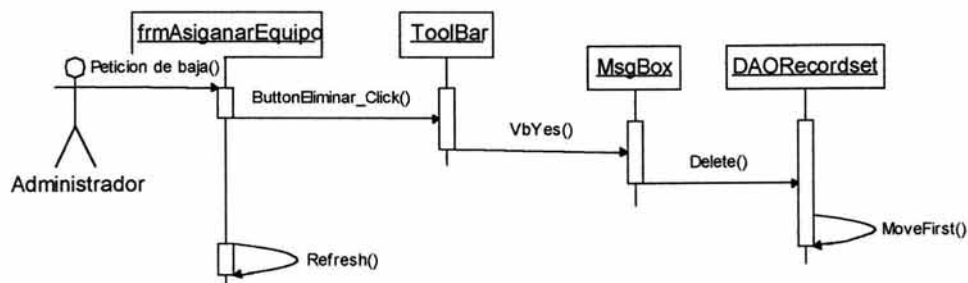


Figura 4.3.21

La figura 4.3.21 describe las especializaciones del Caso de uso: Asignar equipo informático. Modificar equipo asignado su caso alternativo y Eliminar equipo asignado (Ver página 48 tablas 3.1.32 y 3.1.33 respectivamente).

Caso Alterno: Baja de Equipo asignado

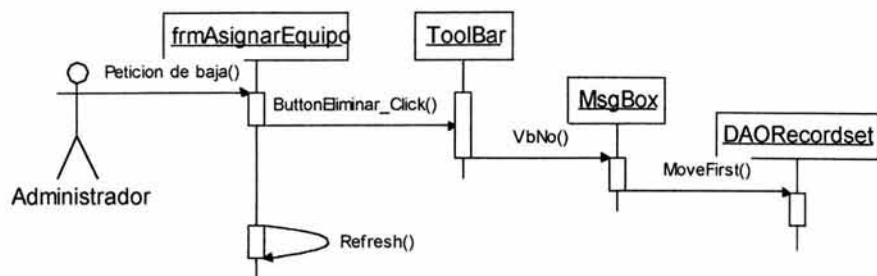


Diagrama de Secuencia Asignar equipo informático Caso Especializado: Salir del equipo asignado

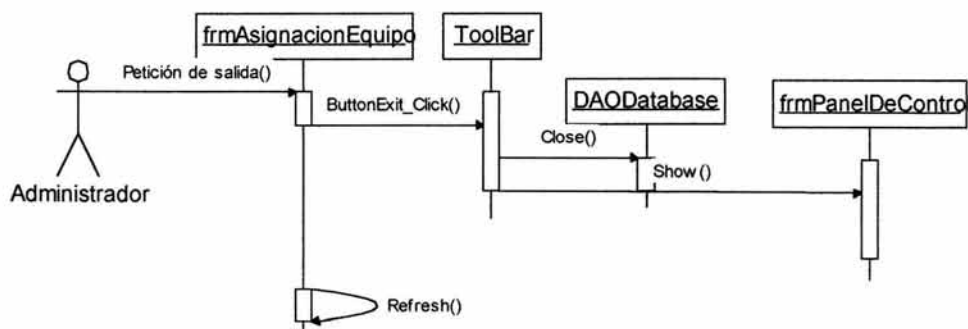
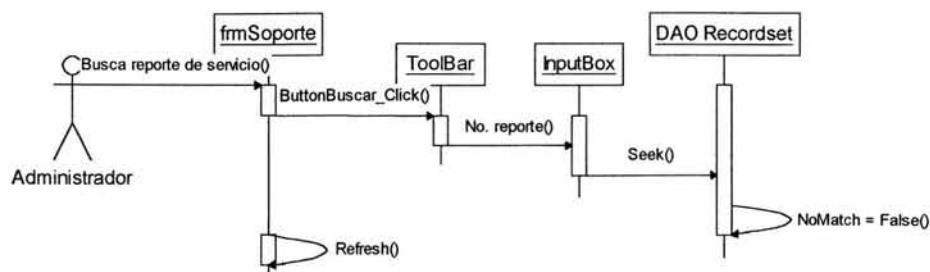


Figura 4.3.22

La figura 4.3.22 describe la especialización del Caso de uso: Asignar equipo informático. Eliminar equipo asignado (Ver página 48 y 49 tablas 3.1.33). El diagrama de secuencia Salir del equipo asignado es el proceso que nos permite concluir con todas las actividades en las asignaciones de equipos.

Diagrama de Secuencia Prestar Soporte Técnico **Caso Especializado: Buscar reporte de servicio**



Caso Alternativo: Buscar reporte de servicio

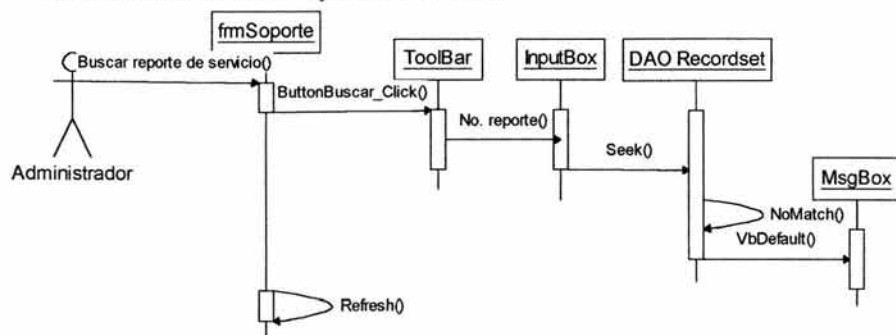


Diagrama de Secuencia Prestar Soporte Técnico **Caso Especializado: Imprimir reporte de servicio**

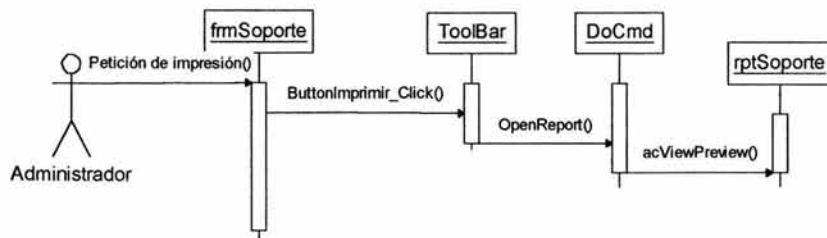
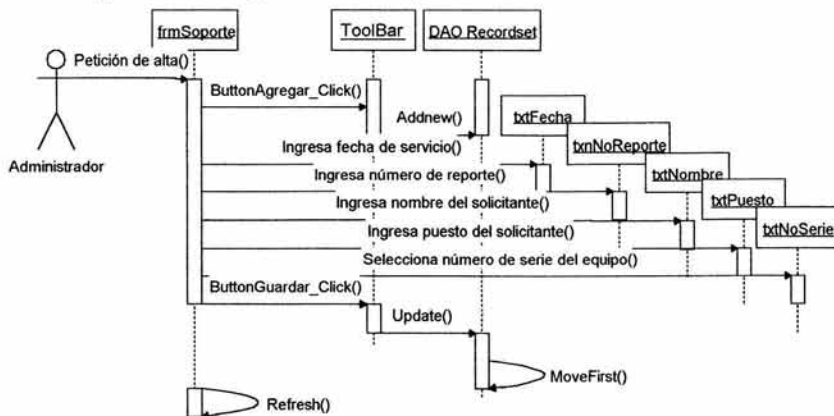


Figura 4.3.23

La figura 4.3.23 describe la especialización del Caso de uso: Prestar Soporte Técnico. Buscar reporte de servicio su caso alternativo e Imprimir reporte de servicio (Ver página 49 tabla 3.1.35 y 3.1.36 respectivamente).

Diagrama de Secuencia Prestar Soporte Técnico Caso Especializado: Ingresar nuevo reporte de servicio



Caso Alternativo: Ingresar reporte de servicio

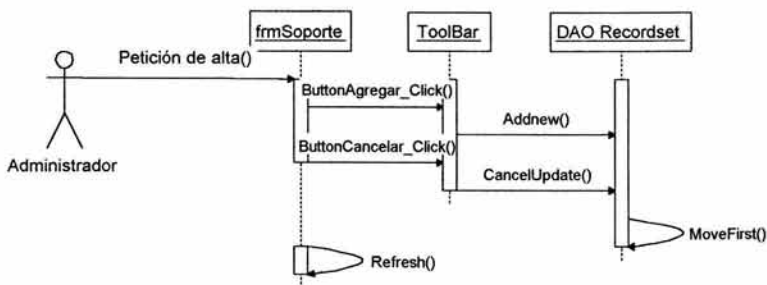


Diagrama de Secuencia Prestar Soporte Técnico Caso Especializado: Modificar reporte de servicio prestado

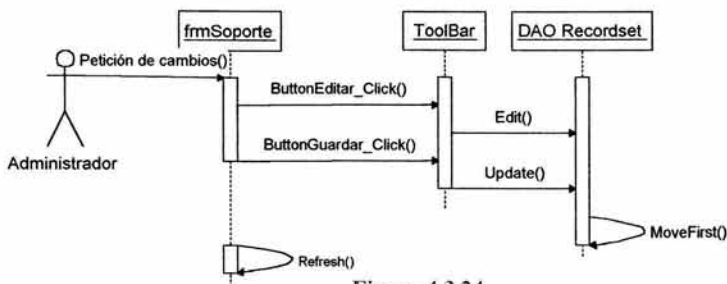


Figura 4.3.24

La figura 4.3.24 describe la especialización del Caso de uso: Prestar Soporte Técnico. Ingresar reporte de servicio su caso alternativo y Modificar reporte de servicio (Ver páginas 49 y 50 tablas 3.1.37 y 3.1.38 respectivamente).

Caso Alterno: Modificar reporte de servicio prestado

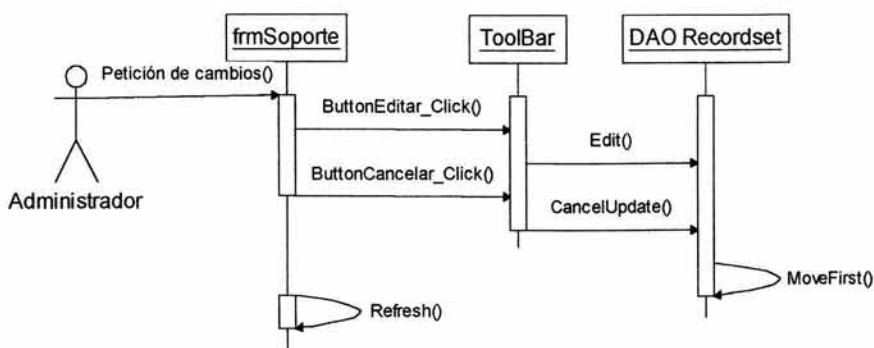
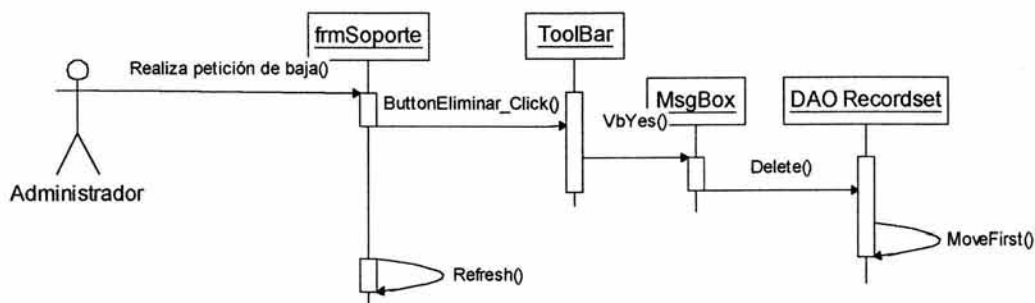


Diagrama de Secuencia Prestar Soporte Técnico Caso Especializado: Baja de reporte



Caso Alterno: Baja de reporte

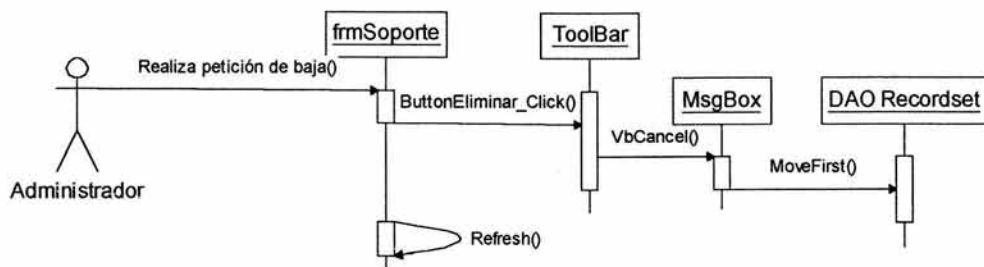


Figura 4.3.25

La figura 4.3.25 describe la especialización del Caso de uso: Prestar Soporte Técnico. Caso alternativo de Modificar reporte de servicio, Eliminar reporte de servicio y su caso alternativo (Ver páginas 50 y 51 tablas 3.1.38 y 3.1.39 respectivamente).

Diagrama de Secuencia Prestar Soporte Técnico

Caso Especializado: Salir de soporte técnico

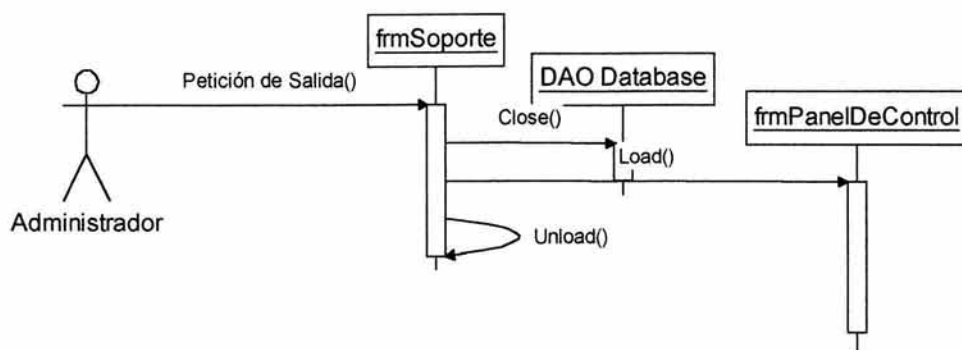


Figura 4.3.26

La figura 4.3.26 especifica el proceso de salida del control de soporte técnico.

Diagrama de Secuencia Salir de la aplicación Caso Especializado: Cerrar sesión

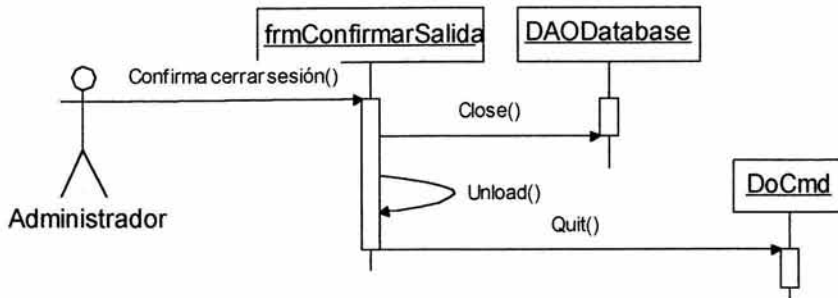


Diagrama de Secuencia Salir de la aplicación Caso Especializado: Cerrar sesión

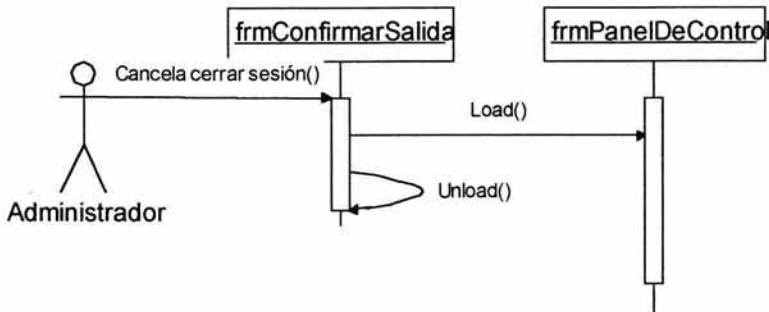


Figura 4.3.27

La figura 4.3.27 especifica el proceso a ejecutar para poder salir de la aplicación de Control de Soporte Técnico. Los diagramas de secuencia representan los dos posibles casos Confirmar y Cancelar.

Diagrama de Secuencia Asignar equipo informático
Caso Especializado: Consulta de equipo asignado por Área

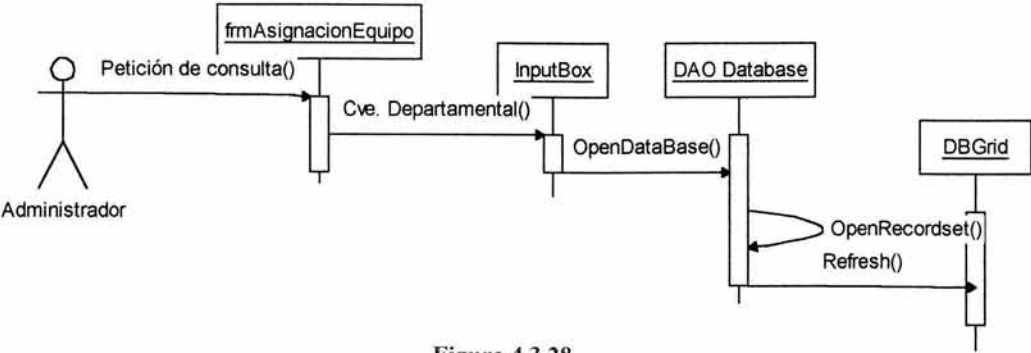


Figura 4.3.28

La figura 4.3.28 especifica el proceso a ejecutar el proceso de consulta de los equipos asignados por área.

CAPÍTULO V

MODELO DE IMPLEMENTACIÓN
DEL NUEVO SISTEMA INTEGRAL
DE SOPORTE TÉCNICO EN LA
DELEGACIÓN GAM

5.1 EL MODELO DE IMPLEMENTACIÓN

En la implementación se toma como base el resultado del modelo de diseño y se implanta el sistema en términos de componentes, es decir ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

La mayor parte de la arquitectura del sistema es capturada durante el diseño, siendo el propósito principal de la implementación el desarrollar la arquitectura y el sistema como un todo. De forma más específica los propósitos de la implementación son:

- Planificar las integraciones de sistema necesarias en cada iteración. Seguimos para ello un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.
- Distribuir el sistema asignado componentes ejecutables a nodos en el diagrama de despliegue. Esto se basa fundamentalmente en las clases activas encontradas durante el diseño.
- Implementar las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementan como componentes de fichero que contienen código fuente.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema.

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizados, y cómo dependen los componentes unos de otros.

COMPONENTES

Un componente es un empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Algunos estereotipos estándares de componentes son:

- <<ejecutable>> es un programa que puede ser ejecutado en un nodo.
- <<File>> es un fichero de código fuente.
- <<library>> es una librería estática o dinámica.
- <<table>> es una tabla de base de datos.
- <<document>> es un documento.

Durante la creación de componentes en un entorno de implementación particular estos estereotipos pueden ser modificados para reflejar el significado real de estos componentes.

Los componentes tienen las siguientes características:

- Tienen relaciones de traza con los elementos de modelo que implementan.
- Es normal que un componente implemente varios elementos, por ejemplo, varias clases, sin embargo, la forma exacta en que se crea esta traza depende de cómo van a ser estructurados y modularizados los ficheros de código fuente, dado el lenguaje de programación que se esté usando.
- Proporcionan las mismas interfaces que los elementos de modelo que lo implementan.
- Puede haber dependencias de compilación entre componentes, denotando qué componentes son necesarios para compilar un componente determinado.

5.2 POLÍTICAS Y ESTÁNDARES DE PROGRAMACIÓN EN LA DELEGACIÓN GAM

A continuación se definen los estándares de programación que se deberán seguir por los equipos de trabajo durante el desarrollo de sistemas en la Coordinación de Informática de la Delegación Gustavo A. Madero.

El establecer convenciones de nomenclatura facilita al equipo de desarrollo a:

- Estandarizar la estructura, el estilo de codificación y la lógica de una aplicación.
- Crear código, preciso, legible y sin ambigüedades.
- Ser consistente con convenciones utilizadas en otros lenguajes.
- Definir nombres de objetos de una manera más comprensible.

El nombre de cada proyecto deberá de contar con un nombre que identifique el objetivo del mismo y en caso de que conste de 8 caracteres se formará como un acrónimo de fácil comprensión y que identifique el objetivo del módulo principal que contiene.

NOMENCLATURA

Objetos

Un objeto es una combinación de código y datos que será tratado como una unidad, como pueden ser los siguientes:

Controles	ctl
Formas	frm
Menús	mnu
Módulos	Mod
Reportes	rpt

El nombre de los controles está compuesto de dos partes: el prefijo y el nombre mismo.

<Prefijo><Nombre>

Prefijo:

El prefijo en el nombre del objeto sirve para determinar fácilmente el tipo de control que se utiliza, de esa forma se puede conocer más fácilmente las propiedades y los eventos a los que puede responder.

El prefijo normalmente es de tres caracteres en minúscula.

Cuando se tiene controles comunes no se hace diferencia en el prefijo, esto es con el fin de no complicar los estándares. Por ejemplo si se tienen botones que desplieguen gráficos ó de barra de herramientas, a todos se les asigna el prefijo cmd.

Nombre:

El nombre del objeto se especifica con un mnemónico con el uso o función del mismo. Los nombres de los objetos que realizan una acción, como los botones, incluyen el verbo en infinitivo de la acción que realizan.

El nombre siempre irá en singular a menos de que se trate de un control que pueda tener más de un dato, como serían las listas, grids, combos, etc.

Tabla 5.2 Nomenclatura para Objetos Estándar.

Prefijo	Tipo de Objeto	Ejemplo
ani	Botón Animado	aniMailBox
bed	Pen Bedit	bedFirstName
cbo	Cuadro Combinado y Lista Desplegable	cboEnglish
chk	Casilla de Verificación	chkReadOnly
clp	Clip de Imagen	clpToolbar
cmd (3d)	Botón de Comando (3D)	cmdOk (cmd3dOk)
com	Comunicaciones	comFax
ctr ctl	Control (de tipo no específico)	ctrCurrent (ctlCurrent)
dat	Control de Datos	datBiblio
dir	Cuadro de Lista de Directorios	dirSource
dlg	Diálogo Común	dlgFileOpen
drv	Cuadro de Lista de Unidades	drvTarget
fil	Cuadro de Lista de Archivos	filSource
frm	Formulario	frmEntry
fra (3d)	Marco (3d)	fraStyle (fra3dStyle)
gau	Medidor	gauStatus
gpb	Grupo de Botones Presionables	gpbChannel
gra	Gráfica	graRevenue
grd	Rejilla	grdPrices
hed	Pen Hedit	hedSignature
hsb	Barra de Desplazamiento Horizontal	hsbVolume
img	Imagen	imgIcon
ink	Trazo de Pluma	inkMap
key	Estado de Tecla	keyCaps
lbl	Etiqueta	lblHelpMessage
lin	Línea	linVertical
lst	Cuadro de Lista	lstPolicyCodes
mdi	Formulario MDI hijo	mdiNote
mpm	Mensaje MAPI	mpmSendMessage
mps	Sesión MAPI	mpsSession
mci	MCI	mciVideo
mnu	Menú	mnuFileOpen
opt (3d)	Botón de Opción (3d)	optRed (opt3dRed)
ole	Control OLE	oleWorksheet
out	Esquema	outOrgChart
pic	Picture	picVGA
pnl3d	Panel 3d	pnl3d
rpt	Reporte	rptQtr1Earnings
shp	Control de Formas	shpCircle
spn	Control Spin	spnPages
txt	Cuadro de Texto	txtLastName
tmr	Cronómetro	tmrAlarm
vsb	Barra de Desplazamiento Vertical	vsbRate

Tabla 5.2.1 Nomenclatura para Objetos de Base de Datos.

Prefijo	Tipo de Objeto	Ejemplo
con	Contenedor	conAccounts
db	Base de datos	dbAccounts
dbe	Motor de Base de datos	
ds	Dynaset	dsSalesByRegion
fldc	Colección de campos	fdcCustomer
fld	Campo	fdAddress
idx	Índice	ixAge
idxc	Colección de índices	ixcNewAge
qd	Definición de Consulta (QueryDef)	qdSalesByRegion
qry	Consulta	qrySalesByRegion
rst	Recordset	rstForecast
ss	Snapshot	ssForecast
tb	Tabla	tbCustomer
td	Definición de Tabla (TableDef)	tdCustomers
wsp	Espacio (Workspace)	

Nomenclatura para Menús.

Las aplicaciones utilizan frecuentemente y en abundancia controles de menú por lo que se requiere de una nomenclatura especial para estos. El prefijo deberá extenderse más allá del "mnu" inicial añadiéndole un prefijo por cada nivel de anidamiento que identifique su trayectoria, siendo la parte final de la cadena la que identifique la acción. Por ejemplo:

Tabla 5.2.2	Secuencia del Menú	Nombre del Punto del Menú
	Help.Contents	mnuHelpContents
	File.Open	mnuFileOpen
	Format.Character	mnuFormatCharacter
	File.Send.Fax	mnuFileSendFax
	File.Send.Email	mnuFileSendEmail

Cuando se utiliza esta convención, todos los miembros de un grupo de menú particular son listados juntos, adicionalmente de saberse claramente de que punto de menú dependen.

Nomenclatura para Controles Adicionales.

Para controles no listados se debe de tratar de formar un prefijo único de tres caracteres. De la misma forma, cuando se utilicen controles derivados, como los cuadros de lista mejorados, se debe de ampliar la lista de prefijos de tal manera que no haya confusión acerca del tipo de control utilizado.

Cada control adicional o de terceros utilizado en una aplicación debe de ser listado en la sección inicial de comentarios, proporcionando el prefijo signado a ese control y el nombre del fabricante.

Tabla 5.2.3	Fabricante	Abreviatura
	MicroHelp (VBTools)	m
	Pioneer Software	p
	Crescent Software	c
	Sheridan Software	s
	Other (Misc)	o

A continuación se listan algunos de los principales controles de proveedores diferentes de Microsoft y sus prefijos.

Tabla 5.2.4

Tipo de Control	Nombre	Abreviatura	Fabricante	Ejemplo	Archivo
Alarm	Alarm	almm	MicroHelp	almmAlarm	MHTI200.VBX
Animate	Animate	anim	MicroHelp	animAnimate	MHTI200.VBX
Callback	Callback	calm	MicroHelp	calmCallback	MHAD200.VBX
Combo Box	DB_Combo	cbop	Pioneer	cbopComboBox	QEVBDDBF.VBX
Combo Box	SSCombo	cbos	Sheridan	cbosComboBox	SS3D2.VBX
Check Box	DB_Check	chkp	Pioneer	chkpCheckBox	QEVBDDBF.VBX
Chart	Chart	chtm	MicroHelp	chtmChart	MHGR200.VBX
Clock	Clock	clkm	MicroHelp	clkmClock	MHTI200.VBX
Button	Command	cmdm	MicroHelp	cmdmCommandButton	MHEN200.VBX
Button	Button				
Button	DB_Command	cmdp	Pioneer	cmdpCommandButton	QEVBDDBF.VBX
Button (Group)	Command	cmgm	MicroHelp	cmgmBtton	MHGR200.VBX
Button	Button (multiple)				
Button	Command	cmim	MicroHelp	cmimCommandButton	MHEN200.VBX
Button	Button (icon)				
CardDeck	CardDeck	crdm	MicroHelp	crdmCard	MHGR200.VBX

Dice	Dice	dicm	MicroHelp	dicmDice	MHGR200.VBX
List Box (Dir)	SSDir	dirs	Sheridan	dirsDirList	SS3D2.VBX
List Box (Drv)	SSDrive	drvs	Sheridan	drvsDriveList	SS3D2.VBX
List Box (File)	File List	film	MicroHelp	filmFileList	MHEN200.VBX
List Box (File)	SSFile	fls	Sheridan	flsFileList	SS3D2.VBX
Flip	Flip	flpm	MicroHelp	flpmButton	MHEN200.VBX
Scroll Bar	Form Scroll	fsrm	MicroHelp	fsrmFormScroll	???
Gauge	Gauge	gagm	MicroHelp	gagmGauge	MHGR200.VBX
Graph	Graph	gpho	Other	gphoGraph	XYGRAPH.VBX
Grid	Q_Grid	grdp	Pioneer	grdpGrid	QEVBDDBF.VBX
Scroll Bar	Horizontal	hsbm	MicroHelp	hsbmScroll	MHEN200.VBX
	Scroll Bar				
Scroll Bar	DB_HScroll	hsbp	Pioneer	hsbpScroll	QEVBDDBF.VBX
Graph	Histo	hstm	MicroHelp	hstmHistogram	HGR200.VBX
Invisible	Invisible	invn	MicroHelp	invnInvisible	MHGR200.VBX
List Box	Icon Tag	itgm	MicroHelp	itgmListBox	MHAD200.VBX
Key State	Key State	kstm	MicroHelp	kstmKeyState	MHTI200.VBX
Label	Label (3d)	lblm	MicroHelp	lblmLabel	MHEN200.VBX
Line	Line	linm	MicroHelp	linmLine	MHGR200.VBX
List Box	DB List	lstp	Pioneer	lstpListBox	QEVBDDBF.VBX
List Box	SSList	lsts	Sheridan	lstsListBox	SS3D2.VBX
MDI Child	MDI Control	mdcm	MicroHelp	mdcmMDIChild	???
Menu	SSMenu	mnus	Sheridan	mnusMenu	SS3D3.VBX
Marque	Marque	mrqm	MicroHelp	mrqmMarque	MHTI200.VB
Picture	OddPic	odpm	MicroHelp	odpmPicture	MHGR200.VBX
Picture	Picture	picm	MicroHelp	picmPicture	MHGR200.VBX
Picture	DB_Picture	picp	Pioneer	picpPicture	QEVBDDBF.VBX
Property Vwr	Property	pvrn	MicroHelp	pvrnPropertyViewer	MHPR200.VBX
	Viewer				
Option (Group)	DB_RadioGroup	radp	Pioneer	radqRadioGroup	QEVBDDBF.VBX
Slider	Slider	sldm	MicroHelp	sldmSlider	MHGR200.VBX
Button (Spin)	Spinner	spnm	MicroHelp	spnmSpinner	MHEN200.VBX
Spreadsheet	Spreadsheet	sprm	MicroHelp	sprmSpreadsheet	MHAD200.VBX
Picture	Stretcher	strm	MicroHelp	strmStretcher	MHAD200.VBX
Screen Saver	Screen Saver	svrm	MicroHelp	svrmSaver	MHTI200.VBX
Switcher	Switcher	swtm	MicroHelp	swtmSwitcher	???
List Box	Tag	tagm	MicroHelp	tagmListBox	MHEN200.VBX
Timer	Timer	ttrm	MicroHelp	ttrmTimer	MHTI200.VBX
ToolBar	ToolBar	tolm	MicroHelp	tolmToolBar	MHAD200.VBX
List Box	Tree	trem	MicroHelp	tremTree	MHEN200.VBX
Input Box	Input (Text)	txtm	MicroHelp	inpmText	MHEN200.VBX
Input Box	DB_Text	txtp	Pioneer	txtpText	QEVBDDBF.VBX
Scroll Bar	Vertical	vsbm	MicroHelp	vsbmScroll	MHEN200.VBX
	Scroll Bar				

Nomenclatura de Variables y Rutinas.

Los nombres de las variables y rutinas deberán de tener la siguiente estructura:

<Prefijo><Nombre><Calificador>

Tabla 5.2.5

Sección	Descripción	Ejemplo
<prefijo>	Describe el uso y alcance de la variable	iGetRecordNext
<nombre>	Describe la variable	iGetNameFirst
<calificador>	Denota un calificador de la variable.	iGetNameLast

Prefijos:

La tabla siguiente define los prefijos de los nombres de variables en base a la notación C Húngara.

Tabla 5.2.6

Prefijo	Abreviado	Uso de la Variable	Tipo de Datos
bln	b	Lógico	Entero
cur	c	Moneda (64 bits)	Moneda
dbl	d	Doble, cantidad con signo (64 bits)	Doble
dat	dt	Fecha y Hora	Variant
err	e	Error	
sng	f	Flotante, cantidad con signo (32 bits)	Sencillo
idx	i	Index	Entero
h	h	Handle	Entero
lng	l	Largo, cantidad con signo (32 bits)	Largo
int	n	Número/Contador	Entero
str	s	Cadena	Alfanumérico
u	u	Sin signo (16 bits)	Largo
udt		Tipo definido por el usuario	
vnt	vnt	Variant	Variant
arr	a	Arreglo	Arreglo

Alcance y uso de prefijos:

Tabla 5.2.7	Prefijo	Descripción
	g	Global
	m	Local al módulo, formulario o reporte
	st	Estática
	(sin prefijo)	Variable No-estática
	v	Variable pasada por valor(local a la rutina)
	r	Variable pasada por referencia(local a la rutina)

Nombre de la variable o rutina:

Cuando el nombre se encuentra compuesto de varias palabras, cada una de ellas comienza con mayúscula (su primera letra).

Para términos usados frecuentemente ó términos muy largos, se podrán usar abreviaturas para mantener los nombres de un largo razonable (hasta 32 caracteres aproximadamente). Cuando se usen abreviaturas hay que tener la precaución de ser consistente en las mismas a todo lo largo del proyecto, ya que el cambiarlas tiende a causar confusión, por ejemplo: si se usa indistintamente cnt ó cont para describir contador.

Cuando una variable guarda una colección de elementos, su nombre esta dado en plural, pero si sólo guarda un elemento su nombre esta dado en singular y deberá ser tan largo como sea necesario para describir su propósito.

Adicionalmente, los nombres de función deben de comenzar con un verbo, por ejemplo: InicializarNombreArreglo o CerrarDialogo.

Calificador:

Las variables y rutinas a menudo son usadas para administrar y manipular un objeto común. En estos casos, se usan calificadores estándar para etiquetar las variables y rutinas derivadas. Aunque el poner el calificador al final parezca un poco extraño, esta práctica ayudará a ordenar los nombres haciendo la lógica y estructura de la aplicación más sencilla.

La siguiente tabla define calificadores comunes:

Tabla 5.2.8

Calificador	Descripción
Pri	Primer elemento de un conjunto.
Ult	Último elemento de un conjunto.
Sig	El elemento siguiente en un conjunto.
Ant	El elemento previo en un conjunto.
Act	El elemento actual en un conjunto.
Min	El valor mínimo de los elementos en el conjunto.
Max	El valor máximo de los elementos en el conjunto.
Save	Utilizada para conservar una variable que debe de ser reinicializada posteriormente.
Tmp	Una variable temporal, usada exclusivamente dentro del procedimiento.
Src	Origen, frecuentemente usada en comparaciones y rutinas de transferencia.
Dst	Destino, usada en conjunto con Src.

Tipos definidos por el usuario.

Declarar los tipos definidos por el usuario en mayúsculas agregando `_Type` al final del nombre, por ejemplo:

```
Type CUSTOMER_TYPE
    sName As String
    sState As String * 2
    lID as Long
End Type
```

Cuando se declara una instancia de una variable correspondiente a un tipo definido por el usuario se debe de agregar Nuevo o New al nombre de la variable, por ejemplo:

```
Dim custNew as CUSTOMER_TYPE
```

Nomenclatura de Constantes.

Por norma general todas las constantes definidas para un proyecto están declaradas en un solo lugar (un módulo en Visual Basic o su similar) para hacer el mantenimiento más sencillo. Invariablemente se deberán de utilizar constantes en lugar de codificar los valores correspondientes.

Los nombres de las constantes se escriben en mayúsculas, y si se usa más de una palabra se deberá separar utilizando guión bajo (`_`). Aunque no se acostumbra utilizar prefijos para definir el tipo y alcance de la constante es útil hacerlo.

CODIFICACIÓN.

Se debe de incluir una descripción de la aplicación, enumerando los objetos primarios de datos, las rutinas, algoritmos, diálogos, bases de datos y sistemas de archivos utilizados en el módulo inicial del sistema.

Los procedimientos de usuario pueden ser de tres tipos: Generales, Públicos o Privados. En cualquiera de los tres casos el nombre de los procedimientos está compuesto de dos partes: el prefijo y el nombre mismo.

```
<Prefijo><Nombre>
```

Prefijo:

El prefijo del módulo se refiere al tipo de información que va a contener. Si es uno que sólo contiene información general se le identifica módulo estándar, pero si es uno que contiene la definición de la clase de un objeto, se le identifica como un módulo de clase. Al tipo del

módulo le corresponderán tres caracteres en minúsculas y pueden ser alguno de los siguientes:

Tabla 5.2.9	<u>Prefijo</u>	<u>Descripción</u>
	std	Módulo Estándar
	cls	Módulo de clase

Nombre:

El nombre del módulo se especifica con un mnemónico con el uso o función del mismo. El nombre puede estar compuesto de más de una palabra para hacer más claro su uso.

Formateo del Código.

La descripción funcional de la rutina debe de iniciar indentada una posición y la primer instrucción posterior a la descripción debe de indentarse una posición del tabulador. Se debe de codificar utilizando indentación de bloques de código uniforme entre dos y cuatro espacios, basados en marcas del tabulador para mostrar la lógica del programa. Esto es, se debe de indentar:

- Después de una declaración *If* cuando se utiliza un *End If*
- Después de una declaración *Else*
- Después de una declaración *Select Case*
- Después de una declaración *Do*
- Líneas siguientes a una división de línea en que se utilice el carácter de continuación
- Después de una declaración *With*
- Después de llamar los métodos *Edit* o *AddNew* en un objeto *Recordset*. Los métodos *Update* y *CancelUpdate* deben de aparecer al mismo nivel de indentación que *Edit* o *AddNew*
- Después de una llamada al método *BeginTrans*
- En la sección de declaraciones de un módulo
- En el cuerpo de las declaraciones de tipos de datos definidos por el usuario
- En el código que este subordinado a una etiqueta

Adicionalmente se deberá dejar un renglón en blanco para agrupar declaraciones relacionadas, en los siguientes casos:

- Antes y después de cada *If...Then*
- Antes y después de cada *Select Case*
- Antes y después de cada *loop*
- Después de declarar un bloque de variables

Uso de Variables y Constantes.

Las variables y constantes no genéricas deben de agruparse por función en lugar de ser divididas en áreas aisladas o archivos especiales.

Es necesario utilizar la opción "Option Explicit" de tal forma que se requiera declarar todas las variables utilizadas, ahorrando tiempo en el desarrollo al reducir el número de problemas generados por errores mecanográficos.

Es recomendable utilizar el tipo de datos correspondiente a la información que se manipulará, ya que es mas eficiente su manejo. Sin embargo, los tipos *variant* son

sumamente útiles cuando se trabaja con bases de datos, mensajes, DDE ú Ole. Normalmente son utilizados en rutinas genéricas en las cuales las operaciones a realizar no necesitan saber el tipo de datos que recibe o pasa un proceso.

Siempre se debe de utilizar el ampersand (&) para concatenar cadenas y reservar el uso del signo más (+) para las operaciones con valores numéricos.

Las variables siempre deben de ser definidas con el menor alcance posible, en virtud de que el manejo de variables globales puede dificultar enormemente el entendimiento de la lógica de la aplicación.

Tabla 5.2.10

Alcance	Lugar de declaración	Visibilidad
Procedimiento	En un procedimiento, Sub o Función (usando Private).	En el procedimiento en que se declara
Formulario, Módulo	Sección de declaraciones de un formulario (Private).	En todos los procedimientos del formulario.
Global	Sección de declaraciones de un módulo (usando Public)	Siempre Visible

El uso de variables globales se debe de reservar para aquellos casos en que no exista una mejor manera de compartir la información entre formularios. En caso de definirse, se deberán de declarar todas las variables globales en un solo módulo y se les dará un nombre que sea significativo y describa su propósito.

Con excepción de las variables globales, los procedimientos y funciones deberán de operar únicamente con objetos pasados a ellas. Normalmente se utilizarán parámetros pasados ByVal, salvo en aquellos casos en que se requiere que sea modificado el valor del argumento pasado.

Comentarios del código.

Todos los métodos y módulos deben de empezar con un comentario describiendo las características funcionales de la rutina (lo que hace) y no deben de describir los detalles de implementación (cómo lo hace).

Los comentarios deben de ser comprensibles y se utilizarán frases completas sin abreviaturas.

Los comentarios pueden ser de:

- De encabezado
- Parámetro
- En línea

Comentarios de Encabezado

Los encabezados deben contener la siguiente información:

Tabla 5.2.11

Título	Descripción
Procedimiento	El nombre del método o módulo al que se le está asignando el comentario.
Objetivo	Se establecerá la función del método o módulo (que hace)
Autor	El nombre de la persona que lo realiza.
Fecha	La fecha de creación del módulo en formato aa-mm-dd
Entradas	Cada una de las parámetros, variables externas no-obvias, controles, archivos abiertos, etc.
Retorno	El valor que devuelve el método o módulo.
Efectos	La forma en que afecta a cada una de las variables externas, controles, archivos utilizados, etc.

Comentarios de Parámetros

Los comentarios de los parámetros del método deberán de indicar que es lo que hace cada uno de ellos y que valores pueden recibir. De la misma manera se deben de comentar los valores de retorno y variables globales cuyo valor es modificado dentro de la rutina.

Comentarios en línea

Los comentarios en línea están incluidos dentro del código mismo, su objetivo es el de clarificar la operación realizada dentro de las secciones del código.

Cada declaración de una variable no trivial debe de incluir un comentario que la describa.

Las variables, controles y rutinas debe de ser nombrado de una manera lo suficientemente explícita para que únicamente requieran ser comentadas aquellas que sean complejas o que su implementación sea clara.

Procedimientos y funciones.

Los procedimientos y funciones deben de ser específicas y autocontenidas, por ejemplo si se requiere acceder una base de datos y poner la información obtenida en una rejilla, se deberá de separar en dos funciones.

Se promoverá la reutilización del código.

Se deberán estructurar los procedimientos y funciones de tal manera que se reduzca la dependencia tratando de reducir los llamados a otras funciones y procedimientos.

Se deberá tener un solo punto de salida en cada procedimiento y funciones, facilitando la depuración del código.

Elementos de un Formulario.

Los formularios se dividen en tres secciones lógicas:

Sección Superior:

Esta sección medirá dos centímetros de alto y en ella se indicará el nombre del formulario en un cuadro de texto a todo lo largo del formulario, con fondo blanco, y utilizando letra Lucida Sans de 12 puntos en negritas.

Sección Central:

Esta sección será de un alto variable, dependiendo de la cantidad de controles que deba de mostrar.

Para las Etiquetas se utilizará letra Verdana de 8 puntos en negritas, se utilizará efecto de grabado, los bordes deberán de ser sólidos en color gris al 25% (12632256) y de un ancho 3 puntos.

En los demás controles (Cajas de Texto, CheckBox, OptionButton, ComboBox, ListBox, etc.) se utilizará letra MS Sans Serif de 8 puntos normal, con color del fondo blanco, se utilizará efecto de bajo relieve, los bordes deberán de ser sólidos en color negro y con trazo fino.

Sección Inferior:

Esta sección medirá dos centímetros de alto y en ella se mostrará la Barra de Navegación del Formulario y se indicará el nombre del objeto que contiene el formulario en un cuadro de texto transparente, en bajo relieve, bordes de color 0 y utilizando letra Arial de 7 puntos normal.

Alineación:

La alineación de las Etiquetas y para los Cuadros de texto será a la Izquierda, mientras que para los campos numéricos es a la derecha.

Por regla general únicamente el formulario correspondiente al Menú General mostrarán el logotipo de la delegación.

Los formularios deberán de tener los siguientes valores en las propiedades enunciadas:

Propiedad	Valor
Color del Fondo	12632256 (gris al 25%)
Barras de Desplazamiento	Ambas
Selectores de Registro	No
Botones de Desplazamiento	No
Separadores de Registro	No
Ajuste Automático de Tamaño	Si
Centrado Automático	Si
Estilo de bordes	Ajustable
Cuadros de Control	No
Botones Minimizar Maximizar	No
Botón Cerrar	No
Botón Qué es esto	No

Barra de Navegación.

Para la generalidad de los formularios navegan a lo largo de un conjunto de registros se deberá de contar con una barra de navegación estándar, como la mostrada en la figura:



Los nombres de los botones mostrados deberán ser:

- btnPrimero
- btnAnterior
- btnSiguiente
- btnUltimo
- btnDeshacer
- btnEditar
- btnAgregar
- btnGrabar
- btnEliminar
- btnConsultar
- btnImprimir
- btnBuscar
- btnCerrar
- btnSalir

Funcionalidad permitida en la Navegación.

La funcionalidad permitida al usuario en cada formulario dependerá de los permisos asignados en el módulo de seguridad del sistema, y en base a eso se habilitarán los botones que ejecutan cada una de las funciones, de tal forma que:

	Función	Botones Habilitados
Altas		btnAgregar btnGrabar btnSalir
Bajas		btnBuscar btnPrimero btnAnterior btnSiguiente btnUltimo btnEliminar btnSalir
Modificación		btnBuscar btnPrimero btnAnterior btnSiguiente btnUltimo btnEditar btnDeshacer btnGrabar btnSalir
Consultas		btnBuscar btnPrimero btnAnterior btnSiguiente btnUltimo btnSalir
Consulta		btnConsultar btnSalir
Impresión		btnImprimir btnSalir

Se deberá inhabilitar la introducción o modificación de los datos mostrados en los diferentes controles mientras que no se haya seleccionado la opción de Agregar un registro ó la de Modificar el Registro Actual.

Elementos de un Reporte:

Los reportes constarán de por lo menos tres secciones:

Encabezado de Página:

Esta sección medirá tres centímetros y medio de alto, en la cual deberá contener un subreporte de dos centímetros de alto que muestre los logotipos correspondientes, y un cuadro de texto a todo lo largo del reporte en el que se indicará el nombre del mismo utilizando letra Lucida Sans de 20 puntos en negritas.

Se dibujará una línea de separación de 2 puntos en color gris al 25%.

Se indicarán los títulos de la columnas que se mostrarán utilizando letra Verdana de 10 puntos en negritas y cursiva.

Se dibujará una línea de separación de 2 puntos en color negro y se dejará un espacio de medio centímetro por debajo de está línea.

Sección Detalle:

Esta sección será de un alto variable, dependiendo de la cantidad de controles que deba de mostrar y se utilizará letra Arial de 8 puntos normal.

Sección Pie de Página:

Esta sección medirá un centímetro de alto y en ella se mostrará la fecha del día, el nombre del área desarrolladora de la aplicación y el número de página impresa del número total de páginas del reporte, lo anterior ajustado al lado izquierdo, al centro y al lado derecho respectivamente y letra Times New Roman de 9 puntos en negritas.

Alineación:

La alineación de las Etiquetas y para los Cuadros de texto será a la Izquierda, mientras que para los campos numéricos es a la derecha.

AMBIENTE.

Para llevar a cabo los desarrollos de sistemas se utilizará:

Gestor de la base de datos :	Microsoft Access 2000 Microsoft SQL Server ver.7.0 Microsoft SQL Server ver. 2000
Herramientas de desarrollo :	Microsoft Visual Basic for Applications ver 6.3 Microsoft Visual Basic ver. 6.0 ODBC versión 3.52

En aquellos casos en que se utilice Access como herramienta de desarrollo y como gestor de base de datos antes de liberar el sistema a producción se deberán de hacer los ajustes necesarios para separar las tablas en una base de datos que servirá de Back-end, y

vinculándolas con la base de datos que servirá de Front-end, misma que contendrá todas las consultas, formularios, reportes, módulos y macros utilizadas por el sistema.

5.3 DISEÑO DE COMPONENTES Y NODOS

Componentes de implementación

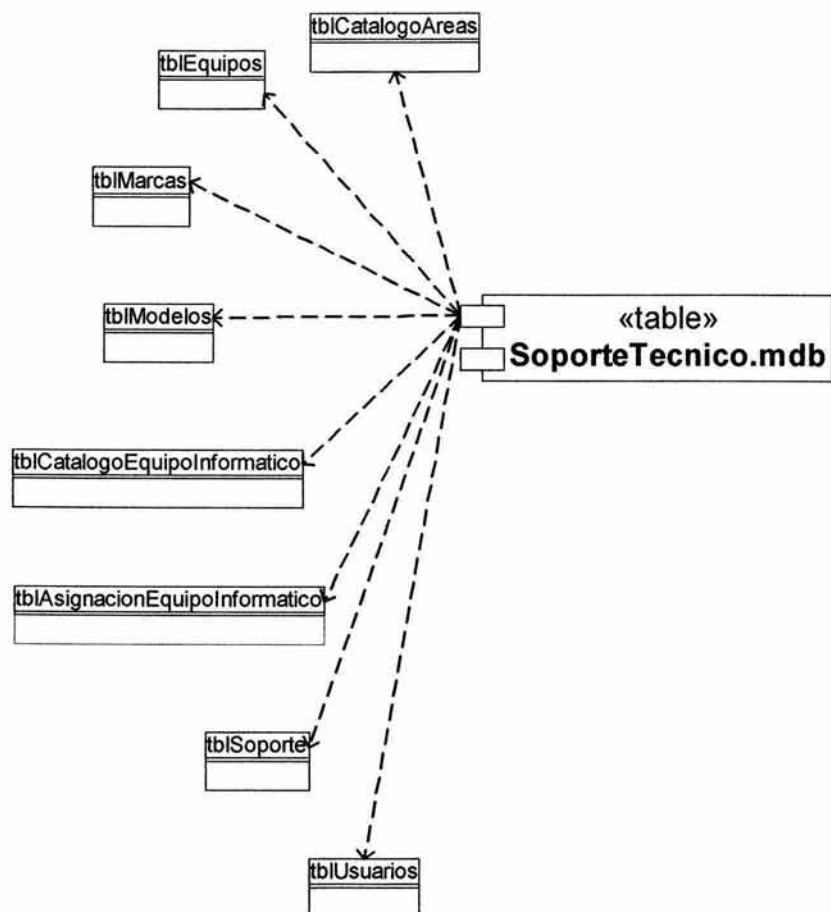


Figura 5.3

En la figura 5.3 se muestran el archivo de base de datos con extensión mdb. El archivo de base de datos está formado por el esquema lógico que forman las clases estáticas tblCatalogoAreas, tblEquipos, tblMarcas, tblModelos, tblCatalogoEquipoInformatico, tblAsignacionEquipoInformatico, tblSoporte y tblUsuarios.

Modelo de implementación

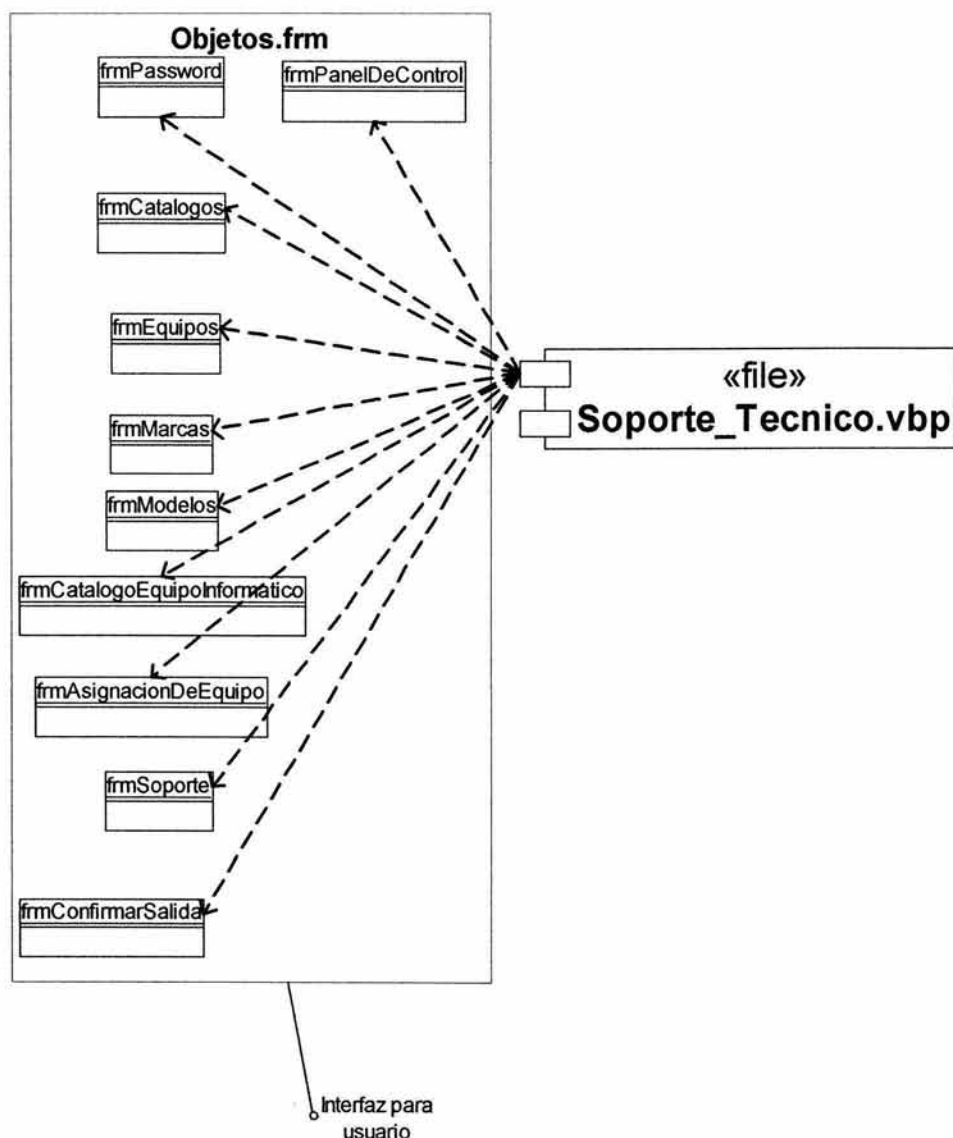


Figura 5.3.1

En la figura 5.3.1 se muestran el archivo de proyecto Soporte_Tecnico.vbp que está conformado por las clases dinámicas frmPanelDeControl, frmPassword, frmCatalogos, frmEquipos, frmMarcas, frmModelos, frmCatalogoEquipoInformatico, frmAsignacionDeEquipo, frmSoporte y frmConfirmarSalida a su vez este conjunto de clases forma el entorno de interfaz gráfica para el actor usuario.

Modelo de implementación

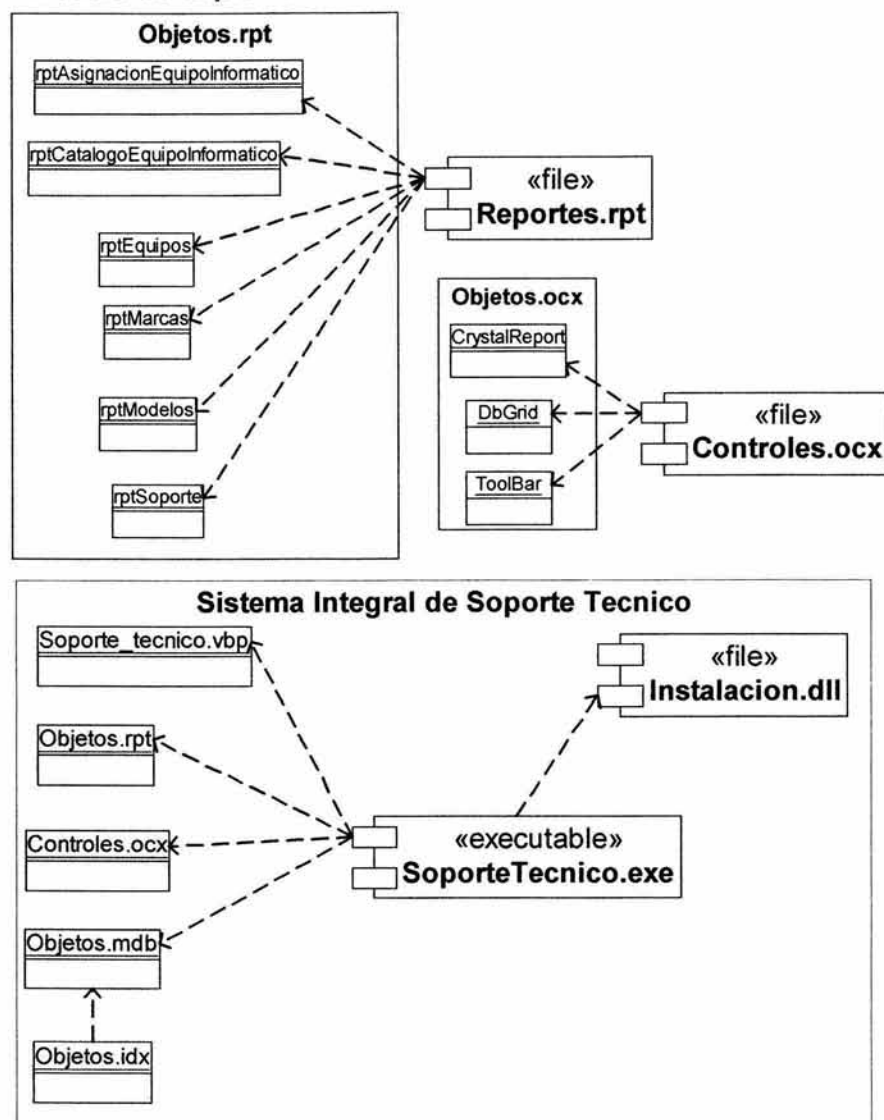


Figura 5.3.2

En la figura 5.3.2 se presenta el modelado del archivo `Reportes.rpt` que está conformado por los objetos con extensión `rpt`, `rptAsignacionEquipoinformatico`, `rptCatalogoEquipoinformatico`, `rptEquipos`, `rptMarcas`, `rptModelos` y `rptSoporte`. Los objetos con extensión `ocx` representan los archivos necesarios para poder utilizar controles especiales en la aplicación (ActiveX).

Arquitectura del sistema

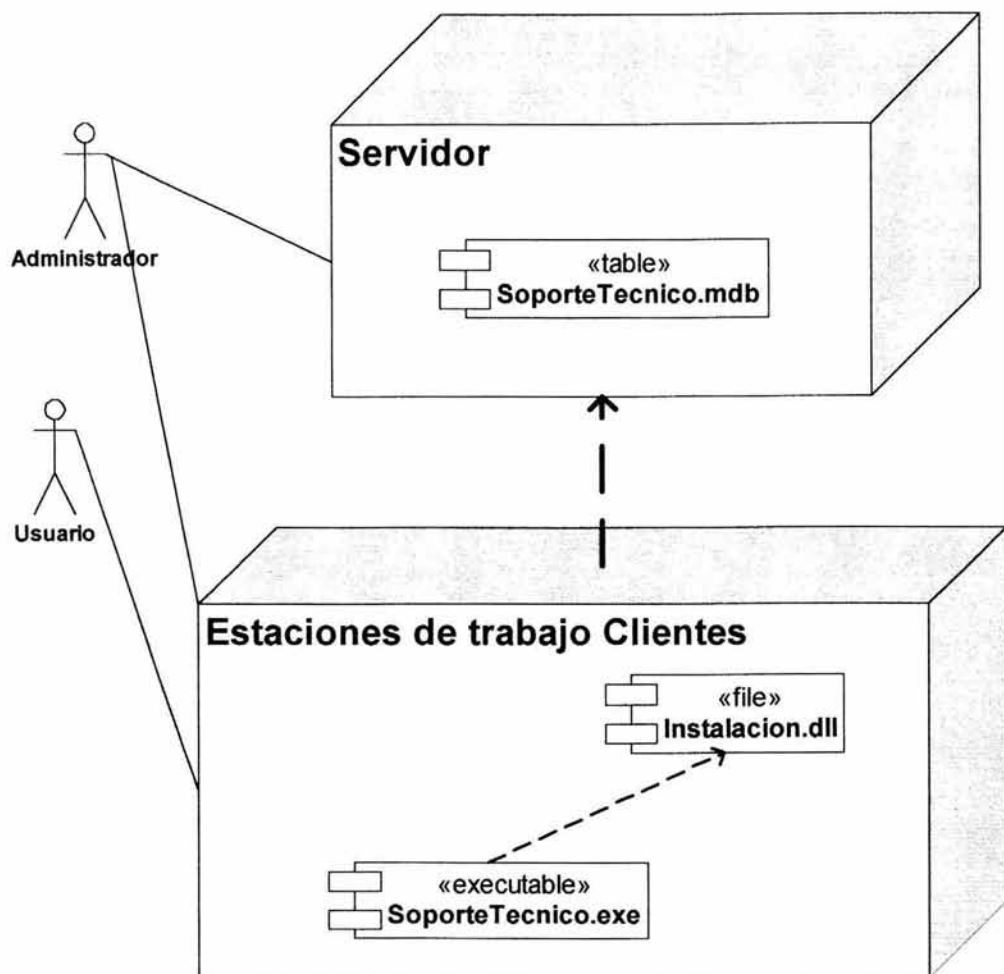


Figura 5.3.3

En la figura 5.3.3 se modela la arquitectura del sistema, en un entorno multiusuario, el archivo de base de datos Soporte Tecnico.mdb reside en el servidor de aplicaciones en un directorio compartido. En las estaciones de trabajo (Clientes) se instala la aplicación. Los actores accesan según su perfil de permisos.

CONCLUSIONES

Después de efectuar el desarrollo del Sistema Integral de Control de Soporte Técnico, tuvimos la labor de verificar los resultados de cada una de las fases del Proceso Unificado de Desarrollo de Software para corroborar que el sistema cumpliera con las finalidades trazadas en el inicio de su construcción.

Se prosiguió a probar los componentes ejecutables del sistema, bajo aspectos de funcionamiento bien específicos.

Se realizaron varios casos de prueba que nos permitieron poner en diversas condiciones de trabajo al Sistema Integral de Control de Soporte Técnico, obteniendo los resultados requeridos por los usuarios finales del sistema, así como una interacción consistente del sistema con todos los requisitos especiales.

La mayoría de los casos de prueba, se basaron en las instancias de los casos de uso, y en la interacción de estos con los requisitos no funcionales del sistema.

Se verificó que cada una de las iteraciones sucesivas en el Proceso Unificado cumpliera con los objetivos del sistema.

Antes de liberar la versión beta del Sistema Integral de Control de Soporte Técnico, se encontraron algunos errores en el código fuente, que fueron fáciles de localizar y corregir gracias a la documentación elaborada durante el Proceso Unificado. El sistema se mostró consistente en los aspectos fundamentales del Proceso Unificado (instancias de los casos de uso y en la arquitectura).

La Jefatura de Desarrollo y Mantenimiento de Sistemas liberó la versión beta del Sistema Integral de Control de Soporte Técnico, en el mes de Diciembre del 2002.

A pesar de que la fase de transición no terminará nunca y, el Sistema Integral de Control de Soporte Técnico sigue evolucionando gracias a la arquitectura implantada en la construcción del sistema, la versión beta del sistema satisface las necesidades básicas de la Unidad Departamental de Soporte Técnico.

Dentro de las expectativas de trabajo, la Jefatura de Unidad Departamental de Desarrollo y Mantenimiento de Sistemas tiene la convicción de que el Sistema Integral de Control de Soporte Técnico, no podrá resistir mucho tiempo antes de que se libere otra versión más reciente del sistema, esto debido a los cambios significativos que existen en las políticas internas del Gobierno de la Ciudad de México, así como en la industria del software y hardware, estos dos últimos quizá los más relevantes puesto que tienen que ver con aspectos de gran trascendencia como lo son:

- Arquitectura de desarrollo
- Gestores de base de datos
- Lenguajes de programación
- Sistemas operativos

Por consiguiente el ciclo de inicio, elaboración, construcción y transición se repite indefinidamente, para el Proceso Unificado de Desarrollo del Sistema Integral de Control de Soporte Técnico.

APÉNDICE

APÉNDICE A

EL PERFIL PROFESIONAL DE BOOCH, RUMBAUGH Y JACOBSON

GRADY BOOCH

Booch es reconocido internacionalmente por sus innovadores trabajos en modelado de arquitectura de software e ingeniería de procesos. Es un renombrado visionario y devoto del trabajo relacionado con desarrollo de técnicas de software en el mundo informático.

Grady Booch es el Jefe del Departamento de Ciencias de la corporación Rational Software desde el año de 1980 y continúa contribuyendo con sus desarrollos en la arquitectura de soluciones de software para el Grupo de IBM.

Booch es uno de los creadores de Rational Rose y otros productos relevantes, Grady ha aportado a la comunidad mundial con grandes novedades en arquitectura de software. Es autor de 13 obras literarias informativas ha cerca del modelado en arquitectura de software y proyectos de ingeniería de procesos.

Grady es miembro de la Association for Computing Machinery (Asociación para la Maquinaria de Computación ACM) del Institute of Electrical and Electronics Engineers (Instituto de Ingenieros en Electrónica e Electricidad IEEE) de la American Association for the Advancement of Science (Asociación Americana para el Desarrollo de la Ciencia AASS) y Computer Professionals for Social Responsibility (Profesionales Responsables de la Computación Socialmente)

Grady Booch recibió su título de Ingeniero por la Academia de la Fuerza Aérea de los Estados Unidos en el año de 1977 y su Maestría por la Universidad de California de Santa Bárbara en el año de 1979.

JAMES RUMBAUGH

El Dr. James Rumbaugh es uno de los encargados del manejo metodologista del desarrollo de software a nivel mundial. Con sus colegas Grady Booch e Ivar Jacobson, James desarrolló el Lenguaje Unificado de Modelado (UML), lenguaje que se ha convertido en el estándar de modelado de software dentro de la industria, adaptado específicamente por la Object Management Group (Grupo de Administración de Objetos OMG) en el año de 1997. James Rumbaugh maneja el desarrollo futuro de UML como representante de Rational. Ha contribuido dirigiendo otros trabajos dentro de Rational Software Corporation, como el Proceso Unificado de Desarrollo de Software y metodologías de desarrollo en tiempo real.

James ha trabajado en el desarrollo de metodologías y herramientas que sirvan de base para el desarrollo de software por más de 30 años dentro de la industria. Jim fué jefe de desarrollo de Object Modeling Technique (Técnicas de Modelado de Objetos OMT) y condujo el análisis orientado a objetos, que hoy en día son metodologías predecesoras de UML.

Antes de la unión de Rational Software Corporation en 1994, James trabajó por mas de 25 años en General Electric Research y Development Center in Shenectady, Nueva York. James trabajó en el desarrollo de DSM object-oriented programming language. También

trabajó en diversas aplicaciones con el sistema VLSI CAD, algoritmos para GE's, termografía original de scanner.

James realizó dentro de sus estudios de doctorado innovaciones en la arquitectura del flujo de datos con el Profesor Jack Dennis en el Instituto Tecnológico de Massachussets (MIT). Ha trabajado en numerosas áreas de computación, incluyendo la semántica de la computación, herramientas para aumentar la productividad de la programación y algoritmos complejos para aplicaciones de estructuras de datos.

James obtuvo su licenciatura en Física por MIT (Instituto Tecnológico de Massachussets), su Maestría en Astronomía por Caltech y su Doctorado en Ciencias de la computación por MIT (Instituto Tecnológico de Massachussets).

IVAR JACOBSON

El Dr. Ivar Jacobson, Vicepresidente de Procesos Estratégicos dentro de Rational, fundó en Suiza la compañía Objectory AB.

Uno de los tres creadores intelectuales de UML el a contribuido en desarrollos en una gran diversidad de áreas relacionadas con software, en donde destacan:

- **Arquitectura de Componentes.** Ivar introdujo el uso de componentes en el desarrollo de software, desarrollo realizado dentro de Ericsson en el año de 1967, inventó los diagramas de secuencia y colaboración para modelar la interacción de los componentes, todos estos antecesores de lo que hoy es UML.
- **Lenguajes de Modelado.** SDL trabajo desarrollado en Ericsson. Lenguaje adaptado por la Comunidad de estandarización para las Telecomunicaciones en 1976.
- **Casos de uso.** Inventor del los casos de uso y especificaciones de requerimientos funcionales a través de estos.
- **Modelado de negocios.** Desarrolló las técnicas para el modelado de negocios.
- **Procesos de software.** Considerado como el padre del Proceso orientado a objetos y componentes de desarrollo de software.

Jacobson es autor de 5 libros de ingeniería y software orientado a objetos así como de otros temas relacionados con el proceso de desarrollo de software. Jacobson recibió su doctorado por el Instituto Tecnológico Royal en Estocolmo.

APÉNDICE B

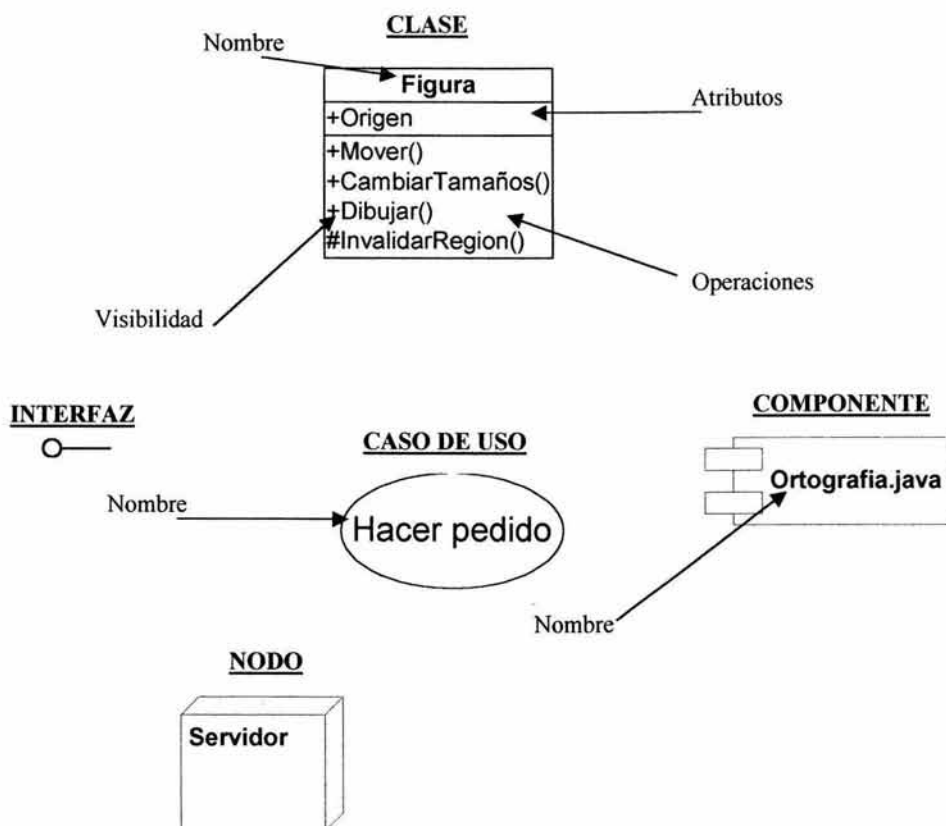
REFERENCIAS D E UML

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Como lenguaje, UML tiene una sintaxis y una semántica bien definidas. La parte más visible de la sintaxis de UML es su notación gráfica.

En el siguiente apéndice se resumen los elementos de la notación de UML.

Elementos estructurales

Los elementos estructurales son los nombres de los modelos de UML. Estos incluyen clases, interfaces, casos de uso, componentes y nodos.



Clase: Descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

Atributo: Propiedad con nombre de un clasificador que describe un rango de valores que pueden contener las instancias de la propiedad.

Operación: Implementación de un servicio que puede ser requerido a cualquier objeto de la clase para que muestre un comportamiento.

Interfaz: Colección de operaciones que se utiliza para especificar un servicio de una clase o componente.

Caso de uso: Descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable, de valor para un actor.

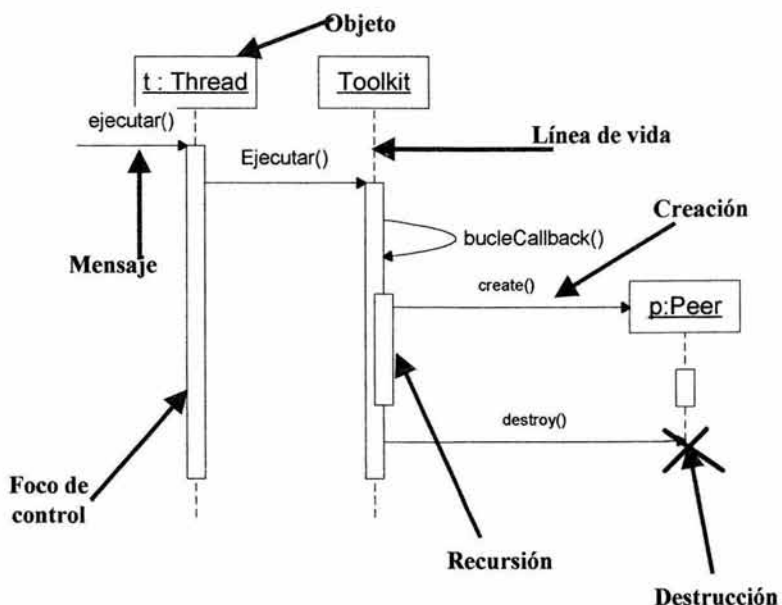
Componente: Parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto.

Nodo: Elemento físico que existe en tiempo de ejecución y que representa un recurso computacional, que normalmente tiene algo de memoria y, a menudo, capacidad de procesamiento.

Elementos de comportamiento

Los elementos de comportamiento son las partes dinámicas de los modelos de UML. Estos incluyen diagramas de interacción y maquinas de estados.

INTERACCIÓN



Objeto: Manifestación concreta de una abstracción, entidad con unos límites bien definidos e identidad que encapsula estado y comportamiento, instancia de una clase.

Mensaje: Especificación de una comunicación entre objetos que transmite información con la expectativa que se desencadenará actividad, la recepción de la instancia de un mensaje se considera normalmente una instancia de un evento.

Foco de control: Símbolo sobre un diagrama de secuencia que muestra el período de tiempo durante el cual un objeto ejecuta una serie de acciones directamente o a través de una operación subordinada.

Línea de vida: Línea de diagrama de secuencia que representa la existencia de un objeto a lo largo de un periodo de tiempo.

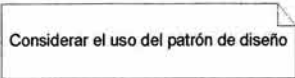
Creación: Proceso que tiene como finalidad crear un objeto

Destrucción: Proceso que tiene como finalidad destruir un objeto en un tiempo determinado.

Elementos de agrupación

Los elementos de agrupación son las partes organizativas de los modelos de UML. Estos incluyen las notas

NOTA



Considerar el uso del patrón de diseño

Nota: Símbolo gráfico que representa restricciones o comentarios asociados a un elemento o a una colección de elementos.

RELACIONES

Dependencia

Una dependencia es una relación semántica entre dos elementos, en la cual un cambio a un elemento (el elemento independiente) puede afectar a la semántica del otro (el elemento dependiente)



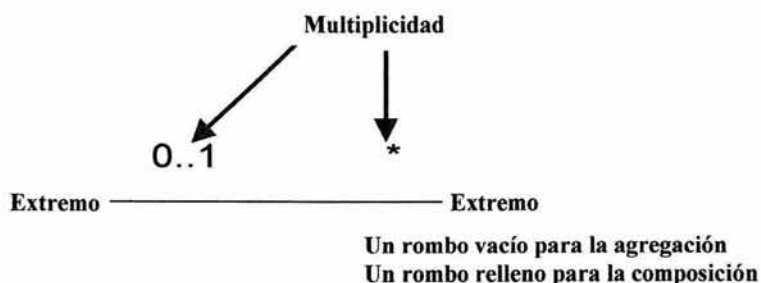
Generalización

La generalización es una relación de especialización/generalización en la cual los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del elemento general (padre).



Asociación

Una asociación es una relación estructural que describe un conjunto de enlaces, un enlace es una conexión entre objetos.



Multiplicidad: Especificación del rango de cardinalidad permisible que puede asumir un conjunto.

Agregación: Forma especial de asociación que especifica una relación todo-parte entre el agregado (el todo) y un componente la parte.

Composición: forma de agregación con fuerte pertenencia y un tiempo de vida coincidente entre las partes y el todo, las partes con una multiplicidad no fija pueden ser creadas después del propio compuesto, pero una vez creadas viven y mueren con el, tales partes también pueden ser eliminadas explícitamente antes de la eliminación del compuesto.

REFERENCIAS BIBLIOGRÁFICAS

Booch, Grady, Rumbaugh, James and Jacobson Ivar. El Proceso Unificado de Desarrollo de Software, Editorial Addison Wesley, Madrid España 2000, Prefacio XVI-XXIII, páginas 5-8, 56,82,149, 167,168, 205,206, 255 (a)

Booch, Grady, Rumbaugh, James and Jacobson Ivar. El Lenguaje Unificado de Modelado, Editorial Addison Wesley, Madrid España 1999, páginas 81, 125,385-389

Schneider, Geri and Winters, Jasón P. Applying Use Cases, Editorial Addison Wesley

Booch, Grady, Rumbaugh, James and Jacobson Ivar. El Lenguaje Unificado de Modelado, Manual de Referencia, Editorial Addison Wesley, Madrid España 2000 (b)

Ericsson, Erick Hans. Bussines Modeling with UML, Editorial, Jhon Wiley &Sons, Inc.

[1] <http://www.vico.org/Navegacio/vicoOfertaServicios.htm>

[2] <http://www.uml.org/>

[3] <http://www.softdocwiz.com/UML.htm>