



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

## FACULTAD DE INGENIERÍA

*Sistema de Inventario de Cómputo para la  
Secretaría de Servicios Académicos de la  
Facultad de Ingeniería*

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTAN:

MERCADO MARTÍNEZ ADRIAN ULISES

SÁNCHEZ FAUSTINO DULCE MARÍA

ASESOR

ING. FILIBERTO MANZO GONZÁLEZ



MÉXICO D.F

2009



## Sistema de inventario



# *Agradecimientos*

---

## **AGRADECIMIENTOS**

Agradezco a Dios por la vida, por cada momento bello de ella, por su amor y paciencia. Porque con él las cosas imposibles se tornan posibles y lo convencional se vuelve maravilloso. Gracias Dios por amarme así.

A mi mamá y mis hermanos les agradezco por todo su apoyo, por su comprensión, su paciencia, por compartir los momentos alegres, pero cuando más por los momentos difíciles que su apoyo y su amor siempre están ahí incondicionalmente.

A mis amigos que con su alegría llenaron cada momento de mi vida.

A la Universidad Nacional Autónoma de México, conjuntamente con la Facultad de Ingeniería que me formaron como Ingeniero, en especial a todos mis profesores que los recuerdo con cariño.

A mi asesor de tesis el Ing. Filiberto Manzo que con paciencia me apoyo en este trabajo de tesis.

A cada una de las personas que me han permitido ser parte de su vida y que han querido que yo forme parte de la suya gracias por alegrar mi existencia en este mundo.

*Adrian Ulises Mercado Martínez*

## **AGRADECIMIENTOS**

Le agradezco principalmente a Dios por darme la gracia de la vida, ya que sin Él nada existe.

A mis padres, Vicente y Susana, por su apoyo incondicional y ejemplo de dedicación a la largo de toda mi carrera, por sus preocupaciones y sus desvelos, y principalmente por el cariño que siempre me han brindado. Sin ellos nada hubiera sido posible.

A mis más grandes amores, Mario y Dany por el tiempo robado, gracias por su paciencia y amor.

A mis hermanos por hacerme ver mis errores y mis aciertos.

A mi Universidad y principalmente a mi Facultad de Ingeniería por darme la oportunidad de pertenecer a ella.

A mis profesores, por su sabiduría y enseñanza, así como sus consejos.

Al Ingeniero Filiberto, mi asesor y amigo, por tenerme la paciencia necesaria y creer en mí.

A mis amigos Adrian, Emmanuel y Rosangel, por haberme brindado su amistad y ayuda en los momentos difíciles.

Y gracias a todas aquellas personas que me han apoyado a lo largo de toda mi vida y principalmente a aquellas que han creído en mí.

¡GRACIAS!

*Dulce M. Sánchez Faustino*

## Índice

<b>I. Portada</b>	i
<b>II. Agradecimientos</b>	ii
<b>III. Índice</b>	iv
<b>1. Capítulo 1. Planteamiento del Problema</b>	
1.1 Introducción	1
1.2 Planteamiento del Problema	1
1.3 Objetivos del Sistema de Inventario de Cómputo	2
1.4 Justificación	3
1.5 Antecedentes	4
1.5.1 Sistemas de información	4
1.5.2 Inventarios	5
1.6 Secretaría de Servicios Académicos	6
1.6.1 Coordinación de Administración Escolar	7
1.6.2 Departamento de Apoyo a la Comunidad	8
1.6.3 Departamento de Publicaciones	9
1.6.4 Unidad de Servicios de Cómputo Administrativo (USECAD)	9
1.6.5 Organigrama	11
<b>2 Capítulo 2. Marco Teórico</b>	
2.1 Introducción	12
2.2 Modelos de Procesos	13
2.2.1 Modelo Lineal Secuencial (MLS)	13
2.2.2 Modelo de Construcción de Prototipos	15
2.2.3 Modelo DRA (Desarrollo Rápido de Aplicaciones)	16
2.2.4 Modelos Evolutivos del Proceso del Software	18
2.2.4.1 Modelo en Espiral	18
2.2.4.2 Modelo de Desarrollo Concurrente	19
2.2.5 Proceso Unificado de Desarrollo (RUP)	20
2.2.5.1 Dirigido por casos de uso	21
2.2.5.2 Centrado en la arquitectura	22
2.2.5.3 Iterativo e incremental	22

	2.2.5.4	La vida del Proceso Unificado	22
	2.2.5.5	Fases Dentro del un Ciclo	23
2.3		Tecnología a Utilizar	24
	2.3.1	SUN SPARC	24
	2.3.1.1	Principales Características	24
	2.3.1.2	Categorías de Instrucciones	25
	2.3.1.3	Ventanas de Registros	25
	2.3.1.4	Interrupciones	26
	2.3.1.5	Protección de Memoria	26
	2.3.2	Solaris (SUN)	26
	2.3.3	Sybase ASE para Solaris	27
	2.3.3.1	Principales Características	27
	2.3.4	Java	29
	2.3.4.1	Principales Características	29
	2.3.4.2	Java Server Pages (JSP)	31
	2.3.5	TOMCAT	33

### **3 Capítulo 3. Definición de Requerimientos**

3.1		Requerimientos de Usuario	35
	3.1.1	Definición de Requerimientos	35
	3.1.2	Clasificación	35
3.2		Modelo del Negocio	35
3.3		Modelo del Negocio para el Sistema de Inventario	36
	3.3.1	Modelo de casos de uso del Negocio	36
	3.3.2	Modelo de Objetos del Negocio	39
3.4		Modelo de Casos de Uso	41
	3.4.1	Actor	42
	3.4.2	Caso de Uso	42
	3.4.3	Flujo de sucesos	42
	3.4.4	Construcción del Modelo de Casos de Uso	43
	3.4.4.1	Encontrar Actores	43
	3.4.4.2	Encontrar Casos de Uso	45
		3.4.4.2.1 Descripción General	45
	3.4.4.3	Reestructuración del modelo de Casos de Uso	50
	3.4.4.4	Detallar cada Caso de Uso	52

## 4 Capítulo 4. *Análisis y Diseño*

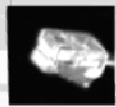
4.1	Modelo de análisis.....	69
4.2	Análisis de la Arquitectura.....	69
4.2.1	Paquete de análisis.....	69
4.2.2	Identificar paquetes del análisis.....	70
4.2.3	Identificar Paquetes de Análisis para el Sistema de Inventario.....	70
4.3	Clases de Análisis.....	73
4.4	Analizar Casos de Uso.....	74
4.5	Analizar Casos de Uso del Sistema de Inventario.....	74
4.6	Analizar Clases.....	89
4.6.1	Identificar Responsabilidades.....	89
4.6.2	Identificar Atributos.....	90
4.6.3	Identificar Asociaciones, Agregaciones y Generalizaciones.....	90
4.7	Análisis de las Clases del Sistema de Inventario.....	90
4.7.1	Identificar Atributos.....	90
4.7.2	Identificar Responsabilidades.....	95
4.7.3	Identificar Asociaciones y Agregaciones.....	100
4.8	Modelo de Diseño.....	104
4.9	Diseño de la Arquitectura.....	104
4.9.1	Identificar Nodos y sus Configuraciones.....	104
4.9.2	Identificar Subsistemas y sus interfaces.....	105
4.9.3	Identificar Clases de Diseño Relevantes	
4.9.4	para la Arquitectura.....	105
4.9.5	Identificación de Mecanismos Genéricos.....	105
4.10	Diseño de la Arquitectura del Sistema de Inventario.....	105
4.10.1	Identificar Nodos y Configuración de Red.....	110
4.10.2	Identificación de Subsistemas y sus interfaces.....	110
4.10.3	Identificación de Mecanismos Genéricos de Diseño.....	112
4.11	Diseño de Caos de Uso.....	114
4.11.1	Identificación de Clases de Diseño.....	114

4.12	Diseño de los Casos de Uso del Sistema de Inventario.....	115
4.13	Diseño de Clases.....	130
4.13.1	Identificar Operaciones.....	130
4.13.2	Identificar Atributos.....	131
4.13.3	Identificar Asociaciones,	
4.13.4	Agregaciones y Generalizaciones.....	131
4.13.5	Describir Métodos.....	131
4.14	Diseño de Clases del Sistema de Inventario.....	132
<b>5</b>	<b>Capítulo 5. Implementación y Prueba</b>	
5.1	Modelo de Implementación.....	140
5.2	Implementación de la Arquitectura.....	140
5.2.1	Componente.....	140
5.2.2	Identificar los Componentes de la Arquitectura.....	141
5.2.3	Identificar Componentes Ejecutables y Asignación de éstos a Nodos.....	141
5.3	Implementación de la Arquitectura del Sistema de Inventario.....	141
5.3.1	Identificación de los Componentes Significativos para la Arquitectura.....	141
5.4	Implementar Clases.....	142
5.5	Implementar Clases del Sistema de Inventario.....	142
5.6	Integrar el Sistema.....	143
5.7	Integrar el Sistema de Inventario.....	144
5.8	Modelo de Prueba.....	146
5.9	Planificar y Diseñar Pruebas.....	146
5.9.1	Caso de Prueba.....	146
5.10	Casos de Prueba del sistema de Inventario.....	147
<b>6</b>	<b>Conclusiones</b> .....	<b>176</b>
<b>7</b>	<b>Bibliografía</b> .....	<b>178</b>
<b>8</b>	<b>Anexos</b> .....	<b>I</b>





Sistema de  
inventario



# *Capítulo 1*

## *Planteamiento del Problema*

---

## **1.1 Introducción**

El cambio acelerado en las tecnologías de información ha provocado mayores exigencias de los usuarios hacia los sistemas informáticos, requiriendo mayor eficiencia y flexibilidad, lo que obliga a tener un mejor manejo de la información. Por tal motivo las computadoras han tenido cada vez mayor impacto en los procesos administrativos y productivos de las organizaciones.

La principal razón por la que la computación y los sistemas informáticos han tenido tan exitosa incursión en las actividades administrativas, ha sido que resulta de gran utilidad para el manejo de la información, en especial de grandes volúmenes de ésta; cobrando mayor importancia cuanto mayor es el valor para las organizaciones.

Es por ello que se desea modernizar el sistema de inventario para el equipo de cómputo con el que cuenta la Secretaría de Servicios Académicos (SSA) de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, ya que presenta una problemática al no contar con un sistema adecuado para el manejo de la información relacionada a este proyecto de inventario, lo que provoca inconsistencia en los datos, redundancia y pérdida de la misma.

## **1.2 Planteamiento del Problema**

Dado que el sistema de inventario de la SSA aún es un sistema de ficheros, los datos se recogen varias veces y se encuentran repetidos en los distintos archivos. Esta redundancia, además de no aprovechar los recursos de manera adecuada, origina a menudo que los resultados para una misma consulta sean distintos.

Además que el sistema actual es un sistema inseguro, al no contar con restricciones para diferentes tipos de usuarios, lo que origina que cualquier usuario

que tenga acceso a los ficheros realice modificaciones sobre los datos perdiéndose así la confidencialidad e integridad de los mismos, aunado a esto, no se asegura la disponibilidad, dado que si un usuario esta trabajando con la información ningún otro usuario puede disponer de ella.

Debido al gran volumen de información que se maneja se dificulta la manipulación de los datos, tales como actualizaciones, bajas y altas, así como la generación de reportes y el manejo de bitácoras.

### **1.3 Objetivos del Sistema de Inventario de Cómputo**

- Mantener la seguridad de la información.
- Evitar la redundancia de datos.
- Mantener la consistencia de datos.
- Mantener la concurrencia de la información.
- Adecuado control de acceso.
- Mantener la integridad de la información.
- Mantener la confidencialidad.
- Agilizar el proceso de búsqueda de información de la base de datos.
- Facilitar la modificación de los datos.
- Presentar a los usuarios una interfaz gráfica amigable que facilite el manejo de la información.
- Presentar reportes de los movimientos en este sistema de inventario, indicando la fecha de su realización, así como el nombre del usuario y el movimiento que se realizó.
- Preparar el reporte solicitado por DGSCA a la Secretaría de Servicios Académicos de la Facultad de Ingeniería sobre los censos del equipo de cómputo que se deben entregar anualmente.

#### **1.4 Justificación**

Un sistema de información computarizado resolverá los problemas referentes a la seguridad, tales como:

- Integridad de la información: se mantiene la integridad controlando el acceso de los usuarios al sistema, mediante cuentas de usuario con distintos privilegios.
- Confidencialidad de la información: con el manejo de tipos de usuario se garantiza que solo los usuarios autorizados tendrán acceso a la información.
- Disponibilidad de la información dado que el sistema se encontrará en línea se podrá tener la información disponible para varios usuarios en el momento que sea requerida.

El sistema propuesto tendrá las siguientes ventajas sobre el sistema de ficheros actual:

- Acceso a la base de datos con encriptación al acceder a la aplicación.
- Alojamiento de la base de datos en Sybase, con un constante monitoreo y administración.
- Creación de un modelo Entidad-Relación, lo que permitirá una mejor visualización de las entidades y relaciones en el modelo conceptual.
- El sistema será portable, es decir, se podrá ejecutar en cualquier sistema operativo, debido a que Java es el lenguaje de programación a utilizar y es multiplataforma al igual que el servidor Tomcat.
- Coherencia en los resultados, debido a que la información se recoge y almacena una sola vez, en el procesamiento de los mismos los resultados de todos ellos son íntegros y perfectamente comparables entre sí.

- Mayor eficiencia en la recolección y validación de los datos en el sistema; los datos se capturan y validan una sola vez, aumentando así el rendimiento de todo el proceso previo al almacenamiento.
- Reducción del espacio de almacenamiento, la disminución de la redundancia provoca que el volumen de los datos sea menor, utilizando menor espacio de almacenamiento en el disco duro.
- Acceso más rápido y sencillo de los usuarios.

## **1.5 Antecedentes**

### **1.5.1 Sistemas de información**

Un sistema de información es un conjunto organizado de cuatro elementos que interactúan entre sí con el fin de facilitar el uso y almacenamiento de datos en una organización. Dichos elementos son:

- Personas.
- Datos.
- Actividades y/o técnicas de trabajo.
- Recursos informáticos y de comunicación.

Se diseña a fin de satisfacer las necesidades de información de una organización y está inmerso en ella. El sistema de información ha de tomar los datos del entorno y sus resultados han de ser la información que dicha organización necesita para su gestión y toma de decisiones. <sup>1</sup>

Características de un sistema de información:

- Tecnológicas, que afectan al rendimiento y seguridad del sistema, desde el punto de vista del equipo.

---

<sup>1</sup> Castaño, Fundamentos y modelos de bases de datos 1ª. Ed. Madrid, España: Alfaomega, 1997

- Funcionales y semánticas, que se refieren a si el sistema hace lo que debe de una forma correcta y si es capaz de adaptarse a requisitos cambiantes.
- Económicas, que ponen énfasis en el coste del sistema y en la eficiencia con que responde a los objetivos.
- Sociales, que son las que tienen un impacto sobre el entorno social, en que se devuelve el sistema.

### **1.5.2 Inventarios**

Un inventario según la definición que da la Real Academia Española es:

“Asiento de los bienes y demás cosas pertenecientes a una persona o comunidad, hecho con orden y precisión”.

El llevar a cabo un buen inventario en las instituciones y en las empresas es necesario para conocer los recursos con los que se cuenta, de esta manera se obtiene el control de su ubicación, estado y así como su disponibilidad en el momento en que se requiere.

Algunas funciones con las que debe cumplir el inventario son:

- Satisfacer la demanda anticipada.
- Proteger contra falta de artículos.
- Tomar ventaja de los ciclos de órdenes.
- Permitir la realización de las operaciones.
- Eliminar irregularidades en los servicios.
- Comprar equipo por lotes.
- Permitir el manejo de materiales perecederos.

## **1.6 Secretaría de Servicios Académicos**

La Secretaría de Servicios Académicos busca crear un ambiente propicio para que el estudiante canalice todos sus esfuerzos a su tarea principal: su formación académica. Asimismo, ofrece servicios de apoyo académico-administrativo en materia de inscripciones y trámites escolares.

### **Objetivos:**

- Coordinar la ejecución de las actividades de administración escolar, de apoyo a maestros, alumnos y en general, de proporcionar los servicios que se requieran en la Facultad para el mejor desempeño de las actividades académicas.
- Facilitar la integración de los alumnos a las agrupaciones estudiantiles, actividades culturales y deportivas, buscando con ello su formación integral fortaleciendo su sentido de pertenencia a la Facultad y a la Universidad.

### **Funciones:**

- Proporcionar a profesores y alumnos los apoyos de administración escolar.
- Difundir actividades de interés para los estudiantes.
- Imprimir materiales didácticos y de difusión.
- Coordinar la utilización de los recintos del Palacio de Minería.
- Coordinar los procesos de admisión e inscripciones.
- Brindar los servicios de cómputo académico-administrativo y de bolsa de trabajo.
- Realizar estudios académicos y preparar los informes que le sean solicitados por la Dirección de la Facultad de Ingeniería.
- Coordinar y promover aquellas actividades deportivas, recreativas y de apoyo al aprendizaje del idioma inglés que propicien una formación integral de los estudiantes de la Facultad.

## *Capítulo 1. Planteamiento del Problema*

- Apoyar y promover la integración y consolidación de las agrupaciones de estudiantes formalmente establecidas.

La Secretaría se conforma de las siguientes áreas:

### **1.6.1 Coordinación de Administración Escolar**

#### **Objetivos:**

- Planear y coordinar la ejecución de las actividades académico-escolares de la Facultad, observando el cumplimiento de la normatividad aplicable.
- Brindar a los estudiantes atención y apoyo respecto a trámites escolares, servicio social, titulación y normatividad escolar.
- Efectuar el análisis y procesamiento de información en materia escolar, así como diseñar procedimientos que contribuyan al establecimiento y mejoramiento de los canales de comunicación con la comunidad estudiantil de la Facultad.

#### **Funciones:**

- Coordinar los procesos de inscripción, primer ingreso, exámenes extraordinarios, cambios de carrera, segunda carrera, carrera simultánea y calificación de actas.
- Coordinar el proceso de titulación de los alumnos de la Facultad de Ingeniería, así como administrar los sistemas de información correspondientes.
- Apoyar el registro de los programas de Servicio Social de la Facultad y coordinar la expedición de cartas de cumplimiento del Servicio Social.



- Atender las solicitudes de la *Dirección General de Incorporación y Revalidación de Estudios*, destacando las relativas al análisis de equivalencia entre los planes de estudio de la Facultad y los de otras Instituciones.
- Administrar y actualizar el *Portal de la Secretaría de Servicios Académicos* y el *Directorio de Agrupaciones Estudiantiles*.
- Brindar apoyo a los proyectos del Plan de Desarrollo relacionados con la administración escolar, la titulación y el servicio social.

### **1.6.2 Departamento de Apoyo a la Comunidad**

#### **Objetivo:**

Coadyuvar a la formación integral del alumno a través de la realización de torneos deportivos y recreativos, así como de actividades artísticas, coordinando la participación de los equipos deportivos representativos de la Facultad en los Torneos Interfacultades y Juegos Puma.

#### **Funciones:**

- Actividades Deportivas: Organiza distintos eventos deportivos.
- Actividades Culturales y Recreativas: Organiza eventos culturales y recreativos para un enriquecimiento general de la comunidad.
- Control de Carteles: Realiza la difusión de los eventos que se realizan en la Facultad de Ingeniería apoyándolos con carteles alusivos a estos eventos.

### **1.6.3 Departamento de Publicaciones**

#### **Objetivo:**

Realizar diversos trabajos de diseño, formación e impresión de material de apoyo didáctico elaborado por el personal docente de la dependencia, así como imprimir diversos tipos de trabajos para la difusión de actividades académicas y culturales presentadas en la Facultad.

#### **Funciones:**

- Área de Diseño: Diseña, captura y forma los trabajos solicitado de acuerdo a las especificaciones requeridas, con la finalidad de que estos tengan una excelente presentación.
- Área de Fotelito: Prepara los negativos y las láminas para realizar los trabajos de impresión. Aunque los fotolitos pueden obtenerse de cualquier tipo original, incluso los producidos por tipografía, en la actualidad se obtienen a través de la unidad fotográfica.
- Área de impresión y fotocopiado: Realiza los trabajos de impresión a color y blanco y negro, compaginación, encuadernación y dar el refine final a los trabajos.

### **1.6.4 Unidad de Servicios de Cómputo Administrativos (USECAD)**

#### **Objetivos:**

- Proporcionar los servicios de cómputo, en forma eficiente y oportuna, para mejorar la vida académica de la Facultad.
- Desarrollar los sistemas de cómputo que soporten las actividades académico-administrativas que generan los órganos de la Facultad.

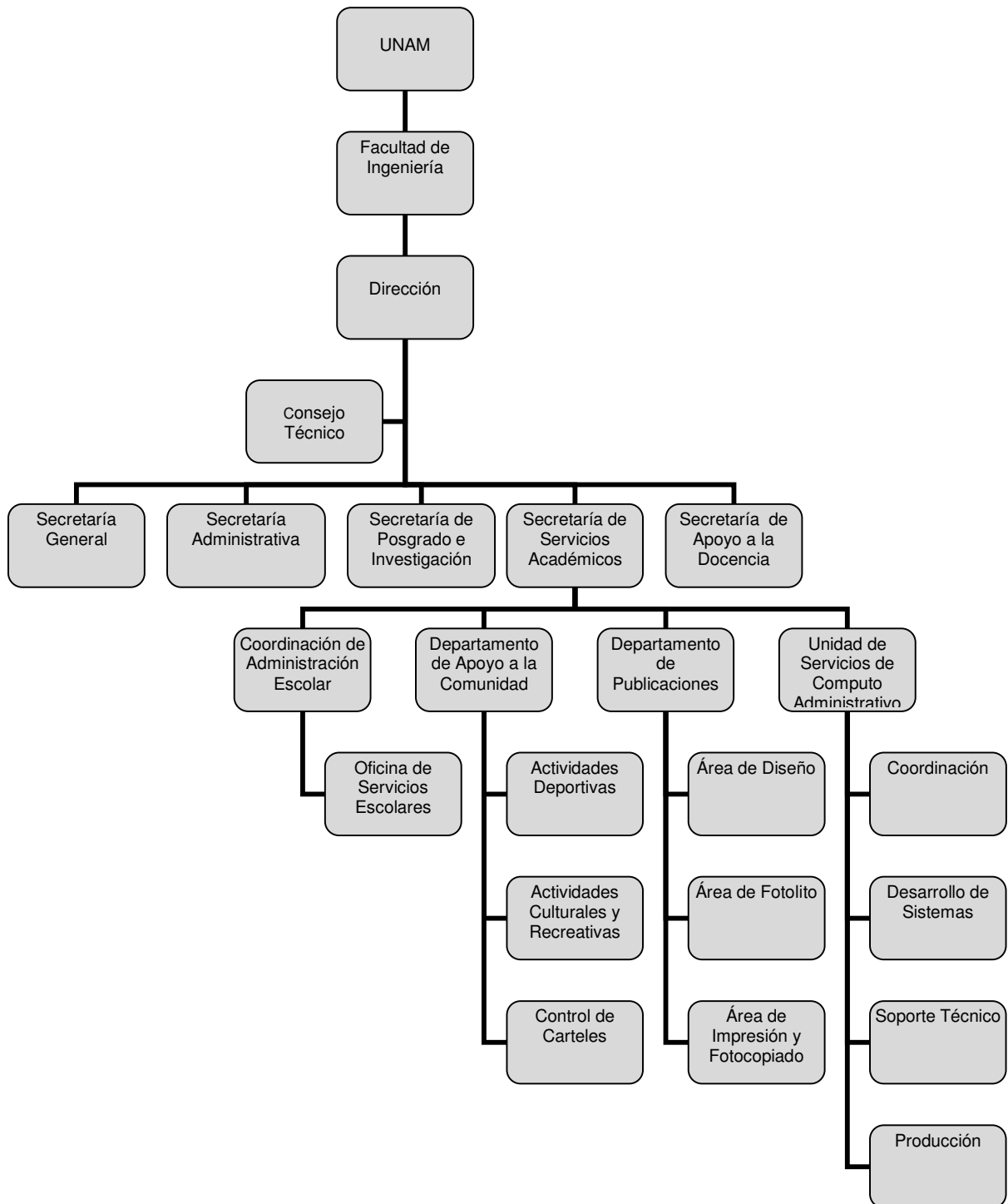
## *Capítulo 1. Planteamiento del Problema*

- Mejorar la calidad de servicio a los usuarios.
- Garantizar la operación normal de los equipos de cómputo de la Secretaría de Servicios Académicos.
- Contar con sistemas de información flexibles en su operación y de fácil mantenimiento.
- Contar con personal capacitado en áreas de cómputo.
- Diseñar e incorporar nuevos productos y servicios que permitan el cumplimiento de los objetivos planteados.
- Crear la infraestructura para que la comunidad de la Facultad tenga acceso a la información que requiera.

### **Funciones:**

- Procesamiento.
- Captura de información.
- Soporte Técnico.
- Capacitación.
- Diseño y desarrollo de sistemas.
- Consultoría.
- Reinscripciones.
- Registro a exámenes extraordinarios.
- Seguridad de los sistemas.
- Diseño de bases de datos.
- Administración y monitoreo.

### 1.6.5 Organigrama





## Sistema de inventario



# Capítulo 2

*Marco Teórico*

---

## **2.1 Introducción**

La Ingeniería de Software es el establecimiento y uso de principios de ingeniería para obtener software confiable y que funcione eficientemente en equipos de cómputo.

La Ingeniería de Software comienza con las Técnicas de Programación Estructurada, incorporándolas a las fases del ciclo vital de software. Y fue seguida por otros métodos estructurados de análisis y métodos estructurados de diseño. Además, comenzaron a usarse tecnologías orientadas a objetos. En un principio la programación era la tarea de oro de la Ingeniería de Software pero ahora la ingeniería y el diseño de requisitos son más importantes.

En los años noventa la gerencia de proyecto ganó interés y llegó a ser un componente importante en la Ingeniería de Software. En la década pasada, los estándares de la Ingeniería de Software han caracterizado la industria del software como una disciplina madura.

En un nivel más técnico, la Ingeniería de Software comienza con una serie de tareas que hacen modelos y que resultan en una especificación completa de requisitos y una representación comprensiva de diseño del software que será construido. Se han desarrollado muchos métodos para hacer modelos de sistemas de información. Sin embargo, los métodos Orientados a Objeto (OO) han llegado a ser el estándar.

Para ciertos sistemas de información críticos, el uso de métodos formales es recomendado en el ciclo de vida del software, para producir sistemas con la integridad más alta. Los métodos formales confían en las técnicas matemáticas que expresan y modelan los requisitos de cualquier producto en el ciclo vital del software.

El Desarrollo de Software Orientado a Objetos comenzó en los años ochenta como una etapa natural de los métodos estructurados. UML (Unified Modeling Language) ha emergido como una unificación de los diversos métodos orientados a objetos y se está convirtiendo en un estándar.

Actualmente, la tecnología de componente es un método para desarrollar sistemas de información que está creciendo. A diferencia de los métodos tradicionales, la tecnología de componente los ensambla para formar una solución de software. Actualmente, hay dos estándares de componentes en competencia: JavaBeans por Sun y DCOM por Microsoft. Los componentes de software son materiales reusables para construir sistemas de software.

## **2.2 Modelos de Procesos**

En este capítulo se analizarán los modelos de procesos más relevantes para identificar cuál será el modelo a utilizar para el desarrollo del sistema.

### **2.2.1 Modelo Lineal Secuencial (MLS)**

También llamado "Ciclo de vida básico" o "Modelo de cascada" tiene su origen en el "Modelo de cascada" desarrollado por Winston Royce, aunque omite los muchos ciclos de este último.

El MLS sugiere un enfoque sistemático o más bien secuencial del desarrollo de software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

El MLS acompaña las siguientes actividades:

- **Análisis de los requerimientos del software:** es la fase en la cual se reúnen todos los requisitos que debe cumplir el software. En esta etapa es

## *Capítulo 2. Marco Teórico*

fundamental la presencia del cliente que documenta y repasa dichos requisitos.

- **Diseño:** es una etapa dirigida hacia la estructura de datos, la arquitectura del software, las representaciones de la interfaz y el detalle procedimental (algoritmo). En forma general se hace un esbozo de lo solicitado y se documenta haciéndose parte del software.
- **Generación del código:** es la etapa en la cual se traduce el diseño para que sea comprensible por la máquina. Esta etapa va a depender estrechamente de lo detallado del diseño.
- **Pruebas:** esta etapa se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en la detección de errores.
- **Mantenimiento:** debido a que el programa puede tener errores, puede eventualmente acoplarse a los cambios en su entorno. Esto quiere decir que no se rehace el programa, sino que sobre la base de uno ya existente se realizan cambios necesarios.

El MLS es el paradigma de desarrollo de software más antiguo que existe, sin embargo esto no ha impedido que se haya creado una desconfianza alrededor de él basada en los siguientes errores reales:

- Los proyectos raramente siguen el paradigma secuencial que propone el proyecto.
- A menudo es difícil que el cliente exponga exactamente todos los requisitos.
- El cliente debe tener paciencia.
- Los responsables del desarrollo de software siempre se retrasan innecesariamente.

Todo lo anteriormente expuesto es cierto pero este paradigma tiene un lugar bien definido e importante en el trabajo de la Ingeniería de Software a parte de proporcionar una plantilla en la que se encuentran métodos para análisis, diseño,



codificación, pruebas y mantenimiento. Con todo y sus errores, sigue siendo el paradigma más utilizado en el desarrollo del software, siendo mucho mejor que un enfoque al azar.

**Ventaja:**

- Proporciona una plantilla en la que se encuentran métodos para análisis, diseño, pruebas y mantenimiento; siendo el modelo más extensamente usado por la ingeniería de software.

**Desventajas:**

- Los sistemas reales rara vez son lineales.
- Requiere paciencia por parte del cliente, ya que un avance significativo solo se ve en las etapas finales.

**2.2.2 Modelo de Construcción de Prototipos**

Este modelo comienza con la recolección de los requisitos. El desarrollador y el cliente definen los objetivos globales para el software, identifica los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces planteamos con rapidez una iteración de construcción de prototipos y se presenta el modelado (en forma de un diseño rápido).

El diseño rápido se basa en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final (por ejemplo, la configuración de la interfaz con el usuario y el formato de los despliegues de salida). El diseño rápido conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La iteración ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo

tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

**Ventajas:**

- No modifica el flujo de ciclo de vida.
- Posibilidad de modificar el sistema en las primeras etapas de desarrollo.
- Oportunidad de suspender el desarrollo de un sistema no funcional.
- Posibilidad de desarrollar un sistema que se acerque más a satisfacer las necesidades y expectativas de los usuarios.
- Reduce costos y aumenta la probabilidad de éxito.
- Exige disponer de las herramientas adecuadas.

**Desventajas:**

- Se presenta dificultad para manejar la elaboración de prototipos como un proyecto en el esfuerzo de sistemas más grandes.
- Los usuarios y los analistas podrían adoptar un prototipo como si fuera un sistema final cuando de hecho es deficiente y su propósito nunca fue servir como sistema terminal.

**2.2.3 Modelo DRA (Desarrollo Rápido de Aplicaciones)**

Es una adaptación “a alta velocidad” del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una construcción basada en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un sistema completamente funcional dentro de periodos más cortos de tiempo.

## *Capítulo 2. Marco Teórico*

Hay tres fases del DRA que vinculan a usuarios y analistas a la evaluación, diseño e implementación.

- Fase de planeación de requerimientos: Involucra a usuarios y analistas para identificar los objetivos de la aplicación y para identificar los requerimientos de información que surgen de dichos objetivos.
- Fase de diseño: Durante esta fase los usuarios responden a los prototipos operativos reales y los analistas refinan los módulos diseñados, basados en respuestas del usuario.
- Fase de implementación: Después de diseñar los aspectos del negocio o no técnicos del sistema y los sistemas sean construidos y se refinen, los nuevos sistemas son probados e introducidos a la organización.

### **Ventaja:**

- Creación rápida de aplicaciones.

### **Desventajas:**

- Para proyectos grandes aunque por escala, el DRA requiere recursos humanos suficientes como para crear el número correcto de equipos DRA.
- Requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo corto.
- Si no hay compromiso de ninguna de las partes, los proyectos DRA fracasarán.
- No todos los tipos de aplicaciones son apropiadas para el DRA.
- El DRA no es adecuada cuando los riesgos técnicos son altos.

## **2.2.4 Modelos Evolutivos del Proceso del Software**

Los modelos evolutivos del proceso del software son modelos iterativos que permiten a los ingenieros el desarrollo de software en versiones cada vez más completas.

### **2.2.4.1 Modelo en Espiral**

Es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. El software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

Generalmente el modelo en espiral se divide en un número de actividades de marco de trabajo, algunas de las cuáles son:

- **Comunicación con el cliente:** Se definen las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- **Planificación:** Son las tareas requeridas para definir recursos, el tiempo y demás información relacionadas con el proyecto.
- **Análisis de riesgo:** Las tareas requeridas para evaluar riesgos técnicos y de gestión.
- **Ingeniería:** Las tareas requeridas para construir una o más representaciones de la aplicación.
- **Construcción y acción:** Tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.

## *Capítulo 2. Marco Teórico*

- Evaluación con el cliente: Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación

### **Ventajas:**

- El coste y la planificación se ajustan con la realimentación ante la evaluación del cliente.
- El gestor de proyecto ajusta el número planificado de iteraciones requeridas para completar el software.
- El modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.
- El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto.
- Reduce los riesgos antes de que éstos lleguen a ser problemáticos.

### **Desventajas:**

- Puede resultar difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.
- Requiere una considerable habilidad para la evaluación de riesgo.
- Si no se realiza correctamente la etapa de análisis, los riesgos se pueden convertir en grandes problemas.

#### **2.2.4.2 Modelo de Desarrollo Concurrente**

El modelo del desarrollo concurrente se puede representar en forma de un esquema como una serie de actividades técnicas importantes, tareas y estados asociados a ellas. Este modelo define una serie de acontecimientos que disparan

transiciones de estado a estado para cada una de las actividades de la ingeniería de software.

El modelo de proceso concurrente se utiliza a menudo como el paradigma de desarrollo de aplicaciones cliente/servidor. Cuando se aplica cliente/servidor, el modelo de proceso concurrente define actividades en dos dimensiones: una dimensión de sistemas y una dimensión de componentes. Los aspectos a nivel de sistemas se afrontan mediante tres actividades: diseño ensamblaje y uso. La dimensión de componentes se afronta con dos actividades: diseño y realización.

La concurrencia se logra de dos formas:

- Las actividades de sistemas y componentes ocurren simultáneamente y pueden modelarse con el enfoque orientado a objetos.
- Una aplicación cliente/servidor típica se implementa con muchos componentes, cada uno de los cuales se puede diseñar y realizar concurrentemente.

#### **Ventajas:**

- El modelo de proceso concurrente es aplicable a todo tipo de desarrollo de software.
- Proporciona una imagen exacta del estado actual de un proyecto.

#### **2.2.5 Proceso Unificado de Desarrollo (RUP)**

El proceso unificado de desarrollo es el resultado de más de tres décadas de desarrollo y uso. Creado por Ivar Jacobson, Grady Booch y James Rumbaugh mismos autores del lenguaje unificado de modelado (UML).

## *Capítulo 2. Marco Teórico*

Es un proceso de desarrollo de software, es decir; el conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema software. La parte que hace potente al Proceso Unificado de Desarrollo es que más de ser 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 proyecto.

El proceso Unificado utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software.

Son tres los aspectos que definen al Proceso Unificado

- Dirigido por casos de uso.
- Centrado en la arquitectura.
- Iterativo e Incremental.

### **2.2.5.1 Dirigido por casos de uso**

Un sistema software es creado para dar servicio a sus usuarios. Por tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean.

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan requisitos funcionales del sistema. Los casos de uso no son sólo una herramienta para especificar los requisitos de un sistema. También guían su diseño, implementación y prueba; esto es, guían el proceso.

### **2.2.5.2 Centrado en la arquitectura**

El papel de la arquitectura software es parecido al papel 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.

La arquitectura software surge de las necesidades de la empresa, como las perciben los usuarios y los inversores, y se refleja en los casos de uso. También se ve influida por otros factores como la arquitectura hardware, sistema operativo, sistema de gestión de bases de datos, protocolos para la comunicación en red, consideraciones implementación, sistema heredados y requisitos no funcionales.

### **2.2.5.3 Iterativo e Incremental**

La construcción de software supone un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas. Cada parte 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.

### **2.2.5.4 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 (fig. 2.1).



## Capítulo 2. Marco Teórico

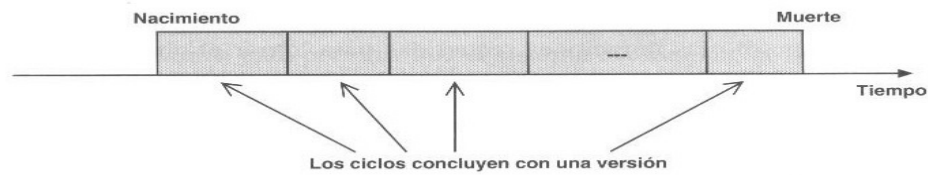


Fig. 2.1 La vida de un proceso consta de ciclos desde su nacimiento hasta su muerte.

Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición.

### 2.2.5.5 Fases Dentro de un Ciclo

Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo, a su vez, se divide en cuatro fases (fig. 2.2). A través de una secuencia de modelos, los implicados visualizan lo que está sucediendo en esas fases. Dentro de cada fase, los directores o los desarrolladores pueden descomponer adicionalmente el trabajo.

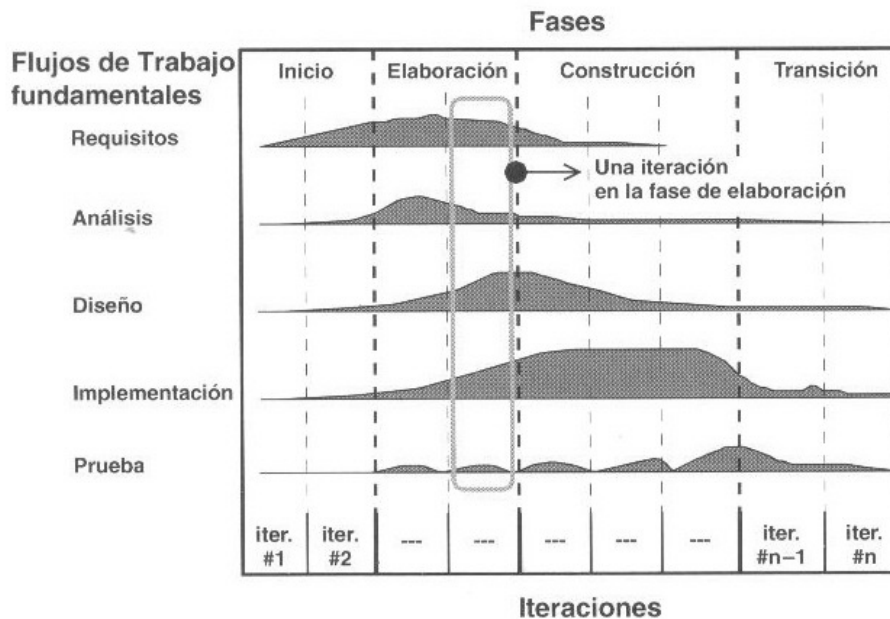


Fig. 2.2 Los cinco flujos de trabajo –requisitos, análisis, diseño, implementación y prueba- tienen lugar sobre las cuatro fases: inicio, elaboración, construcción y transición

## *Capítulo 2. Marco Teórico*

- Fase de inicio: Se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto.
- Fase de elaboración: Se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema.
- Fase de construcción: Se crea el producto. En esta fase, la línea de la arquitectura crece hasta convertirse en el sistema completo.
- Fase de transición: Cubre el periodo durante el cual el producto se convierte en versión beta.

Con base a las características de las distintas metodologías para el desarrollo de software se encontró que la más adecuada a utilizar será el Proceso Unificado de Desarrollo de Software, por presentar las siguientes ventajas para el proyecto:

- Es un proceso estandarizado.
- Está enfocado al diseño Orientado a Objetos.
- Es iterativo e incremental, lo que permite añadir al sistema nuevas funcionalidades.
- Ya que está dirigido por los casos de uso, permite la construcción de un sistema que se apegue más a los requerimientos de los usuarios.
- Utiliza UML como lenguaje de modelado.

### **2.3 Tecnología a Utilizar**

#### **2.3.1 SUN SPARC**

##### **2.3.1.1 Principales Características:**

Su característica distintiva es utilizar ventanas de registros.

- 32 registros de enteros de 32 bits.

- 16 registros de punto flotante de 64 bits (para el caso de doble precisión) que se pueden utilizar como 32 registros de 32 bits (para precisión simple).
- Modos de direccionamiento:
  - Inmediato, (constantes de 13 bits).
  - Directo (offset de 13 bits).
  - Indirecto (registro + offset de 13 bits o registro + registro).
- Utiliza instrucciones retardadas (saltos, load y store).
- Manejo de memoria:
  - Espacio virtual de 4 Gigabytes.
  - Unidad de manejo de memoria (MMU) que trabaja con páginas de tamaño configurable.

### **2.3.1.2 Categorías de Instrucciones**

La arquitectura SPARC tiene cerca de 50 instrucciones enteras, unas pocas más que el anterior diseño RISC, pero menos de la mitad del número de instrucciones enteras del 6800 de Motorola.

Las instrucciones de SPARC se pueden clasificar en cinco categorías:

- LOAD y STORE.
- Instrucciones Aritméticas /Lógicas /Shift.
- Operaciones del Coprocesador.
- Instrucciones de Control de Transferencia.
- Instrucciones de control de registros Read/Write.

### **2.3.1.3 Ventanas de Registros**

La especificación de la arquitectura indica que se pueden tener 32 registros "visibles" divididos en grupos de 8.

- De r0 a r7, Registros GLOBALES.

- De r7 a r15, Registros SALIDA.
- De r15 a r23, Registros LOCALES.
- De r24 a r31, Registros ENTRADA.

#### **2.3.1.4 Interrupciones**

El diseño SPARC soporta un set total de interrupciones, que son manejados por una tabla que soporta 128 interrupciones de hardware y 128 interrupciones de software. Sin embargo las instrucciones de punto flotante pueden ejecutarse concurrentemente con las instrucciones enteras.

#### **2.3.1.5 Protección de Memoria**

Algunas instrucciones SPARC son privilegiadas y pueden ser ejecutadas únicamente mientras el procesador esta en modo supervisor. Estas instrucciones ejecutadas en modo protegido aseguran que los programas de usuario no sean accidentalmente alterados por el estado de la máquina con respecto a sus periféricos y viceversa. El diseño SPARC también proporciona protección de memoria, que es esencial para las operaciones multitarea.

### **2.3.2 Solaris (SUN)**

Solaris es un sistema operativo desarrollado por Sun Microsystems. Es un sistema certificado como una versión de UNIX. Solaris puede considerarse uno de los sistemas operativos más avanzados.

Solaris nace con el fin de convertirse en la plataforma más completa para desarrollar aplicaciones en Java y servicios web.

En total, Solaris incorpora más de 300 mejoras basadas en la estrategia, de interoperabilidad, Internet, innovación e integración. Gracias a estas mejoras se

consigue, según los responsables de Sun, un gran ahorro de tiempo y dinero, un mayor rendimiento y la capacidad de ejecutar aplicaciones de negocio utilizando Java, una gran seguridad, gracias a la integración en el entorno de un cortafuegos empresarial, una gestión más sencilla de los recursos al estar éstos integrados, mayor disponibilidad a través del software Sun Cluster.

### **2.3.3 Sybase ASE para Solaris**

Adaptive Server Enterprise (ASE) es el motor de bases de datos (RDBMS) insignia de la compañía Sybase, Inc. ASE es un sistema de gestión de datos, altamente escalable, de alto rendimiento, con soporte a grandes volúmenes de datos, transacciones y usuarios, y de bajo costo, que permite:

- Almacenar datos de manera segura.
- Tener acceso y procesar datos de manera inteligente.
- Movilizar datos.

#### **2.3.3.1 Principales Características**

ASE incluye características como:

- Un optimizador de consultas completamente renovado y más inteligente.
- Técnicas de particionamiento semántico de tablas que aumentan la velocidad de acceso a los datos.
- Columnas cifradas para mayor seguridad de los datos.
- Columnas computadas "virtuales" y materializadas, e índices funcionales, que brindan mayor rendimiento.
- Mejoras al lenguaje Transact-SQL, para mayor productividad.
- Mejoras a los servicios de Java y XML en la base de datos.
- Mejoras a los servicios para consumo y publicación de Servicios Web.
- Herramientas mejoradas para la administración y el monitoreo.
- Más rendimiento y menor costo total de propiedad.

## Capítulo 2. Marco Teórico

- Arquitectura VSA de Sybase.
- Administrador lógico de recursos y tareas.
- Múltiples esquemas de bloqueo de datos.
- Copias de respaldo en línea y de alto rendimiento.
- Integración transparente con orígenes de datos remotos.
- Programador de tareas.
- Conexiones seguras con SSL.
- Soporte a LDAP para autenticación de usuarios y conectividad cliente/servidor.
- Soporte a múltiples herramientas de desarrollo y lenguajes de programación, como PowerBuilder, Visual Basic, Java, C, PHP, etc.
- Soporte a múltiples protocolos de conectividad, como Open Client (propio de Sybase), ODBC, OLE DB, ADO.NET y JDBC.

Es claro entonces que maneja y accesa muchas bases de datos a la vez y por tal motivo optimiza las operaciones en la base de datos y controla las conexiones de los usuarios.

En muchos sistemas las aplicaciones se ejecutan en dos procesos, uno que corre en el cliente y otro en el servidor. Sin embargo en Sybase y gracias a que todas las conexiones son manejadas por un solo proceso, el número de procesos por cliente se reduce prácticamente a la mitad, pues en este caso sólo se necesita un solo proceso en el servidor para todos los clientes y un proceso para estos últimos. Es por esto que se considera que SQL Server tiene una arquitectura multi-hilos ya que por cada acceso o hilo entra un proceso de forma similar a como trabaja un sistema operativo.

### **Ventajas:**

- Rapidez: Consultas hasta 100 veces más rápidas que un sistema de gestión de base de datos (SGBD) tradicional.

- Menor costo total de la propiedad: Utiliza algoritmos de compresión sofisticados que reducen en volumen de los datos que se van a almacenar en hasta un 70 por ciento.
- Más fácil: Más fácil de mantener que las bases de datos tradicionales y no exige ajustes que requieren mucho tiempo y recursos para la obtención de un rendimiento excelente.
- Más escalable: Ofrece escalabilidad de datos y usuarios casi lineal para admitir miles de usuarios y terabytes de datos. También admite la multiplexación (agrupaciones), una potente función, sobre todo en entornos Linux donde la escala de multiprocesador simétrico es limitada.
- Amplia compatibilidad con servicios Web: Simplifica el desarrollo de aplicaciones de servicios Web con bases de datos, la integración con sistemas existentes y la implementación de aplicaciones en entornos de TI diversos.

#### **2.3.4 Java**

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

##### **2.3.4.1 Principales Características**

- Simple: Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características, menos usadas y más confusas que presentan otros lenguajes.

## Capítulo 2. Marco Teórico

- Reduce errores de programación como:
  - Aritmética de punteros.
  - No existen referencias.
  - Registros (struct).
  - Definición de tipos (typedef).
  - Necesidad de liberar memoria (free).
- Orientado a Objetos: Soporta las características propias del paradigma de la orientación a objetos: encapsulamiento, herencia y polimorfismo.
- Distribuido: Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.
- Robusto: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación, como en tiempo de ejecución.
- Seguro: El código Java pasa muchos tests antes de ejecutarse en una máquina.

Si los ByteCodes pasan la verificación sin generar ningún mensaje de error, entonces sabemos que:

- El código no produce desbordamiento de operandos en la pila.
- El tipo de los parámetros de todos los códigos de operación son conocidos y correctos.
- No ha ocurrido ninguna conversión ilegal de datos, tal como convertir enteros en punteros.
- El acceso a los campos de un objeto se sabe que es legal: public, private, protected.
- No hay ningún intento de violar las reglas de acceso y seguridad establecidas.



- **Portable:** Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, PC o Mac.
- **Interpretado y compilado a la vez:** Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).
- **Multihilo:** Java soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas.
- **Dinámico:** El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red.

#### **2.3.4.2 Java Server Pages (JSP)**

Java Server Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología es un desarrollo de la compañía Sun Microsystems.

Las JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas.

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código

Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.

JSP -> Servidor Aplicaciones (Servlets) -> Cliente (Navegador)

El rendimiento de una página JSP es el mismo que tendría el servidor equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

La principal ventaja de JSP frente a otros lenguajes es que el lenguaje Java es un lenguaje de propósito general que excede el mundo web y que es apto para crear clases que manejen lógica de negocio y acceso a datos. Esto permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento HTML en el archivo JSP.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios.

Los servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, los CGIs (common gateway interface), programas que generan páginas web en el servidor, o ASP (Active Server Pages), un método específico de Microsoft. Sin embargo, se diferencian de ellos en otras cosas.

Para empezar, los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, se puedan usar en cualquier tipo de equipo, siempre que exista

una máquina virtual Java para él. Cada JSP se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo. Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones.

Los JSPs son en realidad servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web

Para ejecutar las páginas JSP, se necesita un servidor Web con un contenedor Web que cumpla con las especificaciones de JSP y de Servlet. Tomcat es una completa implementación.

### **2.3.5 TOMCAT**

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation.

Tomcat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems

Tomcat es un servidor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

## *Capítulo 2. Marco Teórico*

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.



Sistema de  
inventario



# *Capítulo 3*

## *Definición de Requerimientos*

---

### **3. Requerimientos de Usuario**

#### **3.1.1 Definición de Requerimiento**

Es una característica del sistema o bien una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito de éste.

#### **3.1.2 Clasificación**

- **Requerimiento Funcional:** Describe las interacciones entre el sistema y su ambiente, de forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactúa el sistema. Este tipo de requerimientos también especifica el comportamiento de entrada y salida del sistema ante un determinado estímulo. Describiendo los estados aceptables para el sistema.
- **Requerimientos no funcionales:** Describe los aspectos del sistema visible por el usuario que no se relacionan de forma directa con el comportamiento funcional, como restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad, o fiabilidad.

### **3.2 Modelo del Negocio**

El Modelo del Negocio se compone de dos modelos que nos permiten comprender el entorno del sistema: el modelo de casos de uso del Negocio y el modelo de objetos del Negocio.

- **Modelo de casos de uso del negocio:** describe los procesos de negocio de una empresa o institución en términos de caso de uso del negocio, actores del

negocio que se corresponden con los procesos del negocio y los clientes, respectivamente.

- **Modelo de objetos del Negocio:** describe cómo cada caso de uso del negocio se lleva a cabo por el conjunto de trabajadores del negocio. Cada realización de un caso de uso se puede mostrar en diagramas de interacciones y diagramas de actividades de UML.

La simbología utilizada para representar los elementos del modelo del negocio se presentan en la siguiente figura (fig. 3.1).

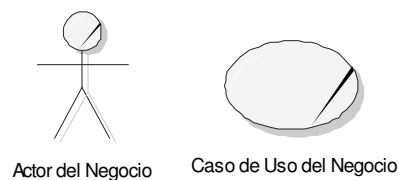


Fig. 3.1. Actor y Caso de Uso del Negocio.

### **3.3 Modelo del Negocio para el Sistema de Inventario**

La captura de requerimientos consiste en hacer una descripción detallada de la funcionalidad que los usuarios desean del sistema, para ello hay que realizar dos modelos, que son: el modelo del negocio, y el modelo de casos de uso.

El modelo del negocio describe el entorno del sistema, mientras que el modelo de casos de uso, el comportamiento funcional y no funcional.

#### **3.3.1 Modelo de casos de uso del Negocio**

El sistema de inventario que maneja la SSA de la FI comprende distintos casos de uso, entre los cuales podemos identificar los siguientes:

### *Capítulo 3. Definición de Requerimientos*

Caso de Uso Captura de Inventario de Equipo. En este caso de uso del negocio la persona encargada de capturar el inventario recolecta la información de cada uno de los equipos que se encuentran dentro de la SSA. Considerando los siguientes pasos:

- a) Revisión del equipo a capturar.
- b) Llenado de la forma de inventario.
- c) Almacenamiento de la forma de inventario.

Caso de Uso Actualización de Inventario de Equipo. En este caso de uso del negocio la persona encargada del inventario ordena la realización de un nuevo inventario de equipo.

- a) Envía una persona para que realice la captura de inventario.
- b) Llenado de nuevas formas de inventario.
- c) Reemplazo de las antiguas por las nuevas formas de inventario.

Caso de Uso Generación de Censos. Este caso de uso del negocio es utilizado por el jefe de soporte técnico para generar los informes de Censo solicitados por la DGSCA.

- a) Ordenación de equipos basados en el tipo de censo.
- b) Transcripción del reporte en el formato solicitado.
- c) Envío de la información a la DGSCA.

Caso de Uso Solicita Compra. Este caso de uso del negocio es utilizado por el jefe de área cuando se requiere una nueva adquisición del algún equipo, o accesorios relacionados con el cómputo.

- a) Llenado de la forma de compra.
- b) Envío de la forma de compra a bienes y suministros.

Caso de Uso *Solicita Mantenimiento*. Este caso de uso del negocio es utilizado por el jefe de área cuando un equipo o accesorio necesita mantenimiento.



### Capítulo 3. Definición de Requerimientos

- a) Llenar la forma de solicitud de mantenimiento.
- b) Envió de la forma de compra a bienes y suministros.

Caso de Uso *Solicita Baja*. Este caso de uso del negocio es utilizado por el jefe de área cuando un equipo o accesorio se ha vuelto obsoleto y se requiere registrar su estado.

- a) Llenado de la forma de baja.
- b) Envió de la forma de baja a bienes y suministros.

El modelo de los casos de uso mencionados anteriormente se representa en la siguiente figura (fig. 3.2).

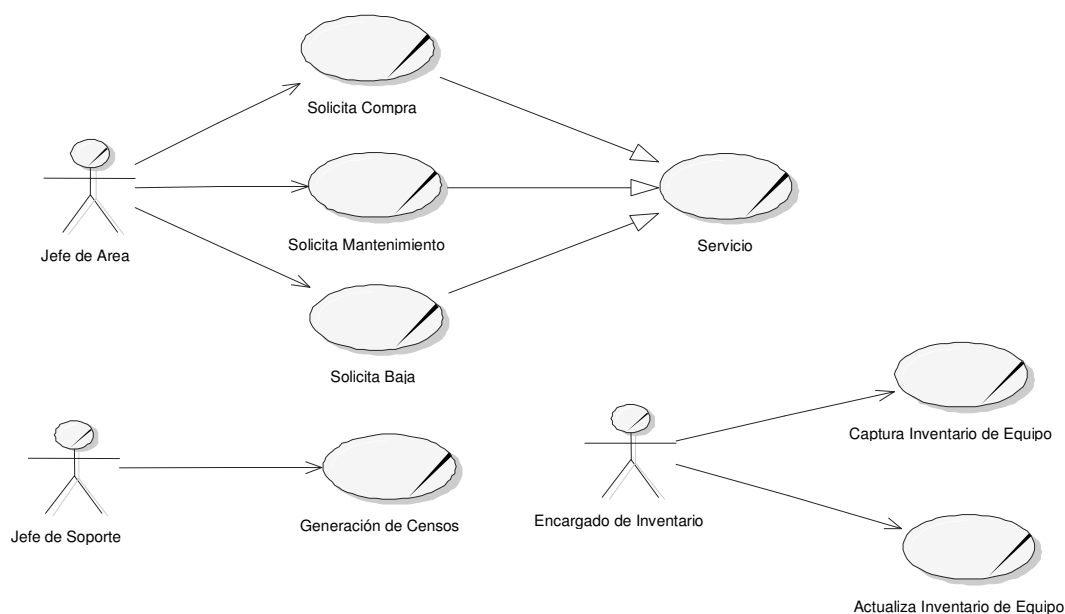
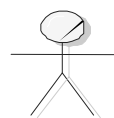


Fig. 3.2 Modelo de Casos de Uso del Negocio.

Los actores que intervienen en el negocio son los siguientes.

- *Jefe de Área*: es aquel actor que se encarga de realizar distintas peticiones al sistema ya sea como la adquisición de un nuevo equipo, actualización del equipo o reparación de equipos (fig. 3.3).

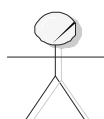
### Capítulo 3. Definición de Requerimientos



Jefe de Area

Fig. 3.3 Actor del Negocio Jefe de área

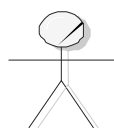
- *Encargado de Inventario*: es un actor que se encarga de levantar el inventario mediante los formatos impresos de la SSA para tener registrados los equipos de cómputo que pose (fig. 3.4).



Encargado de Inventario

Fig. 3.4 Actor del negocio Encargado de Inventario.

- *Jefe de Soporte Técnico*: es el actor responsable de la realización del inventario y de los censos (fig. 3.5).



Jefe de Soporte

Fig. 3.5 Actor del negocio Jefe de Soporte Técnico.

#### 3.3.2 Modelo de Objetos del Negocio

##### *Caso de Uso Solicitar Servicio.*

El jefe de área da los siguientes pasos en el caso de uso del negocio solicitar servicio (fig. 3.6).

- a) El *Jefe de Área* solicita un *Servicio* (Compra, Baja o Mantenimiento).
- b) El *Jefe de Área* llena la forma de servicio.
- c) El *Jefe de Área* envía la forma a bienes y suministros.

### Capítulo 3. Definición de Requerimientos

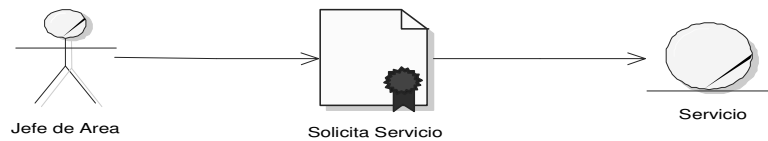


Fig. 3.6 Diagrama de Objeto del Casos de Uso Solicita Servicio.

#### Caso de uso Generar Censo.

El *Jefe de Soporte* técnico realiza los siguientes pasos para generar el censo (fig. 3.7).

- a) El *Jefe de Soporte* técnico fija los criterios para la generalización del censo a través de la interfaz (Genera Censo).
- b) El *Jefe de Soporte* técnico recibe el censo por parte de la interfaz.

La interfaz *Genera Censo* son los estudiantes de servicio social.

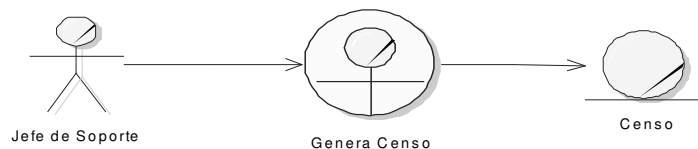


Fig. 3.7 Diagrama de Objeto del Casos de Uso Generar Censo.

#### Casos de Uso Captura y Actualiza Inventario.

El *Jefe de Soporte* técnico utiliza estos casos de uso para capturar y actualizar la información del inventario. Mediante los siguientes pasos (fig. 3.8).

- a) El *Jefe de Soporte* técnico solicita a la interfaz Captura o Actualiza Inventario.
- b) La interfaz retorna el inventario.

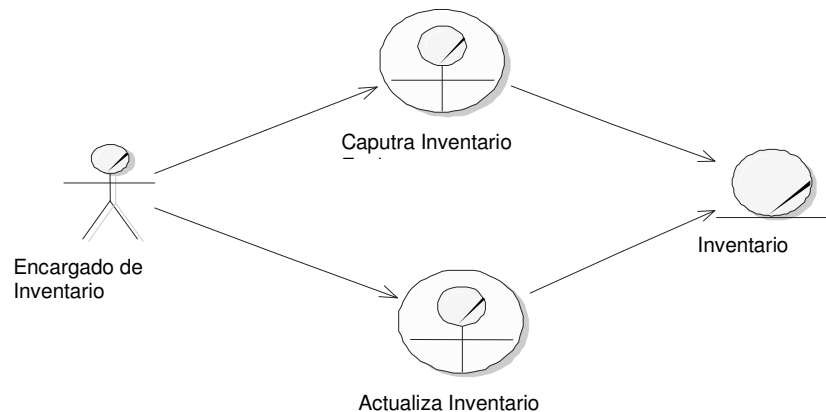


Fig. 3.8 Diagrama de Objeto del Casos de Uso Captura y Actualiza Inventario.

### 3.4 Modelo de Casos de Uso

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

Este artefacto se basa en la descripción de elementos o usuarios externos al sistema y de la funcionalidad del sistema. Un modelo de casos de uso describe los requerimientos funcionales de un actor en términos de las interacciones que éste ejecuta con el sistema. El modelado de casos de uso es una técnica efectiva y a la vez simple para modelar los requerimientos del sistema desde la perspectiva del usuario. Presenta el sistema desde la vista de su uso y esquematiza como proporcionará valor a los usuarios.

El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores para limitar las funciones con que dispondrá el sistema luego de ser implementado, proporcionará la entrada fundamental para el análisis, el diseño, la implementación y las pruebas.

Este modelo está formado por los diagramas de casos de uso y el flujo de sucesos.

### 3.4.1 Actor

Es conjunto coherente de roles que desempeñan los usuarios sobre los casos de uso cuando interactúan con estos. Por lo general un actor tiene correspondencia directa con los usuarios del sistema, también los sistemas externos con los que interactúa el nuevo sistema se representa como un actor (fig. 3.9).



Fig. 3.9 Representación UML de un actor.

### 3.4.2 Caso de uso

Es cada una de las maneras en que un usuario utiliza el sistema. Conduce a un resultado observable de interés para un actor determinado (fig. 3.10).

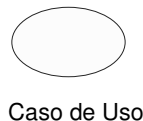


Fig. 3.10 Representación UML de un Caso de Uso.

### 3.4.3 Flujo de sucesos

Es la descripción textual de la secuencia de acciones que realizará el caso de uso, esta descripción se puede dividir en varias secciones:

- Descripción; en esta sección se presenta una pequeña descripción de lo que hace el caso de uso.

- El flujo principal (flujo básico): en el que se describen las actividades de respuesta a un estímulo del usuario y se supone que nada va mal.
- Flujos alternativos: se describen las acciones que realizará el sistema en caso de que ocurra un error en el flujo principal.
- Precondiciones: se describen las condiciones que provocaran que el caso de uso sea ejecutado.
- Poscondiciones: presenta una descripción de los resultados obtenidos después de la ejecución del caso de uso.

### 3.4.4 Construcción del Modelo de Casos de Uso

#### 3.4.4.1 Encontrar Actores

**Administrador:** El *Administrador* representa a un usuario del sistema que tendrá los más altos privilegios que se puedan otorgar en este sistema de inventario, podrá hacer modificaciones en los usuarios, revisar los reportes sobre movimientos de la información en el sistema, dar de alta nuevos usuarios y generar los reportes sobre el censo en el sistema (fig. 3.11).



Fig. 3.11 Actor Administrador.

**Encargado de Mantenimiento:** El *Encargado de Mantenimiento* representa a los usuarios que se encargarán de dar mantenimiento en el sistema, actualizando la información que es utilizada para agilizar la captura del inventario (fig. 3.12).



Fig. 3.12 Actor Encargado de Mantenimiento.

**Encargado de Inventario:** El *Encargado de Inventario* representa al usuario que se encarga de la captura de los equipos, equipos de red y accesorios (fig. 3.13).



Fig. 3.13 Actor Encargado de Inventario.

**Jefe de Área:** El *Jefe de Área* representa a los usuarios que tienen privilegios para ordenar una compra o solicitar mantenimiento a los equipos (fig. 3.14).



Fig. 3.14 Actor Jefe de Área.

El actor usuario es un actor que generaliza los roles comunes para todos los actores del sistema (fig. 3.15).

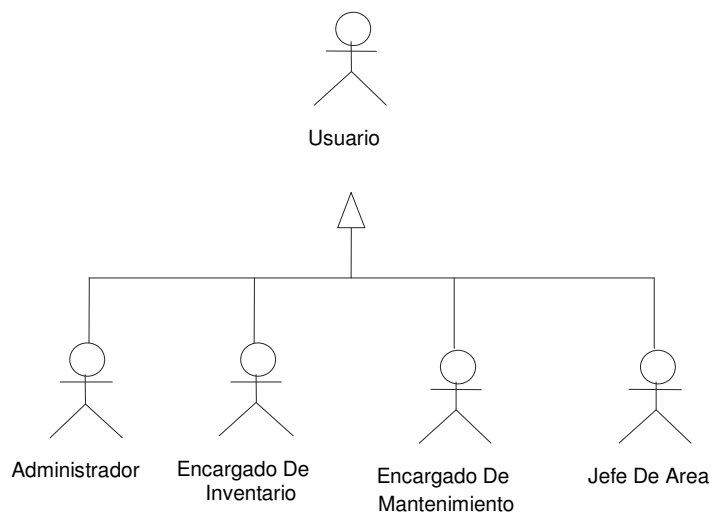


Fig. 3.15 Generalización de los actores.

#### **3.4.4.2 Encontrar casos de uso**

A partir del modelo del Negocio se obtuvieron los casos de uso que corresponden directamente con algunos del modelo mencionado, además de agregar otros para dar funcionalidad al sistema; todos ellos se detallan en la descripción general.

##### **3.4.4.2.1 Descripción general**

A partir del modelo del Negocio y de la captura de requisitos adicionales se obtuvieron los siguientes Casos de Uso, organizados por los procesos del Negocio que soportan.

Para registrar la información de los equipos, mediante los casos de uso *Captura Equipo*, *Agregar Dispositivo*, *Agregar CPU*, *Agregar Periférico*, *Agregar Software* y *Agregar Accesorio* se lleva a cabo la captura de las propiedades hardware y software instalados en el equipo. Se incluye en esta sección los Casos de Uso *Captura Equipo de Red* y *Captura Accesorio*.

Los cambios en la información almacenada de los equipos esta soportada por los Casos de Uso: *Agregar Dispositivo*, *Agregar CPU*, *Agregar Periférico*, *Agregar Software*, *Agregar Accesorio*, *Quita Dispositivo*, *Quita CPU*, *Quitar Periférico*, *Quitar Software*, *Quitar Accesorio*, *Modificar Equipo*, *Modifica Modificar*, *Accesorio* y *Modificar Equipo de Red*.

Tanto el registro de equipos como los cambios en estos registros son utilizados por el actor Encargado de Inventario.

El actor *Encargado de Mantenimiento*, requiere funcionalidad para registrar elementos indispensables para la captura de equipo, accesorio y dispositivo; los



### Capítulo 3. Definición de Requerimientos

Casos de Uso relacionados con este proceso son: *Captura Marca, Captura Tipo, Captura Proyecto, Captura Ubicación, Captura Capacidad, Captura Resolución y Captura Tamaño Monitor.*

El actor *Jefe de Área* requiere que el sistema le permita llevar a cabo peticiones para compra, mantenimiento y baja de equipos, de ahí que se tengan los Casos de Uso: *Revisar Solicitudes, Solicitar Compra, Solicitar Mantenimiento y Solicitar Baja* que corresponden a los servicios respectivos.

Los casos de uso que el sistema presta al actor *Administrador* son los siguientes: *Alta Usuario y Baja Usuario*, por medio de ellos el *Administrador* controla los distintos usuarios que ingresan al sistema. *Consulta Bitácora*; este Caso de Uso es empleado para consultar informes sobre los movimientos que se han hecho en la información del sistema, quién los ha hecho y en qué momento se llevaron a cabo. *Genera Censo*; es un caso de uso muy importante ya que por medio de él el *Administrador* obtendrá los informes requeridos por la DGSCA.

Se presenta una lista de todos los casos de uso encontrados:

- Agrega Accesorio.
- Agrega CPU.
- Agrega Dispositivo.
- Agrega Periférico.
- Agrega Software.
- Alta Usuario.
- Baja Accesorio.
- Baja CPU.
- Baja Periférico.
- Baja Software.
- Captura Capacidad.
- Captura Equipo.
- Captura Equipo de Red.
- Captura Marca.
- Captura Proyecto.
- Captura Resolución.
- Captura Tamaño Monitor.
- Captura Tipo Conexión.
- Captura Tipo Equipo.
- Captura Tipo Equipo Red.

### Capítulo 3. Definición de Requerimientos

- Baja Usuario.
- Captura Accesorio.
- Captura Tipo Mouse.
- Captura Tipo RAM.
- Captura Tipo Software.
- Captura Velocidad.
- Captura Ubicación.
- Cerrar Sesión.
- Generar Censo.
- Consultar Bitácora.
- Inicia Sesión.
- Modificar Accesorio.
- Captura Tipo Lector Óptico.
- Captura Tipo Monitor.
- Modificar CPU.
- Modificar Equipo.
- Modificar Equipo de Red.
- Modificar Periférico.
- Quita Accesorio.
- Quita CPU.
- Quita Periférico.
- Quita Software.
- Solicitar Compra.
- Solicitar Mantenimiento.
- Solicitar Baja.

En la siguiente figura se muestran los casos de uso que son inherentes al administrador que son *Alta Usuario*, *Baja Usuario*, *Baja Accesorio*, *Baja CPU*, *Baja Equipo*, *Baja Equipo de Red*, *Baja Periférico* (fig. 3.16).

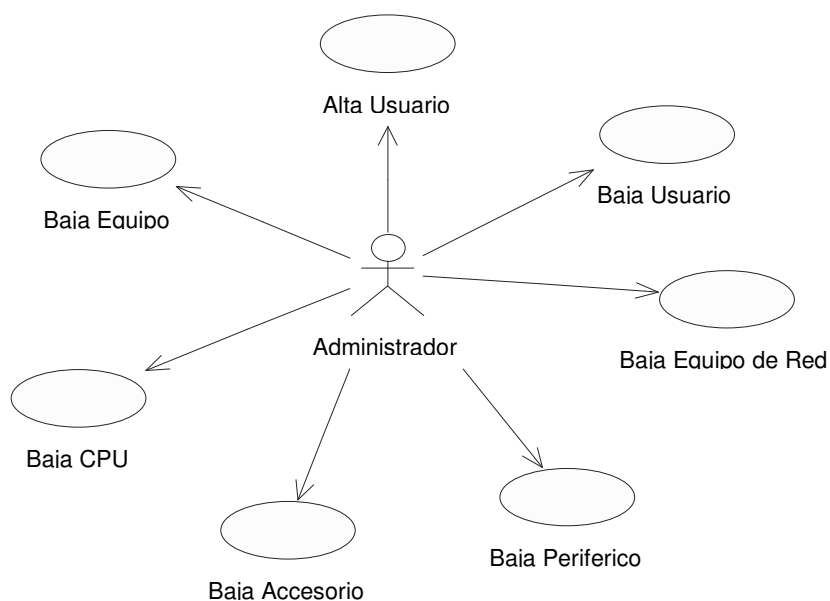


Fig. 3.16 Casos de usos del Administrador.

### Capítulo 3. Definición de Requerimientos

Los casos de uso que manipula el *Encargado de Inventario* son aquellos casos de uso en los cuales se deben capturar datos que sirven para la captura y modificación de los equipos, estos casos de usos son: *Captura Marca*, *Captura Proyecto*, *Captura Resolución*, *Captura Tamaño Monitor*, *Captura Tipo Bocinas*, *Captura Tipo Conexión*, *Captura Tipo Equipo*, *Captura Tipo Equipo Red*, *Captura Tipo Lector Óptico*, *Captura Tipo Monitor*, *Captura Tipo Mouse*, *Captura Tipo RAM*, *Captura Velocidad*, *Captura Tipo Software*, *Captura Ubicación* (fig. 3.17).

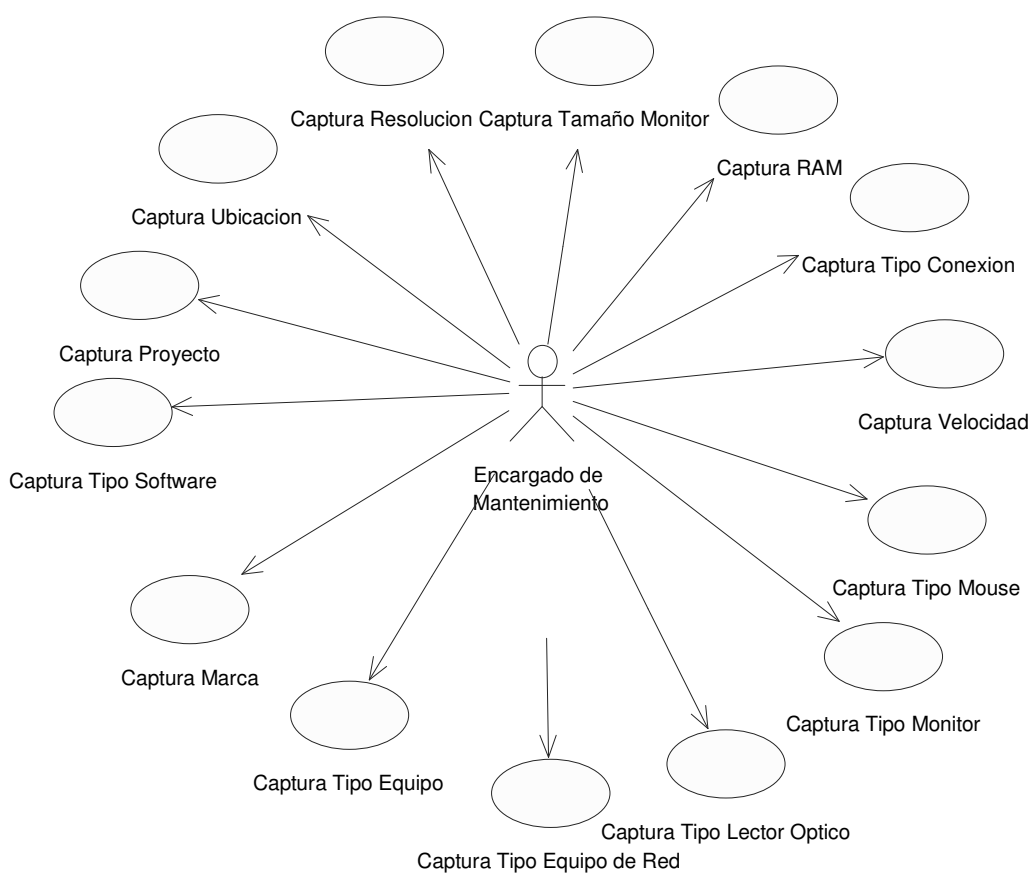


Fig. 3.17 Casos de uso del Encargado de Mantenimiento.

Los casos de uso del *Jefe de Área* (fig. 3.18) son tres casos de uso ya que sus responsabilidades se limitan a hacer solicitudes como compra, mantenimiento, y baja de equipos, para que después estas sean enviadas a quien corresponda atenderlas.

### Capítulo 3. Definición de Requerimientos

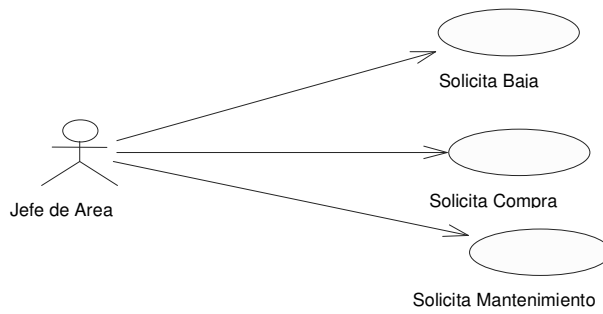


Fig. 3.18 Casos de uso del Jefe de Área.

La siguiente figura representa la relación la que existe entre los casos de uso de actor *Encargado de Inventario* y cómo interactúan entre ellos para poder registrar los equipos en el inventario (fig. 3.19).

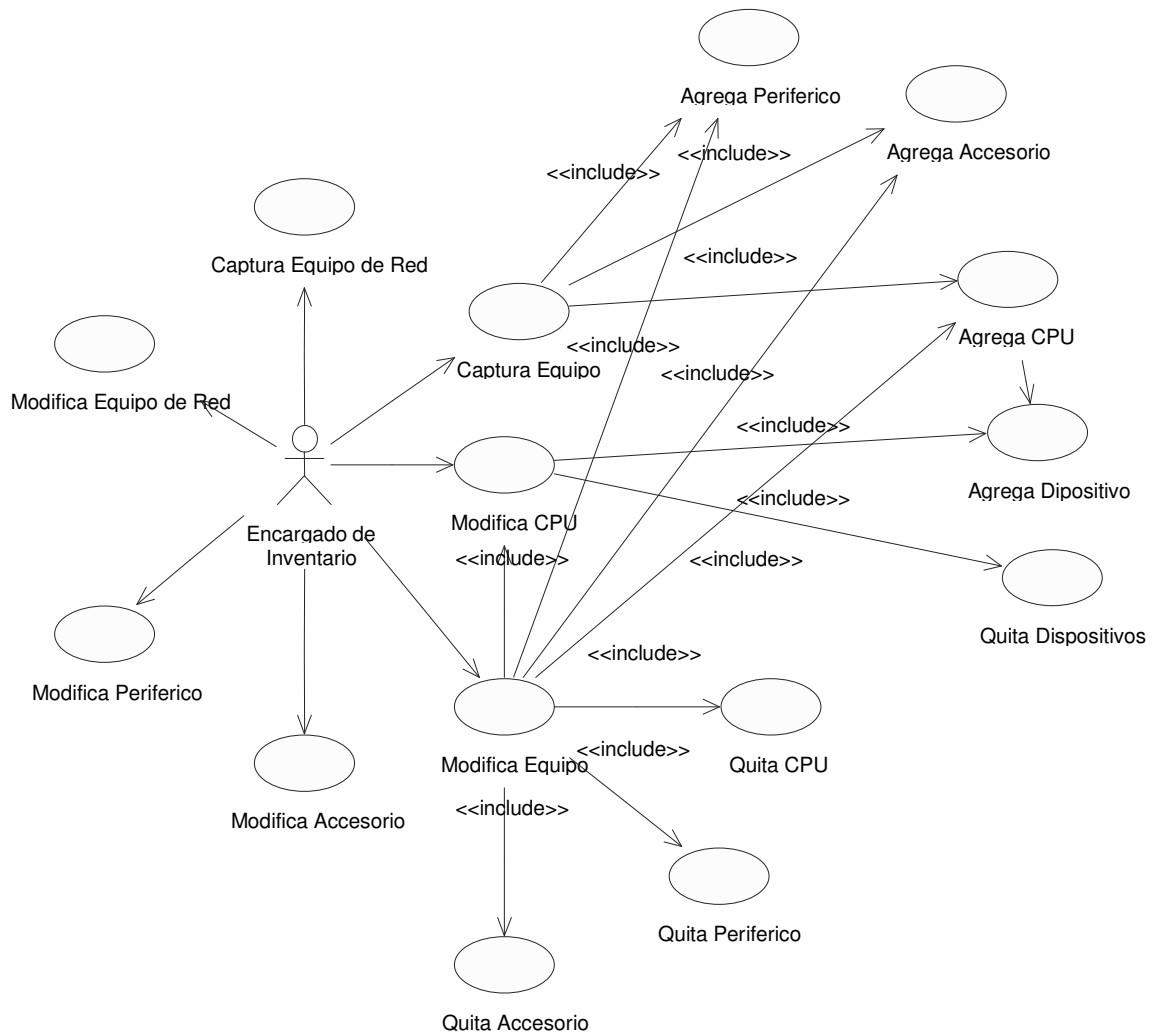


Fig. 3.19 Casos de uso del Encargado de Inventario.

Finalmente la siguiente figura nos muestra los casos de uso del actor Usuario que es la generalización de todos los actores, y los casos de uso comunes a todos los actores que son *Inicia Sesión* y *Cerrar Sesión* (fig. 3.20)

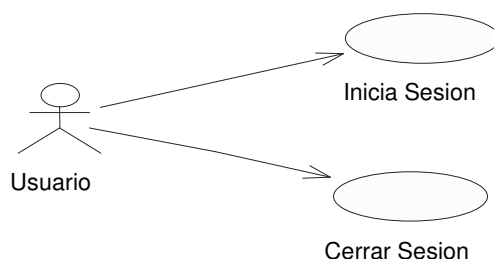


Fig. 3.20 Casos de uso del Usuario.

#### 3.4.4.3 Reestructuración del modelo de casos de uso

Esta actividad consiste en encontrar las generalizaciones y asociaciones entre los casos de uso. En figura 3.21 se muestra la forma en que cómo queda reorganizado el modelo de los casos de uso ya con las generalizaciones y asociaciones encontradas del análisis del modelo.

Los casos de uso para dar de baja las distintas entidades que se manejan se generalizan en el caso de uso *Baja*, los casos de uso que se relacionan con la actividad agregar se generalizan en el caso de uso *Agregar*, los casos de uso para la modificación se generalizan en el caso de uso *Modificar* y finalmente los casos de uso que quitan elementos se generalizan en el caso de uso *Quitar*.

Estas generalizaciones nos sirven para comprender mejor el funcionamiento del sistema, dado que al describir el comportamiento general de los casos de uso es más fácil comprender el funcionamiento particular de cada uno de ellos.

Capítulo 3. Definición de Requerimientos

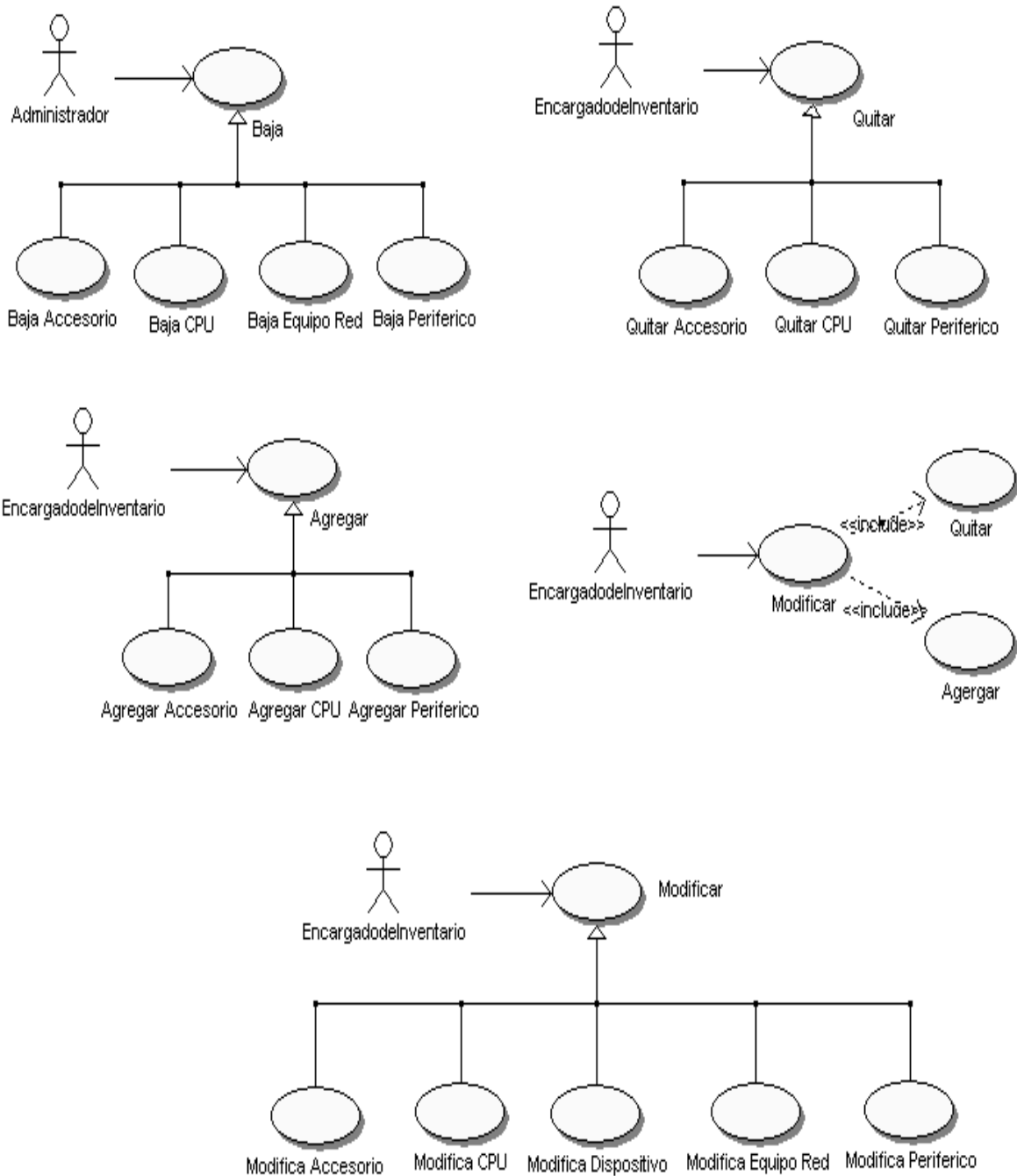


Fig. 3.21 Reestructuración del modelo de caso de uso

#### 3.4.4.4 Detallar cada caso de uso

##### Agregar

###### 1. Descripción

El caso de uso *Agregar* es usado cuando el *Encargado de Inventario* requiere agregar un elemento a cierto *Equipo* registrado en la base de datos del sistema o cuando por primera vez se registra el *Equipo*.

###### 2. Flujo de Eventos

###### 2.1 Flujo Básico

1. Cuando el *Encargado de Inventario* invoca el caso de uso *Agregar*. El sistema muestra una pantalla donde se deberá seleccionar el *Equipo* al cual se le agregará algún *elemento*.
2. El *Encargado de Inventario* elige agregar un elemento nuevo.
  - 2.1 El *Encargado de Inventario* elige agregar un elemento existente.
3. Posteriormente el sistema pedirá al *Encargado de Inventario* que comience a capturar los datos del *elemento*.
4. El *Encargado de Inventario* introduce los datos correspondientes.
5. El *Encargado de Inventario* da clic en el botón Aceptar.
  - 4.1 El *Encargado de Inventario* da clic en el botón Cancelar.
5. Si los datos son correctos el sistema crea el *elemento*.
  - 5.1 Si los datos son incorrectos el sistema no registra el componente.
6. La instancia del caso de uso finaliza.

###### 2.2 Flujos Alternativos

###### 2.2.1 En el punto 2.1

El *Encargado de Inventario* elige el elemento existente de una lista para agregarlo al *Equipo*.

###### 2.2.2 En el punto 4.1

Se borran los datos de la pantalla permitiendo que se agreguen datos de un Accesorio nuevo.

###### 2.2.3 En el punto 5.1

El sistema manda un mensaje de error pidiendo al *Encargado de Inventario* que rectifique los datos.

3. Precondiciones

3.1 Se necesita que Seleccionar un *Equipo* al cual se le agregará nuevo Accesorio.

3.2 Se requiere crear un nuevo *elemento*.

4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *elemento* se asocia a un *Equipo* Existente en el Sistema y este movimiento ha sido registrado en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Encargado de Inventario* decide salir de la aplicación.

**Agregar Dispositivo**

1. Descripción

El caso de uso *Agregar Dispositivo* es usado cuando el *Encargado de Inventario* requiere agregar un *Dispositivo* a cierto *CPU* registrado en la base de datos del sistema o cuando por primera vez se registra el *CPU*.

2. Flujo de Eventos

2.1 Flujo Básico

6. Cuando el *Encargado de Inventario* invoca el caso de uso *Agregar Dispositivo*. El sistema muestra una pantalla donde se deberá seleccionar el *CPU* al cual se le agregará algún *Dispositivo*.

7. El *Encargado de Inventario* elige agregar un elemento nuevo.

2.1 El *Encargado de Inventario* elige agregar un elemento existente.

8. Posteriormente el sistema pedirá al *Encargado de Inventario* que comience a capturar los datos del *Dispositivo*.

9. El *Encargado de Inventario* introduce los datos correspondientes.

10. El *Encargado de Inventario* da clic en el botón Aceptar.

4.1 El *Encargado de Inventario* da clic en el botón Cancelar.

5. Si los datos son correctos el sistema crea el *Dispositivo*.

5.1 Si los datos son incorrectos el sistema no registra el componente.

6. La instancia del caso de uso finaliza.

2.2 Flujos Alternativos



### Capítulo 3. Definición de Requerimientos

#### 2.2.1 En el punto 2.1

El *Encargado de Inventario* elige el elemento existente de una lista para agregarlo al *CPU*.

#### 2.2.2 En el punto 4.1

Se borran los datos de la pantalla permitiendo que se agreguen datos de un *Accesorio* nuevo.

#### 2.2.3 En el punto 5.1

El sistema manda un mensaje de error pidiendo al *Encargado de Inventario* que rectifique los datos.

### 3. Precondiciones

3.1 Se necesita que Seleccionar un *CPU* al cual se le agregará nuevo *Accesorio*.

3.2 Se requiere crear un nuevo *Dispositivo*.

### 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *Dispositivo* se asocia a un *CPU* Existente en el Sistema y este movimiento ha sido registrado en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Encargado de Inventario* decide salir de la aplicación.

## **Alta Usuario**

### 1. Descripción

El caso de uso *Alta Usuario* es usado cuando el *Administrador* requiere registrar un *Usuario* en la base de datos del sistema.

### 2. Flujo de Eventos

#### 2.1 Flujo Básico

1. Cuando el *Administrador* invoca el caso de uso *Alta Usuario*. El sistema muestra una pantalla para capturar los datos del usuario.
2. El *Administrador* introduce los datos correspondientes.
3. El *Administrador* da clic en el botón Aceptar.
  - 3.1. El *Administrador* da clic en el botón Cancelar.

### Capítulo 3. Definición de Requerimientos

4. Si los datos son correctos el sistema da de alta el usuario.
  - 4.1 Si los datos son incorrectos el sistema no registra el usuario.
5. La instancia del caso de uso finaliza.

#### 2.2 Flujos Alternativos

##### 2.2.1 En el punto 3.1

Se borran los datos de la pantalla y se permite que se agregue uno nuevo.

##### 2.2.2 En el punto 4.1

El sistema enviara un mensaje de datos incorrectos y pedirá al *Administrador* rectificar los datos.

#### 3. Precondiciones

- 3.1 El *Administrador* debe registrar un usuario nuevo.

#### 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *Usuario* ha sido registrado en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Administrador* decide salir de la aplicación.

### **Baja**

#### 1. Descripción

El caso de uso *Baja* es usado cuando el *Administrador* requiere dar de baja un Accesorio en la base de datos del sistema, esto quiere decir pasarlo a una tabla de bajas.

#### 2. Flujo de Eventos

##### 2.1 Flujo Básico

1. Cuando el *Administrador* invoca el caso de uso *Baja*. El sistema muestra una pantalla para identificar el *elemento* que va a ser dado de baja.
2. El *Administrador* selecciona de una lista el *elemento*.
3. El *Administrador* da clic en el botón Aceptar.
  - 3.1. El *Administrador* da clic en el botón Cancelar.

4. Se da de baja el *elemento*.
5. La instancia del caso de uso finaliza.

## 2.2 Flujos Alternativos

### 2.2.1 En el punto 3.1

Se regresa a la pantalla con la lista desplegable para seleccionar un nuevo *elemento*.

## 3. Precondiciones

3.1 El *Administrador* debe dar los datos de un Accesorio nuevo a dar de baja.

## 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el Accesorio ha sido dado de baja en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Administrador* decide salir de la aplicación.

## **Baja Usuario**

### 1. Descripción

El caso de uso *Baja Usuario* es usado cuando el *Administrador* requiere dar de baja un *Usuario* en la base de datos del sistema, esto quiere decir pasarlo a una tabla de bajas.

### 2. Flujo de Eventos

#### 2.1 Flujo Básico

1. Cuando el *Administrador* invoca el caso de uso *Baja Usuario*. El sistema muestra una pantalla para identificar el *Usuario* que va a ser dado de baja.
2. El *Administrador* introduce los datos correspondientes.
3. El *Administrador* da clic en el botón Aceptar.
  - 3.1. El *Administrador* da clic en el botón Cancelar.
4. Si los datos son correctos el sistema da de baja el *Usuario*.
  - 4.1 Si los datos son incorrectos el sistema no da de baja el componente.
5. La instancia del caso de uso finaliza.

2.2 Flujos Alternativos

2.2.1 En el punto 3.1

Se borran los datos de la pantalla y se permite que se tomen los datos de otro *Usuario* para dar de baja.

2.2.2 En el punto 4.1

El sistema enviara un mensaje de datos incorrectos y pedirá al *Encargado de Inventario* rectificar los datos.

3. Precondiciones

3.1 El *Administrador* debe dar los datos de un *Usuario* nuevo a dar de baja.

4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *Usuario* ha sido dado de baja en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Administrador* decide salir de la aplicación.

## Captura

1. Descripción

El caso de uso *Captura* es usado cuando el *Encargado de Inventario* requiere registrar un *elemento* en la base de datos del sistema.

2. Flujo de Eventos

2.3 Flujo Básico

1. Cuando el *Encargado de Inventario* invoca el caso de uso *Captura*. El sistema muestra una pantalla para capturar los datos del *elemento*.
2. El *Encargado de Inventario* introduce los datos correspondientes.
3. El *Encargado de Inventario* da clic en el botón Aceptar.
  - 3.1. El *Encargado de Inventario* da clic en el botón Cancelar.
4. Si los datos son correctos el sistema da de alta el elemento.
  - 4.1 Si los datos son incorrectos el sistema no registra el *elemento*.
5. La instancia del caso de uso finaliza.

## 2.4 Flujos Alternativos

### 2.2.1 En el punto 3.1

Se borran los datos de la pantalla y se permite que se agregue un nuevo elemento.

### 2.2.2 En el punto 4.1

El sistema enviara un mensaje de datos incorrectos y pedirá al *Encargado de Inventario* rectificar los datos.

## 3. Precondiciones

3.1 El *Encargado de Inventario* debe registrar un elemento nuevo o alguno que no este registrado en el sistema por lo cual invoca al caso de uso Captura.

## 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *elemento* ha sido registrado en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Encargado de Inventario* decide salir de la aplicación.

## **Captura Equipo**

### 1. Descripción

El caso de uso *Captura Equipo* se realiza cuando el *Encargado de Inventario* requiere registrar un *Equipo* en la base de datos del sistema.

### 2. Flujo de Eventos

#### 2.1 Flujo Básico

1. Cuando el *Encargado de Inventario* invoca el caso de uso *Captura Equipo*. El sistema muestra una pantalla para capturar los datos del *Equipo*.
2. El *Encargado de Inventario* introduce los datos correspondientes.
3. El *Encargado de Inventario* da clic en el botón Aceptar.
  - 3.1. El *Encargado de Inventario* da clic en el botón Cancelar.
4. Si los datos son correctos el sistema da de alta el *Equipo*.
  - 4.1 Si los datos son incorrectos el sistema no registra el *Equipo*.

5. La instancia del caso de uso finaliza.

2.2 Flujos Alternativos

2.2.1 En el punto 3.1

Se borran los datos de la pantalla y se permite que se agregue uno nuevo.

2.2.2 En el punto 4.1

El sistema enviara un mensaje de datos incorrectos y pedirá al Encargado de Inventario rectificar los datos.

3. Precondiciones

3.1 El *Encargado de Inventario* debe registrar un *Equipo* nuevo por lo cual invoca al caso de uso *Captura Equipo*.

4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *Equipo* ha sido registrado en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Encargado de Inventario* decide salir de la aplicación.

**Cerrar Sesión**

1. Descripción

El caso de uso *Cerrar Sesión* se realiza cuando el *Usuario* desea cerrar la sesión del sistema.

2. Flujo de Eventos

2.1 Flujo Básico

1. Cuando el *Usuario* invoca el caso de uso *Cerrar Sesión*. El sistema muestra un mensaje de confirmación del cierre de la sesión.

2. El *Usuario* da clic en el botón Aceptar del mensaje de confirmación del cierre de la sesión.

2.1. El *Usuario* da clic en el botón Cancelar del mensaje de confirmación.

3. La instancia del caso de uso finaliza.

## 2.2 Flujos Alternativos

### 2.2.1 En el punto 2.1

El *Usuario* regresa a la pantalla en la que se encontraba hasta antes del cierre de la sesión.

## 3. Precondiciones

3.1 El *Usuario* desea cerrar la sesión.

## 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *Usuario* decide salir de la aplicación.

4.2 La instancia del caso de uso finaliza cuando se realiza el cierre de la sesión.

## **Consultar Bitácora**

### 1. Descripción

El caso de uso *Consultar Bitácora* es usado cuando el *Administrador* quiere Consultar las modificaciones que se han realizado en la base de datos del sistema. Esta acción puede realizarse mediante un rango de fecha o si se desea saber las modificaciones que se realizaron en un día específico.

### 2. Flujo de Eventos

#### 2.1 Flujo Básico

1. Cuando el *Administrador* invoca el caso de uso *Consultar Bitácora*. El sistema muestra una pantalla donde se deberá seleccionar como quiere que se presente la consulta de las modificaciones que se han realizado, ya sea por rango de fecha o indicando el día que se desea consultar.
2. Ya seleccionada la forma en que se desea que se presenten las modificaciones, el sistema muestra una pantalla para mostrar las modificaciones que se han realizado en ese periodo de tiempo.
3. El *Administrador* da clic en el botón Aceptar que regresara a la página principal.
  - 3.1 El *Administrador* da clic en el botón Cancelar.
4. La instancia del caso de uso finaliza.

#### 2.2 Flujos Alternativos

2.2.1 En el punto 3.1

Se borran los datos de la pantalla y se permite seleccionar otra forma de ver las modificaciones.

3. Precondiciones

3.1 El Sistema debe estar creado y deben existir modificaciones.

3.2 El Sistema debe tener registrada cada modificación considerando todas las fechas.

4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando se han desplegado las modificaciones requeridas de la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Administrador* decide salir de la aplicación.

**Generar Censo**

1. Descripción

El caso de uso *Generar Censo* es usado cuando el *Administrador* quiere Generar los Censos solicitados por la DGSCA.

2. Flujo de Eventos

2.1 Flujo Básico

1. Cuando el Administrador invoca el caso de uso *Generar Censo*. El sistema muestra una pantalla donde se deberá seleccionar el tipo de censo que se desea realizar.
2. Ya seleccionada la forma en que se desea realizar el censo, el sistema muestra una pantalla para mostrar los datos solicitados.
3. El *Administrador* da clic en el botón Aceptar que regresara a la página principal.

3.1 El *Administrador* da clic en el botón Cancelar.

4. La instancia del caso de uso finaliza.

2.2 Flujos Alternativos

2.2.1 En el punto 3.1



Se borran los datos de la pantalla y se permite volver a generar otro censo.

### 3. Precondiciones

3.1 Al menos debe haber un dato guardado en la base de datos para la generación del censo.

### 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando se ha desplegado el censo solicitado.

4.2 La instancia del caso de uso finaliza cuando el *Administrador* decide salir de la aplicación.

## Inicia Sesión

### 1. Descripción

El caso de uso *Inicia Sesión* se realiza cuando el *Usuario* entrar al sistema.

### 2. Flujo de Eventos

#### 2.1 Flujo Básico

1. Para el caso de uso inicia sesión no es necesario que el *Usuario* invoque ningún caso de uso, ya que es la pauta para entrar en el sistema. El sistema muestra una pantalla para capturar los datos del *Usuario* que son requeridos para poder entrar en el sistema.
2. El *Usuario* introduce los datos correspondientes.
3. El *Usuario* da clic en el botón Aceptar.
  - 3.1. El *Usuario* da clic en el botón Cancelar.
4. Si los datos son correctos el *Usuario* puede entrar en el sistema.
  - 4.1 Si los datos son incorrectos el *Usuario* no puede entrar en el sistema.
5. La instancia del caso de uso finaliza.

#### 2.2 Flujos Alternativos

##### 2.2.1 En el punto 3.1

El sistema manda un mensaje de error de que los datos ingresados son incorrectos.

3. Precondiciones

3.1 El *Usuario* desea iniciar sesión.

4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *Usuario* ha podido entrar en el sistema.

**Modificar**

1. Descripción

Este caso de uso *Modifica* es utilizado por el *Encargado de Inventario* para modificar los datos de un *elemento* específico en la base de datos del sistema.

2. Flujo de Eventos

2.1 Flujo Básico

1. Cuando el *Encargado de Inventario* invoca el caso de uso *Modifica*. El sistema muestra una pantalla para realizar la modificación de los datos del *elemento*.
2. El sistema muestra una pantalla en dónde se especifica el *elemento* que se desea modificar y el *Encargado de Inventario* realiza las modificaciones correspondientes del *elemento* específico.
- .3. El *Encargado de Inventario* da clic en el botón Aceptar.
  - 3.1. El *Encargado de Inventario* da clic en el botón Cancelar.
4. Si los datos son correctos el sistema realiza la modificación del *elemento*.
  - 4.1 Si los datos son incorrectos el sistema no realiza ninguna modificación.
5. La instancia del caso de uso finaliza.

2.2 Flujos Alternativos

2.2.1 En el punto 3.1

Se borran los datos de la pantalla y se permite que se modifique uno nuevo.

2.2.2 En el punto 4.1

El sistema enviara un mensaje de datos incorrectos y pedirá al *Encargado de Inventario* rectificar los datos.

3. Precondiciones

3.1 El *Encargado de Inventario* desea modificar un *elemento*.

#### 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *elemento* ha sido modificado en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Encargado de Inventario* decide salir de la aplicación.

### **Modificar Equipo**

#### 1. Descripción

Este caso de uso *Modificar Equipo* es utilizado por el *Encargado de Inventario* para modificar los datos de un *Equipo* específico en la base de datos del sistema.

#### 2. Flujo de Eventos

##### 2.1 Flujo Básico

1. Cuando el *Encargado de Inventario* invoca el caso de uso *Modificar Equipo*. El sistema muestra una pantalla para realizar la modificación de los datos del *Equipo*.
2. El sistema muestra una pantalla en dónde se especifica el *Equipo* que se desea modificar y el *Encargado de Inventario* realiza las modificaciones correspondientes del *Equipo* mediante los casos de uso *Agregar*, *Quitar* y *Modificar CPU*.
3. El *Encargado de Inventario* da clic en el botón Aceptar.
  - 3.1. El *Encargado de Inventario* da clic en el botón Cancelar.
4. Si los datos son correctos el sistema realiza la modificación del *Equipo*.
  - 4.1 Si los datos son incorrectos el sistema no realiza ninguna modificación.
5. La instancia del caso de uso finaliza.

##### 2.2 Flujos Alternativos

###### 2.2.1 En el punto 3.1

Se borran los datos de la pantalla y pide que se modifique uno nuevo.

###### 2.2.2 En el punto 4.1

### Capítulo 3. Definición de Requerimientos

El sistema enviara un mensaje de datos incorrectos y pedirá al *Encargado de Inventario* rectificar los datos.

#### 3. Precondiciones

3.1 El *Encargado de Inventario* desea modificar un *Equipo*.

#### 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *Equipo* ha sido modificado en la base de datos.

4.2 La instancia del caso de uso finaliza cuando el *Encargado de Inventario* decide salir de la aplicación.

### **Quitar**

#### 1. Descripción

Este caso de uso *Quitar* es utilizado por el *Encargado de Inventario* para dar de baja *elemento* del *Equipo* al que pertenecen, ya sea que se van a cambiar de ubicación o simplemente no pertenecerán al *Equipo*.

#### 2. Flujo de Eventos

##### 2.1 Flujo Básico

1. Cuando el *Encargado de Inventario* invoca el caso de uso *Quitar*, el sistema muestra una pantalla para realizar la modificación de los datos del *elemento*.
2. El sistema muestra una pantalla en dónde se especifica el *elemento* que se desea modificar y el *Encargado de Inventario* realiza las modificaciones correspondientes del *elemento* específico.
3. El *Encargado de Inventario* da clic en el botón Aceptar.
  - 3.1. El *Encargado de Inventario* da clic en el botón Cancelar.
4. Si los datos son correctos el sistema realiza la modificación del *elemento*.
  - 4.1 Si los datos son incorrectos el sistema no realiza ninguna modificación.
5. La instancia del caso de uso finaliza.

##### 2.2 Flujos Alternativos

###### 2.2.1 En el punto 3.1

Se borran los datos de la pantalla y pide que se modifique uno nuevo.

2.2.2 En el punto 4.1

El sistema enviara un mensaje de datos incorrectos y pedirá al *Encargado de Inventario* rectificar los datos.

3. Precondiciones

3.1 El *Encargado de Inventario* desea quitar un *elemento* de un *Equipo*.

4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando el *elemento* ha sido quitado del *Equipo* al que pertenece.

4.2 La instancia del caso de uso finaliza cuando el Encargado de Inventario decide salir de la aplicación.

### **Revisar Solicitudes**

1. Descripción

Este caso de uso *Revisar Solicitudes* es utilizado por el *Administrador* para hacer la revisión de las *Solicitudes*, como baja, compra o mantenimiento.

2. Flujo de Eventos

2.1 Flujo Básico

1. Cuando el *Administrador* invoca el caso de uso *Revisar Solicitudes*, el sistema muestra una pantalla en dónde viene un listado de la forma de consultar las solicitudes.
2. El *Administrador* selecciona las solicitudes pendientes.
  - 2.1 El *Administrador* selecciona las solicitudes atendidas.
3. El sistema despliega una pantalla en dónde se muestran las solicitudes pendientes.
4. La instancia del caso de uso finaliza.

2.2 Flujos Alternativos

2.2.1 En el punto 2.1

El sistema despliega una pantalla en dónde se muestran las solicitudes atendidas.

3. Precondiciones

3.1 El *Administrador* desea revisar las *Solicitudes*.

#### 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando se han revisado las *Solicitudes*.

4.2 La instancia del caso de uso finaliza cuando el *Administrador* decide salir de la aplicación.

### **Solicitar**

#### 1. Descripción

El caso de Uso *Solicitar* es utilizado por el *Jefe de Área* para realizar las peticiones sobre compra, mantenimiento o baja para determinado *Equipo*, el caso de uso *Solicitar* imprimirá el formato correspondiente.

#### 2. Flujo de Eventos

##### 2.1 Flujo Básico

1. Cuando el *Jefe de Área* invoca el caso de uso *Solicitar*, el sistema muestra una pantalla donde introducirán los datos del formato deseado.
2. El sistema pedirla al *Jefe de Área* los datos necesarios.
3. Una vez que el formulario para de solicitud este completo. El *Jefe de Área* dará clic en Aceptar.
  - 3.1 El *Jefe de Área* da clic en Cancelar
4. Si los datos son correctos se procesa la solicitud para imprimir.
  - 4.1 Si los datos son incorrectos el sistema pide al *Jefe de Área* *introducir nuevamente los datos*
5. El Sistema Imprime el formato para la *Solicitud*.
6. La instancia del caso de uso finaliza.

##### 2.2 Flujos Alternativos

###### 2.2.1 En el punto 3.1.

Se borran los datos de la pantalla y da la opción de hacer una nueva *Solicitud*.

#### 3. Precondiciones

### *Capítulo 3. Definición de Requerimientos*

3.1 Debe existir alguna necesidad que requiera generar *Solicitudes*.

3.2 Debe no debe existir un formato de *Solicitud* previo.

3.2 El *Jefe de Área* tiene que notificar que realizará una *Solicitud* por lo que invoca al caso de uso.

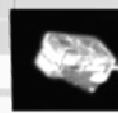
#### 4. Poscondiciones

4.1 La instancia del caso de uso finaliza cuando la *Solicitud* ha sido registrada en la base de datos y se tiene impreso el formato de *Solicitud*.

4.2 La instancia del caso de uso finaliza cuando el *Jefe de Área* decide salir de la aplicación.



Sistema de  
inventario



# *Capítulo 4*

## *Análisis y Diseño*

---



## 4.1 Modelo de análisis

El modelo de análisis es un modelo conceptual que se basa en objetos; este modelo utiliza el lenguaje del desarrollador para presentar una vista interna del sistema, como lo son: la estructura de clases y paquetes estereotipados (fig.4.1).

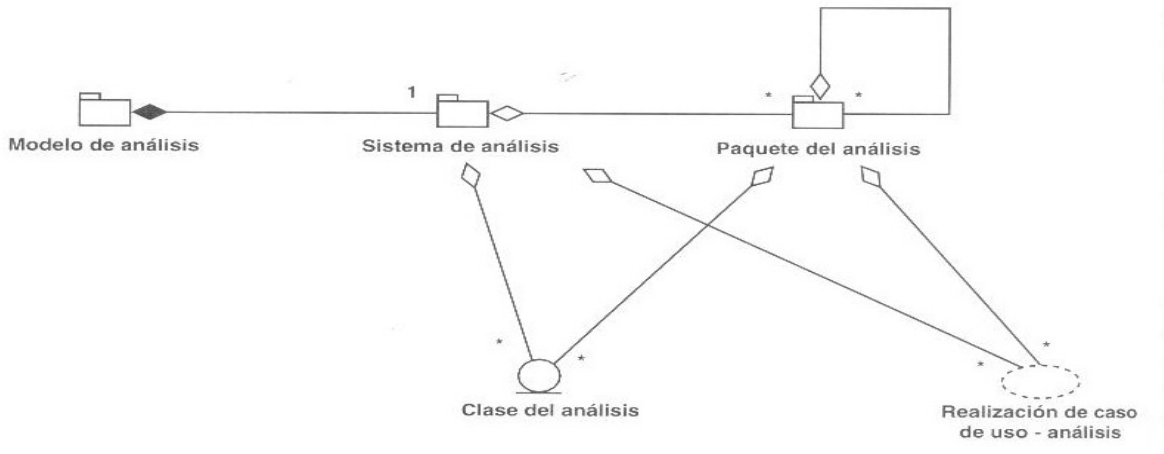


Fig. 4.1. Esquema del modelo de análisis.

## 4.2 Análisis de la Arquitectura

Consiste en identificar los paquetes del análisis, las clases del análisis, que ayudaran a tener una mejor descripción de la arquitectura.

### 4.2.1 Paquete de análisis

Es una estructura que permite la organización del modelo de análisis en piezas simples para su fácil manipulación. Estos paquetes contienen clases de análisis, realizaciones de casos de uso y otros paquetes del análisis.

El paquete de análisis (fig. 4.2) está caracterizado por:

- Representar un análisis de manera separada.
- Se crea en base a los requisitos funcionales y en el dominio del problema.
- Convertirse posiblemente en subsistemas del modelo del diseño.

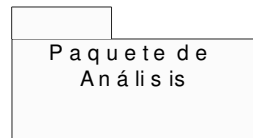


Fig. 4.2 Paquete del análisis.

#### 4.2.2 Identificar paquetes del análisis

Es la actividad que consiste en asociar las funcionalidades que brindan los casos de uso de manera conjunta para un determinado proceso del negocio, para dar soporte a un determinado actor del sistema o que se relaciona mediante generalizaciones y extensiones de uso.

#### 4.2.3 Identificar Paquetes del Análisis para el Sistema de Inventario

Asociando los procesos del negocio a los casos de uso se pueden identificar los siguientes paquetes de análisis para el sistema de inventario de la SSA.

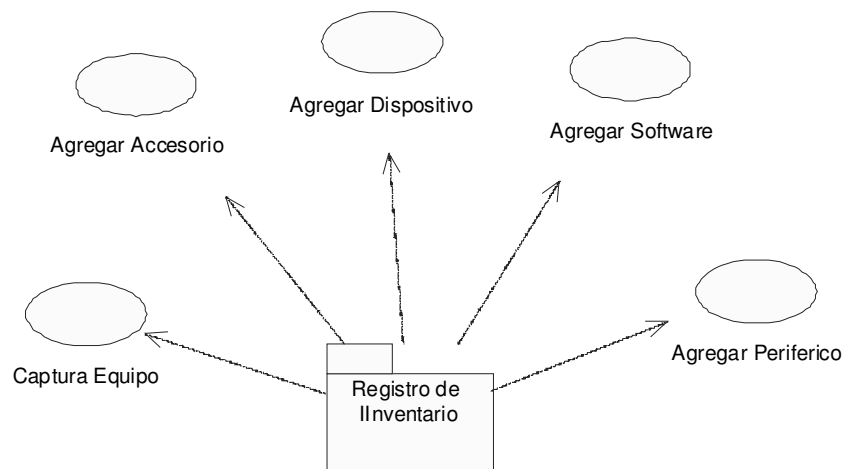


Fig. 4.3 Paquete de análisis Registro de Equipo.

El paquete de análisis Registro de Inventario encapsula los servicios que presta el sistema para capturar todos los datos relacionados con los equipos, desde su descripción hardware hasta el software que tiene instalado (fig.4.3).

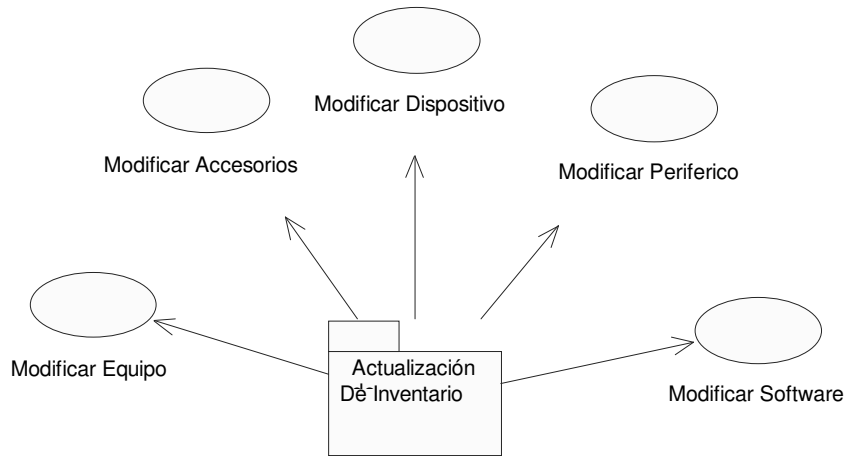


Figura 4.4 Paquete de análisis Actualización de Inventario.

El paquete de análisis de Actualización de Inventario representa a todos los procesos del negocio enfocados a realizar sobre el inventario de los equipos y sobre todos los elementos relacionados con estos (fig.4.4).

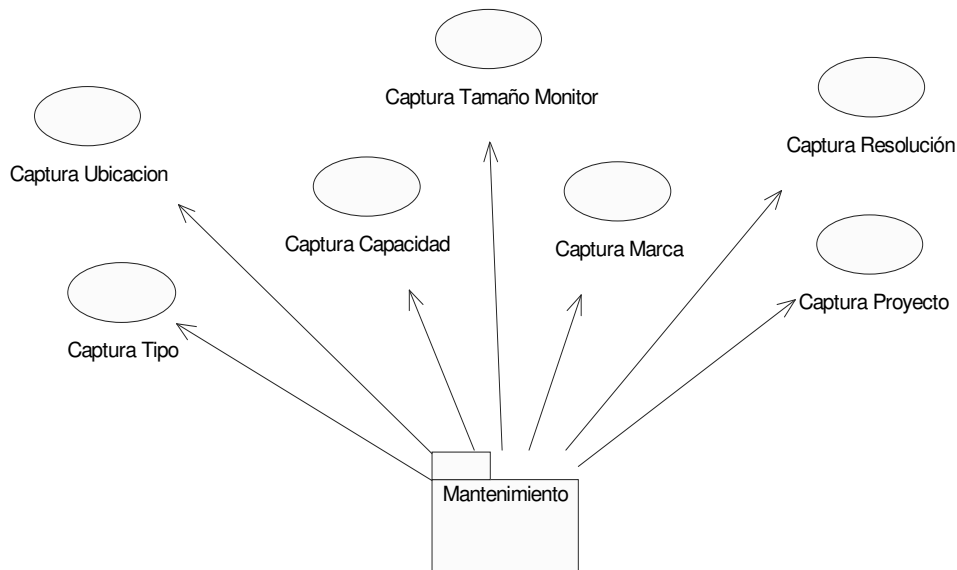


Figura 4.5 Paquete de análisis Mantenimiento.

El paquete Mantenimiento agrupa los casos de uso que son utilizados para realizar actividades de mantenimiento en el sistema, actualizando datos que facilitan la captura de los datos de los equipos (fig.4.5).

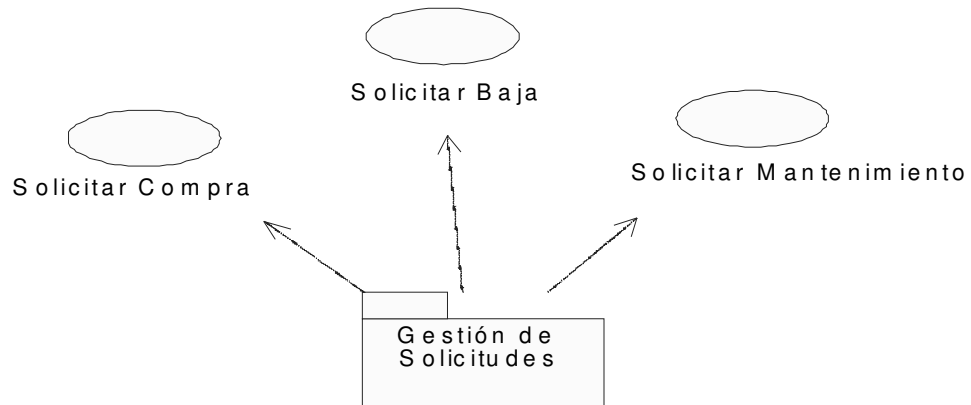


Figura 4.6 Paquete Gestión de Solicitudes.

El paquete Gestión de Solicitudes engloba los procesos relacionados con peticiones que solo el Jefe de Área puede hacer, dado que son tareas relacionadas con procesos administrativos (fig.4.6).

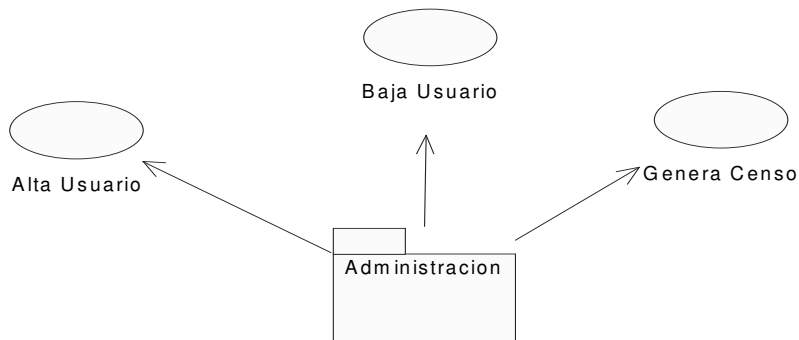


Figura 4.7 Paquete Administración.

El paquete Administración recoge todas las actividades exclusivas para el administrador del sistema (fig.4.7).

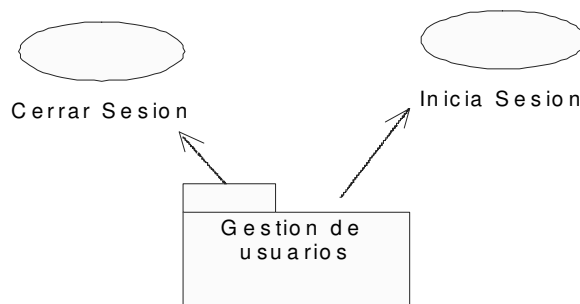


Figura 4.8 Paquete Gestión de usuarios.

El paquete Gestión de usuarios se encarga de los procesos del sistema relacionado con las sesiones que deben iniciar los distintos tipos de usuarios del sistema, así como administrar sus salidas (fig.4.8).

### 4.3 Clases de Análisis

Es una abstracción de una o varias clases y /o subsistemas de lo que será posteriormente el diseño del sistema y consta centrándose principalmente en los requisitos funcionales del sistema.

Todas las clases del análisis deben corresponder a uno de los tres estereotipos de Clases del análisis.

- Clases de Interfaz: Modela los límites del sistema, representados por la forma de comunicación entre los actores y el sistema o sistemas externos y el sistema (fig.4.9).

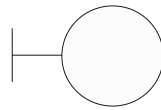


Fig. 4.9 UML para una Clase de Interfaz.

- Clase de Entidad: Modelan la información que ha de persistir en el tiempo: Las clases de entidad muestran datos que contribuyen a comprender la información del sistema (fig.4.10).

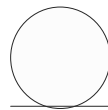


Fig. 4.10 UML para una clase de Entidad.

- Clases de control: Se usan para encapsular el control de un caso de uso: Modelan procesos como coordinación, secuencia, transacciones y control de otros objetos (fig.4.11).

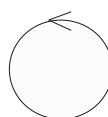


Fig. 4.11 UML para una clase de Control.

#### 4.4 Analizar Casos de Uso

Esta actividad consiste en:

- Identificar las clases del análisis.
- Distribuir los comportamientos del caso de uso.
- Capturar requisitos especiales.

El análisis de casos de uso genera los siguientes artefactos del análisis:

- Diagramas de Realización (Diagramas de clases): Se usa para representar las clases que intervienen en la realización de un Caso de Uso.
- Diagramas de Colaboración (Diagramas de Interacción): Se usa para representar la comunicación que existe entre los objetos que intervienen en el caso de uso.
- Flujo de Sucesos-Análisis: Es la descripción textual de los diagramas de colaboración usada para dar mayor facilidad de comprensión de estos diagramas.

#### 4.5 Analizar Casos de Uso del Sistema de Inventario

##### **Caso de uso Agregar** (fig.4.12)

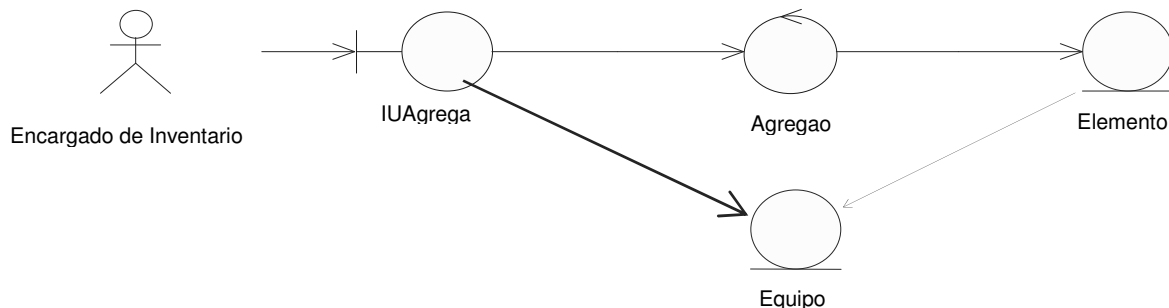


Figura 4.12 Diagrama de clases participantes para el caso de uso Agregar Accesorio.

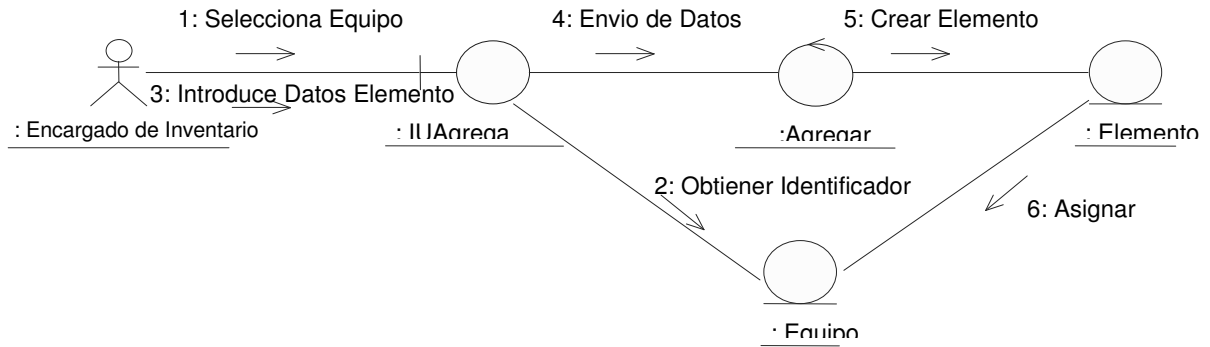


Figura 4.13 Diagrama de colaboración de clases del caso de uso Agregar Accesorio.

**Flujo de sucesos-análisis del Caso de Uso Agregar Accesorio** (fig. 4.13)

El *Encargado de Inventario* selecciona el equipo al cuál le va a agregar un *Elemento* a través de *IUAgregarElemento* (1). De dónde obtiene el identificador del *Equipo* (2). Introduce los datos del *Elemento* que va a ser agregado (3). Se da el envío de los datos de *Elemento* a la base de datos (4). Y se crea el *Elemento* (5). Finalmente se asigna el *Elemento* al *Equipo* que fue seleccionado (6).

**Caso de Uso Agregar Dispositivo** (fig.4.14)

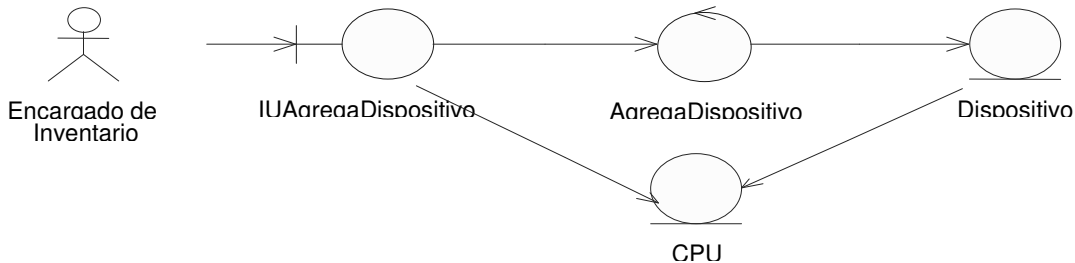


Figura 4.14 Diagrama de clases participantes del caso de uso Agregar Dispositivo.

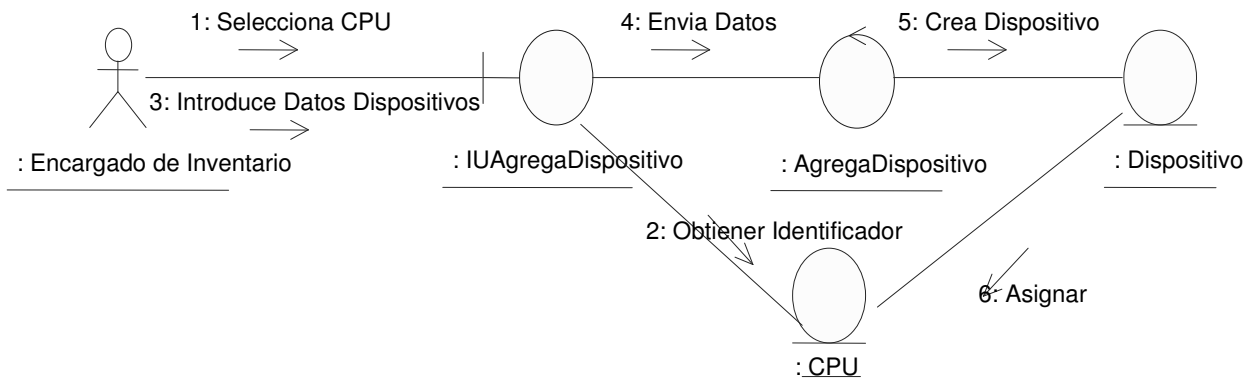


Figura 4.15 Diagrama de colaboración de clases del caso de uso Agregar Dispositivo.

**Flujo de sucesos-análisis del Caso de Uso Agregar Dispositivo** (fig. 4.15)

El *Encargado de Inventario* selecciona el *CPU* al cuál le va a agregar un dispositivo a través de *IUAgregarDispositivo* (1). De dónde obtiene el identificador del *CPU* (2). Introduce los datos del *Dispositivo* que va a ser agregado (3). Se da el envío de los datos de *Dispositivo* a la base de datos (4). Y se crea el *Dispositivo* (5). Finalmente se asigna el dispositivo al *CPU* que fue seleccionado (6).

**Caso de Uso Alta Usuario** (fig.4.16)

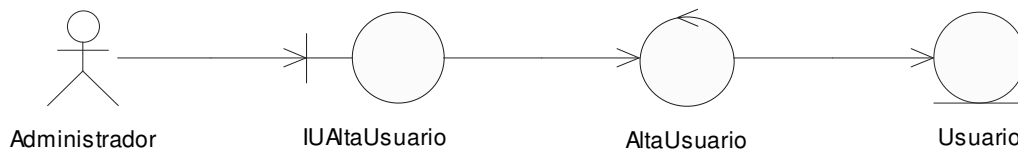


Figura 4.16 Diagrama de clases participantes del caso de uso Alta Usuario.

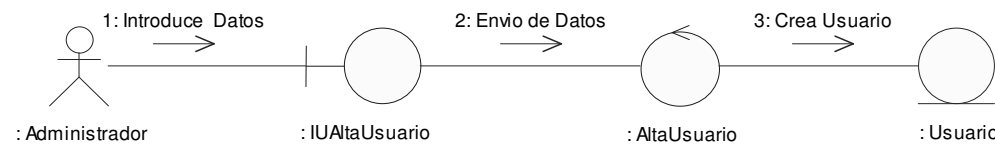


Figura 4.17 Diagrama de colaboración de clases del caso de uso Alta Usuario.

**Flujo de sucesos-análisis del caso de Uso Alta Usuario** (fig. 4.17)

El *Administrador* introduce los datos del nuevo *Usuario* que desea de dar de alta por medio de *IUAltaUsuario* (1). El gestor *AltaUsuario* valida y envía los datos a la base de datos (2). Cuando los datos ingresados son correctos es creado un nuevo *Usuario* (3).



**Caso de uso Baja** (fig.4.18)

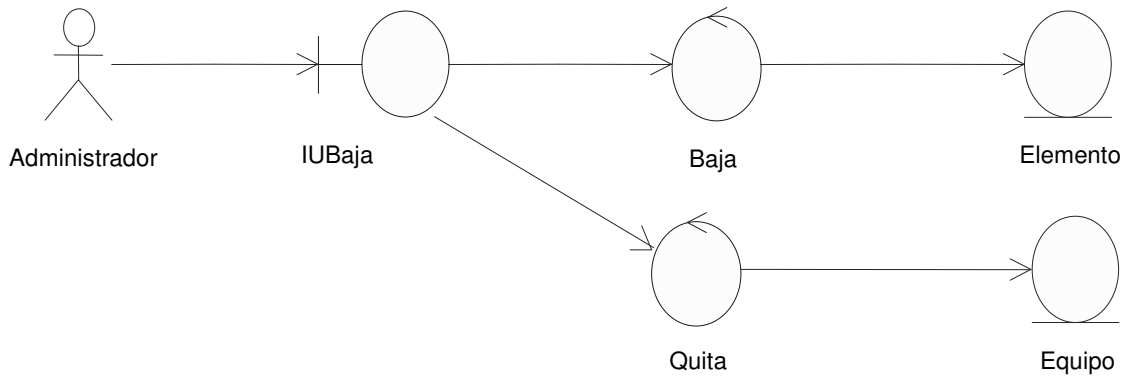


Figura 4.18 Diagrama de clases participantes del caso de uso Baja Accesorio.

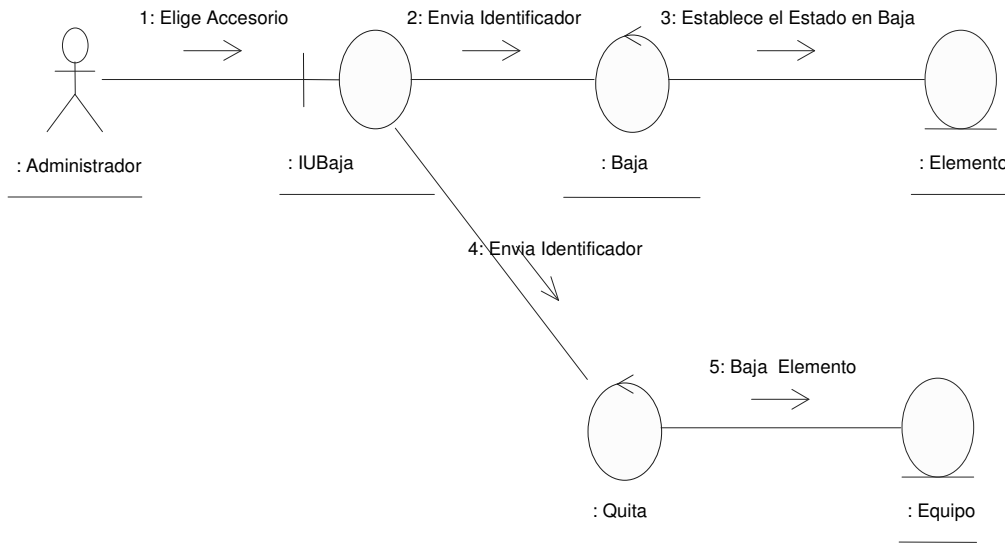


Figura 4.19 Diagrama de colaboración de clases del caso de uso Baja Accesorio.

**Flujo de sucesos-análisis del caso de Uso Baja** (fig. 4.19)

El *Administrador* elige el *Elemento* que desea dar de baja (1). Envía el identificador a la clase de control *Baja* que es el encargado de establecer el estado de baja (2, 3), que a su vez envía el identificador a *Quita*, para dar de baja el *Elemento* del *Equipo* al que pertenece (4,5).

**Caso de Uso Baja Usuario** (fig.4.20)

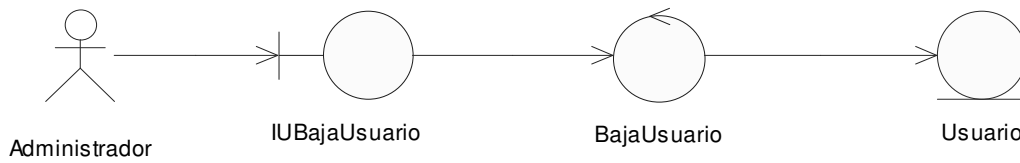


Figura 4.20 Diagrama Clases Participantes del caso de uso Baja Usuario.

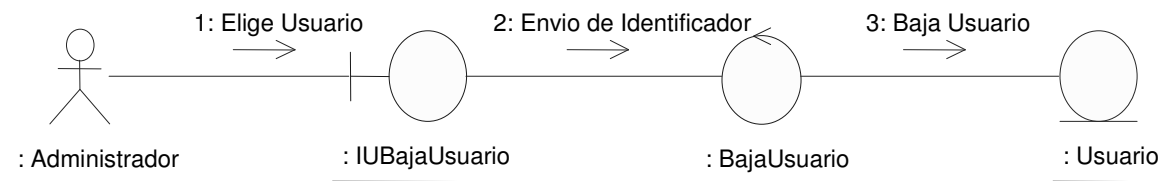


Figura 4.21 Diagrama de colaboración de clases del caso de uso Baja Usuario.

**Flujo de sucesos-análisis del caso de Uso Baja Usuario** (fig. 4.21)

El *Administrador* selecciona el *Usuario* que desea de dar de baja por medio de *IUBajaUsuario* (1). La clase *BajaUsuario* valida la selección y envía el identificar de *Usuario* (2). Se da de baja el *Usuario* (3).

**Caso de uso Captura** (fig.4.22)

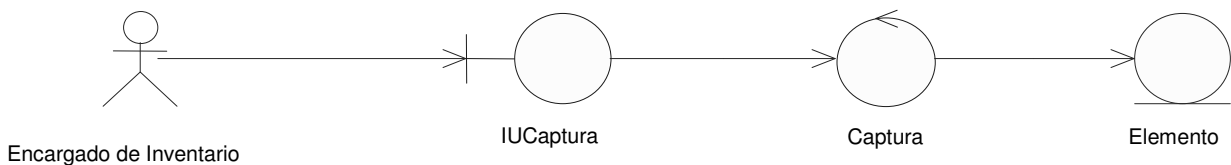


Figura 4.22 Diagrama de Clases Participantes del caso de uso Captura Accesorio.

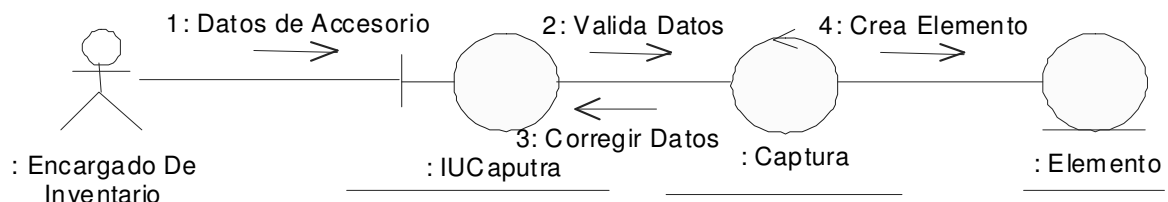


Figura 4.23 Diagrama de colaboración de clases del caso de uso Captura Accesorio.

**Flujo de sucesos-análisis del caso de Uso Captura** (fig. 4.23)

El *Encargado de Inventario* a través de *IUCaptura* introduce los datos del *Elemento* que se desea capturar (1). La clase *Captura*, valida los datos ingresados (2). Si los datos son incorrectos la clase *Captura* pide corregir los datos (3). Si los datos son correctos se crea un nuevo *Elemento* (4).

**Caso de Uso Captura Equipo** (fig.4.24)

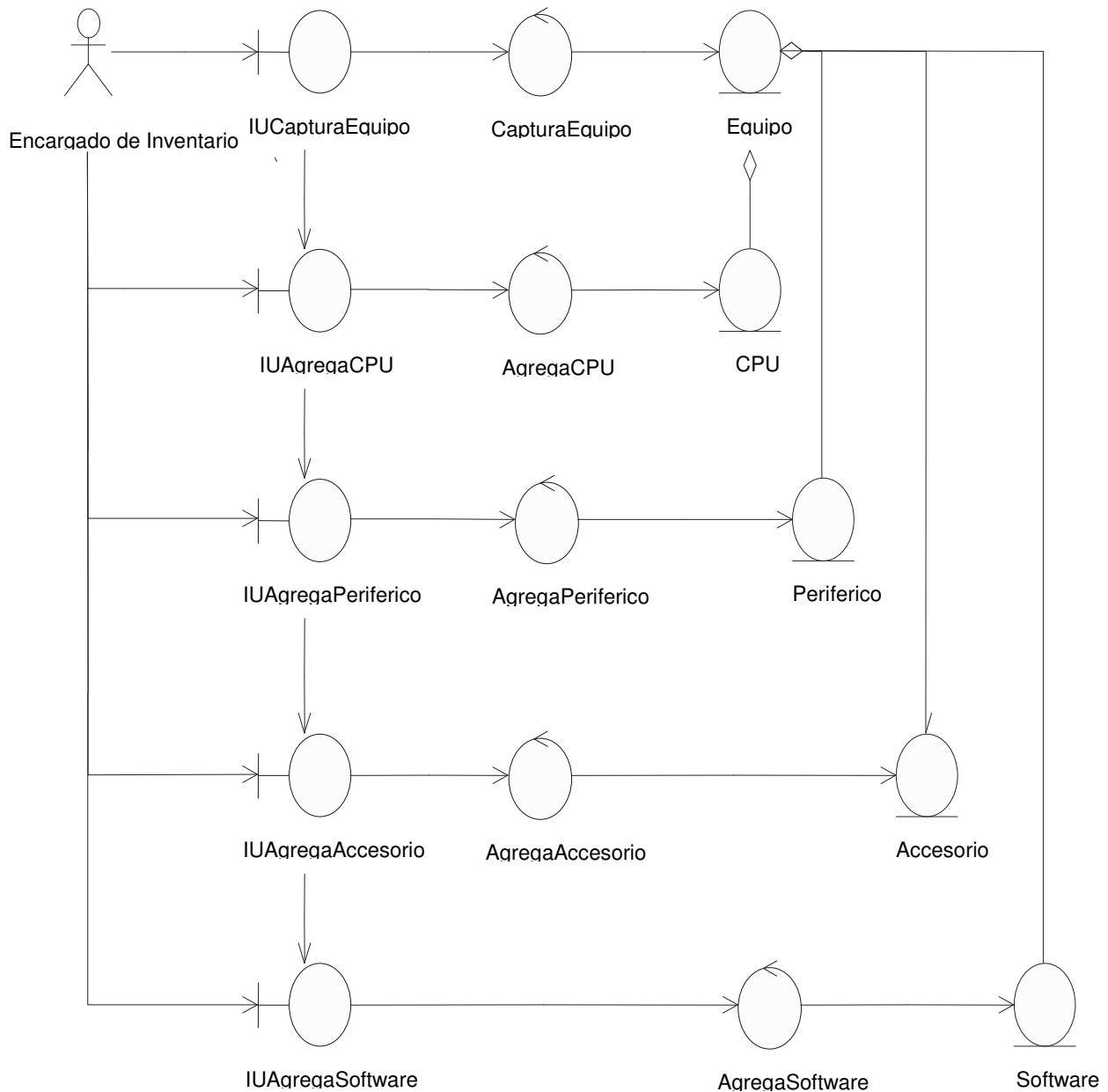


Figura 4.24 Diagrama de clases participantes del caso de uso Captura Equipo.

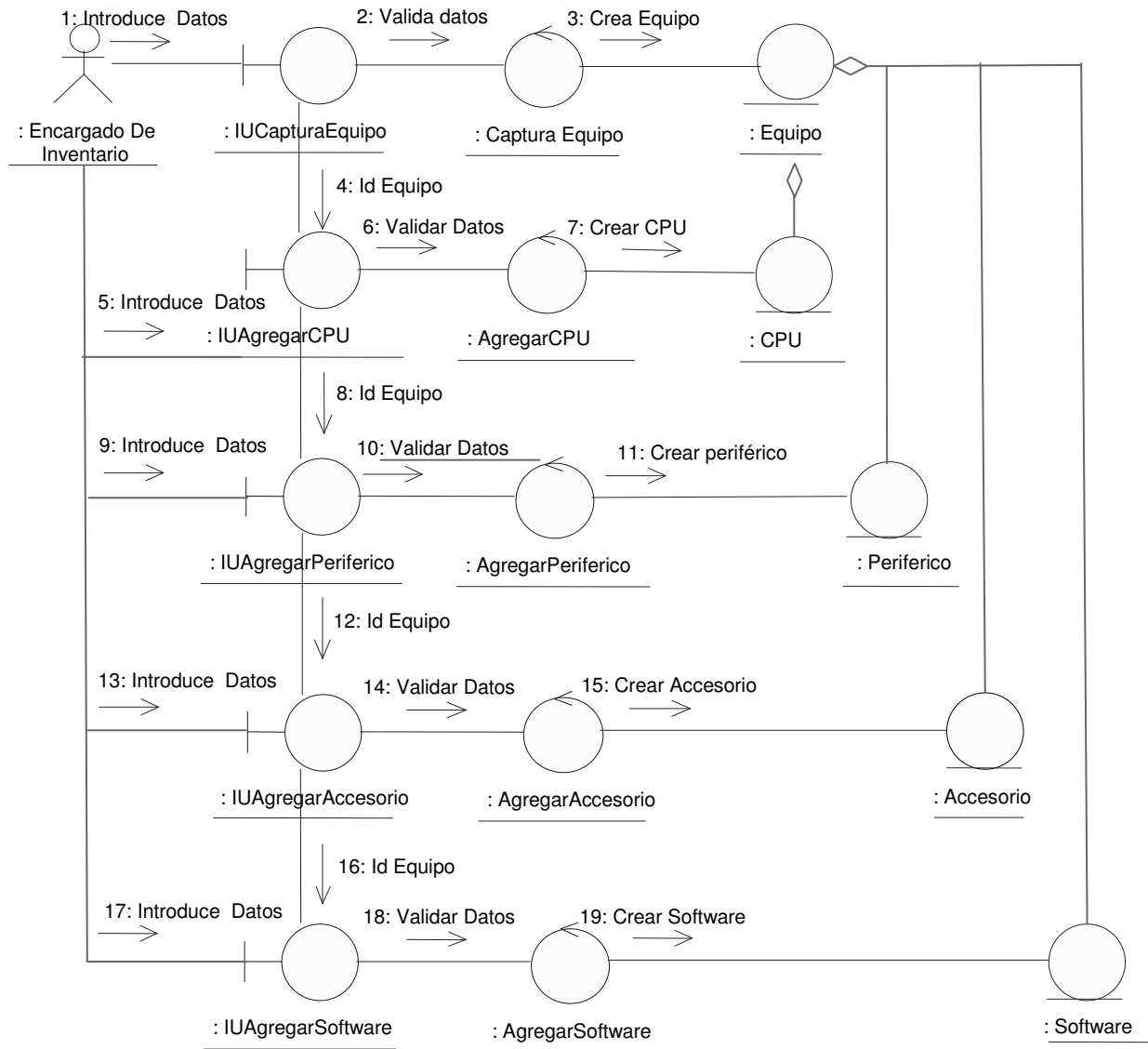


Figura 4.25 Diagrama de colaboración de clases del caso de uso Captura Equipo.

**Flujo de sucesos-análisis del caso de Uso Captura Equipo, Agregar Accesorio, Agregar Dispositivo, Agregar Software, Agregar Periférico** (fig. 4.25)

El *Encargado de Inventario* por medio de *IUCapturaEquipo*, introduce los datos del *Equipo* que desea capturar (1). El gestor *CapturaEquipo*, valida los datos ingresados (2). Si los datos son correctos se crea un nuevo *Equipo* (3). Se envía el *Id de Equipo* (4). A través de la interfaz *IUAgregarCPU* se capturan los datos de *CPU* (5), se validan los datos del *CPU* (6) y finalmente se crea el *CPU* (7). Se envía el *Id de Equipo* a *IUAgregarPeriferico* (8), se capturan los datos del *Periférico* (9), se validan los datos del *Periférico* (10), y finalmente se crea el *Periférico* (11). Se envía el *Id de Equipo* a la interfaz *IUAgregarAccesorio* (12,) con se capturan los

datos del *Accesorio* (13), se validan los datos (14), y se crea un *Accesorio* (15). Se envía el *Id de Equipo* a *IUAgregarSoftware* (16), se capturan los datos (17), se validan los datos del nuevo *Software* agregado al *Equipo* (18), y finalmente se crea la entidad *Software* (19).

**Caso de Uso Cerrar Sesión** (fig.4.26)

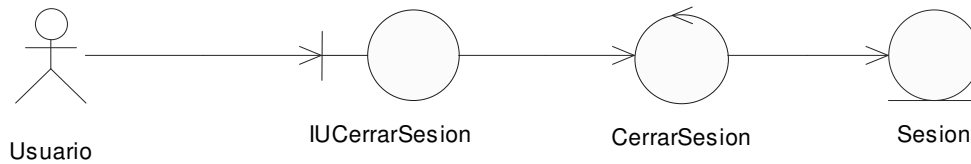


Figura 4.26 Diagrama de clases participantes del caso de uso Cerrar Sesión.

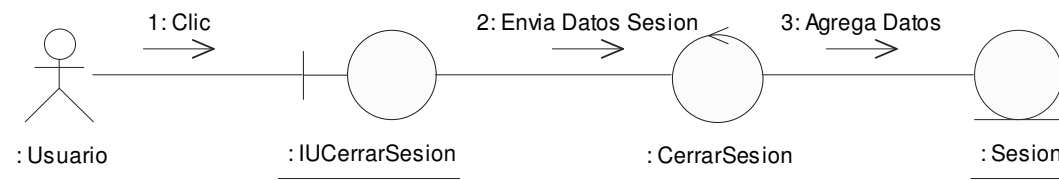


Figura 4.27 Diagrama de colaboración de clases del caso de uso Cerrar Sesión.

**Flujo de sucesos-análisis del caso de Uso Cerrar Sesión** (fig. 4.27)

El *Usuario* cierra la sesión por medio de un clic en la interfaz *IUCierraSesion* (1), se envían de datos de la *Sesión* (2). Se guardan los valores en la entidad *Sesión* que se registro al iniciar sesión. (3).

**Caso de Uso Consultar Bitácora** (fig.4.28)

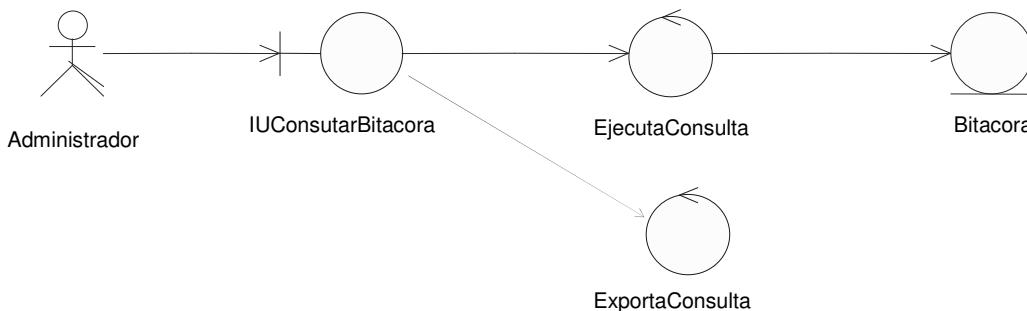


Figura 4.28 Diagrama de clases participantes del caso de uso Consultar Bitácora.

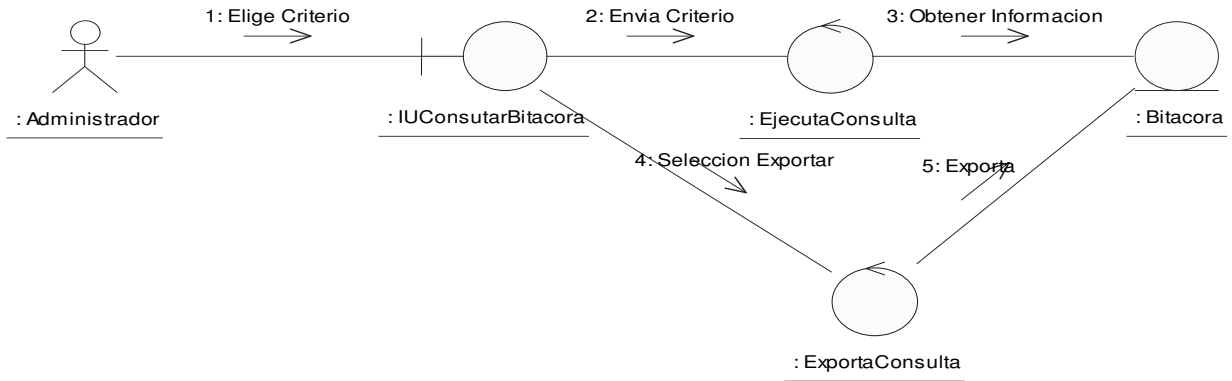


Figura 4.29 Diagrama de colaboración de clases del caso de uso Consultar Bitácora.

**Flujo de sucesos-análisis del caso de Uso Consultar Bitácora** (fig. 4.29)

El *Administrador* por medio de la interfaz *IUConsultarBitacora*, elige el criterio de búsqueda de la *Bitácora* a consultar (1). Se envía el criterio de búsqueda a *Ejecuta Consulta* (2). Se obtiene la información de la *Bitácora* (3). Si el *Administrador* lo decide se puede elegir exportar dicho archivo por medio de *ExportaConsulta* (4). Y finalmente se exporta el archivo (5).

**Caso de Uso Generar Censo** (fig.4.30)

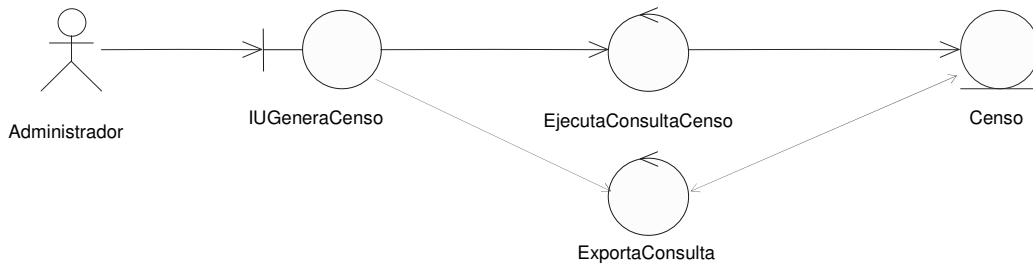


Figura 4.30 Diagrama de clases participantes del caso de uso Generar Censo.

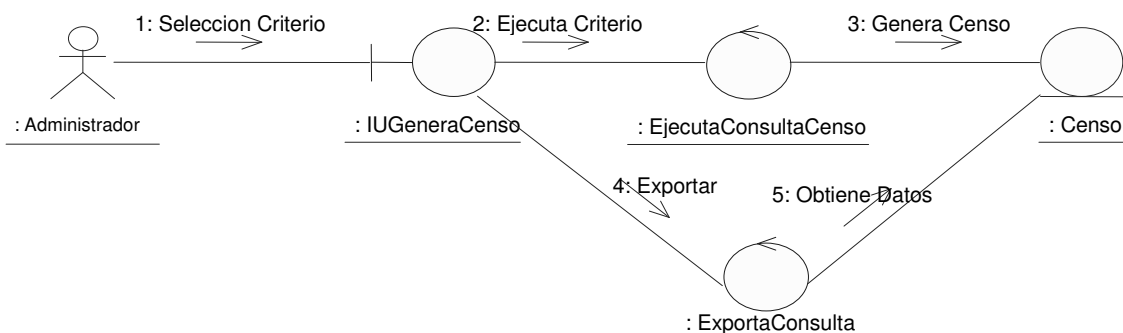


Figura 4.31 Diagrama de colaboración de clases del caso de uso Generar Censo.

**Flujo de sucesos-análisis del caso de Uso Genera Censo** (fig. 4.31)

El *Administrador* por medio de la interfaz *IUGenerarCenso*, elige el criterio para la generación del censo (1). La clase de control *EjecutaConsultaCenso* ejecuta el criterio (2). Se genera el *Censo* (3). Por medio de *ExportaConsulta* se puede elegir exportar dicho archivo (4). Y finalmente se obtienen los datos del censo para ser exportados (5).

**Caso de Uso Inicia Sesión** (fig.4.32)

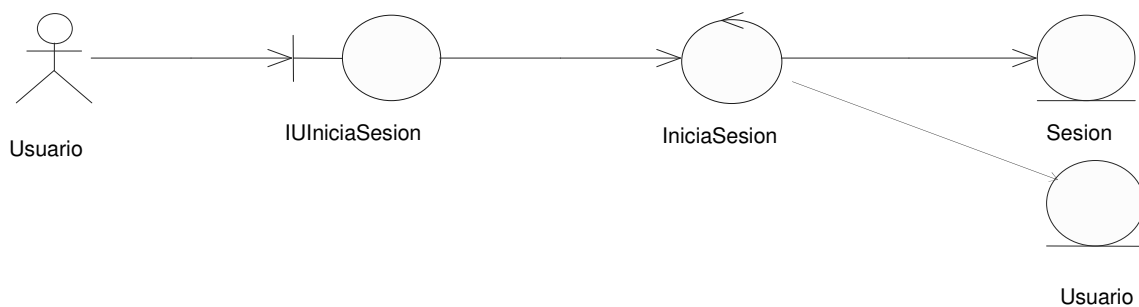


Figura 4.32 Diagrama de clases participantes del caso de uso Inicia Sesión.

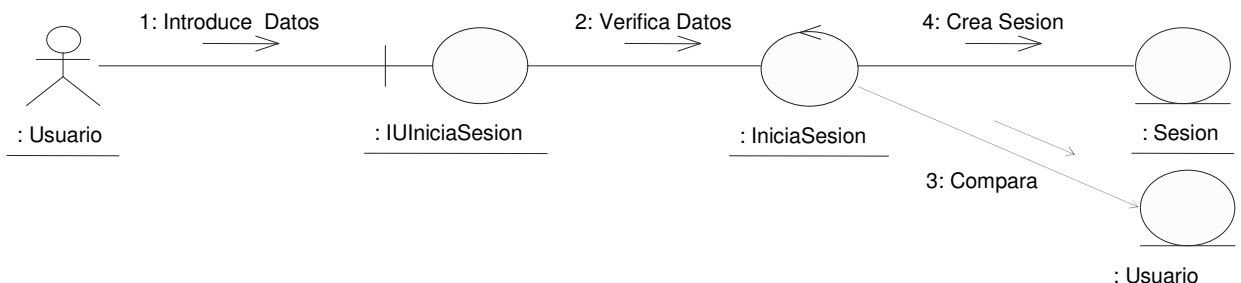


Figura 4.33 Diagrama de colaboración de clases del caso de uso Inicia Sesión.

**Flujo de sucesos-análisis del caso de Uso Inicia Sesión** (fig. 4.33)

El *Usuario* por medio de la interfaz *IInicioSesion* introduce los datos que le permitirán la entrada al sistema (1), la clase de control *IniciarSesion* verifica los datos obtenidos (2). Se compara el *Usuario* con uno ya almacenado en la base de datos (3). Y finalmente se crea la *Sesión* si el *Usuario* es correcto (4).

**Caso de Uso Modifica** (fig.4.34)



Figura 4.34 Diagrama de clases participantes del caso de uso Modifica Accesorio.

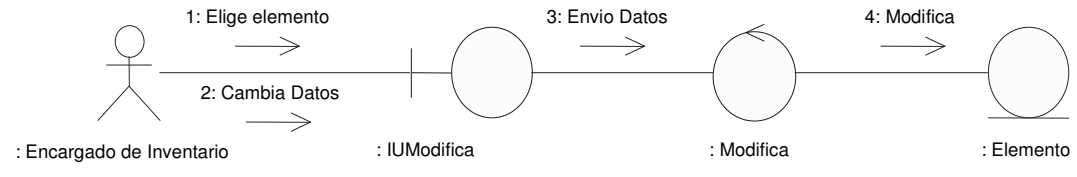


Figura 4.35 Diagrama de colaboración de clases del caso de uso Modifica Accesorio.

**Flujo de sucesos-análisis del Caso de Uso Modifica** (fig. 4.35)

El *Encargado de Inventario* por medio de la interfaz *IUModifica* elige el *Elemento* que desea modificar (1). Cambia los datos de dicho *Elemento* (2). Se realiza el envío de los datos que se van a cambiar median la clase de control *Modifica* (3). Y finalmente se hacen las modificaciones deseadas en la entidad *Elemento* (4).



**Caso de Uso Modifica Equipo** (fig.4.36)

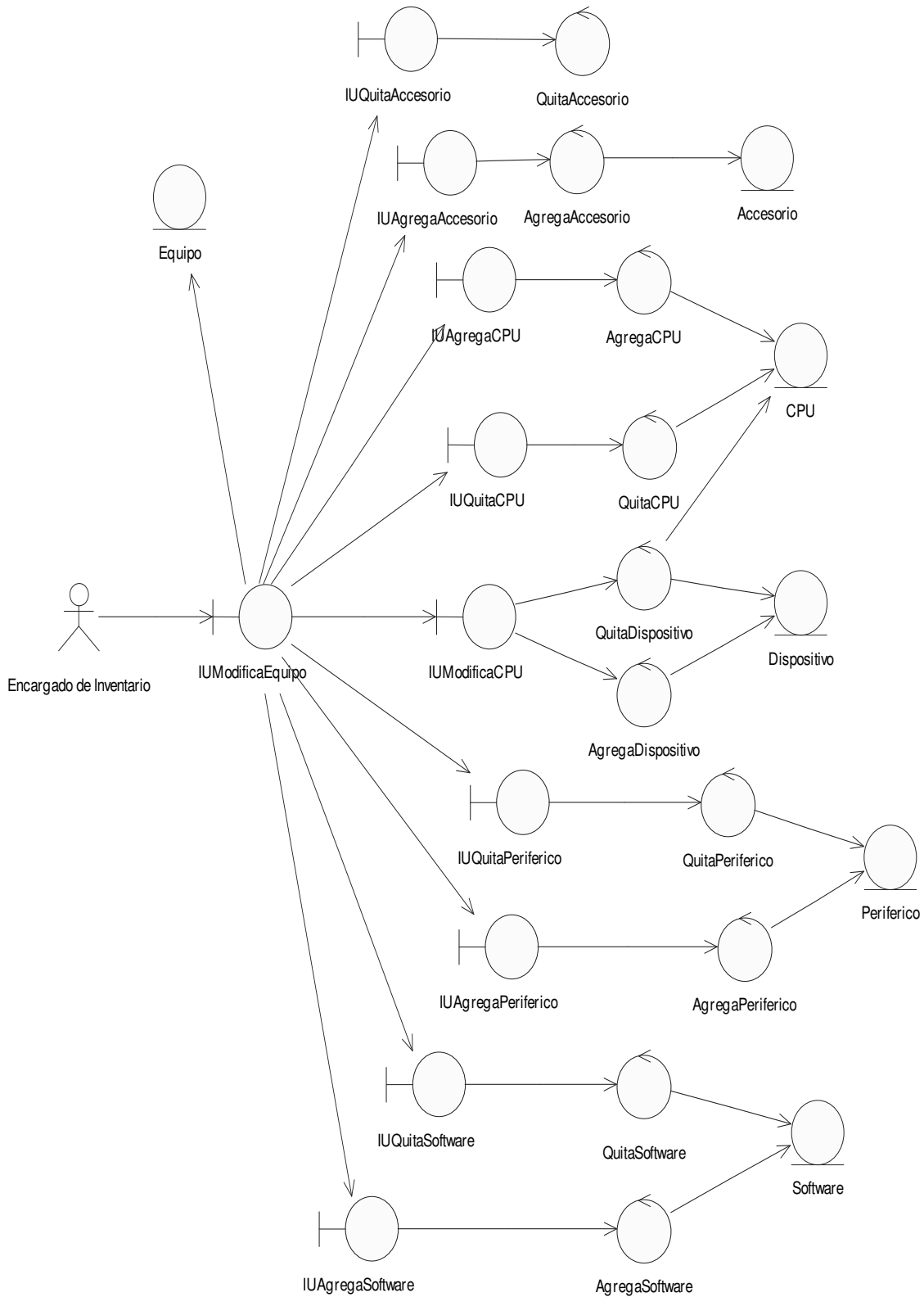


Figura 4.36 Diagrama de colaboración de clases del caso de uso Modifica Equipo.



**Flujo de sucesos-análisis del Caso de Uso Modifica Equipo** (fig. 4.37)

El *Encargado de Inventario* selecciona el *Equipo* por medio de la interfaz *IUModificaEquipo* (1). Obtiene los datos del *Equipo* (2). Por medio de la interfaz *QuitarAccesorio* se elige un *Accesorio* (3), se envía el identificador del accesorio (4), y borra la relación (32). Mediante *AgregarAccesorio* se selecciona un nuevo *Accesorio* (5), *AgregarAccesorio* valida los datos del *Accesorio* que se desea agregar (6), el *Accesorio* es creado (7). A través de *IUAgregarCPU* se elige un *CPU* (8), *AgregarCPU* lo añade al *Equipo* (9), el *CPU* es creado (10). Mediante *IUQuitarCPU* se elige un *CPU* a quitar (11), *QuitarCPU* borra la relación que existe entre el *Equipo* y el *CPU* (12,13). Usando *ModificaCPU* se selecciona el *CPU* a modificar (14). Se elige el *Dispositivo* a quitar mediante *QuitarDispositivos* (15), por medio *QuitarDispositivos*, se envía el identificador del dispositivo y borra la relación (16, 17). Para agregar se usa *IUAgregarDispositivos* en la que se selecciona el *Dispositivo* (18). *AgregarDispositivos* se encarga de crear la relación del *CPU* con el *Dispositivo* (19). *IUAgregar periféricos* permite elegir el *Periférico* para agregar, *AgregarPeriferico*, valida los datos del *periférico* que se desea agregar y el *periférico* es creado (20, 21, 22). Al utilizar *IUQuitarPeriférico* se toma el *Periférico* a quitar, por medio de *QuitarPeriferico*, se envía el identificador del Periférico, se borra la relación (23, 24,25). Para quitar *Software* se usa la *IUQuitarSoftware*, por medio de *QuitarSoftware*, se envía el identificador del *Software* y se borra la relación (26, 27, 28). *IUAgregar Software* permite capturar *Software* para un *Equipo* (29), por medio de *AgregarSoftware*, se valida los datos del *Software* que se desea agregar y el *software* es agregado (29, 30,31)

**Caso de uso Quitar** (fig.4.38)



Figura 4.38 Diagrama de clases participantes del caso de uso Quitar Accesorios.

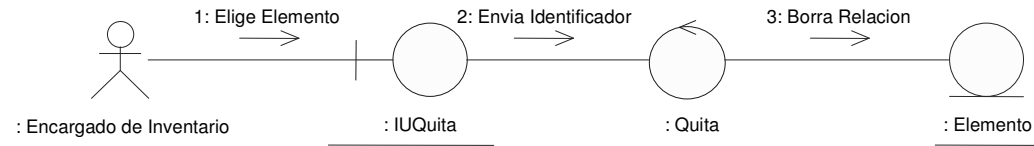


Figura 4.39 Diagrama de colaboración de clases del caso de uso Quitar.

**Flujo de sucesos-análisis del Caso de Uso Quitar** (fig. 4.39)

El *Encargado de Inventario* elige el *Elemento* por medio de *IUQuita* (1). Envía el identificador a *Quita* para validar el *Elemento* que se desea quitar (2), finalmente borra la relación del *Elemento* con el *Equipo* al que pertenecía (3).

**Caso de uso Revisar Solicitudes** (fig.4.40)

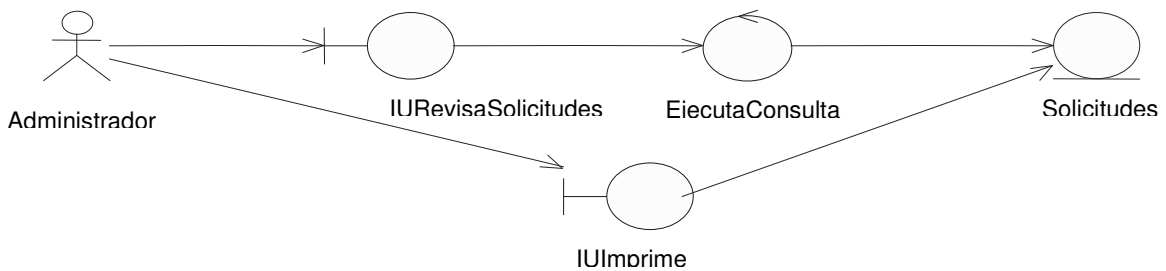


Figura 4.40 Diagrama de clases participantes del caso de uso Revisar Solicitudes.

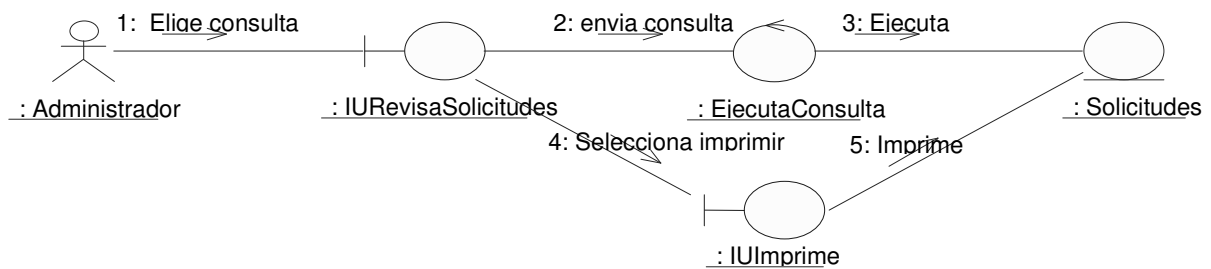


Figura 4.41 Diagrama de colaboración de clases del caso de uso Revisar Solicitudes.

**Flujo de sucesos-análisis del Caso de Uso Revisar Solicitudes** (fig. 4.41)

El *Administrador* elige la consulta para revisar las solicitudes por medio de *IURevisaSolicitudes* (1). Envía la consulta a *EjecutaConsulta*, la cual ejecuta dicha consulta (2,3). Se selecciona imprimir consulta por medio de *IUImprime* (4). Finalmente se imprime la consulta solicitada (5).

**Caso de Uso Solicitar** (fig.4.42)

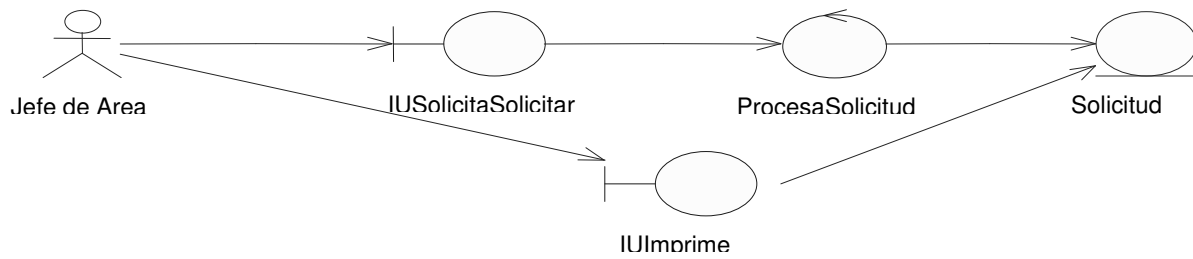


Figura 4.42 Diagrama de clases participantes del caso de uso Solicitar Baja.

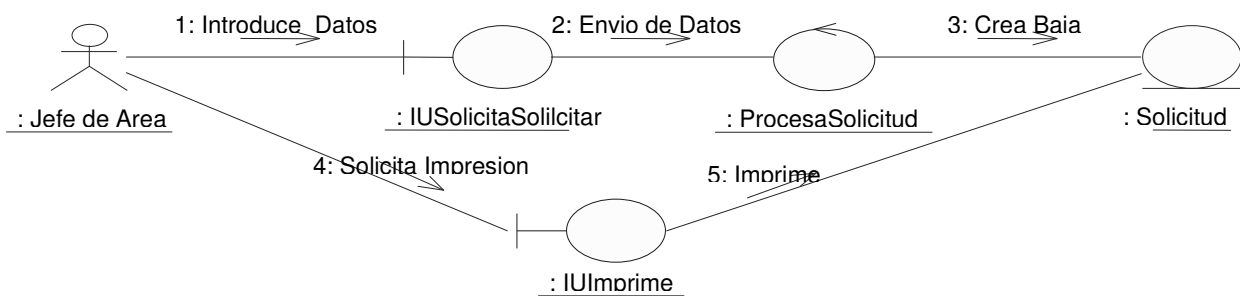


Figura 4.43 Diagrama de colaboración de clases del caso de uso Solicitar Baja.

**Flujo de sucesos-análisis del Caso de Uso Solicitar Baja** (fig. 4.43)

El *Jefe de Área* introduce los datos de la *Solicitud* que desea realizar por medio de *IUSolicitar* (1), se envían los datos a *ProcesarSolicitud* que se encarga de hacer toda la gestión para crear la entidad *Solicitud* (2,3). A través de *IUImpriime* se imprime la *Solicitud* (4). Finalmente se hace la impresión de la *Solicitud* (5).

**4.6 Analizar Clases**

Consiste en identificar las responsabilidades, atributos, asociaciones, entre clases, agregaciones y posibles generalizaciones.

**4.6.1 Identificar responsabilidades**

Se obtienen de la recopilación de todos los roles de la clase en las realizaciones de caso de uso del sistema.

### 4.6.2 Identificar Atributos

Los atributos son información requerida por las responsabilidades de la clase.

Características de los atributos:

- Debe ser un sustantivo
- Conceptuales y no restringidos por el entorno de programación
- No pueden compartirse por varios objetos.

### 4.6.3 Identificar Asociaciones, Agregaciones y Generalizaciones

Consiste en la identificación de las relaciones que existen entre las clases del análisis.

## 4.7 Análisis de las Clases del Sistema de Inventario

### 4.7.1 Identificar Atributos

- Equipo
  - Atributos
    - Nombre: cadena.
    - Número: cadena.
    - Resguardo: cadena.
    - Nombre del Propietario: cadena.
    - Apellido Paterno del Propietario: cadena.
    - Apellido Materno del Propietario: cadena.
- Accesorio
  - Atributos:
    - Número de Inventario: cadena.
    - Número de serie: cadena.

- Marca: cadena.
- Modelo: cadena.
- Tipo: cadena.
- Descripción: cadena.
- CPU
  - Atributos
    - Número de Inventario: cadena.
    - Número de Serie: cadena.
    - Marca: cadena.
    - Modelo: cadena.
    - Estado: cadena.
- Dispositivo
  - Atributos
    - Número de Inventario: cadena
    - Marca: cadena.
    - Modelo: cadena.
    - Número de serie: cadena.
    - Tipo: cadena.
    - Descripción: cadena.
- Periférico
  - Atributos:
    - Número de Serie: cadena.
    - Marca: cadena.
    - Número de inventario: cadena.
    - Tipo: cadena.
    - Subtipo: cadena.
    - Descripción: cadena.
- Software
  - Atributos:
    - Nombre: cadena.
    - Versión: cadena.

- Tipo: cadena.
- Usuario
  - Atributos:
    - Login: cadena.
    - Nombre: cadena.
    - Apellido Paterno: cadena.
    - Apellido Materno: cadena.
    - Cargo: cadena.
    - Password: cadena.
    - Estado: cadena.
    - Tipo: cadena.
  - Equipo de Red
    - Atributos:
      - Número de inventario: cadena.
      - Número de serie: cadena.
      - Marca: cadena.
      - Modelo: cadena.
      - Tipo: cadena.
      - Descripción: cadena.
    - Capacidad
      - Atributos:
        - Identificador: cadena.
        - Capacidad: cadena.
      - Ubicación
        - Atributos:
          - Dependencia: cadena.
          - Departamento: cadena.
          - Ubicación: cadena.
        - Proyecto



- Atributos:
  - Clave: cadena.
  - Nombre: cadena.
- Marca
  - Atributos:
    - Identificador: cadena.
    - Marca: cadena.
- Resolución
  - Atributos:
    - Identificador: cadena.
    - Resolución: cadena.
- Velocidad:
  - Atributos:
    - Identificador: cadena.
    - Velocidad: cadena.
- Tamaño Monitor
  - Atributos:
    - Identificador: cadena.
    - Tamaño de Monitor: cadena.
- Tipo
  - Atributos:
    - Identificador: cadena.
    - Tipo: cadena.
- Bitácora
  - Atributos:
    - Identificador: cadena.
    - Fecha: fecha.
    - Descripción: cadena.

- Sesión
  - o Atributos:
    - Identificador: cadena.
    - Fecha: fecha/hora.
    - Hora de Inicio: fecha/hora.
    - Hora de Fin: fecha/hora.
- Solicitud:
  - o Atributos:
    - Folio: cadena.
    - Identificador de la bitácora: cadena.
    - Login: cadena.
- Compra
  - o Atributos:
    - Folio: cadena.
    - Clave: cadena.
    - Área: cadena.
    - Teléfono: cadena.
    - Nombre del que Autorizo: cadena.
    - Apellido Paterno del que Autorizo: cadena.
    - Apellido Materno del que Autorizo: cadena
    - Anexos: cadena.
    - Cargo: cadena.
    - Observaciones: cadena.
    - Nombre del Responsable: cadena.
    - Apellido Paterno del Responsable: cadena.
    - Apellido Materno del Responsable: cadena.
    - Nombre de la persona de Suficiencia: cadena.
    - Apellido Paterno de la persona de Suficiencia: cadena.
    - Apellido Materno de la persona de Suficiencia: cadena.
    - Nombre del Secretario: cadena.
    - Apellido Paterno del Secretario: cadena.

- Apellido Materno del Secretario: cadena.
- Costo: cadena.
- Fecha: fecha/hora.
- Descripción: cadena.
- Unidad: cadena.
- Cantidad: cadena.
- Precio: cadena.
- Importe: cadena.
- Baja
  - Folio: numero.
  - Fecha: fecha.
- Mantenimiento
  - Folio: cadena.
  - Fecha: fecha.
  - Nombre de la persona que autorizó: cadena.
  - Apellido Paterno de la persona que autorizó: cadena.
  - Apellido Materno de la persona que autorizó: cadena.
  - Teléfono: cadena.
  - Descripción: cadena.
  - Descripción del equipo: cadena.
  - Observaciones: cadena.

#### **4.7.2 Identificar Responsabilidades**

- Agregar Accesorio
  - o Responsabilidades
    - En el caso de uso Agregar Accesorio dicha clase se encarga de validar los datos del accesorio que se desea capturar, con el identificador de equipo al que se desea agregar un accesorio.
    - Llama la conexión a la base de datos para hacer la inserción de los datos del accesorio que se desea agregar.

- Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean insertar.
- Agregar Dispositivo
  - Responsabilidades:
    - En el caso de uso Agregar Dispositivo dicha clase se encarga de validar los datos del dispositivo que se desea capturar, con el identificador de equipo al que se desea agregar un dispositivo.
    - Llama la conexión a la base de datos para hacer la inserción de los datos del dispositivo que se desea agregar.
    - Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean insertar.
- Alta Usuario
  - Responsabilidades:
    - En caso de uso Alta Usuario dicha clase se encarga de validar los datos del usuario que se desea capturar.
    - Llama la conexión a la base de datos para hacer la inserción de los datos del usuario que se desea dar de alta.
    - Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean insertar.
- Baja
  - Responsabilidades
    - En caso de uso Baja Accesorio dicha clase se encarga de validar los datos de los accesorios que se desean eliminar.
    - Llama la conexión a la base de datos para hacer el borrado del Accesorio.
    - Maneja los errores que se presentan tanto en la conexión con la base de datos, como en los datos mismos que se desean eliminar.

- Baja Usuario
  - o Responsabilidades
    - En caso de uso Baja Usuario dicha clase se encarga de validar los datos de los usuarios que se desean eliminar.
    - Llama la conexión a la base de datos para hacer el borrado del usuario.
    - Maneja los errores que se presentan tanto en la conexión con la base de datos, como en los datos mismos que se desean eliminar.
  
- Captura
  - o Responsabilidades:
    - En caso de uso Captura Accesorio dicha clase se encarga de validar los datos del accesorio que se desea capturar.
    - Llama la conexión a la base de datos para hacer la inserción de los datos del accesorio que se desea capturar.
    - Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean insertar.
  
- Captura Equipo
  - o Responsabilidades:
    - En caso de uso captura Equipo dicha clase se encarga de validar los datos del equipo que se desea capturar.
    - Llama la conexión a la base de datos para hacer la inserción de los datos del equipo que se desea capturar.
    - Llama a la clase Agregar Accesorio, para saber que accesorios están relacionados con ese equipo.
    - Llama a la clase Agregar Dispositivo, para saber que dispositivos están relacionados con ese equipo.
    - Llama a la clase Agregar Software para saber que software está almacenado en ese equipo.
    - Llama a la clase Agregar Periférico, para saber que periféricos están relacionados con ese equipo.

- Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean insertar.
  - En caso de uso Captura Tipo dicha clase se encarga de validar los datos del tipo que se desea capturar.
  - Llama la conexión a la base de datos para hacer la inserción de los datos del tipo que se desea capturar.
  - Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean insertar.
- Cerrar Sesión
    - Responsabilidades
      - En caso de uso Cerrar Sesión, dicha clase se encarga de fijar la hora de fin de sesión de usuario.
- Iniciar Sesión
    - Responsabilidades:
      - En caso de uso Inicia Sesión dicha clase se encarga de validar los datos del usuario que desea ingresar al sistema.
      - Llama la conexión a la base de datos para hacer la búsqueda de los datos solicitados por el usuario.
      - Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean buscar.
- Modifica
    - Responsabilidades:
      - En caso de uso Modifica Accesorio dicha clase se encarga de validar los datos del accesorio que se desea modificar.
      - Llama la conexión a la base de datos para hacer la modificación de los datos del accesorio.
      - Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean modificar.

- Modifica Equipo

o Responsabilidades:

- En caso de uso Modifica Equipo dicha clase se encarga de validar los datos del equipo que se desea modificar.
- Llama la conexión a la base de datos para hacer la modificación de los datos del equipo.
- Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean modificar.

- Quitar

o Responsabilidades:

- En caso de uso Quitar Accesorios dicha clase se encarga de eliminar la relación existente entre el accesorio y el equipo al que pertenece.
- Llama la conexión a la base de datos para eliminar la relación entre equipo y accesorio.
- Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean modificar.

- Quitar Dispositivos

o Responsabilidades:

- En caso de uso Quitar Dispositivo dicha clase se encarga de eliminar la relación existente entre el dispositivo y el equipo al que pertenece.
- Llama la conexión a la base de datos para eliminar la relación entre equipo y dispositivo.
- Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean modificar.

- Ejecuta Consulta
  - o Responsabilidades:
    - En caso de uso Ejecuta Consulta dicha clase se encarga de manejar los informes de los movimientos realizados.
    - Llama la conexión a la base de datos para hacer la inserción de los movimientos que han sido registrados.
    - Maneja los errores que se presentan tanto en la conexión de la base de datos, como en los datos mismos que se desean insertar.
- Ejecuta Consulta Censo
  - o Responsabilidades:
    - En caso de uso Ejecuta Consulta Censo dicha clase se encarga de la realización de censos de acuerdo al tipo de censo que se solicite.
    - Llama la conexión a la base de datos para hacer la búsqueda de la información solicitada.
    - Presenta la información en la pantalla del censo solicitado.
- Procesa Solicitud
  - o Responsabilidades
    - Se encarga de generar las solicitudes capturadas.

### **4.7.3 Identificar Asociaciones y Agregaciones**

Un Equipo está formado por Accesorio, Dispositivo, Periférico y Software por lo que podemos representarlo mediante el siguiente diagrama con las relaciones uno a muchos, es decir; un Equipo puede tener varios Accesorios, Dispositivos, Periféricos y Software pero no al revés (fig. 4.44).



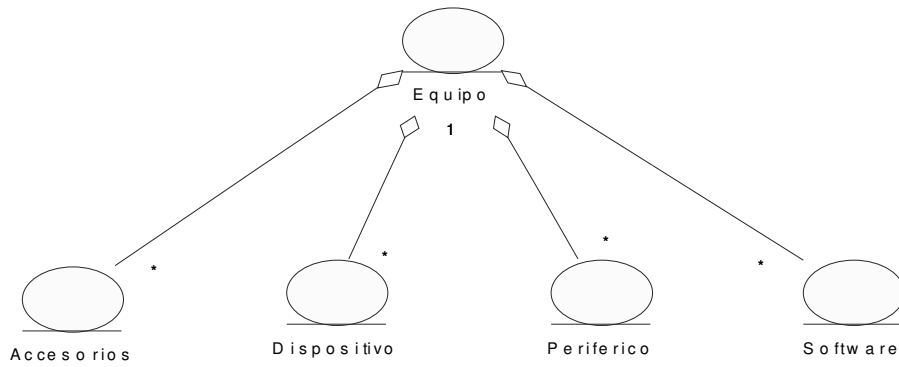


Fig. 4.44 Agregación de clases de Equipo.

Las clases Procesador, Disco Duro, Lector Óptico y RAM tienen atributos similares y representan objetos similares en el aspecto físico. Las cuatro clases son especializaciones de una clase genérica llamada Dispositivo (fig. 4.45).

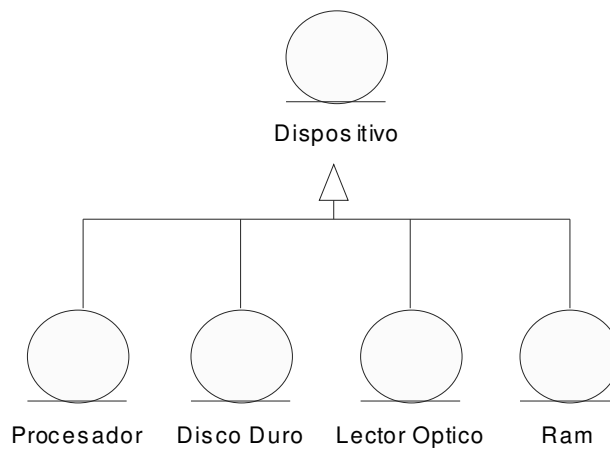


Fig. 4.45 Especialización de la clase Dispositivo.

Al igual que las clases anteriores las clases Monitor, Mouse, Teclado y Bocinas son especializaciones de la clase genérica Periférico (fig. 4.46).

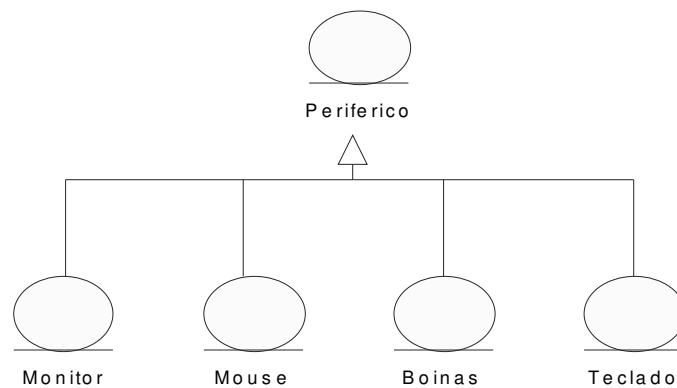


Fig. 4.46 Especialización de la clase Periférico.

Un Equipo tiene asociados un proyecto una ubicación es decir la relación que existe entre estas clases es de uno a uno (fig. 4.47).

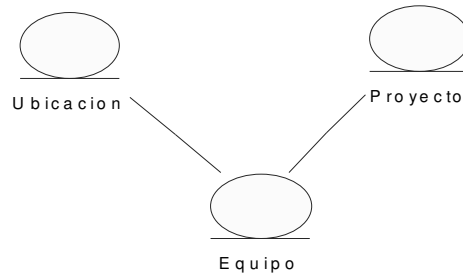


Fig. 4.47 Asociación de la clase Equipo con las clases Ubicación y Proyecto.

Las clases Equipo y Equipo Red son especializaciones de otra clase llamada Equipo Cómputo lo ya que de esta forma se manejarán los censos de una manera más eficiente (fig. 4.48).

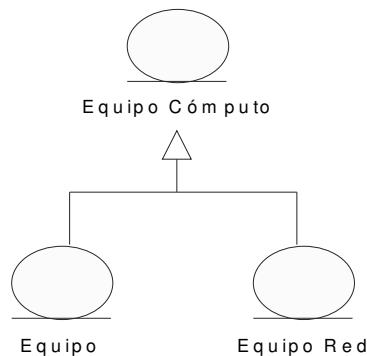


Fig. 4.48 Especialización de la clase Equipo Cómputo.

En la figura siguiente se muestra el diagrama de clases de análisis, es decir, la forma en que se relacionan todas las clases encontradas durante esta fase del proceso (fig. 4.49).

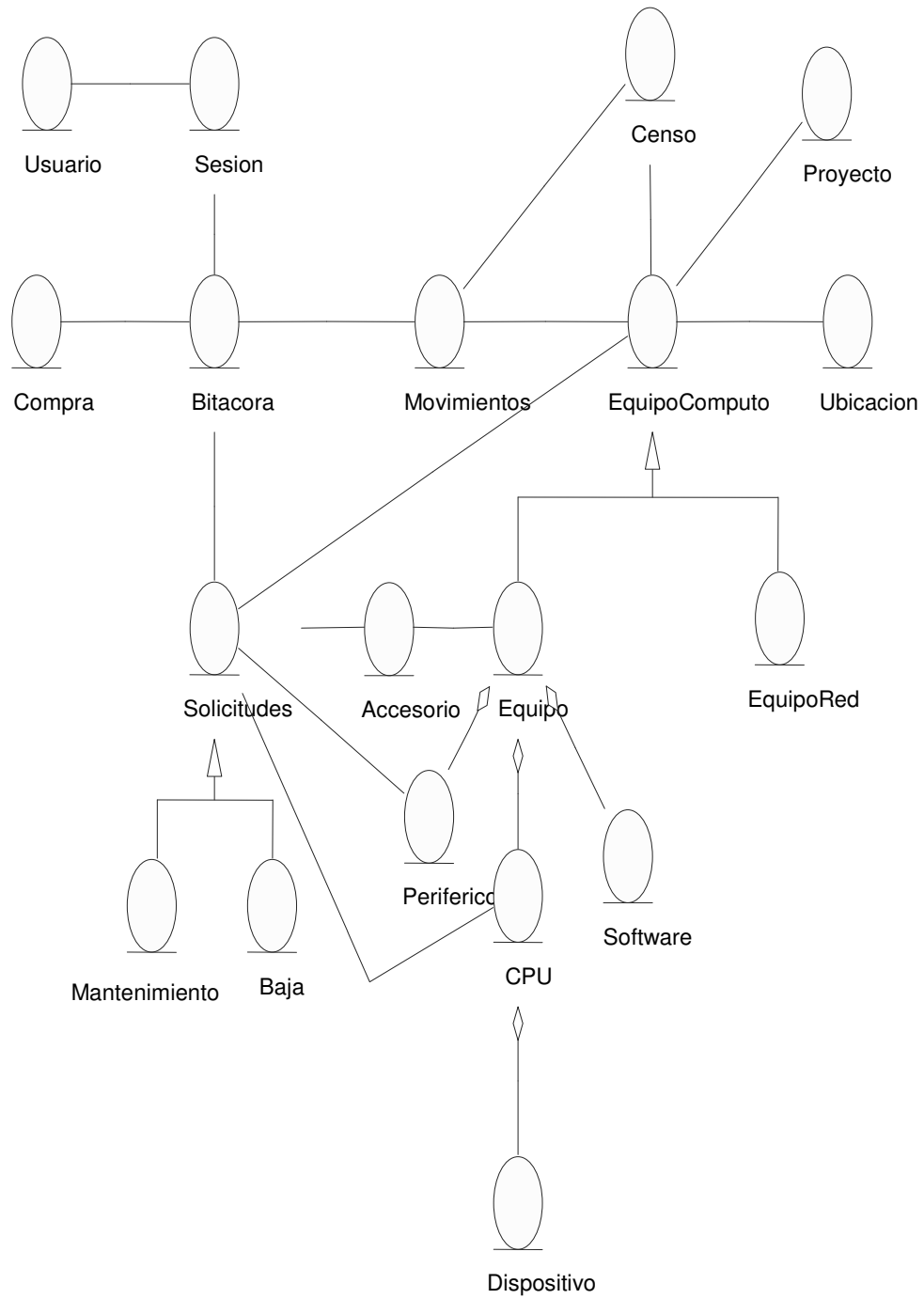


Fig. 4.49 Diagrama de clases del Análisis.

## 4.8 Modelo de diseño

Es un modelo físico y concreto, es decir; es el plano de la implementación del sistema. Por ello es un modelo más formal y debe ser mantenido durante todo el ciclo de vida del software ya que da forma al sistema al apegarse al lenguaje de programación (fig. 4.50).

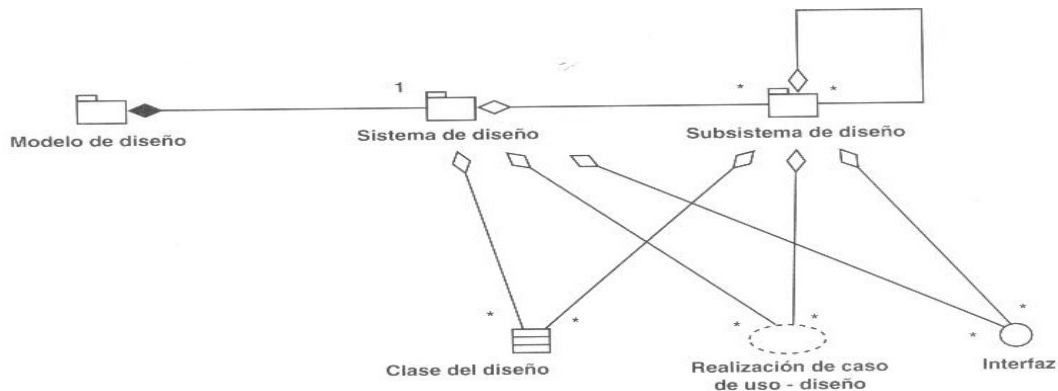


Fig.4.50 Modelo de Diseño.

## 4.9 Diseño de la Arquitectura

Consiste en identificar los nodos y sus configuraciones así como los subsistemas y sus interfaces, las clases del diseño y los mecanismos de diseño genéricos.

### 4.9.1 Identificar Nodos y sus Configuraciones

Es la creación del sistema de despliegue para la cual se utiliza de guía las siguientes preguntas:

- ¿Todos los actores del caso de uso interactuarán con el sistema en el mismo sitio físico?
- ¿Qué requerimientos de computación necesito en cada nodo?
- ¿Qué tipo de conexiones o red existen entre los nodos?
- ¿Qué protocolos manejan?
- ¿Cuáles son sus características?

- ¿Tengo requisitos del tipo Copias Redundantes para caso de fallos?
- ¿Copia de seguridad de BBDD?

#### **4.9.2 Identificar Subsistemas y sus Interfaces**

Los subsistemas constituyen un medio para organizar el modelo de diseño en piezas manejables.

#### **4.9.3 Identificar Clases de Diseño Relevantes para la Arquitectura**

Estas pueden ser:

- Clases activas: Que necesitan estar ejecutándose concurrentemente.
- Clases relacionadas con las comunicaciones entre nodos.

#### **4.9.4 Identificación de Mecanismos Genéricos**

Estos mecanismos se relacionan con los siguientes aspectos:

- Persistencia.
- Distribución transparente de objeto.
- Características de seguridad.
- Detección y recuperación de errores.
- Gestión de transacciones.

### **4.10 Diseño de la Arquitectura del Sistema de Inventario**

Determinada la finalidad del sistema, debemos conseguir el modelo arquitectónico adecuado que nos permita desarrollar la

aplicación Web. Para ello debemos tomar en cuenta los siguientes factores:

- División de la aplicación. Con un diseño modular basado en componentes de forma que las partes individuales del sistema resulten fáciles de entender y no interactúen con otras partes.
- Mantenimiento: Cuanto más simple sea el diseño, más sencillo será su mantenimiento y más complicado resultará cometer errores.
- Separación entre contenido y presentación, de forma que se pueda trabajar en cada uno de estos aspectos de forma independiente.

Tomando en cuenta estos factores, se puede utilizar el paradigma MVC/Modelo2 en la implementación del sistema ya que permite dividir en niveles independientes en función de la lógica y de la presentación y, dividir el nivel lógico en componentes relacionados con cada uno de los casos de uso (fig. 4.51).

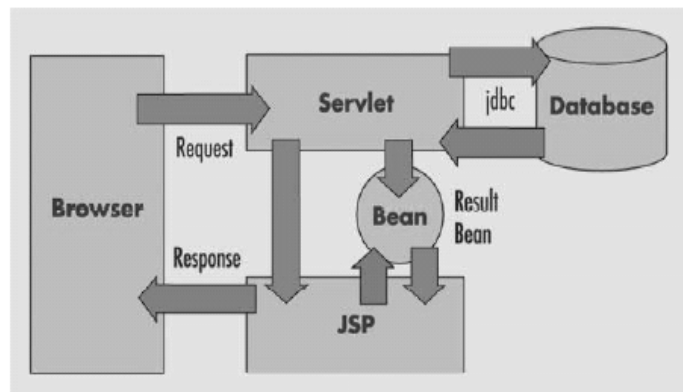


Fig. 4.51 Arquitectura Web MVC/Modelo 2.

La arquitectura modelo 2 presenta las siguientes características: escalabilidad, separación de responsabilidades, portabilidad, gestión de la sesión de usuario cacheado de entidades, aplicación de patrones de diseño. Además de que permite la división de la aplicación en tres capas (fig. 4.52).

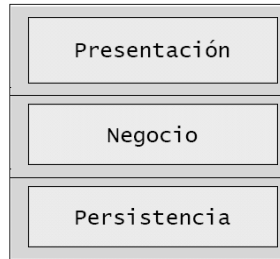


Fig. 4.52 Capas de software de la arquitectura modelo 2.

Dado que la arquitectura MVC/Modelo2 permite la utilización de patrones de diseño se aplicará el Patrón de Diseño DAO.

El problema que viene a resolver este patrón es el de contar con diversas fuentes de datos. De tal forma que se encapsula la forma de acceder a la fuente de datos. Este patrón surge históricamente de la necesidad de gestionar una diversidad de fuentes de datos, aunque su uso se extiende al problema de encapsular no sólo la fuente de datos, sino además ocultar la forma de acceder a los datos. Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los datos o de cuál es la fuente de almacenamiento.

Las aplicaciones pueden utilizar el API JDBC para acceder a los datos de una base de datos relacional. Este API permite una forma estándar de acceder y manipular datos en una base de datos relacional. El API JDBC permite a las aplicaciones J2EE utilizar sentencias SQL, que son el método estándar para acceder a tablas y vistas. La idea de este patrón es ocultar la fuente de datos y la complejidad del uso de JDBC a la capa de presentación o de negocio.

Un DAO define la relación entre la lógica de presentación y empresa por una parte y por otra los datos. El DAO tiene un interfaz común, sea cual sea el modo y fuente de acceso a datos (fig. 4.53).

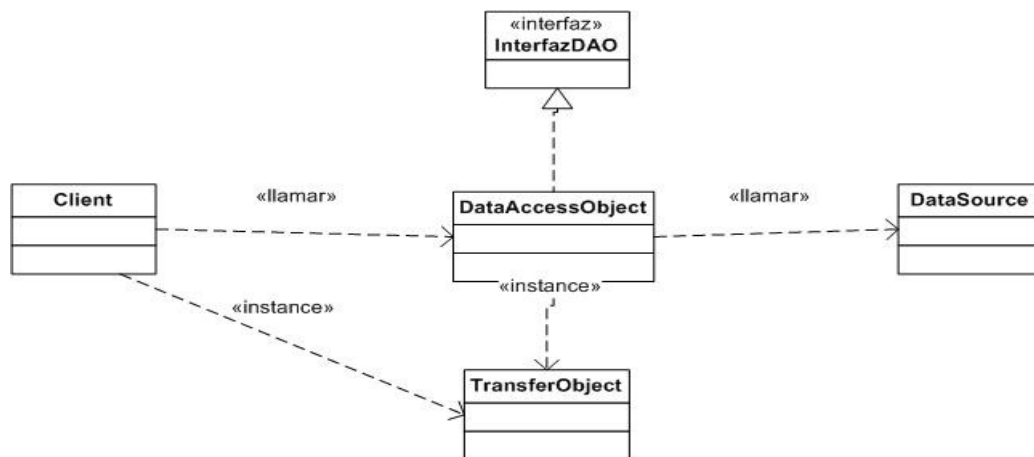


Fig. 4.53 Diagrama de clases del Patrón de Diseño DAO.

Algunas características:

- No es imprescindible, pero en proyectos de cierta complejidad resulta útil que el DAO implemente un interfaz. De esta forma los objetos cliente tienen una forma unificada de acceder a los DAO.
- El DAO accede a la fuente de datos y la encapsula para los objetos clientes. Entendiendo que oculta tanto la fuente como el modo (JDBC) de acceder a ella.
- El TransferObject encapsula una unidad de información de la fuente de datos. El ejemplo sencillo es entenderlo como un "bean de tabla", es decir, como una representación de una tabla de la base de datos, por lo que representamos las columnas de la tabla como atributos del TransferObject. El DAO crea un TransferObject (o una colección de ellos) como consecuencia de una transacción contra la fuente de datos. Por ejemplo, una consulta sobre ventas debe crear tantos objetos (TransferObject) de la clase Venta como registros de la consulta; el DAO devolverá la colección de TransferObject de la clase Venta al objeto Cliente. También puede ocurrir que el objeto Cliente mande un TransferObject para parametrizar una consulta o actualización de datos por parte del DAO.



En el siguiente gráfico se muestran las interacciones entre los elementos del patrón. En este gráfico el TransferObject se denomina ValueObject. Puede observarse las llamadas que recibe y genera el DAO para una consulta y actualización de datos (fig. 4.54):

1. El DAO es creado por el cliente (BusinessObject) (llamada 1 del gráfico).
2. A continuación el cliente solicita los datos al DAO (getData) (2).
3. El DAO responde a la llamada pidiendo los datos a la fuente de datos (2.1).
4. Para cada fila recibida, el DAO crea un TransferObjet (ValueObject del gráfico) (2.2).
5. El DAO devuelve al cliente el(los) TransferObject (2.3).
6. A continuación el cliente define un TransferObject mediante llamadas a setProperty. Por ejemplo, supongamos que buscamos personas de sexo varón y 36 años; para ello el BusinessObject define en el objeto de la clase Persona la edad y sexo que busca. Lo siguiente es fácil de imaginar: el BusinessObject invoca al DAO, pasando a la persona como argumento (3,4, y 5 del gráfico).
7. En DAO.setData() se solicita (5.1 y 5.2) al TransferObjet o ValueObject los datos (edad, sexo, etc.) para realizar el acceso a datos (dataSource.setData()), (5.3).

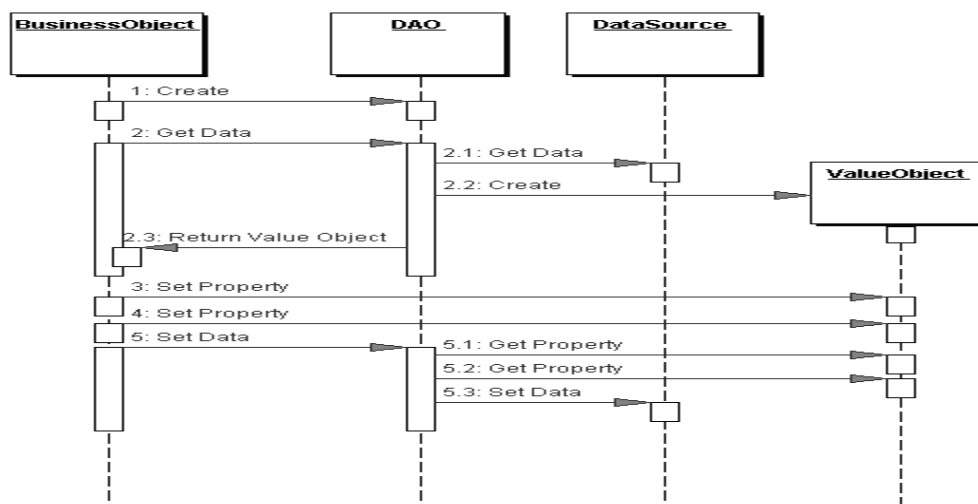


Fig. 4.54 Diagrama de secuencia del Patrón de Diseño DAO.

#### 4.10.1 Identificación de Nodos y Configuración de Red

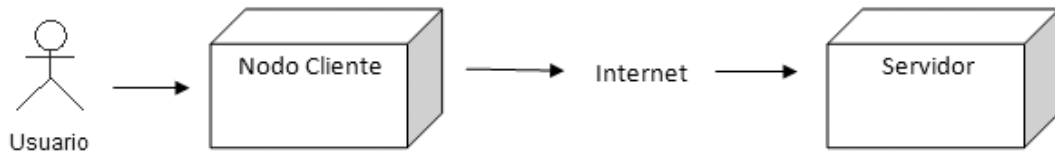


Fig. 4.55 Configuración de nodos.

El sistema de inventario se ejecutara sobre un nodo servidor, y en un cierto número de nodos clientes, en primer lugar tenemos el nodo servidor dónde se almacenarán los objetos del negocio y se realizará el procesamiento de los mismos. Todos los usuarios del sistema acceden al sistema mediante los nodos cliente. Estos nodos se comunican mediante Internet (fig. 4.55).

#### 4.10.2 Identificación de Subsistemas y sus Interfaces

Los paquetes Registro de Inventario, Actualización de Inventario, Mantenimiento, Gestión de Solicitudes, Administración y Gestión de Usuarios del modelo de análisis se utilizan para identificar los correspondientes subsistemas del modelo de diseño (fig. 4.56).

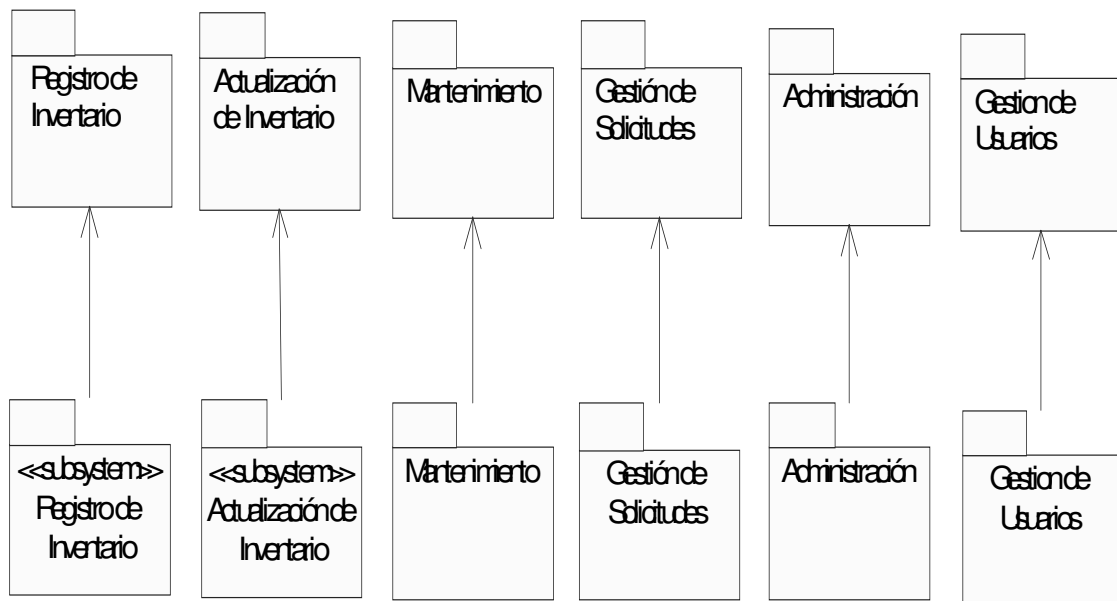


Fig. 4.56 Subsistemas y su correspondiente relación con paquetes de análisis.

**Dependencia y capas (fig. 4.57)**

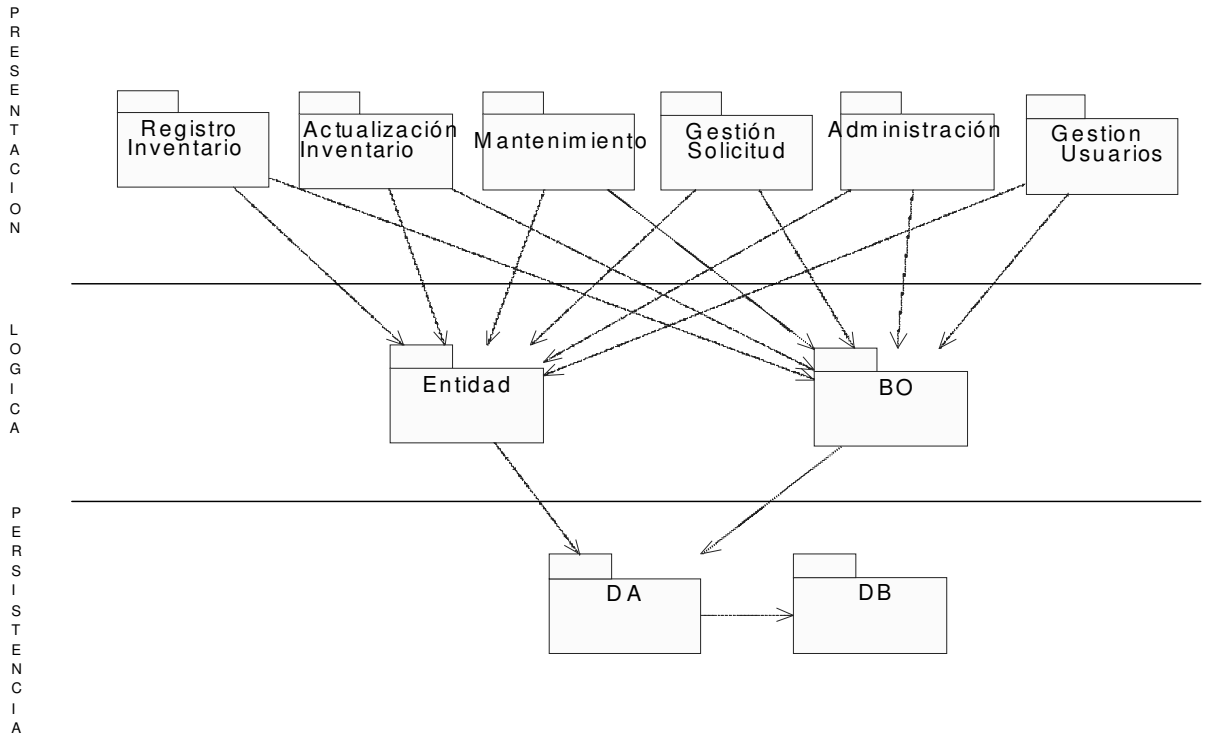


Fig. 4.57a Distribución General en Capas.

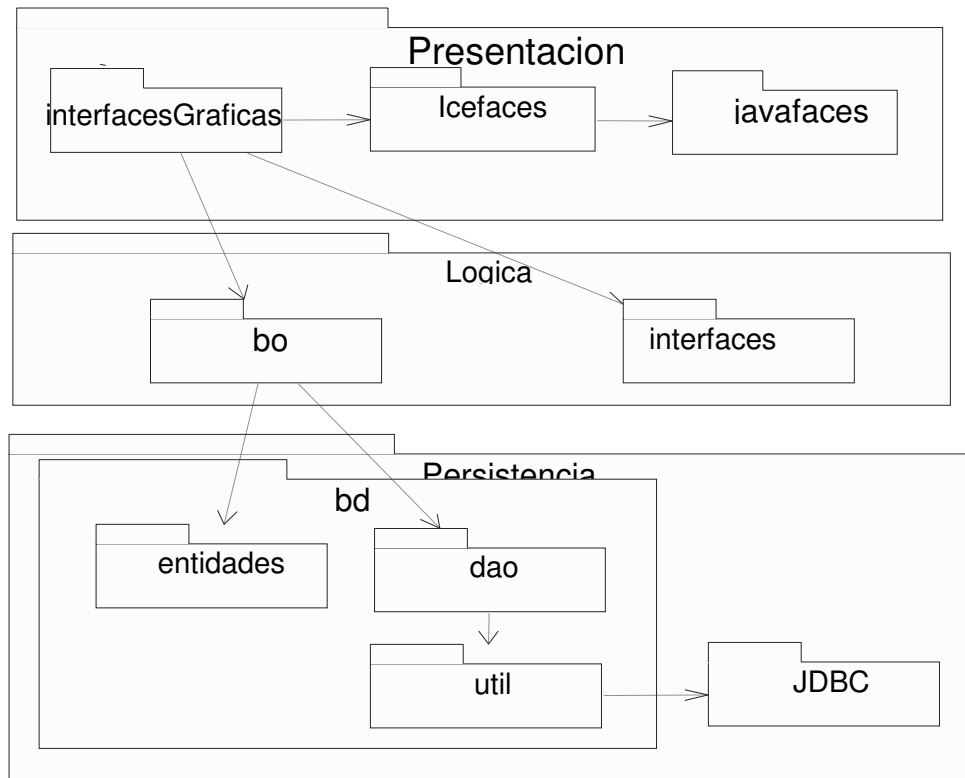


Fig. 4.57b Distribución General en Capas.

En la capa de presentación se encuentran todos los subsistemas que muestran las interfaces de usuario, estos subsistemas se comunican con los subsistemas BO y Entidades para poder obtener acceso a la base de datos, a su vez BO y Entidades se comunica con el subsistema DAO que tiene relación con el paquete BD (fig. 4.57b).

### **4.10.3 Identificación de Mecanismos Genéricos de Diseño**

#### **Seguridad**

Los mecanismos de seguridad que se utilizarán serán los siguientes:

- Autenticación: las diferentes partes que se comunican entre sí deben probar sus credenciales para demostrar que son quienes dicen ser, por medio de la autenticación.
- Autorización: una vez identificados los usuarios, es necesario determinar los recursos de la aplicación a los que tienen acceso. Esto se puede controlar por medio de listas de control de acceso (ACL) que muestran el permiso específico necesario para acceder a cada uno de los recursos.
- Integridad de los datos: la integridad de datos asegura que los datos enviados por una parte no interfieren con terceras partes antes de llegar a su destino.
- Privacidad de los datos. La privacidad de los datos asegura que la información enviada por una de las partes no se puede leer por nadie más antes de llegar a su destino.

#### **Autenticación**

Autenticación básica: si la aplicación Web utiliza la autenticación http, el navegador insta al usuario a que introduzca las credenciales de

seguridad en una ventana. Estas credenciales se envían al servidor en una codificación conocida como base64.

Compendio de autenticación: similar a la anterior. La única diferencia radica en que el nombre de usuario y la contraseña se envían de vuelta al servidor en un formulario codificado.

Autenticación basada en formularios. En los dos casos anteriores los programadores no pueden controlar el aspecto visual y operativo de la ventana que utiliza el navegador para recoger las credenciales del usuarios. Sin embargo, con la autenticación basada en formularios, sí se pueden especificar una página HTML o JSP para tal efecto. Una vez identificado el usuario, el servidor comprueba si tiene el rol adecuado, para acceder al recurso.

Autenticación http: tipo de autenticación basada en Certificados de Clave Publica, mecanismo muy potente.

### **Autorización**

J2EE emplea la autorización de recursos por medio de roles. Un usuario puede tener varios roles diferentes. En aplicaciones Web J2EE, se protegen los recursos mediante la definición de barreras de seguridad en el descriptor de implementación.

### **Persistencia**

El mecanismo de persistencia será implementado mediante la base de datos de Sybase, todas las clases que deban perdurar se guardaran en una tabla correspondiente entre el modelo de clases y el modelo de datos.

## 4.11 Diseño de Casos de Uso

Está compuesto por el artefacto realización de casos de uso del diseño:

El propósito de esta actividad es diseñar cada caso de uso para:

- Identificar clases de diseño.
- Distribuir el comportamiento del caso de uso entre los objetos de diseño que interactúan entre ellos.
- Definir los requisitos sobre las operaciones de diseño.
- Capturar requisitos de implementación para el caso de uso.

### 4.11.1 Identificación de Clases de Diseño

Se identifican las clases de diseño de la siguiente manera:

- Estudiar las clases de análisis que participan en la correspondiente realización de caso de uso de análisis. Identificar clases de diseño que permitan implementar dichas clases de análisis utilizando las clases de diseño ya diseñadas en la tarea de anterior.
- Estudiar los requisitos especiales (de rendimiento, de memoria, de diseño) que deben cumplir las clases de análisis e identificar que clases de diseño son necesarias para cumplirlos.
- Diagramas de clases: Se usan para describir el comportamiento entre las clases del diseño.
- Diagramas de interacción: Se usan diagramas de secuencia para la descripción de la comunicación entre los objetos que participan en la realización de un caso de uso.
- Flujo de Sucesos-Diseño: Descripción textual que explica y complementa los diagramas de secuencia, por lo cual se debe de redactar en términos de los objetos que interactúan en el diagrama. Este flujo no debe confundirse con el Flujo de Sucesos-Análisis.

## 4.8 Diseño de los Casos de uso del Sistema de Inventario

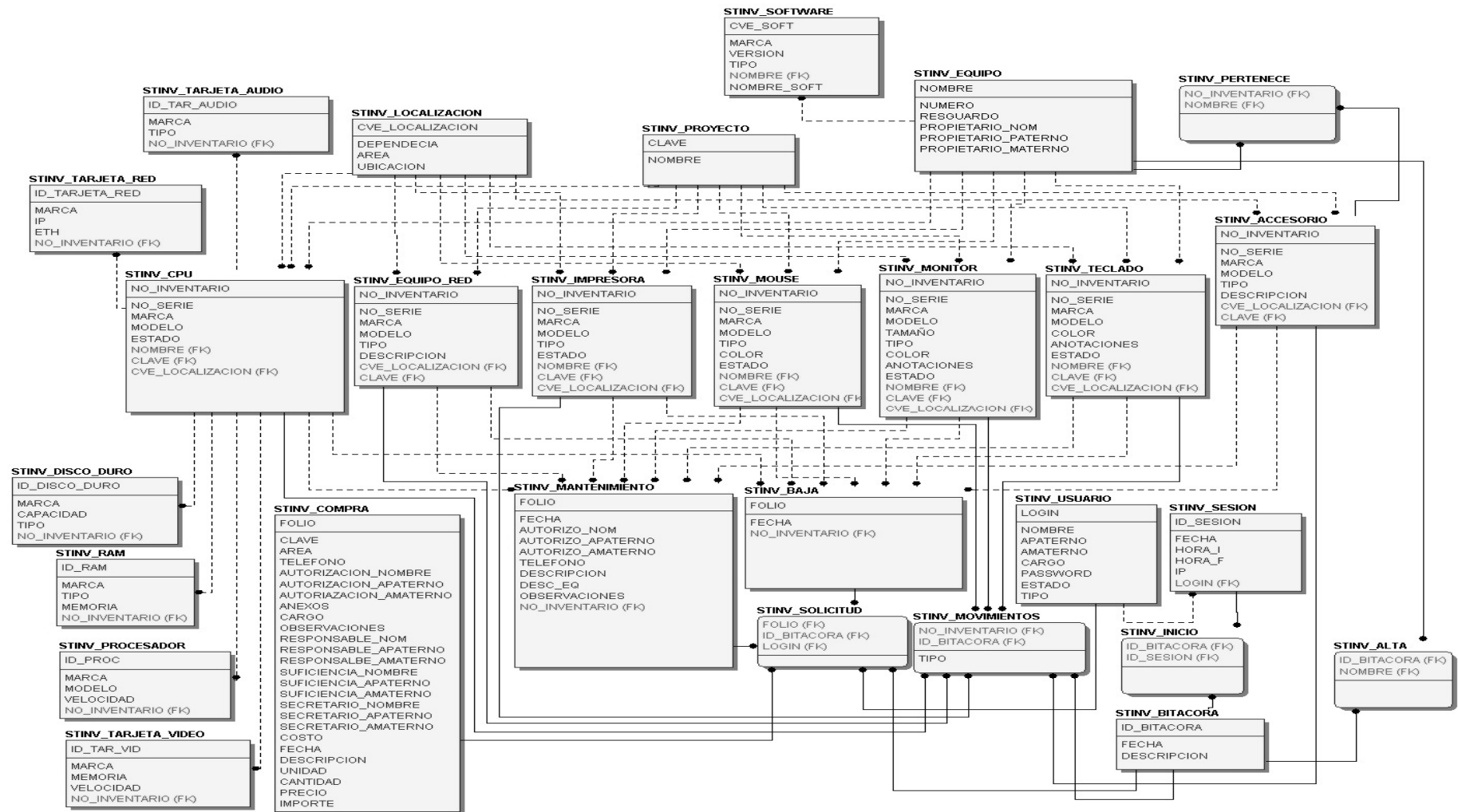


Fig. 4.58 Diagrama entidad-relación

Para lograr que el modelo de datos sea más fácil de manipular es necesario aplicarle las formas normales de las bases de datos. En los siguientes párrafos se describen las formas normales que se aplicaron a los datos.

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

Una tabla está en Primera Forma Normal sólo si:

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.
- La tabla contiene una clave primaria.
- La tabla no contiene atributos nulos.
- Si no posee ciclos repetitivos.

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

Dependencia Funcional. Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales.

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

La tabla se encuentra en 3FN si es 2FN y cada atributo que no forma parte de ninguna clave, depende directamente y no transitivamente, de la clave primaria.

Aplicando las formas normales a la base de datos inicial (fig. 4.58) obtenemos el siguiente modelo relacional mostrado en la figura (fig. 4.59).



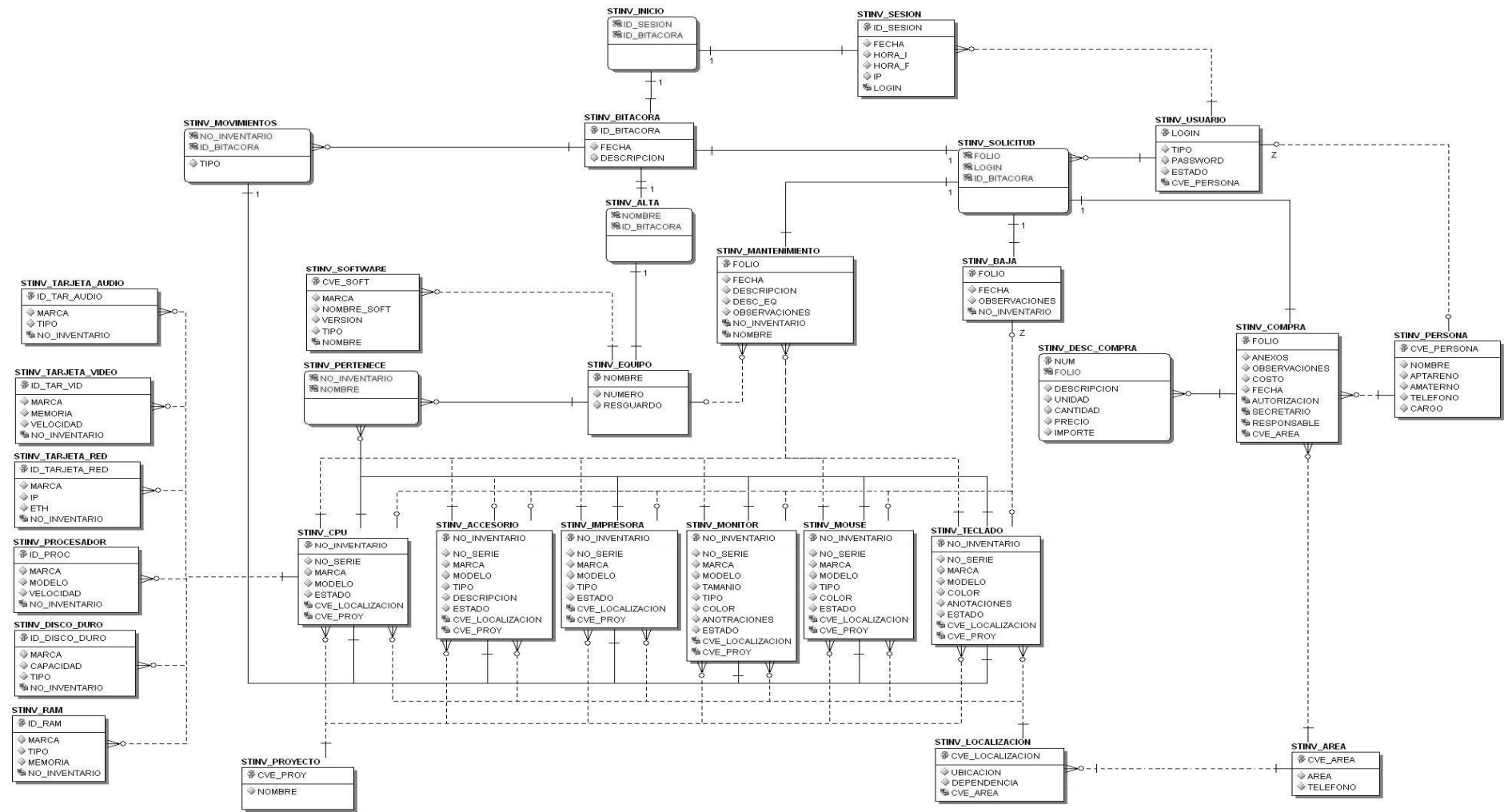


Fig. 4.59 Diagrama entidad-relación normalizada.

## Diagramas de Secuencia

### Caso de uso Agregar (fig. 4.60)

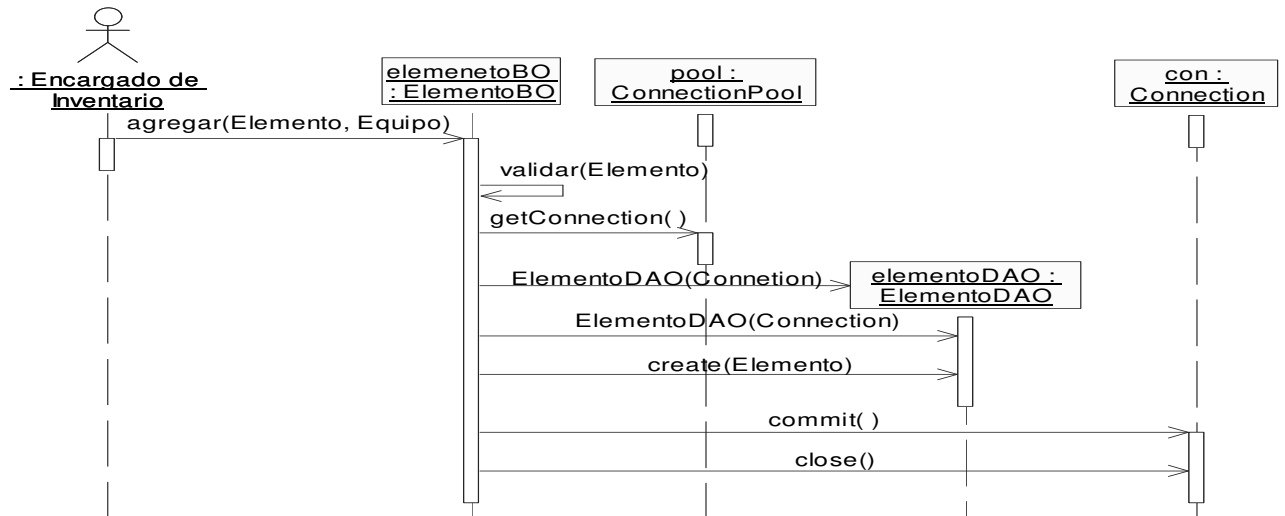


Fig. 4.60 Diagrama de Secuencia del caso de uso Agregar.

El *Encargado de Inventario* interactúa con el sistema por medio de la clase *ElementoBO*, invocando al método *agregar*, enviando los datos del *Elemento* y *Equipo* al que se le va a agregar; que a su vez valida el *Elemento*, toma una conexión de la clase *PoolConnection*. Se realiza la petición de la creación del objeto *ElementoDAO*, y se invoca al método *create*, mediante el método *commit* de la clase *Connection*. Y finalmente se cierra la conexión con el método *close*.

### Caso de uso Agregar Dispositivo (fig. 4.61)

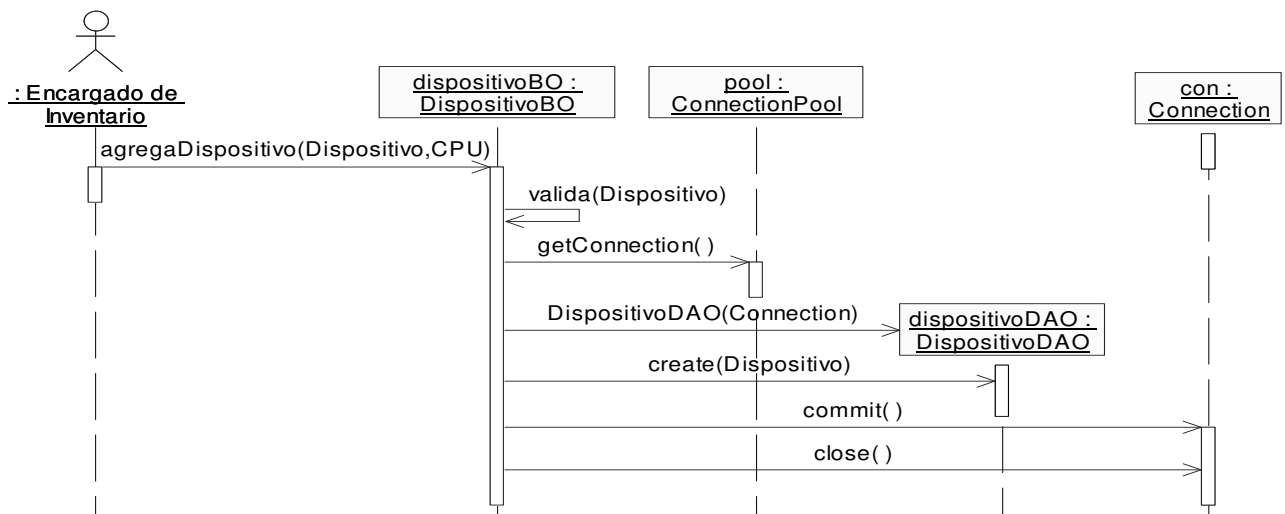


Fig.4.61 Diagrama de Secuencia del caso de uso Agregar Dispositivo.

El *Encargado de Inventario* interactúa con el sistema por medio de la clase *DispositivoBO*, invocando al método *agregar*, enviando los datos del *Elemento* y *CPU* al que se le va a agregar; que a su vez válida el *Elemento*, toma una conexión de la clase *PoolConnection*. Se realiza la petición de la creación del objeto *DispositivoDAO*, y se invoca al método *create*, mediante el método *commit* de la clase *Connection*. Y finalmente se cierra la conexión con el método *close*.

**Caso de uso Alta Usuario** (fig. 4.62)

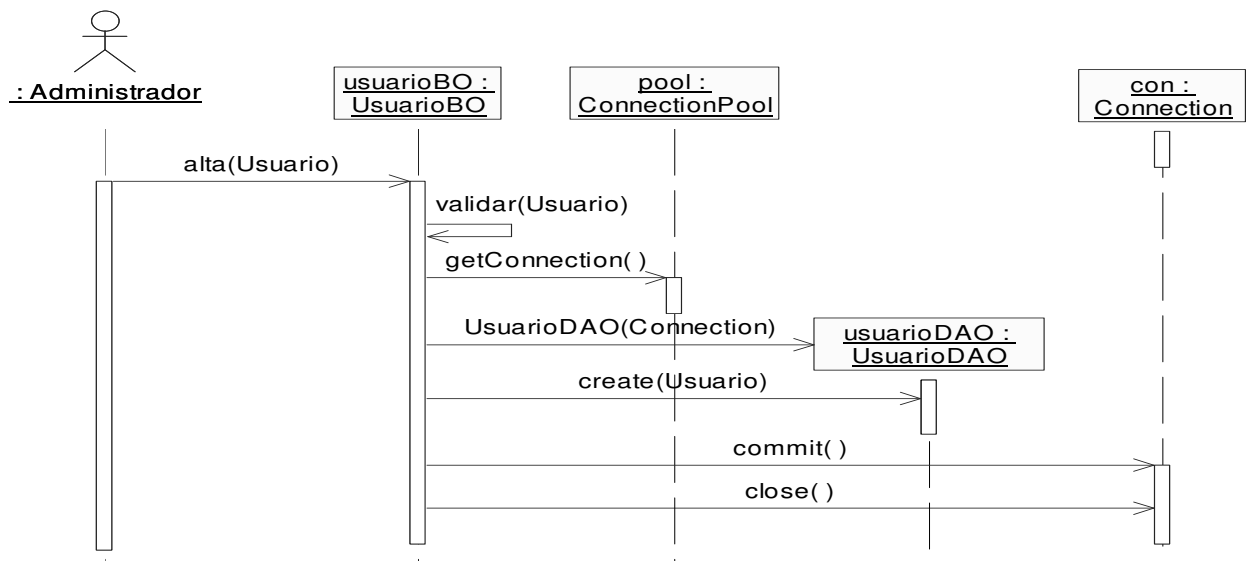


Fig. 4.62 Diagrama de secuencia del caso de uso Alta Usuario.

El *Administrador* llama al método *alta* del objeto *usuarioBO*, enviando los datos de *Usuario* que desea dar de alta; se invoca a su vez el método *valida* que comprueba que los datos de *Usuario* sean correctos. Se toma una conexión del objeto *pool*. Se realiza la petición de la creación del objeto *usuarioDAO*, y se invoca a su método *create*. Se ejecuta la transacción a la base de datos por medio del método *commit* del objeto *con*. Y finalmente se cierra la conexión mediante el método *close*.

**Caso de uso Baja** (fig. 4.63)

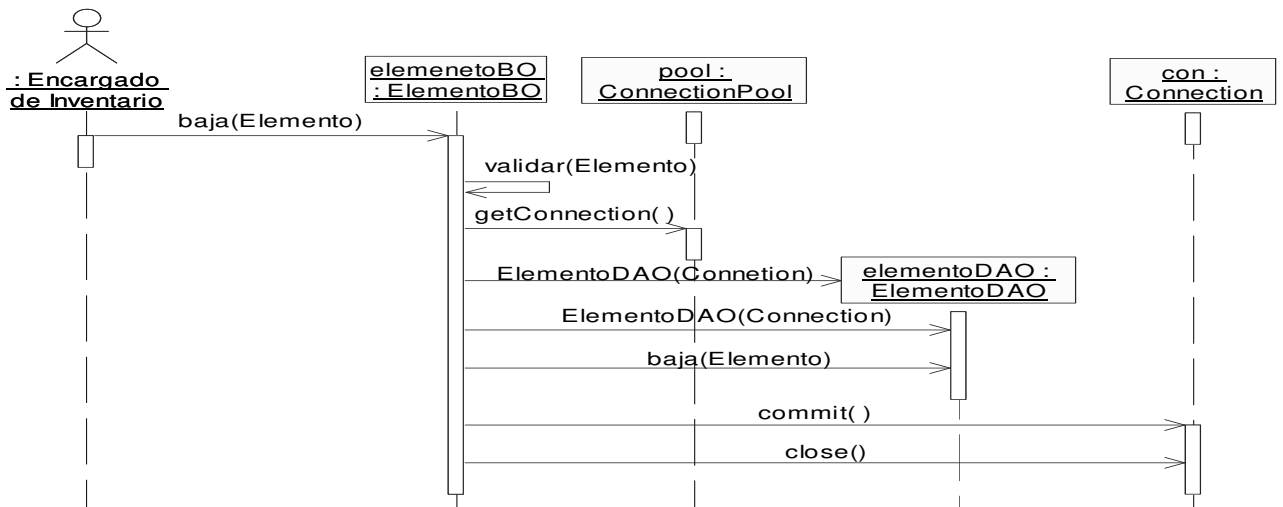


Fig. 4.63 Diagrama de secuencia del caso de uso Baja.

El *Administrador* llama al método *baja* del objeto *elementoBO*, enviando los datos del *Elemento* que desea dar de baja; se invoca a su vez el método *valida* que comprueba que los datos del *Elemento* sean correctos. Se toma una conexión del objeto *pool*. Se realiza la petición de la creación del objeto *elementoDAO*, y se invoca a su método *delete*. Se ejecuta la transacción a la base de datos por medio del método *commit* del objeto *con*. Y finalmente se cierra la conexión mediante el método *close*.

**Caso de uso Baja Usuario** (fig. 4.64)

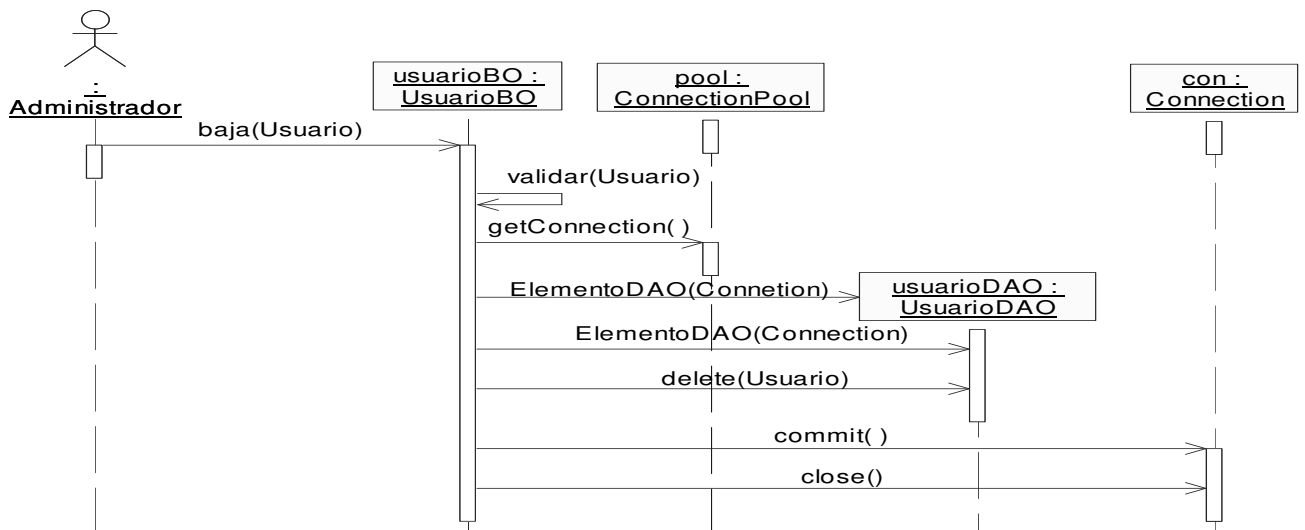


Fig. 4.64 Diagrama de Secuencia del caso de uso Baja Usuario.

El *Administrador* llama al método *baja* del objeto *usuarioBO*, enviando los datos del *Usuario* que desea dar de baja; se invoca a su vez el método *valida* que comprueba que los datos del *Usuario* sean correctos. Se toma una conexión del objeto *pool*. Se realiza la petición de la creación del objeto *usuarioDAO*, y se invoca a su método *delete*. Se ejecuta la transacción a la base de datos por medio del método *commit* del objeto *con*. Y finalmente se cierra la conexión mediante el método *close*.

**Caso de uso Captura** (fig. 4.65)

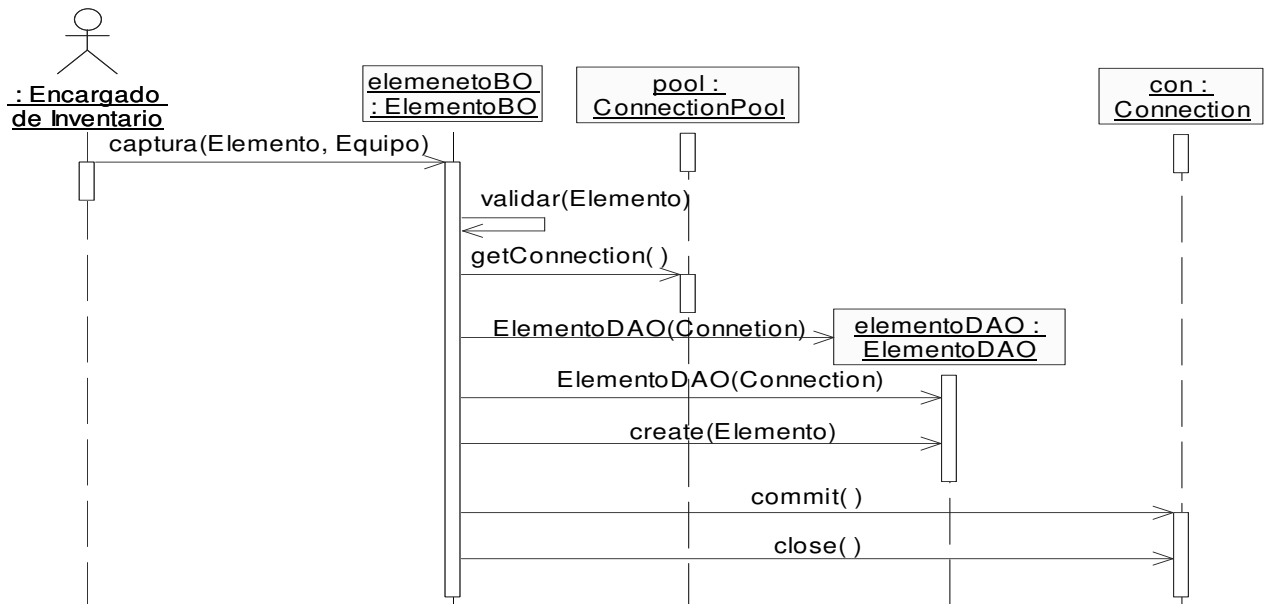


Fig. 4.65 Diagrama de Secuencia del caso de uso Captura.

El *Encargado de Inventario* interactúa con el sistema por medio de la clase *ElementoBO*, invocando al método *captura*, enviando los datos del *Elemento* y *Equipo* al que se le va a agregar; que a su vez válida el *Elemento*, toma una conexión de la clase *PoolConnection*. Se realiza la petición de la creación del objeto *ElementoDAO*, y se invoca al método *create*, mediante el método *commit* de la clase *Connection*. Y finalmente se cierra la conexión con el método *close*.

**Caso de uso Captura Equipo** (fig. 4.66)

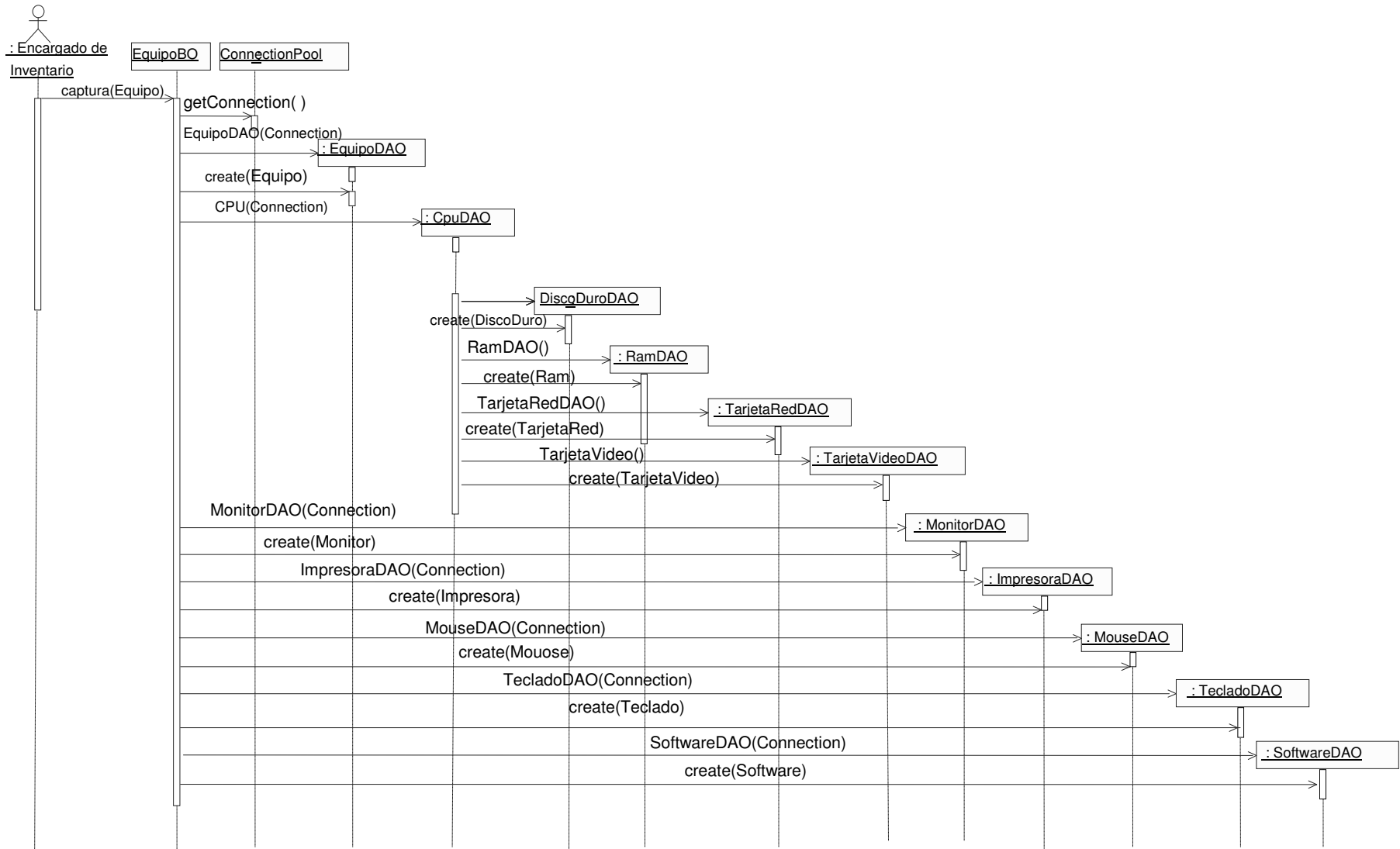


Fig. 4.66 Diagrama de Secuencia del caso de uso Captura Equipo.

El *Encargado de Inventario* invoca al método *captura* del objeto *equipoBO*, el cual toma una conexión por medio del método *getConnection* del objeto *pool*. Se realiza la petición de la creación del objeto *equipoDAO*, y se invoca al método *create*. El objeto *equipoDAO* crea la entidad *equipo* mediante el método *create*, el *equipo* será utilizado para agregar los demás elementos que lo componen. El objeto *equipoBO* llama al constructor del objeto *cpuDAO* y se invoca el método *captura* pasándole un objeto *cpu*, este objeto *cpuDAO*, construye los objetos *ramDAO*, *tarjetaVideoDAO*, *tarjetaAudioDAO*, *tarjetaRedDAO*, *procesadorDAO*, *DiscoDuroDAO* y *lectorOpticoDAO*, en cada uno de estos objetos se invoca el método *create* para registrar el tipo de objeto correspondiente. *EquipoBO* crea también estos otros objetos: *impresoraDAO*, *monitorDAO*, *monitorDAO*, *tecladoDAO* y *softwareDAO*, de los cuales se invoca a su método *create* para crear y asociar los objetos que serán registrados por cada uno de los mencionados anteriormente.

**Caso de uso Cerrar Sesión** (fig.4.67)

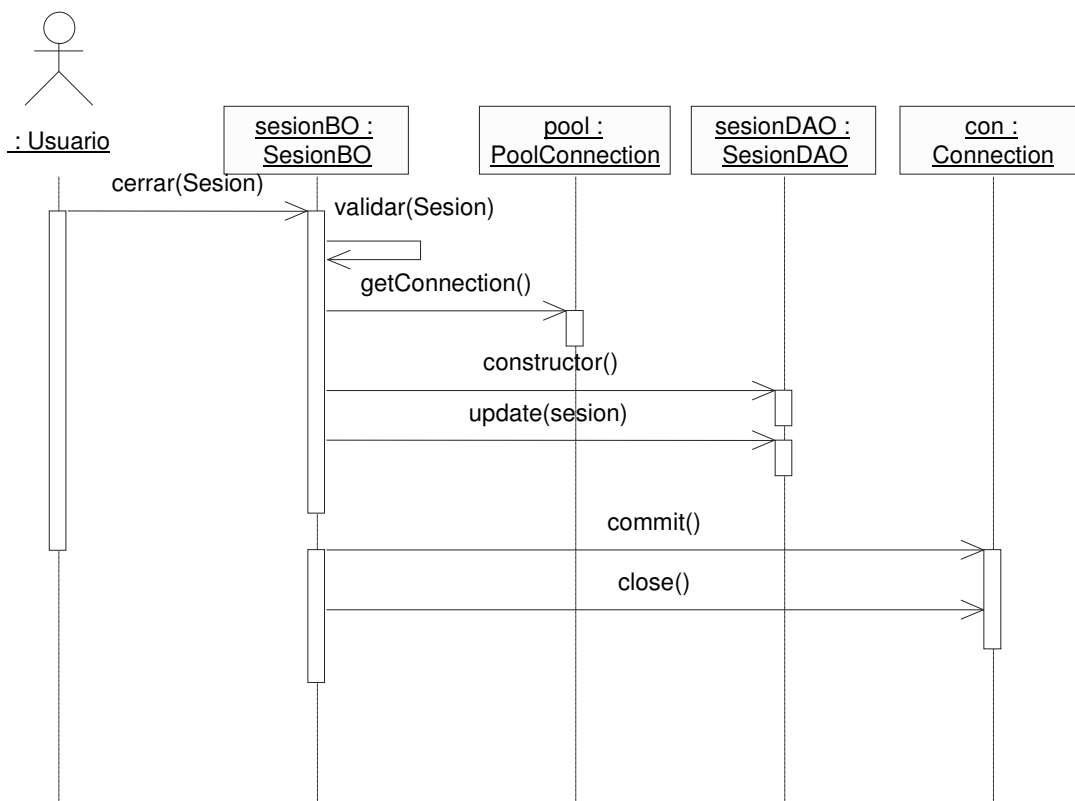


Fig. 4.67 Diagrama de Secuencia del caso de uso Cerrar Sesión.

El *Usuario* invoca al método *inicia* del objeto *sesionBO*, enviando los datos de *sesion*; que a su vez utiliza el método *valida*. Toma una conexión del *pool* con el método *getConnection*. Se realiza la petición de la creación del objeto *sesionDAO*, y se invoca al método *create*, para ejecutar la transacción de la base de datos se utiliza el método *commit* del objeto *con*. Y finalmente se cierra la conexión con el método *close* y se cierra la sesión web.

**Caso de uso Consultar Bitácora** (fig. 4.68)

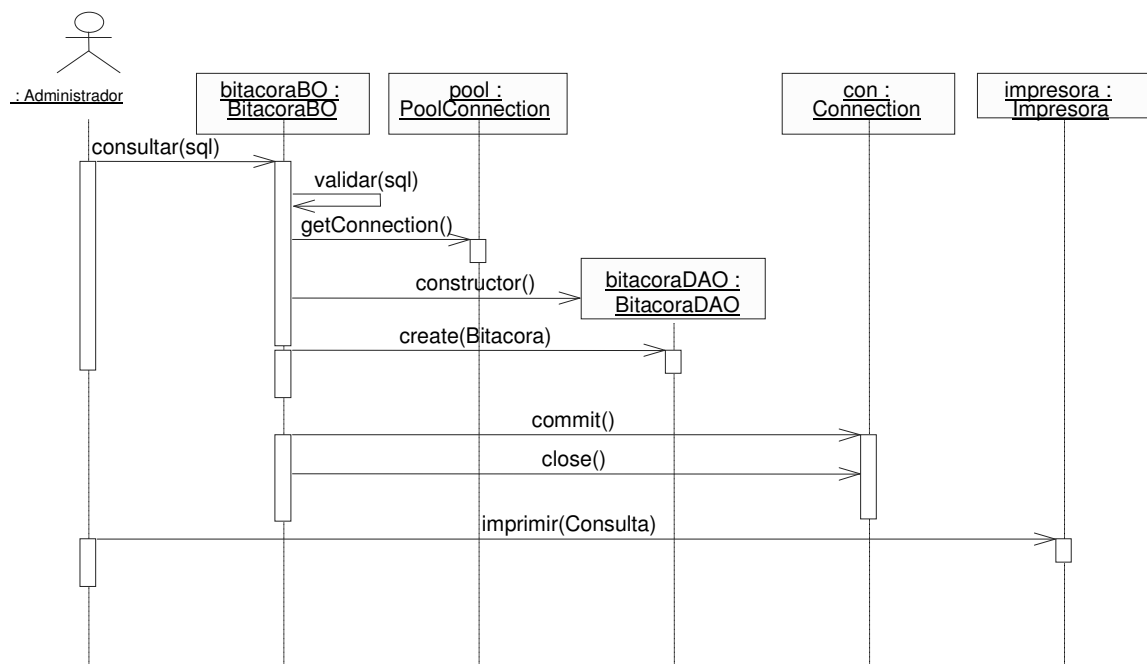


Fig. 4.68 Diagrama de Secuencia del caso de uso Consultar Bitácora.

El *Administrador* interactúa con el sistema por medio del objeto *bitacoraBO*, invocando al método *consultar*, enviando los datos de la *consulta*; que a su vez es validada por el método *valida*. Toma una conexión por medio del método *getConnection* de *pool*. Se realiza la petición de la creación del objeto *bitacoraDAO*, y se invoca al método *create* enviando la transacción por medio de *commit* del objeto *con*. Y finalmente se cierra la conexión con el método *close*.



**Caso de uso Generar Censo** (fig. 4.69)

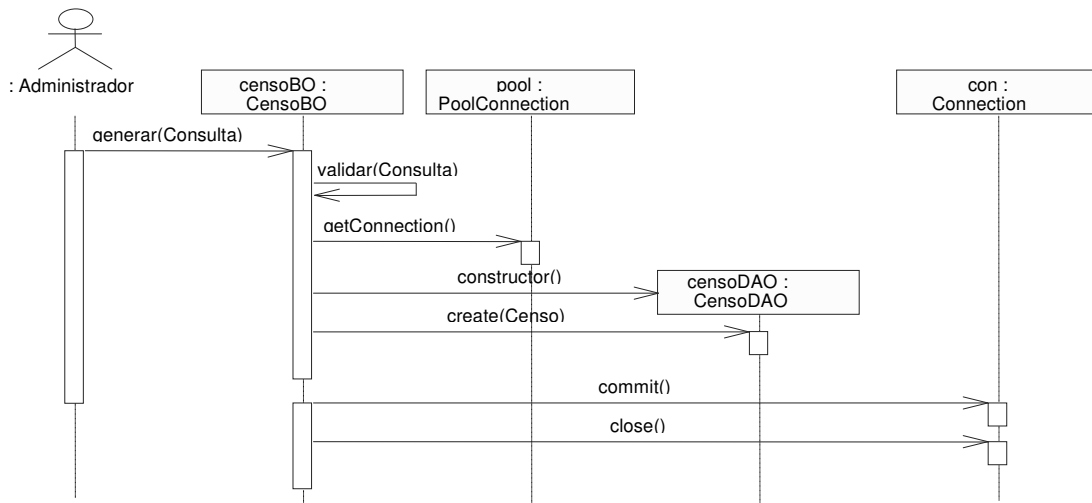


Fig. 4.69 Diagrama de Secuencia del caso de uso Generar Censo.

El *Administrador* invoca el método *generar* de *censoBO*, que a su vez valida la consulta mediante el método *validar*, se ejecuta la consulta a la base de datos mediante el método *create* del objeto *censoDAO*, y se utiliza el método *commit* del objeto *con*. Finalmente se cierra la conexión con el método *close*, y se muestra en pantalla el resultado.

**Caso de uso Iniciar Sesión** (fig. 4.70)

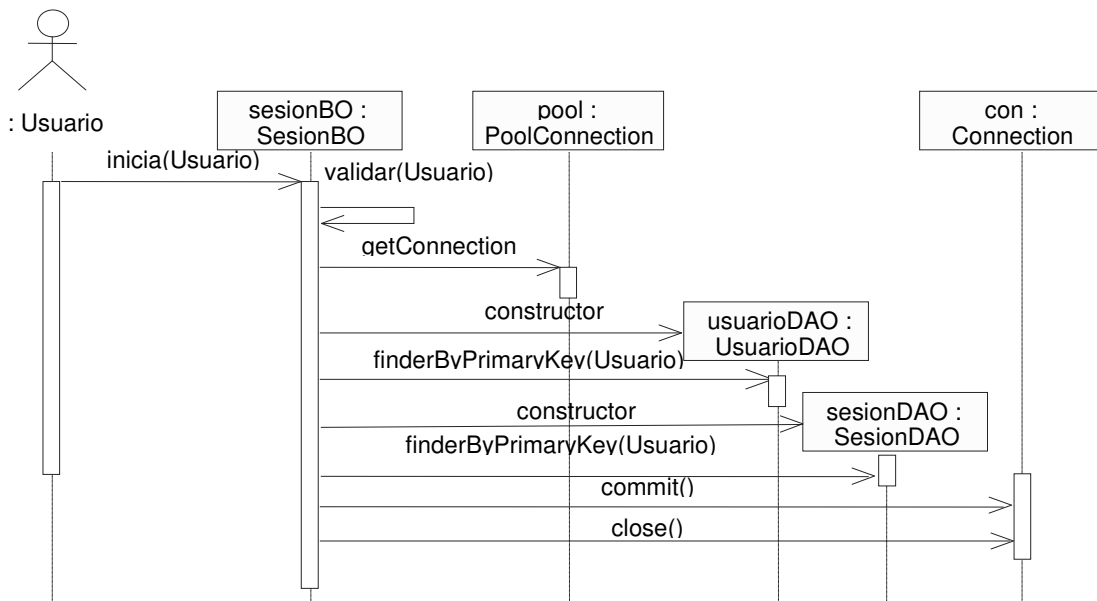


Fig. 4.70 Diagrama de Secuencia del caso de uso Inicia Sesión.

El *Usuario* invoca al método *inicia* del objeto *sesionBO*; consigue una conexión por medio del método *pool*. Se realiza la petición de la creación del objeto *usuarioDAO*, y se invoca al método *finderByPrimayKey*, se crea el objeto *sesionDAO* y se invoca al método *create* enviando la transacción por medio de *con* mediante el método *commit*. Y finalmente se cierra la conexión con el método *close*.

**Modificar** (fig. 4.71)

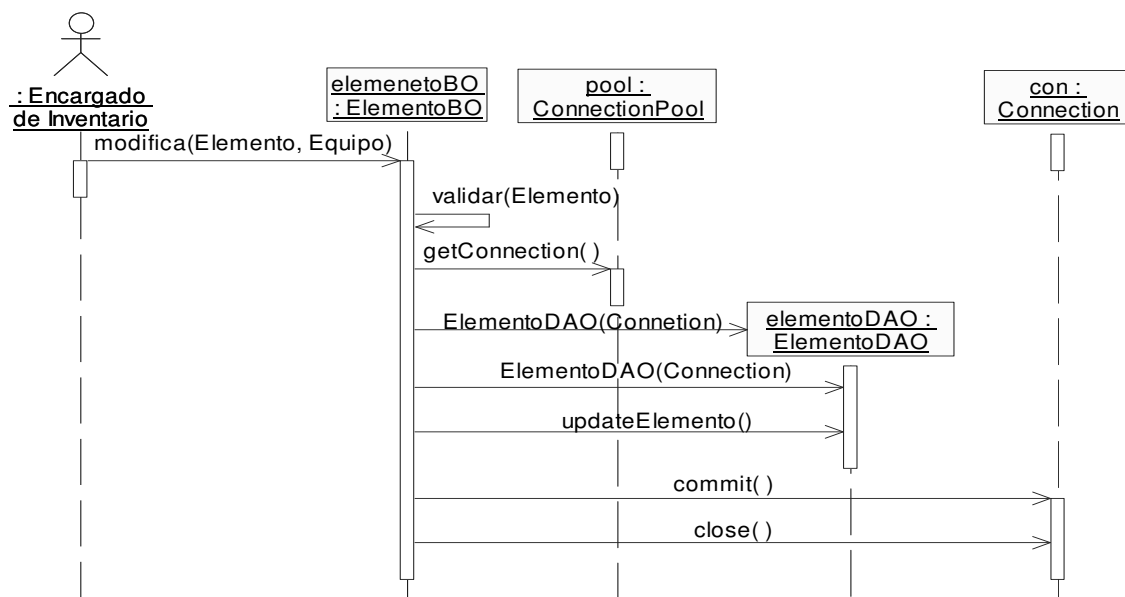


Fig. 4.71 Diagrama de Secuencia del caso de uso Modificar.

El *Encargado de Inventario* interactúa con el sistema por medio de la clase *ElementoBO*, invocando al método *captura*, enviando los datos del *Elemento* y *Equipo* al que se le va a agregar; que a su vez valida el *Elemento*, toma una conexión de la clase *PoolConnection*. Se realiza la petición de la creación del objeto *ElementoDAO*, y se invoca al método *update*, mediante el método *commit* de la clase *Connection*. Y finalmente se cierra la conexión con el método *close*.

**Caso de uso Modifica Equipo** (fig. 4.72)

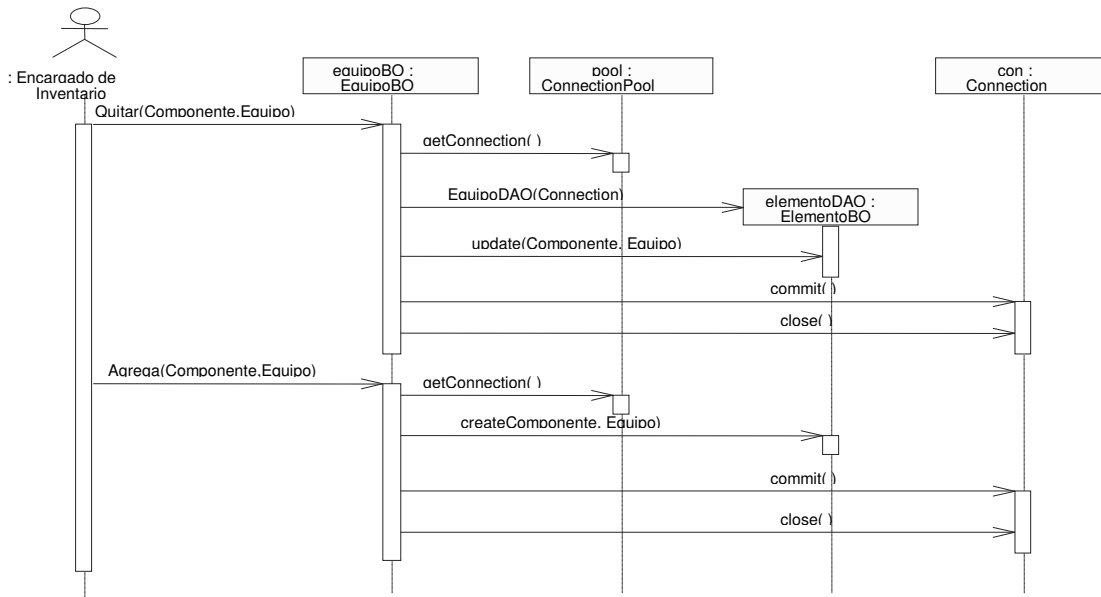


Fig. 4.72 Diagrama de Secuencia del caso de uso Modifica Equipo.

El *Encargado de Inventario* llama al método quitar del objeto *equipoBO*, este a su vez consigue una conexión del objeto *pool* mediante *getConnection*, se crea el objeto *elementoDAO* y se invoca a su método *update* para asignar al elemento que no está asignado a ningún *equipo*. Se ejecuta la transacción por medio del objeto *con* usando *commit* y cerramos la conexión con el método *close*.

El *Encargado de Inventario* llama al método agregar del objeto *equipoBO*, este a su vez consigue una conexión del objeto *pool* mediante *getConnection*, se crea el objeto *elementoDAO* y se invoca a su método *create* para asignar al elemento que no está asignado a ningún *equipo*. Se ejecuta la transacción por medio del objeto *con* usando *commit* y cerramos la conexión con el método *close*.

**Caso de uso Quitar** (fig. 4.73)

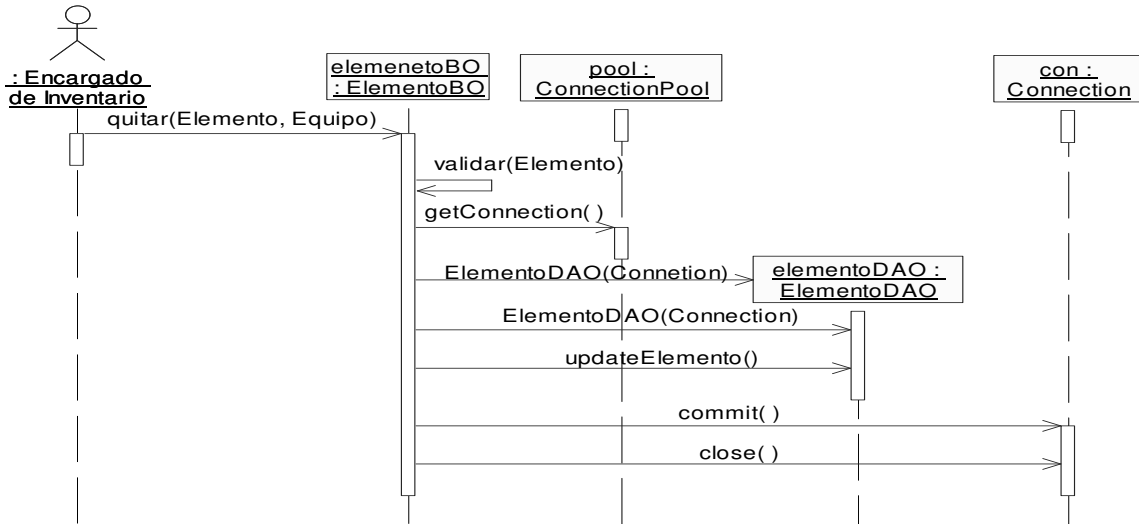


Fig.4.73 Diagrama de Secuencia del caso de uso Quitar.

El *Encargado de Inventario* llama al método *quitar* del objeto *equipoBO*, este a su vez consigue una conexión del objeto *pool* mediante *getConnection*, se crea el objeto *elementoDAO* y se invoca a su método *update* para asignar al elemento que no está asignado a ningún *equipo*. Se ejecuta la transacción por medio del objeto *con* usando *commit* y cerramos la conexión con el método *close*.

**Caso de uso Revisar Solicitudes** (fig. 4.74)

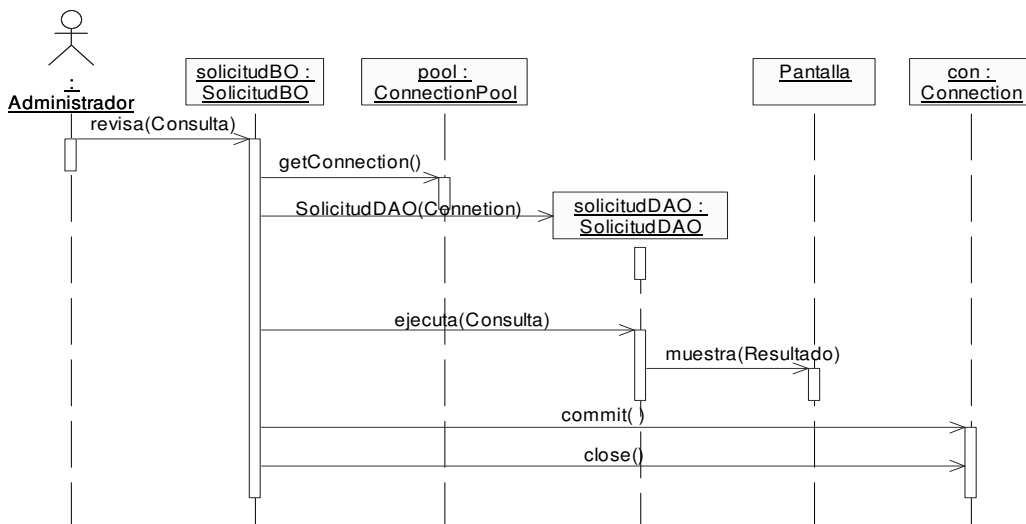


Fig.4.74 Diagrama de Secuencia del caso de uso Revisar Solicitudes.

El *Administrador* invoca al método *revisar* de *SolicitudBO*, consigue una conexión por medio de *getConnection* del objeto *pool*. Se realiza la búsqueda determinada y se muestra en *Pantalla*. Finalmente se realiza la transacción y cierra la conexión con el método *close*.

**Caso de uso Solicitar** (fig. 4.75)

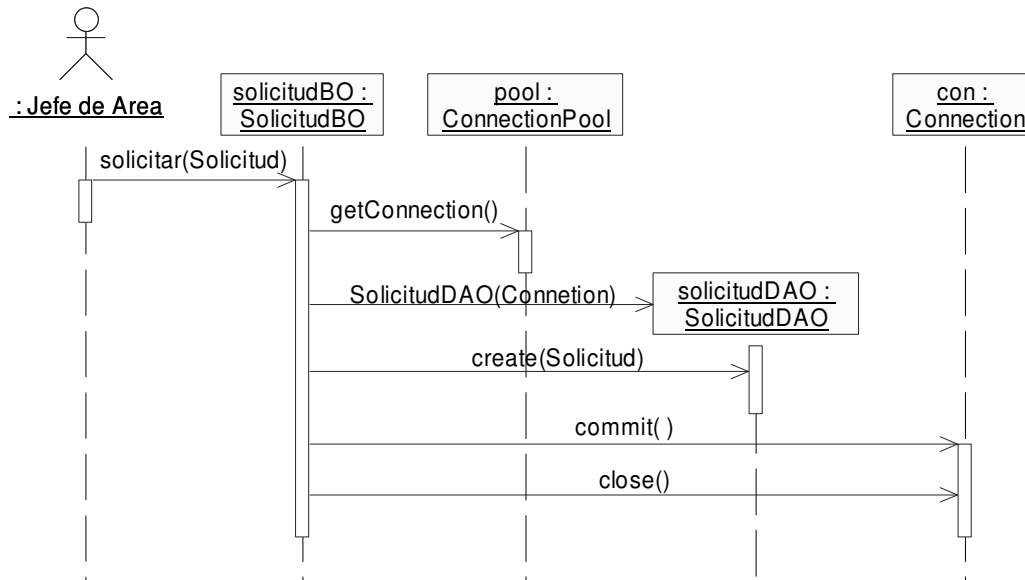


Fig.4.75 Diagrama de Secuencia del caso de uso solicitar.

El *Jefe de Área* interactúa con el sistema por medio del objeto *solicitaBO*, invocando al método *solicitar* al cual recibe como parámetro una *solicitud*. Se consigue una conexión mediante *getConnection* de *pool*, y se crea el objeto *solicitudDAO*; se invoca el método *create*. Finalmente se realiza la transacción con *commit* del objeto *con* y cierra la conexión con el método *close*.

**4.9 Diseño de Clases**

Diseñar una clase es crear una clase de diseño que cumpla su papel en la realización de los casos de uso incluyendo los siguientes aspectos:

- Sus operaciones.
- Sus atributos.
- Sus relaciones como asociaciones, agregaciones y generalizaciones.
- Sus métodos.
- Sus estados.

El diseño de las clases depende del tipo de las clases de análisis de donde proceden.

- Clases interfaz: Su diseño depende mucho de la tecnología que se vaya a emplear.
- Clases Entidad: Su diseño depende de si es necesario emplear tecnología para gestionar una base de Datos.
- Clases Control: Son las más delicadas. Su diseño depende de aspectos tales como que si los objetos están distribuidos en diferentes nodos, o si es posible mezclar sus funcionalidades con otras clases Interfaz o Entidad.

#### **4.9.1 Identificar Operaciones**

Esto se realiza:

- Identificando las responsabilidades de la clase de análisis de la que deriva.
- Identificando los mensajes a los que debe responder en el diagrama de secuencia.

- Contemplando los requisitos especiales de la clase de análisis de la que deriva.
- Inspeccionando las realizaciones de casos de uso de diseño en que participan.
- Añadiendo la visibilidad de cada operación.
- Usando la sintaxis del lenguaje de implementación a utilizar.

#### **4.9.2 Identificar Atributos**

Se debe considerar:

- Los atributos deben ser los requeridos para realizar sus operaciones.
- Hay que tener en cuenta los atributos obtenidos en la fase de Análisis.
- Los tipos de atributos se restringen a los tipos disponibles en el lenguaje de programación a usar.

#### **4.9.3 Identificar Asociaciones, Agregaciones y Generalizaciones**

Consiste en la identificación de las relaciones que existen entre las clases del diseño.

#### **4.9.4 Describir Métodos**

Los métodos se pueden especificar durante el diseño describiéndolos como los algoritmos de las operaciones que realizan para ello se utilizan diagramas de estado.

### **4.10 Diseño de las Clases del Sistema de Inventario**

## Capítulo 4. Análisis y Diseño

Los siguientes diagramas muestran las clases del diseño con cada uno de sus métodos y atributos que serán transcritos directamente al lenguaje de programación.



Fig. 4.76 Clases de entidad.



## Capítulo 4. Análisis y Diseño

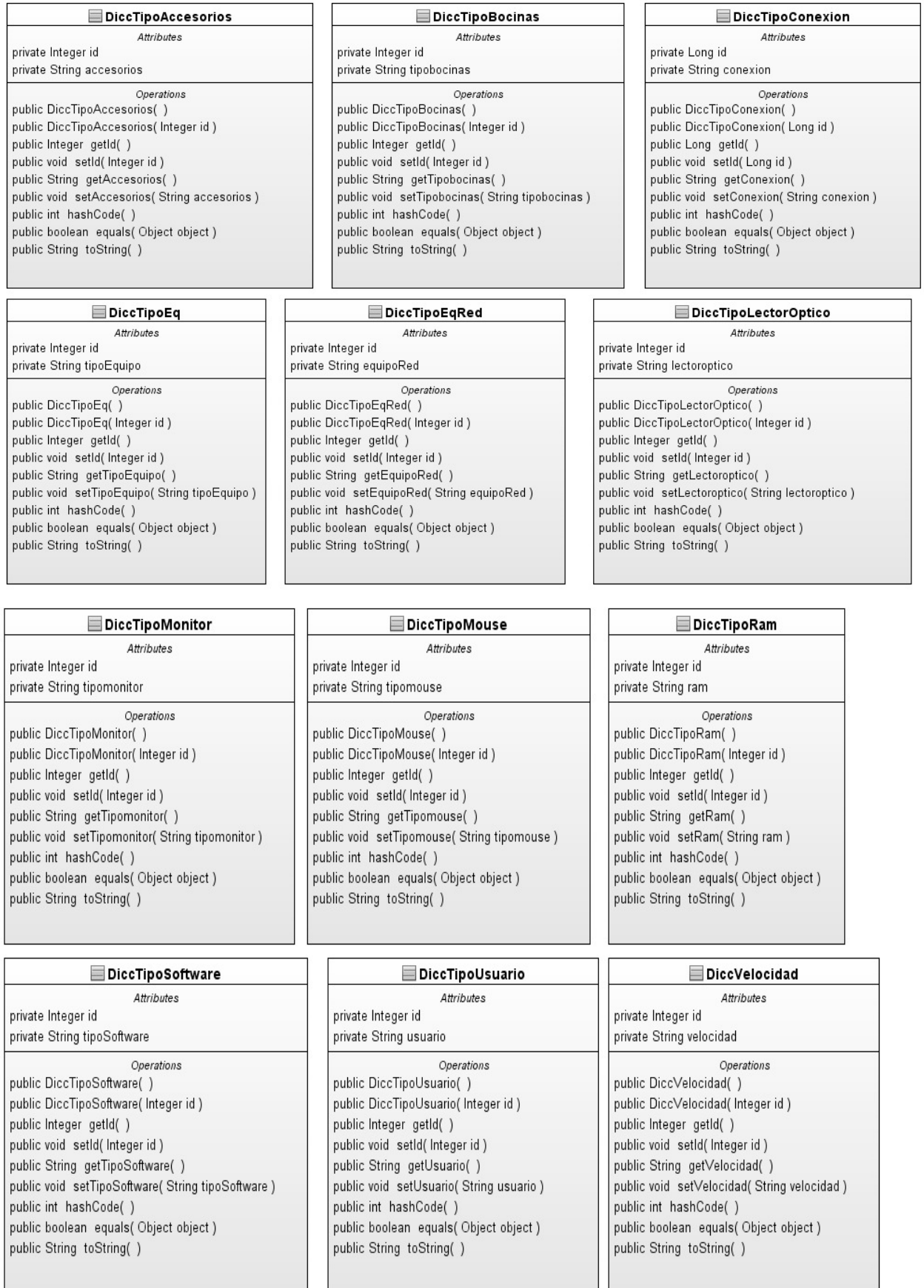


Fig 1.77 Clases de entidad(continuación).

## Capítulo 4. Análisis y Diseño

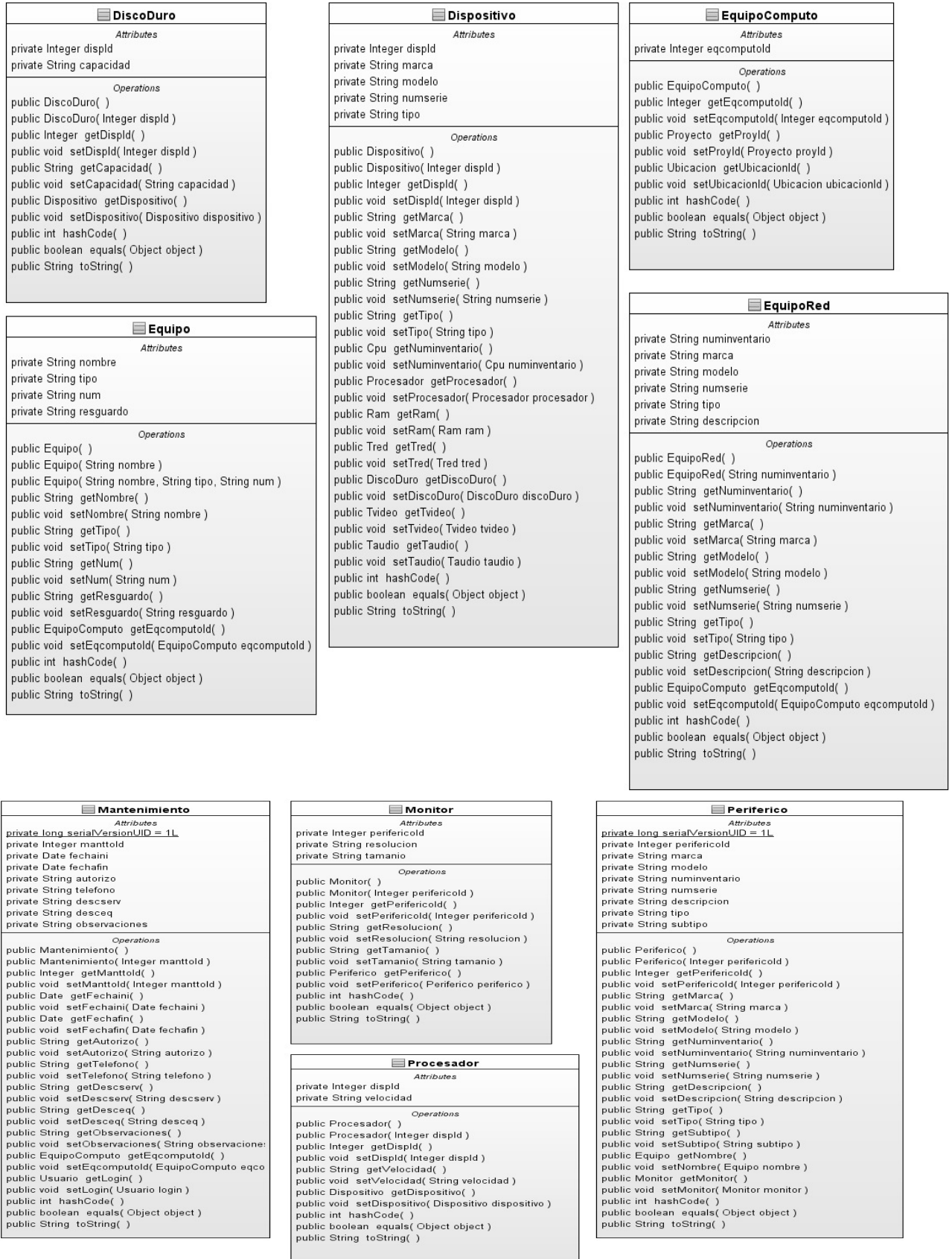


Fig. 4.78. Clases de entidad(continuación).

## Capítulo 4. Análisis y Diseño



Fig. 4.79 Clases de entidad (continuación).



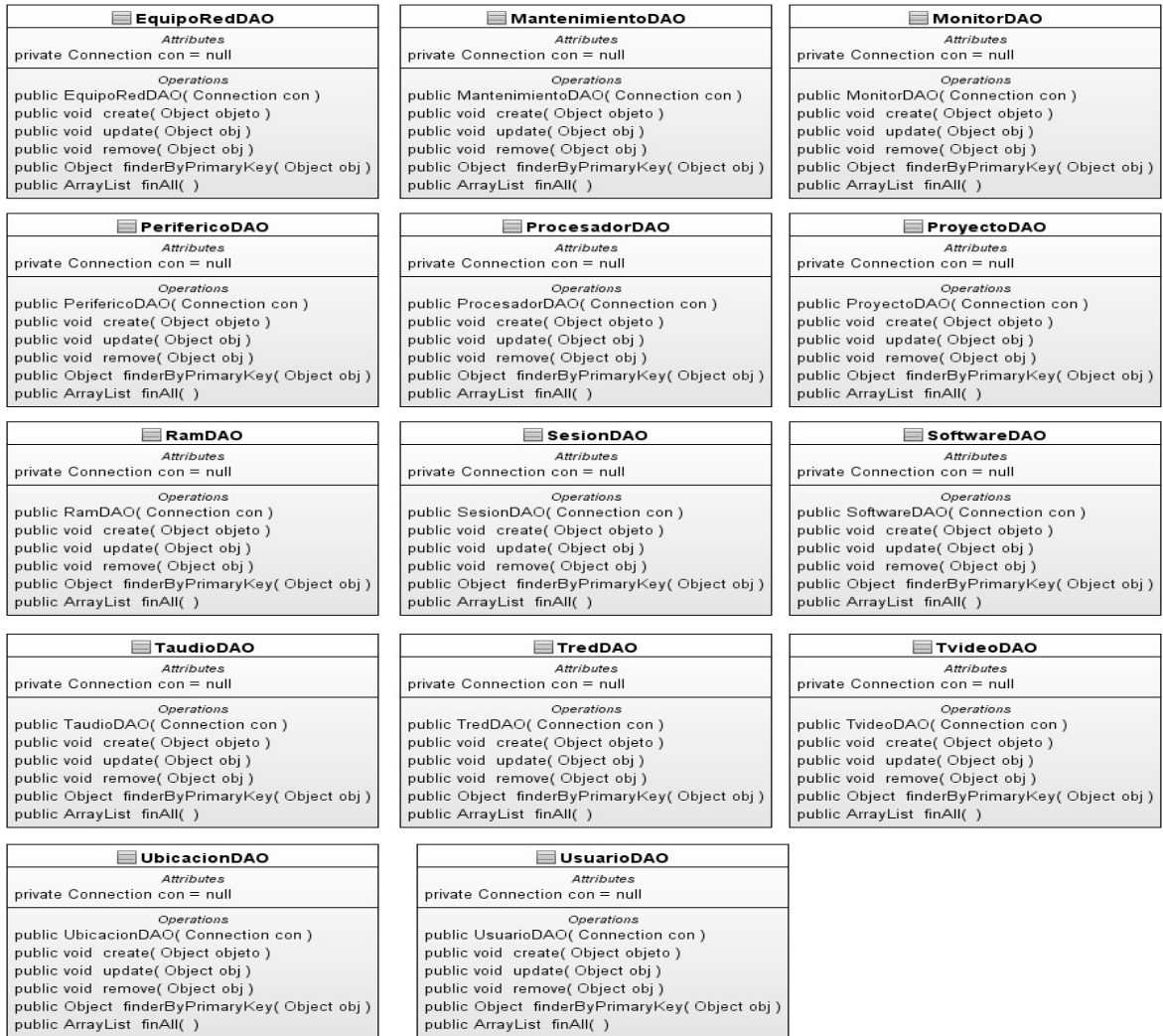


Fig 4.81 Clases de Acceso a Datos(contionuación).

### Clases de Objetos de Negocio (BO)

Estas clases nos permiten y se encargan de la interacción que tiene el usuario con el sistema ya que representan los procesos del negocio directamente, se encargan de la creación, actualización y borrado de objetos sin preocuparse del trasfondo que conlleva a ello como lo es la conexión a la base de datos, ya que esa actividad la realizan las clases DAO.

Las figuras 4.82 y 4.83 nos muestran mediante componentes UML la representación de cada una de las clases del negocio junto con sus métodos.

## Capítulo 4. Análisis y Diseño

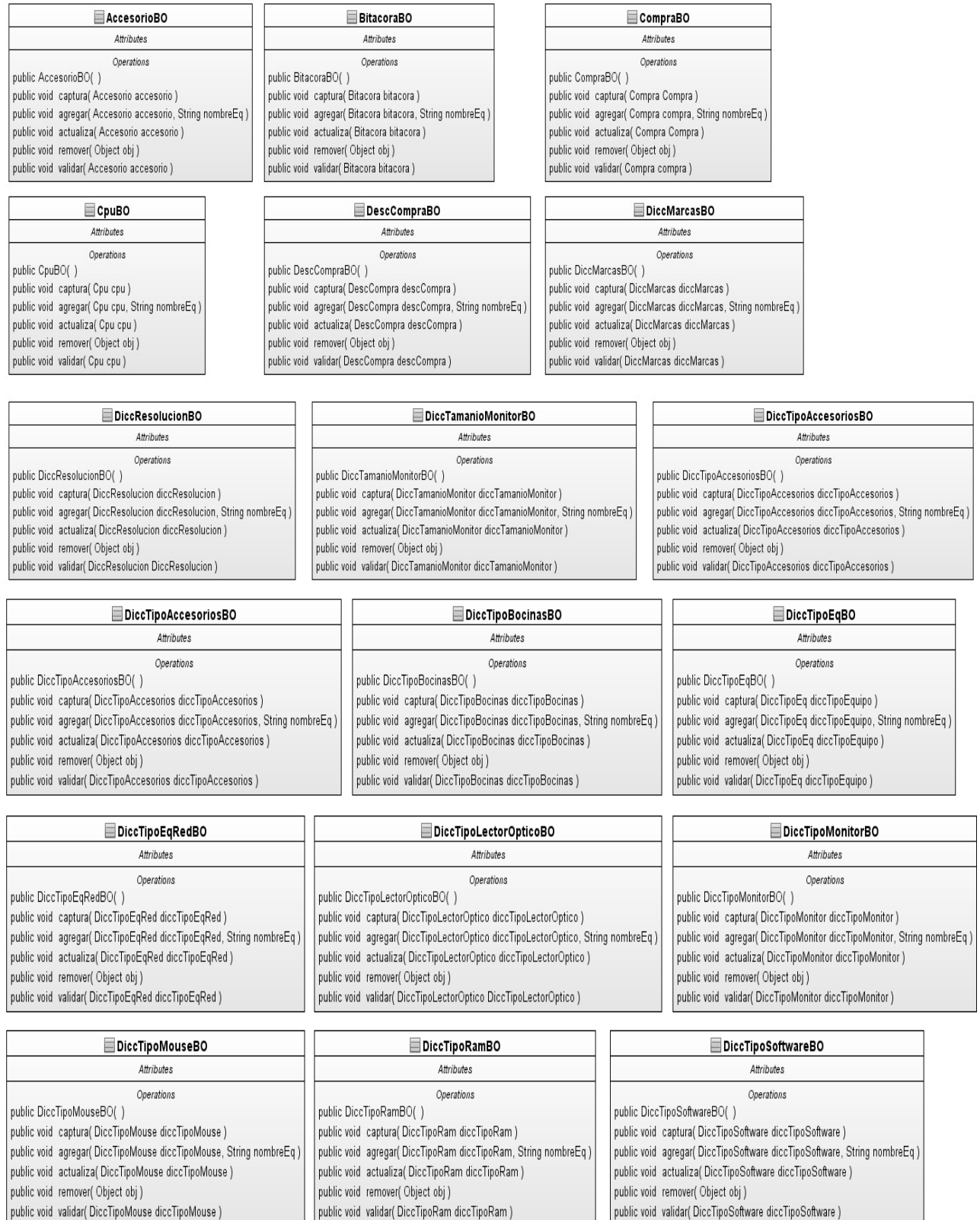


Fig. 4.82 Clases del Negocio(BO).

## Capítulo 4. Análisis y Diseño

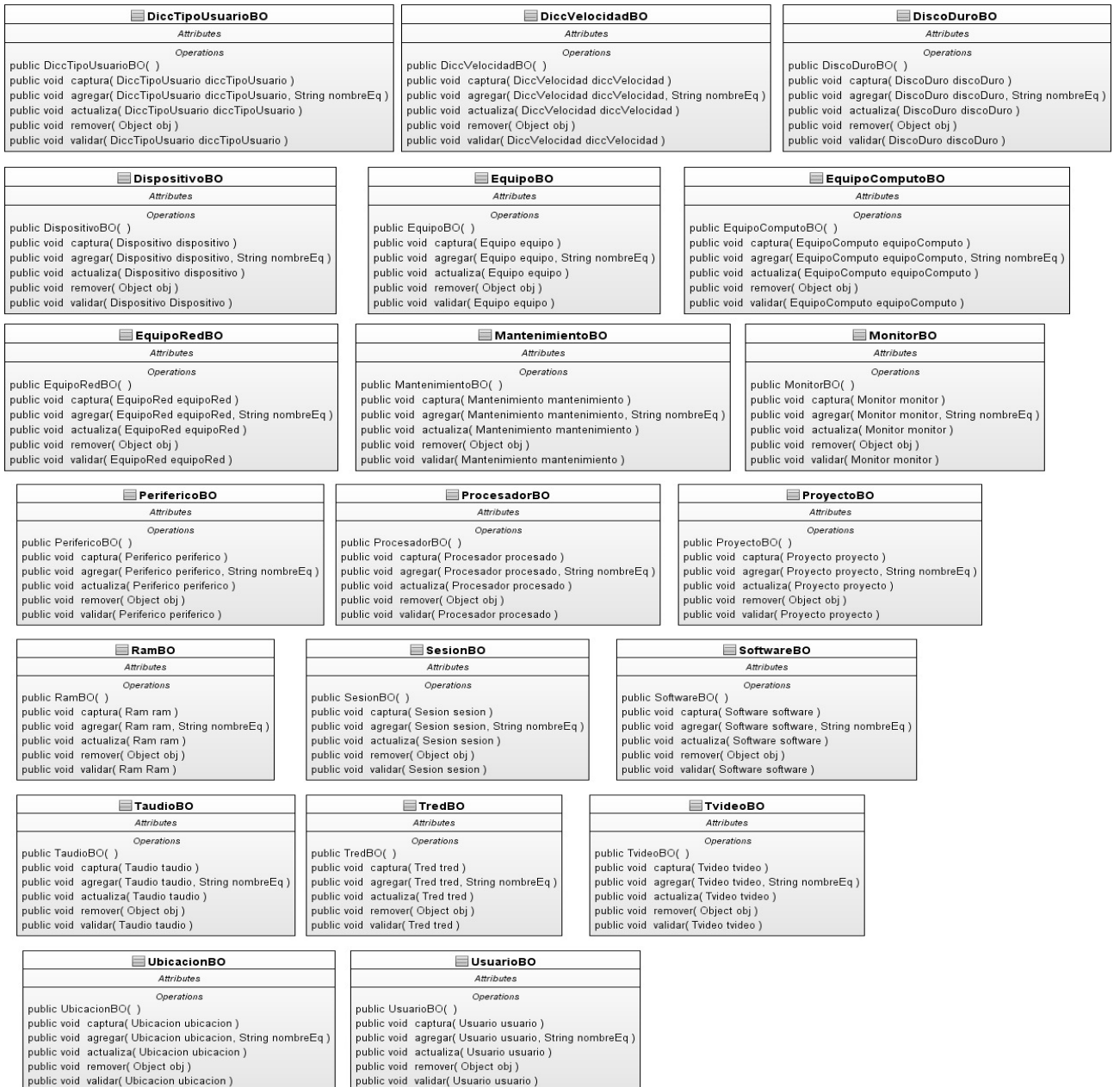
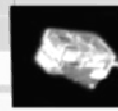


Fig. 4.83 Clases del Negocio (continuación).



Sistema de  
inventario



# *Capítulo 5*

## *Implementación y Prueba*

---



## 5.1 Modelo de Implementación

El modelo de implementación describe el sistema mediante clases que se implementan en términos de componentes como ficheros de código fuente, ejecutables, etc.

También describe la organización de acuerdo a la modularización y estructura disponible en el lenguaje de programación (fig. 5.1).

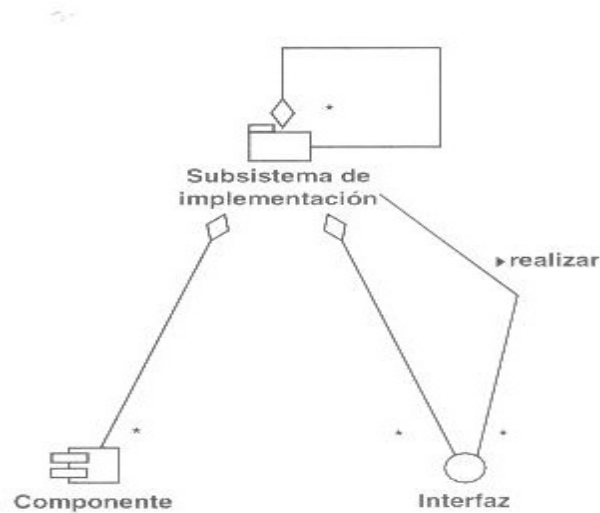


Fig. 5.1 Diagrama del modelo de Implementación.

### 5.1 Implementación de la Arquitectura

Consiste en identificar los componentes significativos para la arquitectura y los nodos en donde se distribuyen.

#### 5.1.1 Componente

Empaquetamiento físico de los elementos del modelo del diseño pueden ser: ejecutables, file, library, table, document. (fig. 5.2).

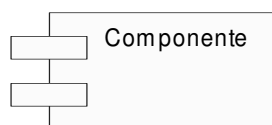


Fig. 5.2 Representación UML de un componente.

### 5.1.2 Identificar los Componentes de la Arquitectura

Esta actividad consiste en definir cuales componentes serán implementados para definir la línea Base de la Arquitectura.

### 5.1.3 Identificar Componentes Ejecutables y Asignación de éstos a Nodos

Se identifica en que nodos deben ejecutarse los componentes significativos para la arquitectura, es decir, las clases activas encontradas en el diseño serán asignadas a un componente ejecutable, denotando así un proceso, que se ejecutará en el nodo asignado.

## 5.2 Implementación de la Arquitectura del Sistema de Inventario

### 5.2.1 Identificar los Componentes Significativos para la Arquitectura

Las interfaces de usuario se ejecutarán en el nodo cliente, al igual que las clases relacionadas con ellas como son: las clases proporcionadas por la librería “ICEFaces” (fig 5.3).

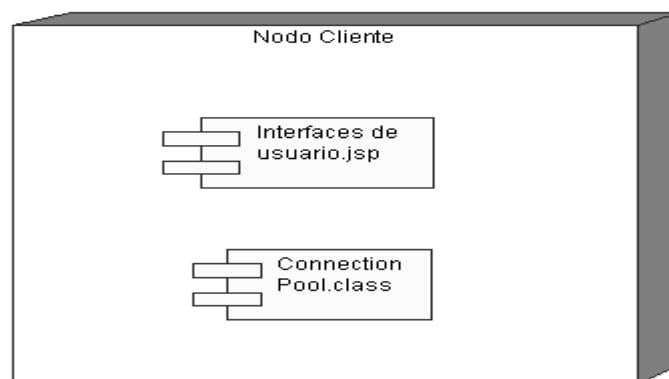


Fig. 5.3 Clases activas que se ejecutarán en el nodo cliente.

Los componentes que se van a ejecutar en el nodo servidor son los generados por las clases BO (Business Object, objetos del negocio), las clases

DAO (Data Access Object, Objeto de acceso a datos) y el controlador para la conexión a las base de datos. (fig. 5.4).

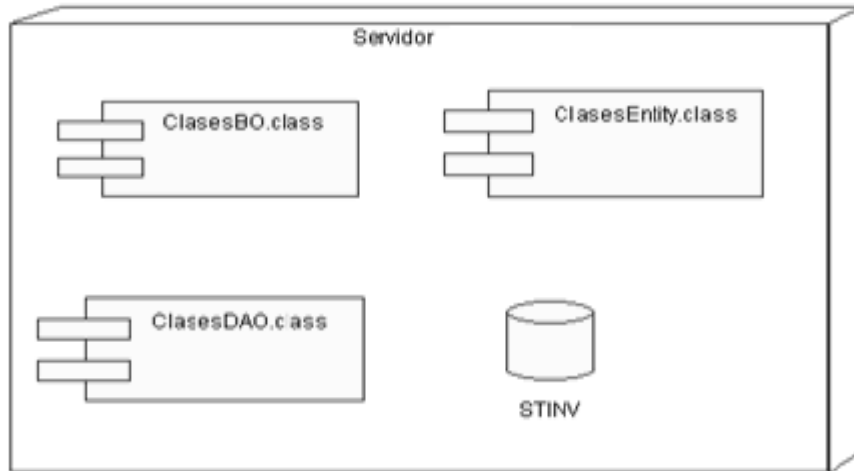


Fig. 5.4 Componentes en el nodo Servidor.

### 5.3 Implementar Clases

Implementar una clase consiste en escribir la clase en un componente fichero, es decir, se implementan sus operaciones generando código fuente. Cada operación corresponde a un método nuevo en la clase.

### 5.4 Implementar Clases del Sistema de Inventario

Cada una de las clases de diseño serán implementadas en archivos .java que en el momento de ser compilados generaran los archivos .class que son los ejecutables en el lenguaje java.

Las interfaces de usuario serán implementadas en archivos .jsp que al ser compilados generan archivos ejecutables en el navegador y en la maquina virtual de java.

La forma en que se comportan los componentes en el caso de uso Captura Equipo (fig. 5.5).

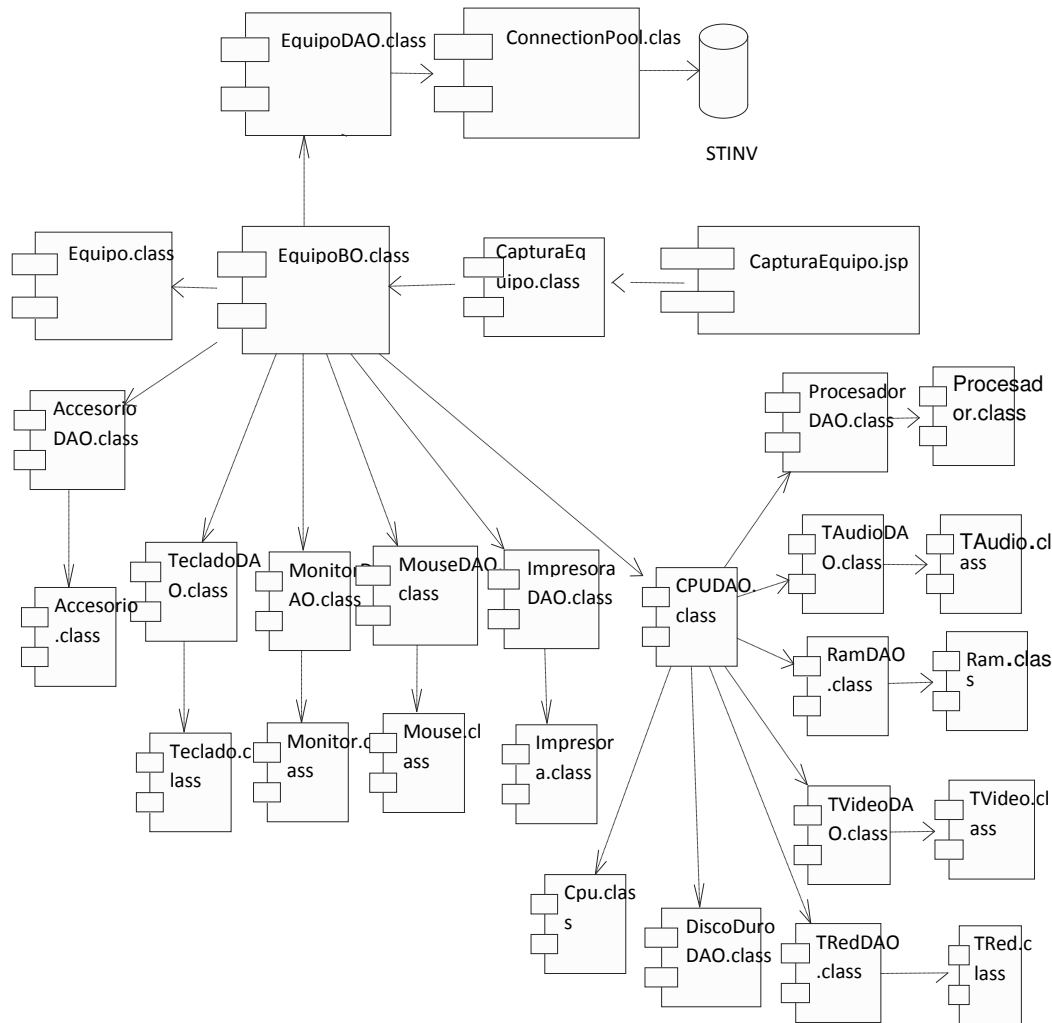


Fig. 5.5 Componentes que participan en el caso de uso Captura Equipo.

## 5.5 Integrar el Sistema

Sus principales objetivos son:

- Crear un plan de integración de los componentes en el sistema.
- Integrar cada componente antes de que sea sometido a las pruebas de integración.

La integración se hará de modo incremental y en cada paso se deberán incluir pocos componentes.

## 5.6 Integrar el Sistema de Inventario

Las construcciones se planearon de la siguiente manera:

- Primera Construcción, en esta construcción se implementa el módulo Registra Equipo, que está compuesto por los siguientes casos de uso:
  - *Captura Equipo.*
  - *Captura Equipo de Red.*
  - *Agrega Accesorio.*
  - *Agrega CPU.*
  - *Agrega Dispositivo.*
  - *Agrega Periférico.*
  - *Agrega Software.*
- Segunda Construcción, esta construcción está enfocada a la implementación del módulo Actualiza Equipo, que está compuesto por los siguientes casos de uso:
  - *Modificar Accesorio.*
  - *Modificar CPU.*
  - *Modificar Equipo.*
  - *Modificar Equipo de Red.*
  - *Modificar Periférico.*
  - *Quita Accesorio.*
  - *Quita CPU.*
  - *Quita Periférico.*
  - *Quita Software.*
- Tercera Construcción, esta construcción estará enfocada a la implementación de la parte administrativa del sistema, y constará de los siguientes casos de uso:
  - *Alta Usuario.*
  - *Baja Usuario.*
  - *Consultar Bitácora.*
  - *Generar Censo.*
  - *Revisar Solicitudes.*

- Cuarta Construcción, esta construcción pertenece a la segunda parte del módulo del caso de administración.
  - *Baja Accesorio.*
  - *Baja CPU.*
  - *Baja Equipo.*
  - *Baja Equipo Red.*
  - *Baja Periférico.*
  - *Baja Usuario.*
- Quinta Construcción, esta construcción se enfoca a la implementación del módulo de solicitudes, que abarca los siguientes casos de uso:
  - *Solicitar Compra.*
  - *Solicitar Mantenimiento.*
  - *Solicitar Baja.*
- Sexta Construcción: esta construcción está enfocada al módulo de mantenimiento, que está constituido por los siguientes casos de uso:
  - *Captura Capacidad.*
  - *Captura Marca.*
  - *Captura Proyecto.*
  - *Captura Resolución.*
  - *Captura Tamaño Monitor.*
  - *Captura Tipo Conexión.*
  - *Captura Tipo Equipo.*
  - *Captura Tipo Equipo Red.*
  - *Captura Tipo Lector Óptico.*
  - *Captura Tipo Monitor.*
  - *Captura Tipo Mouse.*
  - *Captura Tipo RAM.*
  - *Captura Tipo Software.*
  - *Captura Velocidad.*
  - *Captura Ubicación.*

## 5.7 Modelo de Prueba

Describe como se prueban los componentes ejecutables en el modelo de implementación con pruebas de implementación y del sistema también como han de ser probados aspectos específicos del sistema (fig. 5.6).

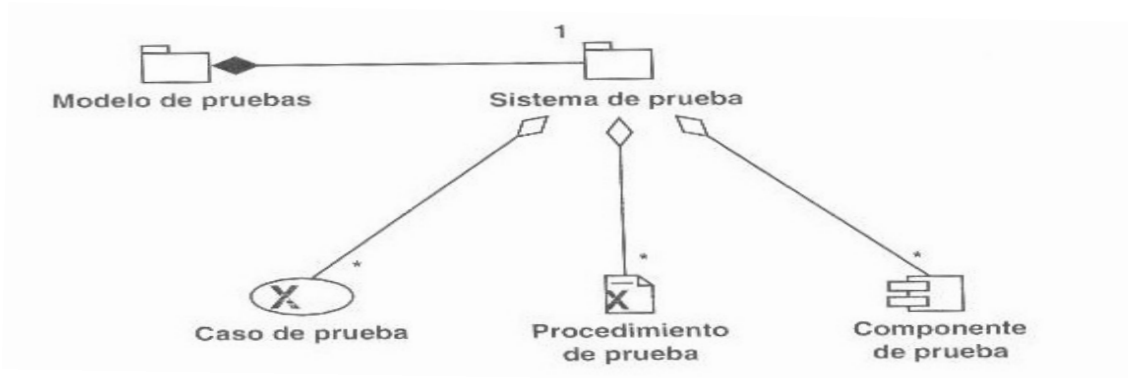


Fig. 5.6 Modelo de Prueba.

## 5.8 Planificar y Diseñar Pruebas

En este paso se planifican y diseñan pruebas, así como el orden en que se van a llevar a cabo.

### 5.8.1 Caso de Prueba

Un caso de uso especifica una forma de probar el sistema, incluyendo la entrada o resultado con el que se ha de probar y las condiciones bajo la que se ha de probar (fig. 5.7).



Fig. 5.7 Símbolo Caso de Prueba en UML.

## 5.9 Casos de Prueba del Sistema de Inventario

### Agregar

#### 1. Descripción

Conjunto de pruebas realizadas sobre el Caso de Uso “Agregar Accesorio”.

Las pruebas realizadas a este caso de uso son:

- *Agregar Accesorios* después de que haya sido solicitado por el *Encargado de Inventario*.
- *Agregar Accesorios* asignando valores nulos.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

#### 2. Agregar Accesorio después de que haya sido solicitado por el Encargado de Inventario.

##### 2.1 Descripción

Nos introducimos en el sistema como *Encargado e Inventario*, accediendo a su funcionalidad y solicitamos Agregar Accesorio, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los accesorios existentes hasta ese momento. Seleccionaremos en el tipo de accesorio que agregar, llenamos los espacios en blanco y damos clic en el botón de aceptar.

##### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

##### 2.3 Entrada

- Introducimos ‘Candy’ en el campo correspondiente a Equipo.
- Se desplegará una pantalla con el tipo de Accesorio que se desea agregar a dicho equipo y la cantidad de cada uno.
- Seleccionamos Impresora.
- Introducimos 1 en el campo cantidad correspondiente a impresora.
- Se desplegará otra pantalla correspondiente a las características de la impresora.
- Introducimos ‘0001’ en el campo No. Serie del accesorio.



- Introducimos 'HP' en el campo Marca del accesorio.
- Introducimos '0001' en el campo Modelo del accesorio.
- Introducimos 'Láser' en el campo Tipo.
- El *Encargado de Inventario* pulsa el botón de 'Enviar' de la aplicación.

## 2.4 Resultado esperado

El accesorio queda almacenado en la base de datos del sistema.

## 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Agregar Accesorios asignando valores nulos

### 3.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Agregar Accesorios*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los existentes hasta ese momento. Seleccionaremos el tipo de accesorio que agregar, llenamos los espacios en blanco con algunas cantidades nulas para observar cómo responde el sistema y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario "*Encargado de inventario*" está dado de alta en la base de datos.

### 3.3 Entrada

- Introducimos 'Candy' en el campo correspondiente a Equipo.
- Se desplegará una pantalla con el tipo de Accesorio que se desea agregar a dicho equipo y la cantidad de cada uno.
- Seleccionamos Impresora.
- Introducimos 1 en el campo cantidad correspondiente a impresora.
- Se desplegará otra pantalla correspondiente a las características de la impresora.
- Introducimos '0000' en el campo No. Serie del accesorio.
- Introducimos 'HP' en el campo Marca del accesorio.
- Introducimos '0001' en el campo Modelo del accesorio.
- Seleccionamos 'Láser' de la lista de Tipo.
- El *Encargado de Inventario* pulsa el botón de 'Enviar' de la

aplicación.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Agregar Dispositivo

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Agregar Dispositivo*”.

Las pruebas realizadas a este caso de uso son:

- *Agregar Dispositivo* después de que haya sido solicitado por el *Encargado de Inventario*.
- *Agregar Dispositivo* asignando valores nulos.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

## 2. Agregar Dispositivos después de que haya sido solicitado por el *Encargado de Inventario*

### 2.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Agregar Dispositivo*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los dispositivos existentes hasta ese momento. Seleccionaremos en el tipo de dispositivo que agregar, llenamos los espacios en blanco y damos clic en el botón de aceptar.

### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

## 2.3 Entrada

- Introducimos 'Candy' en el campo correspondiente a Equipo.
- Se desplegará una pantalla con el tipo de Dispositivo que se desea agregar a dicho equipo y la cantidad de cada uno.
- Seleccionamos Tarjeta de Red.
- Introducimos 1 en el campo cantidad correspondiente a tarjeta de red.
- Se desplegará otra pantalla correspondiente a las características de la tarjeta de red.
- Introducimos '0011' en el campo No. Serie del dispositivo.
- Introducimos " en el campo Marca del dispositivo.
- Introducimos '0001' en el campo Modelo del dispositivo.
- Introducimos '132.248.54.111' en el campo MAC.
- Introducimos '10 Mbits' en el campo velocidad.
- Introducimos " en el campo Tipo.
- El *Encargado de Inventario* pulsa el botón de 'Enviar' de la aplicación.

## 2.4 Resultado esperado

El dispositivo queda almacenado en la base de datos del sistema.

## 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Agregar Dispositivo asignando valores nulos

### 3.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Encargado de Inventario*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los existentes hasta ese momento. Seleccionaremos el tipo de Dispositivo que agregar, llenamos los espacios en blanco con algunas cantidades nulas para observar como responde el sistema y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario "*Encargado de Inventario*" está dado de alta en la base de datos.

### 3.3 Entrada

- Introducimos 'Candy' en el campo correspondiente a Equipo.
- Se desplegará una pantalla con el tipo de Dispositivo que se desea agregar a dicho equipo y la cantidad de cada uno.
- Seleccionamos Tarjeta de Red.
- Introducimos 1 en el campo cantidad correspondiente a tarjeta de red.
- Se desplegará otra pantalla correspondiente a las características de la tarjeta de red.
- Introducimos '0000' en el campo No. Serie del dispositivo.
- Introducimos " en el campo Marca del dispositivo.
- Introducimos '0001' en el campo Modelo del dispositivo.
- Introducimos '132.248.54.111' en el campo MAC.
- Introducimos '10 Mbits' en el campo velocidad.
- Introducimos " en el campo Tipo.
- El *Encargado de Inventario* pulsa el botón de 'Enviar' de la aplicación.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Alta Usuario

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso "Alta Usuario".

Las pruebas realizadas a este caso de uso son:

- *Alta Usuario* después de que haya sido solicitado por el *Administrador*.
- *Alta Usuario* asignando valores nulos.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

## 2. Alta Usuario después de que haya sido solicitado por el Administrador

### 2.1 Descripción

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Alta Usuario*, el sistema nos mostrará unos apartados con los campos, en dónde introduciremos los datos del usuario que deseamos capturar y damos clic en el botón de aceptar.

### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como "*Administrador*".

### 2.3 Entrada

- Introducimos 'Dulce María' en el campo correspondiente a Nombre(s).
- Introducimos 'Sánchez' en el campo correspondiente a Apellido Paterno.
- Introducimos 'Faustino' en el campo correspondiente a Apellido Materno.
- Introducimos '1' en el campo correspondiente a Tipo.
- Introducimos 'dulcemsf23' en el campo correspondiente a Login.
- Introducimos 'xxxxxxxx' en el campo correspondiente a Contraseña.
- Introducimos 'xxxxxxxx' en el campo correspondiente a la confirmación de contraseña.
- El *Administrador* pulsa el botón de 'Enviar' de la aplicación.

### 2.4 Resultado esperado

El *Usuario* queda almacenado en la base de datos del sistema.

### 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Alta Usuario asignando valores nulos

### 3.1 Descripción

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Alta Usuario*, el sistema nos mostrará unos

apartados con los campos, en dónde introduciremos los datos del usuario que deseamos capturar y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario “Administrador” está dado de alta en la base de datos.

### 3.3 Entrada

- Introducimos ‘Dulce María’ en el campo correspondiente a Nombre(s).
- Introducimos ‘Sánchez’ en el campo correspondiente a Apellido Paterno.
- Introducimos ‘00000000’ en el campo correspondiente a Apellido Materno.
- Introducimos ‘1’ en el campo correspondiente a Tipo.
- Introducimos ‘dulcemsf23’ en el campo correspondiente a Login.
- Introducimos ‘xxxxxxxx’ en el campo correspondiente a Contraseña.
- Introducimos ‘xxxxxxxx’ en el campo correspondiente a la confirmación de contraseña.
- El Administrador pulsa el botón de ‘Enviar’ de la aplicación.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Baja

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Baja Accesorio*”.

Las pruebas realizadas a este caso de uso son:

- Dar de baja accesorios después de que haya sido solicitado por el *Administrador*.
- Dar de baja un accesorio asignando valores nulos.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

## **2. Dar de baja un Accesorio después de que haya sido solicitado por el Administrador**

### **2.1 Descripción**

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Dar de baja un Accesorio*, el sistema nos mostrará unos apartados con las características particulares de cada uno de los accesorios existentes hasta ese momento. Seleccionaremos en el tipo de accesorio que deseamos dar de baja, y damos clic en el botón de aceptar.

### **2.2 Condiciones de ejecución**

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como "*Administrador*".

### **2.3 Entrada**

- Seleccionamos de una lista el accesorio que se desea dar de baja.
- Elegimos de los diferentes accesorios mostrados el que deseamos dar de baja.
- Y damos clic en el botón 'Dar de Baja'.
- El sistema muestra un mensaje de confirmación.
- Y damos clic en botón aceptar.

### **2.4 Resultado esperado**

El accesorio es mandado a una tabla de bajas.

### **2.5 Evaluación de la Prueba**

Prueba superada con éxito.

## **3. Dar de baja un Accesorio asignando valores nulos**

### **3.1 Descripción**

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Dar de baja un Accesorio*, el sistema nos mostrará unos apartados con las características particulares de cada uno

de los accesorios existentes hasta ese momento. Seleccionaremos en el tipo de accesorio que deseamos dar de baja, y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario “Administrador” está dado de alta en la base de datos.

### 3.3 Entrada

- Seleccionamos de una lista el accesorio que se desea dar de baja.
- No elegimos ningún accesorio.
- Y damos clic en el botón ‘Dar de Baja’.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Baja Usuario

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Baja Usuario”.

Las pruebas realizadas a este caso de uso son:

- *Baja Usuario* después de que haya sido solicitado por el Encargado de Inventario.
- *Baja Usuario* asignando valores nulos.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación



## 2. Baja Usuario después de que haya sido solicitado por el Administrador

### 2.1 Descripción

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Baja Usuario*, el sistema nos mostrará unos apartados con los campos, en dónde introduciremos los datos del usuario que deseamos dar de baja y damos clic en el botón de aceptar.

### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como "*Administrador*".

### 2.3 Entrada

- Seleccionamos el Login de la persona que deseamos dar de baja del sistema, en este caso de prueba es 'dulcemsf23'.
- Inmediatamente el sistema nos muestra el nombre y apellidos de la persona que deseamos dar de baja, para confirmar que sea la correcta.
- El sistema nos mostrará en el campo Nombre(s) 'Dulce María'.
- El sistema nos mostrará en el campo Apellido Paterno 'Sánchez'.
- El sistema nos mostrará en el campo Apellido Materno 'Faustino'.
- El *Administrador* pulsa el botón de 'Borrar' de la aplicación.

### 2.4 Resultado esperado

El *Usuario* ha sido borrado de la base de datos del sistema.

### 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Baja Usuario asignando valores nulos

### 3.1 Descripción

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Baja Usuario*, el sistema nos mostrará unos apartados con los campos, en dónde introduciremos los datos del usuario que deseamos dar de baja y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario “Administrador” está dado de Alta en la base de datos.

### 3.3 Entrada

- Seleccionamos el Login de la persona que deseamos dar de baja del sistema, en este caso de prueba es ‘000000000’.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Capturar

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Captura Accesorio”.

Las pruebas realizadas a este caso de uso son:

- *Captura accesorios* después de que haya sido solicitado por el Encargado de Inventario.
- *Captura accesorios* asignando valores nulos.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

### 2. Captura Accesorio después de que haya sido solicitado por el Encargado de Inventario

#### 2.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Captura Accesorio*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los accesorios existentes hasta ese momento. Seleccionaremos

en el tipo de accesorio que deseamos capturar, llenamos los espacios en blanco y damos clic en el botón de aceptar.

## 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

## 2.3 Entrada

- Se desplegará una pantalla con el tipo de accesorios que se desea capturar y la cantidad de cada uno.
- Seleccionamos Impresora.
- Introducimos 1 en el campo cantidad correspondiente a impresora.
- Se desplegará otra pantalla correspondiente a las características de la impresora.
- Introducimos ‘0001’ en el campo No. Serie del accesorio.
- Introducimos ‘HP’ en el campo Marca del accesorio.
- Introducimos ‘0001’ en el campo Modelo del accesorio.
- Introducimos ‘Láser’ en el campo Tipo.
- El *Encargado de Inventario* pulsa el botón de ‘Enviar’ de la aplicación.

## 2.4 Resultado esperado

El accesorio queda almacenado en la base de datos del sistema.

## 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Captura Accesorios asignando valores nulos

### 3.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Captura Accesorio*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los accesorios existentes hasta ese momento. Seleccionaremos en el tipo de accesorio que deseamos capturar, llenamos los espacios en blanco y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

### 3.3 Entrada

- Se desplegará una pantalla con el tipo de accesorios que se desea capturar y la cantidad de cada uno.
- Seleccionamos Impresora.
- Introducimos 1 en el campo cantidad correspondiente a impresora.
- Se desplegará otra pantalla correspondiente a las características de la impresora.
- Introducimos ‘0000’ en el campo No. Serie del accesorio.
- Introducimos ‘HP’ en el campo Marca del accesorio.
- Introducimos ‘0001’ en el campo Modelo del accesorio.
- Introducimos ‘Láser’ en el campo Tipo.
- El *Encargado de Inventario* pulsa el botón de ‘Enviar’ de la aplicación.

## Captura Equipo

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Captura Equipo*”.

Las pruebas realizadas a este caso de uso son:

- *Captura Equipo* después de que haya sido solicitado por el *Encargado de Inventario*.
- *Captura Equipo* asignando valores nulos.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

## 2. Captura Equipo después de que haya sido solicitado por el Encargado de Inventario

### 2.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Captura Equipo*, el sistema nos mostrará unos apartados con las características que podemos capturar de la

Equipo completo. Llenamos los espacios en blanco y damos clic en el botón de aceptar.

## 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

## 2.3 Entrada

- Se desplegará una pantalla con las características de equipo que deseamos capturar.
- Introducimos ‘Candy’ en el campo nombre de Equipo.
- Introducimos ‘2’ en el campo número de Equipo.
- Introducimos “” en el campo número de inventario.
- Introducimos “” en el campo número de serie.
- Introducimos ‘HP’ en el campo Marca.
- Introducimos ‘Pavilion DV1000’ en el campo modelo.
- El *Encargado de Inventario* da clic en el botón ‘Continuar’ de la aplicación.
- Se desplegará otra pantalla correspondiente a los dispositivos que contiene ese Equipo.
- Introducimos
- Introducimos ‘0001’ en el campo No. Serie del accesorio.
- Introducimos ‘HP’ en el campo Marca del accesorio.
- Introducimos ‘0001’ en el campo Modelo del accesorio.
- Introducimos ‘Láser’ en el campo Tipo.
- El *Encargado de Inventario* pulsa el botón de ‘Enviar’ de la aplicación.

## 2.4 Resultado esperado

El Equipo queda almacenado en la base de datos del sistema.

## 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Captura Equipo asignando valores nulos

### 3.1 Descripción

Nos introducimos en el sistema como *Encargado de Mantenimiento*, accediendo a su funcionalidad y solicitamos *Captura Equipo*, el sistema nos

mostrará unos apartados con las características que podemos capturar de la Equipo. Llenamos los espacios en blanco y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Mantenimiento*”.

### 3.3 Entrada

- Se desplegará una pantalla con los campos de la Equipo que se desea capturar.
- Introducimos '00 GB' en el campo Equipo.
- El *Encargado de Mantenimiento* pulsa el botón de 'Enviar' de la aplicación.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Cerrar Sesión

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Cerrar Sesión*”.

Las pruebas realizadas a este caso de uso son:

- *Cerrar Sesión* después de que haya sido solicitado por algún usuario.
- *Cerrar Sesión* solo cerrando la ventana de aplicación.

El entorno del cual partiremos para realizar la prueba será cualquier formulario de la aplicación

## 2. Cerrar Sesión después de que haya sido solicitado por algún usuario

### 2.1 Descripción

Nos introducimos en el sistema no importando el usuario, para este caso de uso y solicitamos *Cerrar Sesión* y damos clic en el botón de aceptar.

### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el *Usuario* que acceda a este apartado del sistema este dado de alta en él.

### 2.3 Entrada

- Al iniciar el sistema, se desplegará una pantalla dependiendo del *Usuario* y la funcionalidad que solicitemos.
- Damos clic en el botón 'Cerrar Sesión'
- El sistema nos muestra un mensaje de confirmación del cierre de la sesión. Advirtiéndonos que no guardará nada de los cambios realizados hasta que se de clic en el botón de Enviar o Aceptar, según sea el caso.
- El *Usuario* pulsa en el botón de 'Aceptar' del mensaje de confirmación.
- Se realiza el cierre de la sesión.

### 2.4 Resultado esperado

La sesión ha sido cerrada.

### 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Cerrar Sesión solo cerrando la ventana de la aplicación

### 3.1 Descripción

Nos introducimos en el sistema no importando el *Usuario*, para este caso de uso y solicitamos *Cerrar Sesión* y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el *Usuario* que acceda a este apartado del sistema este dado de alta en él.

### 3.3 Entrada

- Al iniciar el sistema, se desplegará una pantalla dependiendo del usuario y la funcionalidad que solicitemos.
- Damos clic en el botón [x] que aparece del lado derecho y arriba para cada ventana de la aplicación.
- El sistema cierra automáticamente la sesión.

### 3.4 Resultado esperado

El sistema cierra la sesión sin guardar cambios realizados en la sesión

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Consultar Bitácora

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Consultar Bitácora*”.

Las pruebas realizadas a este caso de uso son:

- *Consultar Bitácora* seleccionando un criterio de búsqueda
- *Consultar Bitácora* sin elegir una opción para la consulta de la bitácora.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

### 2. Consultar Bitácora seleccionando un criterio de búsqueda

#### 2.1 Descripción

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Consultar Bitácora*, el sistema nos mostrará unos apartados con las opciones existentes para la consulta de la bitácora.

#### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Administrador*”.



### 2.3 Entrada

- Se desplegará una pantalla con las opciones para generar la bitácora.
- Seleccionamos consultar una bitácora por fecha.
- El *Administrador* pulsa el botón de 'Aceptar' de la aplicación.
- Se desplegará una pantalla con la descripción de la bitácora que el administrador haya elegido.
- Si se desea enviar esta información a imprimir se da clic en 'Exportar'.

### 2.4 Resultado esperado

Se muestra la información de la bitácora seleccionada.

### 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Consultar Bitácora sin elegir una opción para la consulta de la Bitácora

### 3.1 Descripción

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Consultar Bitácora*, el sistema nos mostrará unos apartados con las opciones existentes para la consulta de la bitácora.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como "*Administrador*".

### 3.3 Entrada

- Se desplegará una pantalla con las opciones para generar la bitácora.
- Seleccionamos consultar una bitácora por fecha.
- El *Administrador* pulsa el botón de 'Aceptar' de la aplicación.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que se debe elegir al menos una opción para la consulta de la bitácora.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Generar Censo

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Generar Censo*”.

Las pruebas realizadas a este caso de uso son:

- *Generar Censo* seleccionando un criterio de búsqueda
- *Generar Censo* sin elegir un valor para la generación del censo.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

### 2. Generar Censo seleccionando un criterio de búsqueda

#### 2.1 Descripción

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Generar Censo*, el sistema nos mostrará unos apartados con las opciones existentes para la generación del censo.

#### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Administrador*”.

#### 2.3 Entrada

- Se desplegará una pantalla con las opciones para la generación del censo.
- Seleccionamos generar un censo por ‘Proyecto’.
- El *Administrador* pulsa el botón de ‘Aceptar’ de la aplicación.
- Se desplegará una pantalla con la descripción del censo que el

administrador haya elegido.

- Si se desea enviar esta información a imprimir se da clic en 'Exportar'.

## 2.4 Resultado esperado

Se muestra la información de la generación del censo.

## 2.5 Evaluación de la Prueba

Prueba superada con éxito.

# 3. Generar Censo sin elegir una opción para la consulta del Censo

## 3.1 Descripción

Nos introducimos en el sistema como *Administrador*, accediendo a su funcionalidad y solicitamos *Generar Censo*, el sistema nos mostrará unos apartados con las opciones existentes para la generación del censo.

## 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como "*Administrador*".

## 3.3 Entrada

- Se desplegará una pantalla con las opciones para la generación del censo.
- No seleccionamos nada en las opciones para la generación del censo.
- El *Administrador* pulsa el botón de 'Aceptar' de la aplicación.

## 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que se debe elegir al menos una opción para la generación del censo.

## 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Inicia Sesión

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Inicia Sesión*”.

Las pruebas realizadas a este caso de uso son:

- *Iniciar Sesión* con datos de usuario correctos
- *Iniciar sesión* asignando caracteres especiales.
- *Iniciar sesión* asignando valores nulos.
- Aceptar sin haber asignado valores.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

### 2. Iniciar Sesión con datos de usuario correctos

#### 2.1 Descripción

Nos debemos poder introducir en el sistema con un nombre de *Usuario* y contraseña válidos, accediendo a su funcionalidad. Llenamos los espacios en blanco y damos clic en el botón de aceptar.

#### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el *Usuario* tenga un nombre de *Usuario* y una contraseña con la que pueda acceder al sistema.

#### 2.3 Entrada

- Introducimos ‘Candy’ en el campo correspondiente a usuario.
- Introducimos ‘pumita23’ en el campo correspondiente a contraseña.
- El *Usuario* pulsa el botón de ‘Enviar’ de la aplicación.

#### 2.4 Resultado esperado

El usuario ha podido acceder al sistema.

#### 2.5 Evaluación de la Prueba

Prueba superada con éxito.

### 3. Iniciar sesión asignando caracteres especiales

#### 3.1 Descripción

Nos debemos poder introducir en el sistema con un nombre de *Usuario* y contraseña válidos, accediendo a su funcionalidad. Llenamos los espacios en blanco y damos clic en el botón de aceptar.

#### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el *Usuario* tenga un nombre de *Usuario* y una contraseña con la que pueda acceder al sistema.

#### 3.3 Entrada

- Introducimos "%&/&/(&/&' en el campo correspondiente a *Usuario*.
- Introducimos 'pumita23' en el campo correspondiente a contraseña.
- El *Usuario* pulsa el botón de 'Enviar' de la aplicación.

#### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no es posible introducir caracteres especiales y pasará el foco en el campo en el que se encuentre dicho error.

#### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

### 4. Iniciar Sesión asignando valores nulos

#### 4.1 Descripción

Nos debemos poder introducir en el sistema con un nombre de *Usuario* y contraseña válidos, accediendo a su funcionalidad. Llenamos los espacios en blanco y damos clic en el botón de aceptar.

#### 4.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el *Usuario* tenga un nombre de *Usuario* y una contraseña con la que pueda acceder al sistema.

#### 4.3 Entrada

- Introducimos '0000000' en el campo correspondiente a *Usuario*.
- Introducimos 'pumita23' en el campo correspondiente a contraseña.
- El *Usuario* pulsa el botón de 'Enviar' de la aplicación.

#### 4.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

#### 4.5 Evaluación de la Prueba

Prueba superada con éxito.

### 5. Aceptar sin asignar valores

#### 5.1 Descripción

Nos debemos poder introducir en el sistema con un nombre de *Usuario* y contraseña válidos, accediendo a su funcionalidad. Llenamos los espacios en blanco y damos clic en el botón de aceptar.

#### 5.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario tenga un nombre de *Usuario* y una contraseña con la que pueda acceder al sistema.

#### 5.3 Entrada

- No introducimos nada en el campo correspondiente a *Usuario*.
- Introducimos 'pumita23' en el campo correspondiente a contraseña.
- El *Usuario* pulsa el botón de 'Enviar' de la aplicación.

#### 5.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se debe dejar el nombre de usuario en blanco, y pasará el foco en el campo en el que se encuentre dicho error.

## 5.5 Evaluación de la Prueba

Prueba superada con éxito.

### Modificar

#### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Modificar Accesorio*”.

Las pruebas realizadas a este caso de uso son:

- *Modificar accesorios* después de que haya sido solicitado por el *Encargado de Inventario*.
- *Modificar accesorios* asignando valores nulos.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

#### 2. Modificar Accesorio después de que haya sido solicitado por el Encargado de Inventario

##### 2.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Modificar Accesorio*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los accesorios existentes hasta ese momento. Seleccionaremos en el tipo de accesorio que se desea modificar, y damos clic en el botón de aceptar.

##### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

##### 2.3 Entrada

- Se despliega una pantalla con las características del accesorio que deseamos modificar.

- Seleccionamos Impresora.
- Se desplegará otra pantalla correspondiente a las características de la impresora seleccionada, incluyendo la ubicación del equipo con el que fue capturado y en la cuál es el único campo en el que se puede realizar alguna modificación.
- El sistema nos muestra la ubicación 'USECAD, Soporte Técnico'.
- El *Encargado de Inventario* modifica a 'Usecad, Becarios'
- El *Encargado de Inventario* pulsa el botón de 'Enviar' de la aplicación.

## 2.4 Resultado esperado

Los datos del accesorio han sido modificados y guardados en la base de datos del sistema.

## 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Modificar Accesorios asignando valores nulos

### 3.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Modificar Accesorio*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los accesorios existentes hasta ese momento. Seleccionaremos en el tipo de accesorio que se desea modificar, y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como "*Encargado de Inventario*".

### 3.3 Entrada

- Se despliega una pantalla con las características del accesorio que deseamos modificar.
- Seleccionamos Impresora.
- Se desplegará otra pantalla correspondiente a las características de la impresora seleccionada, incluyendo la ubicación del equipo con el que fue capturado y en la cuál es el único campo en el que se puede



realizar alguna modificación.

- El sistema nos muestra la ubicación ‘USECAD, Soporte Técnico’.
- El *Encargado de Inventario* modifica a ‘0000000’
- El *Encargado de Inventario* pulsa el botón de ‘Enviar’ de la aplicación.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Modificar Equipo

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso “*Modificar Equipo*”.

Las pruebas realizadas a este caso de uso son:

- *Modificar un Equipo* con datos correctos
- Aceptar sin haber asignado valores.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

### 2. Modificar Equipo después de que haya sido solicitado por el Encargado de Inventario

#### 2.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Modificar Equipo*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los Equipo existentes hasta ese momento. Seleccionaremos en el tipo de Equipo que se desea modificar, y damos clic en el botón de aceptar.

#### 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

### 2.3 Entrada

- Se despliega una pantalla correspondiente a las características del Equipo seleccionado, esto quiere decir los componentes (dispositivos o periféricos) que se le pueden agregar o quitar, así como su ubicación.
- Seleccionamos Mouse.
- Se muestran los datos correspondientes a ese Mouse.
- El *Encargado de Inventario* da clic en el botón ‘Quitar’.
- Se muestra un mensaje de confirmación.
- El *Encargado de Inventario* pulsa el botón de ‘Aceptar’ para salir de la aplicación.

### 2.4 Resultado esperado

Los datos del Equipo han sido modificados y guardados en la base de datos del sistema.

### 2.5 Evaluación de la Prueba

Prueba superada con éxito.

## 3. Aceptar sin asignar valores

### 3.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Modificar Equipo*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los Equipo existentes hasta ese momento. Seleccionaremos en el tipo de Equipo que se desea modificar, y damos clic en el botón de aceptar.

### 3.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

### 3.3 Entrada

- Se despliega una pantalla correspondiente a las características del Equipo seleccionado, esto quiere decir los componentes (dispositivos o periféricos) que se le pueden agregar o quitar, así como su ubicación.
- No seleccionamos nada.
- El *Encargado de Inventario* da clic en el botón 'Quitar'.

### 3.4 Resultado esperado

El sistema nos muestra un mensaje de error advirtiéndonos de que no se pueden asignar valores nulos, y pasará el foco en el campo en el que se encuentre dicho error.

### 3.5 Evaluación de la Prueba

Prueba superada con éxito.

## Quitar

### 1. Descripción

Este documento cubre el conjunto de pruebas realizadas sobre el Caso de Uso "Quitar CPU".

Las pruebas realizadas a este caso de uso son:

- *Quitar CPU* después de que haya sido solicitado por el *Encargado de Inventario*.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación

### 2. Quitar CPU después de que haya sido solicitado por el Encargado de Inventario

#### 2.1 Descripción

Nos introducimos en el sistema como *Encargado de Inventario*, accediendo a su funcionalidad y solicitamos *Quitar CPU*, el sistema nos mostrará unos apartados con los nombres y características particulares de cada uno de los CPU existentes hasta ese momento. Seleccionaremos en el tipo de CPU que quitar, llenamos los espacios en blanco y damos clic en el botón de aceptar.

## 2.2 Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el usuario que acceda a este apartado del sistema este dado de alta en él como “*Encargado de Inventario*”.

## 2.3 Entrada

- Introducimos ‘Candy’ en el campo correspondiente a Equipo.
- Se desplegará una pantalla con los datos del CPU que se desea quitar a dicho equipo.
- El *Encargado de Inventario* da clic en el botón ‘Quitar’.
- Se muestra un mensaje de confirmación.
- El *Encargado de Inventario* pulsa el botón de ‘Aceptar’ de la aplicación.

## 2.4 Resultado esperado

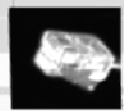
El CPU se ha quitado del equipo, esto quiere decir que se ha eliminado la relación entre el CPU seleccionado y el equipo.

## 2.5 Evaluación de la Prueba

Prueba superada con éxito.



Sistema de  
inventario



# *Conclusiones*

*Conclusiones*

---

### **Conclusiones.**

La construcción del sistema de información para el sistema de inventario de cómputo de la Secretaría de Servicios Académicos, comprueba las ventajas de los sistemas informáticos sobre los procesos manuales. Estas ventajas se mencionan en la siguiente lista:

- Mantener la seguridad de la información.
- Evitar la redundancia de datos.
- Mantener la consistencia de datos.
- Mantener la concurrencia de la información.
- Adecuado control de acceso.
- Mantener la integridad de la información.
- Mantener la confidencialidad.
- Agilizar el proceso de búsqueda de información de la base de datos.
- Facilitar la modificación de los datos
- Presentar a los usuarios una interfaz gráfica amigable que facilite el manejo de la información.
- Presentar reportes de los movimientos en este sistema de inventario, indicando la fecha de su realización, así como el nombre del usuario y el movimiento que se realizó.
- Preparar el reporte solicitado por DGSCA a la Secretaría de Servicios Académicos de la Facultad de Ingeniería sobre los censos del equipo de cómputo que se deben entregar anualmente.

El Proceso Unificado de Desarrollo de Software (RUP, Rational Unified Process) es una metodología poderosa dado que es un proceso iterativo e incremental, lo cual permitió acercarse a la solución final dado que mediante los flujos de trabajo que define, se fueron transformando los requisitos de usuario hasta lograr obtener un software confiable y utilizable.

## *Conclusiones*

De igual forma el Lenguaje Unificado de Modelado (UML) ha contribuido a una mejor comprensión de los requisitos, análisis, diseño, implementación y prueba del Sistema de Inventario; al representar cada una de estas etapas de forma gráfica, mediante diagramas; lo que amplía la forma de conceptualizar el sistema.

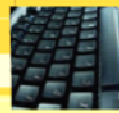
La utilización de patrones de diseño permitieron definir de una forma más clara y entendible la estructura interna del sistema.

Tanto el lenguaje de programación como el servidor web y de base de datos son tecnologías de la más alta calidad. Además de brindar características de seguridad y confiabilidad el lenguaje de programación brinda portabilidad permitiendo que el sistema sea compatible en distintas plataformas y al ser un sistema web hace que ello sea una cualidad deseable.

La utilización de sistema de inventario de cómputo facilitará las tareas para las personas encargadas del manejo de los equipos, debido al incremento y actualización de los equipos, ya que el sistema les permitirá contar con una base sólida de información, la cual podrá ser actualizada reflejando así los cambios de que sufren los equipos, de esta misma manera agilizará los procesos de solicitudes de equipos nuevos, el mantenimiento administrado a los equipos, así como la baja del inventario de equipos obsoletos.



Sistema de  
inventario



*Bibliografía*

---



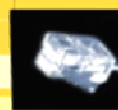
## Bibliografía

- Castaño, Adoración de Miguel.  
“Fundamentos y modelos de bases de datos”  
1ª. Ed. Madrid, España: Alfaomega, 1997
- Rodríguez, Luis Ángel.  
“Seguridad de la información en sistemas de computo”  
1ª Ed. México: Ventura Ediciones S. A. de C. V. 1995
- Tsai, Alice.  
“Sistemas de base de datos: administración y uso”  
1ª Ed. México: Prentice Hall Hispanoamericana. 1990.
- Falkner, Jayso,  
“Fundamentos Desarrollo Web con JSP”  
1ª Ed. Madrid, España: Ediciones Anaya Multimedia. 2002.
- Pfleeger, Shari Lawrence.  
“Ingeniería de Software, teoría y práctica”  
1ª Ed. Buenos Aires: Pearson Education, 2002.
- Alonso Amo, Fernando.  
“Introducción a al ingeniería del software: Modelo de desarrollo de programas”  
1ª Ed. Madrid España: Delta, Publicaciones Universitarias. 2005.
- Bruegge, Bernd.  
“Ingeniería de software orientado a objetos”  
1ª Ed. México :Pearson Educación.2002.

- Booch, Grady.  
“Proceso Unificado de desarrollo de software”  
1ª Ed. Madrid, España: Pearson Educación 2000.
  
- Craig Larman  
“UML y patrones: introduccion al analisis y diseño orientado a objetos”  
2ª. México. Pearson, 2003



Sistema de  
inventario



*Anexos*  
*Diccionario de Datos*

---

TABLA: STINV\_ACCESORIO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NUMERO DE INVENTARIO DEL ACCESORIO	VARCHAR	20	NOT NULL	*	
NO_SERIE	NUMERO DE SERIE DEL ACCESORIO	VARCHAR	20	NOT NULL		
MARCA	MARCA DEL ACCESORIO	VARCHAR	20	NOT NULL		
MODELO	MODELO DEL ACCESORIO	VARCHAR	20	NOT NULL		
TIPO	TIPO DEL ACCESORIO	VARCHAR	15	NOT NULL		
DESCRIPCION	DESCRIPCION DEL ACCESORIO	VARCHAR	255	NULL		
ESTADO	ESTADO EN EL QUE SE ENCUANTRA EL ACCESORIO	CHAR	1	NOT NULL		
CVE_LOCALIZACION	CLAVE DE LA LOCALIZACIÓN	VARCHAR	10	NOT NULL		*
CLV_PROY	CLAVE DEL PROYECTO	VARCHAR	5	NOT NULL		*

TABLA: STINV\_ALTA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_BITACORA	IDENTIFICADOR DE LA TABLA STINV_BITACORA	VARCHAR	10	NOT NULL	*	
NOMBRE	NOMBRE DE LA BITACORA	VARCHAR	20	NOT NULL		

TABLA: STINV\_AREA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
CLV_AREA	CLAVE DE ÀREA	VARCHAR	4	NOT NULL	*	
AREA	NOMBRE DEL ÀREA	VARCHAR	40	NOT NULL		
TELEFONO	TELÈFONO DEL ÀREA	VARCHAR	15	NOT NULL		

TABLA: STINV\_BAJA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
FOLIO	NUMERO DE FOLIO DE LA BAJA	VARCHAR	10	NOT NULL	*	
FECHA	FECHA EN LA QUE SE LEVANTÒ LA BAJA	DATE	-	NOT NULL		
NO_INVENTARIO	NÚMERO DE INVENTARIO DEL EQUIPO QUE SE VA A DAR DE BAJA	VARCHAR	20	NOT NULL		*

TABLA: STINV\_BITACORA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_BITACORA	ID DE LA TABLA STINV_BITACORA	VARCHAR	10	NOT NULL	*	
FECHA	FECHA DE LA REALIZACIÓN DEL INVENTARIO	DATE	-	NOT NULL		
DESCRIPCION	DESCRIPCIÓN DEL MOVIMIENTO QUE FUE HECHO A ALGÚN EQUIPO	VARCHAR	255	NULL		

TABLA: STINV\_COMPRA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
FOLIO	NUMERO DE FOLIO DE LA SOLICITUD DE COMPRA	VARCHAR	10	NOT NULL	*	
ANEXOS	COTIZACIONES, CATÁLOGOS, JUSTIFICACIÓN, OTROS	VARCHAR	5	NOT NULL		
OBSERVACIONES	LAS OBSERVACIONES QUE SE HACEN A LA COMPRA	VARCHAR	255	NOT NULL		
COSTO	COSTO TOTAL DE LO QUE SE VA A SOLICITAR	VARCHAR	30	NOT NULL		
FECHA	FECHA DE SOLICITU DE LA COMPRA	DATE	-	NOT NULL		
AUTORIZACION	NOMBRE DE LA PERSONA QUE AUTORIZÓ LA COMPRA	VARCHAR	4	NOT NULL		*
SUFICIENCIA	NOMBRE DE LA PERSONA ENCARGADA DEL DEPTO. DE SUFICIENCIA	VARCHAR	4	NOT NULL		*
SECRETARIO	NOMBRE DEL SECRETARIO	VARCHAR	4	NOT NULL		*
RESPONSABLE	NOMBRE DEL RESPONSABLE DE LA COMPRA	VARCHAR	4	NOT NULL		*
CLV_AREA	CLAVE DE ÁREA	VACRCHAR	4	NOT NULL		*

TABLA: STINV\_CPU

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NUMERO DE INVENTARIO DEL CPU	VARCHAR	20	NOT NULL	*	
NO_SERIE	NUMERO DE SERIE DEL CPU	VARCHAR	20	NOT NULL		
MARCA	MARCA DEL CPU	VARCHAR	20	NOT NULL		
MODELO	MODELO DEL CPU	VARCHAR	20	NOT NULL		
ESTADO	ESTADO EN EL QUE SE ENCUANTRA EL CPU	CHAR	1	NOT NULL		

CLV_PROY	CLAVE DEL PROYECTO	VARCHAR	5	NOT NULL		*
CVE_LOCALIZACION	CLAVE DE LA LOCALIZACIÓN	VARCHAR	10	NOT NULL		*

TABLA: STINV\_DEPENDENCIA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
CLV_DEPENDENCIA	CLAVE DE LA DEPENDENCIA	VARCHAR	10	NOT NULL	*	
DEPENDENCIA	NOMBRE DE LA DEPENDENCIA	VARCHAR	40	NOT NULL		
CLV_AREA	CLAVE DEL ÁREA	VARCHAR	4	NOT NULL		*

TABLA: STINV\_DESC\_COMPRA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
FOLIO	NUMERO DE FOLIO DE LA SOLICITUD DE COMPRA	VARCHAR	10	NOT NULL	*	
NUM	ELEMENTOS QUE SE VAN A SOLICITAR	INTEGER	-	NOT NULL		
DESCRIPCION	DESCRIPCIÓN DE LO QUE SE VA A SOLICITAR	VARCHAR	255	NOT NULL		
UNIDAD	UNIDAD DE LO QUE SE SOLICITA	VARCHAR	10	NOT NULL		
CANTIDAD	CANTIDAD DE LOS ELEMENTOS QUE SE VAN A SOLICITAR	INTEGER	-	NOT NULL		
PRECIO	PRECIO DE LO QUE SE SOLICITA	REAL	-	NOT NULL		
IMPORTE	IMPORTE DE LO QUE SE SOLICITA	REAL	-	NOT NULL		

TABLA: STINV\_DICC\_CAPACIDAD

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DE LA CAPACIDAD	VARCHAR	10	NOT NULL	*	
CAPACIDAD	CAPACIDAD EN DISCO DURO, RAM, ETC	VARCHAR	15	NOT NULL		

TABLA: STINV\_DICC\_MARCAS

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DE LA MARCA	VARCHAR	10	NOT NULL	*	
MARCA	NOMBRE DE LA MARCA	VARCHAR	50	NOT NULL		

TABLA: STINV\_DICC\_RESOLUCION

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DE LA RESOLUCION	VARCHAR	10	NOT NULL	*	
RESOLUCION	RESOLUCIÓN DE LA PANTALLA DEL MONITOR	VARCHAR	20	NOT NULL		

TABLA: STINV\_DICC\_TAMANIO\_MONITOR

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DEL TAMAÑO DEL MONITOR	VARCHAR	10	NOT NULL	*	
TAMANIOMONITOR	DESCRIPCIÓN DEL TAMAÑO DEL MONITOR	VARCHAR	10	NOT NULL		

TABLA: STINV\_DICC\_TIPO\_ACCESORIOS

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DE LOS ACCESORIOS	VARCHAR	10	NOT NULL	*	
ACCESORIOS	DESCRIPCIÓN DEL TIPO DE ACCESORIOS	VARCHAR	30	NOT NULL		

TABLA: STINV\_DICC\_TIPO\_BOCINAS

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DE LAS BOCINAS	VARCHAR	10	NOT NULL	*	
TIPO_BOCINAS	DESCRIPCIÓN DEL TIPO DE BOCINAS	VARCHAR	30	NOT NULL		

TABLA: STINV\_DICC\_TIPOCONEXION

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DE LA CONEXIÓN	VARCHAR	10	NOT NULL	*	
CONEXIÓN	DESCRIPCIÓN DEL TIPO DE CONEXIÓN	VARCHAR	20	NOT NULL		

TABLA: STINV\_DICC\_TIPOEQ

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DEL EQUIPO	VARCHAR	10	NOT NULL	*	
TIPO_EQUIPO	DESCRIPCIÓN DEL TIPO DEL EQUIPO	VARCHAR	30	NOT NULL		

TABLA: STINV\_DICC\_TIPOEQRED

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DEL EQUIPO DE RED	VARCHAR	10	NOT NULL	*	
EQUIPO_RED	DESCRIPCIÓN DEL TIPO DE EQUIPO DE RED	VARCHAR	30	NOT NULL		

TABLA: STINV\_DICC\_TIPOLECTOROPTICO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DEL LECTOR OPTICO	VARCHAR	10	NOT NULL	*	
LECTOROPTICO	DESCRIPCIÓN DEL TIPO DE LECTOR OPTICO	VARCHAR	15	NOT NULL		

TABLA: STINV\_DICC\_TIPOMONITOR

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DEL MONITOR	VARCHAR	10	NOT NULL	*	
TIPOMONITOR	DESCRIPCIÓN DEL TIPO DE MONITOR	VARCHAR	30	NOT NULL		

TABLA: STINV\_DICC\_TIPOMOUSE

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DEL MOUSE	VARCHAR	10	NOT NULL	*	
TIPOMOUSE	DESCRIPCIÓN DEL TIPO DE MOUSE	VARCHAR	30	NOT NULL		

TABLA: STINV\_DICC\_TIPORAM

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DE LA RAM	VARCHAR	10	NOT NULL	*	
RAM	DESCRIPCIÓN DEL TIPO DE RAM	VARCHAR	20	NOT NULL		

TABLA: STINV\_DICC\_TIPOSOFTWARE

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DEL SOFTWARE	VARCHAR	10	NOT NULL	*	
TIPO_SOFTWARE	DESCRIPCIÓN DEL TIPO DE SOFTWARE	VARCHAR	100	NOT NULL		



TABLA: STINV\_DICC\_TIPOUSUARIO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DEL USUARIO	VARCHAR	10	NOT NULL	*	
USUARIO	DESCRIPCIÓN DEL TIPO DE USUARIO	VARCHAR	5	NOT NULL		

TABLA: STINV\_DICC\_VELOCIDAD

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID	IDENTIFICADOR DE LA VELOCIDAD	VARCHAR	10	NOT NULL	*	
VELOCIDAD	DESCRIPCIÓN DE LA VELOCIDAD	VARCHAR	15	NOT NULL		

TABLA: STINV\_DISCODURO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_DISCO_DURO	IDENTIFICADOR DE LA TABLA STINV_DISCODURO	VARCHAR	20	NOT NULL	*	
MARCA	MARCA DEL DISCO DURO	VARCHAR	20	NOT NULL		
CAPACIDAD	CAPACIDAD DEL DISCO DURO	VARCHAR	5	NOT NULL		
TIPO	TIPO DE DISCO DURO	VARCHAR	10	NOT NULL		
NO_INVENTARIO	IDENTIFICADOR DE LA TABLA ACCESORIO	VARCHAR	20	NOT NULL		*

TABLA: STINV\_EQUIPO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NOMBRE	NOMBRE DEL EQUIPO	VARCHAR	20	NOT NULL	*	
NUM	NUMERO DEL EQUIPO	INTEGER	-	NOT NULL		
RESGUARDO	NUMERO QUE AMPARA LA DEVOLUCIÓN DE UN EQUIPO QUE HA SIDO PRESTADO	VARCHAR	20	NULL		
PROPIETARIO	NOMBRE DEL PROPIETARIO DEL EQUIPO	VARCHAR	4	NOT NULL		*
TIPO	TIPO DE EQUIPO	VARCHAR	20	NOT NULL		

TABLA: STINV\_EQUIPORED

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NUMERO DE INVENTARIO DEL EQUIPO DE RED	VARCHAR	20	NOT NULL	*	
NO_SERIE	NUMERO DE SERIE DEL EQUIPO DE RED	VARCHAR	20	NOT NULL		
MARCA	MARCA DEL EQUIPO DE RED	VARCHAR	20	NOT NULL		
MODELO	MODELO DEL EQUIPO DE RED	VARCHAR	20	NOT NULL		
TIPO	TIPO DEL EQUIPO DE RED	VARCHAR	30	NOT NULL		
IP	IP DEL EQUIPO DE RED	VARCHAR	15	NULL		
COLOR	COLOR DEL EQUIPO DE RED	VARCHAR	10	NOT NULL		
DESCRIPCION	DESCRIPCION DEL EQUIPO DE RED	VARCHAR	255	NULL		
ESTADO	ESTADO EN EL QUE SE ENCUENTRA EL EQUIPO DE RED	CHAR	1	NOT NULL		
CVE_LOCALIZACION	CLAVE DE LA LOCALIZACIÓN DEL EQUIPO DE RED	VARCHAR	10	NOT NULL		*
CLV_PROY	CLAVE DEL PROYECTO	VARCHAR	5	NOT NULL		*

TABLA: STINV\_IMPRESORA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NUMERO DE INVENTARIO DE LA IMPRESORA	VARCHAR	20	NOT NULL	*	
NO_SERIE	NUMERO DE SERIE DE LA IMPRESORA	VARCHAR	20	NOT NULL		
MARCA	MARCA DE LA IMPRESORA	VARCHAR	20	NOT NULL		
MODELO	MODELO DE LA IMPRESORA	VARCHAR	20	NOT NULL		
TIPO	TIPO DE LA IMPRESORA	VARCHAR	15	NOT NULL		
ESTADO	ESTADO EN EL QUE SE ENCUENTRA LA IMPRESORA	CHAR	1	NOT NULL		
CVE_LOCALIZACION	CLAVE DE LA LOCALIZACIÓN DE LA IMPRESORA	VARCHAR	10	NOT NULL		*
CLV_PROY	CLAVE DEL PROYECTO	VARCHAR	5	NOT NULL		*

TABLA: STINV\_INICIO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_BITACORA	IDENTIFICADOR DE LA TABLA STINV_BITACORA	VARCHAR	10	NOT NULL	*	*
ID_SESION	IDENTIFICADOR DE LA TABLA SESION	VARCHAR	10	NOT NULL	*	*

TABLA: STINV\_LOCALIZACION

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
CVE_LOCALIZACION	CLAVE DE LA LOCALIZACIÓN	VARCHAR	10	NOT NULL	*	
UBICACION	NOMBRE DE LA UBICACIÓN	VARCHAR	150	NOT NULL		
CLV_DEPENDENCIA	CLAVE DE LA DEPENDENCIA	VARCHAR	10	NOT NULL		*

TABLA: STINV\_MANTENIMIENTO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
FOLIO	NÚMERO DE FOLIO DE LA SOLICITUD DE MANTENIMIENTO	VARCHAR	10	NOT NULL	*	
FECHA	FECHA EN LA QUE SE REALIZÓ EL MANTENIMIENTO	DATE	-	NOT NULL		
DESCRIPCION	DESCRIPCIÓN DEL MANTENIMIENTO	VARCHAR	255	NOT NULL		
DESC_EQ	DESCRIPCIÓN DEL EQUIPO QUE SE MANDA A MANTENIMIENTO	VARCHAR	255	NOT NULL		
OBSERVACIONES	OBSERVACIONES REALIZADAS PARA EL MANTENIMIENTO	VARCHAR	255	NOT NULL		
AUTORIZO	NOMBRE DE LA PERSONA QUE AUTORIZÓ EL MANTENIMIENTO	VARCHAR	4	NOT NULL		*
NO_INVENTARIO	NÚMERO DE INVENTARIO DEL EQUIPO QUE SE MANDÓ A MANTENIMIENTO	VARCHAR	20	NOT NULL		*

TABLA: STINV\_MONITOR

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NUMERO DE INVENTARIO DEL MONITOR	VARCHAR	20	NOT NULL	*	
NO_SERIE	NUMERO DE SERIE DEL MONITOR	VARCHAR	20	NOT NULL		
MARCA	MARCA DEL MONITOR	VARCHAR	20	NOT NULL		
MODELO	MODELO DEL MONITOR	VARCHAR	20	NOT NULL		
TAMAÑO	TAMAÑO DEL MONITOR	VARCHAR	10	NOT NULL		
TIPO	TIPO DEL MONITOR	VARCHAR	10	NOT NULL		
COLOR	COLOR DEL MONITOR	VARCHAR	15	NOT NULL		
ANOTACIONES	ANOTACIONES CON RESPECTO AL MONITOR	VARCHAR	100	NULL		
ESTADO	ESTADO EN EL QUE SE ENCUENTRA EL MONITOR	CHAR	1	NOT NULL		

CVE_LOCALIZACION	CLAVE DE LA LOCALIZACIÓN	VARCHAR	10	NOT NULL		*
CLV_PROY	CLAVE DEL PROYECTO	VARCHAR	5	NOT NULL		*

TABLA: STINV\_MOUSE

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NUMERO DE INVENTARIO DEL MOUSE	VARCHAR	20	NOT NULL	*	
NO_SERIE	NUMERO DE SERIE DEL MOUSE	VARCHAR	20	NOT NULL		
MARCA	MARCA DEL MOUSE	VARCHAR	20	NOT NULL		
MODELO	MODELO DEL MOUSE	VARCHAR	20	NOT NULL		
TIPO	TIPO DEL MOUSE	VARCHAR	20	NOT NULL		
COLOR	COLOR DEL MOUSE	VARCHAR	15	NOT NULL		
ESTADO	ESTADO EN EL QUE SE ENCUENTRA EL MOUSE	CHAR	1	NOT NULL		
CVE_LOCALIZACION	CLAVE DE LA LOCALIZACIÓN	VARCHAR	10	NOT NULL		*
CLV_PROY	CLAVE DEL PROYECTO	VARCHAR	5	NOT NULL		*

TABLA: STINV\_MOVIMIENTOS

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NUMERO DE INVENTARIO DEL DISPOSITIVO QUE SUFRIÓ ALGUNA MODIFICACIÓN	VARCHAR	20	NOT NULL	*	*
ID_BITACORA	IDENTIFICADOR DE LA TABLA BITÁCORA	VARCHAR	10		*	*
TIPO	TIPO DE MOVIMIENTO REALIZADO	VARCHAR	10	NOT NULL		

TABLA: STINV\_PERSONA

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
CVE_PERSONA	CLAVE DE LA PERSONA	VARCHAR	4	NOT NULL	*	
NOMBRE	NOMBRE DE LA PERSONA	VARCHAR	30	NOT NULL		
APATERNO	APELLIDO PATERNO DE LA PERSONA	VARCHAR	20	NOT NULL		
AMATERNO	APELLIDO MATERNO DE LA PERSONA	VARCHAR	20	NOT NULL		
TELEFONO	TELEFONO DE LA PERSONA	VARCHAR	15	NOT NULL		
CARGO	CRAGO DE LA PERSONA	CHAR	1	NOT NULL		

TABLA: STINV\_PERTENECE

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NÚMERO DE INVENTARIO DE UN EQUIPO	VARCHAR	20	NOT NULL	*	*
NOMBRE	NOMBRE DE LA PERSONA	VARCHAR	20	NOT NULL	*	*

TABLA: STINV\_PROCESADOR

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_PROC	IDENTIFICADOR DE LA TABLA STINV_PROCESADOR	VARCHAR	15	NOT NULL	*	
MARCA	MARCA DEL PROCESADOR	VARCHAR	20	NOT NULL		
MODELO	MODELO DEL PROCESADOR	VARCHAR	20	NOT NULL		
VELOCIDAD	VELOCIDAD DEL PROCESADOR	VARCHAR	10	NOT NULL		
NO_INVENTARIO	NÚMERO DE INVENTARIO DEL CPU AL QUE PERTENECE	VARCHAR	20	NOT NULL		*

TABLA: STINV\_PROYECTO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
PROY_ID	IDENTIFICADOR DE LA TABLA STINV_PROYECTO	VARCHAR	5	NOT NULL	*	
NOMBRE	NOMBRE DEL PROYECTO	VARCHAR	30	NOT NULL		

TABLA: STINV\_RAM

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_RAM	IDENTIFICADOR DE LA TABLA STINV_RAM	VARCHAR	20	NOT NULL	*	
MARCA	MARCA DE LA RAM	VARCHAR	20	NOT NULL		
TIPO	TIPO DE RAM	VARCHAR	15	NOT NULL		
MEMORIA	CAPACIDAD DE LA RAM	VARCHAR	15	NOT NULL		
NO_INVENTARIO	NÚMERO DE INVENTARIO DEL CPU AL QUE PERTENECE	VARCHAR	20	NOT NULL		*

TABLA: STINV\_SESION

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_SESION	IDENTIFICADOR DE LA TABLA STINV_SESION	VARCHAR	10	NOT NULL	*	

FECHA	FECHA EN QUE SE ABRIÒ UNA SESIÒN	DATE	-	NOT NULL		
HORA_I	HORA DE INICIO DE SESIÒN	TIME	-	NOT NULL		
HORA_F	HORA DE FIN DE SESIÒN	TIME	-	NOT NULL		
IP	IP DEL EQUIPO QUE ABRIÒ UNA SESIÒN	VARCHAR	15	NOT NULL		*
LOGIN	LOGIN DE LA PERSONA QUE ABRIÒ UNA SESIÒN	VARCHAR	10	NOT NULL		*

TABLA: STINV\_SOFTWARE

ATRIBUTO	DESCRIPCIÒN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
CLV_SOFT	CLAVE DEL SOFTWARE	VARCHAR	5	NOT NULL	*	
MARCA	MARCA DEL SOFTWARE	VARCHAR	20	NOT NULL		
NOMBRE_SOFT	NOMBRE DEL SOFTWARE	VARCHAR	30	NOT NULL		
VERSION	VERSION	VARCHAR	15	NOT NULL		
TIPO	TIPO DE SOFTWARE	VARCHAR	20	NOT NULL		
NOMBRE	NOMBRE DEL EQUIPO QUE TIENE ESE SOFTWARE	VARCHAR	20	NOT NULL		*

TABLA: STINV\_SOLICITUD

ATRIBUTO	DESCRIPCIÒN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
FOLIO	FOLIO DE LA SOLICITUD	VARCHAR	10	NOT NULL	*	
ID_BITACORA	IDENTIFICADOR DE LA TABLA STINV_BITACORA	VARCHAR	10	NOT NULL		
LOGIN	NOMBRE DE USUARIO QUE REALIZÒ ESA SOLICITUD	VARCHAR	10	NOT NULL		

TABLA: STINV\_TARJETA\_AUDIO

ATRIBUTO	DESCRIPCIÒN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_TAR_AUDIO	IDENTIFICADOR DE LA TABLA STINV_TARJETA_AUDIO	VARCHAR	20	NOT NULL	*	
MARCA	MARCA DE LA TARJETA DE AUDIO	VARCHAR	20	NOT NULL		
TIPO	TIPO DE CONEXIÒN DE LA TARJETA DE AUDIO	VARCHAR	15	NOT NULL		
NO_INVENTARIO	NÚMERO DE INVENTARIO DEL CPU AL QUE PERTENECE	VARCHAR	20	NOT NULL		*

TABLA: STINV\_TARJETA\_RED

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_TAR_AUDIO	IDENTIFICADOR DE LA TABLA STINV_TARJETA_RED	VARCHAR	20	NOT NULL	*	
MARCA	MARCA DE LA TARJETA DE RED	VARCHAR	20	NOT NULL		
IP	IP DE LA TARJETA DE RED	VARCHAR	15	NOT NULL		
ETH	ETHERNET	VARCHAR	15	NOT NULL		
NO_INVENTARIO	NÚMERO DE INVENTARIO DEL CPU AL QUE PERTENECE	VARCAHR	20	NOT NULL		*

TABLA: STINV\_TARJETA\_VIDEO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
ID_TAR_VIDEO	IDENTIFICADOR DE LA TABLA STINV_TARJETA_VIDEO	VARCHAR	20	NOT NULL	*	
MARCA	MARCA DE LA TARJETA DE VIDEO	VARCHAR	20	NOT NULL		
MEMORIA	MEMORIA DE LA TARJETA DE VIDEO	VARCHAR	10	NOT NULL		
VELOCIDAD	VELOCIDA DE LA TARJETA DE VIDEO	VARCHAR	10	NOT NULL		
NO_INVENTARIO	NÚMERO DE INVENTARIO DEL CPU AL QUE PERTENECE	VARCHAR	20	NOT NULL		*

TABLA: STINV\_TECLADO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
NO_INVENTARIO	NUMERO DE INVENTARIO DEL TECLADO	VARCHAR	20	NOT NULL	*	
NO_SERIE	NUMERO DE SERIE DEL TECLADO	VARCHAR	20	NOT NULL		
MARCA	MARCA DEL TECLADO	VARCHAR	20	NOT NULL		
MODELO	MODELO DEL TECLADO	VARCHAR	20	NOT NULL		
COLOR	COLOR DEL TECLADO	VARCHAR	15	NOT NULL		
ANOTACIONES	ANOTACIONES CON RESPECTO AL TECLADO	VARCHAR	100	NULL		
ESTADO	ESTADO EN EL QUE SE ENCUENTRA EL TECLADO	CHAR	1	NOT NULL		
CVE_LOCALIZACION	CLAVE DE LA LOCALIZACIÓN	VARCHAR	10	NOT NULL		*
CLV_PROY	CLAVE DEL PROYECTO	VARCHAR	5	NOT NULL		*

TABLA: STINV\_USUARIO

ATRIBUTO	DESCRIPCIÓN	TIPO	LONGITUD	NULOS	LLAVE PRIMARIA	LLAVE FORANEA
LOGIN	LOGIN DE USUARIO	VARCHAR	10	NOT NULL	*	
TIPO	TIPO DE USUARIO	CHAR	1	NOT NULL		
PASSWORD	PASSWORD DEL USUARIO	VARCHAR	10	NOT NULL		
ESTADO	ESTADO EN EL QUE SE ENCUENTRA EL USUARIO	CHAR	1	NOT NULL		
CVE_PERSONA	CLAVE DE PERSONA	VARCHAR	4	NOT NULL		*