

Universidad de las Ciencias Informáticas

Facultad 2.



Título: Módulo de Control de hardware del SIGLA
“Sistema Integral de Gestión de los
Laboratorios”.

**Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas.**

Autor(es): Sheila Diaraye Sylla Martín.

Danny Torriente Hernández.

Tutor: Salvador González Gómez.

Junio 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Sheila Diaraye Sylla Martin

Firma del Autor

Danny Torriente Hernández

Firma del Autor

Salvador González Gómez

Firma del Tutor

"La independencia no es una bandera, o un himno, o un escudo; la independencia no es una cuestión de símbolo, la independencia depende del desarrollo, depende de la tecnología, depende de la ciencia en el mundo de hoy".

Fidel Castro Ruz.

AGRADECIMIENTOS

A la Revolución Cubana por permitirnos formar parte de este inmenso proyecto.

A nuestros amigos por su apoyo incondicional.

A los profesores que han marcado pautas en nuestra vida estudiantil.

A nuestro tutor por su sabia orientación.

A los compañeros de aula que han extendido siempre su mano amiga.

A la universidad y a todas aquellas personas que de una forma u otra, han contribuido a nuestra formación profesional.

En especiales a nuestras familias, por guiarnos ejemplarmente por el camino certero.

¡GRACIAS!

DEDICATORIA

A mi querida abuela, por haber estado siempre a mi lado dándome fuerzas para seguir adelante y creer en mí.

A mis bellas hermanas, por saber llenar mi corazón en los momentos más difíciles con su amor y ternura.

A mis adorados Muñiz Romero y a mis incondicionales amigos Dayron y Luis, por llenar mi vida de risas y haberme demostrado que existen lazos más allá de la propia sangre.

A mi papito lindo, por no dejarme flaquear durante esta ardua contienda.

Y en especial a esa mujer inquebrantable que me hace sentir única, me ha enseñado que los sacrificios son parte de nuestros logros y que es mi mayor orgullo, mi Mamita.

¡FAMILIA lo logré!

Sheila.

A mis madres Zoraida, Ileana, Xiomara y mis padres Pelly y Denis, por ser las personas más importante en mi vida, por plasmarme esas ansias de superación continua y por toda su confianza.

A todos mis hermanos, por significar tantas cosas lindas en mi vida.

A mi familia por inspirar todos los esfuerzos en estos años.

A mis amigos de todos los tiempos, José, Rubén, Gueimy y Raúl por brindarme esa mano amiga cuando la he necesitado.

A mi novia Marleydí por estar siempre a mi lado y darme todo su apoyo y amor, a su familia por ese cariño sincero que siempre me han brindado. A todos a aquellos que de una forma u otra han contribuido a mi formación.

¡A todos ustedes: Dedico este trabajo!

Danny.

RESUMEN

La dirección de los laboratorios de la UCI, es la entidad laboral del centro universitario encargada de controlar los recursos de hardware incorporados a las áreas docentes-productivas. En ella se realizan actividades y procesos para la gestión y control de los dispositivos de hardware de las estaciones de trabajo. Estos procesos son realizados actualmente de forma independiente en dos aplicaciones, DATAPCLAB (Datos de PC laboratorios) para la realización de actividades de gestión de inventarios de dispositivos de hardware “inserción, modificación y eliminación de sus datos” y GATSERVER (Servidor del grupo de asistencia técnica) para la realización de reportes de desperfecto.

Ambas aplicaciones aunque muy útiles carecen de funcionalidades necesarias para lograr un mayor control de los dispositivos de hardware, entre ellas la incorrecta definición de los roles de accesos de los usuarios involucrado en el negocio, la falta de interacción entre las modificaciones que se le realizan a los dispositivos y los reportes de desperfecto elaborados, y más importante aún estas aplicaciones son solamente ejecutable en la plataforma propietaria Windows.

Atendiendo a las peticiones de los clientes y con el objetivo de mejorar los métodos utilizados para el control de hardware, se decide implementar una aplicación Web dinámica que unifique los procesos de gestión y control de los dispositivos de hardware y establezca una relación entre ellos.

Para la elaboración de la aplicación, se uso como gestor de base de datos PostgreSQL, lenguaje de programación PHP 5.2, Ext. JS 2.0 para el diseño visual y Apache como servidor Web. Se decidió trabajar con la arquitectura cliente/servidor y el patrón Modelo Vista Controlador. También se hizo uso de la metodología de desarrollo RUP, del lenguaje modelado UML y de IDEF0 para la modelación de los procesos del negocio.

Con la implementación de la propuesta, se obtiene una herramienta de control de hardware multiplataforma, de fácil manejo que permitirá a la dirección de los laboratorios gestionar y controlar de manera más rigurosa el manejo de los dispositivos de hardware de las estaciones de trabajo.

Tabla de contenidos

DECLARACIÓN DE AUTORÍA	I
AGRADECIMIENTOS	III
DEDICATORIA	IV
RESUMEN.....	VI
INTRODUCCIÓN.....	1
Capítulo#1: Fundamentación teórica.....	4
Introducción	4
1.1 Sistemas de gestión de hardware existentes.	4
1.1.1. Sistemas de gestión de hardware : ámbito internacional.	4
1.1.1.1 NetSupport DNA.....	4
1.1.1.2 NetSupport Manager 10.	4
1.1.1.3 NetSupport DNA HelpDesk.	5
1.1.2. Sistemas de gestión de hardware: ámbito UCI.	5
1.1.2.1. Servidor de grupo de asistencia técnica: GATSERVER	5
1.1.2.2. Datos de PC laboratorios : DATAPCLAB.....	5
1.2 Lenguaje de modelado y metodología de desarrollo.	6
1.2.1 Lenguaje de modelado: UML.....	6
1.2.2 Metodología de desarrollo : RUP.....	6
1.3 Modelamiento de funciones mediante IDEF0.....	8
1.4 Arquitectura.	8
1.4.1 Arquitectura Cliente/Servidor.	8
1.4.2 Patrón Modelo Vista Controlador (MVC).....	9
1.5 Tecnologías, lenguajes de programación y librería.	10
1.5.1 Lenguaje de programación: PHP 5.2.	10
1.5.2 Sistema gestor de base de datos: PostgreSQL.....	10
1.5.3 Servidor Web: Apache.....	11
1.5.4 Ext JS 2.0: Librería de componentes de diseño.....	11

1.5.4.1	AJAX	12
1.6	Herramientas utilizadas.	15
1.6.1	Visual Paradigm para UML.	16
1.6.2	PgAdmin III.	16
1.6.3	GIMP (General Image Manipulation Program).	17
1.6.4	Aptana.	17
	Conclusiones	18
	Capítulo #2: Características del sistema.	19
	Introducción	19
2.1	Objeto de estudio.....	19
2.1.1	Problema y situación problemática:	19
2.1.2	Objeto de automatización.	20
2.1.3	Información que se maneja.....	20
2.2	Propuesta de sistema.	20
2.3	Modelo de negocio.....	21
2.3.1	Reglas del negocio.	21
2.3.2	Descripción de los actores y trabajadores del negocio.....	21
2.3.3	Diagramas de procesos del negocio.	23
2.3.4	Descripción textual de los procesos de negocio definidos.	24
2.4	Especificación de los requisitos de software.	25
2.4.1	Requerimientos funcionales.....	25
2.4.2	Requerimientos no funcionales:.....	26
2.5	Modelo del sistema.	29
2.5.1	Actores del sistema.	29
2.5.2	Diagrama de casos de usos a automatizar.	29
2.5.3	Descripción de los casos de usos del sistema (CUS).	31
	Conclusiones	52
	Capítulo #3: Análisis y diseño.	53
	Introducción	53

3.1	Análisis. Modelo de clases de análisis por casos de uso.	53
3.1.1	Diagramas de colaboración del análisis.	59
3.2	Diseño.	65
3.2.1	Diagramas de interacción.	65
3.2.2	Diagramas de clases del Diseño.	66
3.3	Diseño de la base de datos.	77
3.3.1	Modelo lógico de datos (diagrama de clases persistentes).	77
3.3.2	Modelo físico de datos (Diagrama entidad-relación).	77
	Conclusiones	78
	Capítulo #4: Implementación.	79
	Introducción	79
4.1	Modelo de implementación.	79
4.1.1	Diagrama de despliegue.	79
4.1.2	Diagrama de componentes.	79
	Conclusiones	81
	Capítulo #5: Estudio de Factibilidad.	82
	Introducción.	82
5.1	Método de Estimación por Puntos de Casos de Uso.	82
5.1.1	Cálculo de Puntos de Casos de Uso sin ajustar.	82
5.1.2	Cálculo de Puntos de Casos de Uso ajustados.	83
5.1.3	De los Puntos de Casos de Uso a la estimación del esfuerzo.	85
5.1.4	Esfuerzo del módulo.	86
5.2	Beneficios tangibles e intangibles.	87
5.3	Análisis de costo.	87
	Conclusiones	88
	CONCLUSIONES	89
	RECOMENDACIONES	90
	Referencias bibliográficas.	91
	BIBLIOGRAFÍA	91

ANEXOS.....	94
Anexo_1: Diagramas de secuencia del diseño.....	94
Anexo_2: Diagrama Entidad-Relación.....	107
GLOSARIO DE TÉRMINOS.....	108

Índice de tablas

Tabla 1: Descripción de actores del negocio.....	22
Tabla 2: Descripción de los trabajadores del negocio.	22
Tabla 3: Descripción del proceso “Gestión de datos de los dispositivos de hardware”.....	24
Tabla 4: Descripción del proceso “Reportar estaciones de trabajo (CPU) defectuosos”.....	25
Tabla 5: Descripción de los actores del sistema.....	29
Tabla 6: Descripción del CUS “Gestionar inventario de dispositivos de hardware”.....	36
Tabla 7: Descripción del CUS “Realizar reporte de incidencia”.....	40
Tabla 8: Descripción del CUS “Realizar reporte de desperfecto”.....	41
Tabla 9: Descripción del CUS “Actualizar estado de reporte.”.....	42
Tabla 10: Descripción del CUS “Mostrar inventario de dispositivos de hardware”.....	50
Tabla 11: Descripción del CUS “Mostrar reportes”.....	52
Tabla 12: Cálculo del Factor de Peso de los Actores sin ajustar.....	82
Tabla 13: Cálculo del Factor de Peso de los Casos de Uso sin ajustar.....	83
Tabla 14: Peso de los factores determinantes de la complejidad técnica del sistema.....	84
Tabla 15: Peso y valor asignado de cada factor que intervienen en el cálculo del Factor de ambiente.....	85
Tabla 16: Cálculo del esfuerzo del módulo.....	86

Índice de ilustraciones

Figura 1: Arquitectura Cliente/Servidor.	9
Figura 2: Diagrama de procesos “Realizar reporte de desperfecto”.	23
Figura 3: Diagrama de procesos “Gestionar inventario de dispositivos de hardware”.....	23
Figura 4: Diagrama de casos de uso del sistema.....	30
Figura 5: Diagrama de clases del CUS “Realizar reporte de incidencia”.	53
Figura 6: Diagrama de clases del CUS “Actualizar estado de reporte”.....	54
Figura 7: Diagrama de clases del CUS “Realizar reporte de desperfecto”.....	54
Figura 8: Diagrama de clases del CUS “Realizar reporte de incidencia”.	55
Figura 9: Diagrama de clases del CUS “Mostrar reportes”.	56
Figura 10: Diagrama de clases del CUS “Gestionar inventario de dispositivos de hardware”.....	57
Figura 11: Diagrama de clases del CUS “Mostrar inventario”.....	58
Figura 12: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección dispositivo.	59
Figura 13: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección Estaciones.	59
Figura 14: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección Laboratorio.....	59
Figura 15: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección UCI.	59
Figura 16: Diagrama de colaboración del análisis CUS “Realizar reporte de incidencia” (Interno).	60
Figura 17: Diagrama de colaboración del análisis CUS “Realizar reporte de incidencia” (Externo).	60
Figura 18: Diagrama de colaboración del análisis CUS “Realizar reporte de desperfecto”.....	61
Figura 19: Diagrama de colaboración del análisis CUS “Actualizar estado de reporte.”.....	61
Figura 20: Diagrama de colaboración del análisis CUS “Gestionar inventario de dispositivos de hardware” Sección Insertar.	62
Figura 21 : Diagrama de colaboración del análisis CUS “Gestionar inventario de dispositivos de hardware” Sección Eliminar.....	63
Figura 22: Diagrama de colaboración del análisis CUS “Gestionar inventario de dispositivos de hardware” Sección Modificar.	64
Figura 23: Diagrama de colaboración del análisis CUS “Mostrar reportes”.	65
Figura 24: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección Área.....	65
Figura 25: Diagrama de clases del diseño CUS “Actualizar estado de reporte”.	66
Figura 26: Diagrama de clases del diseño CUS “Realizar reporte de incidencia” (interno).	67

Figura 27: Diagrama de clases del diseño CUS “Realizar reporte de incidencia”	68
Figura 28: Diagrama de clases del diseño CUS “Realizar reporte de desperfecto”	69
Figura 29: Diagrama de clases del diseño CUS “Mostrar reportes”	70
Figura 30: Diagrama de clases del diseño CUS “Mostrar Inventario” (Dispositivo).....	71
Figura 31: Diagrama de clases del diseño CUS “Mostrar Inventario” (CPU).	72
Figura 32: Diagrama de clases del diseño CUS “Mostrar Inventario” (Laboratorio).	73
Figura 33: Diagrama de clases del diseño CUS “Mostrar Inventario” (Área).	74
Figura 34: Diagrama de clases del diseño CUS “Mostrar Inventario” (UCI).	75
Figura 35: Diagrama de clases del diseño CUS “Gestionar inventario de dispositivos de hardware” ..	76
Figura 36: Diagrama de clases persistentes.	77
Figura 37: Diagrama de despliegue.	79
Figura 38: Diagrama de componentes del módulo.	80
Figura 39: Diagrama de secuencia del diseño CUS “Gestionar inventario de dispositivos de hardware”. Sección Insertar.	94
Figura 40: Diagrama de secuencia del diseño CUS “Gestionar inventario de dispositivos de hardware”. Sección Eliminar.	95
Figura 41: Diagrama de secuencia del diseño CUS “Gestionar inventario de dispositivos de hardware”. Sección Modificar.	96
Figura 42: Diagrama de secuencia del diseño CUS “Realizar reporte de incidencia”. (Interno).	97
Figura 43: Diagrama de secuencia del diseño CUS “Actualizar estado de reporte.”	97
Figura 44: Diagrama de secuencia del diseño CUS “Realizar reporte de incidencia”	98
Figura 45: Diagrama de secuencia del diseño CUS “Realizar reporte de desperfecto.....	99
Figura 46: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección Dispositivo.	100
Figura 47: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección CPU.....	101
Figura 48: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección Laboratorio.	102
Figura 49: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección Área.....	103
Figura 50: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección UCI.	104
Figura 51: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección Almacén.	105
Figura 52: Diagrama de secuencia del diseño CUS “Mostrar reportes”. Sección reportes.	106
Figura 53: Diagrama Entidad-Relación.	107

INTRODUCCIÓN

El incesante progreso de la ciencia informática, juntamente con la creciente globalización de la economía y el conocimiento, conducen a profundos cambios estructurales en todas las naciones.

El lograr un acceso fácil y rápido a la información correcta de manera oportuna, hacer más corta la distancia, mínimos los errores, óptimo el tiempo en la obtención de los datos, hacen de la informática uno de los avances científicos que marca una nueva era en el desarrollo de la sociedad, la era de la informatización. Era que ha propiciado un incremento considerable en el desarrollo de productos de software y la creación de aplicaciones Web dinámicas y convertido ambos productos, en fuentes de ingreso económico de los países desarrollados.

Con el propósito de alcanzar un alto desarrollo tecnológico, económico y social en Cuba, surgió la Universidad de las Ciencias Informáticas (UCI); una universidad docente-productiva, vinculada no solo a la docencia sino también al desarrollo de proyectos productivos. La UCI cuenta con 147 laboratorios, en los cuales se realizan diversas actividades o procesos en función de la docencia y la producción. Por consiguiente, la dirección de los laboratorios de la UCI es considerada una de las áreas de mayor desempeño laboral, debido a los recursos de hardware que en ella se manejan y el personal vinculado a sus procesos de desarrollo.

Estos procesos en su mayoría son realizados de forma manual, lo que en varias ocasiones ha provocado tanto dificultades como ineficiencias a la hora de su realización.

En aras de mejorar el modo en que son realizados estos procesos y eliminar la dependencia de sistemas ajenos a la dirección de los laboratorios, surge el proyecto "SIGLA" (Sistema Integral de Gestión de los Laboratorios). SIGLA es un proyecto que tiene como objetivo automatizar los diferentes procesos de desarrollo organizacional de la dirección de los laboratorios, en la actualidad se compone de los siguientes seis (6) módulos: Administración, Reservación de tiempo de máquina, control de medios informáticos, control de personal, control de solicitudes de software y control de hardware; cada módulo por independiente hace referencia a dichos procesos de desarrollo.

El presente trabajo de diploma se centrará en el módulo de control de hardware del SIGLA. La dirección de los laboratorios de la UCI, maneja más de 25000 dispositivos de hardware, con el fin de obtener un mayor control sobre ellos, se han desarrollado aplicaciones en función de algunas de las actividades dirigidas a su gestión: DATAPCLAB y GATSERVER.

DATAPCLAB fue desarrollada para la realización de actividades de gestión de inventarios de dispositivos de hardware, por su parte, no posee una correcta restricción de sus roles de usuario, ya que le permite a usuarios no autorizados gestionar inventarios, es decir realizar procesos de inserción, modificación y retiro de datos de dispositivos de hardware de las estaciones de trabajo. Permite realizar movimientos de puestos de manera irreal o sea, no tiene un control de las estaciones movidas de un laboratorio para un almacén, ni de las que se encuentran en el almacén. No permite realizar movimientos de estaciones de trabajo de manera independiente y no tiene definido un rol de acceso para técnicos de laboratorios. GATSERVER es una aplicación del Grupo de asistencia técnica, que permite a técnicos de hardware de las áreas de los laboratorios, la realización de reportes de estaciones de trabajo dañadas o con posibles fallas en los dispositivos de hardware que las componen. Esta aplicación al igual que DATAPCLAB, no tiene bien definidas las restricciones de sus roles, permite a usuarios no autorizados realizar reportes de estaciones de trabajo en áreas ajenas a su entorno laboral, además no muestra el histórico de reporte.

En el año 2007, fue propuesta de este módulo, el Trabajo de Diploma presentado por Maybel Marrero León y Yenlys Remis Barrios, para optar por el título de Ingeniero en Ciencias Informáticas. Tomando como base la información brindada en ese trabajo y en aras de refinar las ideas de desarrollo planteadas y alcanzar una solución más detallada, se realizaron encuestas e investigaciones sobre el negocio y se analizaron los diagramas y modelos que fueron elaborados.

Las divergencias existentes entre el negocio propuesto y el actual, el aumento de los requisitos funcionales a cumplirse por el sistema, la necesidad de cambiar las herramientas y tecnologías por otras más novedosas, ha provocado un cambio gradual de dicha propuesta y conlleva a determinar como problema científico “¿Cómo gestionar los procesos de control de dispositivos de hardware en los laboratorios? ”, establecer como objeto de estudio los “Procesos de control de dispositivos de hardware” y como campo de acción la “Gestión de los procesos de control de dispositivos de hardware en los laboratorios de la UCI”.

El Objetivo general de este trabajo de diploma es “Desarrollar una aplicación Web, que integre los procesos de gestión de dispositivos de hardware, que se realizan en la dirección de los laboratorios de la UCI”.

Tareas de investigación establecidas.

- Investigar sobre herramientas, tecnologías, metodologías y lenguajes multiplataforma y establecer los que serán utilizados para la correcta implementación del sistema.

- Investigar sobre las plantillas necesarias para el control de los dispositivos de hardware en los laboratorios de la UCI.
- Revisar la propuesta realizada anteriormente como base para el desarrollo de la aplicación.

Aportes que se podrán obtener con la automatización.

Informatizar el módulo de control de hardware, aumentará la rapidez, eficiencia y tiempo de ejecución de los procesos de gestión de dispositivos de hardware en los laboratorios de la UCI. Permitirá mantener una actualización constante de los cambios referente al proceso de reporte, además de independizar a la dirección de los laboratorios de cierto modo, del grupo de asistencia técnica (GAT).

Estructuración del contenido con una breve explicación de sus partes.

En el Capítulo#1 “Fundamentación teórica”, se realizará un estudio del estado del arte, haciendo mención de aplicaciones Web, con posibles soluciones a procesos de control de dispositivos de hardware y de realización de reportes, en los ámbitos internacional, nacional y en específico de la Universidad de las Ciencias Informáticas. Se establecerán las herramientas, metodologías, arquitectura y tecnologías que utilizaremos para el desarrollo de la aplicación.

En el Capítulo#2 “Características del sistema”, se realizará un estudio profundizado del negocio correspondiente a la situación problemática planteada, y de la necesidad de desarrollo de una aplicación Web para su erradicación. Se detallarán los procesos del negocio a realizar para la gestión de los dispositivos de hardware en el área de los laboratorios, y se realizará un análisis de estos procesos para establecer las condiciones que el sistema deberá cumplir así como las características y restricciones que deberá tener para su correcto funcionamiento.

En el Capítulo#3 “Análisis y diseño”, se analizarán los requerimientos definidos durante la caracterización del sistema. Mediante la realización de los diagramas de análisis y diseño, se modelará el sistema de modo que estos reflejen su estructura interna.

En el Capítulo#4 “Implementación”, se describirá cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue, con el uso de las clases activas y subsistemas encontrados durante el diseño.

Capítulo#5 “Estudio de factibilidad”, se realizará un estudio de factibilidad al Módulo de control de Hardware mediante el Método de Estimación por Puntos de Casos de Uso. Para así poder determinar si es factible o no su implementación.

Capítulo#1: Fundamentación teórica.

Introducción

En el presente capítulo se realizará un estudio, del estado del arte y de aplicaciones Web existentes con posibles soluciones a procesos de control de dispositivos de hardware y de realización de reportes, en los ámbitos internacional, nacional y en específico de la Universidad de las Ciencias Informáticas. Se plantearán las herramientas, metodologías, arquitectura y tecnologías a utilizar en el desarrollo de la aplicación.

1.1 Sistemas de gestión de hardware existentes.

1.1.1. Sistemas de gestión de hardware : ámbito internacional.

1.1.1.1 NetSupport DNA.

NetSupport DNA, es una aplicación para el inventario de hardware e inventario de software que permite gestionar de manera ágil y centralizada sus activos informáticos. Proporciona el inventario de hardware más detallado y profundo del mercado. Desde información sobre las CPU, BIOS, o slots de memoria utilizados en cada PC hasta el sistema operativo, la configuración de red, el monitor o las impresoras. La información se puede mostrar a nivel de cada máquina, o bien agrupada por cualquier categoría. El inventario se puede realizar a intervalos de tiempo configurables, a nivel de empresa o de departamento. NetSupport DNA ofrece soporte completo para Windows 98 y posterior, junto con las siguientes distribuciones de Linux - Red Hat 9, Red Hat Enterprise, SuSE 9 y Fedora Core 6, proporcionando también mecanismos de inventariado para máquinas no conectadas a la red.

1.1.1.2 NetSupport Manager 10.

NetSupport Manager 10 en su conjunto es una solución de control remoto de PCs, que permite la gestión remota de ordenadores y servidores a través de redes locales (LAN) e Internet. Es un producto ideal para departamentos de soporte técnico ya que combina el control remoto con herramientas para realizar el inventario tanto de software como de hardware de las máquinas remotas.

Permite la obtención de una vista a tiempo real del hardware y software instalado en la estación de trabajo objetivo, con un solo clic, sin tener que abandonar nunca el programa NetSupport Control para reunir esta información. NetSupport Manager recopila más de 80 unidades de información, específicamente sobre el hardware o el entorno del PC Cliente. Muestra detalles sobre la configuración del cliente, el hardware, las aplicaciones y parches instalados, así como los procesos y servicios.

Presenta soporte para Windows, Mac, Linux y aparatos móviles desde una sola consola. Opera a través de su propia red o por Internet con seguridad y sin necesidad de configurar un cortafuego.

1.1.1.3 NetSupport DNA HelpDesk.

NetSupport DNA HelpDesk es un módulo de NetSupport DNA que también puede adquirirse de forma independiente. Se trata de una solución de HelpDesk basada en páginas Web que permite gestionar incidencias de soporte a lo largo de todo su ciclo de vida. Desde la apertura de una incidencia y su asignación a un técnico de soporte hasta su resolución, DNA HelpDesk mantiene de forma centralizada el control y toda la información relevante para su gestión.

La herramienta permite a los técnicos de soporte concentrarse en la resolución de problemas en vez de la gestión administrativa y documentación de los mismos. NetSupport DNA HelpDesk también incorpora una base de datos para guardar las resoluciones de problemas, y herramientas para generación de informes y estadísticas.

1.1.2. Sistemas de gestión de hardware: ámbito UCI.

1.1.2.1. Servidor de grupo de asistencia técnica: GATSERVER .

GATSERVER es un sistema de gestión tecnológica concebido para el grupo de soporte técnico de la universidad, es una aplicación que permite la realización de reportes y solicitudes de telefonía, de medios de dispositivos de hardware y otros medios informáticos como monitores, teclados entre otros.

1.1.2.2. Datos de PC laboratorios : DATAPCLAB.

DATAPCLAB es una herramienta de almacenamiento de datos, correspondiente al conjunto de hardware ubicado en los laboratorios de la UCI, datos referentes a especificaciones de las estaciones de trabajo de cada medio, es una aplicación desktop que se divide en 2 partes: Prestaciones y control de las PCs y Entrega Digital de los laboratorios.

Prestaciones y control de las PCs: Es una manera de llevar el control estricto de todos los componentes que pueda tener un Ordenador, permitiéndole al usuario hacer cualquier modificación en cualquiera de las partes deseadas. Además comparte con el usuario todos los datos de los dispositivos y la ubicación de los mismos.

Entrega Digital de los laboratorios: Es la entrega de los laboratorios en un formato digital el cual permite controlar todos los medios y problemas que existan en el mismo.

1.2 Lenguaje de modelado y metodología de desarrollo.

1.2.1 Lenguaje de modelado: UML.

UML (Unified Modeling Language): Es un lenguaje utilizado para el modelado de un sistema, permitiendo en mayor o menor medida representar todas las fases de un proyecto informático, desde el análisis con los casos de uso, hasta la implementación y configuración con los diagramas de despliegue.

UML permite:

- Visualizar gráficamente un sistema de manera que otros puedan entenderlo.
- Especificar cuáles son las características de un sistema antes de su construcción.
- Construir sistemas diseñados a partir de modelos especificados.
- Documentar los elementos gráficos del sistema desarrollado para futuras revisiones.

UML aporta las siguientes ventajas:

- Permite realizar una verificación y validación del modelo realizado.
- El modelado con UML es independiente del lenguaje de implementación, de tal forma que los diseños realizados se pueden implementar en cualquier lenguaje.
- Permite generar código a partir de los modelos y a la inversa, lo que posibilita la constante actualización.
- Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es el suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa y diseño de la estructura de una organización.

1.2.2 Metodología de desarrollo : RUP.

“Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipo de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.”¹

¹ **Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** 2000. El Lenguaje Unificado de Modelado. [Online] 2000. [Cited: Enero 10, 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

RUP (Rational Unified Process) es un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado UML, constituye una metodología estándar muy utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

- Se utiliza en proyectos que se desarrollan a largo plazo.
- Permite una mejor comunicación entre ingenieros de software y de negocio, ya que maneja un lenguaje común, facilita la comprensión de procesos del negocio y su traducción a las funcionalidades que brindará el sistema.
- Genera un volumen considerable de documentación, posibilitando que los cambios realizados en los miembros del equipo no resulte un factor negativo para el avance del proyecto.
- Propone el desarrollo en ciclos e iteraciones con los respectivos artefactos que se generan, siendo esto un elemento importante para alcanzar una categoría de certificación en el desarrollo del software.
- Asegura la producción de software de calidad dentro de plazos y presupuestos predecibles.

En conclusión, los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres frases claves:

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de este momento los casos de uso guían el proceso de desarrollo, ya que los modelos que se obtienen en los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema en la que el equipo del proyecto y los usuarios deben estar de acuerdo. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
- Iterativo e incremental: Propone que cada fase se desarrolle en iteraciones que involucran actividades de todos los flujos de trabajo. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

1.3 Modelamiento de funciones mediante IDEF0

IDEF0 (Integration Definition for Function Modeling) consiste en una serie de normas que definen la metodología para la representación de funciones. Representa de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades.

IDEF0 posee las siguientes características:

- Es capaz de representar gráficamente los principales procesos del negocio de una empresa a cualquier nivel de detalle.
- Permite analizar, documentar y mejorar los procesos del negocio.
- Facilita la comunicación y captura de información.
- Puede ser generado fácilmente por una gran variedad de herramientas gráficas en computadores.
- Posee flexibilidad ante cambios.

La notación IDEF puede ser representada de dos formas:

La forma **AS-IS** (tal como es) es la empleada por el negocio para representar los procesos a la manera actual, antes de la automatización.

La forma **TO-BE** (como debe ser) es utilizada para transformar el proceso a la manera deseada para los objetivos de la organización.

1.4 Arquitectura.

La arquitectura del software alude a la estructura global del software y a las formas en que la estructura proporciona la integridad conceptual de un sistema.

1.4.1 Arquitectura Cliente/Servidor.

La arquitectura cliente-servidor es una nueva tendencia en el desarrollo de redes locales, que tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de funciones: el cliente que maneja la porción de la aplicación y el servidor que administra los procesos de almacenamiento y recuperación de los datos.

Puede presentarse como uno a varios clientes y uno o más servidores, junto con un sistema operativo y una plataforma de comunicación para formar un sistema cooperativo que permita la computación distribuida, el análisis y la presentación de datos.

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

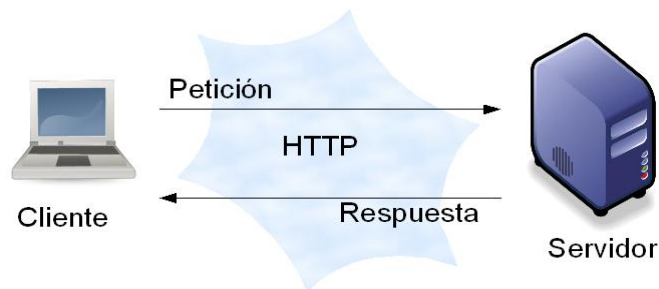


Figura 1: Arquitectura Cliente/Servidor.

Características de la arquitectura Cliente/Servidor:

- El servidor presenta una interfaz única y bien definida a todos sus clientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor no afectan al cliente.

1.4.2 Patrón Modelo Vista Controlador (MVC).

El Modelo-Vista-Controlador se creó para Smalltalk a finales de los setenta; desde entonces su uso se ha ido extendiendo cada día más para la construcción de sistemas con interfaz gráfica. MVC es un patrón de arquitectura utilizado en sistemas Web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.

El **Modelo** representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

La **Vista** es la encargada de presentar la información al usuario. Una vista puede ser una página Web o parte de una página.

El **Controlador** actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página, es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo.

1.5 Tecnologías, lenguajes de programación y librería.

1.5.1 Lenguaje de programación: PHP 5.2.

PHP es un lenguaje sencillo, de sintaxis cómoda y dispone de muchas librerías que facilitan en gran medida el desarrollo de las aplicaciones; convirtiéndolo en el favorito de millones de programadores en todo el mundo.

Características de PHP:

- Dispone de una conexión propia a varios sistemas de base de datos como: MySQL, PostgreSQL y Oracle.
- Incorpora bibliotecas que contienen funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.
- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento.
- Es un lenguaje multiplataforma.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables.
- Posee tratamiento de errores.

PHP 5.2 es una versión de PHP que además incluye:

- Soporte sólido para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte a XML.

1.5.2 Sistema gestor de base de datos: PostgreSQL.

PostgreSQL es un gestor de bases de datos relacional orientado a objetos, libre y gratuito. Está liberado bajo la licencia BSD, lo que significa que se puede disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente.

Presenta las siguientes propiedades:

- Atomicidad: Asegura la realización de una operación, por lo que ante un fallo del sistema esta no queda a medias.

- Consistencia: Posibilita la ejecución de aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
- Aislamiento: Mediante un sistema denominado MVCC (Acceso concurrente multiversión) asegura que una operación no pueda afectar a otras, de esta manera realizar dos transacciones sobre la misma información no genera error.
- Durabilidad: Asegura la permanencia de una operación realizada, y aunque falle el sistema esta no podrá deshacerse.

PostgreSQL presenta las siguientes características:

- Se puede instalar un número ilimitado de veces sin temor de sobrepasar la cantidad de licencias.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos.
- Posee confiabilidad, seguridad y flexibilidad.

1.5.3 Servidor Web: Apache.

Apache es un servidor Web posible de utilizar en distintas plataformas y entornos. Es altamente configurable de diseño modular, posibilitando que los administradores de sitios Web puedan elegir los módulos que serán incluidos y ejecutados en el servidor.

Características de Apache:

- Es una tecnología gratuita y de código abierto, lo que proporciona transparencia en todo el proceso de instalación.
- Es prácticamente universal, por su disponibilidad en multitud de sistemas operativos.
- Posee una alta configurabilidad en la creación y gestión de logs, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.

Este servidor Web tiene una fácil integración con varios lenguajes de programación como: Java, Perl y especialmente PHP. Dicha relación a dado lugar al desarrollo de aplicaciones como el APPSERV y XAMPP los cuales instalan el Apache y el PHP configurados para su uso.

1.5.4 Ext JS 2.0: Librería de componentes de diseño.

ExtJS es una librería de componentes que facilita las herramientas necesarias para la creación de aplicaciones Web con excelentes gráficos; ya que posee una considerable colección de elementos para el diseño de interfaces, ventanas, pestañas, menús, tablas, etc.

Brinda soporte para:

- Construir interfaces gráficas complejas y dinámicas.
- Comunicar datos de forma asíncrona con el servidor.
- Diversos navegadores como: Internet Explorer, Firefox, Safari y Opera.

Actualmente ExtJS es considerado un Framework independiente; ya que a principios del 2007 se creó una compañía para comercializar y dar soporte al mismo, dicha compañía proporciona los servicios de consultoría necesarios para ayudar a los clientes en el aprovechamiento máximo de las ventajas de ExtJS. Es importante señalar que la ExtJS 2.0 tiene dos tipos de licencias, LGPL (Open Source) y la comercial, esta última es obligatoria si se desea obtener soporte.

1.5.4.1 AJAX .

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), facilita la creación de aplicaciones interactivas en la Web que se ejecutan en el navegador de los usuarios y mantienen comunicación asíncrona con el servidor; posibilitando realizar cambios sobre una página sin necesidad de recargarla, aumentando de esta forma la interactividad, velocidad y usabilidad de la misma.

“AJAX no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes.”²

AJAX está conformado por:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Además:

- Provee un mecanismo para mezclar y hacer coincidir XML con XHTML.
- Las aplicaciones son más rápidas e interactivas, al estilo de aplicaciones de escritorio.
- Reduce de manera significativa tener que cargar información continuamente del servidor, ya que actualiza porciones de la página en vez de la página completa.
- Cuando se utiliza AJAX adecuadamente en el desarrollo de una aplicación, se reduce de manera significativa los tiempos de carga inicial.

² Eguíluz Pérez, Javier. 2007. Libros Web. Introducción a AJAX. [Online] 2007. <http://www.librosweb.es/ajax>.

A continuación se explican las características más importantes de algunas de las tecnologías que componen AJAX:

JSON

Notación de Objetos de JavaScript (JavaScript Object Notation), constituye un formato ligero para el intercambio de datos abierto y basado en texto. JSON está pensado principalmente para usarse en aplicaciones Web con el objetivo de transmitir información estructurada de forma asíncrona entre el servidor y los clientes.

JSON es un subconjunto de la notación literal de objetos de JavaScript que se usa generalmente con ese lenguaje. Sin embargo, los tipos básicos y las estructuras de datos de la mayoría de los lenguajes de programación también pueden ser representados en JSON, por lo que es posible usar el formato para intercambiar datos estructurados entre programas escritos en diferentes lenguajes como: ActionScript, C, C#, Java, JavaScript, Perl, PHP, Python, Ruby y Smalltalk.

Características de JSON:

- La sintaxis es muy concisa por lo que requiere menos codificación y procesamiento.
- No se necesita código de aplicación adicional para analizar texto.
- Es legible e independiente de la plataforma.
- Soporta diversos lenguajes de programación.

XML

Es el estándar de Extensible Markup Language (Lenguaje de Etiquetado Extensible), conformado por un conjunto de reglas para definir etiquetas semánticas orientadas a organizar un documento en diferentes partes. Permite al usuario definir sus propios lenguajes de anotación adaptados a sus necesidades y contiene tres características muy importantes que son: extensibilidad, estructura y validación.

XML no sólo tiene aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto y hojas de cálculo.

Ventajas de XML:

- Hasta ahora cada navegador interpreta la información a su manera, y los programadores tienen que trabajar en función del navegador del usuario. Con XML se tiene una sola aplicación que

maneja los datos, y para cada navegador se puede tener una hoja de estilo, escogiendo de estos el más adecuado a utilizar.

- Con el empleo de XML la información es más accesible y reutilizable, porque la flexibilidad de las etiquetas de XML permite que puedan utilizarse sin tener que amoldarse a reglas específicas de un fabricante.

XHTML

“XHTML es el acrónimo en inglés de Extensible Hypertext Markup Language (Lenguaje extensible de marcado de hipertexto). Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos con la de XML, diseñado para describir los datos.”³

XHTML está encaminado al uso de un etiquetado correcto, por lo que exige una serie de requisitos básicos a cumplir en cuanto al código.

Algunos de estos requisitos básicos son:

- Elementos correctamente anidados.
- Etiquetas en minúsculas.
- Elementos cerrados correctamente.
- Atributos de valores entrecomillados.

CSS

Es un lenguaje de hojas de estilos (Cascading Style Sheets) creado para controlar la presentación de documentos estructurados y escritos XHTML y XHTML. Es utilizado para definir el aspecto de todos los contenidos de una página como: el color, el tamaño, el tipo de letra, la separación entre párrafos y la tabulación con la que se muestran los elementos de una lista. El propósito del desarrollo de CSS es separar la estructura y el contenido de la presentación estética en un documento, esto permite un mayor control del documento y sus atributos, convirtiendo al XHTML en un documento muy versátil y liviano.

³ Eguíluz Pérez, Javier. 2007. Libros Web. Introducción a XHTML. [Online] 2007. <http://www.librosweb.es/xhtml>.

“Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.”Entre los beneficios concretos de CSS se encuentran⁴:

- Control de la presentación de muchos documentos desde una única hoja de estilo.
- Control más preciso de la presentación.
- Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, etc.).

JavaScript

Es un lenguaje basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente/servidor dentro del ámbito de Internet. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, convirtiéndolo en un lenguaje interpretado.

Ventajas de JavaScript:

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.
- No requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.
- Es independiente de la plataforma hardware o sistema operativo, y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript.

1.6 Herramientas utilizadas.

Según las definiciones del proyecto SIGLA se determinó el uso de herramientas libres como: Visual Paradigm, PgAdmin III, GIMP y APTANA.

⁴ **HTML.net.** Tutoriales sobre HTML y CSS - Construye tu propio sitio Web. [Online] [Cited: Marzo 15, 2008.] <http://es.html.net/tutorials/css/lesson1.asp>.

1.6.1 Visual Paradigm para UML.

Visual Paradigm para UML (VP-UML) es una herramienta CASE multiplataforma (Windows/Linux/Mac OS X) que soporta el ciclo de vida completo del desarrollo de software. Está diseñado para un amplio rango de usuarios, incluyendo ingenieros de software, analistas de sistema, analistas de negocio, arquitectos de sistemas y quienes estén interesados en la construcción de sistemas de software confiables mediante el uso de la Orientación a Objetos.

Este software facilita una rápida construcción de aplicaciones de calidad y a un menor coste. Visual Paradigm para UML soporta un conjunto de lenguajes (Java, C + +, PHP, Ada y Python), tanto en generación de código como ingeniería inversa.

Entre sus características principales se pueden citar:

- Soporta UML en su versión 2.1.
- Facilita la comunicación de todo el equipo de desarrollo mediante el uso de un lenguaje estándar común.
- Posibilita el desarrollo de la ingeniería directa e inversa.
- Se encuentra disponible en múltiples versiones y plataformas.
- Permite la generación de bases de datos a partir de la transformación de diagramas de Entidad-Relación en tablas de base de datos y viceversa.

1.6.2 PgAdmin III.

Es una aplicación gráfica usada para la gestión de PostgreSQL, posee licencia Open Source. PGAdmin está escrito en C++ y utiliza la librería gráfica multiplataforma wxWidgets, permitiendo que se pueda usar en sistemas operativos como: GNU/Linux, FreeBSD, Solaris, Mac OS y Windows. Es capaz de gestionar versiones a partir de PostgreSQL 7.3, ejecutándose en cualquier plataforma.

PGAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples, hasta desarrollar bases de datos complejas.

El producto también incluye:

- Editor SQL con resaltado de sintaxis.
- Editor de código de la parte del servidor.
- Amplia documentación para ayudar a los usuarios menos experimentados.

1.6.3 GIMP (General Image Manipulation Program).

GIMP es el acrónimo para GNU Image Manipulation Program. Es un programa libre distribuido bajo la licencia GPL, apropiado para tareas como: arreglo fotográfico, composición y edición de imagen. Es especialmente útil para la creación de logotipos y otros gráficos para páginas Web. Inicialmente se desarrolló para sistemas Unix; sin embargo, actualmente existen versiones totalmente funcionales para Windows y para Mac OS.

Características fundamentales de GIMP:

- Lee y escribe la mayoría de los formatos de ficheros gráficos, entre ellos jpg, gif, png, pcx y tiff.
- Es capaz de importar ficheros en pdf y también imágenes vectoriales en formato svg.
- Permite la automatización de muchos procesos mediante macros o secuencias de comandos.
- Dispone de varias herramientas o filtros para la manipulación de los colores y el aspecto de las imágenes.

1.6.4 Aptana.

Aptana Studio es un IDE de desarrollo para aplicaciones de la Web 2.0, gratuito y de código fuente abierto. Está basado en el conocido entorno de desarrollo Eclipse, también de código abierto; pero mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo Web, con soporte a HTML, CSS y JavaScript.

Soporta varias librerías como: Prototype, Yahoo UI y JQuery, pudiendo combinarlas fácilmente en una aplicación. Aptana Studio está disponible como una aplicación independiente o como plug-in para Eclipse y se puede encontrar para tres plataformas fundamentales: Windows, Mac OS y GNU/Linux.

La gestión de proyectos, vista previa, autocompletado de código y gestión de documentación, son algunas de las características similares que presenta Aptana con otros entornos de desarrollo integrado (Eclipse, C++ Builder, Visual Studio. NET), pero además:

- Visualiza los errores de sintaxis a medida que se escribe.
- Presenta una previa visualización de estilos CSS con el editor CSS.

Conclusiones

Partiendo de los resultados del estudio realizado, se llegó a la conclusión que a diferencia del sistema propuesto, las aplicaciones Web estudiadas en el ámbito internacional, no establecen interacción entre los reportes realizados y las posibles modificaciones efectuadas a algún dispositivo de hardware, y no realizan los procesos de inserción, modificación y retiro de datos de dispositivos de hardware de las estaciones de trabajo que manejan.

En el ámbito nacional no se encontró publicado ningún sistema vinculado al tema y en la UCI las herramientas utilizadas carecen de funcionalidades necesarias para la correcta gestión de los procesos de control de hardware.

Teniendo en cuenta el proceso de migración a software libre existente en el país, se definieron las herramientas, metodologías, tecnologías, lenguajes y arquitectura a utilizar en la implementación del sistema.

Capítulo #2: Características del sistema.

Introducción

En este capítulo, se realizará un estudio profundizado del negocio correspondiente a la situación problemática planteada, y de la necesidad de desarrollo de una aplicación Web para su erradicación. Se detallarán los procesos del negocio para el control de hardware en el área de los laboratorios, y realizará un análisis de estos procesos para establecer las condiciones que el sistema deberá cumplir así como las características y restricciones que deberá tener para su correcto funcionamiento.

2.1 Objeto de estudio.

2.1.1 Problema y situación problemática:

En los laboratorios de la UCI, se manejan un aproximado de 5000 estaciones de trabajo, las cuales por individual poseen internamente dispositivos de hardware. Con el fin de obtener un mayor control sobre los dispositivos, se han desarrollado aplicaciones en función de algunas de las actividades dirigidas a su gestión: DATAPCLAB y GATSERVER.

DATAPCLAB fue desarrollada para la realización de actividades de gestión de inventarios de dispositivos de hardware, por su parte, no posee una correcta restricción de sus roles de usuario, ya que le permite a usuarios no autorizados gestionar inventarios, es decir realizar procesos de inserción, modificación y retiro de datos de dispositivos de hardware de las estaciones de trabajo. Permite realizar movimientos de puestos de manera irreal o sea, no tiene un control de las estaciones movidas de un laboratorio para un almacén, ni de las que se encuentran en el almacén. No permite realizar movimientos de estaciones de trabajo de manera independiente y no tiene definido un rol de acceso para técnicos de laboratorios. GATSERVER es una aplicación del Grupo de asistencia técnica, que permite a técnicos de hardware de las áreas de los laboratorios, la realización de reportes de estaciones de trabajo dañadas o con posibles fallas en los dispositivos de hardware que las componen. Esta aplicación al igual que DATAPCLAB, no tiene bien definidas las restricciones de sus roles, permite a usuarios no autorizados realizar reportes de estaciones de trabajo en áreas ajenas a su entorno laboral, además no muestra el histórico de reporte.

La falta de funcionalidades de dichas aplicaciones y la necesidad de una herramienta que establezca una interacción entre los procesos anteriormente citados conforman las bases de nuestra situación problemática.

2.1.2 Objeto de automatización.

El negocio establecido para el control de hardware en el área de los laboratorios, se realiza actualmente por separado. La implementación del módulo de control de hardware tiene como objetivo, brindar una interfaz o aplicación Web dinámica, que establezca la interacción existente entre el proceso de “Reporte de estaciones de trabajo con desperfectos” y los procesos de “Inserción, modificación y eliminación de datos de estaciones de trabajo”, que sirva de herramienta de trabajo a la dirección de los laboratorios.

2.1.3 Información que se maneja.

El documento “Tarjeta de garantía” es una planilla que viene adjunta a las estaciones de trabajo cuando son incorporadas por primera vez a un área de laboratorios. La misma contiene las características de cada dispositivo de hardware de la estación de trabajo.

2.2 Propuesta de sistema.

El sistema que se desea implementar permitirá ejercer los procesos de control de hardware del área de los laboratorios. A diferencia del DATAPCLAB, la aplicación del módulo de control de hardware del SIGLA, le permitirá al usuario en dependencia a su nivel de acceso:

- Realizar reportes de desperfecto.
- Visualizar el inventario de los dispositivos de hardware de las estaciones de trabajo, general de un laboratorio, general de un área y su almacén, general de todas las áreas y de cada uno de sus almacenes.
- Visualizar reportes de desperfecto previamente realizados.
- Visualizar reportes de incidencias, correspondientes ya sea a ejecuciones de procesos de gestión de hardware o a la apertura de alguna de las estaciones de trabajo que no requirieron de proceso de gestión de inventario.

Además el sistema se encargará por sí solo de:

- Actualizar el estado de los reportes de forma dinámica, basándose en las modificaciones que se le realicen a las estaciones de trabajo a los cuales dichos reportes hacen referencia.
- Realizar reporte de incidencia: reporte que se realiza dado cambios realizados a las estaciones de trabajo , ya sea por una inserción, eliminación, modificación de dispositivos de hardware , limpieza o ajustes internos de la estación de trabajo.
- Mantener el control de acceso en dependencia al rol del usuario que acceda a él.

2.3 Modelo de negocio.

En la dirección de los laboratorios de la UCI, se realizan procesos en función del control de los dispositivos de hardware de las estaciones de trabajo. Se entiende en este caso por dispositivos de hardware, al conjunto de componentes internos dentro de un gabinete o torre, entre los cuales podemos mencionar: Motherboard (tarjeta madre), memoria RAM, disco duro, tarjeta de red, de audio, video entre otros.

Estos procesos constan de las siguientes funciones:

- Gestión de datos de los dispositivos de hardware
 - Insertar datos de los dispositivos de hardware.
 - Modificar datos de los dispositivos de hardware.
 - Eliminar datos de los dispositivos de hardware.
- Reportar estaciones de trabajo defectuosas.

2.3.1 Reglas del negocio.

R₁: El especialista general de hardware de la universidad, es el único capacitado para visualizar el inventario de cualquier área perteneciente a la dirección de los laboratorios.

R₂: El técnico de hardware del área, es el único autorizado a realizar procesos de inserción, modificación y eliminación de datos de los dispositivos de hardware.

R₃: El técnico de hardware del área es quien realiza reportes de desperfecto de las estaciones de trabajo al especialista general de hardware.

R₄: El técnico de laboratorio solamente esta capacitado para visualizar el inventario de los dispositivos de hardware del o los laboratorios que le sean asignados en su área de trabajo.

R₅: El técnico de hardware del área, el jefe de área y el jefe de turno tienen permiso para visualizar el inventario general de los dispositivos de hardware, al igual que el existente en los laboratorios de su área de trabajo.

2.3.2 Descripción de los actores y trabajadores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. Los trabajadores por otra parte son quienes realizan las actividades.

Actores del negocio	Justificación
Jefe de turno	Individuo con conocimiento previo de los inventarios de dispositivos de hardware ubicados en su entorno laboral, así como de los cambios que en ellos se realicen.
Jefe de área	Máximo responsable de su área laboral. Individuo con conocimiento previo de los inventarios de dispositivos de hardware ubicados en dicho entorno, así como de los cambios que en ellos se realicen. Recibe notificaciones de todos los procesos de hardware que se realizan en su área de trabajo.
Técnico de laboratorio	Individuo que responde por los dispositivos de hardware de los laboratorios que le son asignados en su área laboral
Especialista general de hardware	Máximo responsable de los dispositivos de hardware ubicados en la UCI, con acceso ilimitado a la información de estos en cualquier área de laboratorios. Se encarga también de firmar la planilla de reporte del grupo UCIXTEL como confirmación de los posibles servicios prestado en una de las áreas.

Tabla 1: Descripción de actores del negocio.

Trabajador del negocio	Justificación
Técnico de hardware del área.	Individuo encargado de la gestión de inventario y gestión de reportes de los dispositivos de hardware de las estaciones de trabajo. Se encarga también de verificar los cambios o reparaciones de dispositivos de hardware realizados a las estaciones de trabajo y firmar la planilla de reporte del grupo UCIXTEL, como confirmación de los posibles servicios que este grupo haya prestado en su entorno laboral.

Tabla 2: Descripción de los trabajadores del negocio.

2.3.3 Diagramas de procesos del negocio.

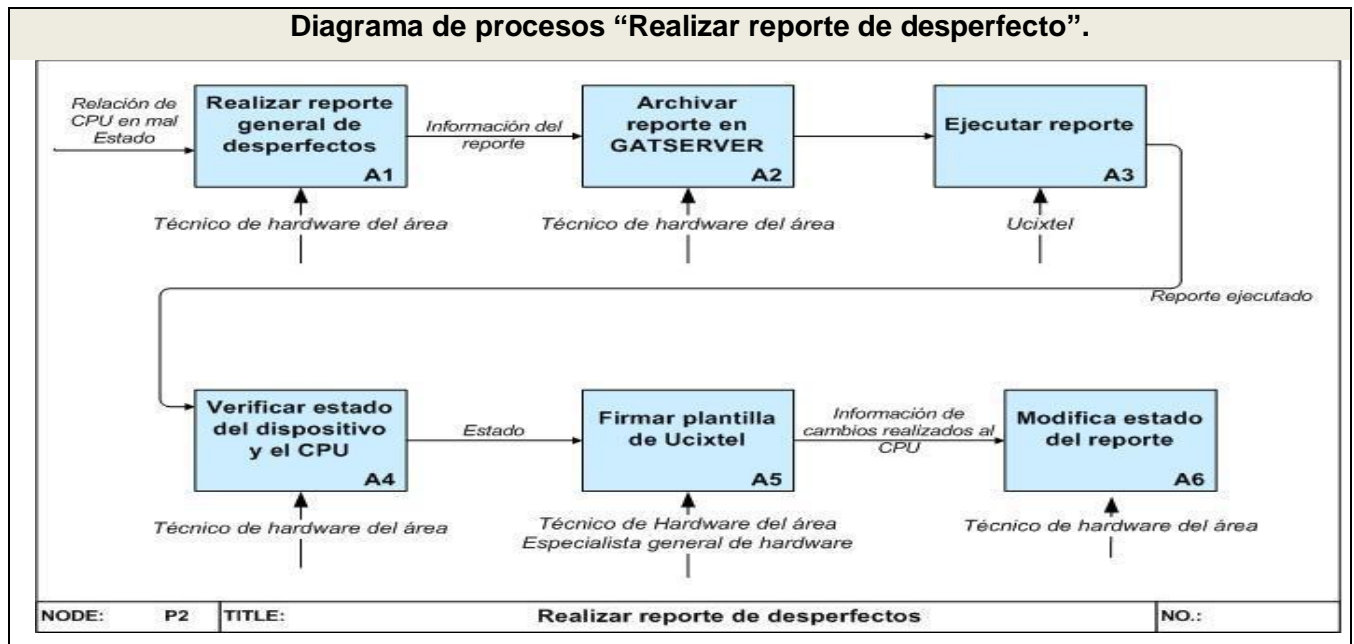


Figura 2: Diagrama de procesos "Realizar reporte de desperfecto".

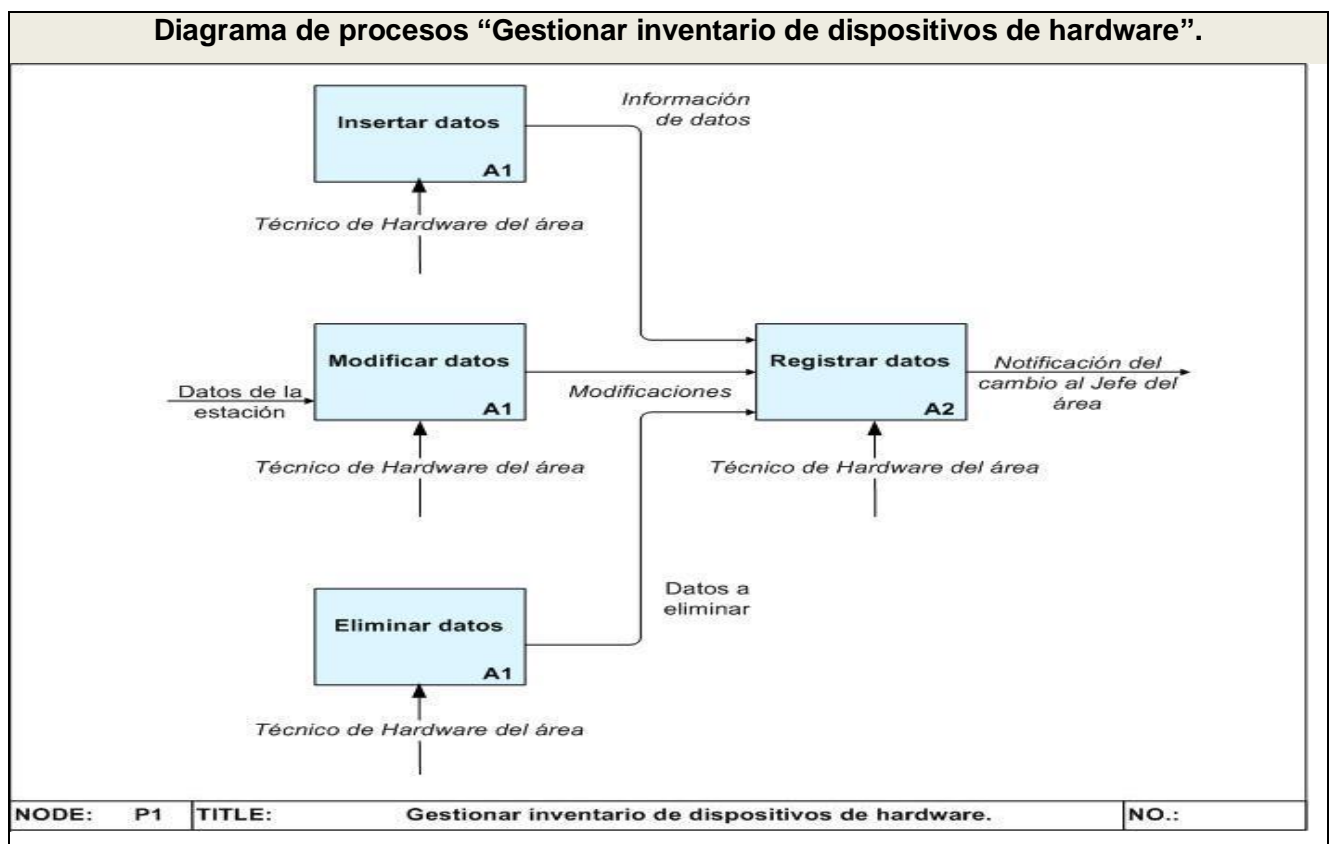


Figura 3: Diagrama de procesos "Gestionar inventario de dispositivos de hardware".

2.3.4 Descripción textual de los procesos de negocio definidos.

A las distintas áreas de los laboratorios se les entrega cada cierto período de tiempo un conjunto de medios informáticos, adjunto a estos reciben una planilla independiente por cada una de las estaciones de trabajo con información referente a cada dispositivo que la conforma.

Teniendo en cuenta lo anteriormente expresado, se describen del siguiente modo los procesos de negocio definidos:

Proceso "Gestión de datos de los dispositivos de hardware"	
Actividades	Descripción de la actividad
Insertar datos:	Puede verse desde dos ámbitos; el primero consistiría en introducir mediante la aplicación DATAPCLAB los datos de los dispositivos de hardware de las estaciones de trabajo incorporadas al área y en un segundo plano consistiría en la incorporación de un nuevo dispositivo de hardware a una estación de trabajo existente. Este proceso solo puede ser realizado por el técnico de hardware del área.
Eliminar datos:	En ocasiones se requiere extraer un dispositivo de forma definitiva de una estación de trabajo, por ya no ser requerido. Este proceso consiste en la eliminación de los datos de este dispositivo de hardware del servidor y solo puede ser realizado por el técnico de hardware del área.
Modificar datos:	Se inicia al presentarse un reemplazo de dispositivo de hardware en estaciones de trabajo, dado un reporte de desperfecto previamente realizado. Consiste en modificar los datos de los dispositivos de hardware de las estaciones de trabajo, solo puede ser realizado por el técnico de hardware del área.
Registrar datos:	Consiste en guardar la información manipulada en los procesos de inserción, modificación y eliminación anteriormente descritos. Este proceso es realizado por el técnico de hardware del área.

Tabla 3: Descripción del proceso "Gestión de datos de los dispositivos de hardware".

Proceso "Reportar estaciones de trabajo (CPU) defectuosos".	
Actividades	Descripción de la actividad
Realizar reporte general de desperfecto:	Consiste en reportar las estaciones de trabajo del área en cuestión, que se encuentran en mal estado o presentan algún problema en su funcionamiento, que proporcionalmente podría implicar una deficiencia en alguno de los dispositivos que componen la estación de trabajo. Esta acción es realizada y archivada en la aplicación GATSERVER y solamente realizada por el técnico de hardware del área.
Ejecutar reporte:	Es el acto de darle solución al reporte realizado. El mismo es ejecutado por el personal del grupo UCIXTEL.
Verificar estado del dispositivo y CPU:	Consiste en verificar el estado del posible dispositivo incorporado a la estación, como parte de la solución dada a un reporte de desperfecto previamente realizado, así como el del correcto funcionamiento del CPU.
Modificar estado de reporte:	Consiste en cambiar el estado del reporte a solucionado.

Tabla 4: Descripción del proceso "Reportar estaciones de trabajo (CPU) defectuosos".

2.4 Especificación de los requisitos de software.

Una especificación de Requerimientos de Software (SRS) define de forma precisa el producto de software que se va a construir, es una descripción completa del comportamiento del sistema a desarrollar. Incluye un conjunto de casos de uso que describen todas las interacciones que se prevean que los usuarios tendrán con el software.

2.4.1 Requerimientos funcionales.

Un requerimiento funcional puede ser una descripción de lo que un sistema debe hacer. Este tipo de requerimiento especifica algo que el sistema entregado debe ser capaz de realizar. Es decir, define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica.

Requerimientos funcionales del sistema a implementar:

R₁: Realizar reportes.

R_{1.1}: Realizar reporte de dispositivos de hardware defectuoso.

R_{1.2}: Realizar reporte de incidencia.

R₂: Actualizar estado de los reportes de desperfecto realizados.

R₃: Gestionar dispositivos de hardware.

R_{3.1}: Insertar datos de los dispositivos de hardware.

R_{3.2}: Modificar datos de los dispositivos de hardware.

R_{3.3}: Eliminar datos de los dispositivos de hardware.

R₄: Mostrar reportes.

R_{4.1}: Mostrar reportes en espera.

R_{4.2}: Mostrar reportes atendándose.

R_{4.3}: Mostrar reportes atendidos.

R₅: Mostrar Inventario.

R_{5.1}: Mostrar inventario general de dispositivos de hardware existentes en los laboratorios de la Universidad.

R_{5.2}: Mostrar inventario general de dispositivos de hardware, por área.

R_{5.3}: Mostrar inventario específico de cada tipo de dispositivo de hardware, por área.

R_{5.4}: Mostrar inventario general de los dispositivos de hardware ubicados en el almacén del área.

R_{5.5}: Mostrar inventario general de los dispositivos de hardware por laboratorios.

R_{5.6}: Mostrar inventario específico de cada tipo de dispositivo de hardware por laboratorios.

R_{5.7}: Mostrar inventario de los dispositivos de hardware de una estación de trabajo específica.

R_{5.8}: Mostrar inventario específico de un tipo dispositivo de hardware de una estación de trabajo.

R₆: Mostrar mensajes de confirmación, dada la ejecución de algún almacenamiento.

R₇: Controlar el nivel acceso de los usuarios.

R₈: Actualizar internamente el estado de reportes de desperfecto de un CPU dada las modificaciones que se le realicen.

2.4.2 Requerimientos no funcionales:

Los requerimientos no funcionales, son propiedades o cualidades que el producto debe tener, están estrechamente relacionados a los requerimientos funcionales y marcan la diferencia entre un producto aceptado y uno con poca aceptación. Permiten a clientes y usuarios valorar las características no funcionales del producto, usabilidad, seguridad entre otras categorías especificadas a continuación.

Estos requerimientos fueron definidos a nivel de proyecto.

Requerimientos de Seguridad:

- La información será almacenada en una base de datos, dejando registradas todas las operaciones mediante copias de seguridad.
- El uso y manejo del sistema estará controlado; ya que la información podrá ser consultada y modificada solamente por el personal autorizado; estableciendo para ello una serie de roles con funcionalidades específicas.
- La aplicación estará funcionando a tiempo completo; de esta forma es posible que los usuarios tengan acceso (según sus permisos) en todo momento a la información solicitada.
- Es muy importante que los servidores estén sincronizados correctamente con el día y hora, ya que si ocurre un fallo el sistema no funcionaría correctamente.

Requerimientos de Confiabilidad:

- Todas las salidas del sistema tendrán 100% de veracidad y precisión.
- El sistema tendrá la capacidad de recuperarse rápidamente ante cualquier fallo mediante las copias de seguridad que serán realizadas.
- Se verificará constantemente la conexión con el servidor y en caso de fallo se le notificará al usuario.
- En caso de alguna dificultad con el funcionamiento del sistema, el tiempo medio de reparación deberá ser menor de un día.

Requerimientos de Rendimiento:

- El sistema debe tener un tiempo de respuesta rápido y eficiente, inferior a diez segundos.
- La aplicación debe ser capaz de soportar gran cantidad de usuarios conectados simultáneamente.

Requerimientos de apariencia o interfaz externa:

- La interfaz será agradable, sencilla y sugerente; ya que su diseño simulará una aplicación de escritorio, presentará por defecto un color azul claro y textos que muestran la acción a realizar con un tipo de letra Verdana de tamaño 10pt.
- Se le brindará al usuario la posibilidad de personalizar la apariencia de interfaz externa.
- Debe contener pocas imágenes y gráficos para acelerar la velocidad de respuesta del sistema; por lo que se emplearán iconos sencillos que representarán la operación a realizar.
- El sistema será diseñado para una resolución de pantalla de 1024 X 768 píxeles.

Requerimientos de Hardware:

- Se requiere tanto para la estación cliente como para el servidor mínimo una PC con procesador Pentium IV o compatible.
- Se necesita además en la estación cliente una tarjeta de red de 10/100 MB/s para conexión con el servidor.
- Para el servidor como mínimo se necesita 256 MB de memoria RAM, 60 GB de espacio libre en disco duro para almacenamiento y tarjeta de red de 10/100 MB/s para la conexión.

Requerimientos de software:

- Las estaciones de trabajo clientes utilizarán como sistema operativo GNU/Linux en sus diversas distribuciones o la familia de Windows superior a Windows 98.
- Las máquinas clientes contarán con un navegador, ya sea el Internet Explorer en versiones superiores a la 5.0 o Mozilla Firefox en versiones superiores a la 1.5, ambos con las opciones de JavaScript y cookies habilitadas.
- El servidor Web funcionará sobre el sistema operativo GNU/Linux y tendrá instalado el Apache en su versión 2.0 y PHP 5.2 con las librerías PDO, PGSQL y JSON habilitadas.
- El servidor de base de datos funcionará sobre el sistema operativo GNU/Linux y tendrá instalado el gestor de bases de datos PostgreSQL 8.2.

Requerimientos de Usabilidad:

- La interfaz será fácil de usar para los diversos usuarios que interactúen con ella.

Requerimientos de Soporte:

- El sistema contará con la documentación apropiada para agilizar su Mantenimiento y Configuración.
- Se ofrecerán servicios de adiestramiento al personal que trabajará con el software.

Requerimientos de Ayuda y documentación en línea:

- El sistema proporcionará en todo momento la documentación necesaria para que los usuarios puedan acceder a la misma en caso de que necesiten realizar alguna consulta.

Requerimientos legales:

- Los módulos de desarrollo del proyecto SIGLA así como la documentación del mismo, son propios de la UCI y específicamente del área de los laboratorios.

2.5 Modelo del sistema.

2.5.1 Actores del sistema.

Actores del sistema	Justificación
Especialista (Especialista general de hardware, Técnico de hardware del área).	Individuo responsable de controlar los dispositivos de hardware de cada estación de trabajo asignada a su área de trabajo y visualizar la información del inventario de estos dispositivos.
Técnico (Especialista, Jefe de área, Jefe de turno, Técnico de laboratorio).	Individuo calificado para visualizar la información del inventario de dispositivos de hardware de su ubicación de trabajo. Siendo estas ubicaciones de trabajo en dependencia al tipo de técnico que acceda al sistema, un laboratorio, un área o las áreas de los laboratorios de la UCI.
Técnico de hardware del área.	Individuo encargado de la gestión de inventarios de dispositivos de hardware (insertar, eliminar o modificar dispositivos de hardware) y de la gestión de reportes de estaciones de trabajo con posibles fallas de sus dispositivos de hardware internos, en su área de trabajo. Es considerado también como un especialista.

Tabla 5: Descripción de los actores del sistema.

2.5.2 Diagrama de casos de usos a automatizar.

Un caso de uso representa un gránulo funcional del sistema bajo análisis, relatado como una secuencia de acciones que uno o más actores llevan a cabo en el sistema para obtener un resultado de valor significativo.

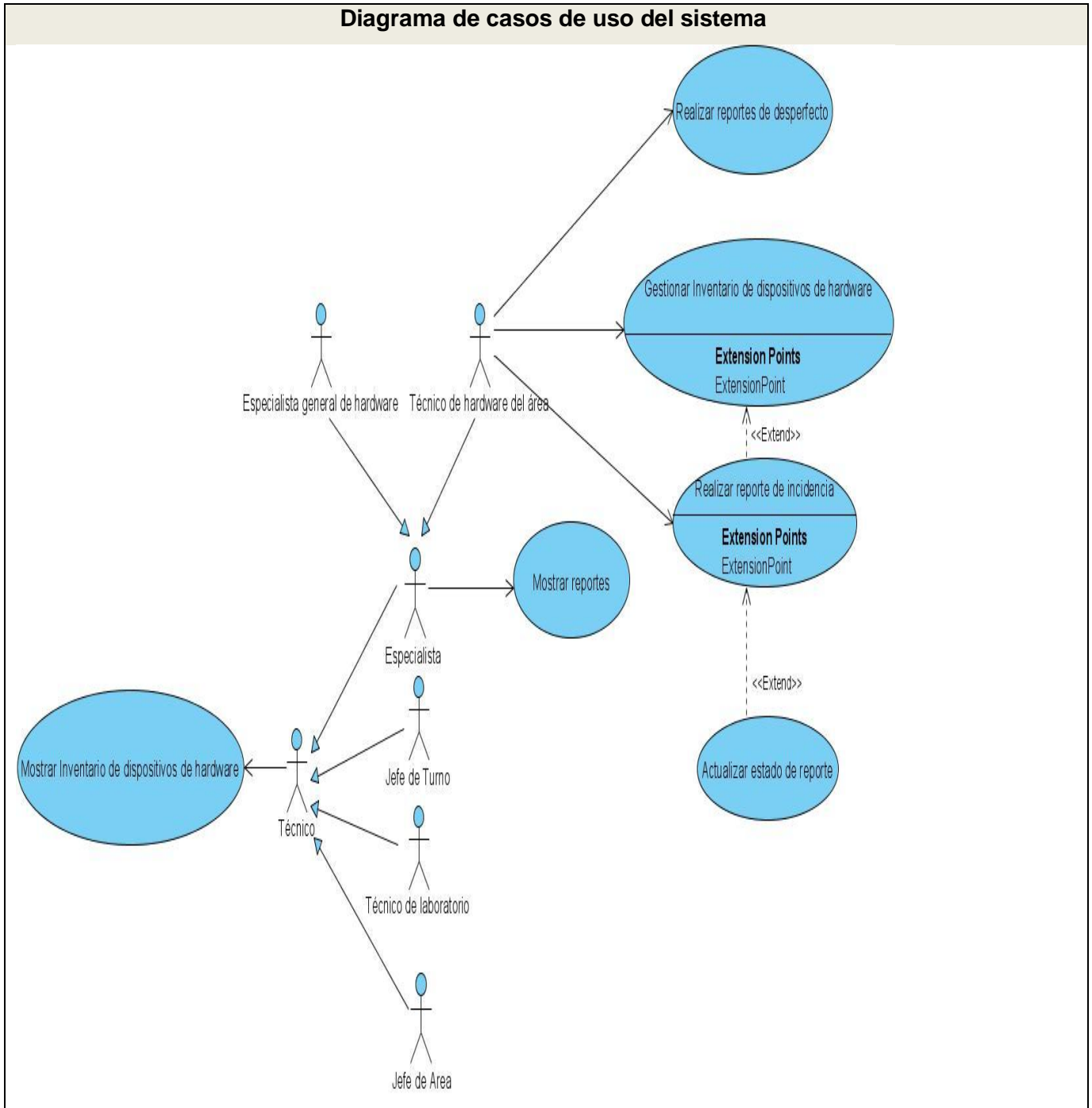
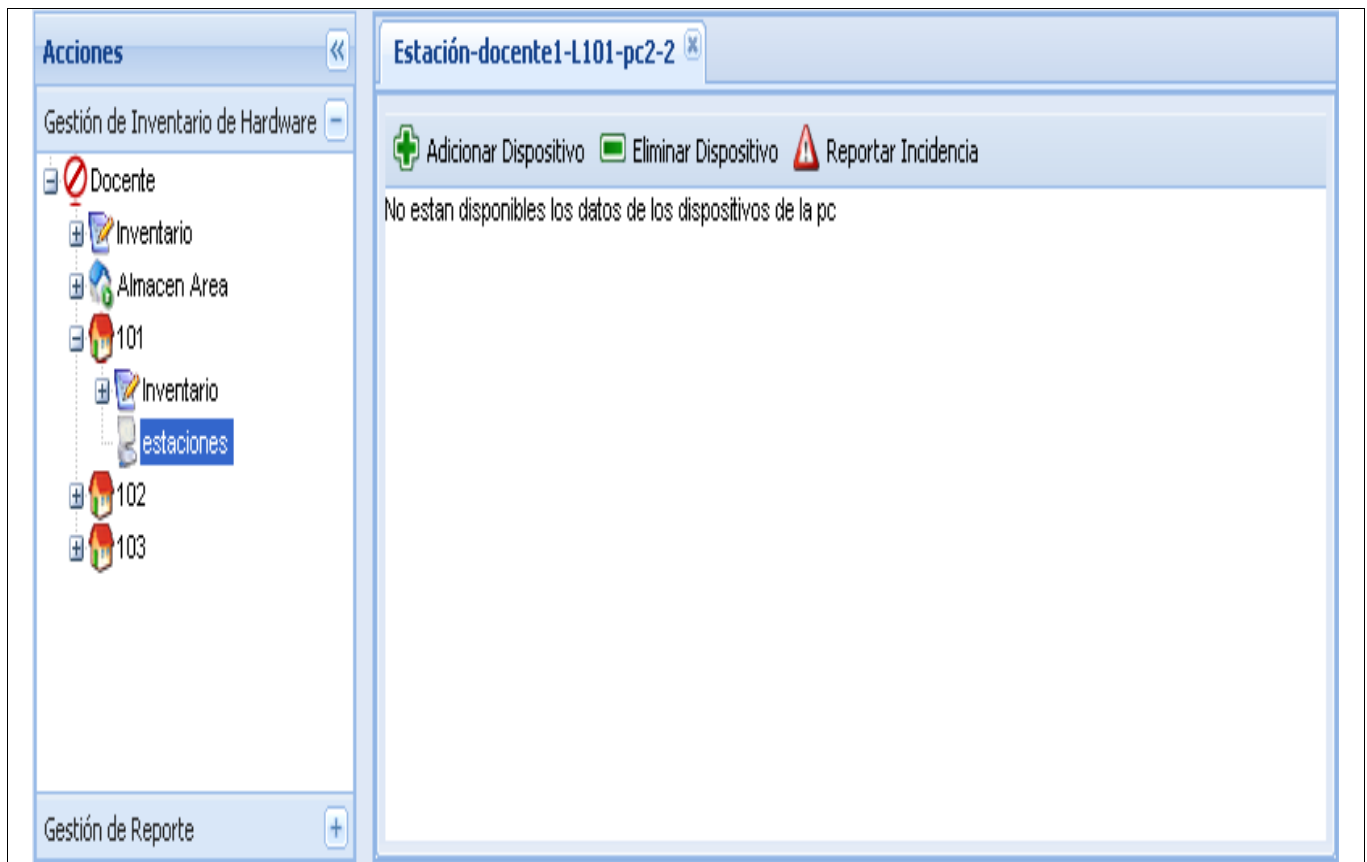


Figura 4: Diagrama de casos de uso del sistema.

2.5.3 Descripción de los casos de usos del sistema (CUS).

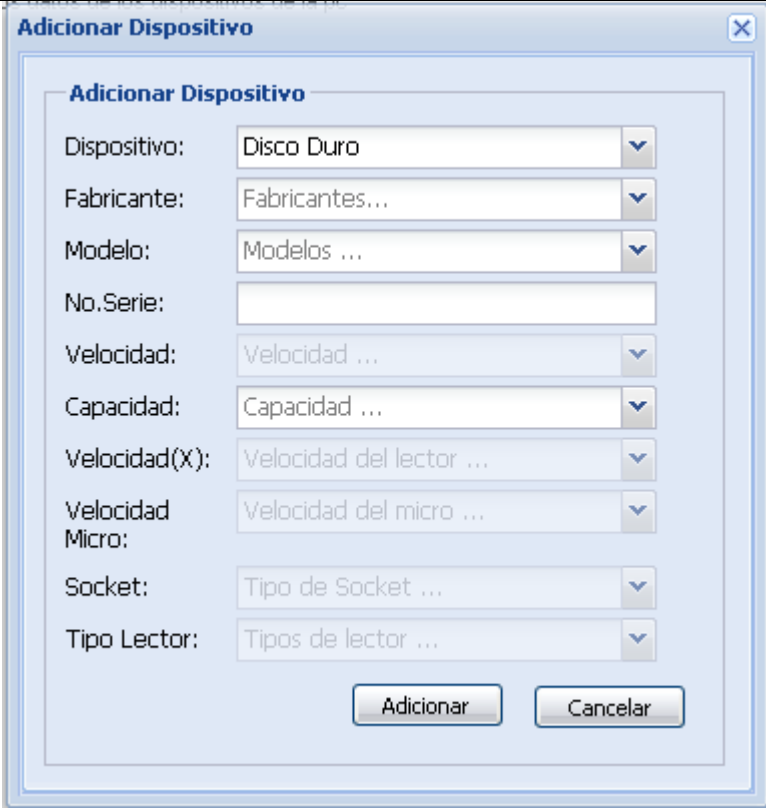
Caso de Uso:	Gestionar inventario de dispositivos de hardware.	
Actor:	Técnico de hardware del área (iniciador).	
Resumen:	<p>El caso de uso consta de las operaciones “insertar, modificar y eliminar” datos de dispositivos de hardware de estaciones de trabajo. Las operaciones siguen un mismo flujo de eventos, solamente se diferencian en el modo de ejecutar la acción a realizar. Pueden realizarse ya sea en un laboratorio o en el almacén del área correspondiente. No se puede modificar un dispositivo de hardware que no haya sido anteriormente insertado.</p> <p>Entendemos en el caso de la inserción, realizar el llenado digital de los datos de los dispositivos de hardware de un estación de trabajo; en el caso de la modificación, a cambiar los datos de los dispositivos de hardware de las estaciones de trabajo, dado un reporte de desperfecto previamente realizado a la misma; por eliminar se entiende a la necesidad de retirar un dispositivo que físicamente ya no formará parte de una estación de trabajo. Las operaciones anteriormente descritas solo pueden ser realizadas por el Técnico de hardware del área.</p>	
Precondiciones:	El Técnico de hardware del área ya se auténtico y se encuentra trabajando en el control de hardware.	
Referencias	R3, R3.1, R3.2, R3.3, R7.	
Casos de uso relacionados	CUS: Realizar reporte de incidencia(extendido)	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del sistema	
1. Selecciona el panel “Gestión de Inventarios de Hardware”.	2. El sistema despliega un árbol con los diversos locales del área de los laboratorios a los cuales el actor tiene acceso.	
3. Selecciona y despliega, el laboratorio (local) donde desea realizar operaciones de inserción, modificación o eliminación de datos de dispositivos	4. El sistema muestra un árbol de opciones posibles de ser seleccionadas por el usuario en dicho local.	

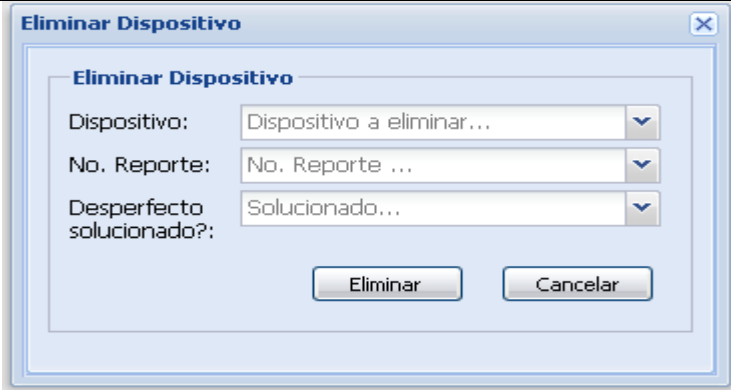
de hardware.	
5. Selecciona la opción “estaciones ”	6. Visualiza un panel con las estaciones de trabajo pertenecientes al local.
7. Selecciona la estación donde desea ejercer las operaciones.	8. Muestra un panel donde aparecen los diversos dispositivos de hardware que posee la estación de trabajo. Sobre el panel estarán activadas las opciones para realizar la operación que el usuario desee ejecutar.
9. La secuencia de pasos depende de la operación que el actor desee realizar: Insertar (Ir a sección insertar). Modificar (Ir a sección modificar). Eliminar (Ir a sección eliminar).	
Flujos Alternos	
3. Selecciona y despliega, el almacén de su área (local).	
Prototipo interfaz	



Sección Insertar

Acción del Actor	Respuesta del sistema
	<ol style="list-style-type: none"> 1. El sistema muestra un formulario con los posibles dispositivos que puedan ser insertados a la estación de trabajo.
<ol style="list-style-type: none"> 2. Selecciona e Inserta los datos del nuevo dispositivo de hardware. 	
<ol style="list-style-type: none"> 3. Selecciona el botón “adicionar” del formulario. 	<ol style="list-style-type: none"> 4. Chequea la validez de los datos introducidos.
	<ol style="list-style-type: none"> 5. Procesa la información en caso de ser válida la entrada de datos.
	<ol style="list-style-type: none"> 6. Invoca el CUS: Realizar reporte de incidencia.

7. Selecciona otra operación a realizar.	
Flujos Alternos	
3. Selecciona el botón “cancelar” del formulario o cierra el formulario. El sistema Muestra un panel donde aparecen los diversos dispositivos de hardware que posee la estación de trabajo. Sobre el panel estarán activadas las opciones para realizar la operación que desee ejecutar	
5. Emite una alerta de invalidación de los valores entrados por el usuario. Muestra el formulario con los errores contenidos en él y le da la posibilidad de reingresar los parámetros	
7. Cierra la aplicación.	
Prototipo interfaz	
	
Sección Eliminar	
Acción del Actor	Respuesta del sistema

	1. Muestra un formulario con los dispositivos de hardware de la estación de trabajo, posibles de eliminar.
2. Selecciona e inserta, los datos solicitados en el formulario.	
3. Selecciona el botón “eliminar”.	4. Chequea la validez de los datos introducidos.
	5. Procesa la información en caso de ser válida la entrada de datos.
	6. Invoca el CUS: Realizar reporte de incidencia.
7. Selecciona otra operación a realizar.	
Flujos Alternos	
Acción del Actor	Respuesta del sistema
2. Selección el botón “cancelar” o cierra el formulario. El sistema Muestra un panel donde aparecen los diversos dispositivos de hardware que posee la estación de trabajo. Sobre el panel estarán activadas las opciones para realizar la operación que desee ejecutar.	
5. Emite una alerta de invalidación de los valores entrados por el usuario. Muestra el formulario con los errores contenidos en él y le da la posibilidad de reingresar los parámetros.	
7. Cierra la aplicación.	
Prototipo interfaz	
	
Sección Modificar	
Acción del Actor	Respuesta del sistema
1. Selecciona el dispositivo cuyos datos desea	2. Muestra un formulario con los datos del


modificar.	dispositivo de hardware seleccionado.
3. Ingresa y selecciona los datos para modificar al dispositivo de hardware en cuestión.	
4. Selecciona el botón “modificar”.	5. Chequea la validez de los datos introducidos.
	6. Procesa la información en caso de ser válida la entrada de datos.
	7. Invoca el CUS: Realizar reporte de incidencia.
8. Selecciona una operación que desee realizar.	
Flujos Alternos	
Acción del Actor	Respuesta del sistema
4. Selecciona el botón “cancelar”. El sistema Muestra un panel donde aparecen los diversos dispositivos de hardware que posee la estación de trabajo. Sobre el panel estarán activadas las opciones para realizar la operación que desee ejecutar.	
6. Emite una alerta de invalidación de los valores entrados por el usuario. Muestra el formulario con los errores contenidos en él y le da la posibilidad de reingresar los parámetros. Realiza el flujo 9.	
8. Cierra la aplicación.	
Prototipo interfaz	
	

Tabla 6: Descripción del CUS “Gestionar inventario de dispositivos de hardware”.

Caso de Uso:	Realizar reporte de incidencia.
---------------------	---------------------------------

Actor:	Técnico de hardware del área (posible iniciador), CUS: Gestionar Inventario de dispositivos de hardware (caso de uso base).
Resumen:	El caso de uso se puede ver desde dos ámbitos distintos. El primero: si es inicializado por el caso de uso base, el sistema genera internamente un reporte con los cambios realizados a una estación de trabajo, es decir, dada una nueva inserción, eliminación o modificación de datos de los dispositivos de hardware que la componen. El segundo: es inicializado por el Técnico de hardware del área, debido a que la incidencia realizada a una estación de trabajo fue por sello (apertura o limpieza de la estación) o por solución de un reporte de desperfecto. Si la incidencia es realizada dado un reporte de desperfecto el actor deberá seleccionar entre los datos que se requieren introducir, el número del reporte de desperfecto al cual responde; de lo contrario solo introduce la descripción del suceso.
Precondiciones:	Debe haberse realizado con anterioridad algún proceso del CUS Gestionar Inventario de dispositivos de hardware.
Caso de uso relacionados:	CUS: Gestionar Inventario de dispositivos de hardware. CUS: Actualizar estado de reporte (extendido).
Referencias	R1, R1.2, R6, R7.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
	<ol style="list-style-type: none"> 1. En dependencia al actor que lo inicializa : <ol style="list-style-type: none"> a. Caso de uso base .Ir a sección CUB. b. Técnico de hardware del área. Ir a sección Técnico
Sección CUB	
Acción del Actor	Respuesta del sistema
	<ol style="list-style-type: none"> 1. El sistema guarda los cambios que se realizaron como un archivo de modificación y crea un reporte de incidencia basándose en la información

Características del sistema

	contenida en dicho archivo.
	2. En caso de haberse guardado correctamente el reporte creado , el sistema emite un mensaje del correcto almacenamiento del mismo
	3. Invoca el CUS: Actualizar estado de reporte
Flujos Alternos	
2. Si el reporte no fue creado correctamente, el sistema emite una alerta, informando que no se pudo realizar correctamente el proceso de gestión de inventario ejecutado.	
Sección Técnico	
Acción del Actor	Respuesta del sistema
1. Selecciona el panel “Gestión de Inventarios de Hardware”.	2. El sistema despliega un árbol con los diversos locales del área de laboratorios a los cuales el Técnico de hardware del área tiene acceso.
3. Selecciona y despliega, el laboratorio (local), donde esta ubicada la estación a la cual le fue realizada una incidencia.	4. El sistema muestra un árbol de opciones posibles de ser seleccionadas por el usuario en dicho local.
5. Selecciona la opción “estaciones ”	6. Visualiza un panel con las estaciones de trabajo pertenecientes al local.
6. Selecciona la estación de trabajo en cuestión.	7. Muestra un panel donde aparecen los diversos dispositivos de hardware que posee la estación de trabajo. Sobre el panel estará activada la opción a ejecutar.
8. Selecciona el botón “Reportar incidencia”	9. Muestra un formulario con los datos ha ser seleccionados e introducidos por el usuario para poder realizar el reporte de incidencia.
10. Selecciona e introduce los datos solicitados.	
11. Selecciona el botón “aceptar” del formulario.	12. Verifica la validez de los datos introducidos.

	<p>13. Procesa la información en caso de ser válida la entrada de datos.</p>
	<p>14. Invoca el CUS: Actualizar estado de reporte, si el reporte de incidencia en curso responde a un reporte de desperfecto.</p>
<p>15. Selecciona otra operación a realizar.</p>	

Flujos alternos

<p>11. Selecciona el botón “cancelar”. El sistema ejecuta automáticamente el flujo 7.</p>
<p>14. Crea la incidencia y la almacena.</p>
<p>13. Emite una alerta de invalidación de los valores entrados por el usuario. Muestra el formulario con los errores contenidos en él y le da la posibilidad de reingresar los parámetros. Realiza el flujo 9.</p>
<p>15. Cierra la aplicación</p>

Prototipo interfaz

The image displays two instances of a 'Nueva Incidencia' (New Incident) dialog box. The left window shows the 'Desperfecto' (Defect) dropdown menu expanded, with options for 'No. Reporte' (Report No.) and 'Desperfecto solucionado?' (Defect solved?). The right window shows the 'Descripcion' (Description) text area. Both windows have 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons at the bottom.

Tabla 7: Descripción del CUS “Realizar reporte de incidencia”.

Caso de Uso:	Realizar reporte de desperfecto.	
Actor:	Técnico de hardware del área (iniciador).	
Resumen:	El caso de uso consiste en realizar un reporte, donde se especifican los problemas o desperfectos de las estaciones de trabajo existentes en el entorno de trabajo del Técnico de hardware del área, este tipo de reporte es individual para cada estación de trabajo.	
Precondiciones:	El Técnico de hardware del área ya se autenticó y se encuentra trabajando en el control de hardware.	
Referencias	R1, R1.1, R1.2, R9, R10.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del sistema	
1. Selecciona el panel “Gestión de reportes”.	2. Muestra un árbol de opciones de reportes.	
3. Selecciona la opción “Nuevo reporte”.	4. El sistema muestra el formulario correspondiente al reporte de desperfecto.	
5. Selecciona e Introduce los datos.		
6. Selecciona la opción “Crear reporte”.	7. Verifica la validez de los datos introducidos.	
	8. Procesa la información y emite una alerta del correcto almacenamiento del archivo.	
9. Cierra la aplicación.		
Flujos Alternos		
Acción del Actor	Respuesta del sistema	
6. Selecciona el botón “cancelar”.	El sistema ejecuta automáticamente el flujo 2.	
8. Emite una alerta de invalidación de los valores entrados por el usuario. Muestra la plantilla con los errores contenidos en ella, ejecuta el flujo 5.		
9. Selecciona otra operación a realizar.		
Prototipo interfaz		

Tabla 8: Descripción del CUS “Realizar reporte de desperfecto”.

Caso de Uso:	Actualizar estado de reporte.
Actor:	CUS “Realizar reporte de incidencia.”(Caso de uso base).
Resumen:	El caso de uso se inicia luego de ser realizado su caso de uso base, consiste en actualizar el estado de los reportes de desperfecto, cambiarlos de un estado de espera a atendiéndose o a atendido, o de atendiéndose a atendido.
Precondiciones:	Debe haberse realizado con anterioridad el CUS “Realizar reporte de incidencia”.
Caso de uso relacionados:	CUS “Realizar reporte de incidencia”.
Referencias	R6, R8.

Flujo Normal de Eventos

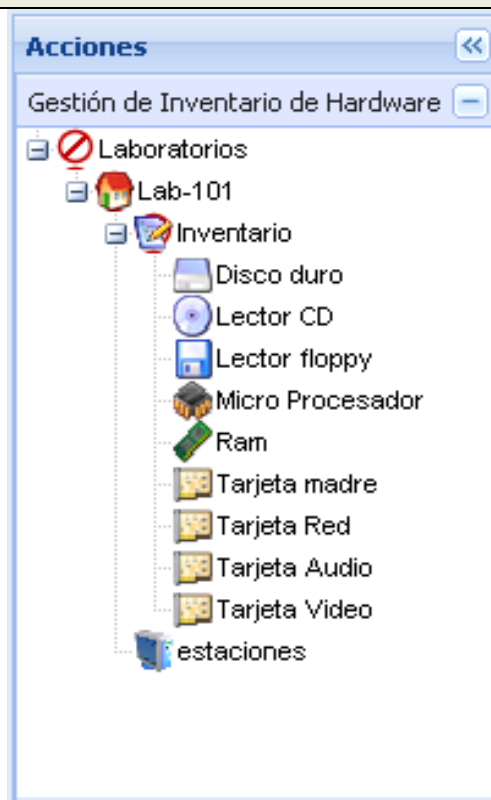
Acción del Actor	Respuesta del sistema
	<p>1. El sistema cambia el estado del reporte de desperfecto, de “Reporte en espera”, a uno de los siguientes estados.</p> <ul style="list-style-type: none"> • Reporte atendiéndose. • Reporte atendido. <p>O de a “Reporte atendiéndose” a “Reporte atendido”.</p>

Tabla 9: Descripción del CUS “Actualizar estado de reporte.”

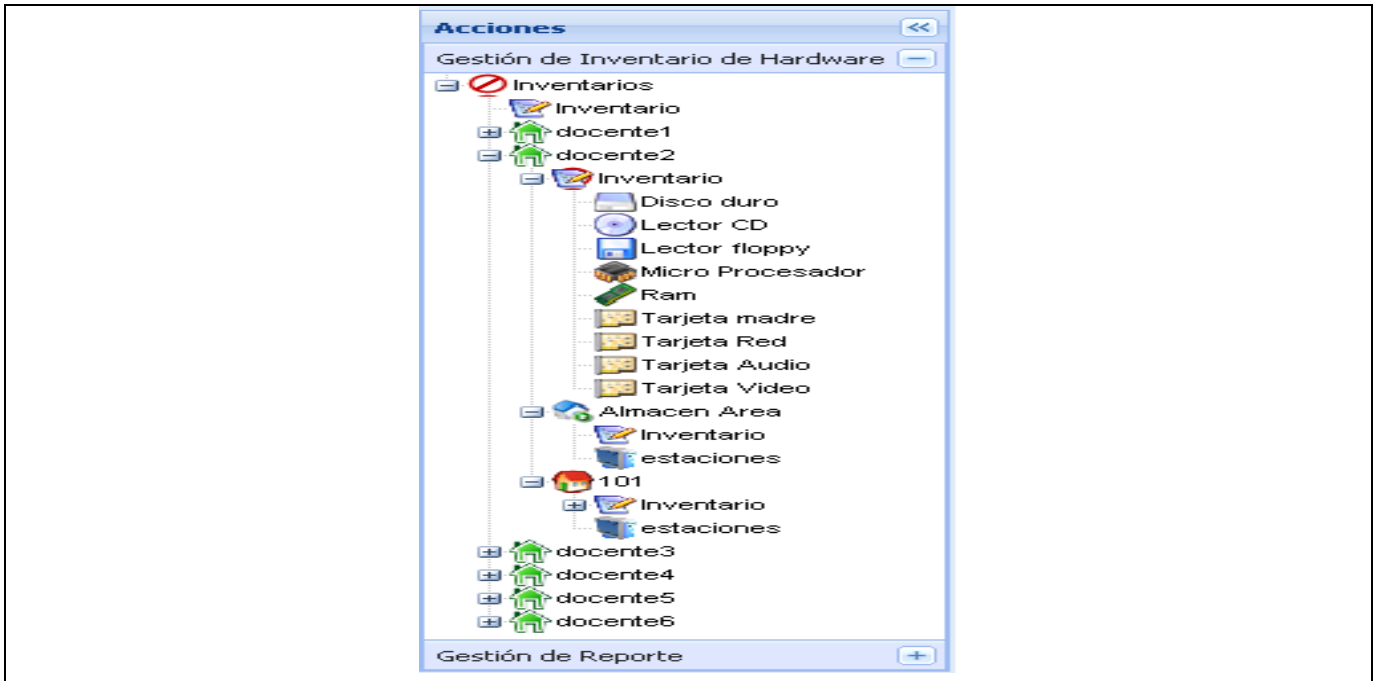
Caso de Uso:	Mostrar inventario de dispositivos de hardware.
Actor:	Técnico (iniciador), Especialista general de hardware, Técnico de hardware del área, Jefe de turno, Jefe de área, Técnico de laboratorio.
Resumen:	<p>El caso de uso consiste en visualizar al Técnico, el inventario de dispositivos de hardware desde el más alto nivel de acceso que tiene permisible, hasta el inventario específico y las propiedades de un dispositivo en una estación de trabajo.</p> <p>Para un Técnico de laboratorio, el nivel más alto de visualización de inventario es el laboratorio o los laboratorios donde trabaja. Para un Jefe de turno, Jefe de área y un Técnico de hardware del área, el nivel más alto de visualización son todos los laboratorios de su área laboral y para un especialista de hardware general son todas las áreas de la UCI. El sistema en dependencia del actor que accede, mantendrá un control de las visualizaciones que le son permitidas.</p>
Precondiciones:	El Técnico se autenticó y se encuentra trabajando en el control de hardware.
Referencias	R5, R5.1, R5.2, R5.3, R5.4, R5.5, R5.6, R5.7, R5.8, R7, R8.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
	<p>1. En dependencia al tipo de técnico autenticado:</p> <p>a. Técnico de laboratorio .Ir a sección Técnico de laboratorio.</p> <p>b. Jefe de turno. Ir a sección Área.</p> <p>c. Jefe de área. Ir a sección Área.</p> <p>d. Técnico de hardware del área. Ir a sección Encargado del área.</p> <p>e. Especialista general de hardware. Ir a sección Especialista general de hardware.</p>
Sección Técnico de laboratorio	
Acción del Actor	Respuesta del sistema

	<p>1. El sistema despliega un árbol con todas las opciones de acceso que le son permisibles al usuario.</p>
<p>2. El usuario desea visualizar alguna de las siguientes opciones:</p> <ul style="list-style-type: none">a. El inventario del laboratorio asignado. Ir a sección Inventario.b. El inventario existente en una estación de trabajo ubicada en el laboratorio. Ir a sección Estaciones.c. Las propiedades de uno de los dispositivos existentes en una estación de trabajo. Ir a sección Dispositivo.	

Prototipo Interfaz

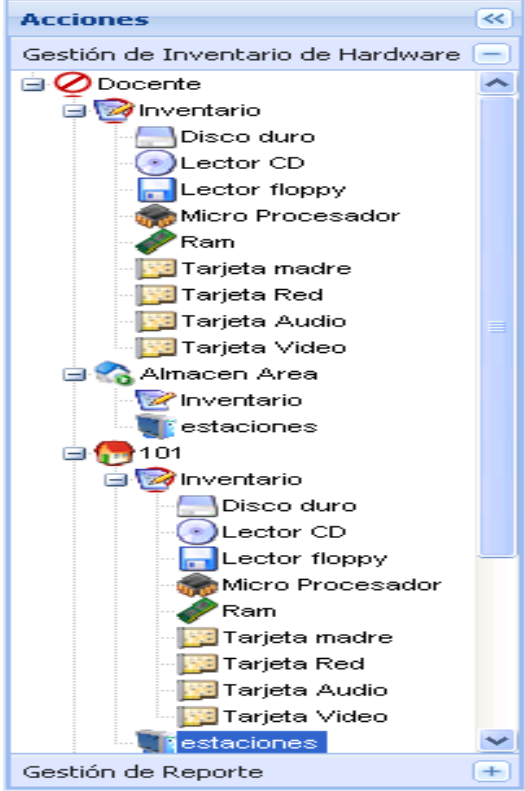


Sección Especialista general de hardware	
Acción del Actor	Respuesta del sistema
	1. El sistema despliega un árbol con todas las opciones de acceso que le son permisibles al usuario.
2. El usuario desea visualizar alguna de las siguientes opciones: <ul style="list-style-type: none">a. Inventario general existente en la UCI. Ir a sección Inventario general.b. Inventario de uno de los almacenes de las distintas áreas de los laboratorios de la UCI. Ir a sección Almacénc. Inventario existente en una de las áreas. Ir a sección Área.d. Inventario existente en un laboratorio, de una de las áreas. Ir a sección Laboratorio.e. Inventario existente en una estación de trabajo ubicada ya sea en un almacén o un laboratorio. Ir a sección Estaciones.f. Las propiedades de uno de los dispositivos existentes en una estación de trabajo. Ir a sección Dispositivo.	
Prototipo Interfaz	



Sección Encargado del área.

Acción del Actor	Respuesta del sistema
	<p>1. El sistema despliega un árbol con todas las opciones de acceso que le son permisibles al usuario.</p>
<p>2. El usuario desea visualizar alguna de las siguientes opciones:</p> <ul style="list-style-type: none"> a. El inventario general del área. Ir a sección Inventario. b. El inventario existente en el almacén del área. Ir a sección Almacén. c. El inventario existente en uno de los laboratorios del área. Ir a sección Laboratorio. d. Inventario existente en una estación de trabajo ubicada ya sea en un almacén o un laboratorio. Ir a sección Estaciones. e. Las propiedades de uno de los dispositivos existentes en una estación de trabajo. Ir a sección 	

Dispositivo.	
Prototipo interfaz	
	
Sección Área	
Acción del Actor	Respuesta del sistema
1. Selecciona y despliega el área cuyo inventario desea visualizar.	2. El sistema despliega un árbol con todas las opciones accesibles del área, sus laboratorios e inventarios
3. Selecciona la opción "inventario" .Ir a sección Inventario.	
Sección Laboratorio	
Acción del Actor	Respuesta del sistema
1. Selecciona y despliega el laboratorio del área cuyo inventario desea visualizar.	2. El sistema despliega un árbol con todas las opciones accesibles del laboratorio.
3. Selecciona la opción "inventario". Ir a sección Inventario.	
Sección Almacén	

Acción del Actor	Respuesta del sistema
1. Selecciona y despliega la opción almacén.	2. El sistema despliega un árbol con todas las opciones accesibles del almacén.
3. Selecciona la opción "inventario". Ir a sección Inventario general.	
Sección Inventario	
Acción del Actor	Respuesta del sistema
1. Desea visualizar: <ul style="list-style-type: none"> a. El inventario general del local .Ir a sección Inventario general. b. El inventario general de algún tipo dispositivo del local. Ir a sección Inventario específico. 	
Sección Inventario general	
Acción del Actor	Respuesta del sistema
1. Selecciona la opción "Inventario".	2. El sistema muestra un panel con los datos generales de los dispositivos del local agrupados por su tipo.
Prototipo interfaz	

Dispositivo	Tipo	Fabricante	Modelo	No.Serie	Area	Laboratorio	Puesto
Tipo: Disco Duro (2 Dispositivos)							
	Disco Duro	ASUS	abc	555555555	docente1	103	pc1
	Disco Duro	ASUS	abc	000999	docente1	101	pc1
Tipo: Lector (1 Dispositivos)							
	Lector	Intel	dasd	12345	docente1	101	pc1
Tipo: Micro Procesador (4 Dispositivos)							
	Micro Proces	ASUS	abc	7890	docente1	101	pc2
	Micro Proces	ASUS	abc	5678	docente1	101	pc2
	Micro Proces	ASUS	abc	555666	docente1	101	pc1
	Micro Proces	ASUS	abc	4567	docente1	102	pc1
Tipo: Ram (5 Dispositivos)							
	Ram	AOPEN	wqe	65656	docente1	102	pc1
	Ram	AOPEN	wqe	000000	docente1	101	pc2
	Ram	Intel	abc	878	docente1	101	pc1
	Ram	Intel	abc	4567	docente1	101	pc2
	Ram	Intel	abc	1111222333	docente1	103	pc1
Tipo: Tarjeta Audio (1 Dispositivos)							
	Tarjeta Audio	Intel	sad	3343	docente1	101	pc2

Sección Inventario específico

Acción del Actor	Respuesta del sistema
1. Selecciona el botón de desglose de la opción "Inventario".	2. Despliega un árbol donde el inventario de dispositivos de hardware, estará agrupado por los tipos de dispositivos de hardware existente en el local.
3. Selecciona desde el desglose, el tipo de dispositivo de hardware cuyas propiedades desea visualizar.	4. Muestra un listado con la información referente al tipo de dispositivo de hardware seleccionado.

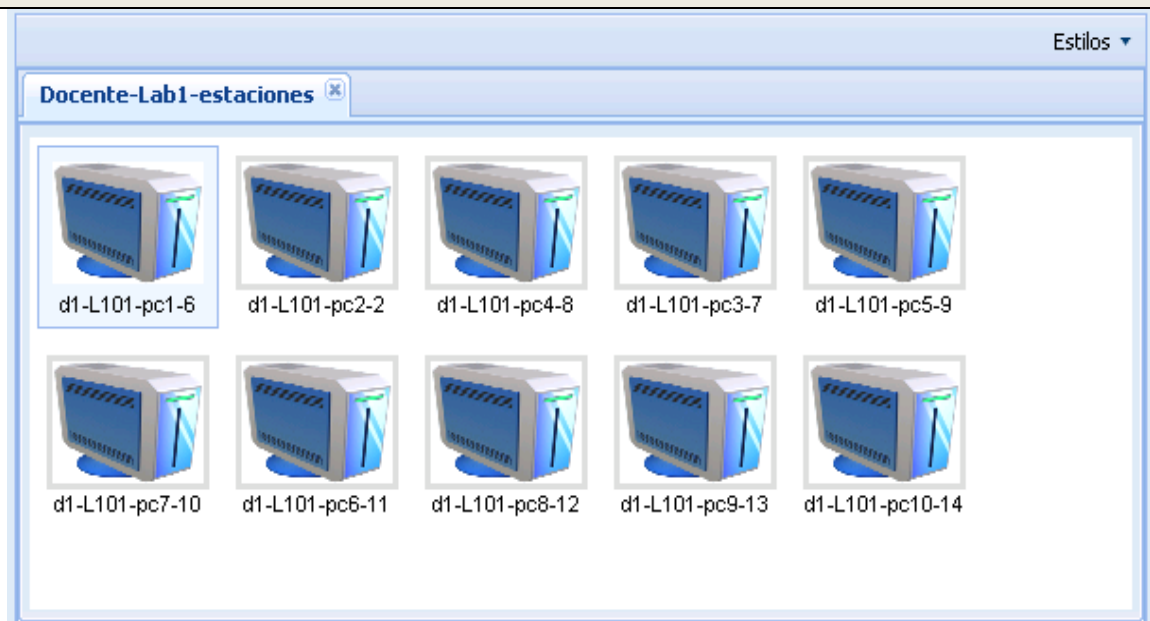
Prototipo interfaz

Dispositivo	Fabricante	Modelo	No.Serie	Area	Laboratorio
	Seagate	IDE	5678	docente1	101
	Seagate	IDE	uui	docente1	101
	Seagate	SATA	hgjhgighj	docente1	101

Sección Estaciones

Acción del Actor	Respuesta del sistema
1. Selecciona la opción “estaciones”.	2. El sistema muestra un panel con las estaciones de trabajo del local en cuestión.

Prototipo de Interfaz



Sección Dispositivo	
Acción del Actor	Respuesta del sistema
1. Selecciona la estación de trabajo.	2. El sistema muestra un panel con los dispositivos de hardware de la estación de trabajo seleccionada.
3. Selecciona el dispositivo cuyas propiedades desea visualizar.	4. El sistema despliega un panel, con las propiedades correspondientes al dispositivo seleccionado.

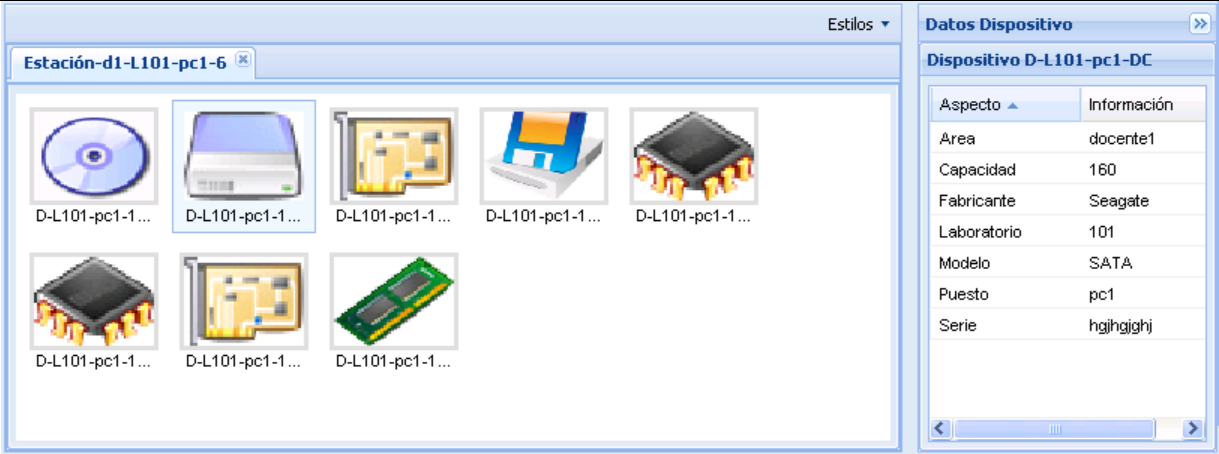
Prototipo de Interfaz


Tabla 10: Descripción del CUS “Mostrar inventario de dispositivos de hardware”.

Caso de Uso:	Mostrar reportes.
Actor:	Especialista (iniciador), Especialista general de hardware, Técnico de hardware del área.
Resumen:	<p>El caso de uso consiste en visualizar los diferentes tipos de reportes realizados, por nivel de atención. El Especialista general de hardware no puede ver los reportes que le fueron realizados al Técnico de hardware del área y viceversa. Ambos ven solamente los reportes que los relacionan en el trabajo. En el caso del especialista de hardware ve las incidencias del Técnico de hardware del área generadas por el sistema y también las creadas por el.</p> <p>El sistema en dependencia al actor que accede, mantiene un control de las</p>

	visualizaciones que le son permisibles.	
Precondiciones:	El especialista se autenticó y se encuentra trabajando en el control de hardware.	
Referencias	R4, R4.1, R4.2, R4.3, R7, R8.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del sistema	
1. Selecciona el panel "Gestión de reportes".	2. Despliega un árbol de opciones de reportes por categorías posibles de ser seleccionadas por el usuario.	
3. Selecciona la categoría "Reportes en espera".	4. El sistema muestra un panel con los reportes correspondientes a la categoría seleccionada.	
5. Selecciona uno de los reportes mostrados para obtener su información específica.	6. El sistema muestra la información del reporte seleccionado.	
7. Cierra la aplicación.		
Flujos Alternos		
Acción del Actor	Respuesta del sistema	
3. Selecciona la categoría "Reportes atendiéndose".		
3. Selecciona la categoría "Reportes atendidos".		
3. Selecciona la categoría "Reportes de incidencia".		
7. Selecciona otra operación a realizar.		
Prototipo de Interfaz		

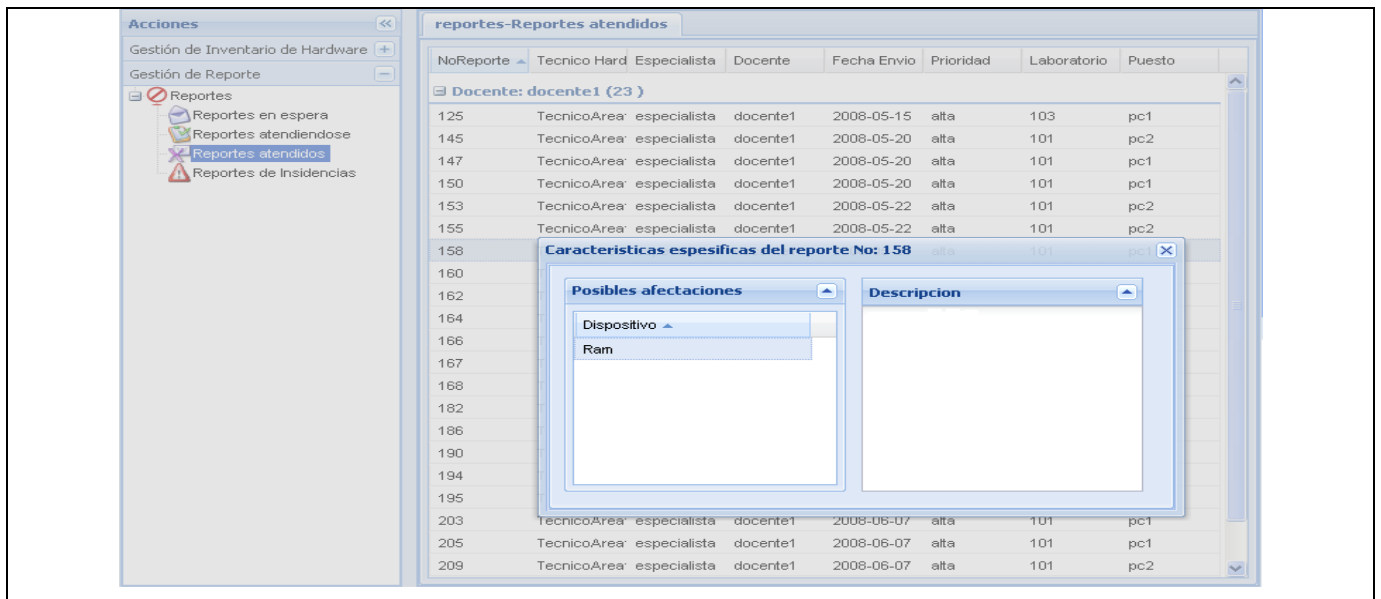


Tabla 11: Descripción del CUS "Mostrar reportes".

Conclusiones

En este capítulo se describieron los procesos y actividades que se realizan en el negocio planteado, definieron las posibles operaciones del sistema mediante casos de usos y trazaron las condiciones que el sistema deberá cumplir, sus propiedades y cualidades. Quedando de este modo, establecidas las características del sistema a implementar, lo que permite obtener una visión general de las funcionalidades que el mismo tendrá.

Capítulo #3: Análisis y diseño.

Introducción

En este capítulo, se analizarán los requerimientos definidos durante la caracterización del sistema. Mediante la realización de los diagramas de análisis y diseño, se modelará el sistema de modo que estos reflejen su estructura interna.

3.1 Análisis. Modelo de clases de análisis por casos de uso.

El análisis está estructurado por clases y paquetes mediante las cuales, esboza como llevar a cabo las funcionalidades dentro del sistema; lo que proporciona la estructura a la vista interna de este último. A continuación se mostrarán los diagramas del modelo de clases del análisis para cada caso de uso del sistema definido.

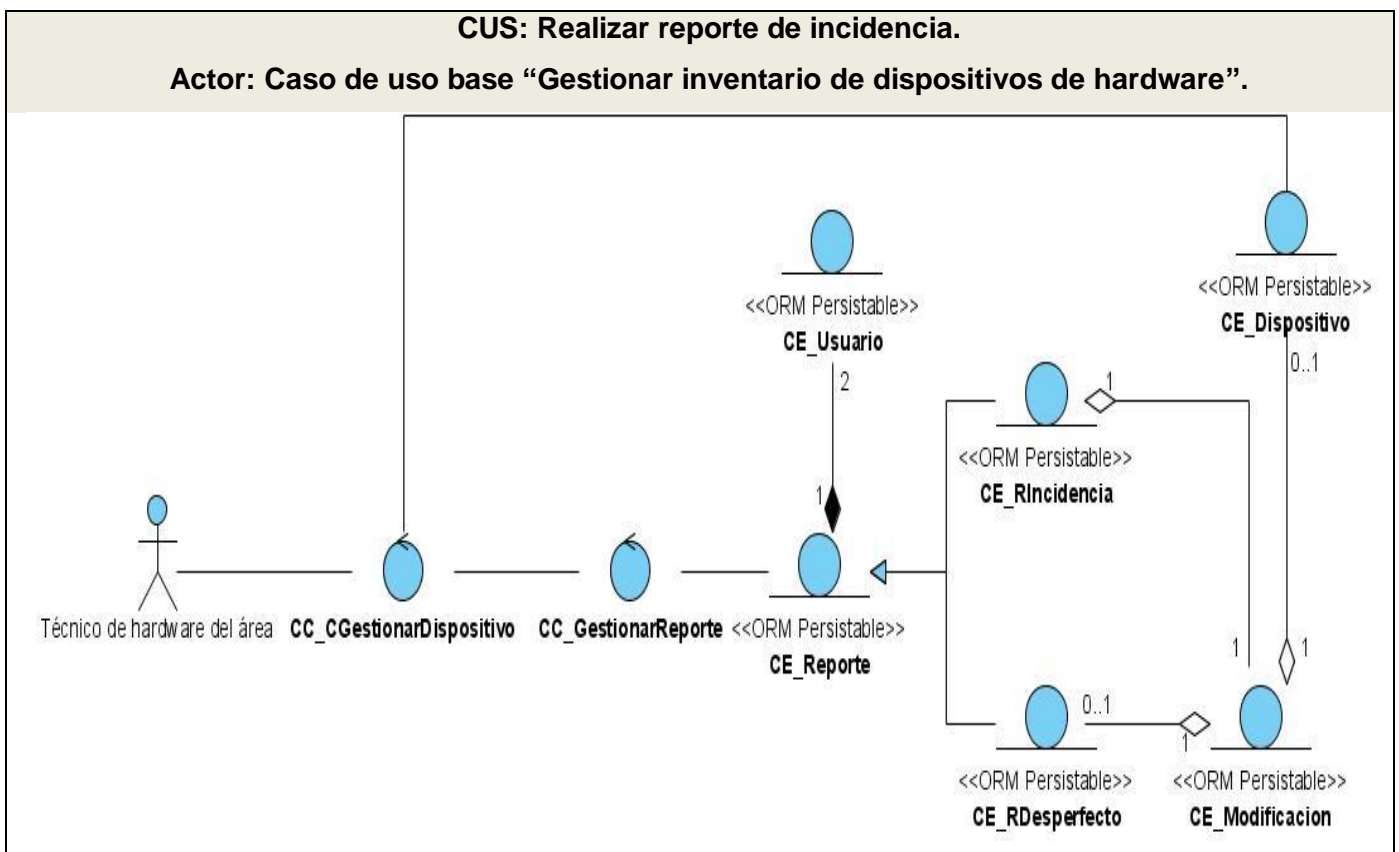


Figura 5: Diagrama de clases del CUS “Realizar reporte de incidencia”.

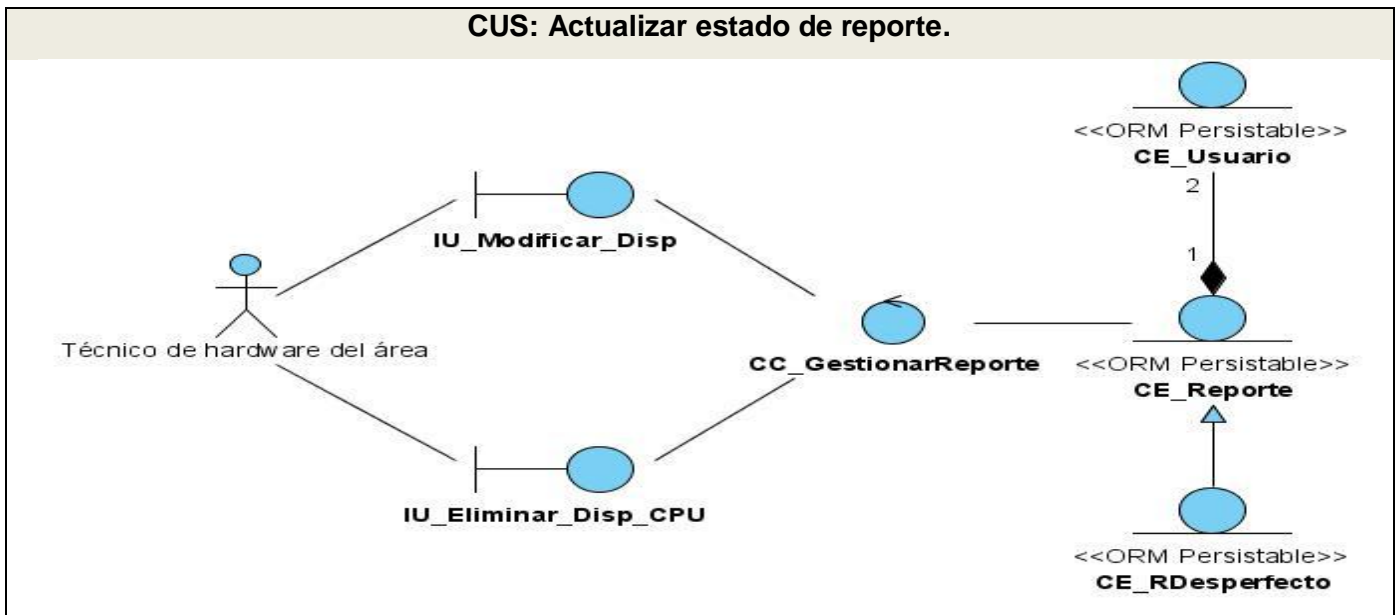


Figura 6: Diagrama de clases del CUS “Actualizar estado de reporte”.

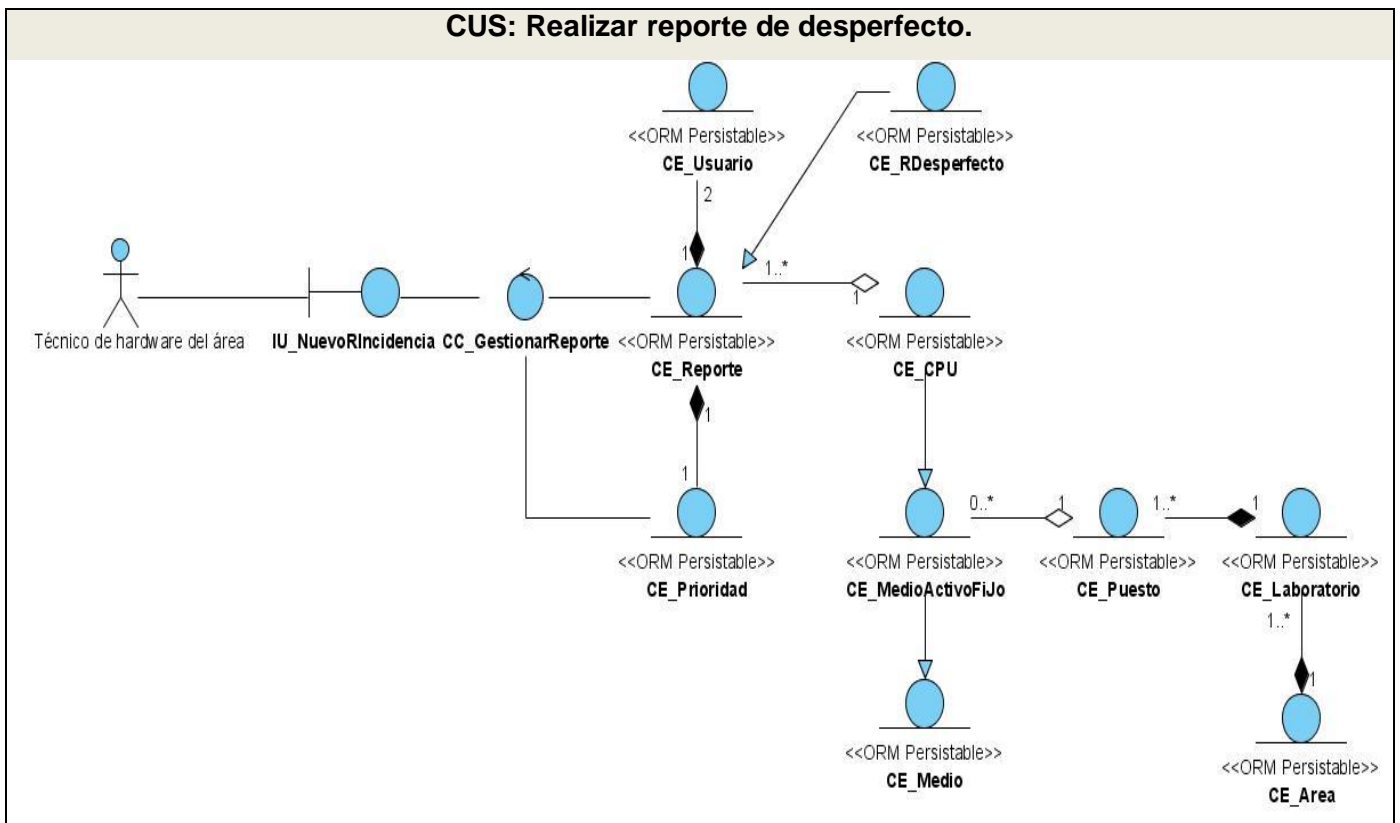


Figura 7: Diagrama de clases del CUS “Realizar reporte de desperfecto”.

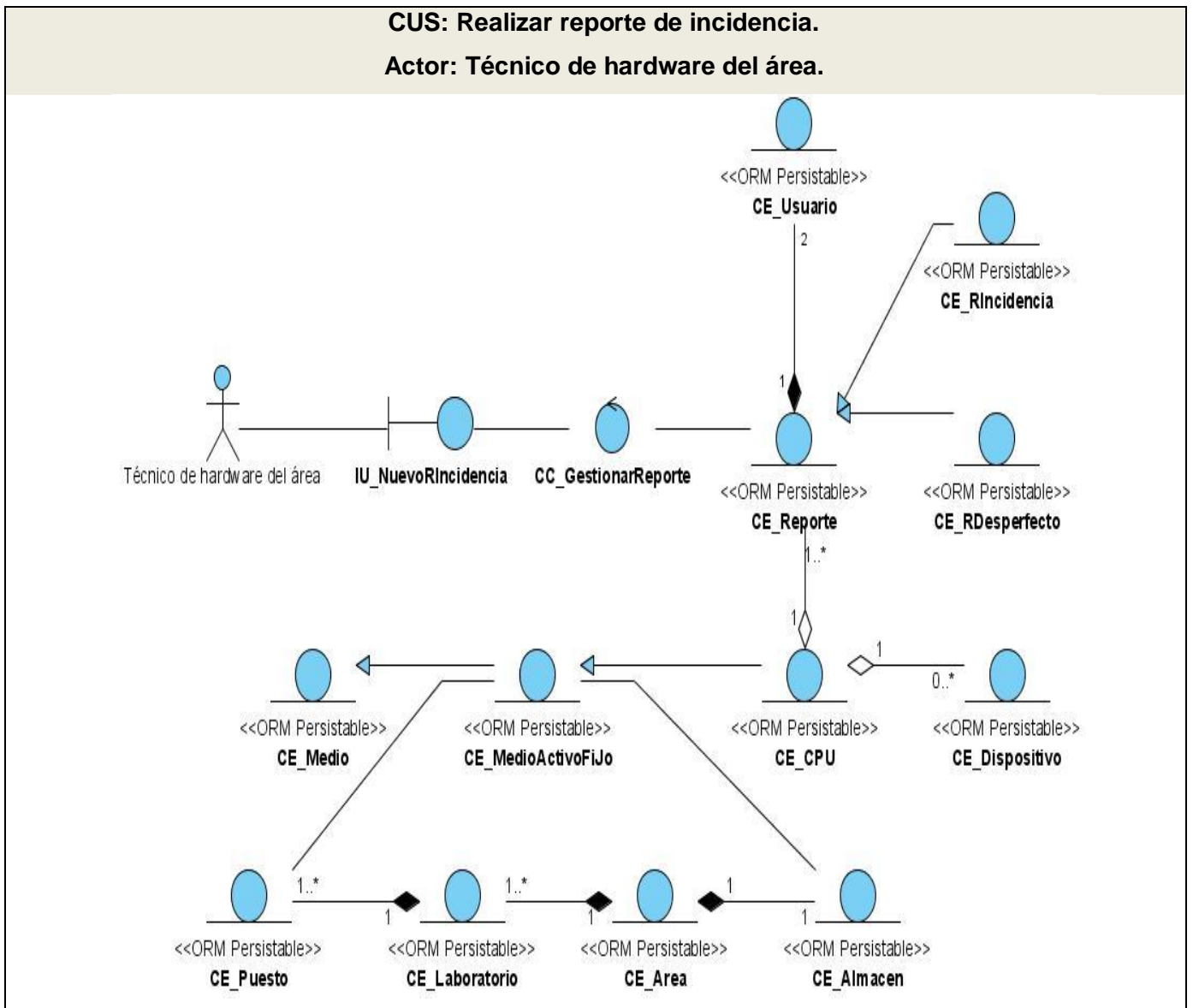


Figura 8: Diagrama de clases del CUS "Realizar reporte de incidencia".

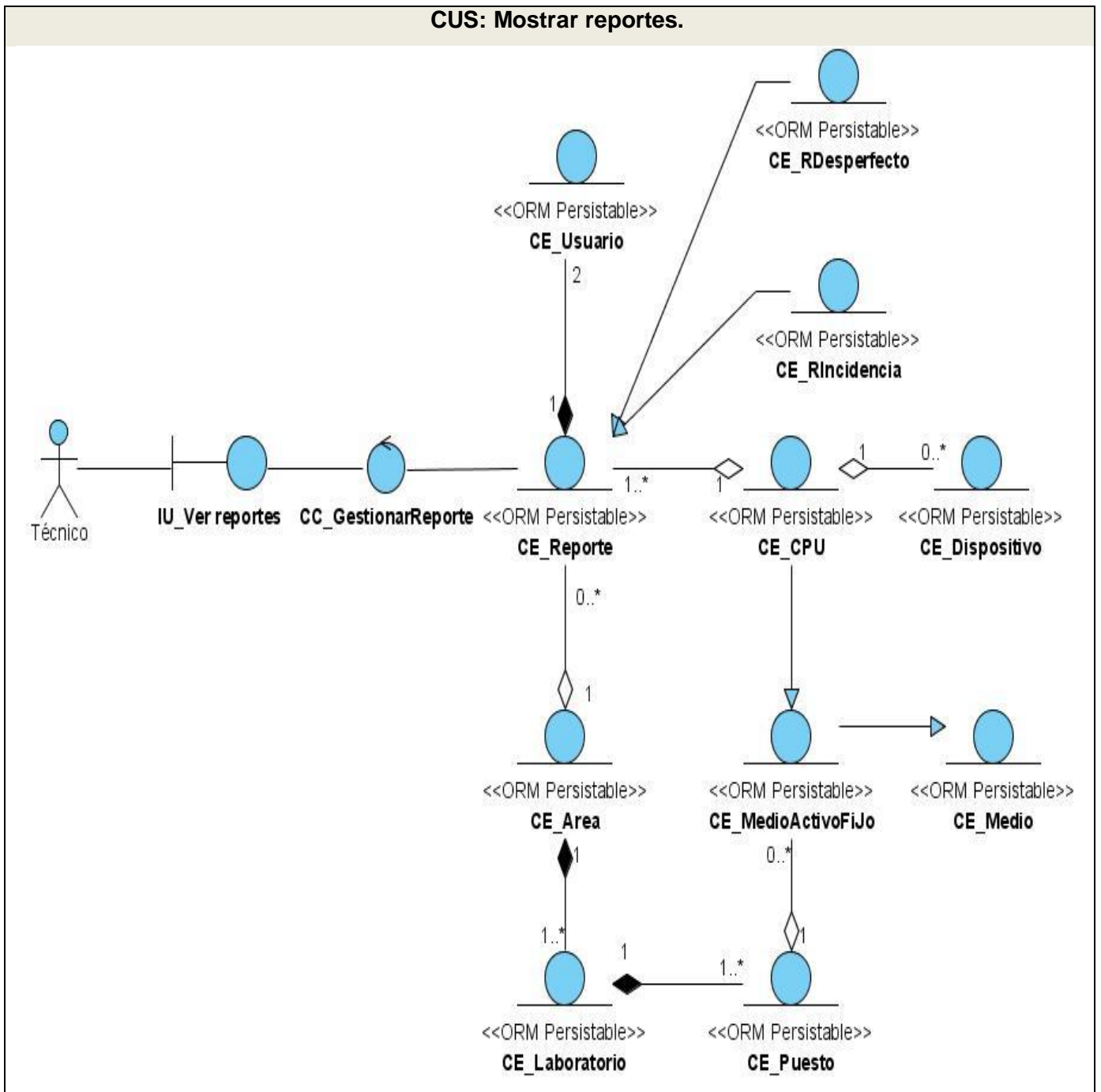
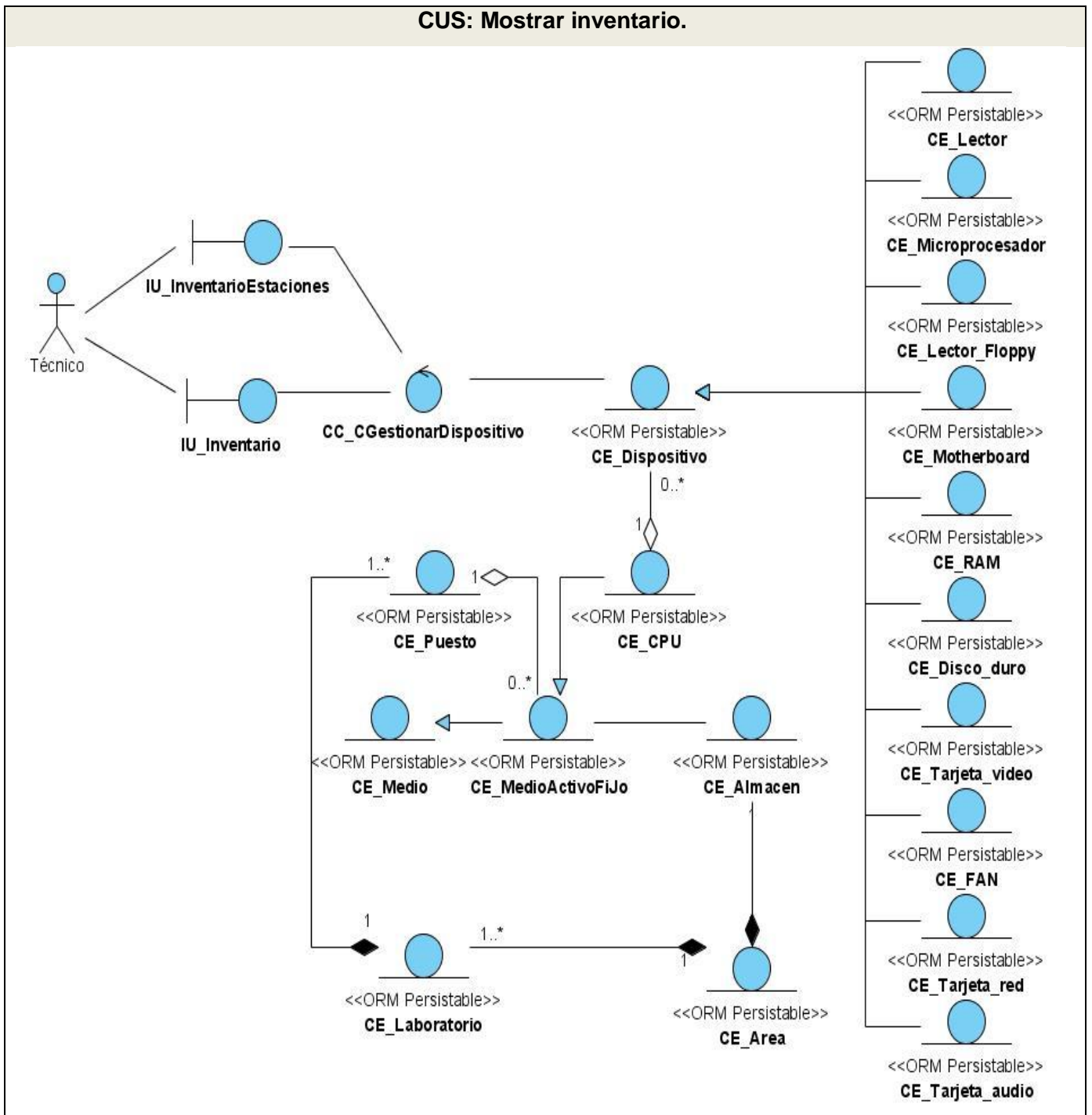


Figura 9: Diagrama de clases del CUS "Mostrar reportes".



3.1.1 Diagramas de colaboración del análisis.



Figura 12: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección dispositivo.



Figura 13: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección Estaciones.

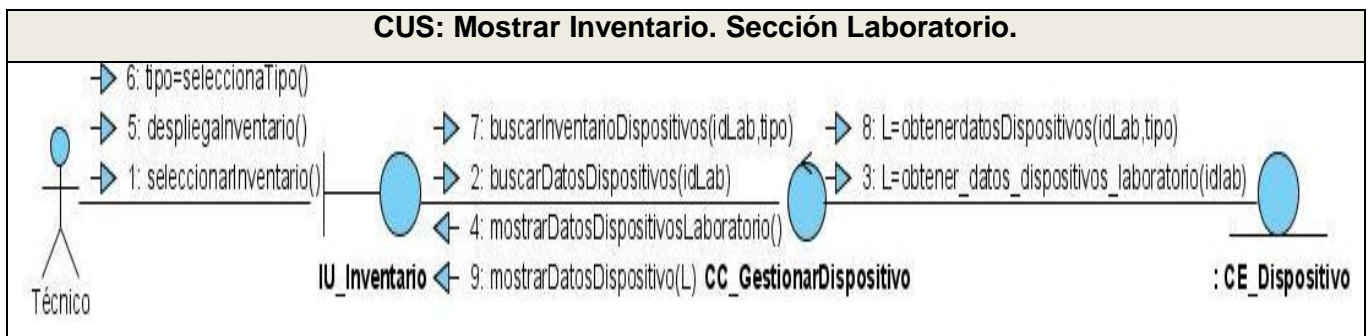


Figura 14: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección Laboratorio.

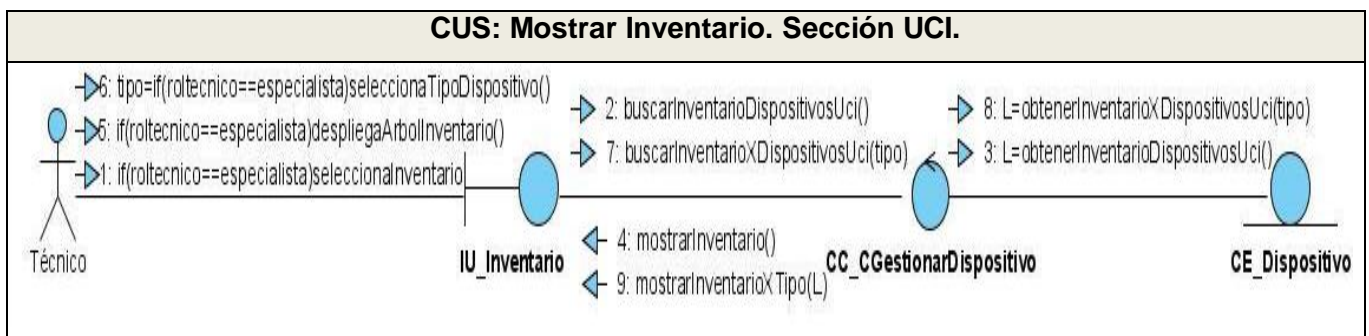


Figura 15: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección UCI.

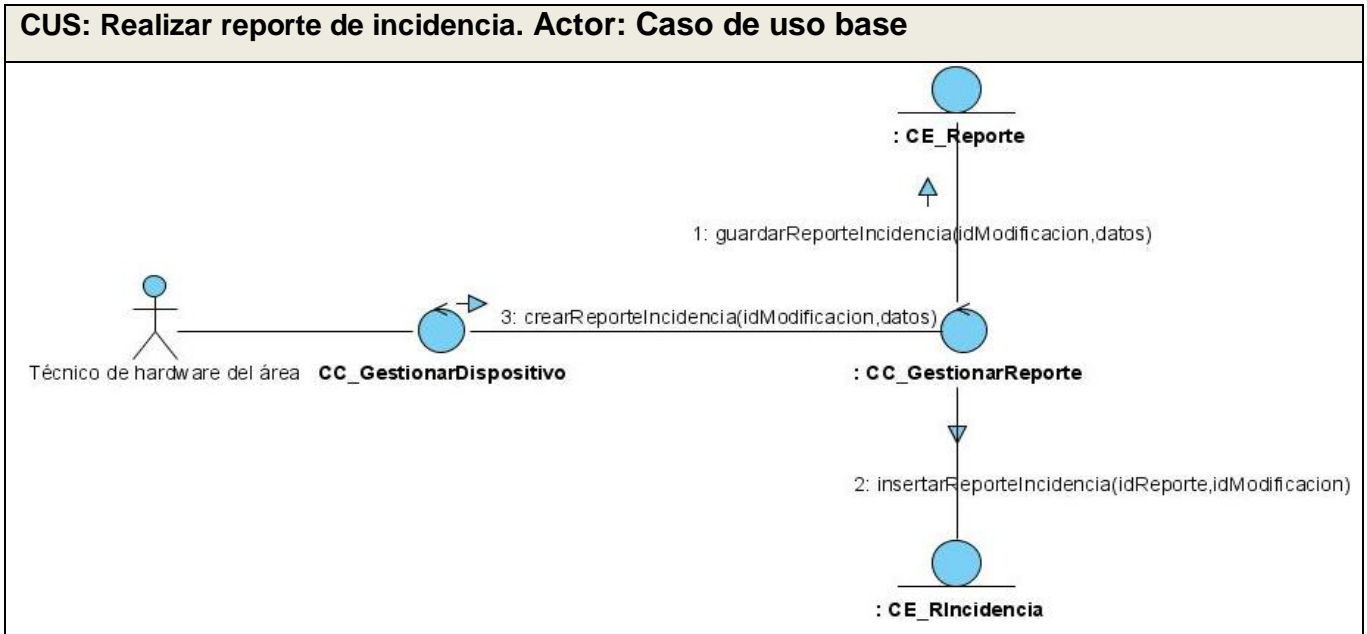


Figura 16: Diagrama de colaboración del análisis CUS “Realizar reporte de incidencia” (Interno).

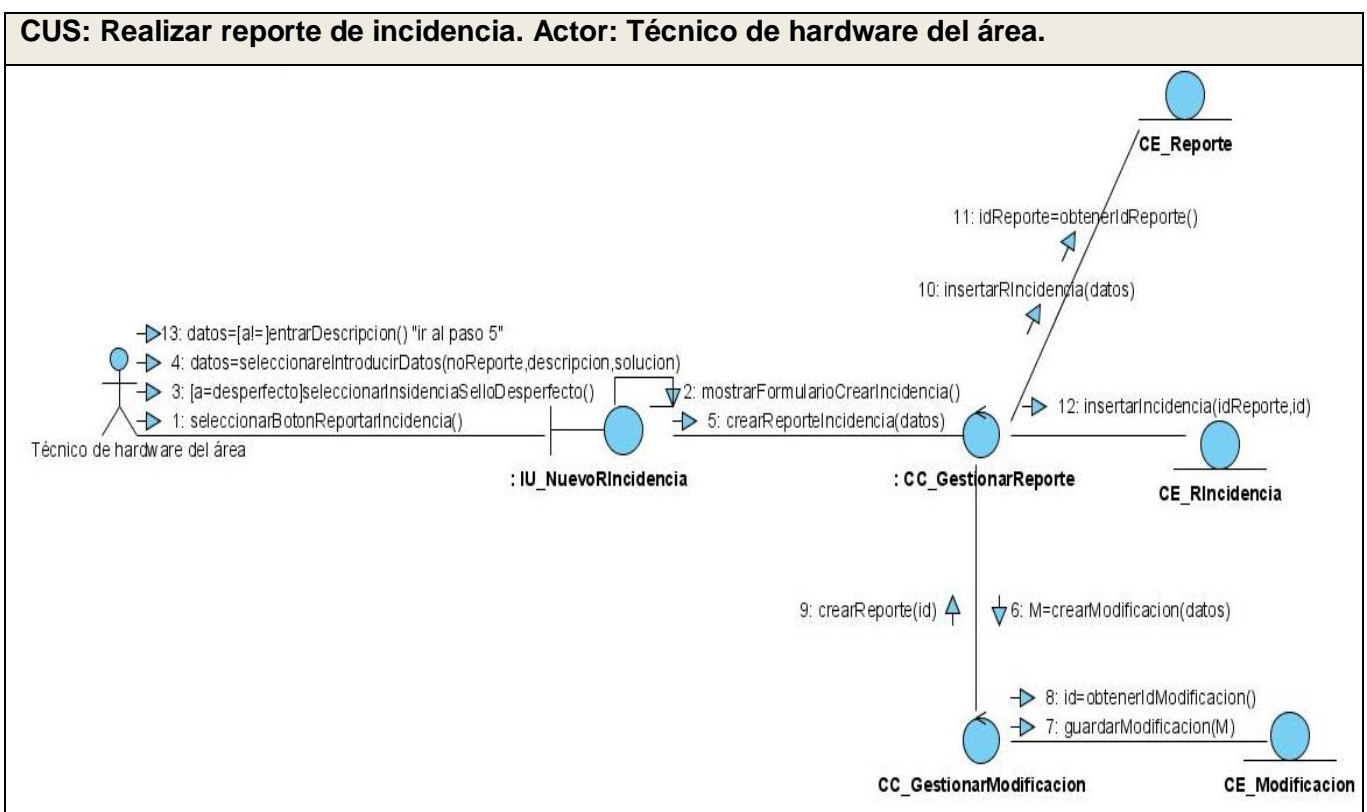


Figura 17: Diagrama de colaboración del análisis CUS “Realizar reporte de incidencia” (Externo).

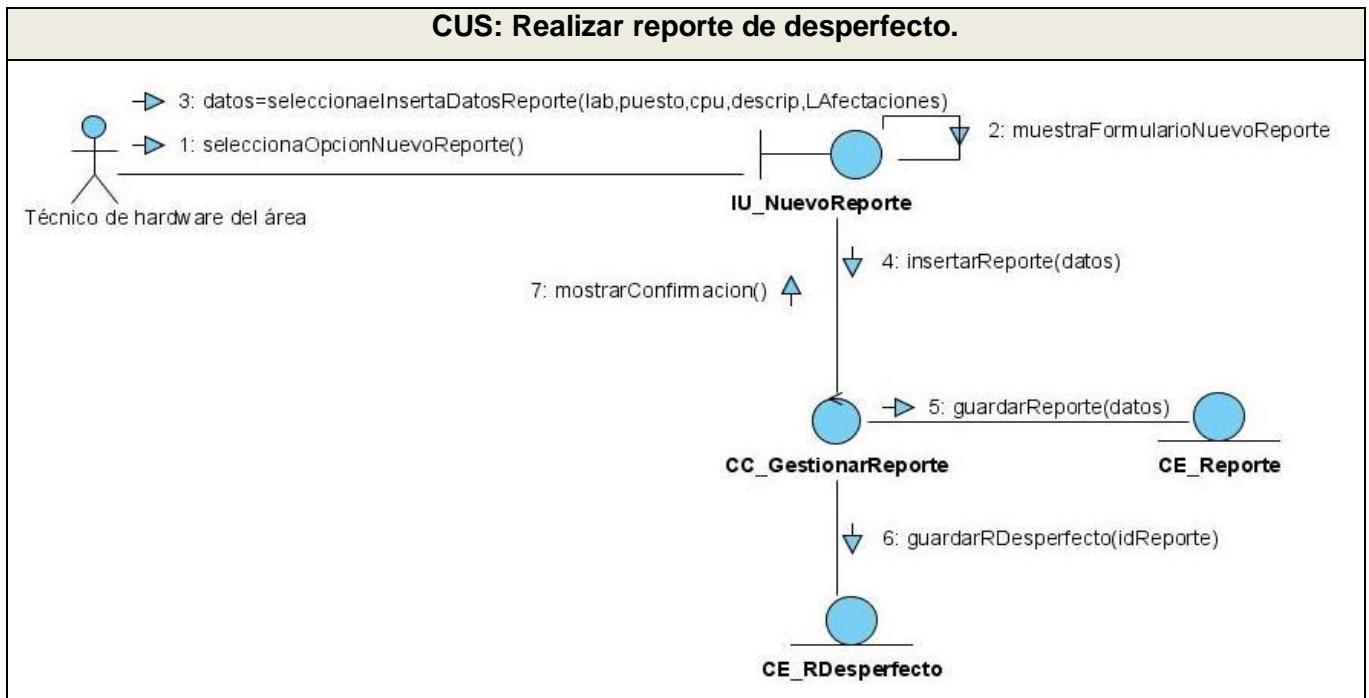


Figura 18: Diagrama de colaboración del análisis CUS “Realizar reporte de desperfecto”.

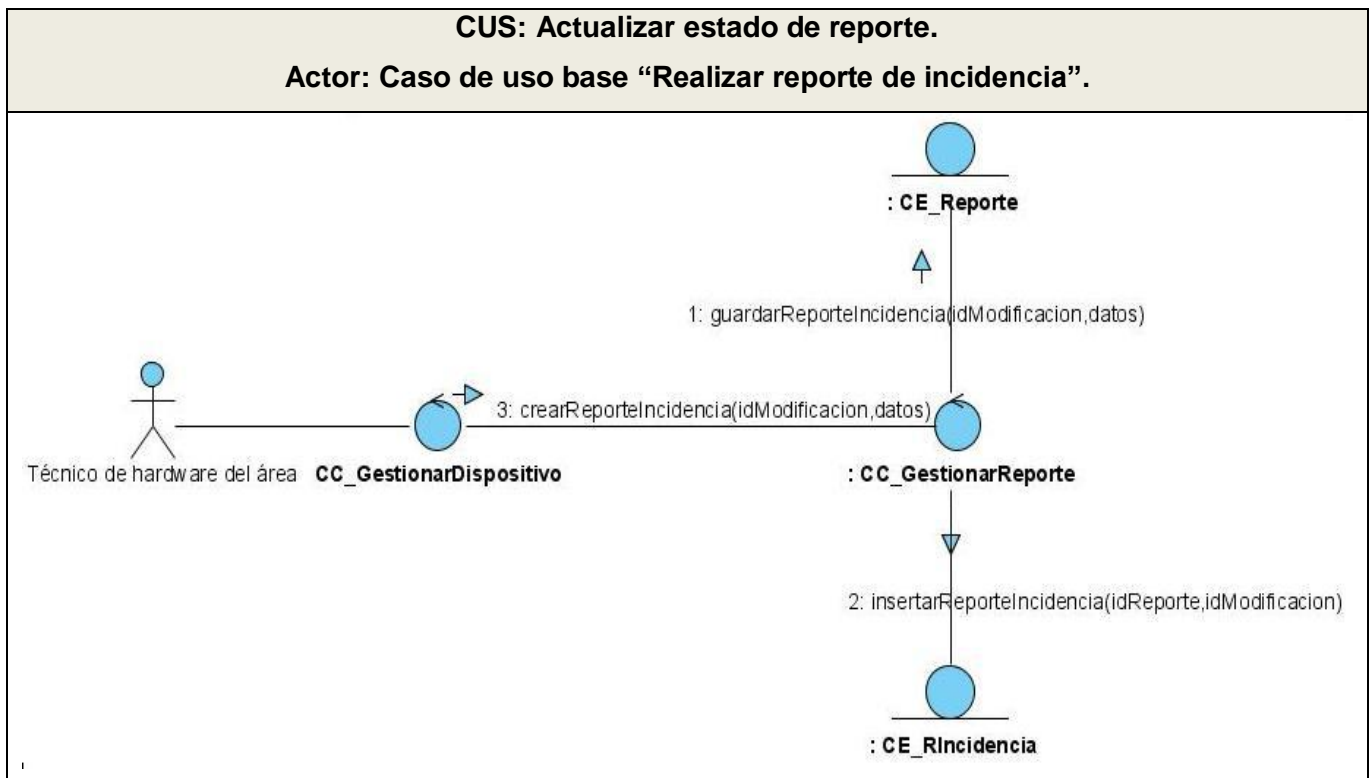


Figura 19: Diagrama de colaboración del análisis CUS “Actualizar estado de reporte.”

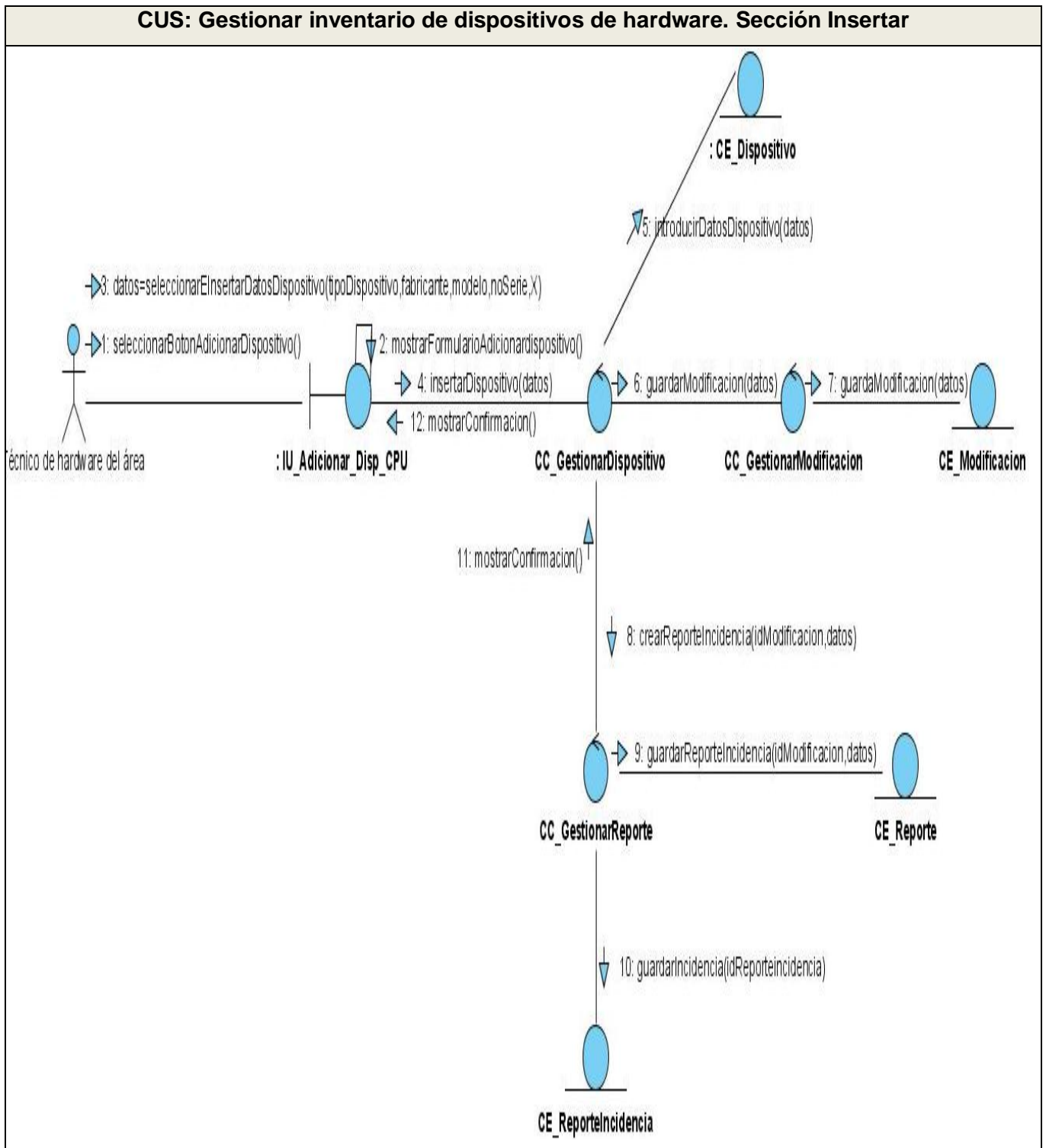


Figura 20: Diagrama de colaboración del análisis CUS “Gestionar inventario de dispositivos de hardware” Sección Insertar.

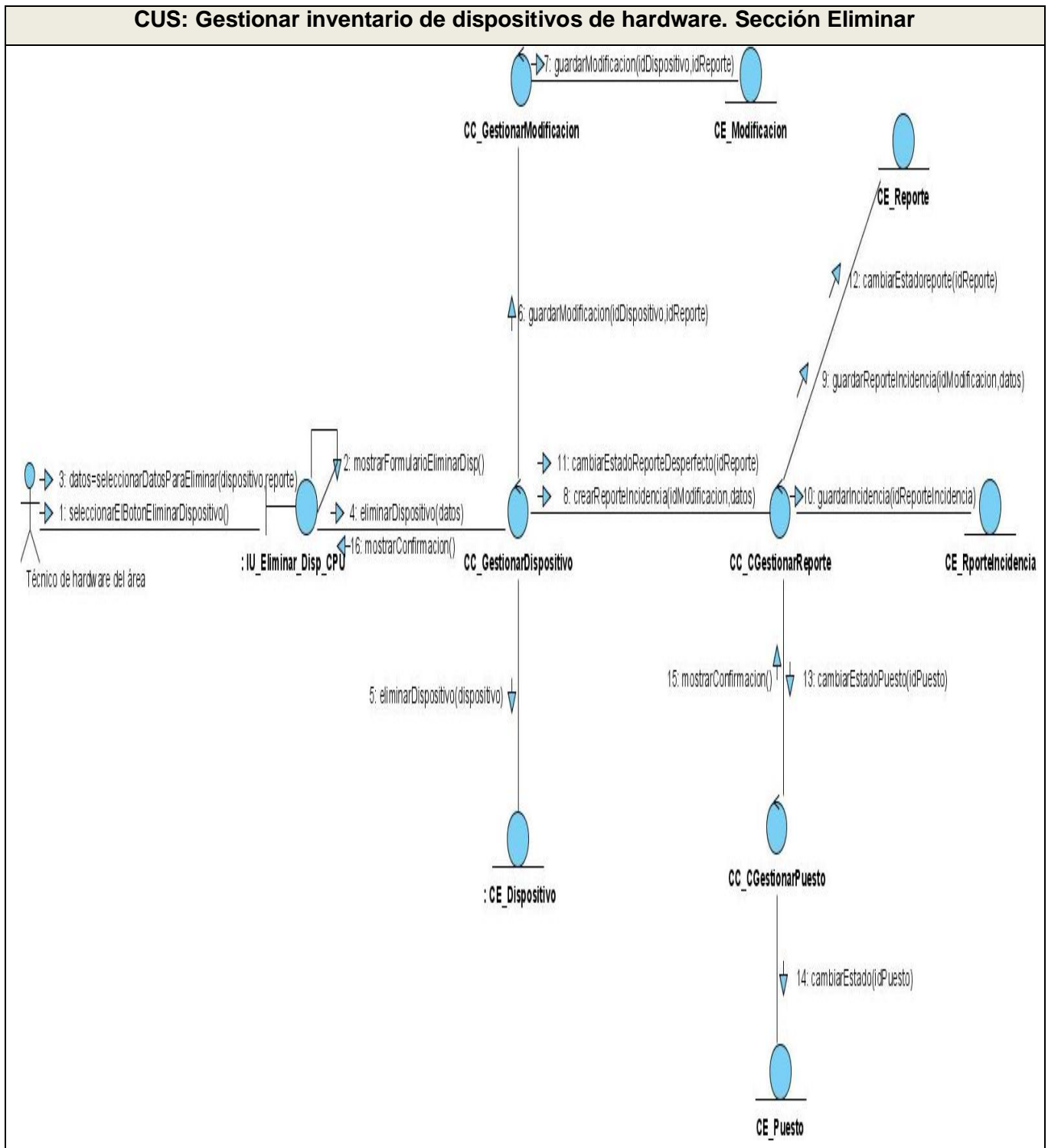


Figura 21 : Diagrama de colaboración del análisis CUS “Gestionar inventario de dispositivos de hardware” Sección Eliminar.

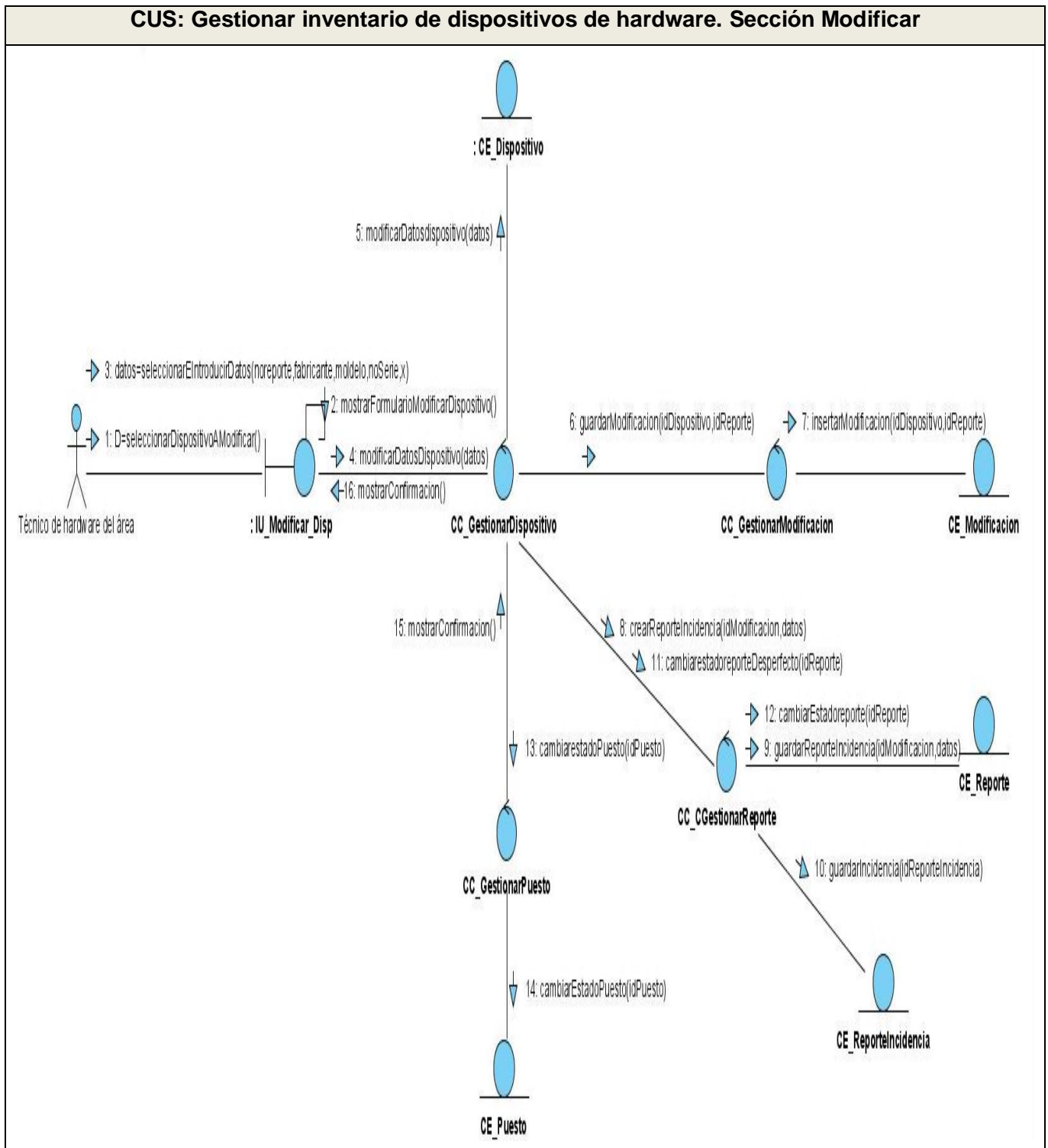


Figura 22: Diagrama de colaboración del análisis CUS “Gestionar inventario de dispositivos de hardware” Sección Modificar.

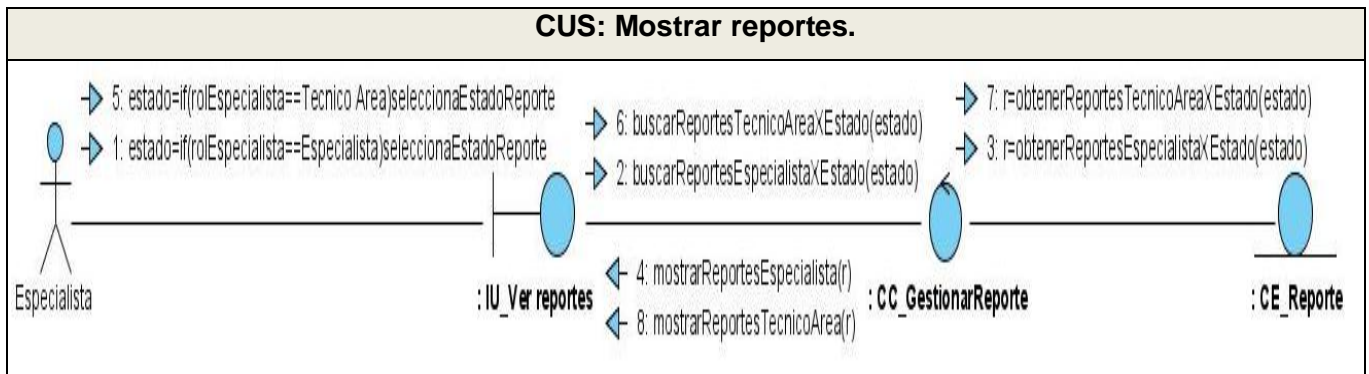


Figura 23: Diagrama de colaboración del análisis CUS “Mostrar reportes”.

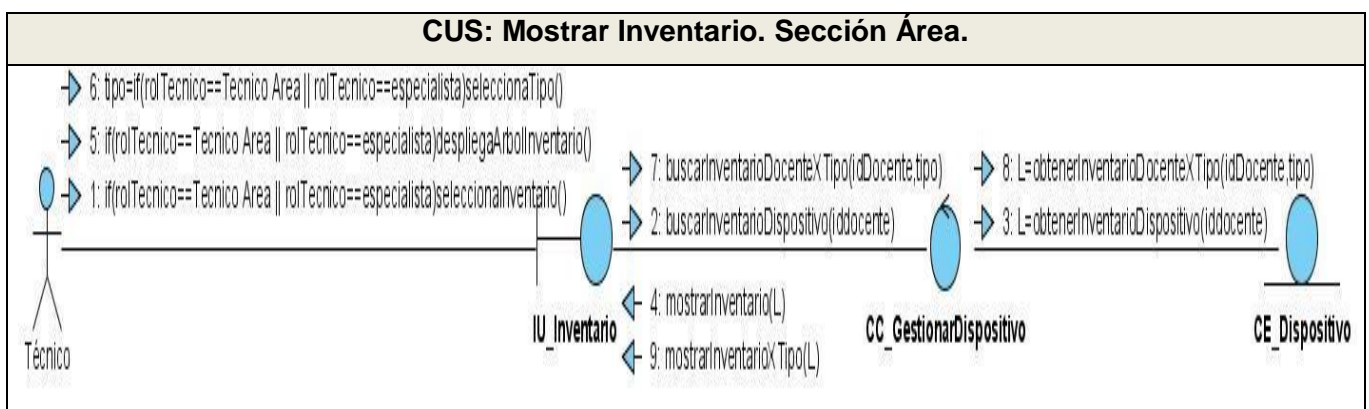


Figura 24: Diagrama de colaboración del análisis CUS “Mostrar Inventario”. Sección Área.

3.2 Diseño.

El diseño permite modelar el sistema en aras de encontrar su forma, y crea una abstracción de la implementación del sistema.

3.2.1 Diagramas de interacción.

Los diagramas de secuencia del diseño se encuentran en el **Anexo_1**.

3.2.2 Diagramas de clases del Diseño.

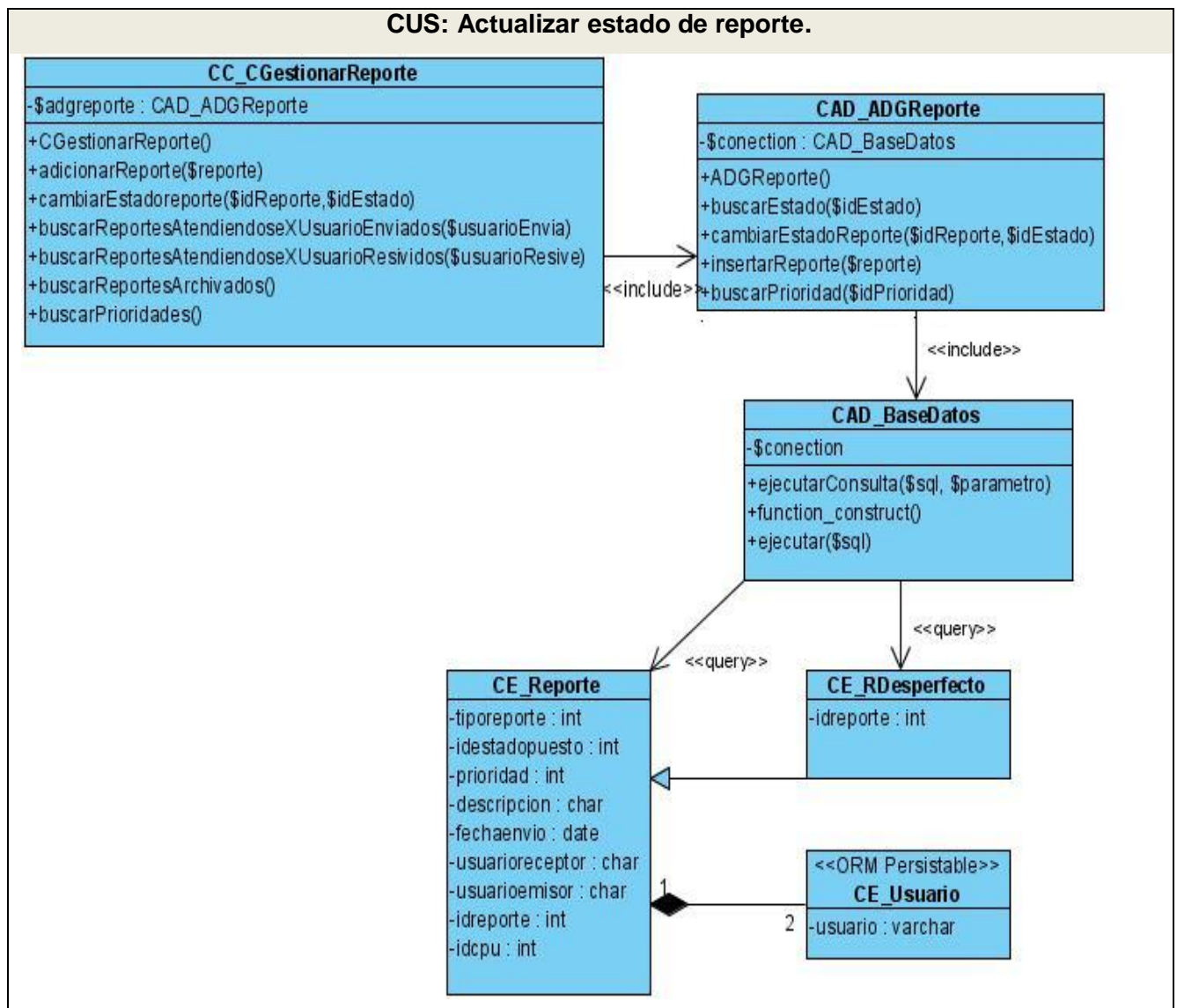


Figura 25: Diagrama de clases del diseño CUS "Actualizar estado de reporte".

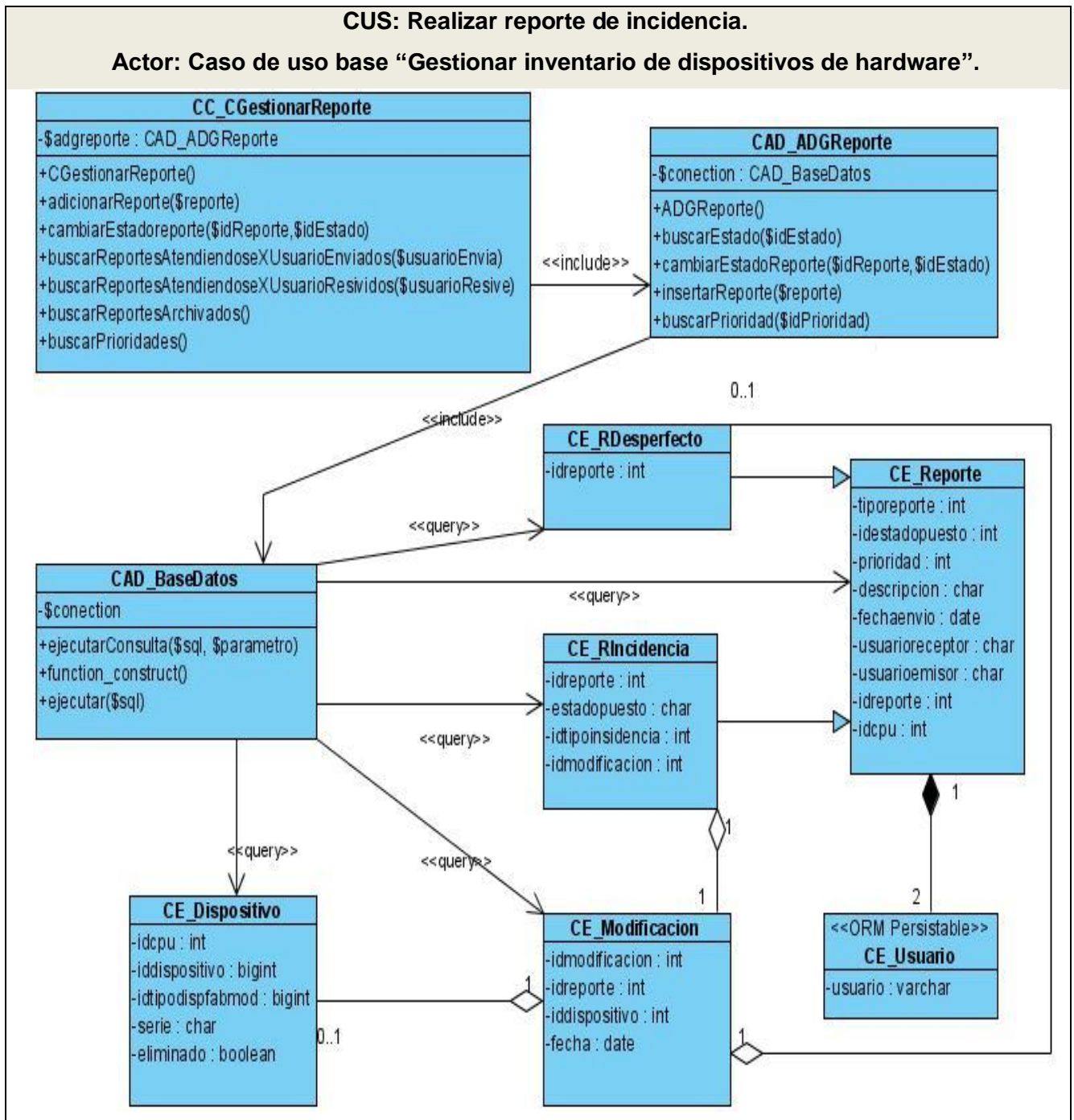


Figura 26: Diagrama de clases del diseño CUS “Realizar reporte de incidencia” (interno).

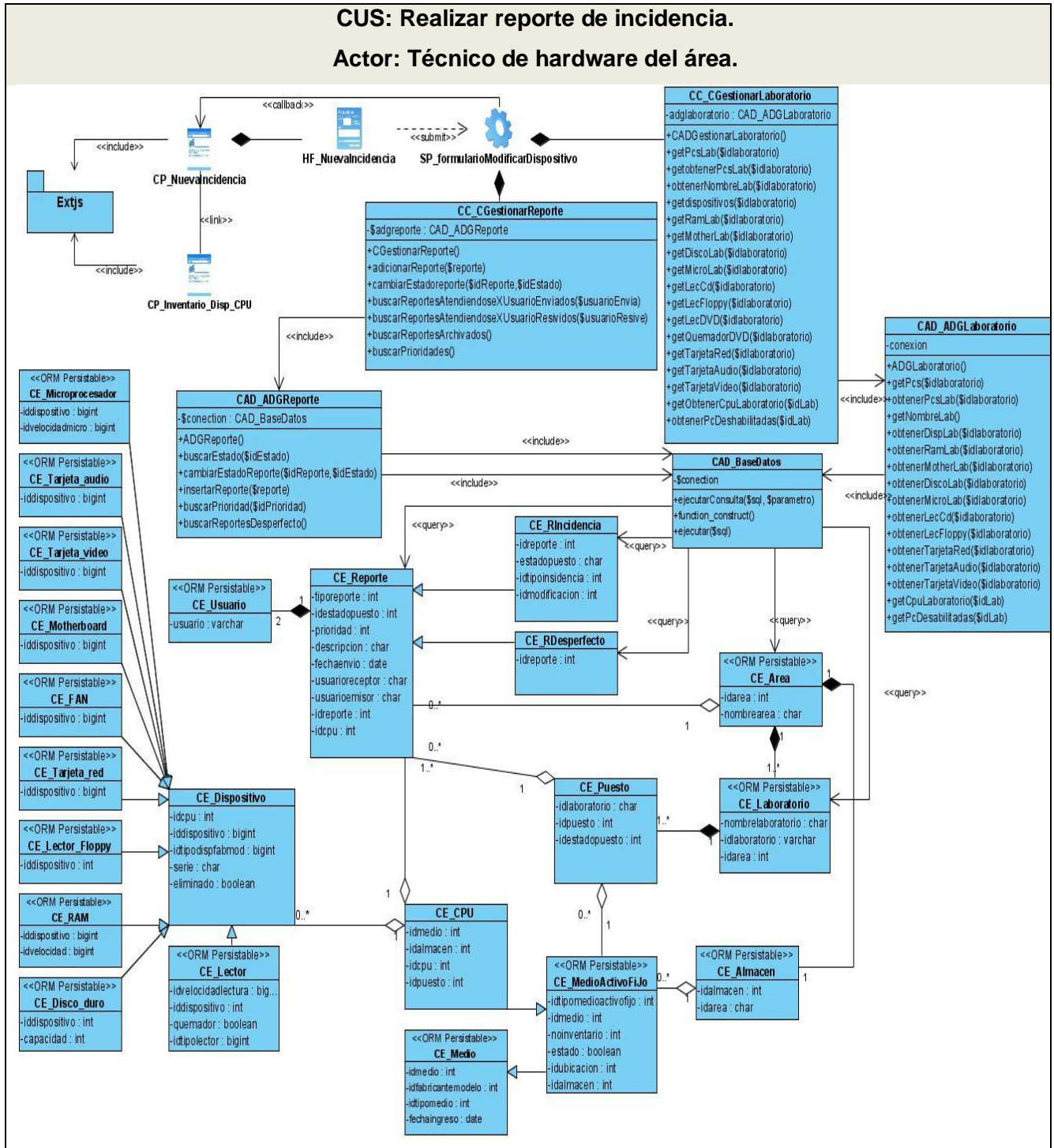


Figura 27: Diagrama de clases del diseño CUS “Realizar reporte de incidencia”.

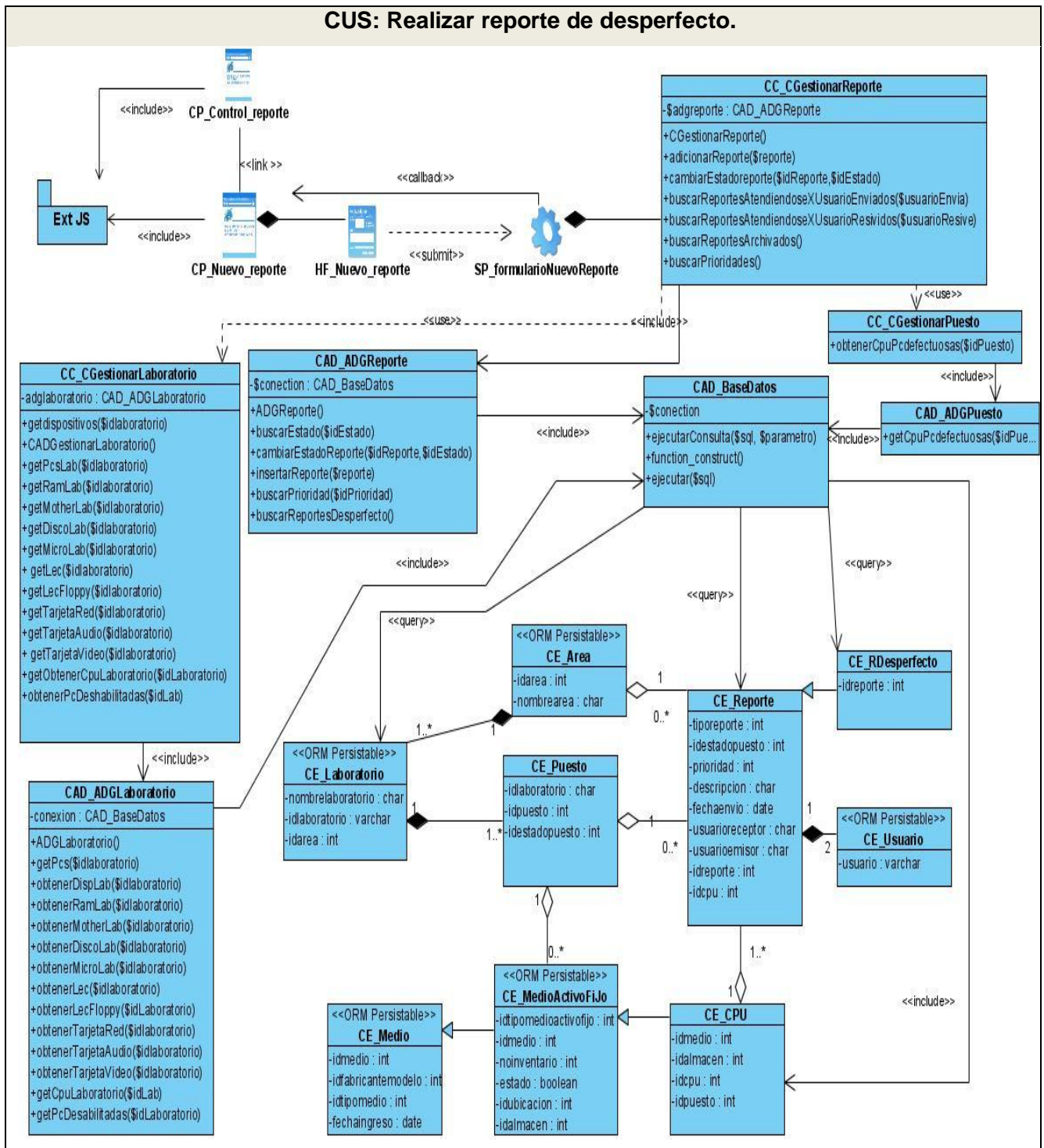


Figura 28: Diagrama de clases del diseño CUS “Realizar reporte de desperfecto”.

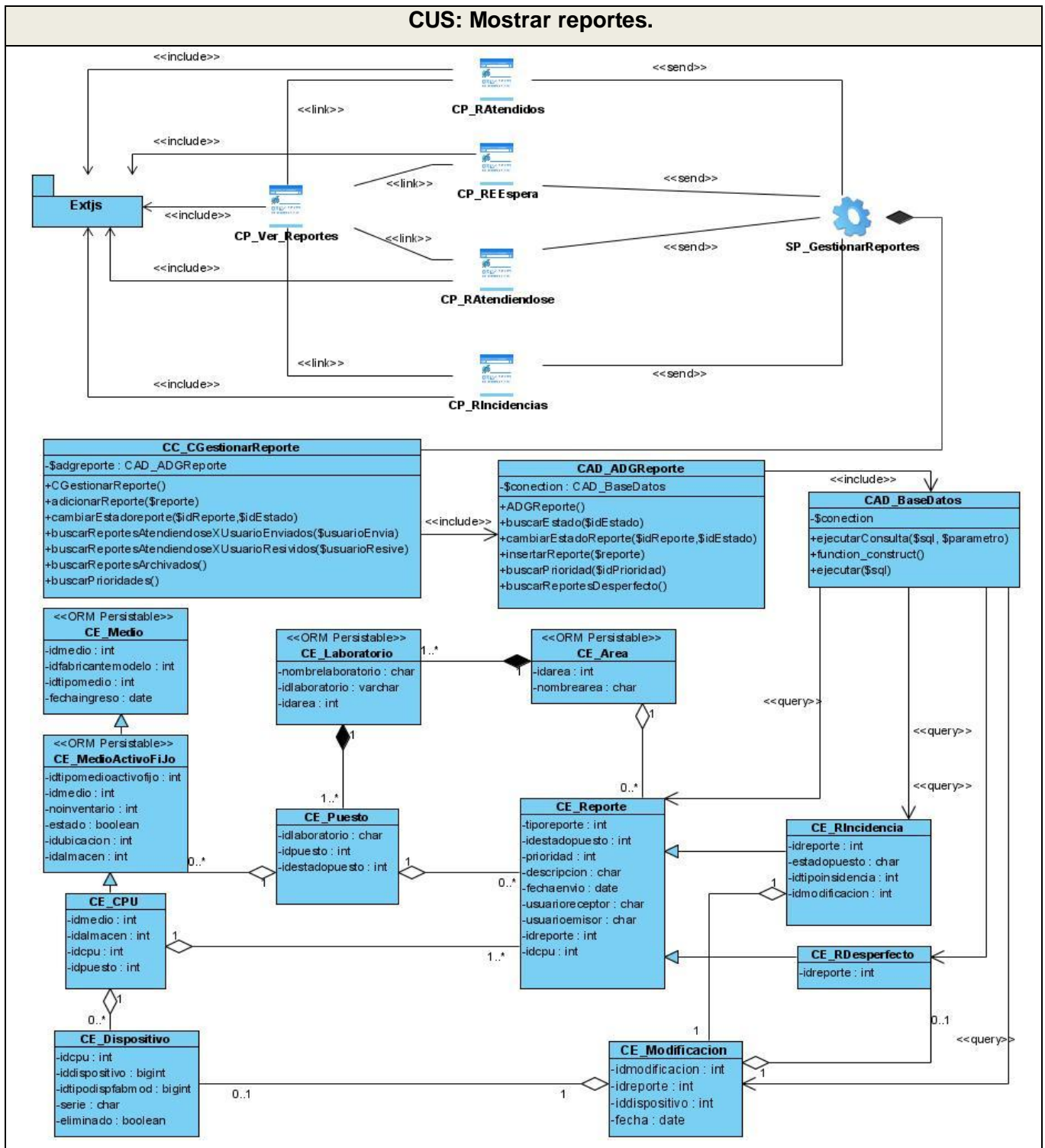


Figura 29: Diagrama de clases del diseño CUS "Mostrar reportes".

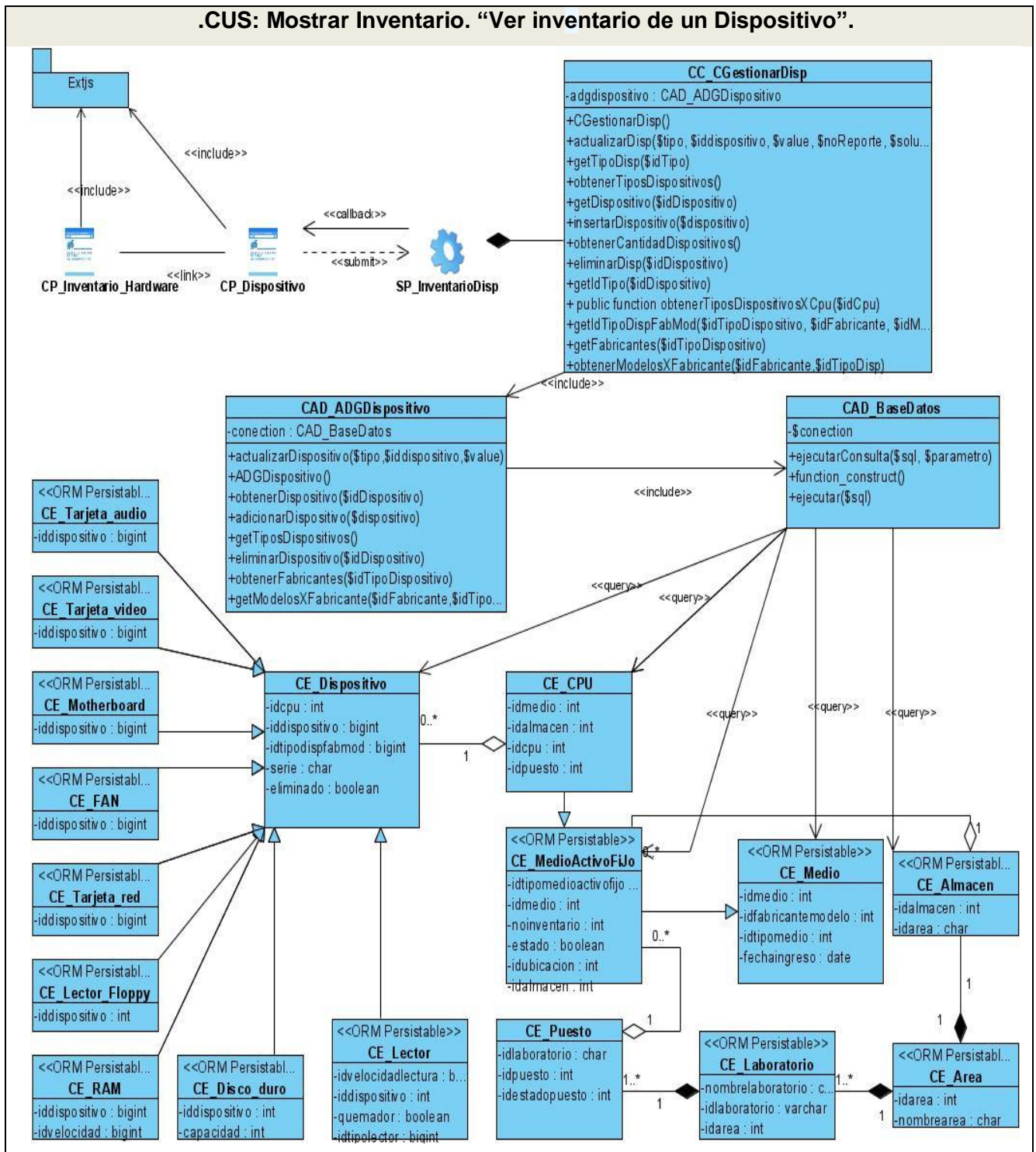


Figura 30: Diagrama de clases del diseño CUS "Mostrar Inventario" (Dispositivo).

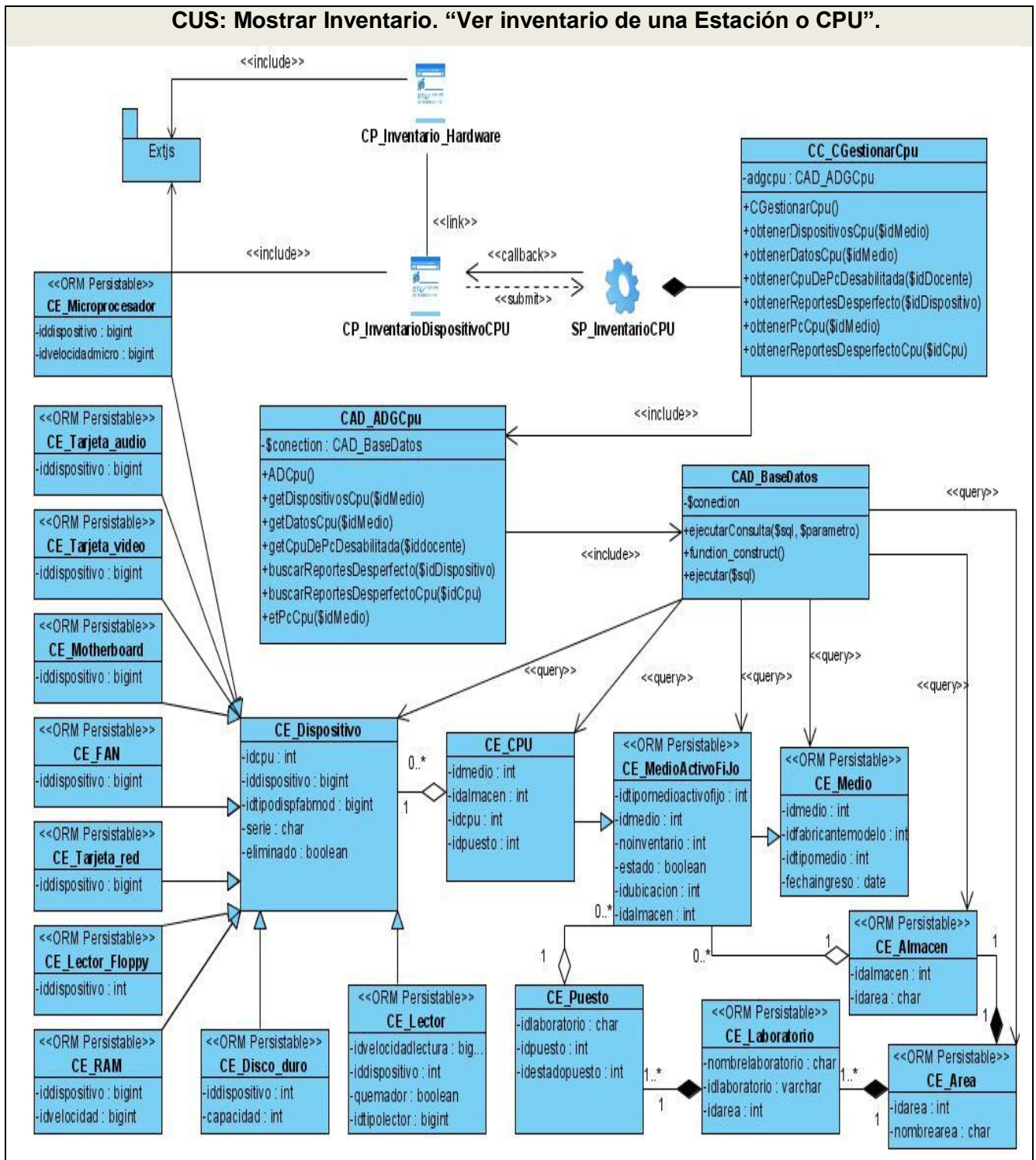


Figura 31: Diagrama de clases del diseño CUS “Mostrar Inventario” (CPU).

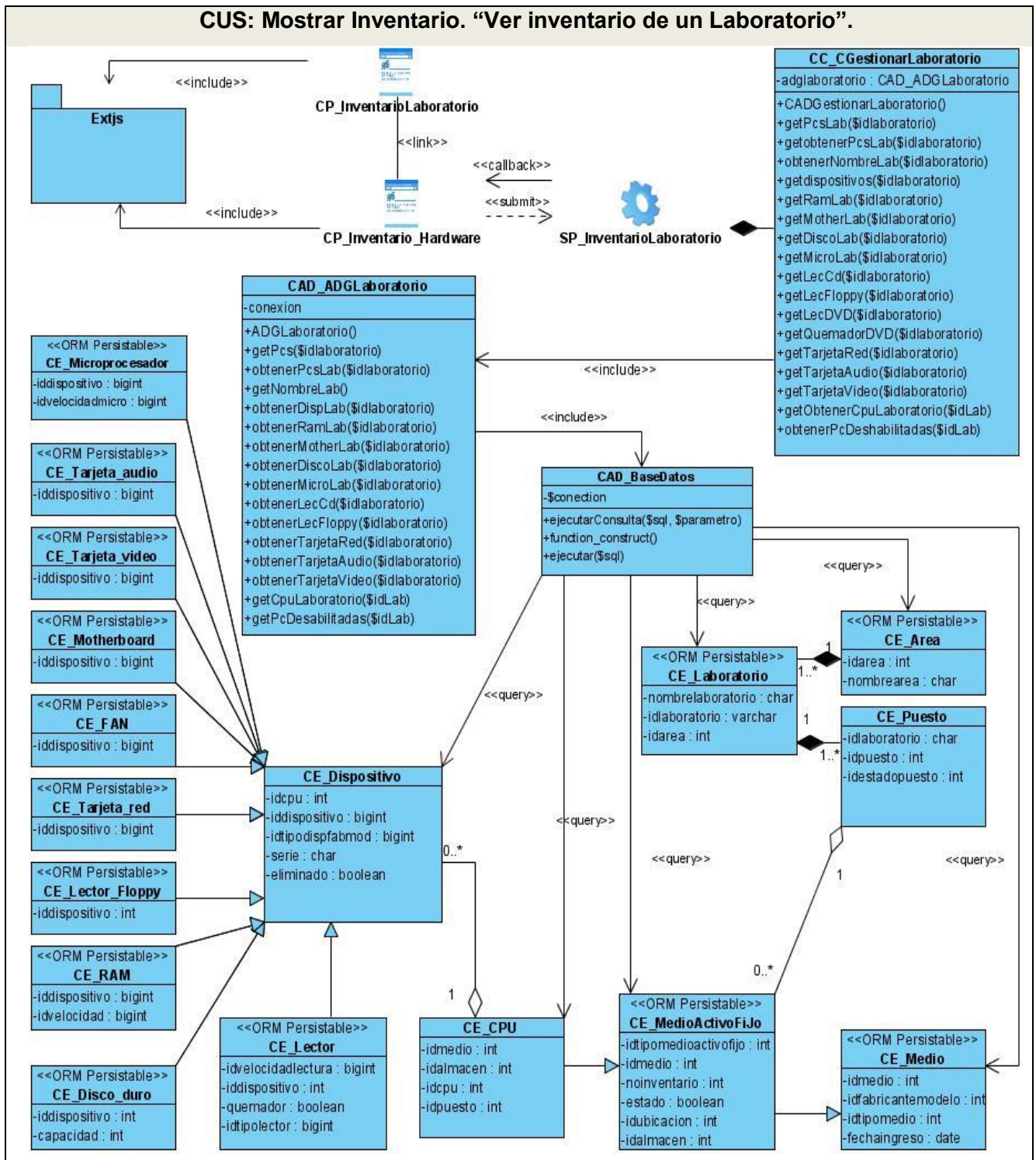


Figura 32: Diagrama de clases del diseño CUS "Mostrar Inventario" (Laboratorio).

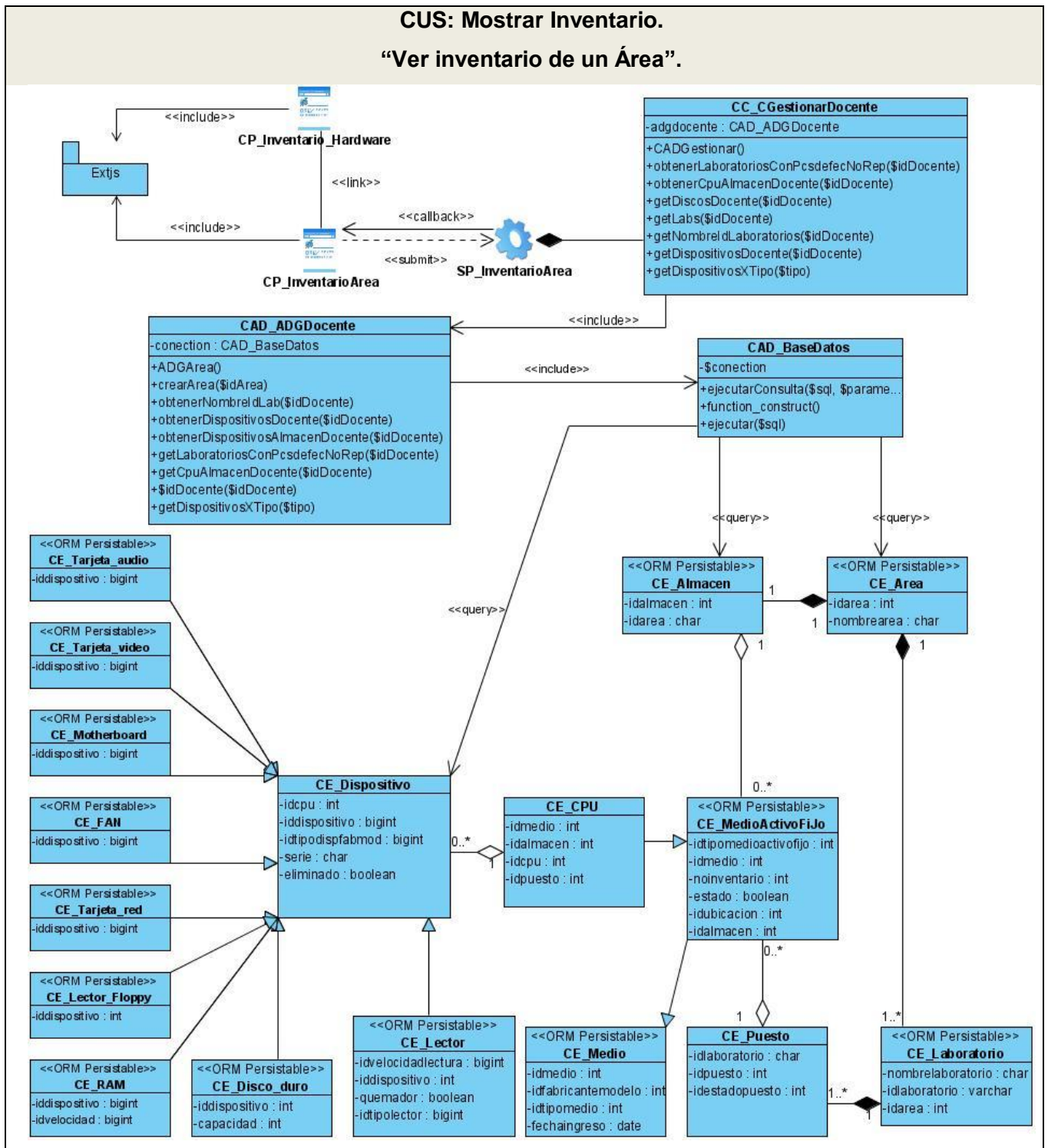


Figura 33: Diagrama de clases del diseño CUS “Mostrar Inventario” (Área).

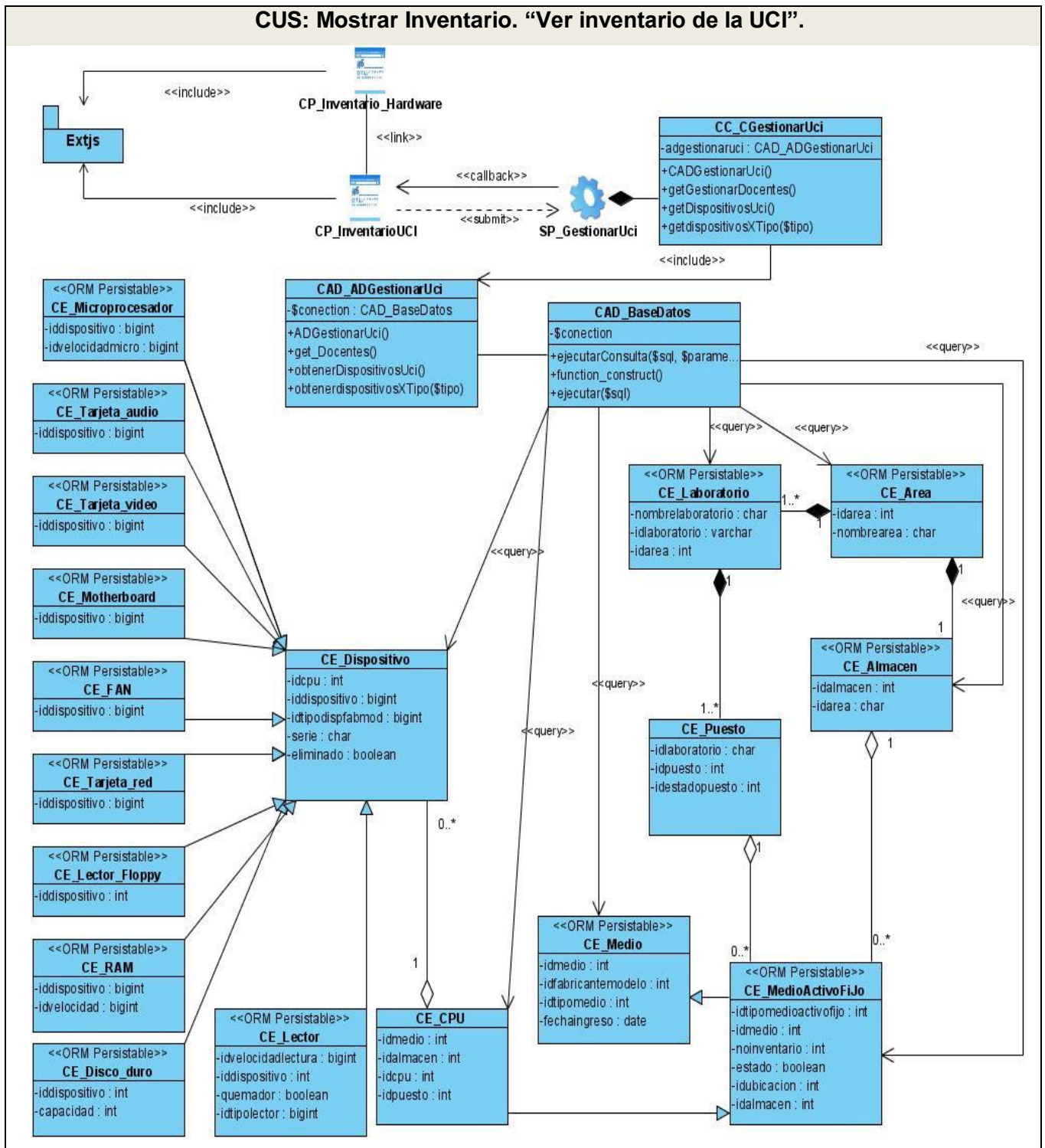


Figura 34: Diagrama de clases del diseño CUS "Mostrar Inventario" (UCI).

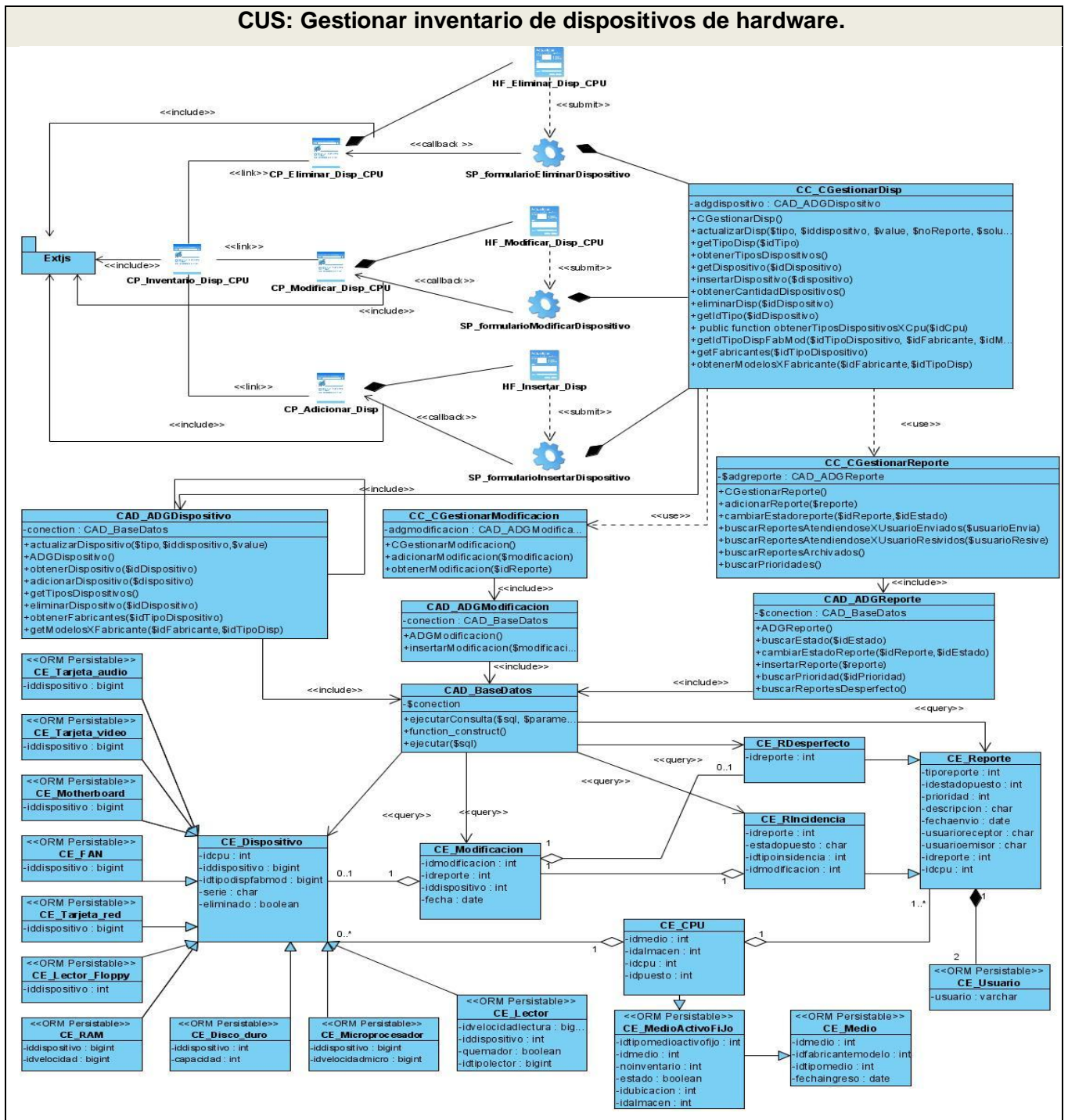


Figura 35: Diagrama de clases del diseño CUS "Gestionar inventario de dispositivos de hardware".

Conclusiones

Con el análisis de los requisitos y el modelado del sistema, se obtuvieron los datos y reglas que en él serán manejados para su correcto funcionamiento, se estableció una estructura interna del sistema y se alcanzó una primera aproximación hacia la implementación.

Capítulo #4: Implementación.

Introducción

En este capítulo se describirá cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue, con el uso de las clases activas y subsistemas encontrados durante el diseño.

4.1 Modelo de implementación.

4.1.1 Diagrama de despliegue.

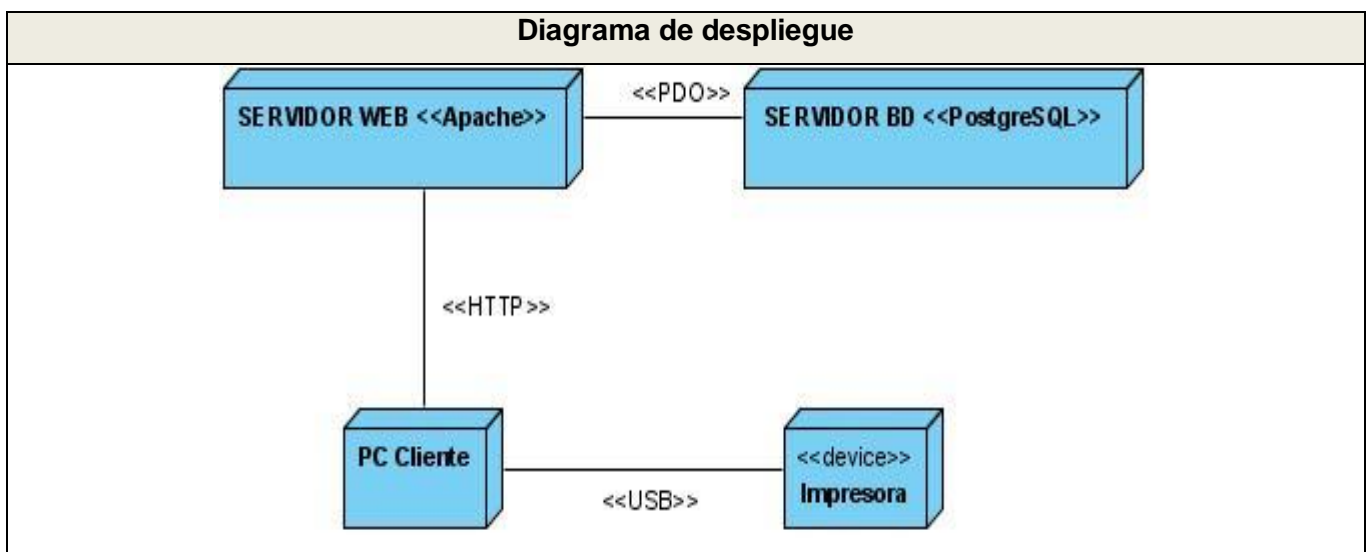


Figura 37: Diagrama de despliegue.

4.1.2 Diagrama de componentes.

En el diagrama de componentes que se ilustrará a continuación, es válido aclarar para una mejor comprensión que los paquetes "Servidoras", "Controladoras" y "Accesos a datos", no establecen relaciones entre los componentes que cada una incluye. En el caso del paquete "Controladoras", puede verse el caso que una controladora invoque a otra.

Entre el paquete "Servidoras" y los componentes de extensión ".js", en dependencia de los requerimientos de la ".js" es que se selecciona la(s) servidora(s) con la(s) que se relaciona. Lo mismo sucede, en la relación entre el paquete "Servidoras" y "Controladoras", se establece relación con la(s) controladora(s) que requiere.

Conclusiones

En este capítulo quedaron definidos los subsistemas de implementación, interfaces y contenidos. Se describieron como se organizan los componentes en el lenguaje de programación utilizado, la arquitectura utilizada en el desarrollo del diagrama de componentes, las configuraciones de red sobre las cuales deberá implementarse el sistema y como los componentes se relacionan unos con otros.

Capítulo #5: Estudio de Factibilidad.

Introducción.

En este capítulo, se realiza el Cálculo de la estimación del Módulo de control de Hardware mediante el Método de Estimación por Puntos de Casos de Uso. Para así poder determinar si es factible o no su implementación.

5.1 Método de Estimación por Puntos de Casos de Uso.

5.1.1 Cálculo de Puntos de Casos de Uso sin ajustar.

Se calcula a partir de la siguiente ecuación: $UUCP = UAW + UUCW$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Hallando el Factor de Peso de los Actores sin ajustar (UAW).

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema. Los criterios se muestran en la siguiente tabla:

Tipo	Descripción	Peso	Cantidad*peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	3*3
UAW			9

Tabla 12: Cálculo del Factor de Peso de los Actores sin ajustar.

Hallando Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la siguiente tabla:

Tipo	Descripción	Peso	Cantidad*peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	1*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	2*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	3*15
UUCW			70

Tabla 13: Cálculo del Factor de Peso de los Casos de Uso sin ajustar.

Los Puntos de Casos de Uso sin ajustar resultan:

$$UUCP = UAW + UUCW \quad UUCP = 9 + 70$$

$$UUCP = 79.$$

5.1.2 Cálculo de Puntos de Casos de Uso ajustados.

Se calcula a partir de la siguiente ecuación: $UCP = UUCP \times TCF \times EF$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Hallando Factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Peso	Valor	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido	2	2	4
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Procesamiento interno complejo	1	1	1
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	2	2
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	1	1
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	1	1
Total				41.5

Tabla 14: Peso de los factores determinantes de la complejidad técnica del sistema.

El Factor de complejidad técnica resulta:

$$\text{TCF} = 0.6 + 0.01 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

$$\text{TCF} = 0.6 + (0.01 * 41.5)$$

TCF = 1.015.

Hallando el Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

En la siguiente tabla se muestra el peso y el valor asignado de cada uno de éstos factores.

Factor	Descripción	Peso	Valor asignado	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	0	0
E2	Experiencia en la aplicación	0.5	0	0
E3	Experiencia en orientación a objetos	1	3	3
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	3	6
E7	Personal part-time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	4	-4
Total				9

Tabla 15: Peso y valor asignado de cada factor que intervienen en el cálculo del Factor de ambiente.

El Factor de ambiente resulta:

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

$$EF = 1.4 - 0.03 * 9.$$

EF =1.13.

Los Puntos de Casos de Uso ajustados resultan:

$$UCP = UUCP * TCF * EF$$

$$UCP = 79 * 1.015 * 1.13$$

UCP = 90.60905.

5.1.3 De los Puntos de Casos de Uso a la estimación del esfuerzo

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad (programación) especificada en los casos de uso.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores de E1 a E6.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

CF = 20 horas-hombre (si $Total_{EF} \leq 2$)

CF = 28 horas-hombre (si $Total_{EF} = 3$ ó $Total_{EF} = 4$)

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$)

$Total_{EF} = \text{Cant EF} < 3 \text{ (entre E1 –E6)} + \text{Cant EF} > 3 \text{ (entre E7, E8)}$

$Total_{EF} = 2+1 = 3$

CF = 28 horas-hombre (porque $Total_{EF} = 3$)

El esfuerzo estimado en horas-hombre sería:

$E = UCP \times CF$

$E = 90.60905 \times 28 \text{ horas-hombre.}$

$E = 2537.0534 \sim 2537.$

E = 2537.

5.1.4 Esfuerzo del módulo.

Actividad	% esfuerzo	(E*% esfuerzo)/ % esfuerzo programación
Análisis	10%	634.25 horas-hombre
Diseño	20%	1268.5 horas-hombre
Programación	40%	2537 horas-hombre
Prueba	15%	951.375 horas-hombre
Sobrecarga	15%	951.375 horas-hombre
Total	100%	6342.5 horas-hombre

Tabla 16: Cálculo del esfuerzo del módulo.

Teniendo en cuenta que se trabajan 10 horas durante 6 días de la semana y 5 horas los domingos.

$E_T = 6342.5$ horas-hombre y se estima que cada mes tiene como promedio 260 horas laborales eso daría un $E_T = 24,4$ mes-hombre.

Se estima que una persona podrá realizar el problema analizado en 24 meses aproximadamente. Lo que equivale a un año entre dos personas.

Costo del Módulo:

Se asume como salario promedio mensual \$100.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres.

CHM = 2 * Salario Promedio.

CHM = 200.00 \$/mes.

5.2 Beneficios tangibles e intangibles.

El beneficio fundamental que brinda el módulo de control de hardware, es auxiliar una herramienta Web flexible, dinámica y de interfaz agradable, que permita la gestión de los procesos de control de hardware en las áreas de los laboratorios de la UCI, tales como la gestión de inventarios de dispositivos de hardware y realización de los reportes desperfecto.

Por tanto, los beneficios inmediatos son generalmente intangibles:

- Obtención de una aplicación Web dinámica y flexible, que agrupe los procesos de gestión de dispositivos de hardware en el área de los laboratorios.
- Mayor control en los procesos de gestión de los dispositivos de hardware.
- Procesamiento de reportes internos y externos.
- Obtención de información actualizada, sobre reportes e inventarios de dispositivos de hardware.
- Control de reportes de desperfecto e incidencias.
- Acceso visual al histórico de reportes que ha sido almacenado.

5.3 Análisis de costo.

Analizando el costo del proyecto y los numerosos beneficios que reportará, se puede concluir, que su implementación es realmente factible.

La tecnología utilizada para el desarrollo de la aplicación es totalmente libre, por tanto, no hay que incurrir en gastos de pago de licencias de uso.

Conclusiones

Mediante el estudio de factibilidad realizado, se concluyó que el sistema propuesto, es realmente factible y será beneficioso no solo para la dirección de los laboratorios de la UCI, sino también a la UCI al no reportarle gastos de pago de licencia.

CONCLUSIONES

En el presente trabajo de diploma, se realizó un profundo análisis del negocio correspondiente al control de hardware en la dirección de los laboratorios de la UCI. Este análisis permitió determinar deficiencias en el mismo, entre ellas ,la dependencia existente de aplicaciones ajenas a las áreas de los laboratorios, la falta de interacción entre los procesos de reportes de desperfecto que se realizan y las modificaciones efectuadas a algún dispositivo de hardware y en opinión de los autores de este trabajo, la más significativa de todas las deficiencias, es que los sistemas que intervienen en el negocio solo se ejecutan en la plataforma propietaria Windows, lo que incumple con el proceso de migración a software libre existente en el país.

La aplicación propuesta, fue implementada y concebida con el uso de herramientas, tecnologías, metodologías elegidas en mutuo acuerdo con los clientes; es una aplicación multiplataforma que cumple el objetivo planteado inicialmente “Desarrollar una aplicación Web que integre los procesos de gestión de dispositivos de hardware, que se realizan en la dirección de los laboratorios de la UCI” y hará del control de hardware un proceso más riguroso y exacto.

RECOMENDACIONES

Unificar el sistema implementado, al desarrollado en el trabajo de diploma titulado “Adaptación y configuración de OCS inventario NG, para el inventario centralizado de hardware y software de la UCI” perteneciente a la facultad 2 e incorporarle los procesos de solicitudes de dispositivos de hardware.

Se propone incorporarle al sistema, un framework del lado servidor como Symphony para así lograr una aplicación robusta con la fusión de este con la Ext. JS.

Referencias bibliográficas.

Eguíluz Pérez, Javier. 2007. Libros Web. *Introducción a JavaScript*. [En línea] 2007. [Citado el: 5 de Diciembre de 2007.] <http://www.librosweb.es/javascript>.

HTML.net. Tutoriales sobre HTML y CSS - Construye tu propio sitio Web. [En línea] [Citado el: 15 de Marzo de 2008.] <http://es.html.net/tutorials/css/lesson1.asp>.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2000. *El Lenguaje Unificado de Modelado*. [En línea] 2000. [Citado el: 10 de Enero de 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

BIBLIOGRAFÍA

[En línea] [Citado el: 24 de 01 de 2008.] www.netsupportsoftware.com/news/pdfs/PC_Plus_DNA_april_2004.pdf

[En línea] [Citado el: 24 de 01 de 2008.]

http://www.totemguard.com/content/products/prod_netsupport_products.php.

[En línea] [Citado el: 15 de 03 de 2008.] <http://www.netsupportsoftware.com/>.

Álvarez Romero, Eduardo y Pueyo, Daniel. 2004-2005. *Integration Definition For Funcion Modeling*. [En línea] 2004-2005. [Citado el: 12 de Diciembre de 2007.] <http://dmi.uib.es/~burguera/download/IDEF0trabajo.doc>.

Apache. 1995-2006. *Versión 2.2 de la documentación del Servidor de HTTP Apache*. [En línea] 1995-2006. [Citado el: 20 de Enero de 2008.] <http://www.apache2.org/>.

Artiles Visual, Leticia, y otros. *PERFECCIONAMIENTO DE LA GESTION UNIVERSITARIA EN EL ISCMH: Proyecto Tocororo*. [En línea] [Citado el: 8 de Enero de 2008.] http://www.ucmh.sld.cu/rhab/vol5_num1/rhcm10106.htm.

ASSETS. 2004. *Qué es ASSETS?* [En línea] 2004. [Citado el: 5 de Diciembre de 2007.] <http://www.assets.co.cu/assets.asp>.

—. **2004.** *Recursos Humanos*. [En línea] 2004. [Citado el: 5 de Diciembre de 2007.] <http://www.assets.co.cu/humanos.asp>.

Departamento de Control de Calidad y Auditoría Informática. 2001. *Sistemas en Arquitectura Cliente/Servidor*. [Subdirección de Sistemas] s.l. : Dirección General de Servicios de Cómputo Académico, 2001.

DesarrolloWeb. *Qué es PHP*. [En línea] [Citado el: 17 de Noviembre de 2007.] <http://www.desarrolloweb.com/articulos/392.php>.

Edinet. 2002. *Qué es XML?* [En línea] 2002. [Citado el: 12 de Diciembre de 2007.] <http://www.edinet.com/sabia2.asp>.

Ext JS. 2006. *Ext JS - JavaScript Library*. [En línea] 2006. [Citado el: 10 de Diciembre de 2007.] <http://extjs.com/>.

Referencias bibliográficas y bibliografía

- Gábilos Software. 2001-2007.** *Gábilos Software*. [En línea] 2001-2007. [Citado el: 2 de Diciembre de 2007.] <http://www.gabilos.com/programasdegestion/prestacionesaraper.htm>..
- GNU Image Manipulation Program. 2001.** *GIMP*. [En línea] 2001. [Citado el: 10 de Diciembre de 2007.]
- HTMLPOINT.com. 1997-2006.** *¿Qué es Apache?* [En línea] 1997-2006. [Citado el: 10 de Diciembre de 2007.] <http://www.htmlpoint.com/faq/apache/01.htm>.
- <http://www.netsupportsoftware.com/>. [En línea] [Citado el: 15 de 03 de 2008.]
<http://www.netsupportsoftware.com/>.
- Lago, Ramiro. 2007.** *Patrón "Modelo-Vista-Controlador"*. [En línea] 2007. [Citado el: 10 de Diciembre de 2007.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
- López Guzmán, Clara. 2000.** *Modelo para el Desarrollo de Bibliotecas Digitales Especializadas*. [En línea] Marzo de 2000. [Citado el: 20 de Enero de 2008.] http://www.bibliodgsca.unam.mx/tesis/tes7c1lg/sec_26.htm.
- Maestros del Web. 1997-2007.** *Conociendo los principales editores de páginas Web del mercado*. [En línea] 1997-2007. [Citado el: 10 de Enero de 2008.] <http://www.maestrosdelweb.com/editorial/conociendo-los-principales-editores-de-paginas-web-del-mercado/>.
- Microsoft. 2006.** *La Arquitectura Orientada a Servicios (SOA) de Microsoft*. [En línea] Diciembre de 2006. [Citado el: 28 de Marzo de 2008.] http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf.
- pgAdmin PostgreSQL Tools . pgAdmin.** [En línea] [Citado el: 10 de Diciembre de 2008.] <http://www.pgadmin.org/>.
- Pitschke, Jurguen. 2002-2008.** *Visual Paradigm Suite*. [En línea] 2002-2008. [Citado el: 20 de Enero de 2008.] <http://www.visual-paradigm.eu/content/view/12/1/lang,en/>.
- ProgramacionWeb.net . 2003-2008.** *MVC - Modelo Vista Controlador*. [En línea] 2003-2008. [Citado el: 9 de Enero de 2008.] <http://www.programacionweb.net/articulos/articulo/?num=505>.
- San Félix, Alvaro del Castillo. 2000.** *El servidor de Web Apache: Introducción práctica*. [En línea] 2000. [Citado el: 8 de Enero de 2008.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>.
- Sánchez González, Carlos. 2004.** *ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras*. [En línea] 28 de Septiembre de 2004. [Citado el: 20 de Enero de 2008.] <http://oness.sourceforge.net/proyecto/html/ch03s02.html>.
- Tedeschi, Nicolás. 2008.** *¿Qué es un Patrón de Diseño?* . [En línea] 2008. [Citado el: 10 de Mayo de 2008.] http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2828/default.aspx.

Visual Paradigm. *Visual Paradigm for UML*. [En línea] [Citado el: 10 de Diciembre de 2007.] <http://www.visual-paradigm.com/product/vpuml/>.

Vizcaíno, Aurora, García, Felix Óscar y Caballero, Ismael. *Una Herramienta CASE para ADOO: Visual Paradigm*. [En línea] [Citado el: 27 de Noviembre de 2007.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.

Welling, Luke y Thomson, Laura. *Desarrollo Web con PHP y MySQL*. [En línea] [Citado el: 8 de Enero de 2008.] <http://bibliodoc.uci.cu/pdf/reg02819.pdf>.

World Wide Web Consortium. 1994-2005. *Guía Breve de Tecnologías XML*. [En línea] 1994-2005. [Citado el: 25 de Noviembre de 2007.] <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>.

—. **1994-2005**. *Guía Breve de Servicios Web*. [En línea] 1994-2005. [Citado el: 30 de Abril de 2008.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

Worsley, John y Drake, Joshua. 2001. *¿Qué es PostgreSQL?* [En línea] 2001. [Citado el: 29 de Noviembre de 2007.] <http://www.sobl.org/traduccion/practical-postgres/node12.html>.

—. **2001.** *Características de PostgreSQL*. [En línea] 2001. [Citado el: 29 de Noviembre de 2007.] <http://www.sobl.org/traduccion/practical-postgres/node19.html>.

XperimentoS. *Los mejores 12 Frameworks JavaScript*. [En línea] [Citado el: 25 de Noviembre de 2007.] <http://www.xperimentos.com/2007/09/04/los-mejores-12-frameworks-javascript/>.

ANEXOS

Anexo_1: Diagramas de secuencia del diseño.

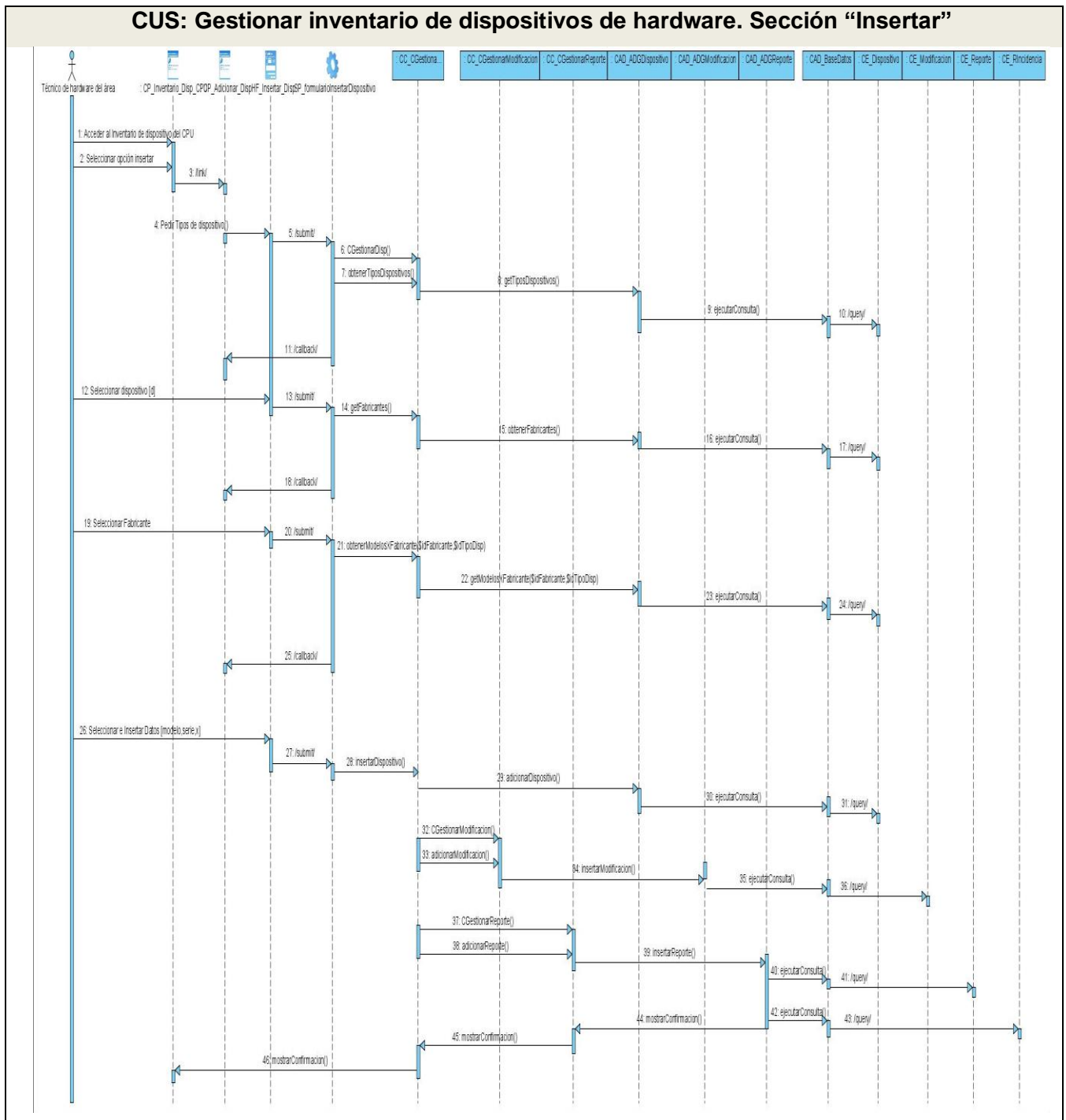


Figura 39: Diagrama de secuencia del diseño CUS "Gestionar inventario de dispositivos de hardware". Sección Insertar.

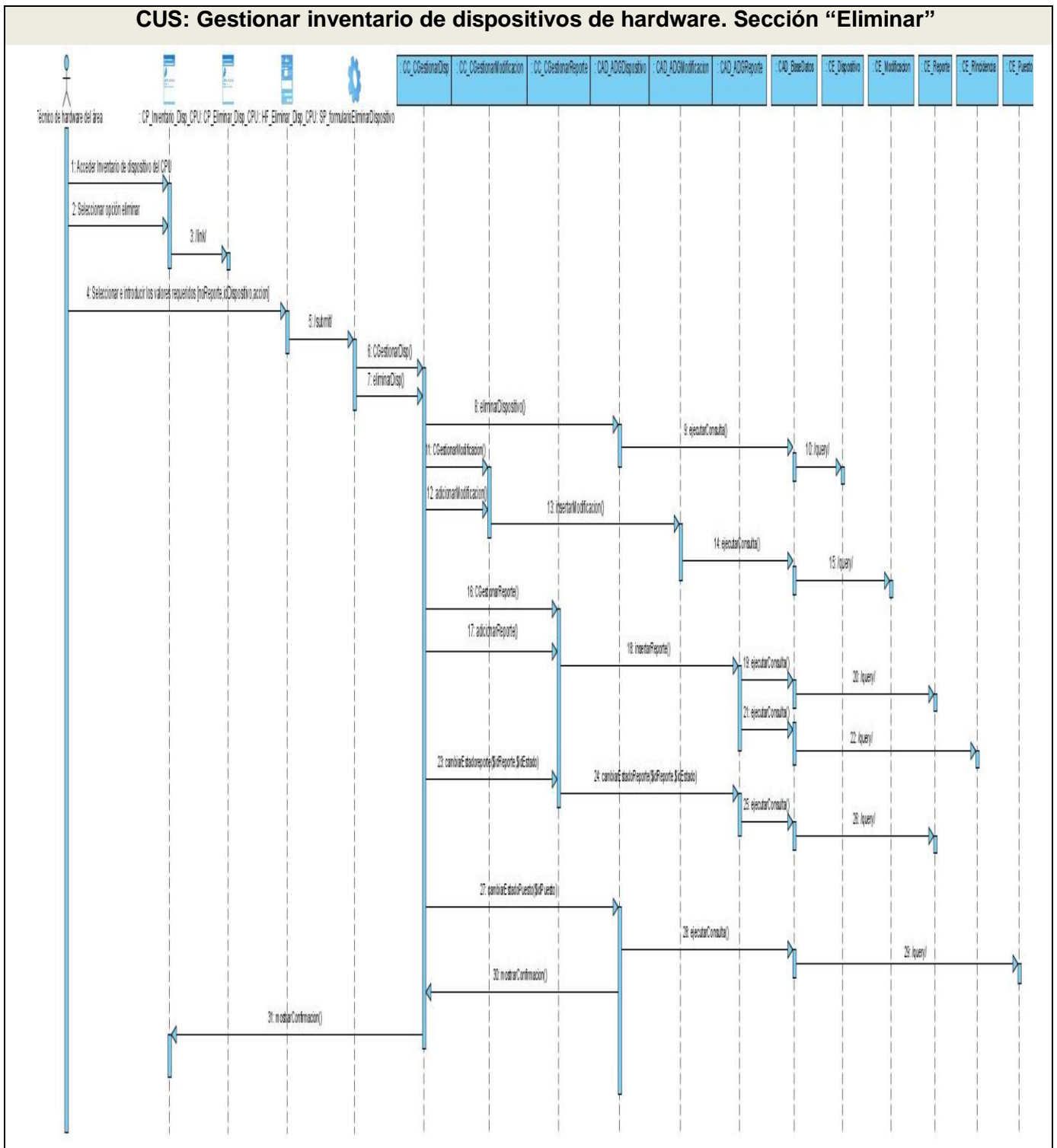


Figura 40: Diagrama de secuencia del diseño CUS "Gestionar inventario de dispositivos de hardware". Sección Eliminar.

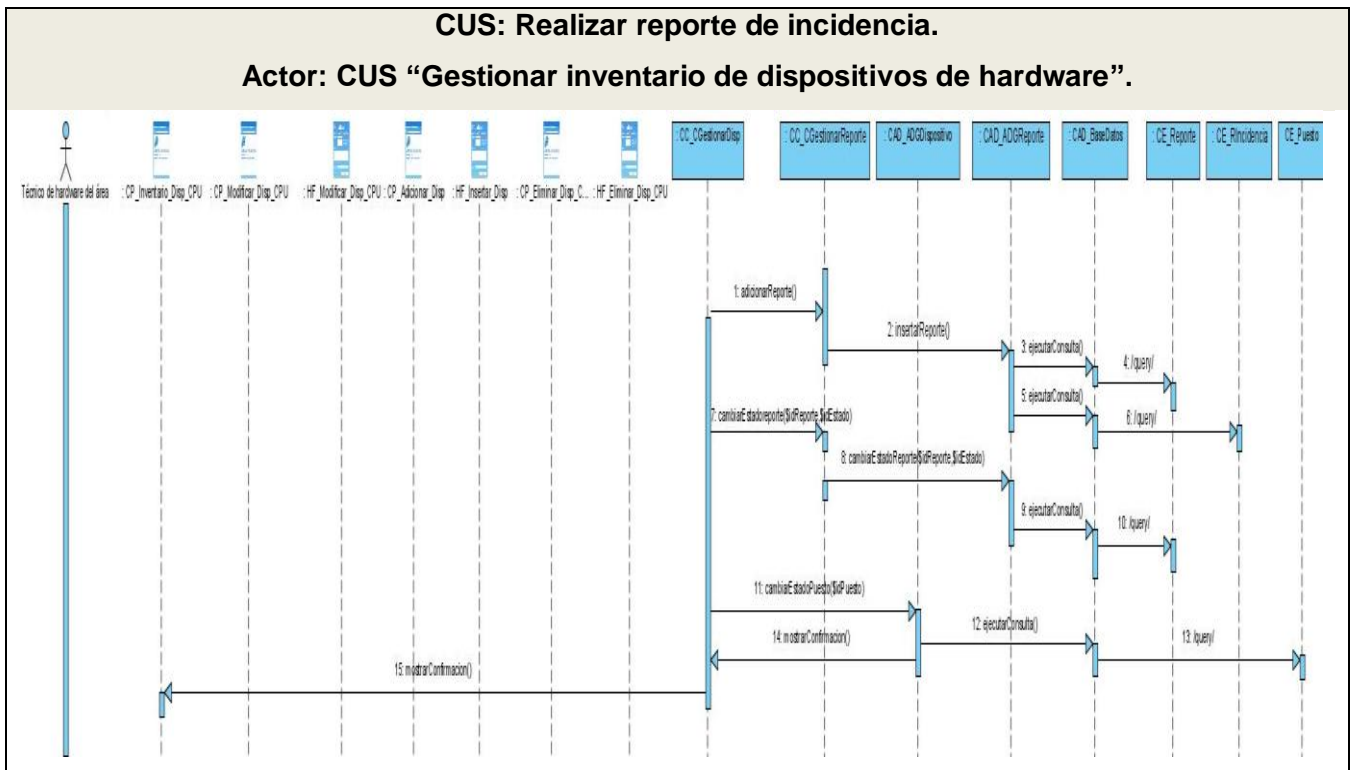


Figura 42: Diagrama de secuencia del diseño CUS “Realizar reporte de incidencia”. (Interno).

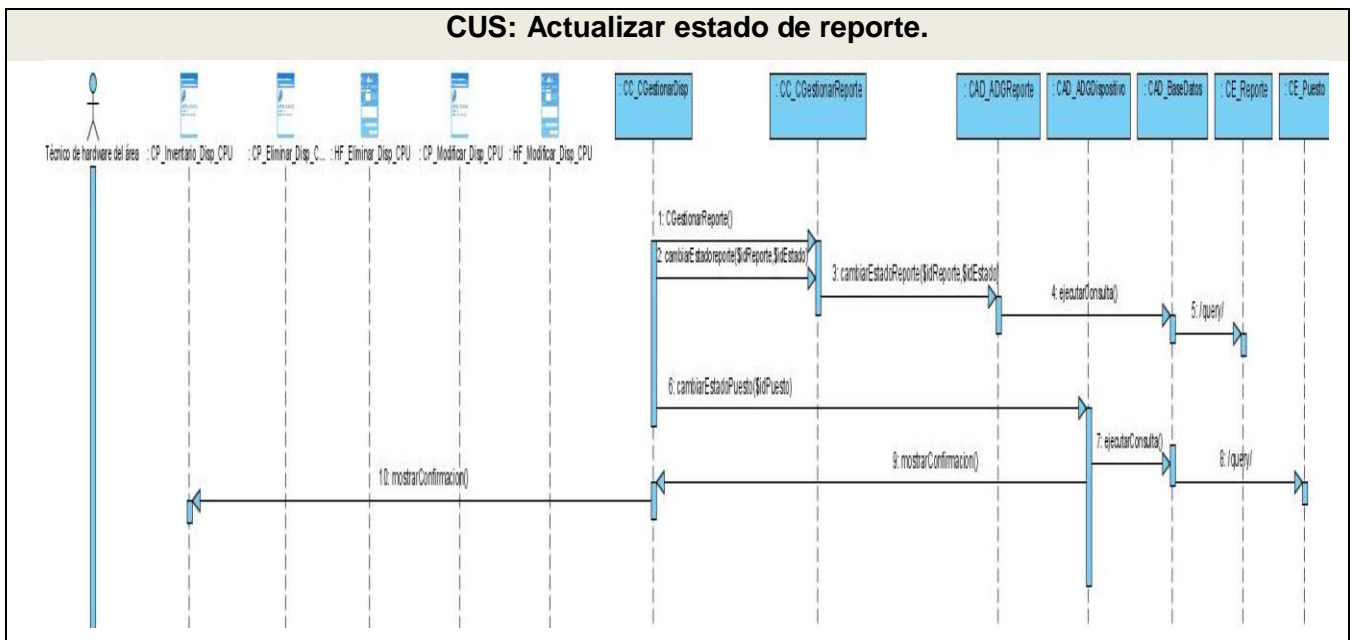


Figura 43: Diagrama de secuencia del diseño CUS “Actualizar estado de reporte.”

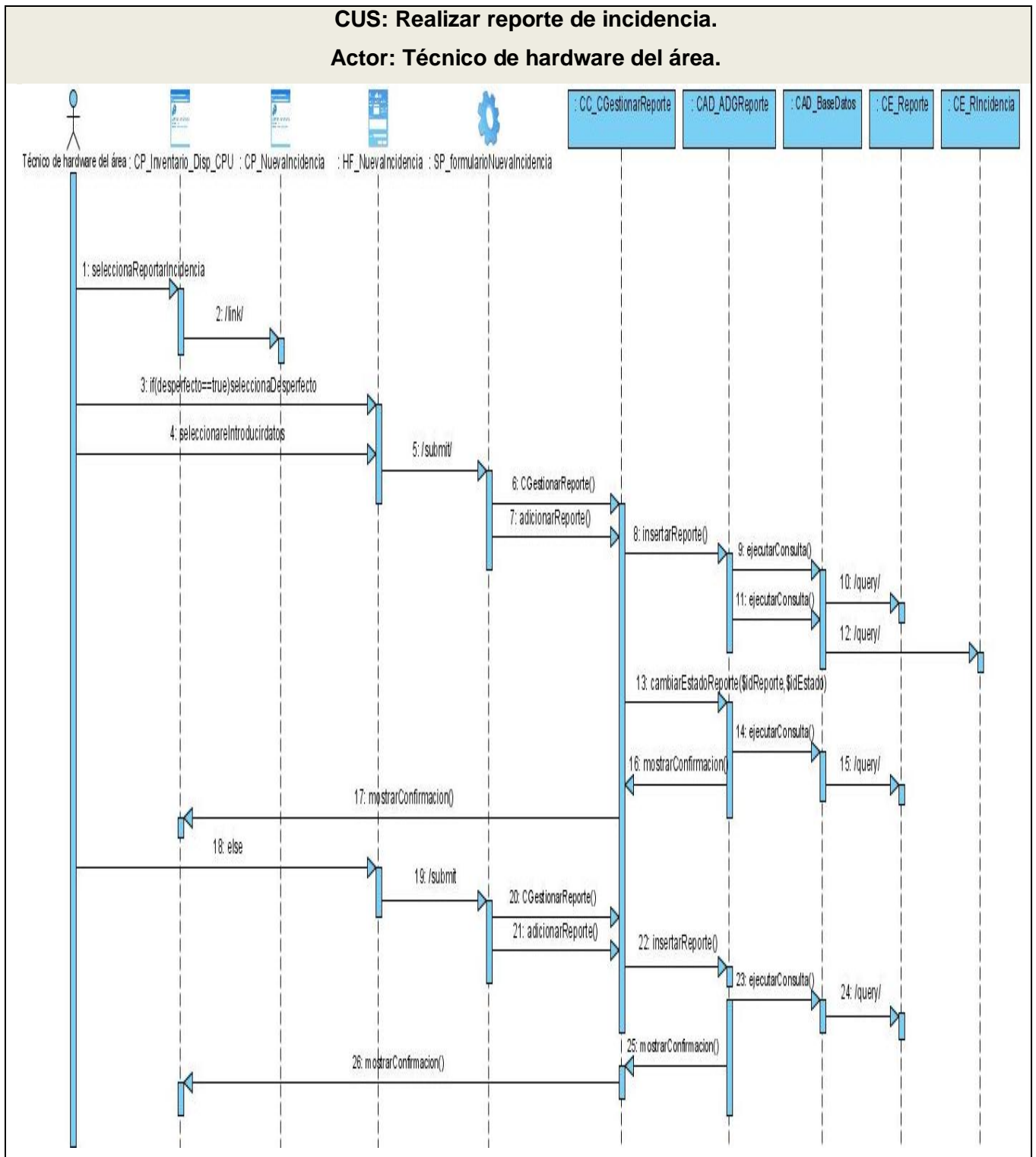


Figura 44: Diagrama de secuencia del diseño CUS “Realizar reporte de incidencia”.

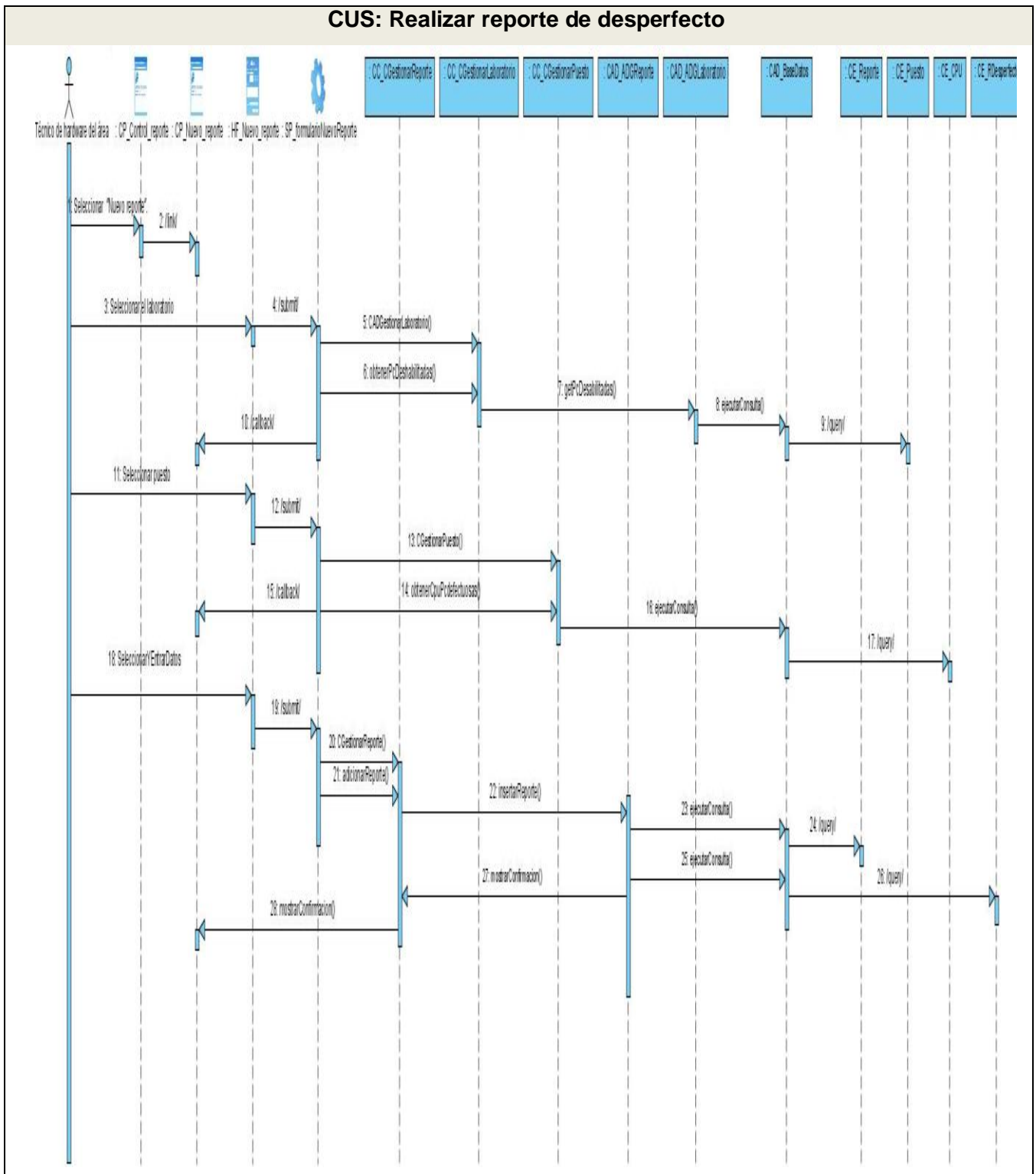


Figura 45: Diagrama de secuencia del diseño CUS "Realizar reporte de desperfecto."

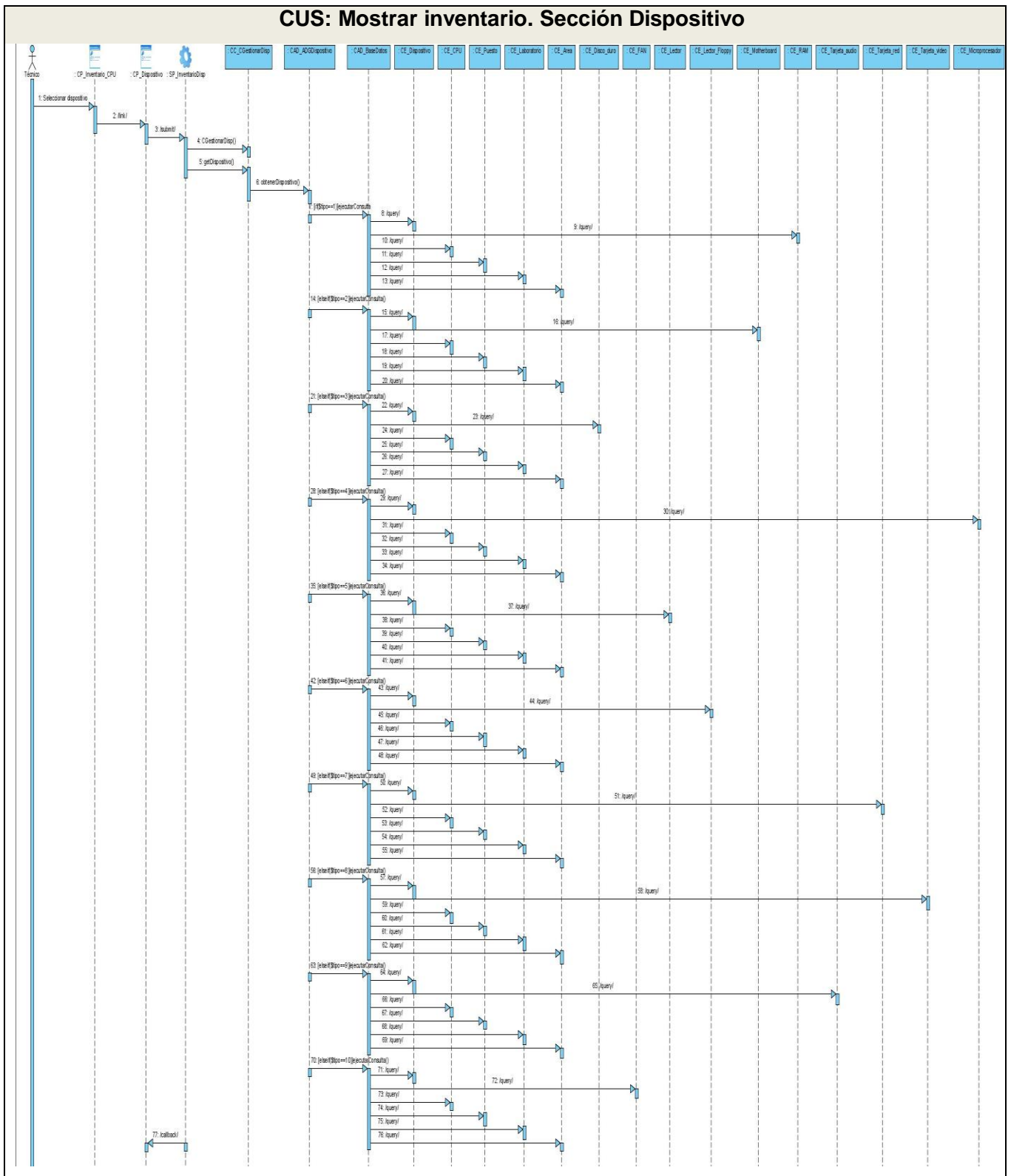


Figura 46: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección Dispositivo.

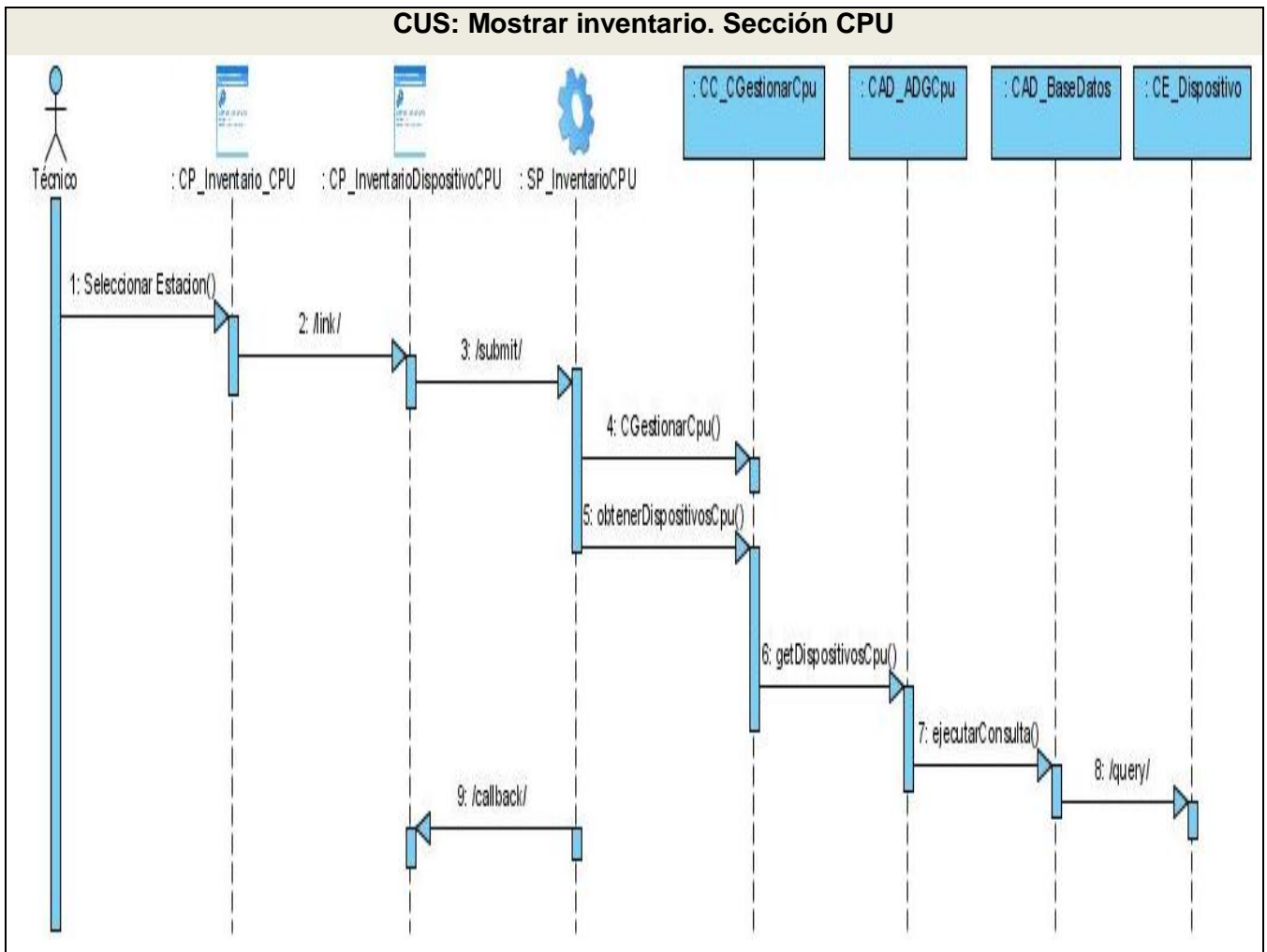


Figura 47: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección CPU.

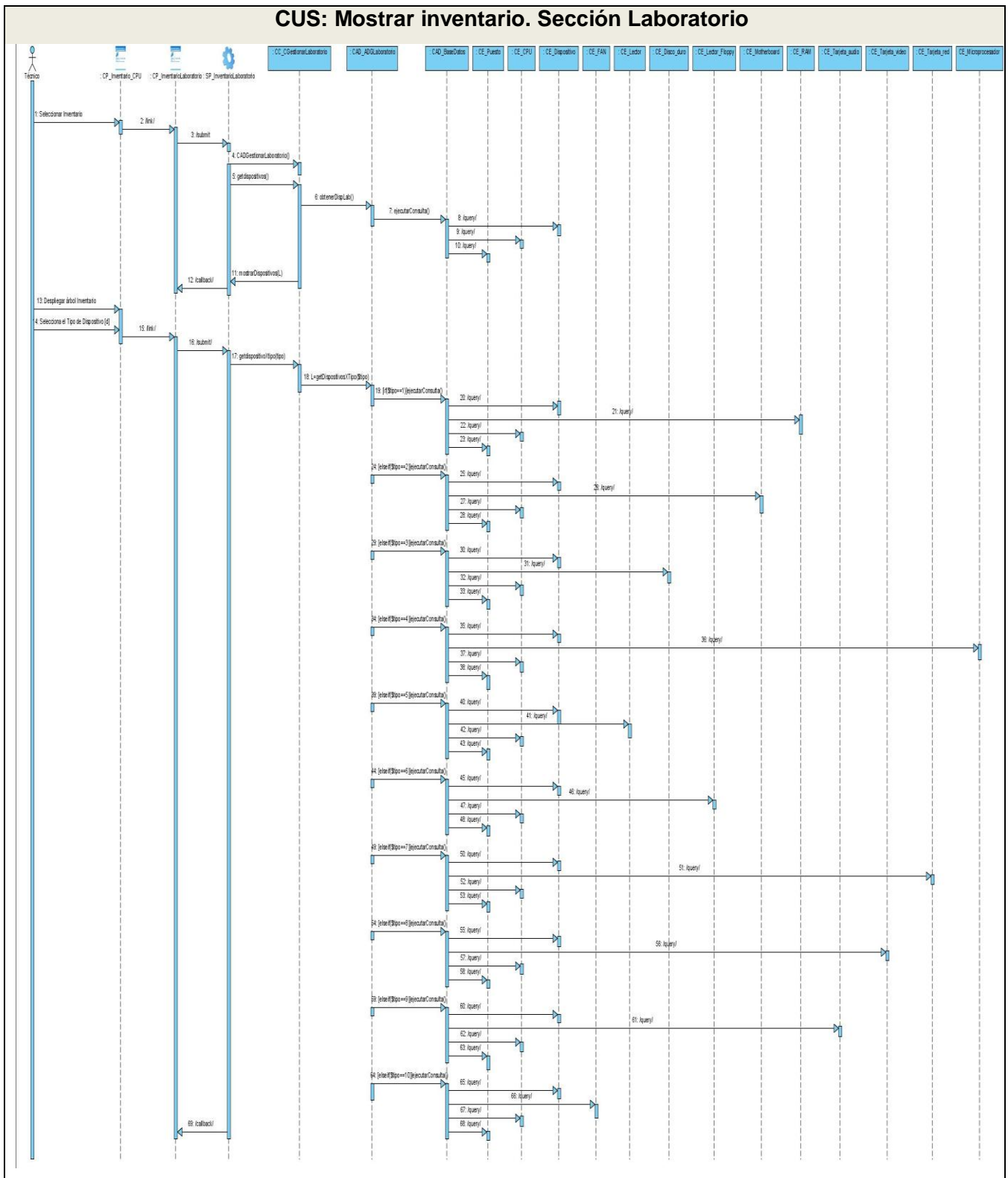


Figura 48: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección Laboratorio.

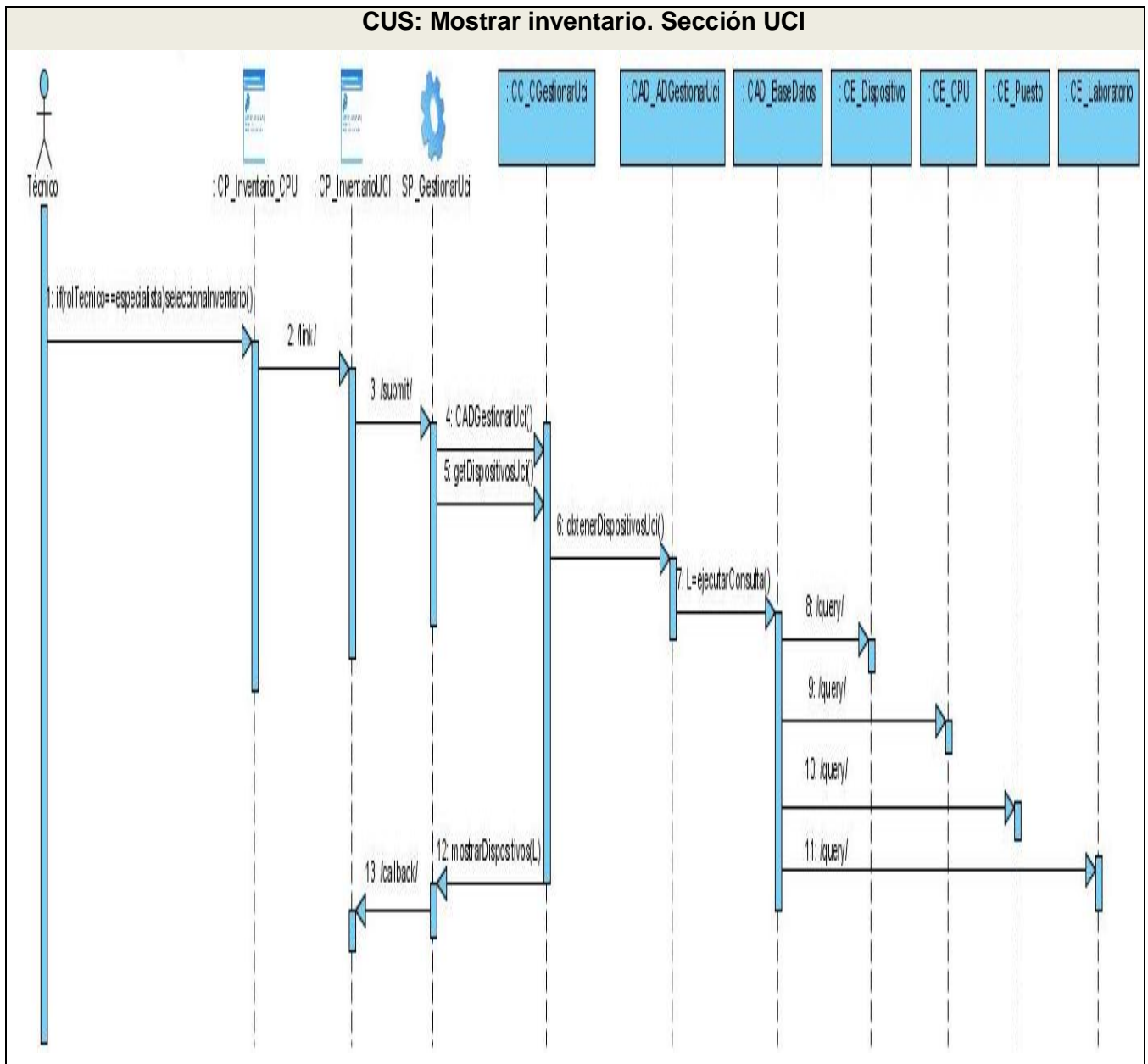


Figura 50: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección UCI.

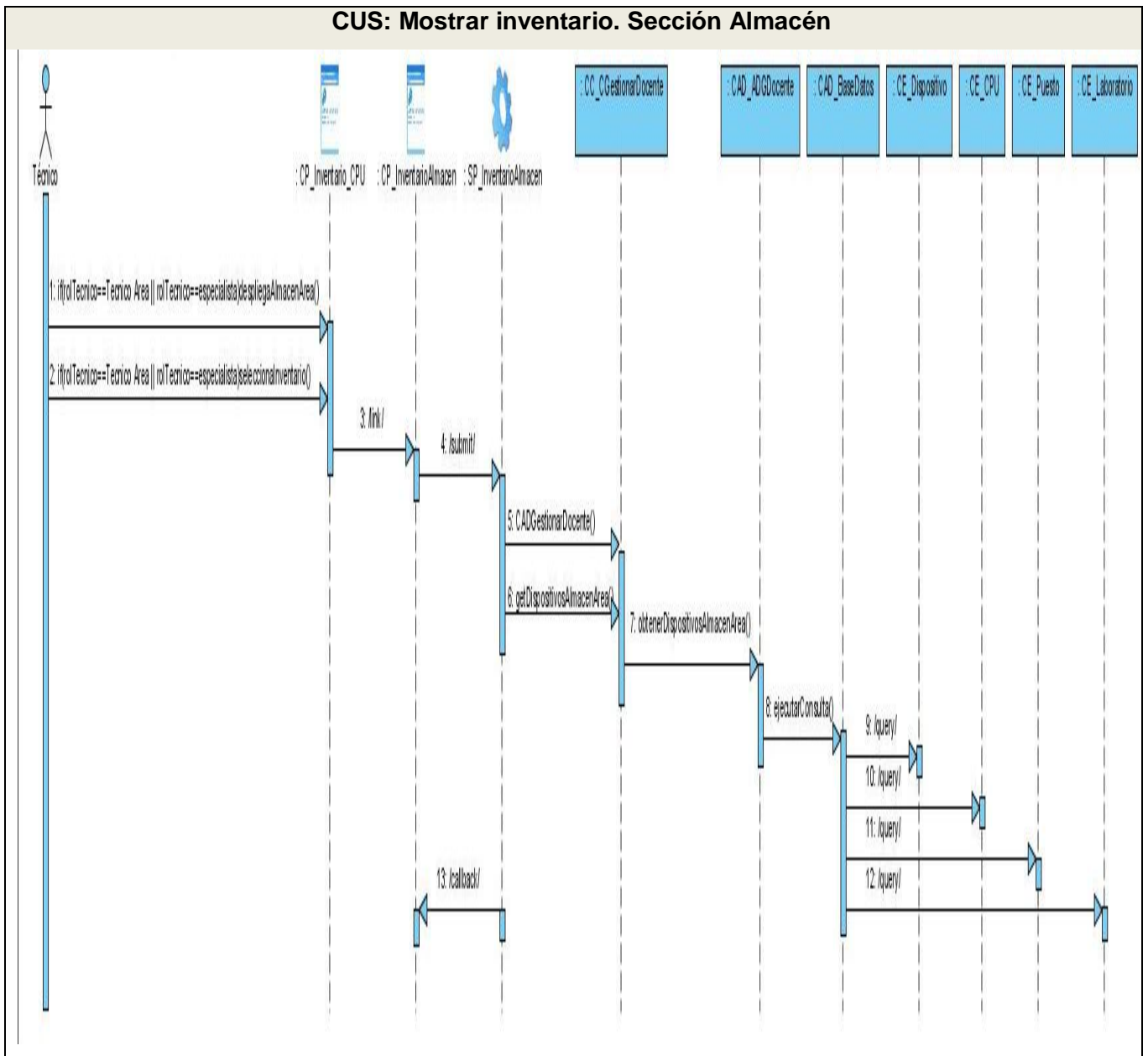


Figura 51: Diagrama de secuencia del diseño CUS “Mostrar inventario”. Sección Almacén.

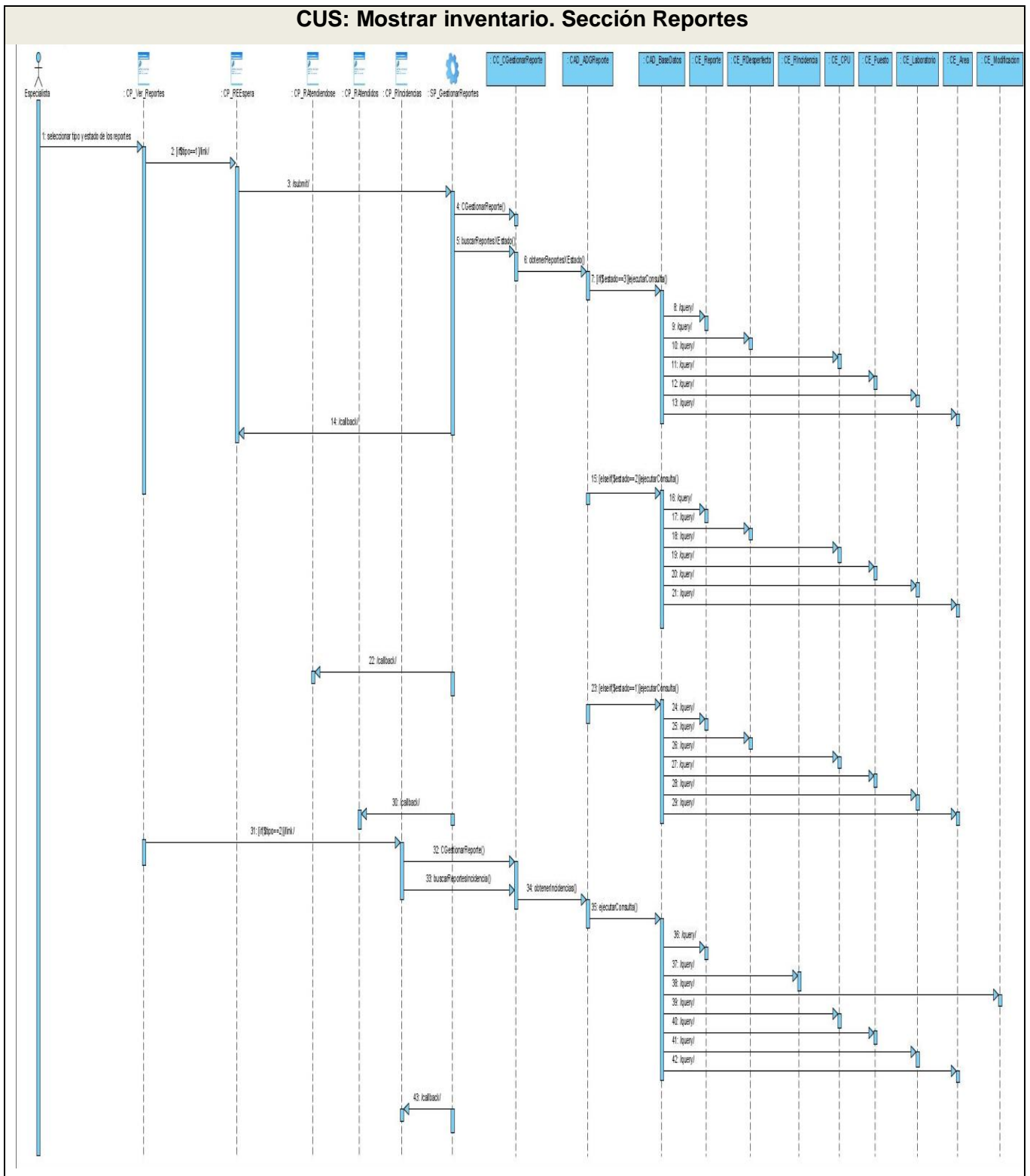


Figura 52: Diagrama de secuencia del diseño CUS “Mostrar reportes”. Sección reportes.

Anexo_2: Diagrama Entidad-Relación.

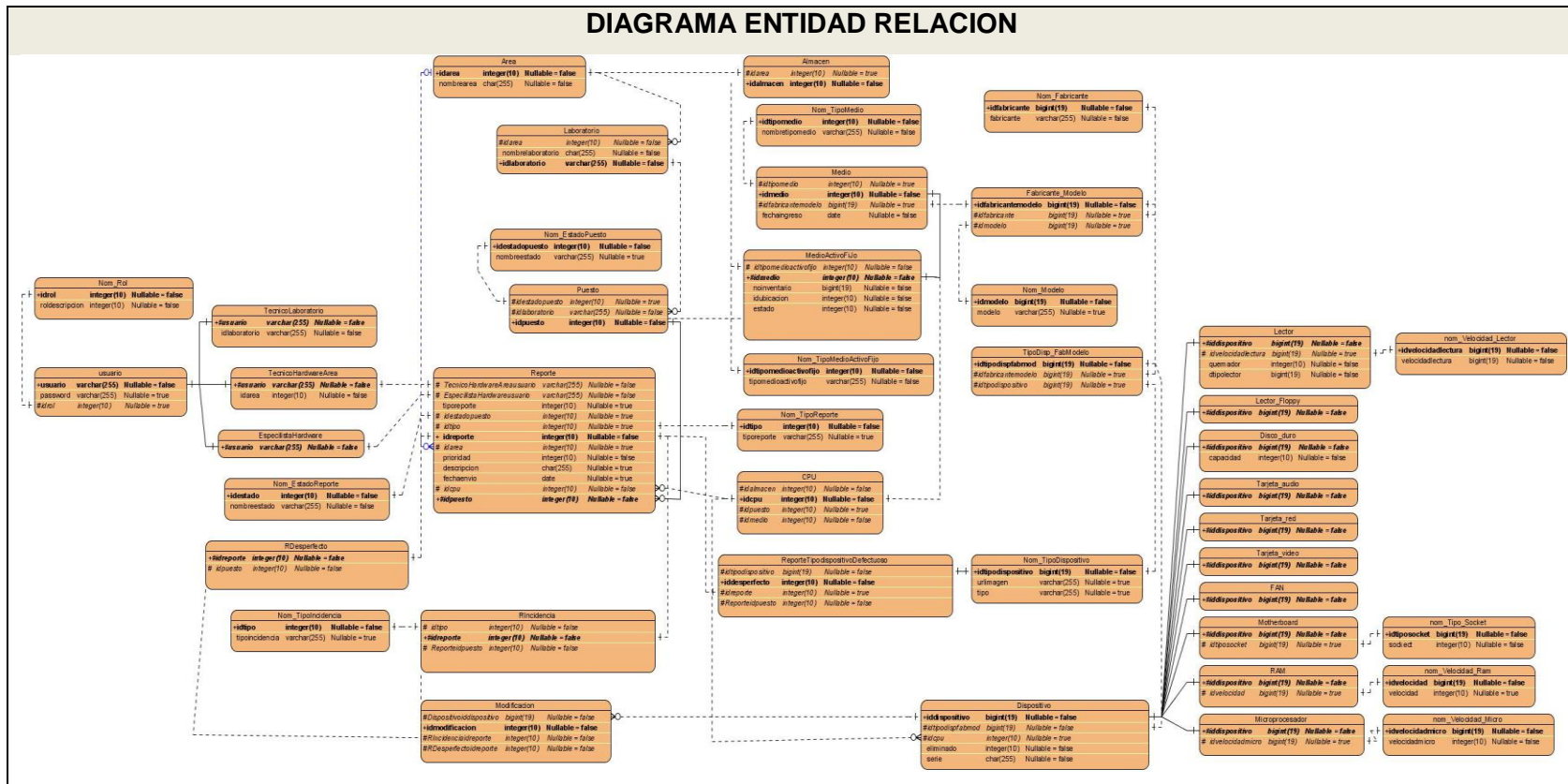


Figura 53: Diagrama Entidad-Relación.

GLOSARIO DE TÉRMINOS

AppServ: Es una herramienta Open Source para Windows que facilita la instalación de Apache, MySQL y PHP en una sola herramienta.

BIOS: Sistema Básico de entrada/salida Basic Input-Output System, es un código de interfaz que localiza y carga el sistema operativo en la RAM; es un software muy básico instalado en la placa base que permite que ésta cumpla su cometido. Proporciona la comunicación de bajo nivel, y el funcionamiento y configuración del hardware del sistema que, como mínimo, maneja el teclado y proporciona salida básica (emitiendo pitidos normalizados por el altavoz de la computadora si se producen fallos) durante el arranque.

Código abierto (open source): es el término con el que se conoce al software distribuido y desarrollado libremente.

DOM (Document Object Model): es esencialmente un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML.

Eclipse: es un entorno de desarrollo integrado de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores

Framework: En el desarrollo de software, es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

GIF (CompuServe GIF o Graphics Interchange Format): es un formato gráfico utilizado ampliamente en la World Wide Web, tanto para imágenes como para animaciones.

Herramientas CASE: las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y de dinero.

HTML: es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Etiquetas de Hipertexto¹. Es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

IDE (Integrated Development Environment) : es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

jQuery: es un nuevo tipo de bibliotecas de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones, y agregar interacción con la tecnología AJAX a nuestras páginas Web.

JSON (JavaScript Object Notation): es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Lenguaje de marcado: es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

Licencia BSD: es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre. Permite el uso del código fuente en software no libre.

MySQL: es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Oracle: es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

PgSQL (PostgreSQL Structured Query Language): es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las sentencias SQL básicas.

PHP Data Objects o PDO: es una extensión que provee una capa de abstracción de acceso a datos para PHP 5, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos manejadores de bases de datos.

Plug-in : es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como controlador en una aplicación, para hacer así funcionar un dispositivo en otro programa.

Prototype: es un framework escrito en JavaScript que se orienta al desarrollo sencillo y dinámico de aplicaciones Web. Es una herramienta que implementa las técnicas AJAX.

Scripts: guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades.

Slots: En computación, ranura utilizada para dar soporte de conexión al microprocesador.

SQL: Lenguaje de consulta estructurado en inglés (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

SVG (Scalable Vector Graphics): es un lenguaje para describir gráficos vectoriales bidimensionales, tanto estáticos como animados (estos últimos con ayuda de SMIL), en XML.

UNIX: es un sistema operativo portable, multitarea y multiusuario;

W3C: World Wide Web Consortium.

WxWidgets: son bibliotecas multiplataforma y libres, para el desarrollo de interfaces gráficas programadas en lenguaje C++.

XAMPP: es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los interpretes para lenguajes de script: PHP y Perl. El programa esta liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas.

XHTML (eXtensible Hypertext Markup Language): es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web

XMLHttpRequest (XHR): es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web.

XSLT o Transformaciones XSL: es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.

Yahoo User Interface (YUI): serie de librerías escritas en JavaScript, para la construcción de aplicaciones interactivas (RIA). Liberadas bajo licencia BSD por parte de la compañía Yahoo. Dichas librerías son utilizadas para el desarrollo Web específicamente para ser usadas como la programación de aplicaciones de escritorio, con componentes vistosos y personalizables y con una amplia implementación con AJAX.