

# Universidad de las Ciencias Informáticas

## Facultad 5



**Título:**

## **Sistema de gestión del Departamento de Tecnología de la Facultad 5**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autores:** Nivardo Ibarra Ocaña

Ernesto Marrero Vera

**Tutores:** Ing. Gerandys Hernández Casanova

Ing. Yoan Parra Nápoles

La Habana, Junio 2013



*"No se vive celebrando victorias sino superando derrotas"*

## **Declaración de Auditoría.**

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 \_\_\_\_\_.

Ernesto Marrero Vera.

Nivardo Antonio Ibarra Ocaña

\_\_\_\_\_  
**Firma del Autor**

\_\_\_\_\_  
**Firma del Autor**

Ing. Gerandys Hernández Casanova

Ing. Yoan Parra Nápoles.

\_\_\_\_\_  
**Firma del Tutor**

\_\_\_\_\_  
**Firma del Tutor**

**Datos de Contacto.**

**Tutor:** Ing. Gerandys Hernández Casanova

**Graduado de:** Ingeniero en Ciencias Informáticas

**Institución:** Universidad de las Ciencias Informáticas

**E-mail:** ghernandez@uci.cu

**Tutor:** Ing. Yoan Parra Nápoles

**Graduado de:** Ingeniero en Ciencias Informáticas

**Institución:** Universidad de las Ciencias Informáticas

**E-mail:** yparra@uci.cu

## **Agradecimientos.**

### *De Marrero*

*A mis padres Orelbe y Eneida por haberse sacrificado tanto por mí, por estar siempre a mi lado apoyándome en las buenas y en las malas, dándome su amor y confianza en todo momento, por darme el mejor regalo que le puedan dar a una persona, la vida y la oportunidad de vivirla plenamente. A quienes nunca vacilaron para darme la mano en cualquier circunstancia. A quienes con sus palabras y optimismo me levantaban los ánimos y me daban fuerzas para seguir adelante. A quienes me enseñaron todas las cosas que sé y me inculcaron valores importantes que me han ayudado a ser la persona que soy. Gracias a ambos por estar siempre ahí cuando los necesité.*

*A mi tía Diley que es una segunda madre para mí te quiero tía.*

*A mi abuelo, mis tías y tíos, por mantenerse siempre a mi lado dándome todo el apoyo que necesitaba para estar hoy donde estoy.*

*A mi hermano Erick y a mi primo Alejandro gracias por estar siempre a mi lado y darme el apoyo que siempre he necesitado.*

*Al mis tutores gracias por aguantarme todo este tiempo.*

*A los colegas.*

*De Nívardo*

*A mi madre y a mi abuela que siempre han creído en su varón, que me lo han dado todo para estar hoy donde estoy. Antes no pudo ser como queríamos pero bueno sucedió que es lo que importa. Abuela mima te dije que íbas a ver a tu predilecto como dice duni ingeniero .Mi vida va dedicada a ustedes. Perdón por todas las veces que he fallado.*

*A mi papá Robert.*

*A mi papá Nívardo.*

*A las locas de Vista Alegre que me malcrian siempre que estoy allá.*

*A mi hermano.*

*A mi tía Sonia y a mi tía Gisella.*

*A mis amigos de siempre el vlado, el marcelo, ramiro y carlos.*

*A todos mis compañeros de la universidad.*

*A marrero por todo el trabajo que pasamos juntos.*

*Al gera y al parra, gracias por confiar en mí y apoyarme estos meses.*

## **Dedicatoria.**

### ***De Marrero***

*Dedico este trabajo a mis padres, mi hermano, mi tía, mi primo, mi abuelo y demás familiares que siempre confiaron en mí y dieron todo para que me superara profesionalmente. En especial a mi madre y mi padre que con su perseverancia me han dado fuerzas para seguir adelante hasta lograr mi objetivo, que con sus consejos han estado presente en todos los momentos buenos y malos, confiando en que podía vencer los obstáculos que se presentaran.*

### ***De Nívardo***

*Este trabajo va dedicado a mi abuela y a mi madre, quería dedicárselo a una sola persona pero no puedo separar al amor que siento por ustedes. Las que nunca me han reprochado nada, de las que nunca he recibido un regaño ni cuando lo merecía. Con ustedes he aprendido a amar a la familia y ponerla por encima de todo, a ser hombre de bien. Gracias por la vida que me han dado, podría escribir un millón de cosas buenas de ustedes pero saben que no soy de muchas palabras y se me ponen los ojos llorosos solo de pensar todo lo que significan para mí. Mil veces gracias, firma: el varón.*

## **Resumen.**

Actualmente para toda empresa o institución resulta esencial el control de sus medios básicos ya que los mismos se utilizan de manera continua en sus actividades diarias. La Universidad de las Ciencias Informáticas (UCI), líder en el desarrollo tecnológico en el país y privilegiada en cuanto a cantidad y calidad de medios disponibles se hace imprescindible la gestión práctica y segura de los mismos. Aunque en la UCI se han tomado decisiones relevantes en este aspecto, todavía estos esfuerzos no han sido suficientes para erradicar todos los problemas.

En este documento se presentan los resultados de una investigación para el desarrollo de un sistema que permita apoyar la gestión de los medios básicos, partiendo de un análisis de diferentes aplicaciones, herramientas y metodologías. Se diseñó e implementó una serie de requisitos identificados a partir del problema encontrado.

**Palabras Clave:** Aplicación Web, Gestión de Medios Básicos, Symfony2.

## Índice.

Resumen.....	VI
Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	4
1.1.    Introducción.....	4
1.2.    Sistemas Similares Existentes.....	4
1.2.1.    GLPI.....	4
1.2.2.    VERSAT Sarasola.....	5
1.2.3.    RODAS XXI.....	5
1.3.    Metodología de Desarrollo de Software.....	6
1.3.1.    Rational Unified Process(RUP).....	7
1.3.2.    Extreme Programming (XP).....	8
1.4.    Lenguaje de Modelado.....	9
1.5.    Herramienta para el modelado.....	9
1.5.1.    Rational Rose.....	10
1.5.2.    Visual Paradigm.....	10
1.6.    Lenguaje de programación.....	11
1.8.1.    PHP(“PHP Hypertext Pre-processor”).....	11
1.8.2.    HTML.....	12
1.8.3.    CSS(“Hojas de Estilo en Cascada”).....	12
1.8.4.    JavaScript.....	12
1.9.    Entorno de Desarrollo Integrado.....	13
1.9.1.    Eclipse.....	13
1.9.2.    NetBeans.....	13
1.10.    Servidor Web.....	13
1.11.    Marco de trabajo (Framework).....	14
1.11.1.    CakePHP.....	14
1.11.2.    Symfony 2.....	15
1.12.    Sistema Gestor de Base de Datos.....	16
1.12.1.    Microsoft SQL Server.....	16
1.12.2.    PostgreSQL.....	16

---

Conclusiones Parciales. ....	17
Capítulo 2. Características y Diseño del Sistema. ....	18
2.1. Introducción. ....	18
2.2. Descripción del Sistema. ....	18
2.3. Modelo del Dominio. ....	18
2.4. Requisitos Funcionales(RF). ....	19
2.5. Requisitos No Funcionales (RNF). ....	21
2.6. Actores del Sistema. ....	22
2.7. Diagrama de Caso de Uso del Sistema. ....	22
2.8. Descripción de Caso de Uso. ....	23
2.9. Patrones de Diseño. ....	26
2.9.1. Patrón Data Mapper. ....	27
2.9.2. Patrones GRASP ( <i>General Responsibility Assignment Software Patterns</i> ). ....	27
2.10. Patrones de Arquitectura. ....	28
2.10.1. Patrón Modelo Vista Controlador (MVC). ....	28
2.11. Diseño. ....	30
2.11.1. Diagrama de colaboración. ....	30
2.11.2. Diagrama de clase del diseño. ....	31
2.11.3. Diagrama de Clases Persistentes. ....	32
2.11.4. Modelo de Datos. ....	32
2.11.5. Diagrama Entidad Relación (DER). ....	33
Conclusiones Parciales. ....	33
Capítulo 3. Implementación y Prueba. ....	34
3.1. Introducción. ....	34
3.2. Diagrama de Despliegue. ....	34
3.3. Diagrama de Componente. ....	34
3.4. Código Fuente. ....	35
Estándares de codificación. ....	35
3.4.1. Nomenclatura. ....	35
3.4.2. Estructura. ....	36
3.5. Pruebas de Software. ....	36
Niveles de Prueba. ....	37

---

Métodos de Prueba.....	37
3.6. Pruebas Realizada.....	37
3.7. Diseño de Caso de Prueba.....	37
3.8. Resultado de las Pruebas.....	40
Conclusiones Parciales.....	41
Conclusiones.....	42
Recomendaciones.....	43
Glosario de términos.....	44
Referencias Bibliográficas.....	45
Anexos.....	47

## **Introducción.**

En el escenario del desarrollo tecnológico cubano actual, la Universidad de las Ciencias Informáticas (UCI) desempeña un rol vital. Por la gran cantidad de medios informáticos con los que cuenta, es de gran importancia la gestión de los mismos, los cuales, desde los inicios de la universidad fueron controlados por la dirección de laboratorios, la cual estaba estructurada de la siguiente forma, 5 áreas docentes y 2 productivas (Infraestructura Productiva y Producción), 3 departamentos, uno para soluciones informáticas, otro de control y soporte técnico y el otro de recursos humanos y la dirección a la que todos estaban subordinados. A partir del curso 2012/2013 se decidió en la universidad pasar la tecnología y los recursos humanos de dicha dirección a cada una de las facultades con el objetivo de lograr una mejor organización en los procesos docentes y productivos de cada facultad según sus necesidades, mejorando el tiempo de respuesta ante algún problema o necesidad de cambio.

Para la ejecución de la transferencia mencionada se creó el Departamento de Tecnología en cada facultad, el cual está estructurado por: un jefe de departamento, los especialistas encargados de software, hardware y recursos humanos, un responsable de medios básicos y los técnicos de laboratorios, así como los asesores tecnológicos de cada centro de producción. Este departamento es el encargado de controlar toda la parte técnica de las facultades: laboratorios, oficinas, departamentos docentes, aulas, centros productivos y el personal técnico vinculado a esta.

En aras de preservar los medios puestos en función del desarrollo profesional de los estudiantes y trabajadores de la UCI, se hace necesario documentar y facilitar procesos tales como el alta, baja y préstamos o traslados de los medios por cada laboratorio, así como el control de útiles por parte de los departamentos de tecnología, que también se encargan de realizar los reportes de las afectaciones de medios y útiles.

En estos momentos estos procesos se realizan de forma manual por parte del responsable de los medios básicos, encargado de manejar esta información.

- ✓ Los datos se almacenan en documentos Excel, Word y en formato duro.
- ✓ No se tiene un claro control de los medios básicos, debido a los diferentes medios de almacenamiento.
- ✓ La información está desorganizada y desactualizada producto al cúmulo de esta.

- ✓ El proceso es lento y engorroso.
- ✓ Se cometen muchos errores humanos a la hora de almacenar los datos de los medios básicos.

Tomando en cuenta la problemática planteada surge como **problema a resolver**: ¿Cómo automatizar los procesos de gestión de información de los medios básicos del Departamento de Tecnología de la Facultad 5?

Constituye **objeto de estudio** del problema anterior: los procesos de organización y control de información en el Departamento de Tecnología de la Facultad 5, enmarcado en el **campo de acción**: los procesos de control de información sobre los medios básicos. Definiéndose como **objetivo general**: Desarrollar una aplicación web para automatizar la gestión de información de los medios básicos del Departamento de Tecnología de la Facultad 5.

## Tareas investigativas:

- ✓ Estudio bibliográfico del tema para la elaboración del marco teórico de la investigación a partir del estado del arte existente actualmente.
- ✓ Selección de la metodología y herramientas a emplear para llevar a cabo el desarrollo del software.
- ✓ Identificación de los requisitos para la selección de las características y funcionalidades del sistema.
- ✓ Elaboración del diseño de la aplicación para obtener un mejor entendimiento del sistema a desarrollar.
- ✓ Diseño de una Base de Datos para gestionar la información.
- ✓ Implementación del sistema para gestionar los medios básicos del Departamento de Tecnología de la facultad 5.
- ✓ Elaboración del diseño de casos de prueba partiendo del flujo de trabajo de cada uno de los requisitos identificados.
- ✓ Implementación de las pruebas para proporcionar información objetiva e independiente sobre la calidad del producto.
- ✓ Análisis de los resultados de las pruebas.

Nótese que la **idea a defender** plantea que con el desarrollo de una herramienta para automatizar la gestión de información de los medios básicos del Departamento de Tecnología

de la Facultad 5 se pudiese encontrar una respuesta para facilitar el trabajo de organización en el proceso del control de los medios básicos.

En cuya solución se utilizarán como **métodos científicos**:

## **Métodos Teóricos:**

- ✓ **Análisis Histórico-Lógico:** Se utiliza para hacer un estudio del estado del arte de las herramientas existentes para aplicaciones similares y sus usos en el ámbito nacional e internacional, así como las ventajas y desventajas que poseen.

## **Métodos Empíricos:**

- ✓ **Entrevistas:** Se realizan entrevistas a miembros del Departamento de Tecnología de la facultad 5 con el fin de comprender el problema existente.

Como **posible resultado** se obtendría una aplicación web que automatice el proceso de gestión de información de los medios básicos del Departamento de Tecnología de la Facultad 5.

## **El documento está estructurado por tres capítulos:**

**Capítulo 1. Fundamentación Teórica:** En este capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. Se hace un análisis de los lenguajes de programación, herramientas y tecnologías a usar en el desarrollo del sistema.

**Capítulo 2. Características y Diseño del Sistema:** En este capítulo se presenta la propuesta de solución, a través de los requerimientos del sistema. Se obtendrá el diseño de la base de datos y los diagramas de clases. Además se establece la arquitectura y los patrones de diseño a usar en la realización de la aplicación.

**Capítulo 3. Implementación y Pruebas:** En este capítulo se realizará la implementación de la solución del sistema y las pruebas necesarias para la validación de la calidad del mismo.

# *Capítulo 1. Fundamentación Teórica.*

---

## **Capítulo 1. Fundamentación Teórica.**

### **1.1. Introducción.**

En el presente capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. Se realiza un estudio de sistemas similares a la solución propuesta así como las tendencias de estos en la actualidad. Se describen además las tecnologías, herramientas, lenguaje y metodología a usar en el desarrollo del sistema.

### **1.2. Sistemas Similares Existentes.**

Tanto a nivel internacional como nacional se han desarrollado distintos sistemas que apoyan la gestión y control de medios básicos en una institución. Se dedican directamente a la gestión de estos o integran entre sus procesos otras funcionalidades.

#### **1.2.1. GLPI.**

**GLPI** (Gestionnaire Libre de Parc Informatiqué) que traducido al español es: Administrador Libre de Recursos Informáticos, con una interfaz para su administración. Dispone de funciones que facilitan el trabajo de los administradores así como un sistema de seguimiento de trabajo con mail de notificación. La herramienta está desarrollada para entornos apache-php-mysql, por lo que es instalada en servidores Windox y Linux, permite gestionar todo el soporte y mantenimiento de una empresa. Sus principales funcionalidades son llevar el inventario de todos los recursos técnicos y sus características almacenadas en una base de datos. Posee recursos que puedan facilitar el trabajo al administrador como: programar, solicitar tarea, realizar seguimientos de las mismas y enviar emails para las comunicaciones. Permite la administración de multi-usuario, sistemas de permisos y sistemas de exportación de pdf y hoja de cálculo.

Este sistema presenta deficiencias como: los usuarios que no son administradores o técnicos pueden modificar datos importantes, lo que lleva a la inseguridad de los medios bajo el sistema. Las actualizaciones de los datos del inventario real con los ya existentes implican tiempo por parte de las personas encargadas de los mismos y esto puede generar pérdida de eficiencia y eficacia a la hora de que la información del sistema corresponda con la real. En cuanto a las reservas no hay

## *Capítulo 1. Fundamentación Teórica.*

---

un filtro, permiso o sanción que permita impedir a un usuario tomar o no, en préstamo un equipo, ni un mecanismo que permita ver que está utilizando la misma.(1)

### **1.2.2. VERSAT Sarasola.**

Es un sistema de gestión contable financiero, comenzó su elaboración en julio de 1999 en Villa Clara y 2 años más tarde se utilizó por primera vez en el central George Washington. El nombre de Versat se debe a la versatilidad y Sarasola, hace honor a Andrés Sarasola, experimentado contador acreedor al Premio Nacional de Contabilidad. El software incluye varios módulos, Control de Activos Fijos e Inventarios, Finanzas y Planificación, Contabilidad General y Facturación. El resultado es un sistema integral, que se actualiza constantemente en función de nuevas demandas y facilita la organización y el control en el área económica. Unas de sus principales características es su confiabilidad, se divide en subsistemas que en el momento de configuración determinan quien tiene acceso a cada uno de los sistemas, posibilita rectificaciones y no desapareciendo el original. Este hace un cuadro sistemático donde lleva el conteo de lo que queda en el almacén. Actualmente se encuentra instalado en 3 700 entidades del país como el MINAZ (Ministerio del Azúcar), Comercio Interior, Agricultura, Industria Ligera. También otras 100 entidades de servicios médicos cubanos lo usan en Venezuela, Argelia y China. Representa un ejemplo de sustitución de importaciones en el sector de las aplicaciones informáticas. (2)

### **1.2.3. RODAS XXI.**

Desarrollado por la empresa cubana CITMATEL (Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados), el Sistema Integral Económico Administrativo RODAS XXI posibilita automatizar el funcionamiento de cualquier empresa. Ha sido diseñado para trabajar sobre los sistemas operativos: Windows XP, 2000 ó 2003, con un servidor con Microsoft SQL Server 2000. Cuenta actualmente con 8 módulos que pueden ser usados en su totalidad o cada uno de forma independiente:

- ✓ Finanzas
- ✓ Contabilidad

# Capítulo 1. *Fundamentación Teórica.*

---

- ✓ Activos Fijos
- ✓ Nóminas
- ✓ Inventario
- ✓ Facturación
- ✓ Recursos Humanos
- ✓ Telecobranzas

El módulo de Activos Fijos de RODAS XXI le permite tener un control detallado de los activos fijos de su entidad, realizando en el mismo momento que se registra un movimiento, su contabilización. Se pueden realizar todo tipo de operaciones de activos fijos con facilidad en el momento que se desee, generando el documento asociado al movimiento de que se trate de forma automática previa configuración del sistema para ello. Permite el control por separado de los activos fijos que se encuentran en almacén de los que se encuentran en explotación. Es posible además realizar ajustes a los activos fijos con facilidad mediante una opción que brinda el sistema.(3)

Tanto RODAS XXI como VERSAT Sarasola no constituyen una solución factible para el Departamento de Tecnología de la Facultad 5 ya que están desarrolladas bajo software propietario, por lo que no cumplen con la independencia tecnológica que se quiere alcanzar en el país. Además son aplicaciones de escritorio lo que implicaría que el usuario deba instalar las mismas en cada estación de trabajo.

### **1.3. Metodología de Desarrollo de Software.**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

Es como un libro de recetas de cocina, en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.(4)

# Capítulo 1. *Fundamentación Teórica.*

---

Una metodología es un conjunto de procedimientos que permiten producir y mantener un producto software, se definen una serie de pasos a seguir para obtener un software de calidad. Las metodologías se desarrollan con el objetivo de dar solución a los problemas existentes en la producción de software, que cada vez son más complejos. Estas abarcan procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de software. Sus procesos se descomponen a nivel de tareas o actividades elementales, donde cada tarea está identificada por un procedimiento que define la forma de llevarla a cabo.(5)

## 1.3.1. Rational Unified Process(RUP).

El Proceso Unificado, Rational Unified Process en inglés, y sus siglas RUP, es un proceso de ingeniería de software. Proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales dentro de un horario y presupuesto predecible.(6)

**Las características principales de RUP son:**

- ✓ **Guiado/Manejado por casos de uso:** La razón de ser de un sistema software es servir a usuarios; un caso de uso es una facilidad que el software debe proveer a sus usuarios y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo.
- ✓ **Centrado en arquitectura:** La arquitectura involucra los elementos más significativos del sistema. Se representa mediante varias vistas que se centran en aspectos concretos con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o que se desarrollan y maduran sus componentes.
- ✓ **Iterativo e Incremental:** RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.(7)

# Capítulo 1. Fundamentación Teórica.

## Fases:

- ✓ **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- ✓ **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- ✓ **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- ✓ **Transición:** El objetivo es llegar a obtener el release del proyecto.(8)

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los 3 restantes como de apoyo.

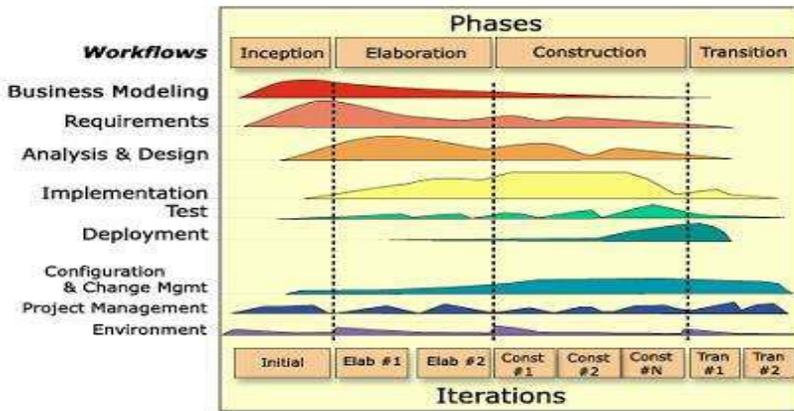


Figura1.1 Fases e Iteraciones de RUP.

### 1.3.2. Extreme Programming (XP).

Un desarrollo interesante en los últimos años ha sido la adaptación del método de desarrollo de prototipos rápidos en algo llamado XP. Esto es adecuado para los equipos de desarrollo de software de tamaño pequeño/medio y aborda algunas de las principales dificultades que encuentran los programadores. El principio más conocido que los profesionales deben tener en cuenta es la programación en parejas. Esto significa que toda actividad de diseño y programación se lleva a cabo por equipos de dos. (9)

La idea central detrás de XP es poner en marcha prácticas cotidianas que concilien la humanidad y la productividad. Lo que esto significa en la práctica es que se le da más importancia a la comunicación cara a cara y a la retroalimentación, que a la recogida de requisitos formales.

# Capítulo 1. *Fundamentación Teórica.*

---

Por otra parte, los defensores XP exponen en el valor de hacer algo sencillo y limpio hoy, en lugar de preocuparse por los problemas futuros que puedan suponer complejidad.(10)

Después de analizar estas metodologías de desarrollo de software se decidió utilizar RUP, ya que constituye un marco de trabajo genérico que puede ser adaptado a una gran diversidad de sistemas de software independientemente del tamaño del proyecto, el tipo de organización y los diferentes niveles de aptitud. Es una metodología adaptable al contexto y a las necesidades que permite tener bien claro y accesible el proceso de desarrollo que se quiere alcanzar. Además basa su trabajo principalmente en la documentación, donde en todo su ciclo de vida expone un conjunto de actividades que permite la obtención de todos los artefactos necesarios para el desarrollo de la aplicación.

## **1.4. Lenguaje de Modelado.**

Unified Modeling Language (UML), Lenguaje Unificado de Modelado traducido al español es el lenguaje más utilizado para el diseño orientado a objetos.

Permite a un diseñador describir el proyecto con una rica variedad de esquemas. Por ejemplo, los diagramas de clases UML pueden mostrar los elementos importantes o relevantes de las clases, y las relaciones entre ellos, de forma concisa e intuitiva. Otros tipos de diagramas UML pueden ilustrar cómo colaboran las clases entre sí y cómo los usuarios interactúan con objetos de clase. (11)

UML ofrece soporte para clases, clases abstractas, relaciones, comportamiento por interacción y empaquetamiento. Estos elementos se pueden representar mediante nueve tipos de diagramas, que son: de clases, de objetos, de casos de uso, de secuencia, de colaboración, de estados, de actividades, de componentes y de desarrollo. (12)

Debido a la selección de la metodología de desarrollo de software se hace necesario usar UML como lenguaje de modelado ya que RUP utiliza el lenguaje unificado de modelado para preparar todos los esquemas de un sistema de software. UML es de hecho una parte fundamental de RUP, se desarrollaron de forma simultánea.

## **1.5. Herramienta para el modelado.**

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a

# Capítulo 1. *Fundamentación Teórica.*

---

aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como: el proceso de realizar un diseño del proyecto, cálculo de costes, implementación automática de parte del código con el diseño dado, compilación automática y documentación o detección de errores. (5)

## **1.5.1. Rational Rose.**

Rational Rose Enterprise Edition es una herramienta CASE desarrollada por Rational Corporation basada en el Lenguaje Unificado de Modelación (UML), que permite crear los diagramas que se van generando durante el proceso de Ingeniería en el Desarrollo del Software.

Rational Rose brinda muchas facilidades en la generación de la documentación del software que se esté desarrollando, además de que posee un gran número de estereotipos predefinidos que facilitan el proceso de modelación del software. Esta herramienta permite completar una gran parte de las disciplinas (flujos fundamentales) de RUP tales como captura de requisitos, análisis y diseño, implementación, control de cambios y gestión de configuración.

**Entre las características principales de Rational se pueden destacar:**

- ✓ Admite como notaciones: UML, OMT y Booch.
- ✓ Permite desarrollo multiusuario.
- ✓ Genera documentación del sistema.

## **1.5.2. Visual Paradigm.**

Visual Paradigm es una herramienta CASE que usa UML como lenguaje de modelado. La misma soporta, completamente el ciclo de vida del desarrollo de software: Análisis y Diseño Orientados a Objetos, Construcción, Pruebas y Despliegue. También permite la generación automática de reportes en formato pdf y html, el reconocimiento de artefactos de ingeniería a partir de reconocimiento de textos, implementa una actualización automática del modelo de diseño y código permitiendo mantener la documentación de ambos modelos actualizadas con los cambios que ocurran en ambos sentidos, optimizando la descripción textual de elementos de código a partir de la descripción visual. (13)

# Capítulo 1. *Fundamentación Teórica.*

---

## **Dentro de sus principales características están:**

- ✓ Disponibilidad en múltiples plataformas (Windows y Linux).
- ✓ Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- ✓ Licencia gratuita y comercial.
- ✓ Generación de base de datos. Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ✓ Ingeniería inversa de base de datos. Desde Sistemas Gestores de Base de Datos existentes a diagramas Entidad-Relación.

Es capaz de importar y exportar elementos de otras herramientas CASE como Rational Rose.

Visual Paradigm es la herramienta CASE que se utilizará en el modelado del sistema debido a que es multiplataforma, además de ser una potente herramienta CASE con licencia libre para uso de la comunidad de desarrolladores.

## **1.6. Lenguaje de programación.**

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que éste pueda comunicarse con los dispositivos hardware y software existentes. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo.(14)

### **1.8.1. PHP (“PHP Hypertext Pre-processor”).**

PHP es un lenguaje script para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML (Hyper Text Markup Language), debido a esto y a que es Open Source (código abierto), es el más popular y extendido en la web. (15)

#### **Entre sus principales características están:**

- ✓ Es un lenguaje Multiplataforma.
- ✓ Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ✓ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su

# Capítulo 1. *Fundamentación Teórica.*

---

resultado HTML al navegador. Esto hace a la programación usando este lenguaje segura y confiable.

- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

## **1.8.2. HTML.**

HTML es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento.

El lenguaje HTML es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo.

## **1.8.3. CSS (“Hojas de Estilo en Cascada”).**

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML (eXtensible HyperText Markup Language). Es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Describe cómo es que se va a mostrar el documento incluyéndole colores, imágenes y una serie de propiedades utilizadas para la confección del mismo.

## **1.8.4. JavaScript.**

Es un lenguaje interpretado, por lo que no es necesario compilar los programas para ejecutarlos. JavaScript nació con la necesidad de permitir a los autores de sitio web crear páginas que permitan intercambiar con los usuarios, ya que se necesitaba crear webs de mayor complejidad. Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto se puede cambiar totalmente el aspecto de nuestra página al gusto del usuario.

# Capítulo 1. *Fundamentación Teórica.*

---

## **1.9. Entorno de Desarrollo Integrado.**

Un Entorno Integrado de Desarrollo en inglés Integrated Development Environment (IDE) es un programa compuesto por un conjunto de herramientas para un desarrollador que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios.(16)

### **1.9.1. Eclipse.**

Fue desarrollado por IBM (International Business Machine) Canadá como el sucesor de sus herramientas para VisualAge. Actualmente es desarrollado por la Fundación Eclipse. Es un entorno de desarrollo integrado, de código abierto y multiplataforma. La base para él, son las aplicaciones de Cliente Enriquecido, opuesto a las aplicaciones de Cliente-Liviano basado en navegadores. Dispone de un editor de texto con resaltado de sintaxis, contiene una lista de tarea, la compilación es en tiempo real, tiene pruebas unitaria con JUNIT y asistentes (wizards) para creación de proyectos.

### **1.9.2. NetBeans.**

Fue fundado por Sun Microsystems en junio 2000 y sigue siendo su patrocinador, es un proyecto exitoso de código abierto con una gran base de usuarios. Es un entorno de desarrollo integrado y una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Es rápido y facilita desarrollar aplicaciones de escritorio, móviles y web con Java, HTML5, PHP, C / C + +, además es un producto libre y gratuito sin restricciones de uso.

Para el desarrollo de la aplicación se selecciona el NetBeans ya que este tiene un mejor completamiento de código, cuenta con una interfaz de usuario sencilla, tiene un soporte PHPUnit y se puede configurar en las propiedades del proyecto. Es muy fácil de usar para la programación web y orientada a objetos.

## **1.10. Servidor Web.**

La piedra angular de cualquier sitio o portal Web es, con toda seguridad, su servidor Web; el Software encargado de atender las peticiones de los clientes y ejecutar las

# Capítulo 1. *Fundamentación Teórica.*

---

páginas Web solicitadas. Existen multitud de aplicaciones para montar servidores Web, muchos de ellos distribuidos como Software libre y siendo, sin lugar a dudas, el más popular de todos: Apache. (13)

Para trabajar con lenguaje PHP es necesario una tecnología de servidor que sea capaz de interpretarlo, para ello se selecciona la tecnología de servidor Apache. El Servidor Apache HTTP (Hypertext Transfer Protocol) es un servidor Web desarrollado por la Apache Software Foundation (Fundación del Software Apache). (17)

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

- ✓ Multiplataforma.
- ✓ Es un servidor web conforme con el protocolo HTTP/2.0.
- ✓ Eficiente: Gran parte del esfuerzo se ha puesto en la optimización de código C de Apache para el rendimiento. Como resultado, se ejecuta más rápido y consume menos recursos que muchos otros servidores.
- ✓ Estabilidad/Fiabilidad: El código fuente de Apache está abierto al público. Cuando se encuentran errores, se comunican y se arreglan rápidamente. Se realizan actualizaciones y se anuncian a partir de entonces.

## **1.11. Marco de trabajo (Framework).**

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (18)

Los marcos de trabajo son un conjunto de conceptos, prácticas y criterios que sirven de base y facilitan el desarrollo de software. Para el desarrollo de aplicaciones web es buena práctica el uso de marcos de trabajo PHP que definen una filosofía de trabajo e implementan patrones de programación orientados a la Web.

### **1.11.1. CakePHP**

CakePHP es un marco de trabajo para el desarrollo de aplicaciones web siguiendo la arquitectura Modelo-Vista-Controlador. Requiere para su

# Capítulo 1. *Fundamentación Teórica.*

---

funcionamiento de PHP versión 4 o superior y un servidor de Base de Datos MySQL, aunque es capaz de trabajar con otros Sistemas Gestores de Base de Datos como PostgreSQL, SQLite o ADODB. Reduce significativamente los costos de desarrollo y acelera el proceso del mismo.

Es una estructura fundamental para ayudar a los programadores a crear aplicaciones web. Tiene como objetivo principal permitir el trabajo de forma estructurada y rápida, sin pérdida de flexibilidad. CakePHP pone a disposición de los desarrolladores todas las herramientas que se necesitan para empezar a programar lo que realmente hay que hacer: la lógica específica de la aplicación. (28)

## 1.11.2. **Symfony 2.**

Symfony2 es un completo marco de trabajo para el desarrollo de aplicaciones web basado en el patrón Modelo-Vista-Controlador. Está orientado hacia desarrolladores altamente preparados que están construyendo aplicaciones de grado corporativas y ha sido probado en numerosos proyectos reales. Requiere para su uso de PHP 5.3 o superior y es compatible con la mayoría de los Sistemas de Gestión de Base de Datos. Incorpora las mejores ideas del resto de los marcos de trabajo, tanto de los que están desarrollados en PHP como los que no.

Symfony2 es una colección de más de veinte bibliotecas independientes que se pueden utilizar dentro de cualquier proyecto PHP y contienen lo necesario para casi cualquier situación, independientemente de cómo se desarrolla el proyecto. Proporciona un potente conjunto de herramientas de uso opcional para el desarrollo rápido de aplicaciones web. Los usuarios normales pueden iniciar rápidamente el desarrollo mediante el uso de una distribución Symfony2, que proporciona un esqueleto de proyecto con parámetros por defecto. (29)

Se selecciona Symfony2 como marco de trabajo debido a que es rápido, flexible, cuenta con una abundante documentación y permite el desarrollo de aplicaciones mantenibles. Ha sido concebido para favorecer el rendimiento, es 3 veces más rápido que Symfony 1.4 y que Zend Framework 1.10. Su uso es altamente factible ya que proporciona las herramientas que en gran medida mejoran la productividad de los desarrolladores, además de considerarse estable y sostenible.

# Capítulo 1. *Fundamentación Teórica.*

---

## **1.12. Sistema Gestor de Base de Datos.**

Una Base de Datos (BD) es una colección de datos pertenecientes a un mismo entorno, organizados para su búsqueda y recuperación. Existen programas computarizados denominados Sistemas Gestores de Bases de Datos (SGBD) que permiten almacenar y posteriormente recuperar y actualizar estos datos de forma rápida y estructurada.

Un SGBD es un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar la misma con base en peticiones. La información en cuestión puede ser cualquier elemento que sea de importancia para el individuo u organización; en otras palabras, todo lo que sea necesario para auxiliarle en el proceso general de su administración. (30)

### **1.12.1. Microsoft SQL Server.**

Es una plataforma de BD que se utiliza en el procesamiento de transacciones a gran escala, almacenamiento de datos y en las aplicaciones de comercio electrónico. Es SGBD producido por Microsoft basado en el modelo relacional. Incluye un potente entorno gráfico, que permite el uso de Lenguaje de Definición de Datos (Data Definition Language (LDD)) que define las estructuras de la BD como crear, modificar, eliminar objetos como: tablas, vistas, índice, base de datos y permite el uso de Lenguaje de Manipulación de Datos (Lenguaje de Manipulación de Datos (LMD)) que posibilita la manipulación de los datos, la realización de consultas, actualización y borrado de tuplas de tablas. Permite trabajar en modo Cliente-Servidor donde la información se guarda en el servidor y el cliente accede a la misma. Administra otros servidores SQL permitiendo seguridad en tablas, dándole permisos de escritura, lectura y ejecución. Es fácil de usar, optimiza consultas y recuperación de datos. Su principal desventaja es la cantidad de memoria RAM que utiliza para la instalación y utilización del software. Además es propietario y solo puede ser utilizado en el sistema operativo Windows.

### **1.12.2. PostgreSQL.**

Es un sistema de gestor de datos relacional orientado a objetos, creado en la Universidad de California. Es de código abierto, posee un Control de Concurrencia Multi-Versión (MVCC) permitiendo trabajar con grandes cantidades de datos. Crea subconsultas, valores por defectos y ofrece funcionalidades como

# Capítulo 1. *Fundamentación Teórica.*

---

llaves primarias, identificadores, conversiones de tipo, entrada de enteros y hexadecimales. Está disponible para 34 plataformas con la última versión estable. Facilita la construcción de la aplicación en diferentes lenguajes como: ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG; PYTHON y RUBY y funciona en todos los sistemas operativos. Debido a la liberación de licencia se puede modificar, usar y distribuir de forma gratuita para cualquier fin.

El SGBD seleccionado es PostgreSQL debido a que es un sistema de alto rendimiento, flexible, multiplataforma, ofrece gran capacidad de almacenamiento, ahorros considerables de costos de operación y está diseñado para entornos con altos volúmenes de tráfico. Es el sistema libre más avanzado, con soporte de distintos tipos de datos y permite la creación de tipos de datos propios. Además es un sistema estable y confiable.

## **Conclusiones Parciales.**

Luego de realizado un estudio sobre las tecnologías y herramientas que cumplieran con las tendencias actuales del desarrollo web, orientado al código abierto y multiplataforma, se definió PHP como lenguaje de programación del lado del servidor y HTML, JavaScript y CSS del lado del cliente. Además, fueron seleccionados Symfony2 como framework de desarrollo, NetBeans como IDE, PostgreSQL como SGBD y Apache como servidor web. RUP será la metodología que guiará el proceso de desarrollo de software apoyada en UML como lenguaje de modelado y Visual Paradigm como herramienta CASE. Todas estas tecnologías escogidas con el fin de crear un sistema con la calidad requerida.

# Capítulo 2. Características y Diseño del Sistema.

## Capítulo 2. Características y Diseño del Sistema.

### 2.1. Introducción.

En este capítulo se describe la propuesta de solución del sistema según el problema descrito, además se especifica como el mismo será realizado a partir de las funcionalidades previstas. Para ello se presenta el diagrama de casos de uso del sistema con las correspondientes especificaciones, se realiza el diseño de la aplicación describiendo la arquitectura a seguir y los patrones de diseño. Se muestra igualmente los diagramas de clase, el esquema de la base de datos y los diagramas de interacción.

### 2.2. Descripción del Sistema.

El sistema que se propone a desarrollar permitirá la gestión de los medios básicos que controla el Departamento de Tecnología de la Facultad 5, de una forma práctica y segura. El sistema contará con una autenticación de usuarios así como una debida gestión de estos garantizando los permisos específicos para interactuar con las distintas funcionalidades.

### 2.3. Modelo del Dominio.

El modelo de dominio es una representación visual de los conceptos del mundo real significativos para un problema. El diagrama de conceptos del modelo de dominio del presente trabajo se muestra a continuación.

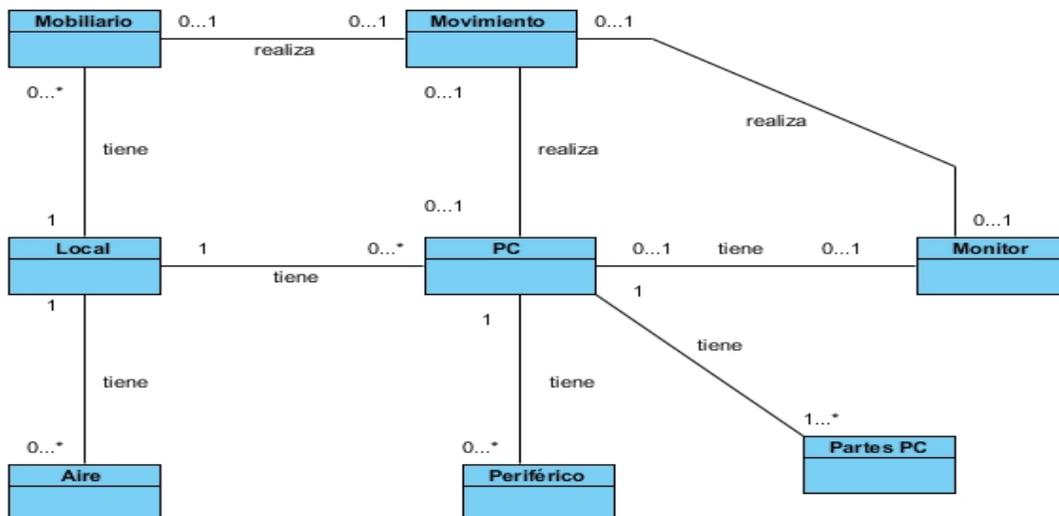


Figura 2.1 Modelo del dominio.

## Capítulo 2. Características y Diseño del Sistema.

---

Para una mejor comprensión del diagrama mostrado, se describe a continuación cada una de las clases que intervienen en el mismo:

- ✓ **Mobiliario:** Representa los medios básicos no tecnológicos controlados por el Departamento de Tecnología.
- ✓ **Local:** Es la entidad de la cual se controlarán los medios básicos.
- ✓ **Aire:** Representa los aires acondicionados de los locales.
- ✓ **PC:** Representa las computadoras controladas por el departamento.
- ✓ **Partes PC:** Representa las piezas internas de las computadoras.
- ✓ **Monitor:** Representa el dispositivo de salida asociado a las computadoras.
- ✓ **Periférico:** Representa las partes externas de la computadoras que constituyen útiles controlados por el departamento.
- ✓ **Movimiento:** Representa el modelo de Movimiento de Medios Básicos.

### 2.4. Requisitos Funcionales (RF).

Los requisitos funcionales de software son declaraciones de los servicios que el sistema debe proporcionar, cómo el sistema debe reaccionar a entradas particulares y cómo el sistema debe comportarse en determinadas situaciones. En algunos casos, los requisitos funcionales también pueden revelar lo que el sistema no debe hacer. (19)

El sistema a desarrollar a través de la realización del presente trabajo debe cumplir los siguientes requisitos:

**RF1-Autenticar Usuario.**

Esta funcionalidad permite al usuario acceder al sistema.

**RF2-Insertar Usuario.**

Esta funcionalidad permite insertar un usuario en la base de datos.

**RF3-Eliminar Usuario.**

Esta funcionalidad permite eliminar un usuario de la base de datos.

**RF4- Modificar Usuario.**

Esta funcionalidad permite modificar los atributos de un usuario en la base de datos.

**RF5-Insertar Aire.**

Esta funcionalidad permite insertar un aire en la base de datos.

**RF6-Eliminar Aire.**

## *Capítulo 2. Características y Diseño del Sistema.*

---

Esta funcionalidad permite eliminar un aire de la base de datos.

**RF7-** Modificar Aire.

Esta funcionalidad permite modificar los atributos de un aire en la base de datos.

**RF8-**Insertar Local.

Esta funcionalidad permite insertar un local en la base de datos.

**RF9-**Eliminar Local.

Esta funcionalidad permite eliminar un local de la base de datos.

**RF10-** Modificar Local.

Esta funcionalidad permite modificar los atributos de un local en la base de datos.

**RF11-**Insertar Mobiliario.

Esta funcionalidad permite insertar un mobiliario en la base de datos.

**RF12-**Eliminar Mobiliario.

Esta funcionalidad permite eliminar un mobiliario de la base de datos.

**RF13-** Modificar Mobiliario.

Esta funcionalidad permite modificar los atributos de un mobiliario en la base de datos.

**RF14-**Insertar Periféricos.

Esta funcionalidad permite insertar un periférico en la base de datos.

**RF15-**Eliminar Periféricos.

Esta funcionalidad permite eliminar un periférico de la base de datos.

**RF16-** Modificar Periféricos.

Esta funcionalidad permite modificar los atributos de un periférico en la base de datos.

**RF17-**Insertar Partes PC.

Esta funcionalidad permite insertar una parte de una pc en la base de datos.

**RF18-**Eliminar Partes PC.

Esta funcionalidad permite eliminar una parte de una pc de la base de datos.

**RF19-** Modificar Partes PC.

Esta funcionalidad permite modificar los atributos de una parte de una pc de la base de datos.

**RF20-**Insertar PC.

Esta funcionalidad permite insertar una PC en la base de datos.

**RF21-**Eliminar PC.

Esta funcionalidad permite eliminar una pc de la base de datos.

**RF22-** Modificar PC.

Esta funcionalidad permite modificar los atributos de una pc de la base de datos.

## *Capítulo 2. Características y Diseño del Sistema.*

---

### **RF23-Insertar Monitor.**

Esta funcionalidad permite insertar un monitor en la base de datos.

### **RF24-Eliminar Monitor.**

Esta funcionalidad permite eliminar un monitor de la base de datos.

### **RF25- Modificar Monitor.**

Esta funcionalidad permite modificar los atributos de un monitor de la base de datos.

### **RF26-Realizar Reporte.**

Esta funcionalidad permite verificar los medios básicos que se encuentran en préstamo o en reparación.

### **RF27-Realizar Movimiento.**

Esta funcionalidad permite generar el modelo de Movimiento de Medios Básicos.

## **2.5.Requisitos No Funcionales (RNF).**

Con el fin de ofrecer la calidad deseada para el usuario final, un sistema debe ofrecer una amplia variedad de atributos que no se describen específicamente por los requisitos funcionales. Estos constituyen restricciones sobre los servicios o funciones ofrecidas por el sistema. (20)

### ✓ **Usabilidad**

El sistema deberá poseer una interfaz web sencilla, lo más atractiva y clara posible para el usuario, además de poder ser usado por cualquier persona con conocimientos básicos en el manejo de la computadora y el entorno Web en sentido general.

### ✓ **Seguridad**

La autenticación será la primera acción del usuario en el sistema, proporcionándole los privilegios adecuados, de forma tal que la información sensible sea vista y manejada por la persona adecuada.

### ✓ **Rendimiento**

El sistema debe tener una respuesta rápida ante cualquier solicitud del usuario.

## Capítulo 2. Características y Diseño del Sistema.

### ✓ Portabilidad

El sistema estará desarrollado con tecnologías multiplataforma permitiendo que el servidor sea instalado tanto en Windows como en Linux, llevando a cabo esta operación sin necesidad de efectuar cambios significativos.

### ✓ Software

La máquina servidor deberá contar con el servidor web Apache 2.2 o superior, PostgreSQL 8.3 o superior como SGBD y PHP 5.3 o superior como lenguaje de programación.

El acceso al sistema se deberá realizar a través de navegadores web con soporte HTML5 como Mozilla Firefox 12 o superior, Google Chrome 18 o superior o Internet Explorer 9 o superior, Firefox recomendado.

### ✓ Hardware

El servidor de la aplicación deberá contar con un procesador Pentium IV a 2 GHz o superior, con una memoria RAM de 512 MB como mínimo.

### 2.6. Actores del Sistema.

Actores del sistema	Descripción
<b>Administrador</b>	Representa al encargado de asignar los permisos establecidos, así como de gestionar todos los usuarios.
<b>Responsable de Activos Fijos</b>	Representa al encargado del control de los activos fijos en el Departamento de Tecnología de la Facultad 5, responsable de la gestión y control de movimiento de los medios básicos en el departamento.
<b>Usuario</b>	Representa a los usuarios del sistema que se han de autenticar para acceder a las funcionalidades que le son permitidas.

*Tabla 2.1 Actores del Sistema.*

### 2.7. Diagrama de Caso de Uso del Sistema.

Es un modelo donde se muestra la relación entre los actores y los casos de uso del sistema.

# Capítulo 2. Características y Diseño del Sistema.

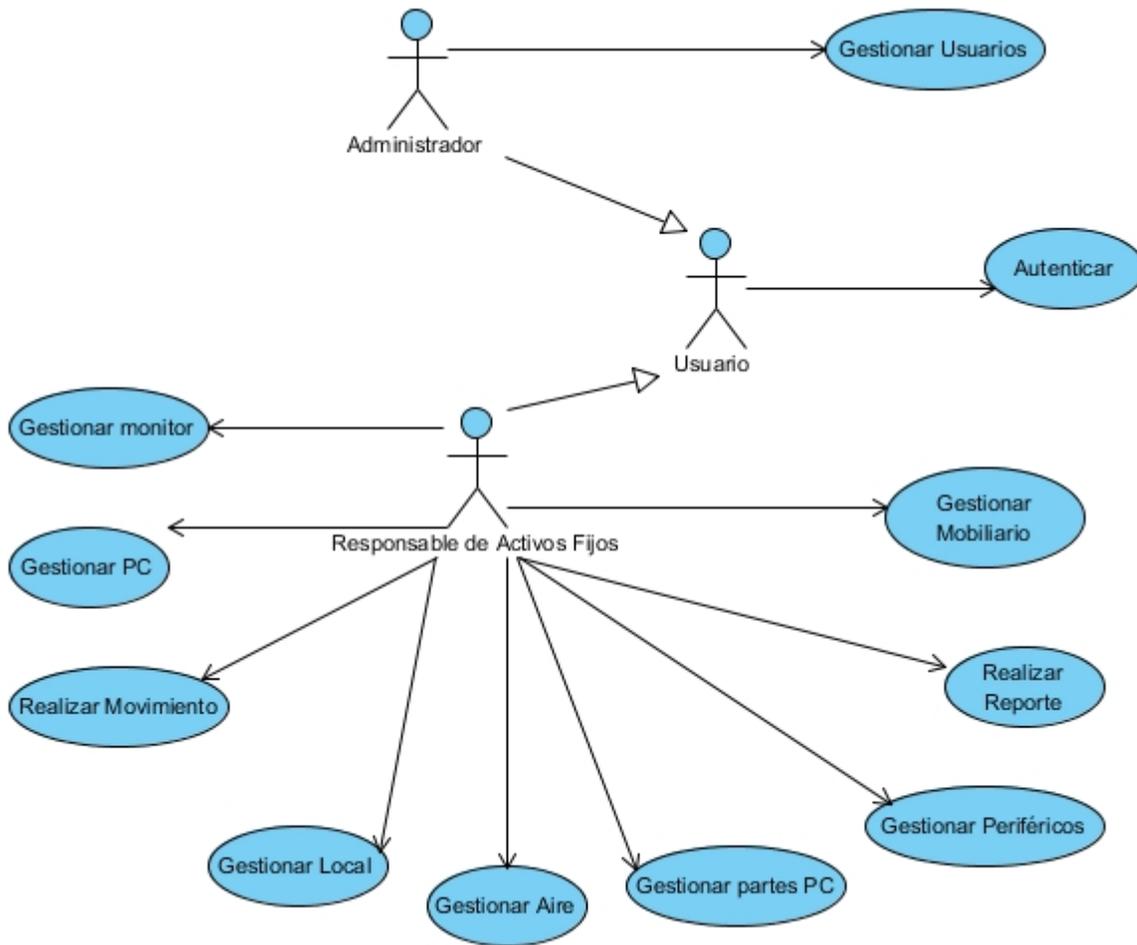


Figura 2.2 Diagrama de Caso de Uso del Sistema.

## 2.8. Descripción de Caso de Uso.

La descripción de los casos de usos son los pasos que sigue el actor para interactuar con el sistema. A continuación se muestra la descripción del Caso de Uso (CU) Gestionar Mobiliario y las descripciones correspondientes al resto de los CU se encuentran:([Anexo 1](#))

CU Gestionar Mobiliarios.

<b>Objetivo</b>	Modificar, Eliminar y Adicionar los mobiliarios.
<b>Actores</b>	Responsable de Activos Fijos (RAF).
<b>Resumen</b>	El caso de uso se inicia cuando el RAF hace alguna actualización, elimina o inserta algún mobiliario.
<b>Complejidad</b>	Media.

## Capítulo 2. Características y Diseño del Sistema.

<b>Prioridad</b>	Media.	
<b>Precondiciones</b>	Que se encuentre autenticado un usuario con rol RAF.	
<b>Poscondiciones</b>	Se modificó, eliminó o adicionó un Mobiliario.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>		<b>Sistema</b>
1.	1.1 Hace clic en la opción “Mobiliario”.	1.2 Muestra una ventana con las opciones “Modificar”, “Eliminar”, “Adicionar” y una lista de mobiliario. <ul style="list-style-type: none"> <li>• Si desea Añadir Mobiliario, va a la sección “Insertar Mobiliario”.</li> <li>• Si desea Eliminar Mobiliario, va a la sección “Eliminar”.</li> <li>• Si desea Modificar Mobiliario, va a la sección “Editar”.</li> </ul>
<b>Sección “Editar”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Selecciona el mobiliario que desea modificar.  1.2 Hace clic en la opción “editar”.	1.3 Muestra un formulario con los datos del mobiliario que solicitaste. <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Tipo.</li> <li>• Estado.</li> <li>• Inventario.</li> <li>• Ubicación.</li> <li>• Local.</li> </ul>
2	2.1 Introduce los datos.  2.2 Hace clic en la opción “Editar”.	2.3 Valida los datos (Alternativo 1).  2.4 Modifica los datos.

## Capítulo 2. Características y Diseño del Sistema.

		2.5 Muestra un mensaje “Se modificó correctamente el mobiliario”.
3		3.1 Termina caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje de alerta “Por favor, inserte correctamente los datos”.
2		2.1 Termina el caso de uso.
<b>Sección “Eliminar”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Selecciona el mobiliario que desea eliminar. 1.2 Hace clic en la opción “Eliminar”	1.3 Muestra un mensaje de confirmación: “Estás seguro que deseas eliminar”.
2	2.1 Presiona la opción “Aceptar”	2.2 Elimina el mobiliario. 2.3 Muestra un mensaje “Se eliminó correctamente el mobiliario”.
3		3.1 Termina el caso de uso.
<b>Sección “Insertar medio básico”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Presiona la opción “Insertar mobiliario”.	1.2 Muestra una ventana solicitando la siguiente información: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Inventario</li> <li>• Tipo</li> <li>• Estado</li> <li>• Ubicación</li> </ul>

## Capítulo 2. Características y Diseño del Sistema.

		<ul style="list-style-type: none"> <li>Local</li> </ul>
2	2.1 Introduce los datos.  2.2 Hace clic en la opción "Insertar".	2.3 Valida los datos (Alternativo 1).  2.4 Inserta los datos.  2.5 Muestra un mensaje "Se insertó correctamente el mobiliario".
3		3.1 Termina el caso de uso
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrecto.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico, muestra un mensaje de alerta "Por favor, inserte correctamente los datos".
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>RF11, RF12, RF13</b>	

**Tabla 2.2 Caso de Uso Gestionar Mobiliario.**

### 2.9. Patrones de Diseño.

Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. Representa un esquema o microarquitectura que supone una solución a problemas (dominios de aplicación) semejantes; una estructura común que tienen aplicaciones semejantes. (21)

Estos brindan una solución generalmente ya probada y documentada a problemas que se dan durante el proceso de desarrollo de software. Emplean un conjunto de buenas prácticas que facilitan el trabajo, definen una estructura de clases que da respuesta a uno o varios problemas en particular y presentan la ventaja de que son fáciles de comprender, además de que no dependen del lenguaje, haciéndolos genéricos. Lo complejo es cuando se tiene que decidir cuál usar, pues presentan diferentes

## Capítulo 2. Características y Diseño del Sistema.

---

soluciones, ya sea a través del empleo de uno u otro, o la combinación de varios. De ahí la importancia de conocer y estudiar los diferentes patrones que existen para poder determinar su uso.(22)

### 2.9.1. Patrón Data Mapper.

El patrón Data Mapper tiene como objetivo separar las estructuras de los objetos de las estructuras de los modelos relacionales y realizar la transferencia de datos entre ambos. Con el uso de un Data Mapper, los objetos pertenecientes al modelo, son ignorantes del esquema presente en la base de datos y, por supuesto, no necesitan hacer uso de código SQL.(23)

En Symfony2 la estructura de la base de datos es mapeada a clases del sistema mediante Doctrine2 que es un ORM (Object-Relational Mapping), mapeo objeto-relacional traducido al español y está implementado basado en Data Mapper, o sea que usando este patrón se interactúa con el contenido de la base de datos a través de las instancias de estas clases.

### 2.9.2. Patrones GRASP (*General Responsibility Assignment Software Patterns*).

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP), describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.(24)

- ✓ **Experto:** Asignar una responsabilidad al más competente en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. (24)

Este patrón está presente ya que Doctrine2 es la librería usada para realizar la abstracción al modelo de datos, generando las clases con funcionalidades propias de las entidades. Por tanto, cada clase creada por esta librería a partir de una entidad es experta en manejar su información. Ejemplos de estas clases son: *Local.php*, *Mobiliario.php* y *PC.php*.

- ✓ **Bajo acoplamiento:** Asignar una responsabilidad para mantener bajo acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. (24)

## Capítulo 2. Características y Diseño del Sistema.

---

Las clases pertenecientes a la vista, las del modelo y las controladoras son parcialmente independientes unas de otras, o sea, no existe una relación directa entre estas clases debido a la separación que hace el patrón arquitectónico. En la relación existente entre las clases *Local.php* perteneciente al modelo, *LocalController.php* perteneciente al componente controlador y *new.html.twig* perteneciente a la vista se evidencia el uso de este patrón.

- ✓ **Controlador:** Asignar la responsabilidad del manejo del flujo de los eventos del sistema a una clase. Garantiza que los procesos sean manejados por la capa Controladora y no por la de Presentación.(24)

Las peticiones web son manejadas por un solo controlador (app.php), que es el punto de entrada único de la aplicación. Cuando este controlador recibe una petición, asocia el nombre de una acción con la URL entrada por el usuario.

### 2.10. Patrones de Arquitectura.

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño que ofrecen soluciones a problemas de arquitectura en la ingeniería de software. Son una vista del sistema que contiene los componentes principales del mismo y la forma en que estos interactúan para alcanzar el objetivo del sistema. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor.

#### 2.10.1. Patrón Modelo Vista Controlador (MVC).

La arquitectura MVC separa la lógica de negocio y la presentación, por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. (25)

El principio más importante de la arquitectura MVC es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. (25)

## Capítulo 2. Características y Diseño del Sistema.

### ✓ **Modelo.**

Es el componente que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.(25)

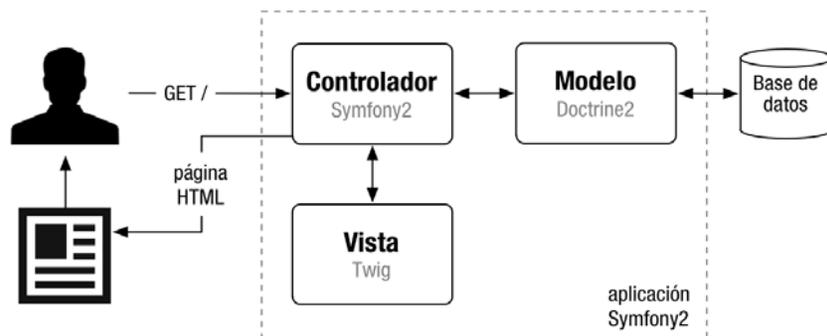
### ✓ **La Vista.**

Es el componente que maneja la presentación visual de los datos contenidos por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo. (25)

### ✓ **El Controlador.**

Es el componente que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. (25)

Symfony2 basa su funcionamiento interno en la arquitectura Modelo-Vista-Controlador por tanto será la arquitectura a usar en el desarrollo del sistema.



**Figura 2.3** Arquitectura de Symfony2.

# Capítulo 2. Características y Diseño del Sistema.

## 2.11. Diseño.

### 2.11.1. Diagrama de colaboración.

Describe lo que hacen las clases de una aplicación mediante mensajes recibos o enviados por estas. En la figura 2.4 se muestra el diagrama de colaboración del CU Gestionar Mobiliario.

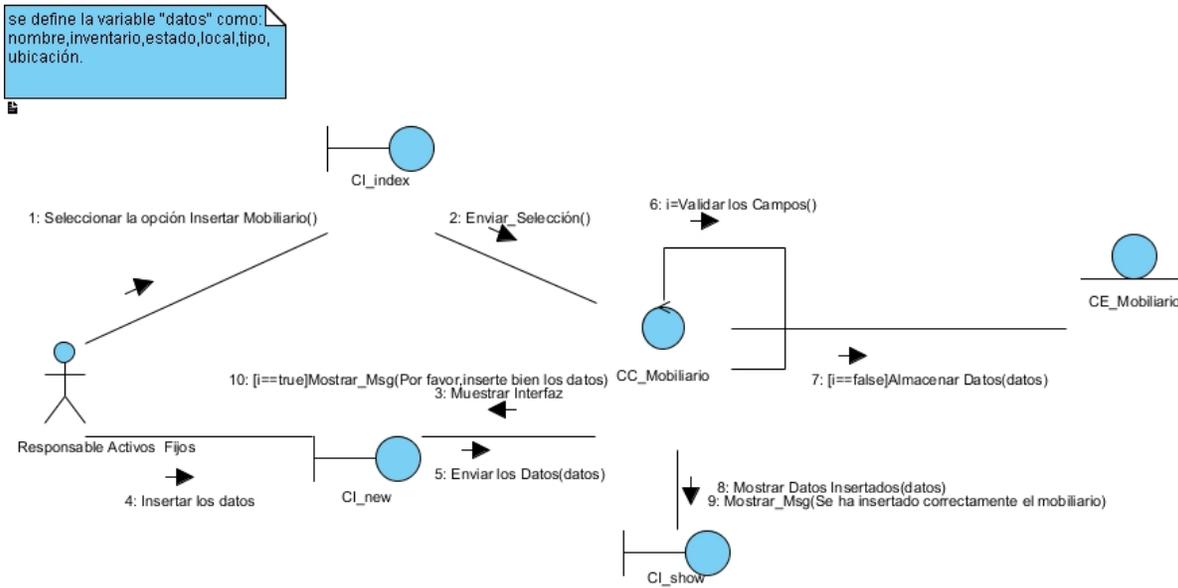


Figura 2.4 Diagrama de colaboración del CU Gestionar Mobiliario (Insertar Mobiliario).

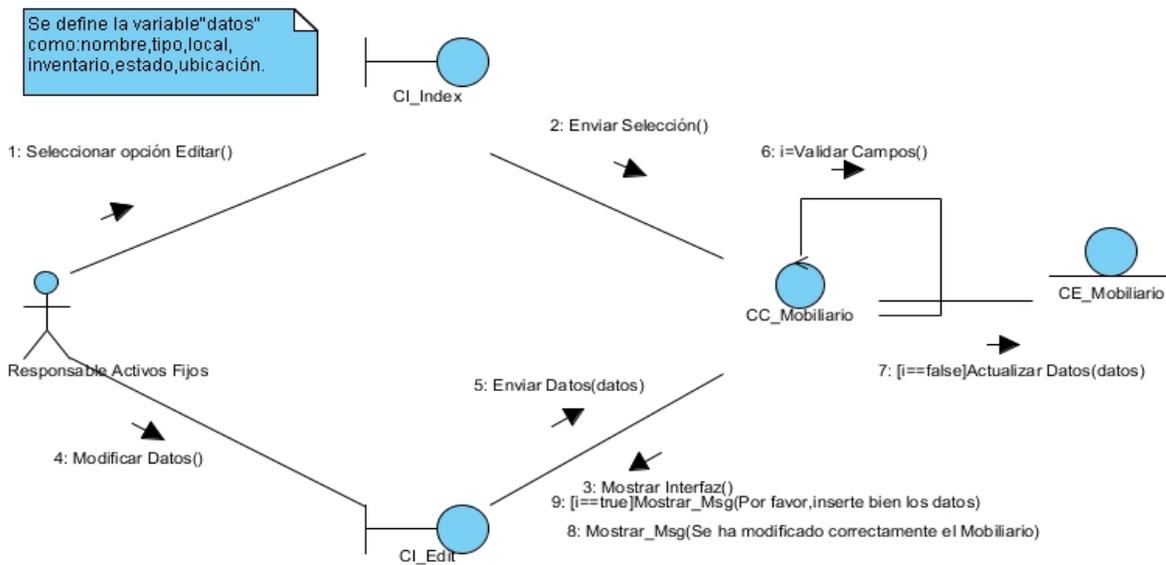


Figura 2.5 Diagrama de colaboración del CU Gestionar Mobiliario (Modificar Mobiliario).

# Capítulo 2. Características y Diseño del Sistema.

se define la variable "datos" como:  
nombre, inventario, tipo, estado, local,  
ubicación.



Figura 2.6 Diagrama de colaboración del CU Gestionar Mobiliario (Eliminar Mobiliario).

## 2.11.2. Diagrama de clase del diseño.

Describe gráficamente las interfaces de una aplicación y se elabora para tener datos concretos de la implementación del sistema. A continuación se muestra el diagrama de clase del diseño del CU Gestionar Mobiliario y los diagramas correspondientes ver: [Anexo 2](#)

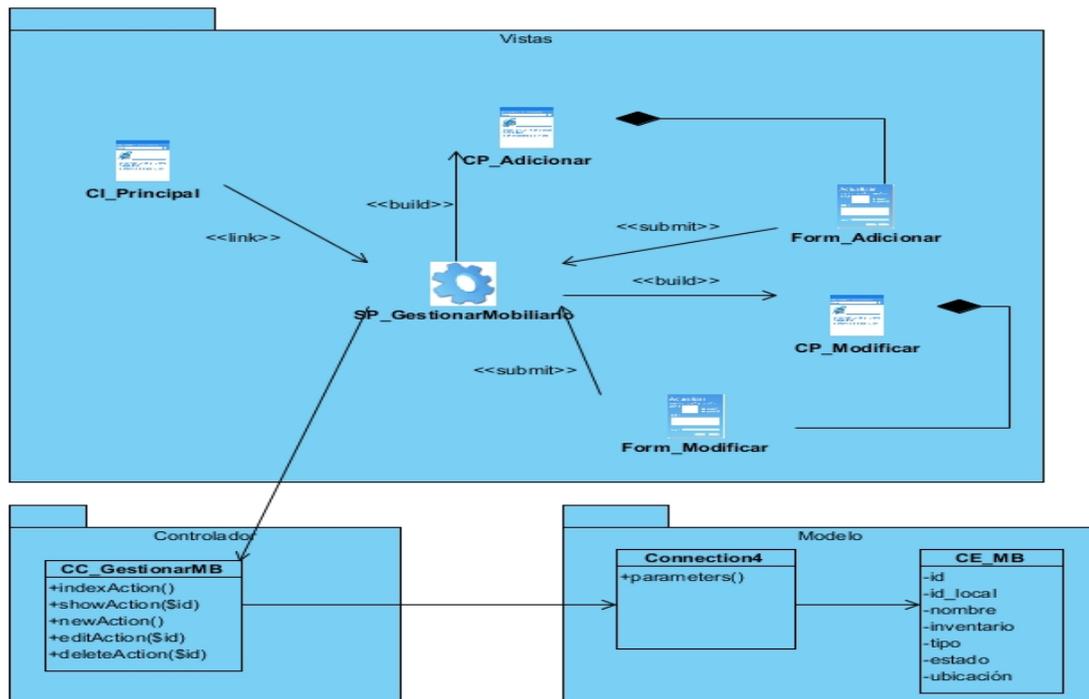


Figura 2.7 Diagrama de clase del diseño del Caso de Uso Gestionar Mobiliario.

# Capítulo 2. Características y Diseño del Sistema.

## 2.11.3. Diagrama de Clases Persistentes.

Las clases persistentes son clases en una aplicación que implementan las entidades del problema, el nombre de estas representan las tablas y los atributos las columnas. Se muestra el diagrama en la figura 2.8.

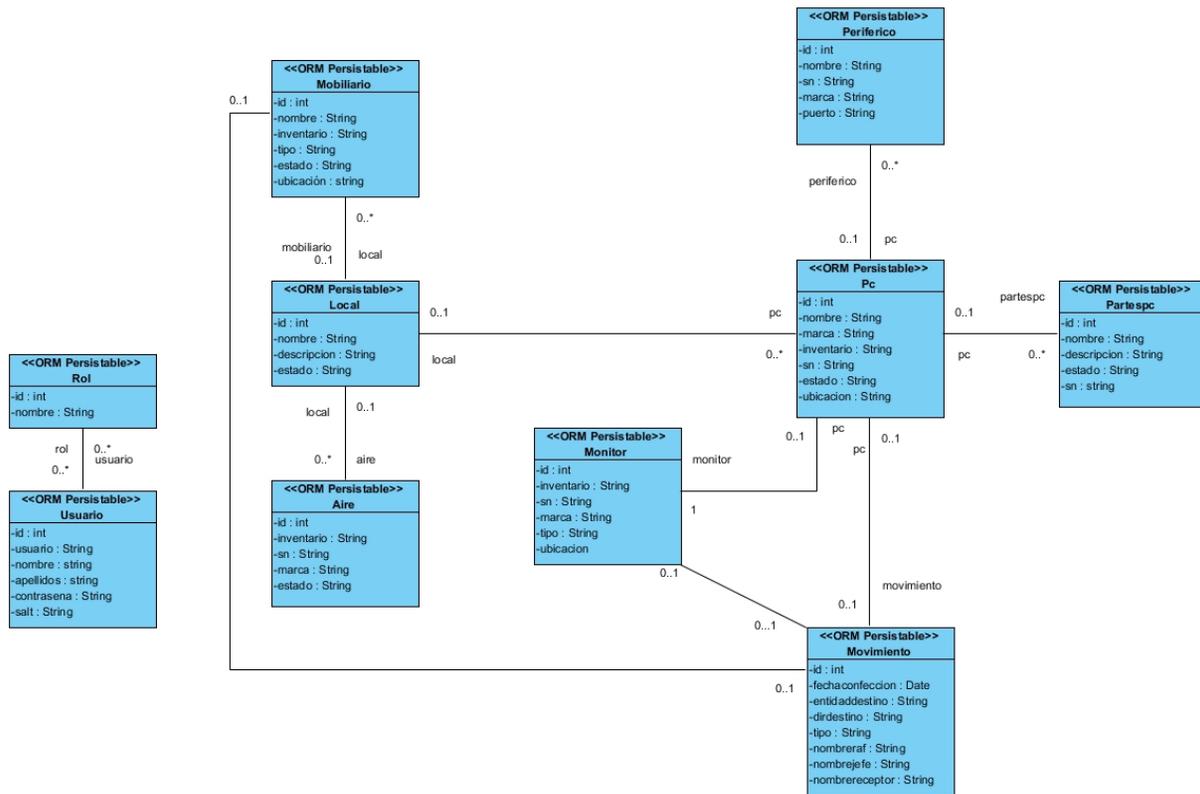


Figura 2.8 Diagrama de clases persistentes.

## 2.11.4. Modelo de Datos.

Este ayuda a representar las entidades que el sistema debe entender, los atributos que describen cada entidad y las relaciones entre ellas. Los Modelos de Datos Conceptuales están orientados a la descripción de las estructuras de datos y restricciones de integridad. Se usan fundamentalmente durante la etapa de análisis de un problema dado y están orientados a representar los elementos que intervienen en ese problema y sus relaciones. El ejemplo más típico es el Modelo Entidad-Relación.

# Capítulo 2. Características y Diseño del Sistema.

## 2.11.5. Diagrama Entidad Relación (DER).

El DER hace un mejor entendimiento del dominio de la información del problema y funciona como base para producir el diseño de la base de datos. En la figura 2.8 se representa el DER.

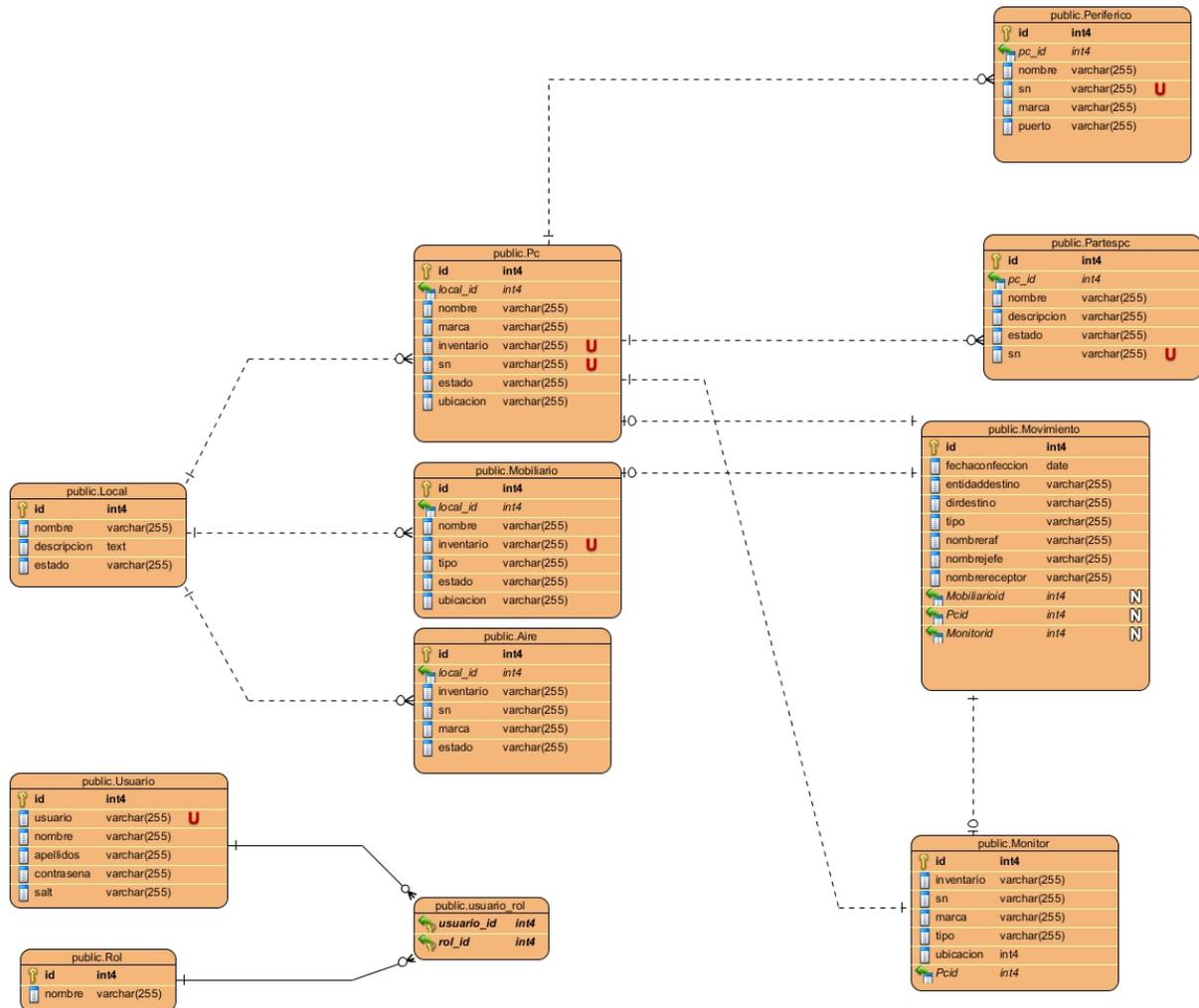


Figura 2.9 Diagrama Entidad Relación.

### Conclusiones Parciales.

En el presente capítulo se definieron los requerimientos necesarios para el desarrollo de la aplicación, además de describir cada uno de los casos de uso, basado en los requisitos funcionales. Tomando en cuenta esta descripción se obtuvieron los diagramas correspondientes al diseño del sistema, apoyado de los patrones analizados. Una vez descrito como será desarrollada la herramienta quedan creadas las condiciones para la implementación del mismo.

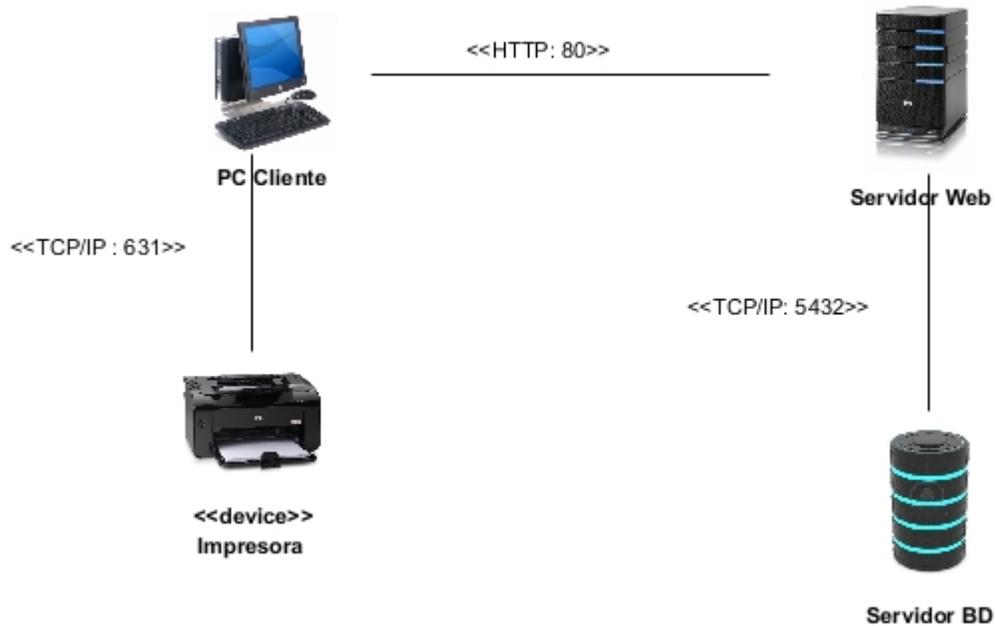
## Capítulo 3. Implementación y Prueba.

### 3.1.Introducción.

En este capítulo se describe el flujo de trabajo de implementación, donde se define la organización del código y como se implementan las clases y objetos en términos de componentes. Una vez realizado el software se realizan las pruebas para verificar las funcionalidades de este, validando así los requisitos propuestos.

### 3.2.Diagrama de Despliegue.

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.



**Figura 3.1 Diagrama de Despliegue.**

### 3.3.Diagrama de Componente.

Dentro del modelo de implementación se encuentran los diagramas de componentes, este muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables. Prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Se relacionan con los diagramas de clases, ya que un componente normalmente se corresponde con una o más clases, interfaces o

# Capítulo 3. Implementación y Prueba.

colaboraciones pero un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clase. (26)

En la figura 3.2 se muestra el diagrama de componente para el CU Gestionar Mobiliario y para los otros diagramas ver: ([Anexo 3](#))

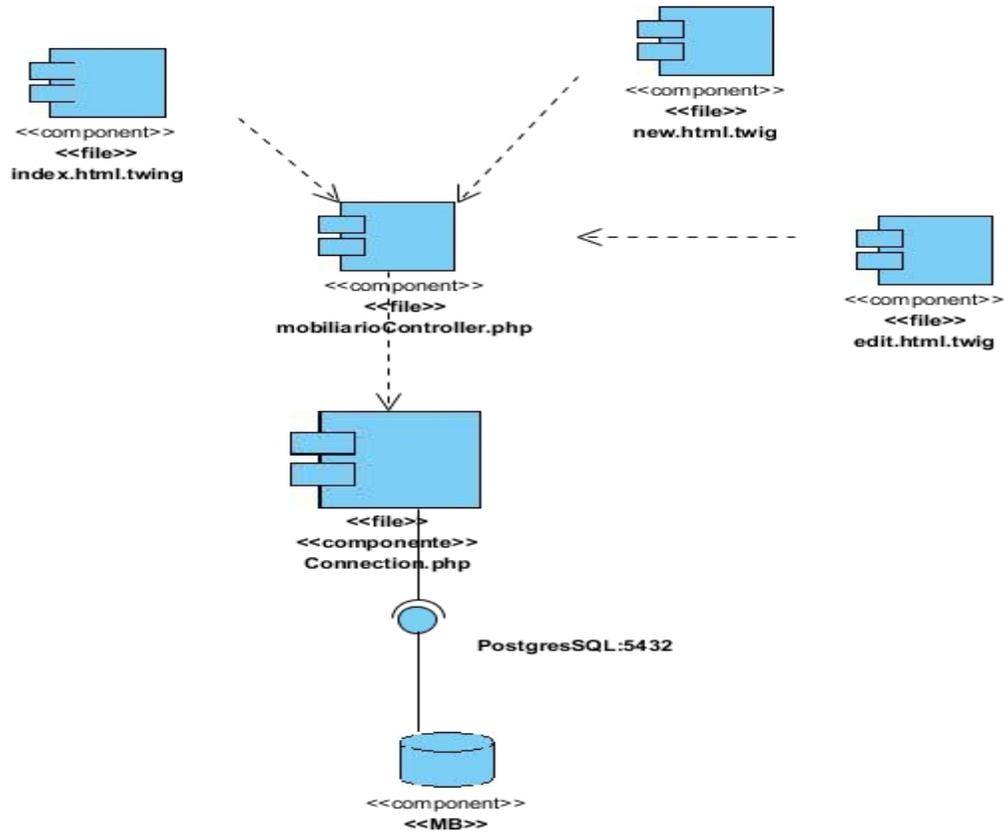


Figura 3.2 Diagrama de componentes del CU Gestionar Mobiliario.

## 3.4. Código Fuente.

También llamado código base, es un texto que se ha escrito en un lenguaje de programación concreto destinado a ser legible por humanos para que cualquiera lo analice, lo modifique, lo comparta o lo copie.

### Estándares de codificación.

Son reglas que se siguen para la escritura del código fuente, comprende todos los aspectos de la generación de un código.

#### 3.4.1. Nomenclatura.

- ✓ Utiliza espacios de nombres (namespace) para todas las clases. Ejemplo:

## Capítulo 3. Implementación y Prueba.

---

```
<?php  
  
namespace DepartamentoT\MBBundle\Entity;
```

**Figura 3.3** Ejemplo de la utilización de namespace en las clases.

- ✓ Usa mayúsculas intercaladas, no subrayados, para variables, funciones y nombres de métodos. Ejemplo:

```
public function indexAction()  
{  
    $em = $this->getDoctrine()->getEntityManager();  
  
    $entities = $em->getRepository('MBBundle:Local')->findAll();  
  
    return $this->render('MBBundle:Local:index.html.twig', array(  
        'entities' => $entities  
    ));  
}
```

**Figura 3.4** Ejemplo de funciones y métodos que usan mayúscula intercalada.

### 3.4.2. Estructura.

- ✓ Añade un solo espacio después de cada delimitador coma. Ejemplo:

```
return $this->render('MBBundle:Local:index.html.twig', array(  
    'entities' => $entities  
));
```

**Figura 3.5** Ejemplo añadir un espacio después de una coma.

- ✓ Añade un solo espacio alrededor de los operadores (=, &&,...). Ejemplo:

```
$em = $this->getDoctrine()->getEntityManager();
```

**Figura 3.6** Ejemplo añadir un espacio después del operador (=).

### 3.5. Pruebas de Software.

Luego de concluido un proyecto de desarrollo de software es necesario verificar la calidad del mismo antes de realizar su entrega. En el flujo de trabajo de pruebas es donde se examina el resultado de la implementación. Estas no son más que un conjunto de técnicas y métodos que garantizan el buen desempeño de una aplicación, para la validación del producto y representan un elemento crítico en la garantía de la calidad del producto.

## Capítulo 3. Implementación y Prueba.

---

Las pruebas regulan la ejecución de los proyectos y garantizan la calidad del software desarrollado. Desde el modelo de desarrollo en cascada hasta la aparición de las metodologías ágiles, las pruebas han pasado de ser una simple etapa en el proceso de desarrollo a constituirse en un conjunto de etapas que controlan la duración del ciclo de vida, la calidad y la confiabilidad del software desarrollado. (27)

### Niveles de Prueba.

- ✓ Pruebas unitarias.
- ✓ Pruebas de componentes.
- ✓ Pruebas de integración.
- ✓ Pruebas de sistema.
- ✓ Pruebas de aceptación.

### Métodos de Prueba.

- ✓ **Caja Blanca:** Se realiza un análisis riguroso de los detalles procedimentales, comprobando las rutas lógicas del programa. El objetivo principal de este método es probar que todos los caminos del código están correctos.
- ✓ **Caja Negra:** Tipo de prueba diseñada para demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una respuesta correcta.

### 3.6. Pruebas Realizada.

Para validar la solución propuesta se realizaron pruebas funcionales, las cuales permiten probar el correcto funcionamiento de los requisitos funcionales, lo que incluye la entrada de datos, el procesamiento de estos y la obtención de los resultados.

Para la creación de casos de prueba de sistema se usa el método de caja negra, procedimiento que permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata los errores.

### 3.7. Diseño de Caso de Prueba.

A continuación se describe una de las pruebas realizada al Caso de Uso Gestionar Mobiliario, especificando la información de entrada, los resultados esperados y obtenidos una vez realizado el Caso de Prueba y a los otros Casos de Prueba ver:

([Anexo 4](#))

## Capítulo 3. Implementación y Prueba.

### Caso de Prueba Gestionar Mobiliario.

✓ **Breve Descripción.**

Este caso de prueba permite comprobar las funcionalidades de modificar, eliminar e insertar al sistema.

✓ **Flujo Central.**

Seleccionar la opción Mobiliario.

✓ **Condiciones de Ejecución.**

El Responsable de activos fijos debe haberse autenticado.

### EC1.Insertar Mobiliario.

#### Flujo Central.

Seleccionar la opción Insertar Mobiliario.

Introducir los datos necesarios.

#### Iteraciones.

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducir los datos correcto y presiona el botón "Insertar".		El sistema debe mostrar una interfaz con los datos entrados, luego de haber insertado en la base de datos. Mostrar un mensaje "se insertó correctamente el mobiliario".	Muestra una interfaz con los datos pasados por parámetro, luego de ser insertado en la base de datos. Muestra un mensaje "se insertó correctamente el mobiliario".
	Pulsar "Insertar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	El sistema muestra el mensaje.
	Pulsar "Insertar" y haber entrado datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	El sistema permite que entre cualquier dato.

## *Capítulo 3. Implementación y Prueba.*

	Pulsar “Insertar” y haber entrado un número de inventario ya creado.	El sistema debe mostrar un mensaje, ya existe ese mobiliario.	El sistema permite que se inserten más de un mobiliario con el mismo número de inventario.
--	--	---	--

**Tabla 3.1 Caso de Prueba “CU Gestionar Mobiliario (Insertar)”.**

### **SC2.Editar Mobiliario.**

- ✓ **Flujo Central.**  
 Seleccionar la opción Editar.  
 Modificar los datos deseados.
- ✓ **Condiciones de ejecución.**  
 Exista un mobiliario creado.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Modificar los datos correctamente y presiona el botón “Editar”.		El sistema debe mostrar una interfaz con todos los datos modificados, luego de ser modificado en la base de datos.  Mostrar un mensaje “se modificó correctamente el mobiliario”.	Muestra una interfaz con los datos modificados, luego de ser modificado en la base de datos.  Muestra un mensaje “se modificó correctamente el mobiliario”.
	Pulsar “Editar” sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar “Editar” y haber entrado los datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	El sistema permite que entre cualquier dato.

## *Capítulo 3. Implementación y Prueba.*

	Pulsar “Editar” y haber entrado un número de inventario ya creado.	El sistema debe de mostrar un mensaje, ya existe ese mobiliario creado.	El sistema permite tener mobiliario con el mismo número de inventario.
--	--	---	--

**Tabla 3.2 Caso de Prueba “CU Gestionar Mobiliario (Modificar)”.**

### **SC3.Eliminar Mobiliario.**

- ✓ **Flujo Central.**  
Selecciona la opción Eliminar.
- ✓ **Condiciones de Ejecución.**  
Que exista un mobiliario creado.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Presionar el botón “Eliminar”.		Eliminar los datos.	Elimina los datos.

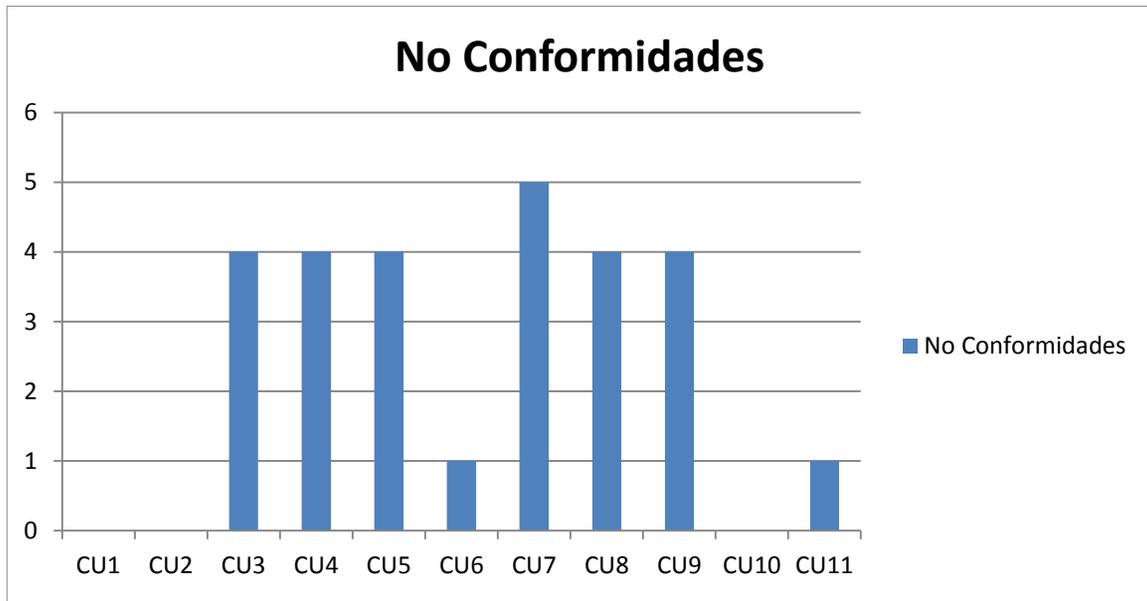
**Tabla 3.3 Caso de Prueba “CU Gestionar Mobiliario (Eliminar)”.**

### **3.8.Resultado de las Pruebas.**

Se realizaron las pruebas establecidas de acuerdo a los 27 diseños de caso de pruebas definido, donde se detectaron no conformidades quedando estas distribuidas de la siguiente forma:

- ✓ Existen mobiliarios con el mismo número de inventario.
- ✓ Se eliminan todos los medios básicos asociados al local cuando se elimina este.
- ✓ Se eliminan monitor y periféricos asociados a la PC cuando se elimina esta.
- ✓ Existen PC con más de un mismo periférico.
- ✓ Los campos admiten cualquier carácter extraño.
- ✓ Existen números de serie duplicados.
- ✓ No se actualiza en la base de datos cuando se haga un movimiento.

## Capítulo 3. Implementación y Prueba.



**Figura 3.7 Gráfico de la cantidad de no conformidades por caso de uso.**

Los casos de uso que se muestran en el gráfico siguen el orden este:

- ✓ CU Autenticar.
- ✓ CU Gestionar Usuario.
- ✓ CU Gestionar Mobiliario.
- ✓ CU Gestionar Periféricos.
- ✓ CU Gestionar Aire.
- ✓ CU Gestionar Local.
- ✓ CU Gestionar PC.
- ✓ CU Gestionar Partes PC.
- ✓ CU Gestionar Monitor.
- ✓ Realizar Reporte.
- ✓ Realizar Movimiento.

Estas pruebas no solo fueron realizadas para detectar errores, sino para corregirlos. Se realizaron 3 iteraciones para corregir los errores detectados en el sistema.

### **Conclusiones Parciales.**

Al finalizar el desarrollo de este capítulo, en el cual se detalla la fase de implementación se consigue un sistema que cumple con los requerimientos definidos. Durante la realización de las pruebas de caja negra fueron resueltas las no conformidades detectadas.

### **Conclusiones.**

El cumplimiento de los objetivos trazados para el presente trabajo de diploma permite arribar a las siguientes conclusiones:

- ✓ El análisis integral de los elementos referidos a la gestión de medios básicos, la metodología y tecnologías seleccionadas, constituyen el punto de partida hacia la aplicación web ofrecida que optimiza el proceso.
- ✓ El sistema, implementado haciendo uso de tecnologías distribuidas bajo licencia de software libre, está en correspondencia con la independencia tecnológica que se quiere alcanzar en nuestro país.
- ✓ La aplicación de pruebas funcionales usando el método de caja negra permitió obtener un sistema con calidad para gestionar los medios básicos del Departamento de Tecnología de la Facultad 5.

### **Recomendaciones.**

Luego de realizado el presente trabajo y teniendo en cuenta los resultados obtenidos, se recomienda:

- ✓ Integrar todos los procesos de gestión del Departamento de Tecnología de la Facultad 5.
- ✓ Se haga extensivo el uso de la aplicación a los centros productivos, oficinas y departamentos docentes.

## **Glosario de términos**

**JUnit:** Es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

**Release:** Nueva versión de una aplicación informática.

**Sun Microsystems:** Empresa informática que se dedicaba a vender estaciones de trabajo, servidores, componentes informáticos, software (sistemas operativos) y servicios informáticos. Fue adquirida en el año 2009 por Oracle Corporation, fabricante de semiconductores y software.

**URL:** Un localizador de recursos uniforme, más comúnmente denominado URL (sigla en inglés de uniform resource locator), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación

**VisualAge:** Nombre de una familia de entornos de desarrollo integrados de computadoras de IBM, que incluye soporte para múltiples lenguajes de programación.

# Referencias Bibliográficas.

---

## Referencias Bibliográficas.

1. **Jorge Mario Cortes Cortes.** scribd. [En línea] Universidad Nacional de Colombia, 2008.  
<http://es.scribd.com/doc/69594036/GLPI..>
2. **Sosa, Lianet Arias.** Contar con informática. *Diario Granma.* jueves, 2010, 49.
3. **Cecilia Placerez, Niurka Armas, Antonio Lima, Omar Ernesto Rodriguez.** citmatel. *tecnologia de la informacion de servicios telematicos.* [En línea] [Citado el: 27 de febrero de 2013.]  
[www.rodosxxi.cu/rodosxxi.php](http://www.rodosxxi.cu/rodosxxi.php).
4. **Barzanallana, Rafael Menéndez.** *Apuntes Informática Aplicada a la Gestión Pública.* 2010.
5. **Araujo, Yuriana, Hilda López, Alexander Mendoza, Luis Torrealba, German Ortiz, Roberto Guerra.** *Metodología RUP.* 2010.
6. **Edition, Philippe Kruchten Publisher: Addison Wesley Second.** *The Rational Unified Process An Introduction, Second Edition.* March 14, 2000.
7. **Martínez, Alejandro Martínez y Raúl.** *Guía a Rational Unified Process.* 2011.
8. **Sanchez, María A. Mendoza.** *Metodologías De Desarrollo De Software.* Junio 7 del 2004.
9. **Williams, Rob.** 2007. *Real time Systems Development.*
10. **Myer, Thomas.** *Professional CodeIgniter (Wrox Professional Guides).* 25 Julio 2008.
11. **Ezust, Alan Ezust And Paul.** *Introduction to Design Patterns in C++ with Qt (2nd Edition).* 2012.
12. **Erika Camacho, Fabio Cardeso, Gabriel Nuñez.** *Guía Arquitectura v2.* 2004 .
13. **Ben Laurie, Peter Laurie.** *Apache: The Definitive Guide (3rd Edition).* 2002.
14. **Lenguajes de Programación.** [En línea] 2009. [Citado el: 27 de febrero de 2013.]  
<http://www.lenguajes-de-programacion.com..>
15. **Andi Gutmans, Stig Sæther Bakken.** *PHP 5 Power Programming.* 2004.
16. **Myatt, Adam.** *Pro NetBeans 5.5 IDE Enterprise Edition.* 2007.
17. **Jason Gerner, Morgan Owens, Elizabeth Naramore, Matt Warden.** *Professional LAMP: Linux, Apache, MySQL and PHP5 Web Development.* 2005.
18. **EcuRed.** [En línea] 14 de diciembre de 2010. [Citado el: 8 de marzo de 2013.]  
<http://www.ecured.cu/index.php/Framework>.

## Referencias Bibliográficas.

---

19. **Sommerville, Ian.***Ingeniería de Software.*Septima Edición. 2005.
20. **March, Addison Wesley.***The Rational Unified Process An Introduction* 2da Edición. 14, 2000.
21. **Kuchana, Partha.***Software architecture design patterns in Java.* abril de 2004.
22. **Allan Shalloway, James R Trott.***Design Patterns Explained.* 2010.
23. **Cesar de la Torre Llorente, Unai Zorrilla Castro.***Guía de Arquitectura N-Capas orientadas al Dominio con .NET 4.0.* 2010.
24. **Larman, Craig.***UML y Patrones.* 2006.
25. **Pantoja, Ernesto Bascón.***El patrón de diseño Modelo-Vista-Controlador (MVC).* 2008.
26. **Ramírez, Elisa Arizaca.***Diagrama de componentes.* 2009.
27. **Gèlvez, Hugo Alexander Parada.***Contribucion a la Gestion de los Procesos de Pruebas de Software y Servicios.* 2010 .
28. **Foundation, Cake Software.***CakePHP Cookbook.* 2013.
29. **Boyarynov, Eugene.***Symfony 2: The Book.* 2011.
30. **Date, C J.***Introducion a los sistemas de Bases de Datos.* 2009.

## Anexos.

### Anexo1.Descripcion de caso de uso.

CU Gestionar usuario.

<b>Objetivo</b>	Modificar, Eliminar y Adicionar usuarios.	
<b>Actores</b>	Administrador.	
<b>Resumen</b>	El caso de uso se inicia cuando el administrador crea, elimina o modifica un usuario.	
<b>Complejidad</b>	Baja.	
<b>Prioridad</b>	Media.	
<b>Precondiciones</b>	Que exista al menos un usuario administrador autenticado.	
<b>Poscondiciones</b>	Se modificó, eliminó o adicionó un usuario.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>		<b>Sistema</b>
1.	1.1 Hace clic en la opción "Administrador".	1.2 Muestra una ventana con las siguientes opciones "Modificar", "Eliminar", "Adicionar" y el listado de los usuarios. <ul style="list-style-type: none"> <li>• Si desea Modificar usuario, va a la sección "editar".</li> <li>• Si desea Eliminar usuario, va a la sección "Eliminar".</li> <li>• Si desea Adicionar usuario, va a la sección "Insertar Usuario".</li> </ul>
<b>Sección "editar".</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Selecciona el usuario que desea modificar. 1.2 Presiona opción "editar".	1.3 Se muestra un formulario con los datos de ese usuario. <ul style="list-style-type: none"> <li>• Usuario.</li> </ul>

		<ul style="list-style-type: none"> <li>• Rol.</li> <li>• Contraseña.</li> <li>• Nombre.</li> <li>• Apellidos.</li> </ul>
2	<p>2.1 Inserta los datos.</p> <p>2.2 Hace clic en el botón "Editar"</p>	<p>2.3 Valida los datos del usuario (alternativo 1).</p> <p>2.4 Modifica los datos del usuario.</p> <p>2.5 Muestra un mensaje "Se modificó correctamente el usuario".</p>
		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Datos incorrecto.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico, muestra un mensaje de alerta "Por favor, inserte correctamente los datos".
2		2.1 Termina el caso de uso.
<b>Sección "Eliminar".</b>		
<b>Actor</b>		<b>Sistema</b>
1	<p>1.1 Selecciona el usuario que desea eliminar.</p> <p>1.2 Presiona opción "Eliminar".</p>	1.3 Muestra un mensaje de confirmación "Esta seguro que desea eliminar"
2	2.1 Presiona "Aceptar"	2.2 Elimina el usuario.
3		3.1 Termina el caso de uso.
<b>Sección "Insertar Usuario".</b>		
<b>Actor</b>		<b>Sistema</b>

1	1.1 Presiona opción "Insertar Usuario "	1.2 Muestra una ventana solicitando la siguiente información: <ul style="list-style-type: none"> <li>• Usuario (Obligatorio)</li> <li>• Rol (Obligatorio)</li> <li>• Contraseña (Obligatorio)</li> <li>• Nombre.</li> <li>• Apellidos.</li> </ul>
2	2.1 Introducir datos del usuario.  2.2 Hace clic en la opción "Insertar".	2.3 Verifica los datos (Alternativo 1).  2.4 Adiciona usuario.  2.5 Muestra un mensaje "Se insertó correctamente el usuario".
3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Existe error en los datos.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje de alerta "Por favor, inserte bien los datos".
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>RF2, RF3, RF4</b>	

CU Autenticar.

<b>Objetivo</b>	El usuario puede acceder a la aplicación.
<b>Actores</b>	Administrador y FAR

<b>Resumen</b>	El caso de uso se inicia cuando el usuario se autentifica en el sistema.	
<b>Complejidad</b>	Baja.	
<b>Prioridad</b>	Media.	
<b>Precondiciones</b>	Que se encuentre creada la cuenta del usuario.	
<b>Poscondiciones</b>	Se autenticó un usuario.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.2 Muestra un formulario solicitando: <ul style="list-style-type: none"> <li>• Usuario(Obligatorio)</li> <li>• Contraseña (Obligatorio)</li> </ul>
2	2.1 Introduce los datos.	2.2 Valida los datos (Alterno 1). 2.3 Muestra una interfaz.
3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Usuario y Contraseña incorrecto.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.2 del flujo básico, muestra un mensaje de alerta "Usuario o Contraseña incorrecto".
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	RF1	
<b>Asuntos pendientes</b>		

## CU Gestionar Periféricos.

<b>Objetivo</b>	Modificar, Eliminar y Adicionar periféricos.	
<b>Actores</b>	RAF.	
<b>Resumen</b>	El caso de uso se inicia cuando el RAF desea actualización, elimina o inserta algún periférico.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Media.	
<b>Precondiciones</b>	Que se encuentre autenticado un usuario con rol RAF.	
<b>Poscondiciones</b>	Se modificó, eliminó o adicionó un periférico.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>		<b>Sistema</b>
1.	1.1 Hace clic en la opción "Periféricos".	1.2 Muestra una ventana con un formulario con las opciones "Modificar", "Eliminar", "Adicionar". <ul style="list-style-type: none"> <li>• Si desea Añadir Periféricos, va a la sección "Insertar Periféricos".</li> <li>• Si desea Eliminar Periféricos, va a la sección "eliminar".</li> <li>• Si desea Modificar Periféricos, va a la sección "editar".</li> </ul>
<b>Sección "editar".</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Selecciona el periférico que desee modificar. 1.2 Hace clic en la opción "editar"	1.3 Muestra un formulario con los datos del periférico. <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Estado.</li> <li>• SN.</li> <li>• Marca.</li> </ul>

		<ul style="list-style-type: none"> <li>• Puerto.</li> <li>• PC.</li> </ul>
2	<p>2.1 Introduce los datos que desee modificar.</p> <p>2.2 Hace clic en el botón "Editar".</p>	<p>2.3 Valida los datos (Alternativo 1).</p> <p>2.4 Modifica los datos.</p> <p>2.5 Muestra un mensaje "Se modificó correctamente el periférico".</p>
3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrecto.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje de alerta "Por favor, inserte correctamente los datos".
2		2.1 Termina el caso de uso.
<b>Sección "Eliminar".</b>		
<b>Actor</b>		<b>Sistema</b>
1	<p>1.1 Selecciona el periférico que desee eliminar.</p> <p>1.2 Hace clic en la opción "Eliminar".</p>	1.3 Muestra un mensaje de confirmación "Estas seguro que deseas eliminar".
2	2.1 Presiona la opción "Aceptar"	2.3 Elimina el periférico.
3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>Sección "Insertar Periféricos".</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Presiona la opción "Insertar Periféricos".	<p>1.1 Se muestra un formulario solicitando la siguiente información:</p> <ul style="list-style-type: none"> <li>• Nombre.</li> </ul>

		<ul style="list-style-type: none"> <li>• Marca.</li> <li>• Puerto.</li> <li>• SN.</li> <li>• PC.</li> <li>• Estado.</li> </ul>
2	<p>2.1 Introduce los datos.</p> <p>2.2 Hace clic en la opción "Insertar".</p>	<p>2.3 Valida los datos (Alternativo 1).</p> <p>2.4 Inserta los datos.</p> <p>2.5 Muestra un mensaje "Se insertó correctamente el periférico".</p>
3		3.1 Termina el caso de uso
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrecto.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico, muestra un mensaje de alerta "Por favor, inserte correctamente los datos".
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>no</b>	<b>RF14, RF15, RF16</b>

CU Gestionar partes PC.

<b>Objetivo</b>	Modificar, Eliminar y Adicionar partes PC.
<b>Actores</b>	RAF.
<b>Resumen</b>	El caso de uso se inicia cuando el RAF desea actualización, elimina o inserta algún parte de la PC.
<b>Complejidad</b>	Media.
<b>Prioridad</b>	Media.

<b>Precondiciones</b>	Que se encuentre autenticado un usuario con rol RAF.	
<b>Poscondiciones</b>	Se modificó, eliminó o adicionó un componente de la PC.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>		<b>Sistema</b>
1.	1.1 Hace clic en la opción "Partes PC".	<p>1.2 Muestra una barra con las opciones "Modificar", "Eliminar", "Adicionar" y una lista de Partes PC.</p> <ul style="list-style-type: none"> <li>• Si desea Añadir Partes PC, va a la sección "Insertar Partes PC".</li> <li>• Si desea Eliminar Partes PC, va a la sección "Eliminar".</li> <li>• Si desea Modificar Partes PC, va a la sección "Editar".</li> </ul>
<b>Sección "Editar".</b>		
<b>Actor</b>		<b>Sistema</b>
1	<p>1.1 Selecciona la Parte PC que quieres modificar.</p> <p>1.2 Hacer clic en el botón "Editar".</p>	<p>1.3 Muestra un formulario con los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Descripción.</li> <li>• Ubicación.</li> <li>• Estado.</li> <li>• PC.</li> </ul>
2	<p>2.1 Introduce los datos que desea modificar.</p> <p>2.2 Hace clic en el botón "Editar".</p>	<p>2.3 Validar los datos (Alternativo 1).</p> <p>2.4 Modificar los datos.</p> <p>2.5 Muestra un mensaje "Se modificó correctamente una componente de la</p>

		PC”.
3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrecto.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje de alerta “Por favor, inserte correctamente los datos”.
2		2.1 Termina el caso de uso.
<b>Sección “Eliminar”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Selecciona la Parte PC que desea eliminar. 1.2 Hace clic en el botón “Eliminar”.	1.3 Muestra un mensaje de confirmación “Estas seguro que desea eliminar”.
2	2.1 Presiona la opción “Aceptar”.	2.2 Elimina el componente de la pc.
3		3.1 Termina el caso de uso.
<b>Sección “Insertar Partes PC”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Hace clic en la opción “Insertar Partes PC”.	1.2 Muestra un formulario solicitando la siguiente información: <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Descripción.</li> <li>• Ubicación.</li> <li>• Estado.</li> <li>• PC.</li> </ul>
2	2.1 Introduce los datos.	2.3 Valida los datos (Alternativo 1).

	2.2 Presiona el botón "Insertar".	2.4 Inserta los datos.  2.5 Muestra un mensaje "Se insertó correctamente un componente de la PC".
3		3.1 Termina el caso de uso
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrecto.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico, muestra un mensaje de alerta "Por favor, inserte correctamente los datos".
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>no</b>	<b>RF17, RF18, RF19</b>

CU Gestionar Aire.

<b>Objetivo</b>	Modificar, Eliminar y Adicionar Aire.
<b>Actores</b>	RAF.
<b>Resumen</b>	El caso de uso se inicia cuando el RAF desea actualización, elimina o inserta algún Aire.
<b>Complejidad</b>	Media.
<b>Prioridad</b>	Media.
<b>Precondiciones</b>	Que se encuentre autenticado un usuario con rol RAF.
<b>Poscondiciones</b>	Se eliminó, modificó y actualizó una PC.
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
<b>Actor</b>	<b>Sistema</b>

1.	1.1 Hace clic en la opción "Aire".	1.2 Muestra una barra con las opciones "Modificar", "Eliminar", "Adicionar" y lista de Aire. <ul style="list-style-type: none"> <li>• Si desea Añadir Aire, va a la sección "Insertar Aire".</li> <li>• Si desea Eliminar Aire, va a la sección "Eliminar".</li> <li>• Si desea Modificar Aire, va a la sección "Editar".</li> </ul>
<b>Sección "Editar".</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Selecciona el Aire que desee modificar.  1.2 Hace clic en el botón "Editar".	1.3 Muestra un formulario con los datos: <ul style="list-style-type: none"> <li>• Inventario.</li> <li>• Estado.</li> <li>• Marca.</li> <li>• SN.</li> <li>• Local.</li> </ul>
2	2.1 Modifica los datos.  2.2 Hace clic en el botón "Editar".	2.3 Valida los datos (Alternativo 1).  2.4 Modifica los datos.  2.5 Muestra un mensaje "Se modificó correctamente el Aire".
3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje de alerta "Por

		favor, inserte correctamente los datos”.
2		2.1 Termina el caso de uso.
<b>Sección “Eliminar”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Selecciona el Aire que desea eliminar. 1.2 Hace clic en el botón “Eliminar”.	1.3 Muestra un mensaje de confirmación “Por favor desea eliminar”.
2	2.1 Presiona la opción “Aceptar”.	2.2 Elimina el Aire.
3		3.1 Termina el caso de uso.
<b>Sección “Insertar Aire”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Hace clic en la opción “Insertar Aire”.	1.2 Muestra un formulario con los datos.  <ul style="list-style-type: none"> <li>• Inventario.</li> <li>• Estado.</li> <li>• Marca.</li> <li>• SN.</li> <li>• Local.</li> </ul>
2	2.1 Introduce datos.  2.2 Hace clic en el botón “insertar”.	2.3 Valida los datos (Alternativo 1).  2.4 Inserta los datos.  2.5 Muestra un mensaje se “Se insertó correctamente la PC”.
3		3.1 Termina el caso de uso
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>

1		1.1 En el paso 2.3 del flujo básico muestra un mensaje “Por favor, inserte correctamente los datos”.
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>RF5, RF6, RF7.</b>	

CU Gestionar Local.

<b>Objetivo</b>	Modificar, Eliminar y Adicionar local.	
<b>Actores</b>	RAF.	
<b>Resumen</b>	El caso de uso se inicia cuando el RAF desea actualización, elimina o inserta algún local.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Media.	
<b>Precondiciones</b>	Que se encuentre autenticado un usuario con rol RAF.	
<b>Poscondiciones</b>	Se eliminó, modificó y actualizó un local.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>	<b>Sistema</b>	
1.	1.1 Hace clic en la opción “Local”.	1.2 Muestra una barra con las opciones “Modificar”, “Eliminar”, “Adicionar” y lista de local. <ul style="list-style-type: none"> <li>• Si desea Añadir Local, va a la sección “Insertar Local”.</li> <li>• Si desea Eliminar Local, va a la sección “Eliminar”.</li> <li>• Si desea Modificar Local, va a la sección “Editar”.</li> </ul>
<b>Sección “Editar”.</b>		

Actor		Sistema
1	1.1 Selecciona el local que desee modificar.  1.2 Hace clic en el botón "Editar".	1.3 Muestra un formulario con los datos:  <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Descripción.</li> <li>• Estado.</li> </ul>
2	2.1 Modifica los datos.  2.2 Hace clic en el botón "Editar".	2.3 Valida los datos (Alternativo 1).  2.4 Modifica los datos.  2.5 Muestra un mensaje "Se modificó correctamente el local".
3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
Actor		Sistema
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje de alerta "Por favor, inserte correctamente los datos".
2		2.1 Termina el caso de uso.
<b>Sección "Eliminar".</b>		
Actor		Sistema
1	1.1 Selecciona el local que desea eliminar. 1.2 Hace clic en el botón "Eliminar".	1.3 Muestra un mensaje de confirmación "Por favor desea eliminar".
2	2.1 Presiona la opción "Aceptar".	2.2 Elimina el local.
3		3.1 Termina el caso de uso.
<b>Sección "Insertar Local".</b>		
Actor		Sistema

1	1.1 Hace clic en la opción "Insertar Local".	1.2 Muestra un formulario con los datos.  <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Estado.</li> <li>• Descripción.</li> </ul>
2	2.1 Introduce datos.  2.2 Hace clic en el botón "insertar".	2.3 Valida los datos (Alternativo 1).  2.4 Inserta los datos.  2.5 Muestra un mensaje "Se insertó correctamente el local".
3		3.1 Termina el caso de uso
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.3 En el paso 2.3 del flujo básico muestra un mensaje "Por favor, inserte correctamente los datos".
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>		<b>RF8, RF9, RF10</b>

CU Gestionar Monitor.

<b>Objetivo</b>	Modificar, Eliminar y Adicionar Monitor.
<b>Actores</b>	RAF.
<b>Resumen</b>	El caso de uso se inicia cuando el RAF desea actualización, elimina o inserta algún Monitor.
<b>Complejidad</b>	Media.
<b>Prioridad</b>	Media.

<b>Precondiciones</b>	Que se encuentre autenticado un usuario con rol RAF.	
<b>Poscondiciones</b>	Se eliminó, modificó y actualizó un Monitor.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>		<b>Sistema</b>
1.	1.1 Hace clic en la opción "Monitor".	<p>1.2 Muestra una barra con las opciones "Modificar", "Eliminar", "Adicionar" y lista de Monitor.</p> <ul style="list-style-type: none"> <li>• Si desea Añadir Monitor, va a la sección "Insertar Monitor".</li> <li>• Si desea Eliminar Monitor, va a la sección "Eliminar".</li> <li>• Si desea Modificar Monitor, va a la sección "Editar".</li> </ul>
<b>Sección "Editar".</b>		
<b>Actor</b>		<b>Sistema</b>
1	<p>1.1 Selecciona el Monitor que desee modificar.</p> <p>1.2 Hace clic en el botón "Editar".</p>	<p>1.3 Muestra un formulario con los datos:</p> <ul style="list-style-type: none"> <li>• Inventario.</li> <li>• Marca.</li> <li>• SN.</li> <li>• Tipo</li> <li>• Ubicación.</li> <li>• PC.</li> </ul>
2	<p>2.1 Modifica los datos.</p> <p>2.2 Hace clic en el botón "Editar".</p>	<p>2.3 Valida los datos (Alternativo 1).</p> <p>2.4 Modifica los datos.</p> <p>2.5 Muestra un mensaje "Se modificó correctamente el Monitor".</p>

3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje de alerta “Por favor, inserte correctamente los datos”.
2		2.1 Termina el caso de uso.
<b>Sección “Eliminar”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Selecciona el Monitor que desea eliminar. 1.2 Hace clic en el botón “Eliminar”.	1.3 Muestra un mensaje de confirmación “Por favor desea eliminar”.
2	2.1 Presiona la opción “Aceptar”.	2.2 Elimina el Monitor.
3		3.1 Termina el caso de uso.
<b>Sección “Insertar Monitor”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Hace clic en la opción “Insertar Monitor”.	1.2 Muestra un formulario con los datos.  <ul style="list-style-type: none"> <li>• Inventario.</li> <li>• Marca.</li> <li>• SN.</li> <li>• Tipo</li> <li>• Ubicación.</li> <li>• PC.</li> </ul>
2	2.1 Introduce datos.  2.2 Hace clic en el botón “insertar”.	2.3 Valida los datos (Alternativo 1).  2.4 Inserta los datos.

		2.5 Muestra un mensaje se “Se insertó correctamente el Monitor”.
3		3.1 Termina el caso de uso
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje “Por favor, inserte correctamente los datos”.
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>RF23, RF24, RF25.</b>	

CU Gestionar PC.

<b>Objetivo</b>	Modificar, Eliminar y Adicionar PC.	
<b>Actores</b>	RAF.	
<b>Resumen</b>	El caso de uso se inicia cuando el RAF desea actualización, elimina o inserta alguna PC.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Media.	
<b>Precondiciones</b>	Que se encuentre autenticado un usuario con rol RAF.	
<b>Poscondiciones</b>	Se eliminó, modificó y actualizó una PC	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>		<b>Sistema</b>
1.	1.1 Hace clic en la opción “PC”.	1.2 Muestra una barra con las opciones “Modificar”, “Eliminar”, “Adicionar” y lista de PC.  <ul style="list-style-type: none"> <li>• Si desea Añadir PC, va a la</li> </ul>

		<p>sección “Insertar PC”.</p> <ul style="list-style-type: none"> <li>• Si desea Eliminar PC, va a la sección “Eliminar”.</li> <li>• Si desea Modificar PC, va a la sección “Editar”.</li> </ul>
<b>Sección “Editar”.</b>		
<b>Actor</b>		<b>Sistema</b>
1	<p>1.1 Selecciona el local que desee modificar.</p> <p>1.2 Hace clic en el botón “Editar”.</p>	<p>1.3 Muestra un formulario con los datos:</p> <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Inventario.</li> <li>• Ubicación.</li> <li>• Marca.</li> <li>• Estado.</li> <li>• SN.</li> </ul>
2	<p>2.1 Modifica los datos.</p> <p>2.2 Hace clic en el botón “Editar”.</p>	<p>2.3 Valida los datos (Alternativo 1).</p> <p>2.4 Modifica los datos.</p> <p>2.5 Muestra un mensaje “Se modificó correctamente la PC”.</p>
3		3.1 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje de alerta “Por favor, inserte correctamente los datos”.
2		2.1 Termina el caso de uso.
<b>Sección “Eliminar”.</b>		

Actor		Sistema
1	1.1 Selecciona la PC que desea eliminar. 1.2 Hace clic en el botón "Eliminar".	1.3 Muestra un mensaje de confirmación "Por favor desea eliminar".
2	2.1 Presiona la opción "Aceptar".	2.2 Elimina la PC.
3		3.1 Termina el caso de uso.
<b>Sección "Insertar PC".</b>		
Actor		Sistema
1	1.1 Hace clic en la opción "Insertar PC".	1.2 Muestra un formulario con los datos. <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Inventario.</li> <li>• Ubicación.</li> <li>• Marca.</li> <li>• Estado.</li> <li>• SN.</li> <li>• Local.</li> </ul>
2	2.1 Introduce datos.  2.2 Hace clic en el botón "insertar".	2.3 Valida los datos (Alternativo 1).  2.4 Inserta los datos.  2.5 Muestra un mensaje se "Se insertó correctamente la PC".
3		3.1 Termina el caso de uso
<b>Flujos alternos</b>		
<b>NºEvento1 Los datos son incorrectos.</b>		
Actor		Sistema
1		1.1 En el paso 2.3 del flujo básico muestra un mensaje "Por favor,

		inserte correctamente los datos”.
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>RF22, RF21, RF20.</b>	

CU Realizar Reporte.

<b>Objetivo</b>	Llevar el control de los medios básicos que se encuentran en reparación o préstamo.	
<b>Actores</b>	RAF.	
<b>Resumen</b>	El caso de uso se inicia cuando el RAF desee saber los medios básicos que están en préstamo o reparación.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Media.	
<b>Precondiciones</b>	Que exista medios básicos o PC en reparación o en préstamo.	
<b>Poscondiciones</b>	Se realizó el reporte.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>	<b>Sistema</b>	
1.	1.1 Hace clic en la opción “Realizar Reporte”.	1.2 Muestra un formulario con un select con las opciones “Préstamo” y “Reparación”: <ul style="list-style-type: none"> <li>• Si desea buscar medios básicos y PC prestados selecciona la opción: “préstamo”.</li> <li>• Si desea buscar medios básicos y PC en reparaciones selecciona la opción:</li> </ul>

		"reparación".
<b>Sección "préstamo".</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Hace clic en el botón "Buscar".	1.3 Muestra un listado con los medios básicos y PC en préstamo. <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• inventario.</li> </ul>
2		2.1 Termina el caso de uso.
<b>Sección "reparación".</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Hace clic en el botón "Buscar".	1.2 Muestra un listado con los medios básicos y PC en reparación. <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• inventario.</li> </ul>
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>RF26</b>	

CU Realizar Movimiento.

<b>Objetivo</b>	Tener el control y documentado los medios básicos.
<b>Actores</b>	FAR
<b>Resumen</b>	El caso de uso se inicia cuando el responsable de activos fijos tenga que hacer algún movimiento de medios básicos.
<b>Complejidad</b>	Baja.
<b>Prioridad</b>	Media.

<b>Precondiciones</b>	Que se encuentre autenticado un usuario con rol RAF.	
<b>Poscondiciones</b>	Se realizó el movimiento.	
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
<b>Actor</b>		<b>Sistema</b>
1	1.1 Seleccionar la opción "Movimiento".	1.2 Muestra una Interfaz con un botón "Realizar Movimiento".
2	2.1 Oprime el botón "Realizar Movimiento".	2.3 Muestra una interfaz con un formulario, con la siguiente información: <ul style="list-style-type: none"> <li>• Medio Básico.</li> <li>• Fecha de confección.</li> <li>• Tipo.</li> <li>• Entidad.</li> <li>• Dirección Destino.</li> <li>• Nombre.</li> </ul>
3	3.1 Inserta los datos.  3.2 Oprime el botón "Guardar".	3.3 Valida los Datos (alternos 1).  3.4 Muestra una interfaz con una tabla con los datos insertados.
4	4.1 Selecciona imprimir.	4.2 Termina el caso de uso.
<b>Flujos alternos</b>		
<b>NºEvento1 datos incorrectos.</b>		
<b>Actor</b>		<b>Sistema</b>
1		1.1 En el paso 3.3 del flujo básico, muestra un mensaje de alerta "Por favor, inserte bien los datos".
2		2.1 Termina el caso de uso.
<b>Requisitos funcionales</b>	<b>RF27</b>	

Anexo2. Diagramas De Clase

Diagrama de clase CU Gestionar Local.

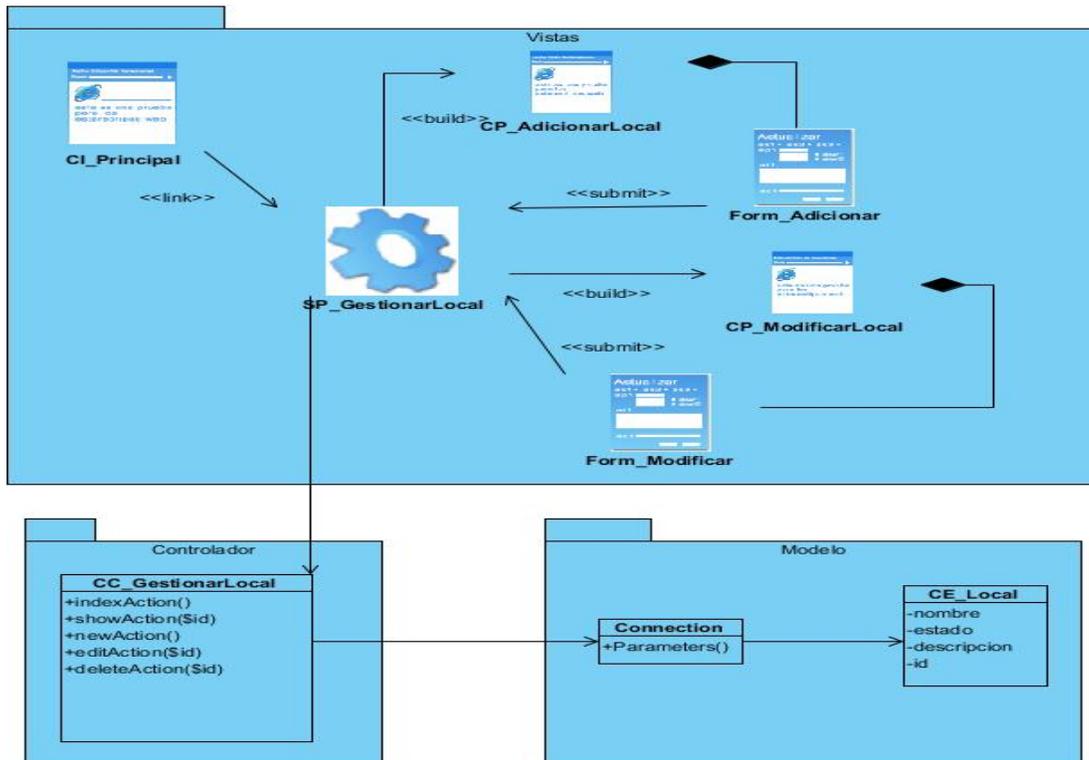


Diagrama de clase CU Gestionar Aire.

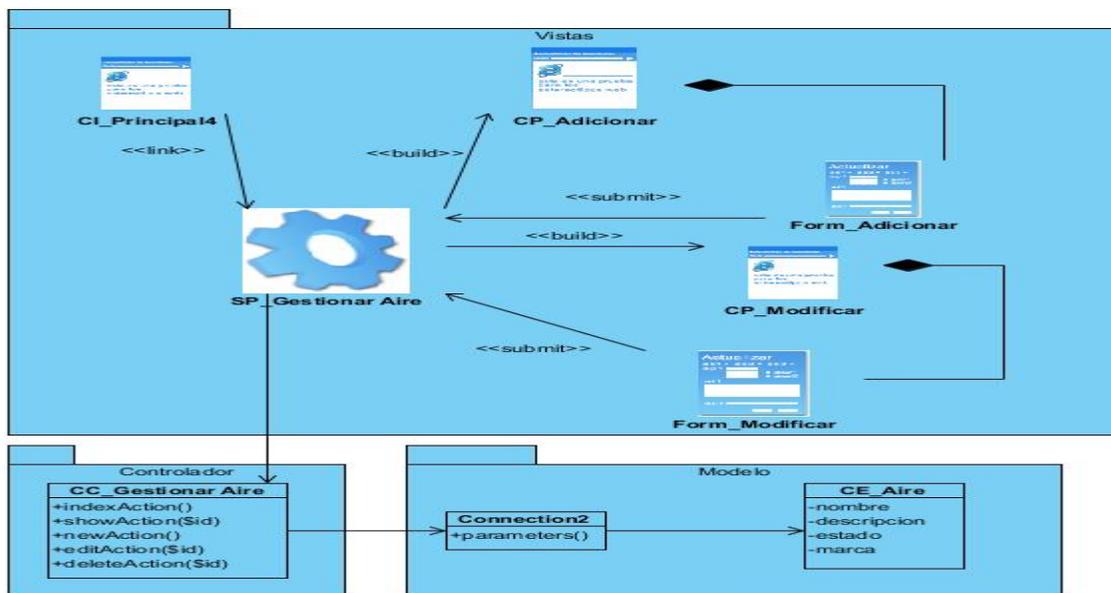


Diagrama de clase CU Realizar Reporte.

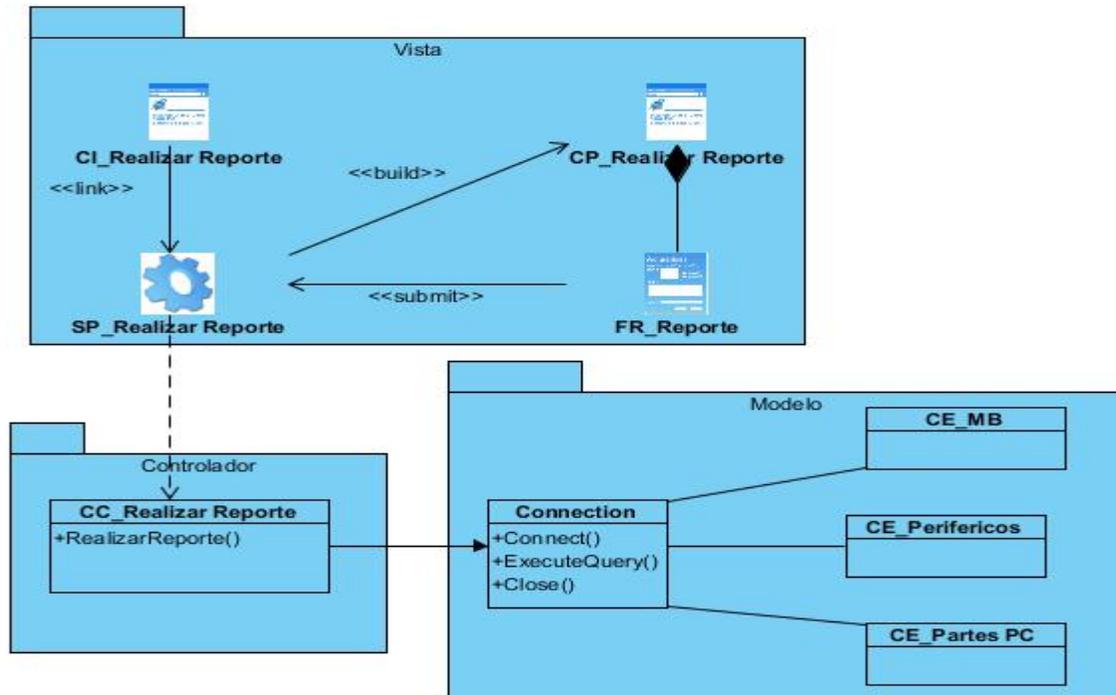


Diagrama de clase CU Gestionar PC.

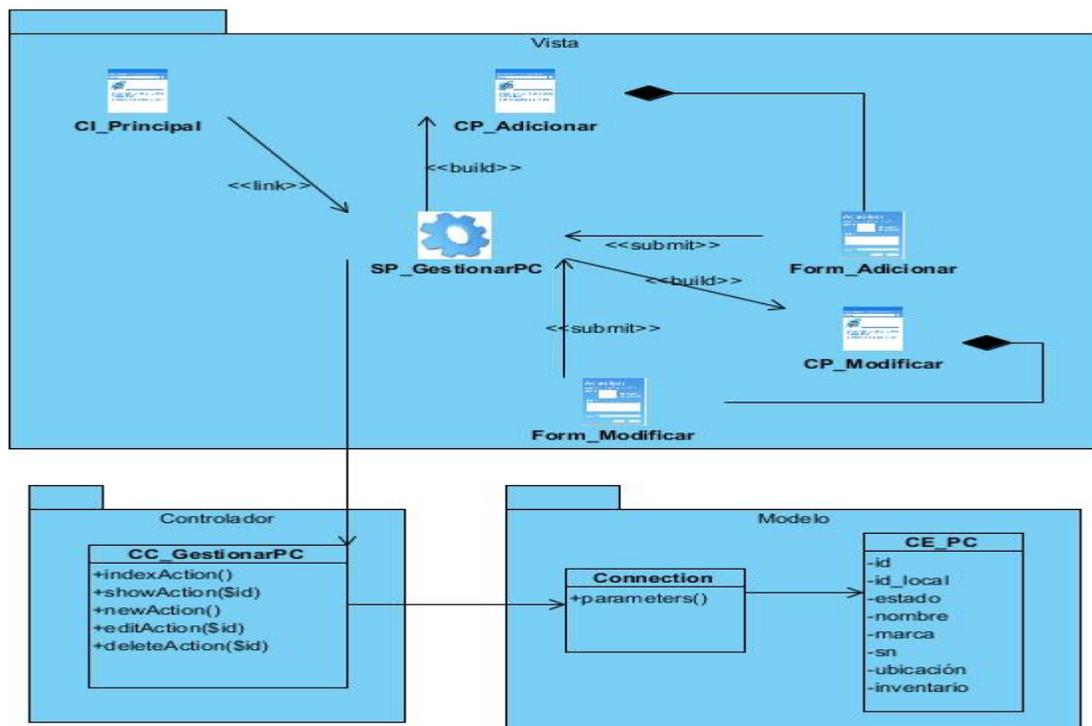


Diagrama de clase CU Gestionar Partes PC.

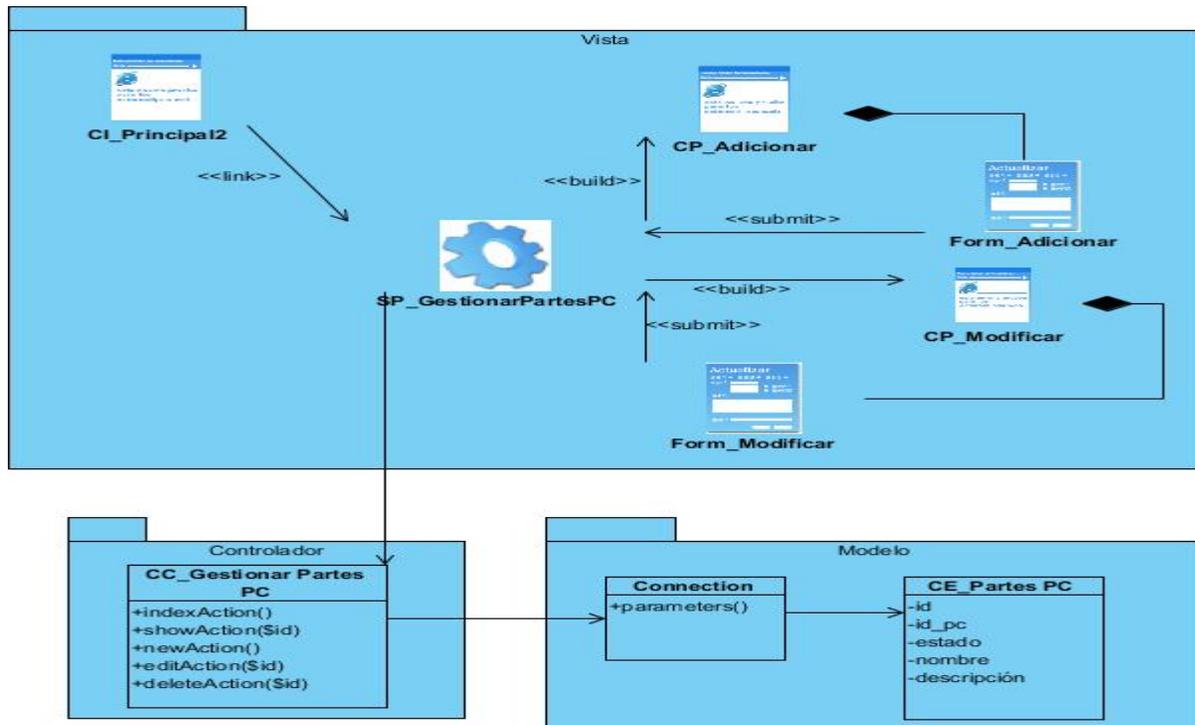


Diagrama de clase CU Gestionar Periférico.

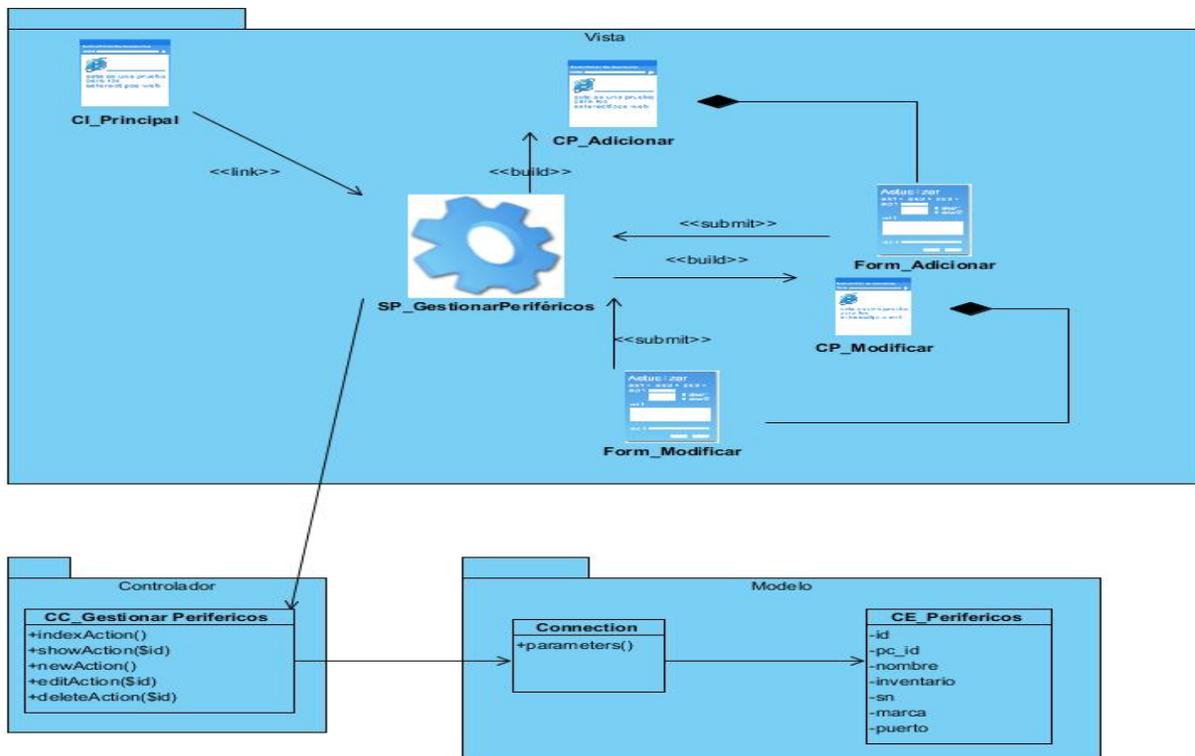


Diagrama de clase CU Gestionar Monitor.

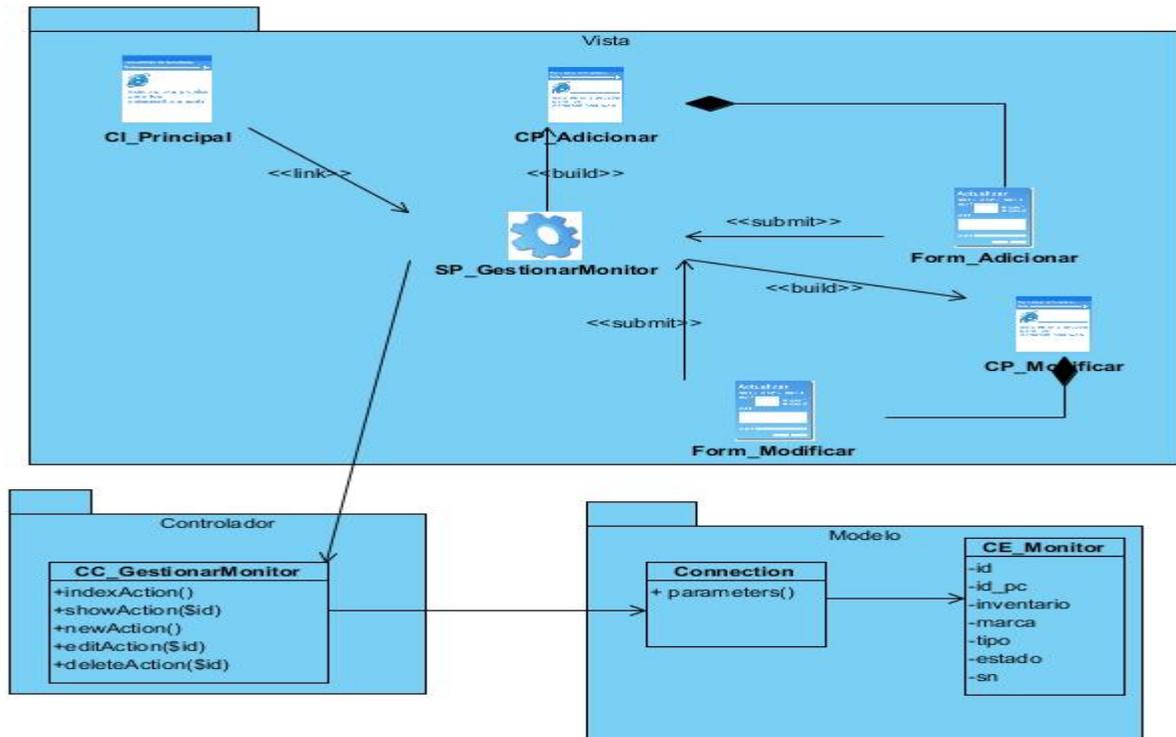


Diagrama de clase CU Realizar Movimiento.

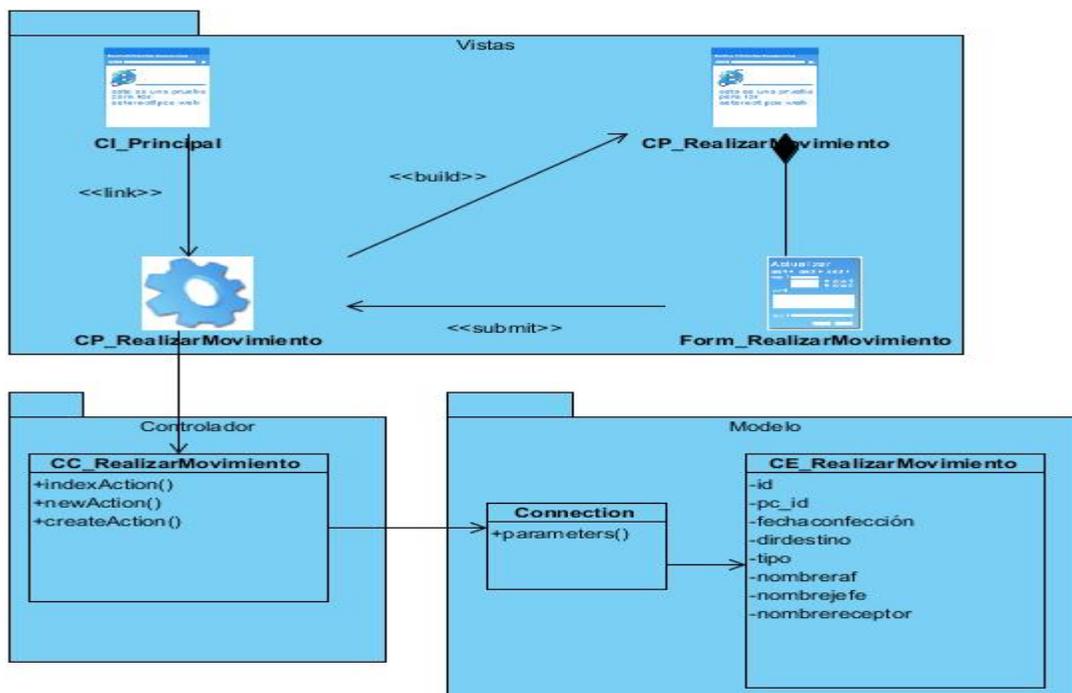


Diagrama de clase CU Gestionar Usuario.

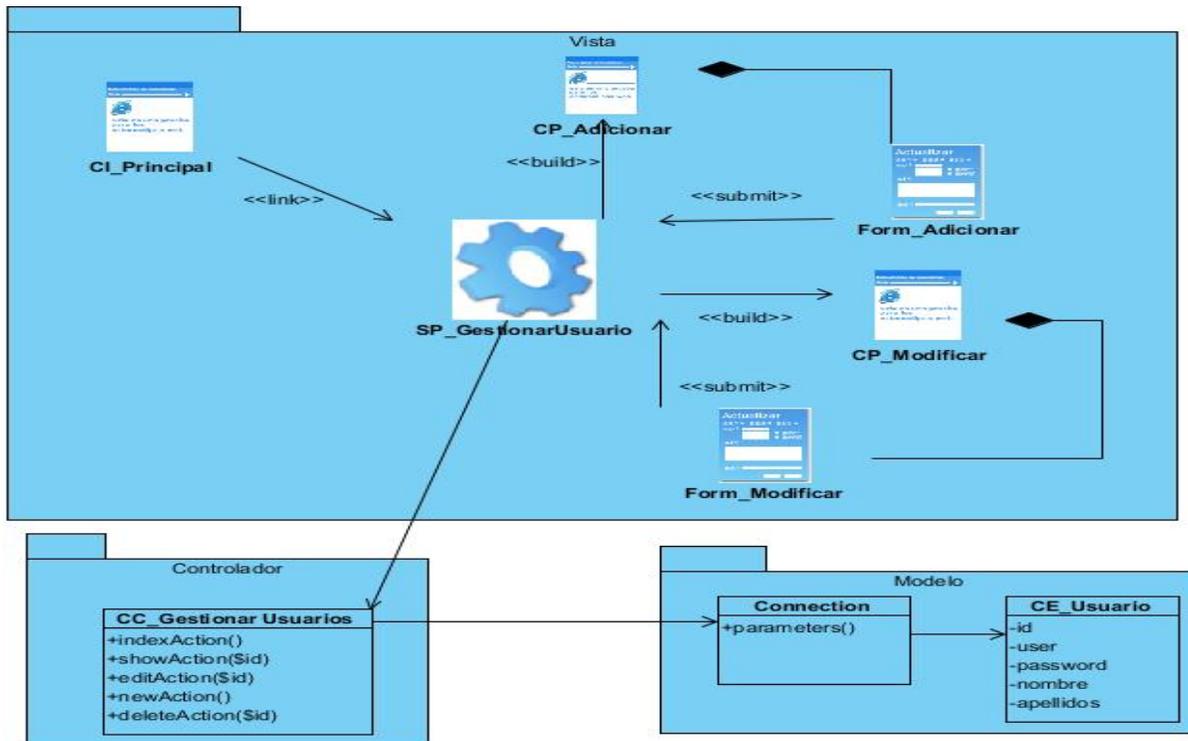
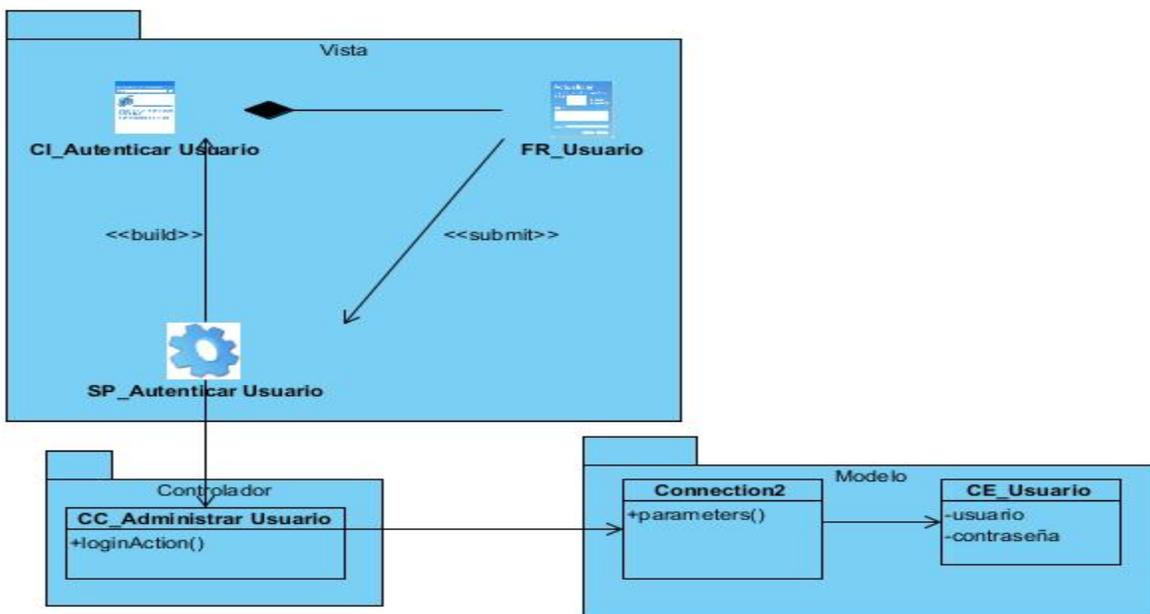
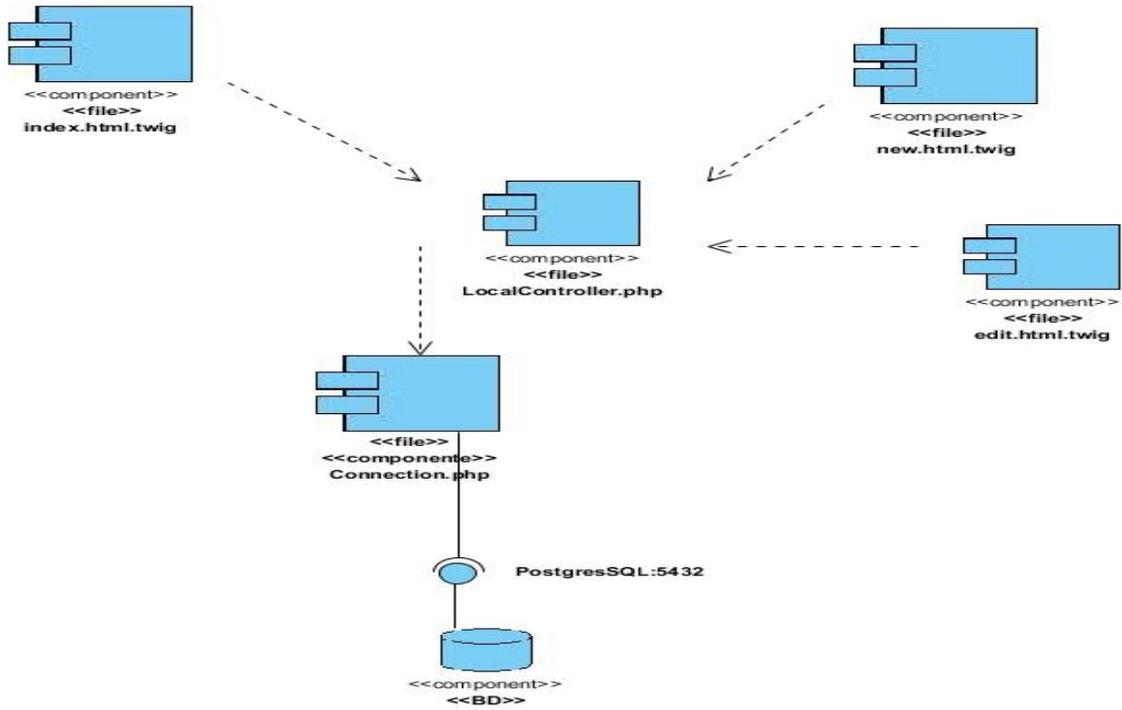


Diagrama de clase CU Autenticarse.



### Anexo 3. Diagramas de Componentes

#### Diagrama de componente CU Gestionar Local.



#### Diagrama de componente CU Gestionar Aire.

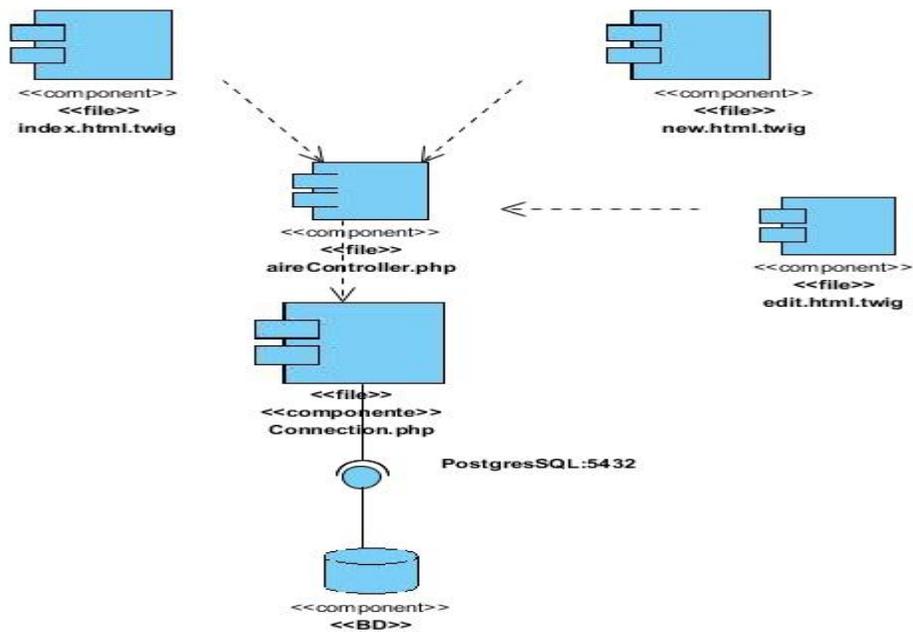


Diagrama de componente CU Gestionar PC.

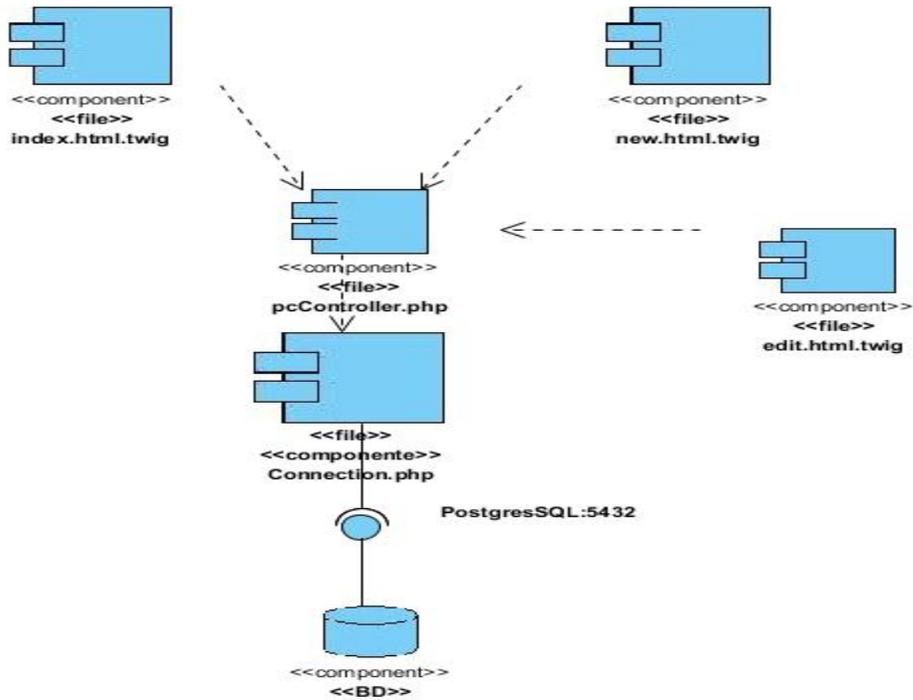


Diagrama de componente CU Gestionar Partes PC.

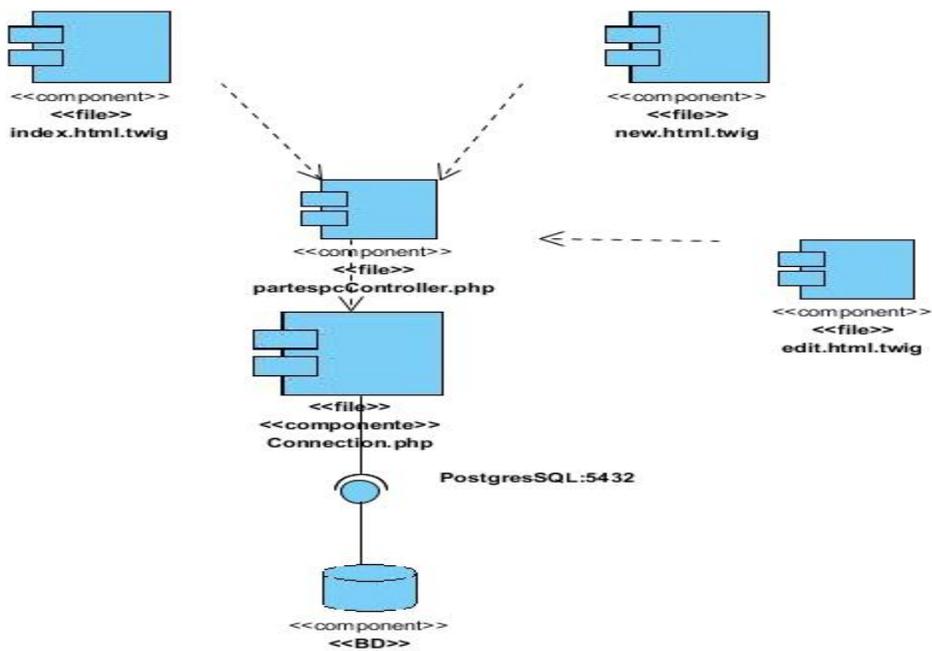


Diagrama de componente CU Gestionar Periférico.

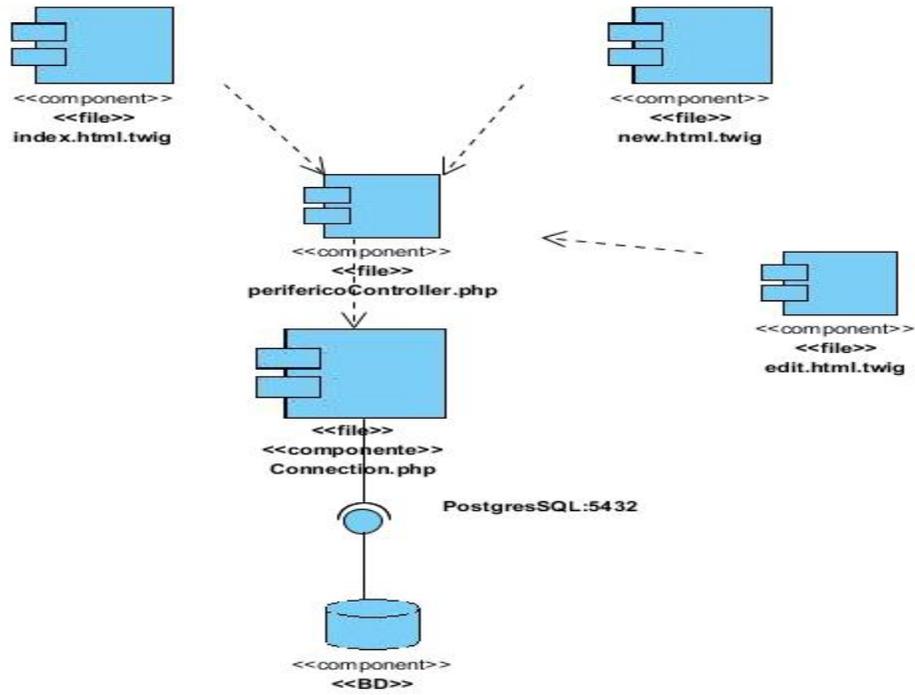


Diagrama de componente CU Gestionar Monitor.

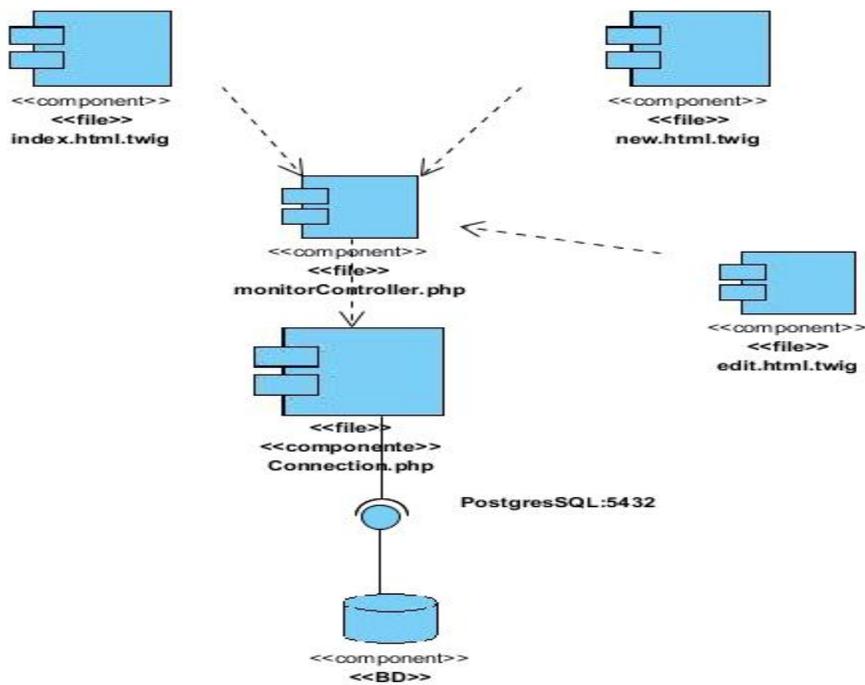


Diagrama de componente CU Realizar Movimiento.

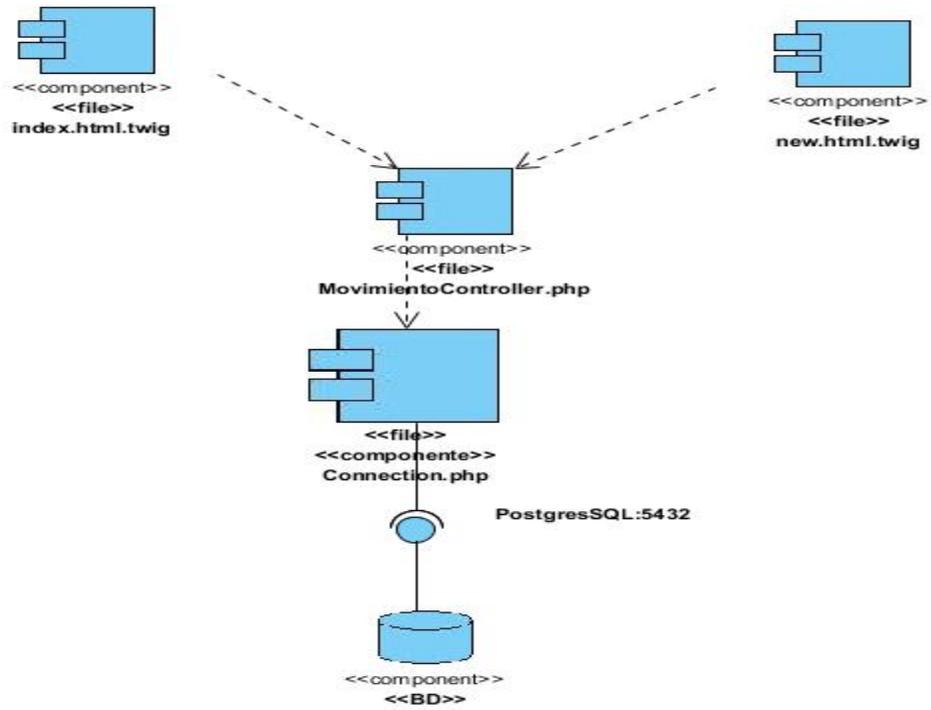


Diagrama de componente CU Gestionar Usuario.

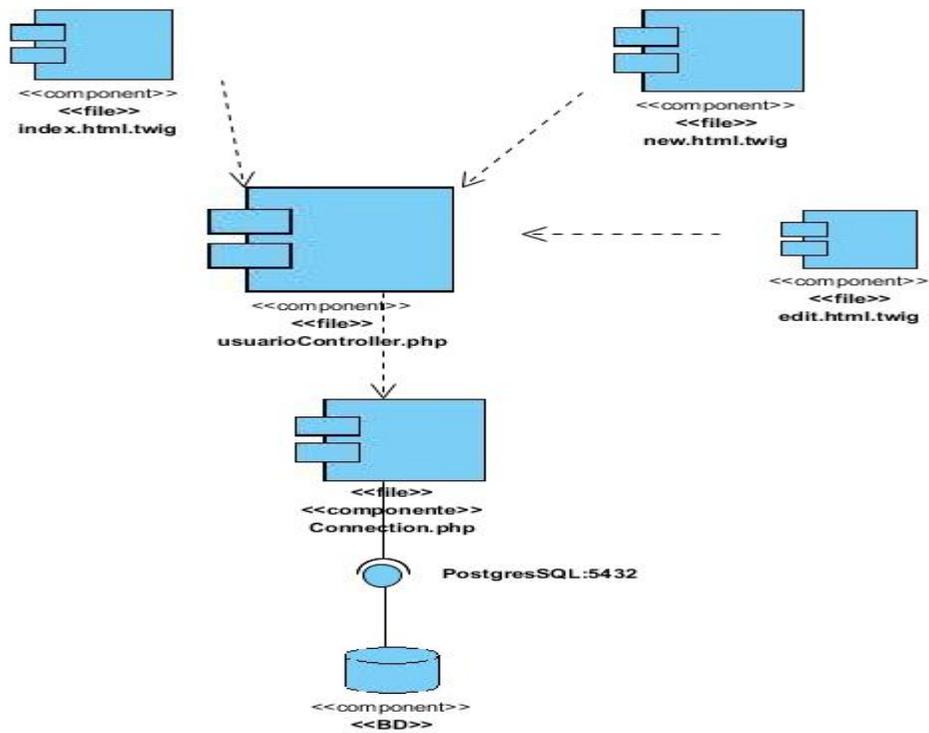


Diagrama de componente CU Autenticar.

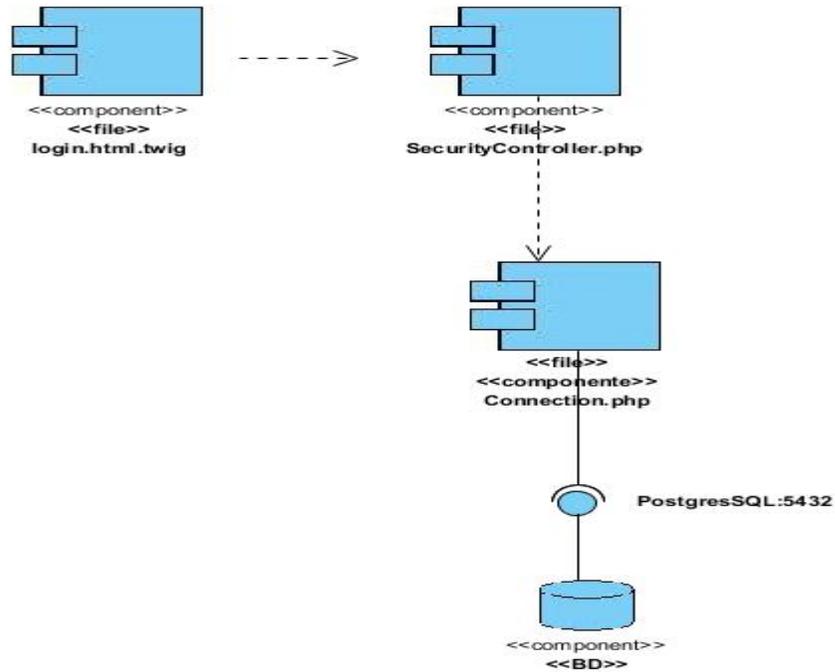
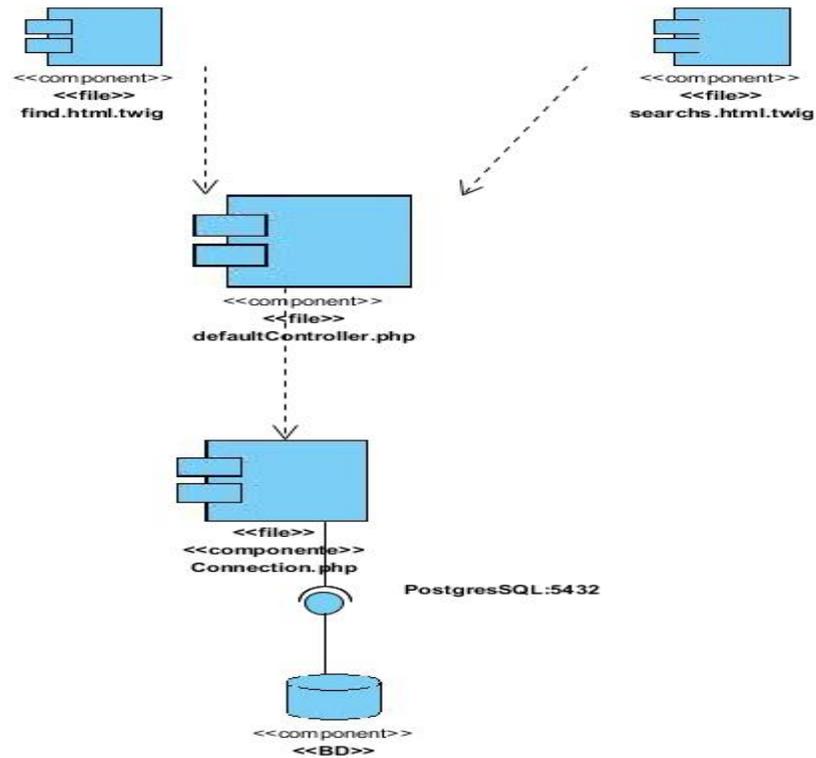


Diagrama de componente CU Realizar Reporte.



## Anexo 4. Diseño de Casos de Prueba

### Caso de prueba CU Autenticar Usuario.

✓ **Breve Descripción.**

Este caso de prueba permite comprobar la funcionalidad de tener un control de acceso al sistema.

✓ **Flujo Central.**

Introducir usuario y contraseña.

El sistema te muestra una Interfaz.

✓ **Condiciones de Ejecución.**

El usuario debe de estar autenticado en el sistema.

✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducir datos correctos y presiona el botón "Entrar".		El usuario tenga acceso al sistema.	El usuario tiene acceso al sistema.
	Pulsar "Entrar" sin haber entrado los datos.	El sistema debe mostrar un mensaje usuario y contraseña incorrecta.	El sistema muestra el mensaje.

### Caso de Prueba Gestionar Local.

✓ **Breve Descripción General.**

Este caso de prueba permite comprobar la funcionalidad de modificar, eliminar, adicionar un local al sistema.

✓ **Flujo Central.**

Seleccionar la opción Local.

✓ **Condiciones de Ejecución.**

El responsable de activos fijo debe haberse autenticado.

**SC1.Insertar local.**

✓ **Flujo Central.**

Seleccionar la opción Insertar Local.

Introducir datos necesarios.

**Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducir los datos correcto y presiona el botón "Insertar".		El sistema debe mostrar una interfaz con los datos entrados, luego de ser almacenados en la base de datos. Mostrar un mensaje "se insertó correctamente el local".	Muestra interfaz con los datos insertados luego de ser almacenados en la base de datos. Muestra un mensaje "se insertó correctamente el local".
	Pulsar "Insertar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.

**SC2.Editar Local.**

✓ **Flujo Central.**

Seleccionar la opción Editar.

Modificar los datos deseados.

✓ **Condiciones de ejecución.**

Exista un Local Creado.

✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Modificar los datos correctamente y presiona el botón "Editar".		Muestra una interfaz con todos los datos modificados, luego de actualizar la base de datos.	El sistema muestra la interfaz con los datos modificados, luego de haber actualizado la base de datos.
	Pulsar "Editar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	El sistema muestra el mensaje.

### SC3.Eliminar Local.

- ✓ **Flujo Central.**  
Selecciona la opción Eliminar.
- ✓ **Condiciones de Ejecución.**  
Que exista un local creado.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
	Presionar el botón "Eliminar".	El sistema debe eliminar los datos del local.	El sistema elimina el local creado y todo lo que está relacionado a él.

## Caso de Prueba Gestionar Usuario.

✓ **Breve Descripción.**

Este caso de prueba permite comprobar las funcionalidades de modificar, eliminar e insertar un usuario al sistema.

✓ **Flujo Central.**

Seleccionar la opción Usuario.

✓ **Condiciones de Ejecución.**

El administrador del sistema debe haberse autenticado.

### EC1.Insertar Usuario.

#### Flujo Central.

Seleccionar la opción Insertar Usuario.

Introducir los datos necesarios.

#### Iteraciones.

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducir los datos correcto y presiona el botón "Insertar".		Muestra una interfaz con los datos entrados luego de ser almacenados en la base de datos.	El sistema muestra la interfaz con los datos insertados luego de ser almacenados en la base de datos.
	Pulsar "Insertar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	El sistema muestra mensaje por favor rellene este campo.
	Pulsar "Insertar" y haber entrado datos incorrectos.	El sistema debe mostrar mensaje de error.	El sistema muestra mensaje de error.

**SC2.Editar Usuario.**

- ✓ **Flujo Central.**  
 Seleccionar la opción Editar.  
 Modificar los datos deseados.
- ✓ **Condiciones de ejecución.**  
 Exista un usuario Creado.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Modificar los datos correctamente y presiona el botón "Editar".		Muestra una interfaz con todos los datos modificados, luego de ser actualizados en la base de datos.	El sistema muestra interfaz con los datos modificados, luego de ser actualizados en la base de datos.
	Pulsar "Editar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	
	Pulsar "Editar" y haber entrado los datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	

**SC3.Eliminar Usuario.**

- ✓ **Flujo Central.**  
 Selecciona la opción Eliminar.
- ✓ **Condiciones de Ejecución.**  
 Que exista un usuario creado.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Presionar el botón "Eliminar".		El sistema debe eliminar los datos del usuario.	Elimina los datos del usuario.

## Caso de Prueba Gestionar Aire.

✓ **Breve Descripción.**

Este caso de prueba permite comprobar las funcionalidades de modificar, eliminar e insertar al sistema.

✓ **Flujo Central.**

Seleccionar la opción Aire.

✓ **Condiciones de Ejecución.**

El Responsable de activos fijos debe haberse autenticado.

### EC1.Insertar Aire.

#### Flujo Central.

Seleccionar la opción Insertar Aire.

Introducir los datos necesarios.

#### Iteraciones.

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducir los datos correcto y presiona el botón "Insertar".		El sistema debe mostrar una interfaz con los datos del aire entrados por el usuario. Mostrar un mensaje "se insertó correctamente el aire".	Muestra una interfaz con los datos del aire. Muestra un mensaje "se insertó correctamente el aire".

	Pulsar "Insertar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra un mensaje "por favor rellene este campo".
	Pulsar "Insertar" y haber entrado datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite entrar cualquier dato.
	Pulsar "Insertar" y haber entrado un mismo número de inventario.	El sistema debe de mostrar un mensaje "Existe un aire con ese número de inventario".	Permite la entrada de dos aire con el mismo número de inventario.

## SC2.Editar Aire.

- ✓ **Flujo Central.**  
 Seleccionar la opción Editar.  
 Modificar los datos deseados.
- ✓ **Condiciones de ejecución.**  
 Exista un aire Creado.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Modificar los datos correctamente y presiona el botón "Editar".		El sistema debe de mostrar una interfaz con todos los datos modificados. Mostrar un mensaje "Se modificó correctamente el aire".	Muestra una interfaz con los datos modificados. Muestra un mensaje "Se modificó correctamente el aire".

	Pulsar "Editar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Editar" y haber entrado los datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite la entrada de cualquier tipo de datos.
	Pulsar "Editar" y haber entrado un mismo número de inventario.	El sistema debe mostrar un mensaje "Existe un aire con ese número de inventario".	Permite que existan más aire con el mismo número de inventario.

### SC3.Eliminar Aire.

- ✓ **Flujo Central.**  
Selecciona la opción Eliminar.
- ✓ **Condiciones de Ejecución.**  
Que exista un Aire creado.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Presionar el botón "Eliminar".		El sistema debe de eliminar los datos.	Elimina los datos.

### Caso de Prueba Gestionar Periféricos.

- ✓ **Breve Descripción.**  
Este caso de prueba permite comprobar las funcionalidades de modificar, eliminar e insertar al sistema.
- ✓ **Flujo Central.**  
Seleccionar la opción Periféricos.

✓ **Condiciones de Ejecución.**

El Responsable de activos fijos debe haberse autenticado.

**EC1.Insertar Periféricos.**

**Flujo Central.**

Seleccionar la opción Insertar Periféricos.

Introducir los datos necesarios.

**Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducir los datos correcto y presiona el botón "Insertar".		El sistema debe de mostrar una interfaz con los datos del periférico entrado. Mostar un mensaje "se insertó correctamente el periférico".	Muestra una interfaz con los datos del periférico. Muestra un mensaje "se insertó correctamente el periférico".
	Pulsar "Insertar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Insertar" y haber entrado datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite la entrada de cualquier tipo de dato.
	Pulsar "Insertar" y haber entrado el mismo número de serie.	El sistema debe de mostrar un mensaje "Existe un periférico con ese número de serie"	Permite que existan periféricos con el mismo número de serie.

**SC2.Editar Periféricos.**

**Flujo Central.**

- Seleccionar la opción Editar.
- Modificar los datos deseados.

- ✓ **Condiciones de ejecución.**  
Exista un medio Periféricos.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Modificar los datos correctamente y presiona el botón "Editar".		El sistema debe de mostrar una interfaz con todos los datos del periférico modificados. Mostrar un mensaje "se modificó correctamente el periférico".	Muestra una interfaz con todos los datos de periférico modificado. Muestra el mensaje "se modificó el periférico".
	Pulsar "Editar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Editar" y haber entrado los datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite la entrada de cualquier tipo de datos.
	Pulsar "Editar" y haber entrado el mismo número de serie.	El sistema debe de mostrar un mensaje "Existe un periférico con ese número de serie"	Permite que existan periféricos con el mismo número de serie.

**SC3.Eliminar Periféricos.**

- ✓ **Flujo Central.**  
Selecciona la opción Eliminar.
- ✓ **Condiciones de Ejecución.**  
Que exista un medio Periféricos.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Presionar el botón "Eliminar".		El sistema debe de eliminar los datos del periférico.	Elimina los datos del periférico.

**Caso de Prueba Gestionar Partes PC.**

- ✓ **Breve Descripción.**  
Este caso de prueba permite comprobar las funcionalidades de modificar, eliminar e insertar al sistema.
- ✓ **Flujo Central.**  
Seleccionar la opción Partes PC.
- ✓ **Condiciones de Ejecución.**  
El Responsable de activos fijos debe haberse autenticado.

**EC1.Insertar Partes PC.**

**Flujo Central.**

Seleccionar la opción Insertar Partes PC.

Introducir los datos necesarios.

**Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido

Introducir los datos correcto y presiona el botón "Insertar".		El sistema debe de mostrar una interfaz con los datos de la parte PC. Mostrar un mensaje "se insertó correctamente una parte PC".	Muestra una interfaz con los datos de la parte PC. Muestra un mensaje "se insertó correctamente una parte PC".
	Pulsar "Insertar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Insertar" y haber entrado datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite cualquier tipo de datos.
	Pulsar "Insertar" y haber entrado el mismo número de serie.	El sistema debe de mostrar un mensaje "Existe una parte PC con ese número de serie"	Permite que existan partes PC con el mismo número de serie.

## SC2.Editar Partes PC.

- ✓ **Flujo Central.**  
 Seleccionar la opción Editar.  
 Modificar los datos deseados.
- ✓ **Condiciones de ejecución.**  
 Exista un medio Partes PC.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
----------------	------------------	--------------------	--------------------

Modificar los datos correctamente y presiona el botón "Editar".		El sistema debe de mostrar una interfaz con los datos modificados. Mostrar un mensaje "se modificó correctamente una parte PC".	Muestra una interfaz con los datos de una parte PC ya modificados. Muestra un mensaje "se modificó correctamente una parte PC"
	Pulsar "Editar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Editar" y haber entrado los datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite cualquier tipo de datos.
	Pulsar "Editar" y haber entrado el mismo número de serie.	El sistema debe de mostrar un mensaje "Existe una parte PC con ese número de serie"	Permite que existan partes PC con el mismo número de serie.

### SC3.Eliminar Partes PC.

- ✓ **Flujo Central.**  
Selecciona la opción Eliminar.
- ✓ **Condiciones de Ejecución.**  
Que exista un medio Partes PC.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Presionar el botón "Eliminar".		El sistema debe de eliminar los datos.	Elimina los datos.

**Caso de Prueba Gestionar PC.**

✓ **Breve Descripción.**

Este caso de prueba permite comprobar las funcionalidades de modificar, eliminar e insertar al sistema.

✓ **Flujo Central.**

Seleccionar la opción PC.

✓ **Condiciones de Ejecución.**

El Responsable de activos fijos debe haberse autenticado.

**EC1.Insertar PC.**

**Flujo Central.**

Seleccionar la opción Insertar PC.

Introducir los datos necesarios.

**Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducir los datos correcto y presiona el botón "Insertar".		El sistema debe de mostrar una interfaz con los datos modificados. Mostrar un mensaje "se insertó correctamente una PC".	Muestra una interfaz con los datos del la PC. Muestra un mensaje "se insertó correctamente una PC".
	Pulsar "Insertar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Insertar" y haber entrado datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite cualquier tipo de datos.

	Pulsar "Insertar" y haber entrado el mismo número de serie.	El sistema debe de mostrar un mensaje "Existe una PC con ese número de serie"	Permite que existan PC con el mismo número de serie.
	Pulsar "Insertar" y haber entrado un mismo número de inventario.	El sistema debe de mostrar un mensaje "Existe una PC con ese número de inventario".	Permite la entrada de PC con el mismo número de inventario.

## SC2.Editar PC.

- ✓ **Flujo Central.**  
 Seleccionar la opción Editar.  
 Modificar los datos deseados.
- ✓ **Condiciones de ejecución.**  
 Exista un medio PC.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Modificar los datos correctamente y presiona el botón "Editar".		El sistema debe de mostrar una interfaz con los datos modificados. Mostrar un mensaje "se modificó correctamente una PC".	Muestra una interfaz con los datos de una PC ya modificados. Muestra un mensaje "se modificó correctamente una PC".
	Pulsar "Editar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.

	Pulsar "Editar" y haber entrado los datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite cualquier tipo de datos.
	Pulsar "Editar" y haber entrado el mismo número de serie.	El sistema debe de mostrar un mensaje "Existe una PC con ese número de serie"	Permite que existan PC con el mismo número de serie.
	Pulsar "Editar" y haber entrado un mismo número de inventario.	El sistema debe de mostrar un mensaje "Existe una PC con ese número de inventario".	Permite la entrada de PC con el mismo número de inventario.

### SC3.Eliminar PC.

- ✓ **Flujo Central.**  
Selecciona la opción Eliminar.
- ✓ **Condiciones de Ejecución.**  
Que exista un medio PC.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Presionar el botón "Eliminar".		El sistema debe de eliminar los datos de la PC.	Elimina los datos de la PC creada y todo lo que está relacionado a esta.

### Caso de Prueba Gestionar Monitor.

- ✓ **Breve Descripción.**  
Este caso de prueba permite comprobar las funcionalidades de modificar, eliminar e insertar al sistema.
- ✓ **Flujo Central.**  
Seleccionar la opción Monitor.
- ✓ **Condiciones de Ejecución.**

El Responsable de activos fijos debe haberse autenticado.

**EC1.Insertar Monitor.**

**Flujo Central.**

Seleccionar la opción Insertar Monitor.

Introducir los datos necesarios.

**Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducir los datos correcto y presiona el botón "Insertar".		El sistema debe de mostrar una interfaz con los datos modificados. Mostrar un mensaje "se insertó correctamente una monitor".	Muestra una interfaz con los datos de una monitor ya modificados. Muestra un mensaje "se insertó correctamente una monitor".
	Pulsar "Insertar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Insertar" y haber entrado datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite cualquier tipo de datos.
	Pulsar "Insertar" y haber entrado un mismo número de inventario.	El sistema debe de mostrar un mensaje "Existe un monitor con ese número de inventario".	Permite la entrada de un monitor con el mismo número de inventario.

**SC2.Editar Monitor.**

- ✓ **Flujo Central.**  
 Seleccionar la opción Editar.  
 Modificar los datos deseados.
- ✓ **Condiciones de ejecución.**  
 Exista un medio Monitor.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Modificar los datos correctamente y presiona el botón "Editar".		El sistema debe de mostrar una interfaz con los datos modificados. Mostrar un mensaje "se modificó correctamente un monitor".	Muestra una interfaz con los datos de un monitor ya modificados. Muestra un mensaje "se modificó correctamente un monitor".
	Pulsar "Editar" sin haber llenado todos los campos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Editar" y haber entrado los datos incorrectos.	El sistema debe mostrar un mensaje por favor, entre bien los datos.	Permite la entrada de cualquier tipo de datos.
	Pulsar "Editar" y haber entrado un mismo número de inventario.	El sistema debe de mostrar un mensaje "Existe un monitor con ese número de inventario".	Permite la entrada de un monitor con el mismo número de inventario.

**SC3.Eliminar Monitor.**

- ✓ **Flujo Central.**  
Selecciona la opción Eliminar.
- ✓ **Condiciones de Ejecución.**  
Que exista un medio Monitor.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Presionar el botón "Eliminar".		El sistema debe de eliminar los datos del monitor.	Elimina los datos el monitor.

**Caso de Prueba Realizar Movimiento.**

- ✓ **Breve Descripción.**  
Este caso de prueba permite comprobar las funcionalidades realizar un movimiento.
- ✓ **Flujo Central.**  
Seleccionar la opción Realizar movimiento.
- ✓ **Condiciones de Ejecución.**  
El Responsable de activos fijos debe haberse autenticado.  
Existir algún medio básico registrado.
- ✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Introducirlos datos correcto, presionar el botón "Guardar" e imprimir.		El sistema guarda los datos en la base de datos e imprime el modelo.	Guarda los datos en la base de datos e imprime el modelo.

	Pulsar "Guardar" sin haber entrado los datos.	El sistema debe mostrar un mensaje por favor, rellene este campo.	Muestra el mensaje.
	Pulsar "Guardar" haber definido la ubicación.	El sistema debe de actualizar la ubicación de los medios básicos en la base de datos.	No actualiza en la base de datos.

### Caso de Prueba "Realizar Reportes".

✓ **Breve Descripción.**

Este caso de prueba permite comprobar las funcionalidades de realizar un reporte.

✓ **Flujo Central.**

Seleccionar la opción Realizar Reportes.

✓ **Condiciones de Ejecución.**

El Responsable de activos fijos debe haberse autenticado.

Existir algún medio básico registrado y alguna PC.

✓ **Iteraciones.**

Clases válidas	Clases inválidas	Resultado esperado	Resultado obtenido
Selecciona la opción "préstamo".		Mostrar un listado con los medios básicos en préstamo.	El sistema muestra una interfaz con los medios básicos en préstamo.
Seleccionar la opción "reparación".		Mostrar un listado con los medios básicos en reparación.	El sistema muestra una interfaz con los medios básicos en reparación.

	Selecciona la opción "préstamo" y pulsa "buscar".	Mostrar un mensaje "no se encuentra ningún medio básico en reparación".	Muestra el mensaje.
	Seleccionar la opción "reparación" y pulsar "buscar".	Mostrar un mensaje "no se encuentra ningún medio básico en préstamo".	Muestra el mensaje.