

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 4



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

**Sistema para la informatización del proceso de  
confirmación metrológica del Departamento de  
Calidad en Cubana de Aviación**

**Autora:** Yusleivy Robles Velázquez

**Tutora:** Ing. Lisset Salazar Gómez

**Cotutor:** Ing. Ángel Alberto Vázquez Sánchez

**La Habana, junio de 2014**

“Año 56 de la Revolución”

# Pensamiento

---

*No dejéis que el ruido de las opiniones ajenas silencie vuestra propia voz interior.*

*Steve Jobs.*

# Declaración de autoría

Declaro ser autora de la presente tesis y reconozco 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 \_\_\_\_.

\_\_\_\_\_

Firma del Autor

Yusleivy Robles Velázquez

\_\_\_\_\_

Firma del Tutor

Ing. Lisset Salazar Gómez

\_\_\_\_\_

Firma del Cotutor

Ing. Ángel Alberto Vázquez Sánchez

# Dedicatoria

---

*A mi hermano, amigo y compañero Lionnis, porque la vida no le permitió llegar hasta aquí, porque este era nuestro pacto y aquí lo cumplo por los dos.*

*A mis padres Enoelia y Alejandro por ser el motor y timones de este barco. Son los culpables que yo haya podido nutrirme de estos cinco años de experiencia.*

*Mi mamá además de darme la vida, me da la fuerza para seguir la luz que alumbra al final del camino.*

*Mi papá me da la fuerza, la madurez y el carácter para luchar y lograr mis metas.*

Actualmente el desarrollo empresarial, va acompañado de sistemas encargados de la gestión de la información, dando paso a la inmediatez que requiere la toma de decisiones en estos tiempos. La Empresa Cubana de Aviación se encuentra inmersa en la construcción de un sistema que contribuya en la informatización del proceso metrológico. En la presente investigación se abordan los elementos esenciales de los flujos de trabajo: requisitos, diseño, implementación y pruebas correspondientes a trabajo de diploma. Además se realiza un estudio del estado del arte de la propuesta de solución y se utiliza el conjunto de herramientas y tecnologías definidas.

La investigación fue guiada por la metodología tradicional RUP y arrojó como principales resultados la documentación y los artefactos de ingeniería de software generados durante el desarrollo del sistema.

**Palabras claves:** confirmación metrológica, gestión de la información.

## Índice

Introducción	1
Capítulo 1. Aspectos teóricos que fundamentan la investigación	5
1.1 Introducción	5
1.2 Conceptos asociados al dominio del sistema	5
1.3 Tendencias actuales de los sistemas de gestión	6
1.3.1 Sistemas informáticos vinculados al campo de acción	7
1.4 Descripción del proceso de confirmación metrológica	8
1.5 Estudio de metodologías, herramientas y tecnologías	9
1.5.1 Metodología de desarrollo de software	10
1.5.2 Lenguaje de Modelado	12
1.5.3 Lenguajes de Desarrollo	13
1.5.4 Framework de desarrollo	17
1.5.5 Entorno de Desarrollo Integrado	22
1.5.6 Gestor de Base de Datos	23
Conclusiones del capítulo	24
Capítulo 2. Características del sistema	26
2.1 Introducción	26
2.2 Modelo del proceso de Negocio	26
2.2.1 Actores del negocio	26
2.2.2 Trabajadores del negocio	27

2.2.3 Proceso del Negocio	27
2.2.4 Diagrama de casos de usos del Negocio	27
2.2.4 Especificación de los casos de usos del Negocio	28
2.3 Definición de los requisitos funcionales	30
2.3.1 Definición de los requisitos no funcionales	33
2.4. Actores del sistema	35
2.4.1 Caso de Uso del sistema	36
2.4.2 Diagrama de CU del sistema	36
2.4.3 Descripción de los CU del sistema	37
Conclusiones del capítulo	41
Capítulo 3: Arquitectura y diseño	42
3.1 Introducción	42
3.2 Arquitectura del sistema	42
3.3 Patrones de diseño	43
3.4 Diagrama de clases del Diseño	45
3.5 Diagrama Entidad-Relación	46
3.6 Diagrama de despliegue	48
Conclusiones del capítulo	49
Capítulo 4. Implementación y pruebas del sistema	50
4.1 Introducción	50
4.2 Modelo de implementación	50

<b>4.2.1 Descripción de los componentes</b>	<b>51</b>
<b>4.3 Estándares de codificación</b>	<b>53</b>
<b>4.4 Pruebas del sistema</b>	<b>54</b>
<b>Estrategia de prueba</b>	<b>54</b>
<b>Conclusiones del capítulo</b>	<b>59</b>
<b>Conclusiones Generales</b>	<b>60</b>
<b>Recomendaciones</b>	¡Error! Marcador no definido.
<b>Glosario de Términos</b>	¡Error! Marcador no definido.
<b>Bibliografía</b>	<b>62</b>
<b>Anexos</b>	¡Error! Marcador no definido.
<b>Anexo1 “Entrevista”</b>	<b>66</b>
<b>Anexo2 “Diagrama de Actividades del Negocio”</b>	¡Error! Marcador no definido.
<b>Anexo3 “Descripción de Casos de Usos del sistema”</b>	<b>68</b>
<b>Anexo4 “Diagramas de clases del diseño”</b>	<b>89</b>
<b>Anexo5 “Descripción de las tablas de la Base de Datos”</b>	<b>93</b>

## Introducción

La Empresa Cubana de Aviación S.A. con su sede central en el Aeropuerto Internacional “José Martí”, es la línea aérea insigne de Cuba. Con una amplia experiencia acumulada a lo largo de sus 85 años de existencia (fue fundada el 8 de octubre de 1929). Figura entre las primeras aerolíneas que a nivel mundial abrieron el paso a la era de los vuelos comerciales.

Cubana de Aviación se ha caracterizado por el continuo perfeccionamiento de sus servicios (transporte de pasajeros, carga, correo, un servicio de vuelos privados) al mismo nivel de los estándares internacionales, distinguiéndose por la atención al cliente, la seguridad y la puntualidad. La visión de la empresa es ubicar la aerolínea en un modelo de excelencia y eficiencia, alcanzando así la rentabilidad de sus operaciones. Entre los elementos que abarca esa visión, sobresale la seguridad de los vuelos, que en términos generales es garantizada por el buen funcionamiento de los equipos de medición.

Los equipos de medición<sup>1</sup> requieren de un proceso de confirmación metrológica<sup>2</sup> que incluye calibración y/o verificación y el control de calidad de los mismos; mediante el cual la empresa Cubana de Aviación garantiza las buenas condiciones metrológicas de los equipos. Esta confirmación determina las características metrológicas del equipo, las cuales pueden influir sobre los resultados de la medición. (1)

Para la empresa, la confirmación metrológica de un equipo de medición es una condición indispensable para su uso. Es realizada por el técnico<sup>3</sup>, tras aceptar el resultado de la verificación o calibración del equipo en cuestión. La evidencia de la condición del equipo está dada por el certificado que se emite una vez recibido uno de estos dos servicios de aseguramiento, o en su defecto la Declaración de Conformidad del fabricante o ente autorizado por este. La verificación<sup>4</sup> es obligatoria para los equipos de medición declarados con carácter legal por la Norma DG-01 la

---

<sup>1</sup> Instrumento de medición, software, patrón de medida, material de referencia o aparato auxiliar, o una combinación de éstos, necesario para llevar a cabo un proceso de medición. (ISO 10012:07)

<sup>2</sup> Conjunto de operaciones requeridas para asegurarse de que el equipo de medición es conforme a los requisitos correspondientes a su uso previsto; incluye la reparación, ajuste, calibración y verificación. (ISO 10012)

<sup>3</sup> El técnico es la persona responsable de todo lo referente a un equipo de medición determinado.

<sup>4</sup> Procedimiento que incluye el examen y marcado y/o emisión de un certificado de verificación, que constata y confirma que el instrumento de medición cumple con los requisitos reglamentarios. (VIM)

# Introducción

---

cual indica la relación de instrumentos a los que el estado les realiza la verificación. La calibración<sup>5</sup>, es ejecutada periódicamente a todos los equipos de medición que se dispongan, con interés de ratificar por medio de un ente especializado su buen estado metrológico. En algunos equipos como los especiales de aviación, la calibración se complementa con un control de calidad, para así corroborar el cumplimiento de parámetros de seguridad preestablecidos. Ver entrevistas realizadas al Jefe de Calidad de la empresa en ([Anexo1](#))

Actualmente en el Departamento de Calidad de la Empresa Cubana de Aviación no se cuenta con una gestión informatizada de la metrología; mediante la cual se pueda brindar una información con la inmediatez que requiere la toma de decisiones empresariales en la actualidad. Además que consolide y aumente la veracidad y uniformidad de la información, permitiendo el registro de datos que posibilitan el análisis estadístico. Esto genera, demora en los trámites del proceso, documentación excesiva y desconocimiento en el tiempo de estancia y del paradero de los equipos debido a la existencia de tantos laboratorios.

Lo anteriormente expuesto, constituye la situación problemática identificada que fundamentó la investigación, cuyo **problema a resolver** radica en: ¿Cómo contribuir a la gestión de la información en el proceso de confirmación metrológica en el departamento de Calidad en la empresa Cubana de Aviación?

Se define como **objetivo general** desarrollar un sistema de gestión para controlar los aspectos metrológicos de interés de la empresa. El **objeto de estudio** es el proceso de confirmación metrológica. Centrándose en el **campo de acción** proceso de confirmación metrológica en el Departamento de Calidad de la Empresa Cubana de Aviación.

Como **idea a defender**: La implementación de un sistema de gestión de la información metrológica facilitará el control de las diferentes fases del proceso metrológico garantizando la inmediatez de la información en la toma de decisiones.

## **Objetivos específicos:**

---

<sup>5</sup> *Conjunto de operaciones realizadas sobre un sistema de medida para que proporcione indicaciones prescritas, correspondientes a valores dados de la magnitud a medir.*

1. Elaborar el marco teórico conceptual a partir del estudio de las principales tendencias, metodologías y herramientas existentes relacionadas con el proceso de confirmación metrológica.
2. Definir los requisitos funcionales y no funcionales que permitan mejorar la gestión de la información del proceso de confirmación metrológica.
3. Diseñar e implementar un sistema para la gestión del proceso de confirmación metrológica.
4. Validar el sistema desarrollado mediante pruebas.

Para poder implementar correctamente el sistema se plantean las **tareas** siguientes:

- ❖ Sistematización de los referentes teóricos y metodológicos que sustentan la implementación del sistema.
- ❖ Estudio de los sistemas de gestión existentes para el control de los procesos de confirmación metrológica.
- ❖ Seleccionar la metodología de desarrollo de software, lenguaje de programación y herramientas tecnológicas que se utilizarán en el desarrollo de la aplicación.
- ❖ Diagnóstico del estado actual del proceso de confirmación metrológico.
- ❖ Identificar los requisitos funcionales y no funcionales del sistema.
- ❖ Diseñar el sistema haciendo uso de los diagramas pertinentes y modelos definidos por la metodología de desarrollo seleccionada.
- ❖ Fundamentación de la propuesta a partir de los requisitos definidos con el cliente.
- ❖ Diseñar la base de datos del software.
- ❖ Implementar las clases del diseño e interfaces del sistema.
- ❖ Realizar los casos de prueba de aceptación.
- ❖ Probar el sistema y corregir las no conformidades arrojadas.

Los métodos científicos fueron fundamentales en el desarrollo de la investigación puesto que contribuyeron a caracterizar el objeto de estudio (2). A continuación se explica cómo fueron aplicados:

## **Métodos teóricos:**

**Analítico-Sintético:** Se evidencia cuando se realiza un análisis de toda la teoría y documentación, que permiten la extracción de los elementos fundamentales relacionados con el proceso de confirmación.

**Histórico-Lógico:** Utilizado al tener en cuenta la caracterización de la evolución histórica de los sistemas de gestión de la información para concebir el sistema actual.

**Modelación:** Se emplea como forma de representación de todos los datos esenciales relacionados con el campo de acción.

## **Métodos empíricos:**

**Observación:** Fue utilizado en diferentes momentos de la investigación para agrupar elementos relacionados con la problemática en cuestión.

El documento de tesis está estructurado por cuatro capítulos de la siguiente forma:

**Capítulo 1. Aspectos teóricos que fundamentan la investigación:** Aborda los conceptos relacionados con el objeto de estudio y justifica la selección de las tecnologías, metodologías y herramientas que serán utilizadas en el desarrollo del sistema.

**Capítulo 2. Características del sistema:** Explica detalladamente el proceso de captura y refinamiento de los requisitos del software.

**Capítulo 3. Arquitectura y diseño del sistema:** Mediante los diagramas generados por la metodología de desarrollo utilizada se describe la estructura interna del sistema a través de artefactos.

**Capítulo 4. Implementación y pruebas del sistema:** A partir de la documentación generada en el capítulo anterior se documenta el proceso de implementación y las estrategias de prueba a utilizar.

## Capítulo 1. Aspectos teóricos que fundamentan la investigación

### 1.1 Introducción

Durante el desarrollo de este capítulo se expone el marco teórico en el que se desarrolla la investigación, identificando la terminología que se utilizará a lo largo del desarrollo del sistema. Por otra parte se realiza un análisis de soluciones similares con el objetivo de identificar mejoras y funcionalidades comunes en este tipo de aplicaciones. Se realiza un estudio de las metodologías de desarrollo de software y herramienta lo que sirve como entrada al análisis de los procesos que se desea automatizar.

### 1.2 Conceptos asociados al dominio del sistema

**En la investigación del trabajo se hace uso de conceptos asociados al dominio del sistema, como son:**

**Metrología:** "Ciencia de la medida y su aplicación". (VIM)

**Medición:** Instrumento de medición, software, patrón de medida, material de referencia o aparato auxiliar, o una combinación de éstos, necesario para llevar a cabo un proceso de medición. (ISO 10012:07)

**Confirmación Metrológica:** Conjunto de operaciones requeridas para asegurarse de que el equipo de medición es conforme a los requisitos correspondientes a su uso previsto; incluye la reparación, ajuste, calibración y verificación. (ISO 10012)

**Verificación:** Procedimiento que incluye el examen y marcado y/o emisión de un certificado de verificación, que constata y confirma que el instrumento de medición cumple con los requisitos reglamentarios. (VIM)

**Calibración:** Conjunto de operaciones realizadas sobre un sistema de medida para que proporcione indicaciones prescritas, correspondientes a valores dados de la magnitud a medir. (VIM).

Estos conceptos asociados al dominio del sistema, son los utilizados por la empresa que hará uso del sistema. Luego del análisis de las terminologías establecidas por el VIM (Vocabulario Internacional de Metrología) (3) y la ISO 10012 (Norma Internacional. Sistemas de gestión de las mediciones) (4). Se puede decir que en el trabajo se hace uso del término confirmación metrológica ya que este aporta una descripción detallada del proceso a informatizar.

## 1.3 Tendencias actuales de los sistemas de gestión

Las empresas que operan en el siglo XXI se enfrentan a muchos retos significativos, como son: la rentabilidad, competitividad, globalización, capacidad de adaptación, tecnología y otros. Por lo que los sistemas de gestión juegan un gran papel para el crecimiento de las mismas, pues ayudan mediante una serie de estrategias como la optimización de procesos y el pensamiento disciplinario lograr los objetivos de la empresa. (5)

La implementación de un sistema de gestión puede ayudar a mejorar la efectividad operativa, reducir costos, aumentar la satisfacción de clientes y partes interesadas y lograr mejoras continuas. (5)

La gestión de la información es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma.” (6)

La gestión de la información se establece como una disciplina transversal que aparece entrelazada en todas las diferentes capas de una organización, en todos los conceptos de administración como recursos humanos, marketing, finanzas, estrategia, operaciones, entre otros. (6)

Un sistema de gestión ayuda a lograr los objetivos de una organización mediante una serie de estrategias que incluyen la optimización de procesos, el enfoque centrado en la gestión de la información, entre otros. (6)

La implementación de un sistema de gestión brinda ventajas como (6):

- Control efectivo de las actividades de la organización.
- Integración de nuevas tecnologías y herramientas.
- Disponibilidad de mayor y mejor información para los usuarios en tiempo real.
- Elimina la barrera de la distancia trabajando con un mismo sistema en puntos distantes.
- Disminuye errores, tiempo y recursos.

Pero también presentan desventajas como (6):

- El tiempo que pueda tomar su implementación.
- La resistencia al cambio de los usuarios.

- Los problemas técnicos como fallas de hardware y/o software.
- La inadecuada implementación de las funcionalidades para el apoyo de actividades de la organización.

Los sistemas de gestión de información permiten la comprensión de cada organización desde un enfoque analítico, evaluador y creativo que permite develar las oportunidades que merezcan ser explotadas y contrarrestar las amenazas, además de establecer los factores que resulten críticos y buscar nuevas oportunidades. (7)

Por todo lo planteado, se puede decir, que la gestión de la información es todo lo relacionado con la obtención de información con la inmediatez que requiere la toma de decisiones empresariales. Es un modo de controlar un proceso o una entidad, según el permiso de acceso en cualquier momento, con el menor costo.

### **1.3.1 Sistemas informáticos vinculados al campo de acción**

Debido al nivel de confidencialidad, seguridad, complejidad e importancia que caracteriza a los sistemas de gestión encargados del proceso de confirmación metrológica, en las empresas aéreas internacionales no es fácil encontrar un sistema que ejemplifique el proceso. La publicidad de los mismos sería un riesgo, puesto que violaría la privacidad de las empresas.

En la exploración del tema se encontraron sistemas de gestión de la información pertenecientes a aerolíneas internacionales como son: *Amadeus IT Group*, *Atennea Air*, *Worldspan* y *AMIS*. Los cuales no tienen que ver con el campo de acción pero proporcionaron una visión de la importancia que poseen dichos sistemas para las empresas.

**Amadeus IT Group** sistema de distribución global, centra su principal fuente de negocio en (8):

- El sistema de gestión de reservas el cual permite a la aerolínea comercializar su inventario (asientos/vuelos). Contiene la información de horarios y tarifas además de una base de datos.
- El sistema de inventario define los asientos en que vuelos se pueden vender y a quien. Con el Sistema de Inventario y facturación la aerolínea define sus reglas en función de criterios como rentabilidad.

**Atennea Air es el software de gestión de diferentes compañías aéreas. Está diseñado para integrar en una única solución todas las áreas de la compañía, reaprovechando la información generada por cada departamento, para que sea utilizada por el resto, en**

**función de sus necesidades y permisos. La compañía trabaja con el mismo nivel de actualización de la información, evitando errores por doble digitación, e incrementando la productividad del personal. (9)**

**Worldspan** es un sistema de distribución global de información para las agencias de viajes. Ofrece a sus clientes toda la información a través de e-mail, proporciona la ventaja de acceder a esta información en tiempo real en cualquier momento y en cualquier lugar, antes y durante el transcurso de su viaje.

**AMIS** (Sistema de Información de Gestión de la Aviación) es un sistema informático integrado muy potente para la gestión de las actividades de las operaciones técnicas de una compañía aérea o un operador de la flota de aviones. El sistema ha sido licenciado a nivel mundial para las grandes y las pequeñas empresas que operan muchos tipos de fijas y helicópteros. Se basa en la tecnología de sistemas abiertos y se ejecuta en micros a los marcos principales en los distintos sistemas de gestión de bases de datos relacionales y Sistema Operativo Unix / Linux. Debido a su arquitectura, la AMIS opera las 24 horas del día / 7 días a la semana sin tiempo de inactividad en absoluto. Las copias de seguridad se realizan sobre la marcha. El sistema abarca actualmente diez módulos. (10)

Luego del análisis de los sistemas de gestión encontrados, se considera tener en cuenta, algunas de las funcionalidades de los mismos. Debido a la importancia que aporta a dichos sistemas, estas son:

- Integrar en una única solución todas las áreas de la compañía.
- Distribuir la información a través de correo.
- Posibilitar el acceso a la información en cualquier momento.

Realizada la investigación se demuestra la necesidad de implementar un sistema que satisfaga la necesidad del país de informatizar dicho proceso y así garantizar la calidad de los equipos, siendo estos un elemento esencial en la seguridad de los vuelos.

#### **1.4 Descripción del proceso de confirmación metrológica**

El proceso de confirmación metrológica, garantiza un estricto control de calidad en el cumplimiento de los requisitos y la no utilización de equipos de medición que se encuentren funcionando por fuera de sus especificaciones paramétricas.

De acuerdo a la norma ISO: 10012 se define como “El conjunto de operaciones necesarias para asegurar que el equipo de medición cumple con los requisitos para su uso previsto”.

El proceso consta de tres etapas: (1)

## **Calibración:**

Es el procedimiento metrológico por medio del cual se compara un equipo de medición con un patrón de referencia determinando con suficiente exactitud el valor que posee dicho equipo. Este proceso es realizado por los laboratorios de calibración, quienes entregan un certificado de calibración donde reportan los errores o correcciones y la incertidumbre de medición del equipo.

## **Verificación Metrológica:**

Es el procedimiento mediante el cual se interpretan los resultados obtenidos en el certificado de la calibración para determinar si el equipo cumple o no cumple con los requisitos para los cuales la empresa lo tiene destinado. Las actividades que se deben desarrollar para la realización de este procedimiento son (1):

- ✓ **la evaluación de consistencia:** consta de un conjunto de actividades que permiten determinar las características metrológicas con las cuales debe cumplir un equipo de medición para que se encuentre en condiciones óptimas de funcionamiento.
- ✓ **la evaluación de conformidad:** determina si un instrumento se encuentra conforme o no a los requerimientos de medición establecidos por la evaluación de consistencia utilizando los datos (error e incertidumbre) proporcionados por el certificado de calibración.

## **Control de calidad:**

Son todas aquellas actividades que se deben realizar una vez que se conoce el resultado de la evaluación de conformidad del equipo de medición (1). Cuando dicho equipo no se encuentra conforme con las especificaciones establecidas, es necesario actuar sobre sus características por medio de los procedimientos de ajuste, reparación o mantenimiento. En caso que el equipo de medición se encuentre conforme con los requerimientos establecidos para su uso, las acciones que se deben realizar son: el cálculo del intervalo de calibración y la elaboración del informe de inspección.

## **1.5 Estudio de metodologías, herramientas y tecnologías**

Las metodologías, herramientas y tecnologías seleccionadas para el desarrollo de un sistema deben ser escogidas cautelosamente, ya que pueden suponer el fracaso de éste o pueden aumentar su complejidad, para lo cual se deben conocer cuáles son las distintas alternativas y las necesidades del cliente, para proceder a seleccionar las que se adapten conforme a estas necesidades.

## **1.5.1 Metodología de desarrollo de software**

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software. (11)

Estas tecnologías proporcionan una serie de elementos que permiten gestionar y controlar el sistema, definir las herramientas a utilizar, dividir su desarrollo en etapas y precisar qué hacer en cada una de ellas.

Las metodologías orientadas al control riguroso de los procesos se clasifican como pesadas, mientras que las orientadas a la interacción con el cliente y el desarrollo incremental del software son las denominadas ágiles (11) (12) (13). Debido a los intereses de la empresa a la que pertenece la solución que se implementa se decide el uso de la metodología tradicional RUP<sup>6</sup> para que la misma cuente con una documentación exhaustiva sobre el proceso de desarrollo.

### **Proceso Unificado de Racional**

RUP es un marco genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos (13). Esta metodología de desarrollo se ha hecho muy popular entre las tradicionales por el nivel de detalle de la documentación que genera sirviendo de base al equipo de trabajo para dar soporte a la solución implementada, además es adaptable a pequeños y grandes proyectos.

RUP divide el desarrollo del proyecto en fases y flujos de trabajo como se indica en la Figura 1 Fases y flujos de trabajo de RUP:

---

<sup>6</sup> Del inglés Rational Unified Process.

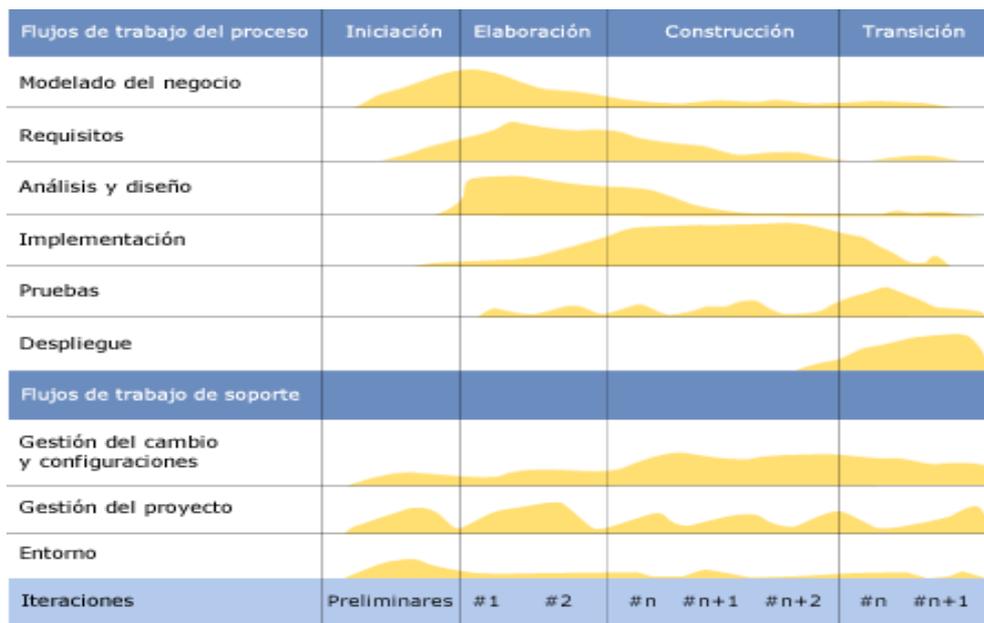


Figura 1 Fases y flujos de trabajo de RUP

Según (13) las principales características de esta metodología de desarrollo son las que a continuación se explican:

- Dirigido por casos de uso:

Un caso de uso es un segmento de funcionalidad que aporta un resultado concreto al usuario; constituye además una guía del proceso de desarrollo a través de sus fases y flujos (13). A grandes rasgos, identificar estos casos de uso es el punto de partida para el desarrollo del sistema y apoyándose en ellos se crean una serie de modelos de diseño e implementación que dan soporte a la solución.

El software en cuestión engloba gran cantidad de funcionalidades. Para llevarlas a cabo, fueron identificados y detalladamente documentados los casos de uso, además, los modelos obtenidos representan la realización de los mismos y son el resultado de algunos de los flujos de trabajo.

- Centrado en la arquitectura:

La arquitectura de un sistema software se describe mediante diferentes vistas del sistema en construcción. El concepto de arquitectura de software incluye los aspectos estáticos y dinámicos más significativos del sistema (13). Los casos de uso implementados fueron agrupados de acuerdo al impacto que tenían sobre la concepción del producto y en la arquitectura del mismo. Esta

organización del sistema que se logra a través de la arquitectura permitió tener visión general del producto final para comprenderlo y desarrollarlo en tiempo y forma.

- Iterativo e incremental:

Esta característica de la metodología fue primordial para arribar a un resultado concreto debido a que se dividió el desarrollo del sistema en partes más pequeñas para implementar en cada iteración el conjunto de casos de uso que amplían la funcionalidad, disminuyendo así el impacto de los riesgos más significativos. Desarrollar por iteraciones permitió que tras cada una de ellas el equipo de trabajo pudiera ver resultados a corto plazo y hacer una planificación más realista de la iteración siguiente (13).

La elección de esta metodología robusta está basada en que cumple con una serie de normas establecidas por el cliente como son: la interacción formal con el mismo, la elaboración de artefactos específicos (diagrama de clases de diseño, diagrama de componentes, entre otros), la definición de los requisitos a implementar en el sistema. La metodología RUP, en ocasiones es considerada ágil, debido al especial énfasis en cuanto a su adaptación a las condiciones del negocio. Por lo que solo es generado los artefactos necesarios para la elaboración.

## 1.5.2 Lenguaje de Modelado

Como lenguaje de modelado RUP propone UML<sup>7</sup>. El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan (14). El uso de esta tecnología permitió especificar, visualizar, construir y documentar los artefactos generados durante cada flujo de trabajo. Es importante señalar que como lenguaje de modelado, UML indica qué debe hacerse y no cómo debe hacerse, por lo que es un error conceptual confundirlo con una metodología para desarrollar software.

## Herramienta CASE

Las herramientas de ingeniería de software asistida por ordenador o herramientas CASE<sup>8</sup> son elementos indispensables para concebir una aplicación informática. El uso de las mismas reduce los gastos y el tiempo en el ciclo de desarrollo del software ya que permiten modelar fácilmente

---

<sup>7</sup> Del inglés *Unified Modeling Language*.

<sup>8</sup> Del inglés *Computer Aided Software Engineering*.

procesos y sistemas, con previo dominio de al menos un lenguaje de modelado soportado por la herramienta. (15)

La Empresa Cubana de Aviación a la que pertenece la solución que se implementa utiliza la versión 8.0 de Visual Paradigm. Esta herramienta CASE proporciona abundantes tutoriales y demostraciones interactivas para la realización de proyectos bajo la metodología RUP, permitiendo modelar todos y cada uno de los diagramas que incorpora UML. (16) La principal razón por la que fue seleccionada esta herramienta fue por la necesidad del uso de tecnologías libres durante el desarrollo del sistema para gestionar el contenido de la información necesaria. Por otro lado es importante destacar que se integra con el entorno de desarrollo NetBeans7.0 que es el seleccionado para la codificación del software, además permite generar la documentación en diferentes formatos como: jpeg, .html, .pdf.

### 1.5.3 Lenguajes de Desarrollo

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático (17).

Partiendo de las características del software a desarrollar (aplicación web con soporte de datos), se hace necesaria la selección de un conjunto de lenguajes de desarrollo web que permitan cumplir con los requerimientos propuestos. Para ello se deben tener en cuenta un conjunto de variables que permitirán escoger el lenguaje de desarrollo idóneo para la implementación del software, como son: la familiaridad y conocimiento de las estructuras del lenguaje, soporte de los IDE<sup>9</sup> para el lenguaje escogido, así como el dominio de la herramienta escogida. La selección del lenguaje es una de las tareas más importantes realizadas por el equipo de trabajo durante el proceso de desarrollo del software porque influye en la decisión sobre el gestor de base de datos y sobre las herramientas para automatizar el proceso de desarrollo.

Estos lenguajes pueden dividirse en dos grandes grupos, los lenguajes del lado del cliente y los lenguajes del lado del servidor.

### Lenguajes del lado del cliente

Los lenguajes del lado del cliente son aquellos que basan su procesamiento en el cliente web, o sea, no son compilados o interpretados por el servidor, sino que son devueltos por salida estándar

---

<sup>9</sup> Del inglés *Integrated Development Environment*.

en el proceso de una petición cliente-servidor, para ser interpretados por el navegador del usuario. Los criterios para la selección de las versiones más recientes de los lenguajes del lado del cliente para el desarrollo de la aplicación, estuvieron basados principalmente en el grado de conocimiento y habilidades con dichos lenguajes, la cantidad de material de consulta disponible sobre los mismos y en la popularidad de estos para su uso en aplicaciones web a nivel mundial (18).

Esta decisión de utilizar versiones actualizadas se tomó para dar cumplimiento a los requerimientos necesarios y lograr una aplicación acorde a las últimas tendencias y uso de estándares en la web; aclarando que solo se utilizarán componentes, características y funcionalidades de los lenguajes compatibles e interpretados por las versiones de los navegadores recomendados para el uso de la aplicación, logrando de esta forma un comportamiento y visión estándar del software en dichos navegadores.

- **JavaScript** (19)

JavaScript maneja objetos dentro de la página web y sobre ese objeto se pueden definir diferentes eventos, dichos objetos facilitan la programación de páginas interactivas; 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.

- **HTML5**

HTML5 es la actualización de HTML, el lenguaje en el que es creada la web. Definido formalmente por un cuerpo de normas internacionales conocido como World Wide Web Consortium (W3C), HTML5 consta de más de cien especificaciones que se relacionan con la nueva generación de tecnologías web. De hecho, HTML5 es un término aglutinador que describe un conjunto de especificaciones de HTML, CSS y JavaScript diseñado para permitir que los desarrolladores creen la nueva generación de sitios y aplicaciones web. (20) Es una plataforma de código abierto desarrollado en términos de derechos de licencia libre. (21)

- **CSS3**

Las hojas de estilo en cascada o CSS<sup>10</sup> son un mecanismo sencillo para añadir estilo (por ejemplo, fuentes, colores, espacios) a los documentos Web (22). El W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o na-

---

<sup>10</sup> Del inglés *Cascading Style Sheets*.

veadores. En diferencia a CSS2, que fue una gran especificación que definía varias funcionalidades, CSS3 está dividida en varios documentos separados, llamados módulos. Cada módulo añade nuevas funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad.

Se decide utilizar CSS3 en la implementación ya que ofrece una gran variedad de nuevas formas de crear un impacto con los diseños, siendo además la mejor vía de separar los contenidos de la presentación; permite definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los textos. (22) Además mantiene el propósito del equipo de desarrollo de crear una aplicación que presente elementos de las tendencias actuales de diseño en la web.

## Lenguajes del lado del servidor

Los lenguajes del lado del servidor son aquellos que son procesados por el servidor y generan las páginas que son devueltas al cliente en cada petición. Además se encargan de manipular la información que persiste en la base de datos, la seguridad del sistema y en la mayoría de los casos realizan la lógica del negocio de la aplicación.

Para la selección del lenguaje del lado del servidor se realizó una comparación entre tres de los lenguajes más populares entre los desarrolladores web:

- **PHP5:** PHP<sup>11</sup> es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. (23)
- **Java:** Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros (24).
- **ASP.NET:** Este es una tecnología comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecno-

---

<sup>11</sup> Del inglés *PHP Hypertext Preprocessor*.

logía ASP, fue lanzado mediante una estrategia de mercado denominada .NET. El ASP.NET fue desarrollado para resolver las limitantes que tenía su antecesor ASP. Creado para desarrollar web sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Para publicar las páginas web necesita tener instalado IIS con el Framework .Net (1)(23).

Estos lenguajes gozan de gran popularidad entre los grupos de desarrollo de aplicaciones web de gestión. Inicialmente se comenzó a realizar el filtro de la selección teniendo en cuenta las características que debe tener el lenguaje escogido para que pueda resolver los problemas planteados. Entre estas características se encuentran la posibilidad de conexión con los gestores de base de datos libres más utilizados (versiones libres de PostgreSQL y MySQL, siendo este último un requisito no funcional de establecido por la empresa. Además debe incluir gran cantidad de funciones nativas que agilicen el proceso de desarrollo.

Luego de realizar un estudio sobre los diferentes lenguajes de programación escogidos para la comparación, se concluye que de una forma u otra los tres lenguajes de programación cuentan con las características fundamentales para el proceso de desarrollo del software, a pesar de la superioridad de unos sobre otros en algunos aspectos.

Como segundo paso de la comparación se tuvieron en cuenta criterios que podían incidir tanto en la calidad del producto final, como en el cumplimiento del cronograma. También fueron valoradas las necesidades del proceso de desarrollo y las capacidades del equipo de programación, aspectos que se muestran como un punto de vista relativo en la siguiente tabla:

Tabla 1.5.3 Comparación entre PHP5, JAVA, ASP.NET

Criterios	PHP5	JAVA	ASP.NET
Dominio del lenguaje y las tecnologías que engloba <sup>12</sup> .	Si	No	No
Referencia de desarrollos en el equipo de desarrollo de la empresa Cubana de Aviación.	Si	No	Si
Legibilidad del código.	Si	Si	Si
Adaptabilidad del lenguaje con la aplicación de integración.	Si	No	No

<sup>12</sup> Dígase configuración de servidor web o dominio de un framework de desarrollo para el lenguaje en cuestión.

Basado en la comparación anterior y teniendo en cuenta el dominio de habilidades se decidió utilizar como lenguaje del lado del servidor a PHP5, permitiendo poner en práctica los conocimientos adquiridos durante los tres últimos años de carrera.

## 1.5.4 Framework de desarrollo

En la actualidad es una tendencia en el desarrollo de software la utilización de frameworks (o marcos de trabajo). Utilizarlos acelera el proceso de desarrollo, reutilizar el código ya existente, promover buenas prácticas de desarrollo con el uso de patrones y mejorar la seguridad de la aplicación. Un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se pueden añadir las últimas piezas para construir una aplicación concreta. (25)

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (25) Un framework web, por tanto, se puede definir como un conjunto de componentes que forman parte de un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. (26)

Existen frameworks que permiten desarrollar diversos tipos de aplicaciones, desde videojuegos hasta aplicaciones médicas. Los frameworks se clasifican en dependencia de su principal funcionalidad, por tal motivo se decide que para el desarrollo de la aplicación debe utilizarse un framework para la interfaz de usuario o capa de presentación, uno para la modularización o capa de lógica de negocio y uno para la capa de acceso a datos.

### Capa de Presentación

Al ser JavaScript uno de los lenguajes del lado del cliente, se sobrentiende la utilización de un framework desarrollado para este lenguaje para la capa de presentación. Dando respuestas a las necesidades y requisitos de la aplicación se necesita seleccionar un framework que permita mostrar de forma dinámica los contenidos, aumentando la interactividad y experiencia del usuario.

Otros requisitos indispensables del marco de trabajo que se seleccione deben ser la manipulación del árbol de objetos dinámicos (DOM<sup>13</sup>), el manejo de eventos y animaciones fácilmente. Además

---

<sup>13</sup> Del inglés Document Object Model.

debe ser compatible con las principales versiones de los navegadores web más utilizados. Debe ser de código abierto, libre y gratis.

Teniendo en cuenta que los frameworks jQuery y Prototype presentan las características anteriores, se realizó una comparación entre ellos, teniendo como premisas las necesidades, competencias y condiciones del equipo de desarrollo. Esta comparación arrojó como resultado el framework jQuery en su versión 1.9 debido a las ventajas que proporciona la utilización del mismo para el desarrollo de la aplicación (27):

- Es fácil de usar y aprender, además de ser el framework JavaScript más conocido por el equipo de desarrollo de la aplicación ya que se cuenta con experiencia de sistemas anteriores implementados por el equipo de trabajo utilizando dicho framework.
- Permite manejar JSON<sup>14</sup>, XML<sup>15</sup> y CSS.
- Presenta soporte a extensiones o plugins, los cuales amplían las facilidades de uso y mejoran los tiempos de producción al permitir reutilizar componentes desarrollados por terceros.
- Existe una amplia documentación y comunidad de usuarios, aspectos estos muy importantes en la retroalimentación y rectificación de errores con el uso de esta herramienta.

Puede definirse a jQuery como una biblioteca de clases JavaScript rápida y concisa que simplifica el documento HTML: manejando eventos y animaciones para el desarrollo web rápido. jQuery está diseñado para cambiar la forma en que se escribe JavaScript. (28)

## Capa de lógica de negocio

Debido a que la selección del lenguaje del lado del servidor es PHP5, se necesita un framework desarrollado en este lenguaje y capaz de dar solución a las necesidades de la implementación del producto. En la actualidad existen gran cantidad de frameworks de desarrollo creados con PHP5 y que gozan de mucha popularidad a la hora de seleccionarlos para su uso en un proyecto determinado.

---

<sup>14</sup> Del inglés *JavaScript Object Notation*.

<sup>15</sup> Del inglés *eXtensible Markup Language*.

Entre estos frameworks puede citarse Symfony, CakePHP, Zend Framework y CodeIgniter. Para el desarrollo de la aplicación se seleccionó el framework Symfony en su versión 2.4.3 debido a las ventajas que representa su uso frente a los demás para resolver las necesidades y requisitos del proceso de desarrollo del software, además de contar con una amplia comunidad de usuarios, siendo esto importante para la retroalimentación, superación y reparación de errores.

## **Symfony**

Symfony es un framework PHP que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes como las configuraciones, validaciones, manejo de formularios, entre otros, dejando que el programador se dedique a crear las funcionalidades específicas de cada proyecto. Actualmente existe mucha información referente al trabajo con este framework, ya que están disponibles miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales. (29)

Junto a las características anteriores, la principal ventaja de la utilización de Symfony es la integración que permite entre la aplicación desarrollada y el sistema ya existente en la empresa del cual forma un apéndice, permitiendo adaptarse a la arquitectura de dicho software, ya que Symfony considera un proyecto como un conjunto de servicios y operaciones disponibles bajo un determinado nombre de dominio y que comparten el mismo modelo de objetos. (25)

Dentro del framework, las aplicaciones se agrupan de una forma lógica e independiente como un proyecto Symfony, permitiendo compartir el modelo de datos y las librerías. Al mismo tiempo están compuestas por módulos, los cuales representan una página web o un grupo de páginas con un propósito relacionado. Los módulos almacenan las acciones que representan a cada una de las posibles operaciones; trabajar con estas es muy similar a trabajar con páginas de una aplicación web tradicional. A estas acciones se les puede hacer corresponder una vista específica o pueden retornar la respuesta en formato de texto.

Dichas características proporcionan una estructura al código fuente que permite escribir un código más legible para un futuro mantenimiento. Además, el patrón de arquitectura modelo-vista-controlador implementado por dicho framework, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web, logrando desarrollar un producto bajo buenas prácticas de desarrollo de aplicaciones web actuales.

Otras necesidades a resolver por este framework en el proceso de implementación son: la capacidad de lograr una aplicación de software libre instalable en las plataformas Unix, Linux y Windows;

además de lograr una aplicación extensible, explotando al máximo el mecanismo de plugins. La aplicación desarrollada debe ser segura, y Symfony realiza gran parte del trabajo para lograr ese objetivo, protegiendo al sistema contra ataques de tipo XSS<sup>16</sup>, CSRF<sup>17</sup>, etcétera.

Los frameworks de ruta y formulario contenidos en Symfony ahorran gran parte del trabajo de implementación, logrando direcciones amigables y una correcta validación de los datos enviados por los usuarios. Además cuenta con un conjunto de líneas de comando que realizan tareas automatizadas ahorrando tiempo de implementación, por ejemplo, permite crear nuevos módulos perfectamente funcionales integrando interacción con la base de datos mediante CRUD<sup>18</sup> sin necesidad de crear código fuente.

La independencia de la base de datos utilizada, el soporte para la integración con un mapeador de objetos relacional (se logra a través de un potente framework de acceso a datos) y la integración con jQuery, son otras de las características que influyeron en la decisión de la utilización de dicho framework por parte del equipo de desarrollo.

Por último y no menos importante vale destacar el framework para pruebas con que cuenta Symfony, lo cual significa una potente herramienta utilizada por los desarrolladores, permitiendo encontrar errores y acortar el tiempo de desarrollo.

## Capa de acceso a datos

La utilización de un ORM<sup>19</sup> en el desarrollo de un software responde a la necesidad de tener una interfaz que traduzca la lógica de los objetos a la lógica relacional, o sea, poder acceder de forma efectiva a la base de datos desde un contexto orientado a objetos. En otras palabras, un ORM es una técnica de programación que permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es

---

<sup>16</sup> Del inglés *Cross Site Scripting*.

<sup>17</sup> Del inglés *Cross Site Request Forgery*.

<sup>18</sup> Del inglés *Create, Retrieve, Update y Delete*.

<sup>19</sup> Del inglés *Object Relation Mapper*.

decir, las tablas de la base de datos pasan a ser clases y los registros, objetos que podemos manejar con facilidad. (26)

Los ORMs se sitúan en la parte superior de la capa de abstracción de la base de datos, lo que permite abstraerse del gestor utilizado. Además resuelven un conjunto de tareas comunes y de mantenimiento en la aplicación como son (27):

- Reutilización: La reutilización permite invocar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.
- Encapsulamiento: La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.
- Portabilidad: Un ORM permite cambiar el gestor de base de datos en cualquier momento del desarrollo de la aplicación, sin que ocurran grandes cambios en la implementación. Esto se debe a que no se utilizan sintaxis propias del lenguaje de la base de datos actual, sino una sintaxis propia del ORM que es capaz de traducir a diferentes tipos de bases de datos.
- Seguridad: Los ORM suelen implementar mecanismos de seguridad que protegen la aplicación de los ataques más comunes como inyecciones SQL.
- Mantenimiento del código: Gracias a la correcta organización de la capa de datos, modificar y mantener el código del proyecto es una tarea sencilla.

Por tal motivo es necesaria la utilización de un ORM para el lenguaje PHP5 y que sea soportado por el framework Symfony 2.4.3.

Este último presenta soporte e integración con Propel y Doctrine (25), ambos con características similares y que resuelven las necesidades existentes en la aplicación ya expresadas anteriormente. Luego entonces, se decide Doctrine como ORM debido a que es el utilizado en el proyecto con el que se integra el presente trabajo, lo cual permitirá reutilizar el modelo de datos generado junto con sus funcionalidades. Además de ser rápido y eficaz, posee una amplia documentación y presenta soporte para migraciones. Doctrine implementa los métodos mágicos que facilitan aún más las consultas a la base de datos, permitiendo obtener información a partir de cualquier campo de una tabla.

Doctrine es un ORM para PHP 5.2.3 y posterior basado en YAML<sup>20</sup>. Además de todas las ventajas que conlleva un ORM, uno de sus puntos fuertes es su lenguaje DQL<sup>21</sup> inspirado en el HQL<sup>22</sup> de Hibernate. Soporta validación de datos en los modelos y relaciones entre ellos. (27)1.5.5 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado (IDE) es un programa compuesto por una serie de herramientas, tales como editor de texto, compilador, intérprete, depurador, sistema de ayuda para la construcción de interfaces gráficas de usuario, etc., que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. (28)

La selección de un IDE para la codificación, debe realizarse de forma que cumpla con las características que se listan a continuación:

- Debe tener soporte y completamiento de código para los lenguajes definidos para la implementación del software.
- Debe tener soporte, integración y completamiento de código para los frameworks que se usarán para el desarrollo del software.
- Debe permitir crear nuevos proyectos y aplicaciones directamente desde el IDE.
- Debe tener integración con sistemas de control de versiones como SVN<sup>23</sup>.
- Los requerimientos de hardware deben estar acorde a las condiciones de desarrollo presentes.

Los entornos de desarrollo analizados tras estas consideraciones fueron:

- Eclipse 3.5.1: Es un IDE de código abierto independiente de la plataforma, es una aplicación de cliente enriquecido, emplea plugins para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Eclipse es, en el fondo, únicamente un armazón sobre el que se

---

<sup>20</sup> Del inglés *YAML Ain't Markup Language*.

<sup>21</sup> Del inglés *Doctrine Query Language*.

<sup>22</sup> Del inglés *Hibernate Query Language*.

<sup>23</sup> Abreviatura de *Subversion*.

pueden montar herramientas de desarrollo. La arquitectura de plugins permite, además de integrar diversos lenguajes, introducir otras aplicaciones. (29)

- NetBeans IDE 7.0: En un entorno de desarrollo integrado libre y de código abierto para desarrolladores de software. Contiene todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java, así como con C/C++, PHP, JavaScript y Groovy. Entre sus características se encuentra un sistema de proyectos basado en control de versiones y refactorización. Todas las funciones del IDE son provistas por plugins, al igual que Eclipse.

Finalmente se decide seleccionar a NetBeans IDE 7.0 como entorno de desarrollo integrado para utilizar durante la implementación de la aplicación. Esta selección está respaldada por los siguientes criterios: cumple con las características requeridas, de ser este IDE el utilizado en el desarrollo del sistema del cual la propuesta desarrollada va a formar parte; además el equipo de desarrollo está familiarizado y tiene experiencia en el trabajo con el mismo.

NetBeans IDE 7.0 se destaca por brindar soporte nativo a frameworks como jQuery y Symfony (tanto para el completamiento de código como para las tareas desarrolladas mediante líneas de comando), los cuales fueron previamente seleccionados para el desarrollo de la aplicación; por otro lado brinda completamiento de código para los métodos mágicos que usa el ORM Doctrine.

## **1.5.6 Gestor de Base de Datos**

Un sistema gestor de base de datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Entre las características que debe presentar un SGBD se encuentra: abstracción de la información ahorrando detalles del almacenamiento físico de los datos al usuario, independencia de las aplicaciones que se sirvan de sus datos, redundancia mínima evitando repetir información, consistencia, o sea, que todos los datos repetidos se actualicen de forma simultánea, seguridad frente a usuarios malintencionados o ataques que intenten destruir o acceder a la información, integridad al garantizar la validez de los datos almacenados, respaldo y recuperación, además de controlar el acceso concurrente a la información, que podría derivar en inconsistencias.

Para esta herramienta no se realizó ninguna comparación entre los diferentes SGBD disponibles debido a que la aplicación es un sistema de gestión de la información para un software en desarrollo y por tanto debe trabajar sobre el gestor de base de datos definido en el documento de arquitectura de software de la empresa: MySQL en su versión 5.6.12.

MySQL es un SGBD relacional, licenciado bajo la GPL de GNU. Su diseño multihilos le permite manipular bases de datos de gran tamaño. Fue creada por la empresa sueca MySQL AB, que mantiene el derecho de autor del código fuente del servidor SQL, así como también de la marca. Aunque es un software libre, MySQLAB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece y la posibilidad de integrar este gestor en un software propietario, de no ser así se vulneraría la licencia GPL.

Esta decisión no estuvo acompañada de ninguna comparación previa debido a que su utilización no es decisiva para el desarrollo del software, además existía conocimiento por parte del equipo de desarrollo sobre el trabajo con la herramienta.

## **Servidor web**

Se decidió escoger como servidor web a Apache 2.0 ya que es el servidor web con licencia de software libre y de código. Apache 2.0 contiene un conjunto de características y funcionalidades indispensables para que la aplicación desarrollada pueda funcionar correctamente. Entre estas características se encuentran:

- Soporta PHP5.
- Es altamente configurable, rápido, flexible, eficiente y adaptado a los nuevos protocolos web.
- Su condición de ser multiplataforma permite mantener este servidor web en todos los sistemas operativos en los que se despliegue el software.
- Es un servidor seguro que permite protección de ficheros.

Vale destacar que cuenta con una gran comunidad de usuarios que facilita encontrar solución tanto a errores del propio servidor como a errores en la configuración del mismo.

## **Conclusiones del capítulo**

El estudio realizado en este capítulo ha permitido definir las herramientas y tecnologías a utilizar en el desarrollo de la propuesta de solución. Quedando seleccionada la metodología RUP para guiar el proceso de desarrollo, como lenguaje de modelado fue elegido UML y como herramienta

CASE para realizar el modelado del software se optó por Visual Paradigm 8.0. Tras determinarse que la propuesta de solución será una aplicación web, las tecnologías para su desarrollo se distinguen por su uso en el servidor y en el cliente. En el lado del servidor el lenguaje a utilizar será PHP5, este lenguaje será utilizado aprovechando las potencialidades que provee el framework de desarrollo Symfony 2.4.3. En el lado del cliente será utilizado HTML5 como lenguaje de maquetado, como lenguaje para proveer estilos visuales se empleará CSS3, este lenguaje será usado a través del framework de diseño Bootstrap, como lenguaje de scripting en el cliente será empleado JavaScript a través de la librería JQuery 1.9. En el acceso a la base de datos será utilizado el ORM Doctrine. Para el almacenamiento de la información se determinó a MySQL como gestor de bases de datos. El servidor web a utilizar será Apache 2.0 y por sus disímiles características que lo hacen un potente IDE fue seleccionado NetBeans 7.0 para desarrollar la propuesta de solución.

No se hizo fácil la obtención de sistemas similares que ayuden en el estudio para la informatización del proceso. De los sistemas similares encargados de la gestión de la información, se obtuvieron características a informatizar como: integrar en una única solución todas las áreas de la compañía.

## Capítulo 2. Características del sistema

### 2.1 Introducción

La captura de requisitos, la identificación de los casos de uso y la descripción de los mismos son tareas que tienen un impacto significativo en la caracterización del sistema que se desea implementar. Tomando como punto de partida el modelo de negocio, en este capítulo se relacionan detalladamente los requerimientos funcionales y no funcionales del software junto a las descripciones de los casos de uso del sistema

### 2.2 Modelo del proceso de Negocio

**Un proceso de negocio es una colección de actividades diseñadas para producir una salida específica para un cliente o mercado particular. Así, un proceso es un ordenamiento específico de actividades de trabajo a través del tiempo y del espacio, con un comienzo, un fin, entradas y salidas claramente identificados: una estructura para la acción. (30)**

#### 2.2.1 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor. (30)

A continuación se muestra la tabla con los actores del negocio y la descripción de los mismos:

**Tabla 1. Actores del Negocio**

Actor	Descripción
Técnico	Es la persona responsable de realizar todo lo referente a un equipo de medición determinado.
Jefe de Área	Es la persona que se encarga de controlar el cumplimiento del procedimiento metrológico del área.

## 2.2.2 Trabajadores del negocio

Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos. (30)

A continuación se muestran los trabajadores del negocio y su correspondiente descripción:

**Tabla 2. Trabajadores del Negocio**

Trabajador	Descripción
<b>Jefe de Calidad</b>	<b>Es la persona encargada de planificar, organizar y garantizar la implementación de las normas y regulaciones metrológicas.</b>

## 2.2.3 Proceso del Negocio

El proceso de negocio comienza, una vez que a uno de los técnicos del taller de la empresa se le presente la necesidad de controlar un equipo. El control del equipo inicia una vez que se crea un nuevo equipo y este es incluido a un área. El técnico es el encargado de verificar su certificación que puede estar emitida por un laboratorio o por el fabricante. En caso que se encuentre en norma, el equipo de medición es enviado para su uso, en otro caso es enviado al laboratorio de verificación o calibración.

El proceso puede comenzar también, con alguna de las acciones del jefe de área como: inventariar equipo, capacitar personal, realizar reportes. La acción, inventariar equipo empieza cuando el jefe de área verifica la lista de equipo y comprueba la existencia de los mismos. En caso se encuentre el equipo, crea el informe de inventario; en otro caso solicita al técnico el informe anterior. Para capacitar al personal, se detectan las necesidades de capacitación del personal del área y asigna el curso. Se realizan reportes de inventarios y capacitaciones, estos son generados con la información almacenada de los mismos.

## 2.2.4 Diagrama de casos de usos del Negocio

El diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y los actores que intervienen en el mismo, además de las relaciones que se establecen entre estos. (30)

A continuación se muestra la figura correspondiente al diagrama de casos de uso del negocio.

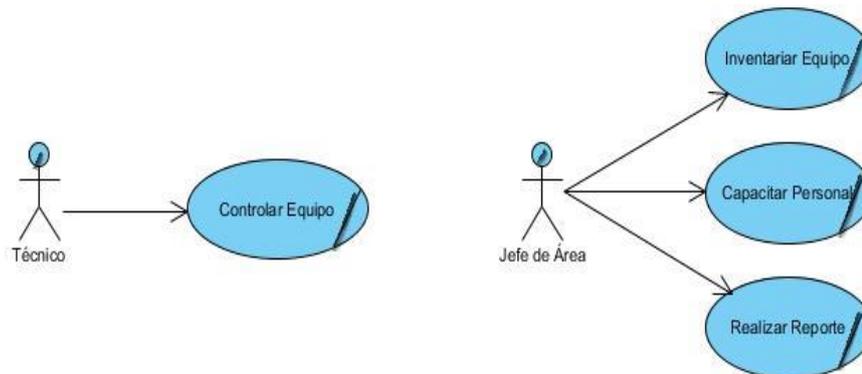


Figura 2.2.3 Diagrama de Caso de Uso del Negocio

## 2.2.4 Especificación de los casos de usos del Negocio

La especificación de los casos de uso del negocio, propicia una mayor comprensión de los procesos del negocio, tanto por parte del cliente como de los desarrolladores. Se describe cómo se llevan a cabo las actividades dentro del negocio y quiénes las realizan. Además de que se muestra el flujo de los procesos de manera gráfica, por medio de los diagramas de actividades, en los cuales, las actividades representan alguna acción que, el actor o los trabajadores realizan, y las entidades la información con las cuales estos interactúan. (30)

A continuación se muestran los diagramas de actividades de los casos de uso del negocio Inventariar Equipo y Controlar Equipo, el resto de los diagramas se pueden encontrar en el [Anexo2](#):

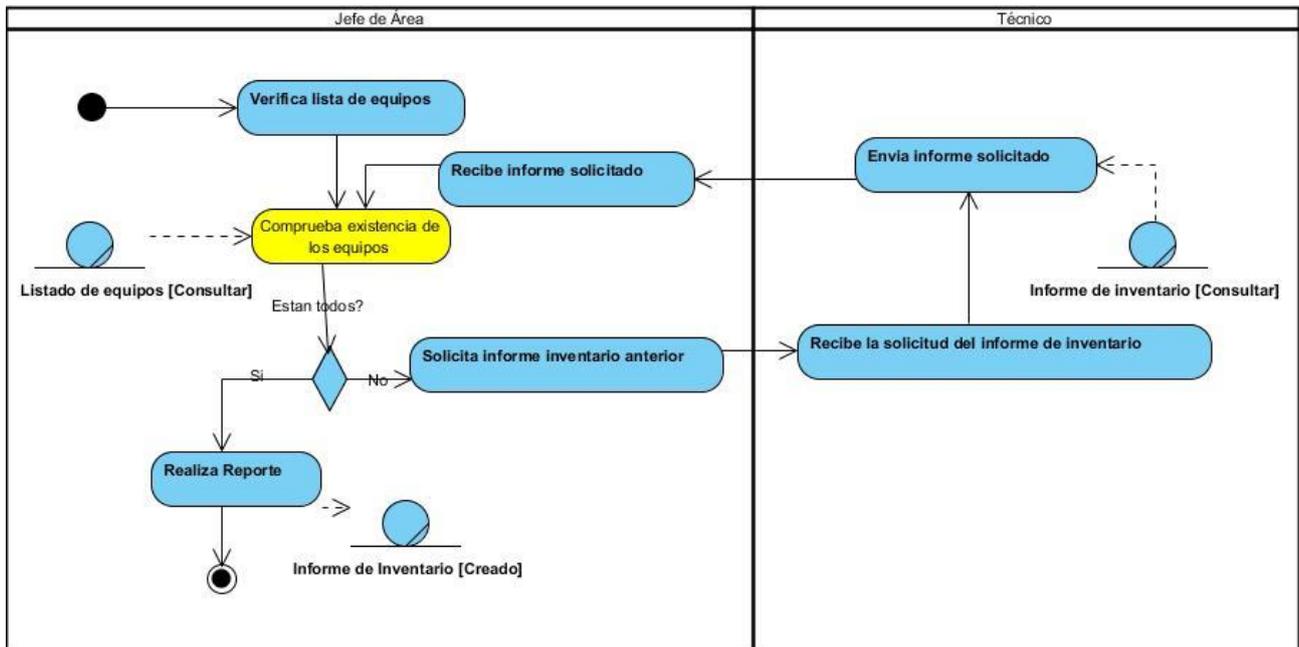


Figura 2.2.4 Diagrama de Actividades del CU "Inventariar equipo".

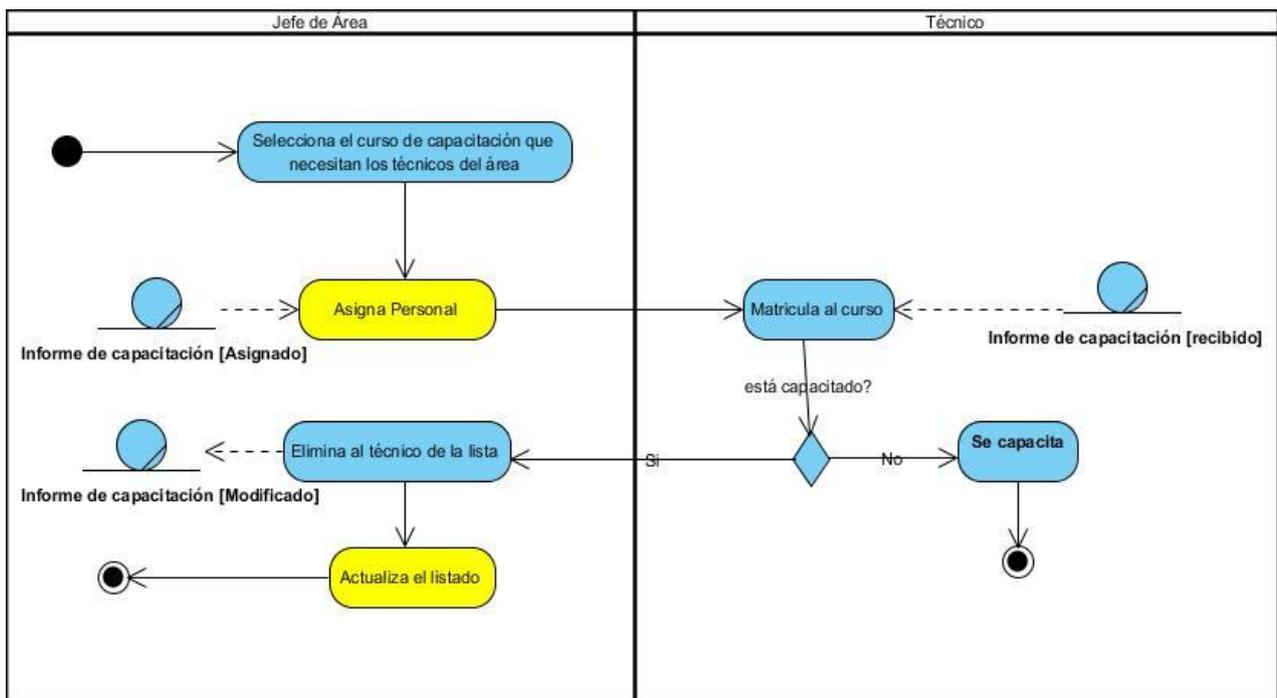


Figura 2.2.4 Diagrama de Actividades del CU "Capacitar Personal".

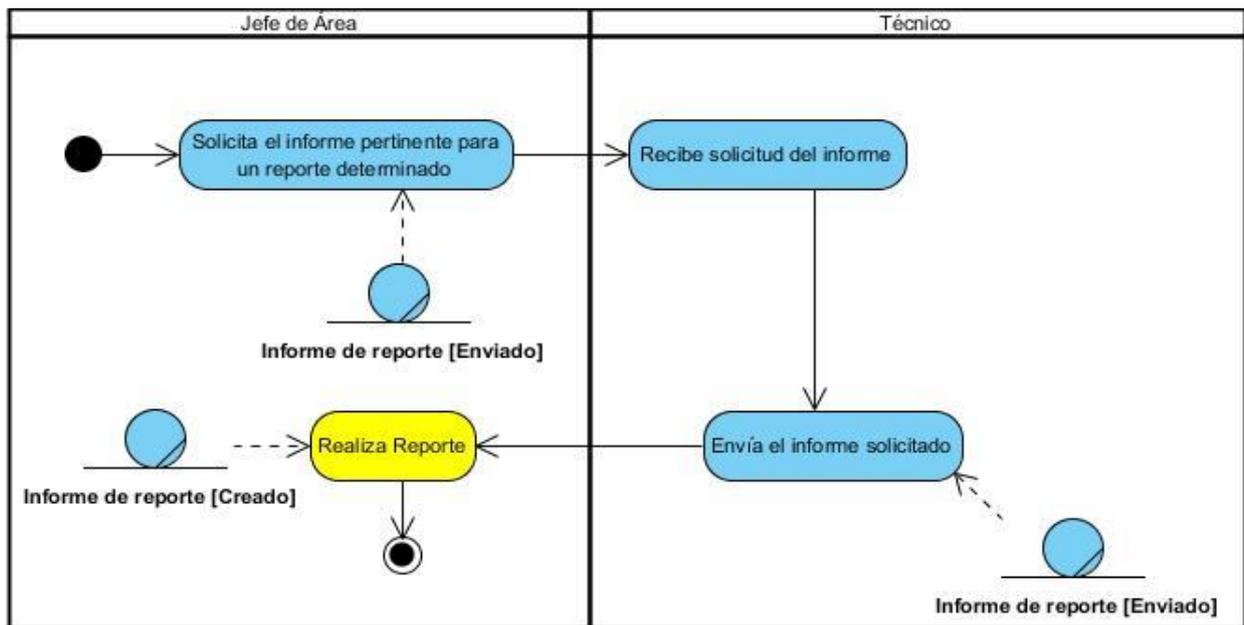


Figura 2.2.4 Diagrama de Actividades del CU "Realizar Reporte".

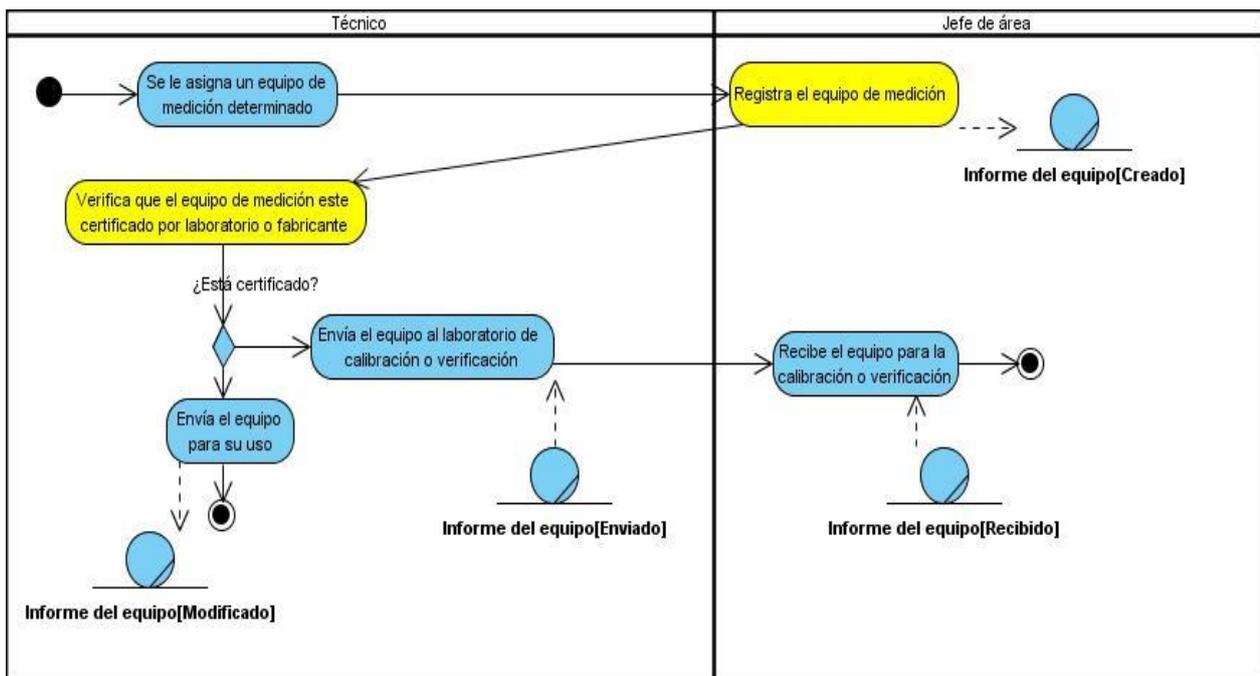


Figura 2.2.4. Diagrama de Actividades del CU "Controlar Equipo".

## 2.3 Definición de los requisitos funcionales

Los requisitos funcionales son "capacidades o condiciones que el sistema debe cumplir, especifican las acciones que el sistema debe ser capaz de realizar, sin tener ningún tipo de

restricción física y cómo debe reaccionar ante situaciones particulares. Estos deben ser de fácil entendimiento para el usuario y para los desarrolladores”. (13)

**Tabla 4. Descripción de los requisitos funcionales del sistema.**

No.	Nombre	Descripción
RF1	Autenticar usuario	Permite al usuario interactuar con el sistema según los niveles de acceso predeterminado por la empresa Cubana de Aviación.
RF2	Crear equipo de medición	Permite adicionar equipos de medición en el sistema según se vayan disponiendo.
RF3	Mostrar equipo de medición	Permite mostrar los detalles de un equipo de medición determinado.
RF4	Eliminar equipo de medición	Permite eliminar un equipo determinado.
RF5	Editar equipo de medición	Permite actualizar el estado de un equipo determinado.
RF6	Enviar equipo al área	Permite enviar un nuevo equipo a un área determinada.
RF7	Buscar equipo	Permite buscar los detalles de un equipo determinado.
RF8	Crear área	Permite crear una nueva área en el sistema.
RF9	Mostrar área	Permite mostrar los detalles de un área determinada.
RF10	Editar área	Permite actualizar el estado de un área determinada.
RF11	Eliminar área	Permite eliminar un área determinada.
RF12	Enviar equipo al	Permite enviar un equipo a un laboratorio específico.

## Capítulo II

	<b>laboratorio</b>	
<b>RF13</b>	<b>Crear plan</b>	<b>Permite crear un nuevo plan de aseguramiento para los equipos del área.</b>
<b>RF14</b>	<b>Mostrar plan</b>	<b>Permite mostrar los detalles de un plan determinado.</b>
<b>RF15</b>	<b>Editar plan</b>	<b>Permite actualizar el estado del plan de aseguramiento.</b>
<b>RF16</b>	<b>Eliminar plan</b>	<b>Permite eliminar un plan determinado.</b>
<b>RF17</b>	<b>Enviar plan por correo</b>	<b>Permite enviar un plan por correo.</b>
<b>RF18</b>	<b>Crear capacitación</b>	<b>Permite crear un nuevo curso de capacitación.</b>
<b>RF19</b>	<b>Mostrar capacitación</b>	<b>Permite mostrar los detalles del curso de capacitación.</b>
<b>RF20</b>	<b>Editar capacitación</b>	<b>Permite actualizar el estado del curso de capacitación.</b>
<b>RF21</b>	<b>Eliminar capacitación</b>	<b>Permite eliminar un curso de capacitación.</b>
<b>RF22</b>	<b>Crear laboratorio</b>	<b>Permite crear un nuevo laboratorio.</b>
<b>RF23</b>	<b>Mostrar laboratorio</b>	<b>Permite mostrar los detalles del laboratorio.</b>
<b>RF24</b>	<b>Editar laboratorio</b>	<b>Permite actualizar el estado del laboratorio.</b>
<b>RF25</b>	<b>Eliminar laboratorio</b>	<b>Permite eliminar un laboratorio.</b>
<b>RF26</b>	<b>Evaluar Servicio</b>	<b>Permite evaluar el servicio brindado por el laboratorio.</b>

RF27	Mostrar servicio	Permite mostrar los detalles del servicio.
RF28	Editar servicio	Permite actualizar el estado del servicio.
RF29	Eliminar servicio	Permite eliminar un servicio.
RF30	Crear clasificador	Permite crear un nuevo clasificador.
RF31	Mostrar clasificador	Permite mostrar los detalles del clasificador.
RF32	Editar clasificador	Permite actualizar el estado del clasificador.
RF33	Eliminar clasificador	Permite eliminar un curso de clasificador.
RF34	Crear personal	Permite crear un nuevo personal.
RF35	Mostrar personal	Permite mostrar los detalles del personal.
RF36	Editar personal	Permite actualizar el estado del personal.
RF37	Eliminar personal	Permite eliminar un curso de personal.
RF38	Crear grupo	Permite crear un nuevo grupo.
RF39	Mostrar grupo	Permite mostrar los detalles del grupo.
RF40	Editar grupo	Permite actualizar el estado del grupo.
RF41	Eliminar grupo	Permite eliminar un curso de grupo.

### 2.3.1 Definición de los requisitos no funcionales

Los requisitos no funcionales son “propiedades, cualidades o restricciones que el producto debe cumplir, tales como restricciones de tiempo, estándares, de desarrollo, entre otros con el objetivo de lograr un producto confiable, atractivo y seguro”. (13)

Los requisitos no funcionales de la aplicación han sido agrupados por categorías y se muestran a continuación.

- **Requisitos de Usabilidad.**

El software será usado por los usuarios del taller de la empresa Cubana de Aviación y por el equipo de desarrollo de la entidad. El sistema debe ser fácil para el uso de los usuarios, así obtener un mejor entendimiento.

- **Requisitos de software.**

La estación de trabajo donde se instale la aplicación debe contar con sistema operativo Linux o Windows, ya que estos han sido los dos escenarios en los que ha sido instalada y configurada la aplicación. Para publicarla se requiere del servidor web Apache 2.0 y como servidor de base de datos debe utilizarse MySQL. El navegador web a utilizar debe ser Mozilla Firefox de la versión 10 en adelante, debido al uso de las nuevas etiquetas semánticas de HTML5 y de las reglas de CSS3.

- **Requisitos de hardware.**

A pesar de ser una aplicación web con soporte de datos, el host donde estará alojada la misma debe contar con al menos 1 GB<sup>24</sup> disponibles de HDD<sup>25</sup>, debido a que los elementos que se gestionen serán almacenados en directorios físicos dentro de la estructura de carpetas de la aplicación.

- **Restricciones del diseño y la implementación.**

- ❖ Lenguaje PHP en su versión 5.
- ❖ Lenguaje JavaScript.
- ❖ Lenguaje HTML en su versión 5.
- ❖ Lenguaje CSS en su versión 3.
- ❖ IDE NetBeans 7.0.
- ❖ Framework Symfony 2.4.3.
- ❖ Framework jQuery.
- ❖ ORM Doctrine.

---

<sup>24</sup> Gigabytes.

<sup>25</sup> Del inglés: Hard Disk Drive.

## - Requisitos de apariencia o interfaz externa

- ❖ El menú de acciones, permitirá que el usuario pueda disponer de él en la medida de sus necesidades, visible todo el tiempo para propiciar el fácil acceso a las funcionalidades del software.
- ❖ Los servicios que brinda, contarán con términos asociados al tema propiciando que el usuario tenga un buen entendimiento de las utilidades que brinda.

## - Requisitos de Seguridad

- ❖ Permitirá además la verificación sobre acciones irreversibles; es decir, se le solicitará al usuario la confirmación al realizar operaciones como la eliminación.
- ❖ El uso y manejo del sistema estará controlado por el administrador. Toda la información podrá ser consultada solamente por el personal autorizado de acuerdo al rol del mismo.

## - Requisitos de Fiabilidad

- ❖ Disponibilidad de la aplicación en todo momento.
- ❖ Exactitud en las salidas de la aplicación.

## 2.4. Actores del sistema

Un actor es un conjunto de roles que los usuarios desempeñan cuando interactúan con los casos de uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que se ha identificado los actores del sistema, se identifica el entorno externo del sistema. (30)

A continuación se muestran los actores del sistema en la tabla 5:

Tabla 5. Actores del sistema

Actor	Descripción
Usuario	Es el que realiza la autenticación en el sistema ya sea administrador, Técnico o Jefe de Área.
Jefe de Calidad	Es el encargado de administrar, configurar y dar mantenimiento al sistema, además tiene permisos para gestionar los roles.
Técnico	Es el que realiza todo lo referente a la gestión del equipo de medición, ya sea adicionar, eliminar, modificar o mostrar.

<b>Jefe de Área</b>	Es el que lleva control del inventario, genera reportes y gestiona los laboratorios, áreas y capacitación del personal.
---------------------	---

## 2.4.1 Caso de Uso del sistema

Los casos de uso que cubren las principales tareas o funciones que el sistema ha de realizar son los siguientes:

- ❖ Autenticar Usuario
- ❖ Gestionar Personal
- ❖ Gestionar Equipo
- ❖ Gestionar Laboratorio
- ❖ Gestionar Área
- ❖ Gestionar Capacitación
- ❖ Gestionar Clasificador
- ❖ Gestionar Plan de aseguramiento
- ❖ Gestionar Servicio
- ❖ Gestionar Clasificador.
- ❖ Gestionar Grupo.

## 2.4.2 Diagrama de CU del sistema

Un diagrama de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. El diagrama de casos de uso permite que los desarrolladores y los clientes lleguen a un acuerdo sobre los requisitos, es decir sobre las condiciones y posibilidades que debe cumplir el sistema. El diagrama de casos de uso describe lo que hace el sistema para cada tipo de usuario. Cada usuario se representa mediante uno o más actores.

Identificados los requisitos, los actores y casos de usos del sistema anteriormente se estructura el diagrama de casos de usos del sistema, el cual representa la relación de los actores del sistema con los casos de uso. Para la confección del mismo se emplearon un conjunto de patrones de casos de uso tales como:

- ❖ CRUD completo para modelar las diferentes operaciones que se pueden realizar con una información, como: crear, leer, actualizar y eliminar. Ejemplo de ello son los casos de usos Gestionar Equipo, Gestionar Área, Gestionar Laboratorio y otros.
- ❖ CRUD parcial para modelar las diferentes operaciones que se pueden realizar con una información, pero en este caso no cumple con todas las operaciones como en el caso anterior. Ejemplo de ello es el caso de uso Gestionar Servicio.
- ❖ Múltiples Actores (rol común) para modelar cómo, varios actores pueden tener un rol común. Ejemplo de ello es el caso de uso Autenticar Usuario, el cual es accedido por los tipos de usuarios: Usuario, Jefe de Calidad, Jefe de Área y Técnico.

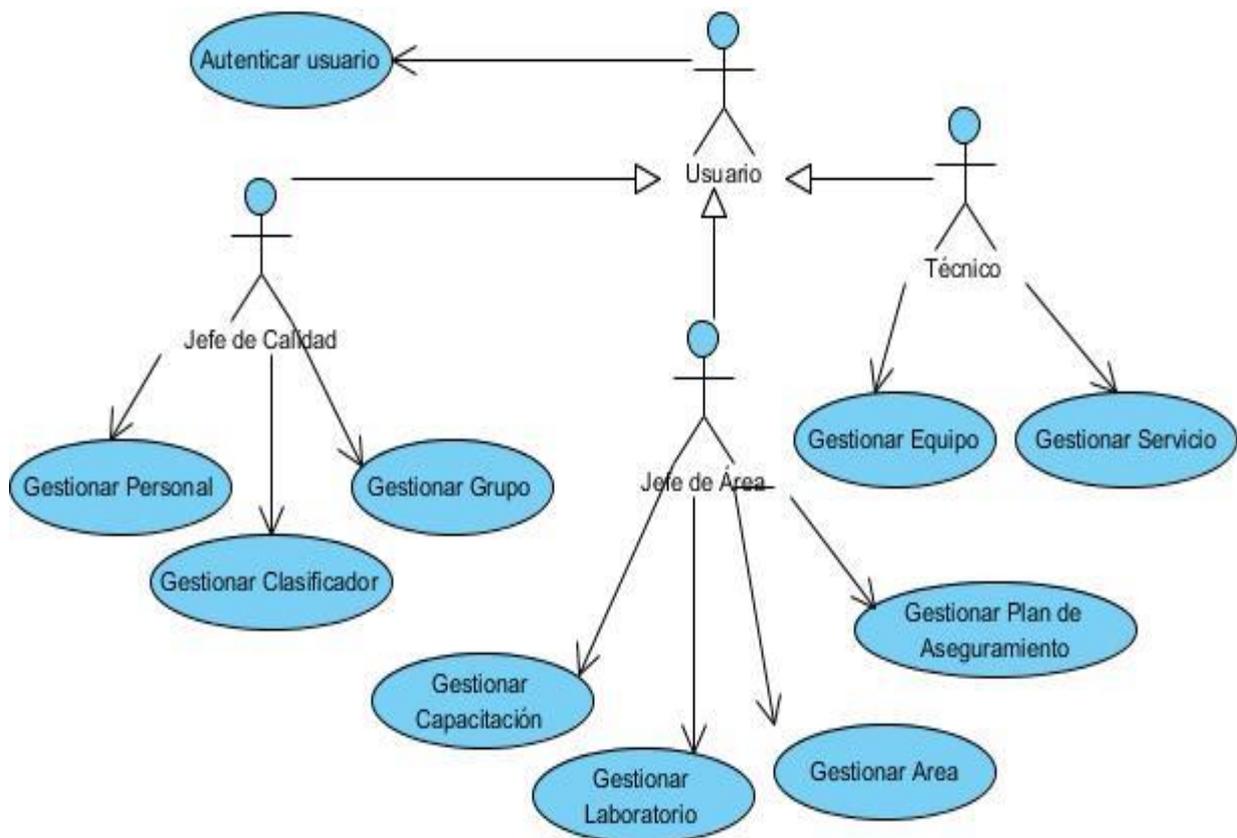


Figura 2.4.3.1. Diagrama de Caso de Uso del sistema

## 2.4.3 Descripción de los CU del sistema

Mediante las descripciones de los casos de uso se especifican la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. (30) A continuación se muestra un ejemplo de especificación de casos de uso, correspondiente al caso de uso Gestionar Equipo:

Tabla 2.4.4.1 Descripción del CU del sistema "Gestionar Equipo"

# Capítulo II

CU	Gestionar Equipo	
Actores	Técnico (Inicia): Incluye crear, ver, modificar o eliminar equipo. Usuario: Ver equipos.	
Resumen	El caso de uso se inicia cuando el actor Técnico selecciona la opción que le permite realizar una acción sobre un equipo. El actor Usuario puede ver un equipo.	
Precondiciones	El Técnico debe estar autenticado en el sistema.	
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción "Equipo" en la página principal.	
2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Crear.</li> <li>• Ver. Ver sección "Ver Equipo".</li> <li>• Editar. Ver sección "Editar Equipo".</li> <li>• Eliminar. Ver sección "Eliminar Equipo"</li> </ul>
3.	Selecciona la opción Crear.	
4.		Brinda la posibilidad de introducir los datos del equipo. <ul style="list-style-type: none"> <li>• Denominación</li> <li>• Serie</li> <li>• Magnitud</li> <li>• Área</li> <li>• Ubicación</li> </ul> Permite: <ul style="list-style-type: none"> <li>• Guardar.</li> <li>• Volver al Listado.</li> </ul>
5.	Introduce los datos del equipo.	
6.	Selecciona la opción de guardar los datos.	
7.		Valida los datos.
8.		Crea el nuevo equipo.
9.		Muestra un mensaje de información "El equipo se ha insertado exitosamente."

# Capítulo II

10.		El caso de uso termina.
<b>Flujo Alternos</b>		
4.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
4.a.1		Regresa a la interfaz anterior.
4.a.2		El caso de uso termina.
7.a Existen datos incompletos		
	Actor	Sistema
7.a.1		Muestra un indicador sobre los campos vacíos.
7.a.2		Regresar al paso 5.
7.b Existen datos incorrectos		
	Actor	Sistema
7.b.1		Muestra un indicador sobre los campos incorrectos.
7.b.2		Regresar al paso 5.
<b>Sección “Ver Equipo”</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1.	Selecciona la opción ver asociada a un equipo.	
2.		Muestra la pantalla principal del producto con los datos asociados al mismo. Permite: <ul style="list-style-type: none"> <li>• Volver al Listado</li> </ul>
3.	Selecciona la opción volver al listado.	
4.		Muestra la interfaz “Equipo”.
5.		El caso de uso termina.
<b>Sección “Editar Equipo”</b>		
<b>Flujo básico</b>		
	Actor	Sistema

# Capítulo II

1.	Selecciona la opción modificar asociada a un producto.	
2.		Muestra los datos editables del equipo. Permite además: <ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Volver al Listado.</li> </ul>
3.	Modifica los datos que necesita y selecciona la opción guardar.	
4.		Valida los datos.
5.		Muestra el listado de los equipos.
6.		El caso de uso termina.
<b>Flujos alternos</b>		
2.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
2.a.1		Regresa a la interfaz anterior.
2.a.2		El caso de uso termina.
4. a Existen datos incorrectos.		
	Actor	Sistema
4.a.1		Muestra un indicador sobre los datos incorrectos.
4.a.2		Regresa al paso 3.
<b>Sección “Eliminar Equipo”</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1	Selecciona la opción eliminar asociada a un equipo.	
2		Muestra el mensaje de confirmación “¿Está seguro que desea eliminar este producto? Para confirmarlo teclee la palabra ELIMINAR en la caja de texto.”
3.	Selecciona la opción “Eliminar”	
4.		Elimina el producto.
5.		Muestra el listado con el resto de los equipos

		registrados en el sistema.
6.		El caso de uso termina.

Las demás tablas de las descripciones de los casos de usos del sistema pueden verse en el [Anexo3](#).

## Conclusiones del capítulo

Con la descripción y modelado del negocio se logró alcanzar una visión más completa de los requisitos del sistema. Lo que permitió la deducción de los casos de uso, obteniéndose como resultado la descripción de los casos de usos del sistema que describen el proceso de metrología del departamento de calidad de la empresa Cubana de Aviación.

## Capítulo 3: Arquitectura y diseño

### 3.1 Introducción

Según la IEEE<sup>26</sup> 610.12-90 el diseño del software es el proceso de definición de la arquitectura, componentes, interfaces y otras características de un sistema o componente que resulta de este proceso. Estas actividades son las que se describirán a lo largo de este capítulo, sirviendo al mismo tiempo para justificar la utilización de determinados patrones de diseño y algunos elementos importantes dentro del modelo de datos del sistema.

### 3.2 Arquitectura del sistema

Para el desarrollo del sistema se utiliza el patrón arquitectónico Modelo-Vista-Controlador (MVC) en el cual está basado el framework Symfony. Este patrón separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Además permite mantener el controlador y el modelo original para aplicaciones que se accedan desde navegadores estándar o móviles cambiando solamente la vista para cada dispositivo (29) (33). La separación de la vista y el modelo trae ventajas; es posible tener diferentes representaciones de la misma información, haciendo uso del mismo código dentro del modelo. Es posible además programar el código del modelo, abstrayéndose de la representación visual que se le dará a la información. El patrón MVC define tres roles o niveles (25):

- **El modelo:** representa la información relacionada con el dominio de la aplicación, es decir, su lógica del negocio. Abstrae la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes del tipo de gestor de base de datos utilizado. El ORM Doctrine permite utilizar el lenguaje DQL en las clases del modelo generadas para manipular los contenidos de la base de datos. Dichas clases son generadas a partir del esquema de la base de datos escrito en XML o YAML.
- **La vista:** transforma el modelo en interfaces de usuario permitiendo la interacción con el sistema. Solo se encarga de mostrar información. Esta capa está separada en tres partes: los elementos comunes a todas las páginas de la aplicación, denominada layout; las plantillas, encargadas de visualizar las variables definidas en el controlador; y la lógica de las

---

<sup>26</sup> Del inglés: *Institute of Electrical and Electronics Engineers*.

vistas, definida a través de los slots y componentes (fragmentos de vistas que contienen lógica de aplicación y pueden ser reutilizables).

- **El controlador:** se encarga de procesar las peticiones realizadas desde el navegador realizando los cambios necesarios en el modelo o en la vista. Este se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP). Symfony divide esta capa en un controlador frontal y acciones. El controlador frontal es el único punto de entrada a la aplicación, carga la configuración y determina las acciones a ejecutarse. Este se encuentra ubicado en el directorio web del sistema. Las acciones verifican la integridad de las peticiones, contienen la lógica de la aplicación y preparan los datos necesarios para actualizar la vista.

La capa del controlador contiene los objetos encargados de acceder a los parámetros de la petición, a las cabeceras de las respuestas y a los datos de sesión del usuario. Además brinda la posibilidad de ejecutar filtros antes o después de cada acción; según las comprobaciones sistemáticas realizadas en el filtro, puede o no ser modificado el procesamiento de la petición (25).

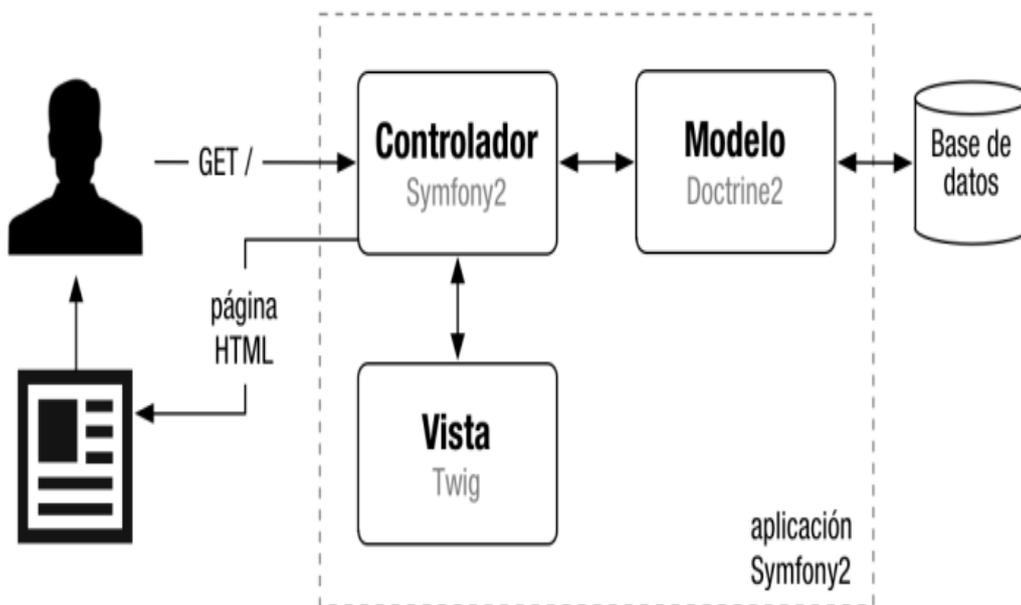


Figura 3.2 Patrón modelo-vista-controlador en Symfony.

### 3.3 Patrones de diseño

Un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos (31). El framework Symfony utiliza en su implementación un conjunto de patrones de diseño para dar solución a problemas específicos del diseño orientado a objetos durante el flujo de ejecución de una petición. Para el desarrollo del sistema no fue necesario utilizar patrones de diseño diferentes a los que utiliza el framework, por tal motivo se explicarán los utilizados directamente en la implementación de la aplicación.

## Patrones GRASP (31)

- **Experto:** este patrón es el principio básico de asignación de responsabilidades. Pretende asignar una responsabilidad al experto en información, o sea, la clase que cuenta con información necesaria para cumplir la responsabilidad, ya sea crear un objeto o implementar un método (31). La utilización de este patrón dentro del sistema permite un diseño con mayor cohesión y se mantiene el encapsulamiento, ayudando a entender y mantener el código fácilmente. El patrón Experto fue utilizado en la capa de abstracción del modelo. Las clases generadas poseen un grupo de funcionalidades que facilitan el acceso y la manipulación de los datos de las entidades persistentes en la base de datos.
- **Creador:** en el desarrollo de un sistema orientado a objetos conviene contar con un principio general para asignar la responsabilidad de creación. En este sentido, el patrón Creador ayuda a identificar quién debe ser el responsable de crear o instanciar un objeto o clase en dependencia de la relación y visibilidad entre la clase creada y la clase creador (31). La ventaja de utilizar este patrón es eliminar la dependencia entre clases, facilitando el mantenimiento y la reutilización. Dentro del sistema este patrón se evidencia en las acciones del controlador, las cuales crean objetos del modelo o los formularios que representan las entidades.
- **Alta cohesión:** la información que almacena una clase debe ser coherente y debe estar relacionada con la clase, asignando responsabilidades con una alta cohesión (31). Dentro del sistema, este patrón fue utilizado en las acciones, las cuales contienen varias funcionalidades que están relacionadas. Un ejemplo de ello lo constituyen los diferentes "actions" que definen las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades de los mismos.

- **Bajo acoplamiento:** asigna una responsabilidad a una clase para mantener bajo acoplamiento, o sea, disminuir la dependencia entre clases, evitando que una modificación en alguna de ellas repercuta en gran medida en el resto, posibilitando además una mayor reutilización (31). Este patrón es utilizado por el framework Symfony, y por ende en el sistema, en las clases pertenecientes a las acciones que heredan únicamente de “sfAction” para alcanzar un bajo acoplamiento entre clases. Además, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja.
- **Controlador:** asigna la responsabilidad del manejo de mensajes de los eventos del sistema a una o varias clases. Sirve como intermediario entre una determinada interfaz y la acción que debe ejecutarse, es decir, recibe los datos del usuario y los envía a las distintas clases según el método llamado (31). Este patrón aumenta la reutilización de código. Dentro del sistema el patrón Controlador es utilizado en las clases de la capa del controlador del patrón MVC, como son: “sfController”, “sfWebController”, “sfContext”, “sfAction” y las clases que heredan de la misma (los “actions”), así como el index.php del ambiente (en el caso del sistema integrado al proyecto, “backend.php”).

## Patrones GOF

- **Singleton:** patrón de tipo creación, a nivel de objetos. Garantiza que una clase solo tiene una única instancia, proporcionando un punto de acceso global a la misma (32). El objeto “sfContext” de Symfony sirve de mecanismo para interactuar con los objetos únicos del núcleo del framework, ya que almacena una referencia a todos estos objetos. Además la clase “sfRouting” que se encarga de enrutar todas las peticiones que de hagan a la aplicación implementa este patrón mediante el método “getInstance”.
- **Decorator:** patrón de tipo estructura, a nivel de objetos. Añade responsabilidades adicionales a un objeto dinámicamente (32). Symfony utiliza el patrón Decorator para la vista, el layout o plantilla global decora el contenido de la plantilla.

## 3.4 Diagrama de clases del Diseño

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso y se centra en los requisitos funcionales y no funcionales. Las abstracciones del modelo del diseño tienen una correspondencia directa con los elementos físicos del ambiente de implementa-

ción. Los objetivos del diseño de un caso de uso son: identificar las clases (subsistemas o interfaces) necesarias para llevar a cabo el caso de uso y definir los requisitos de las operaciones sobre dichos artefactos, permitiendo así reflejar con claridad las restricciones que deben ser cumplidas a la hora de la implementación (13). En la Figura 6 Diagrama de clases del diseño para el caso de uso Gestionar Equipo se presenta el diagrama de clases del diseño para el caso de uso Gestionar Equipo; el mismo está estructurado de acuerdo con el patrón arquitectónico MVC y muestra la interacción con el ORM Doctrine, el framework de presentación jQuery y el resto de los elementos que provee el uso de Symfony.

A continuación se representa la clase diseño del CU Gestionar Equipo, para ver el resto de los diagramas puede dirigirse al [Anexo4](#).

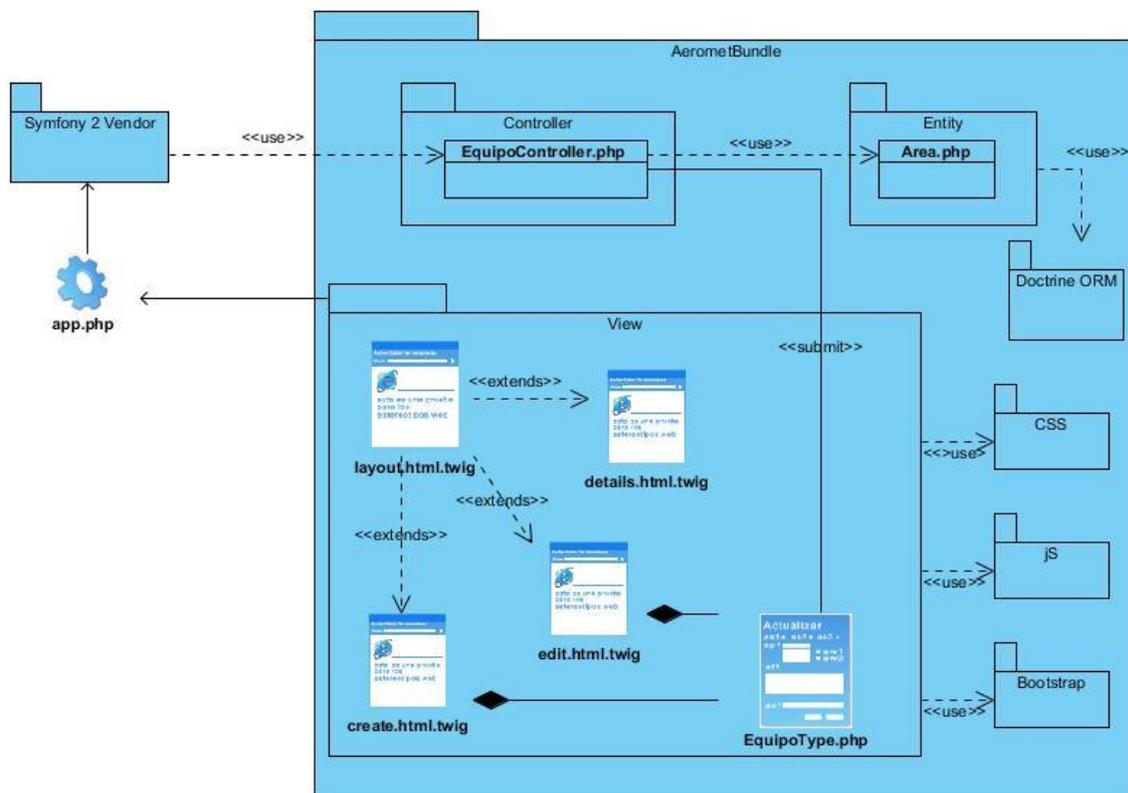


Figura 3.4.1 Diagrama de la clase de diseño. CU Gestionar Equipo.

### 3.5 Diagrama Entidad-Relación

El Diagrama Entidad-Relación (DER) le permite al ingeniero del software identificar los objetos de datos y sus relaciones por medio de una notación gráfica, define en su análisis estructural los datos que son adicionados, almacenados, modificados y producidos dentro de una aplicación.

(31)

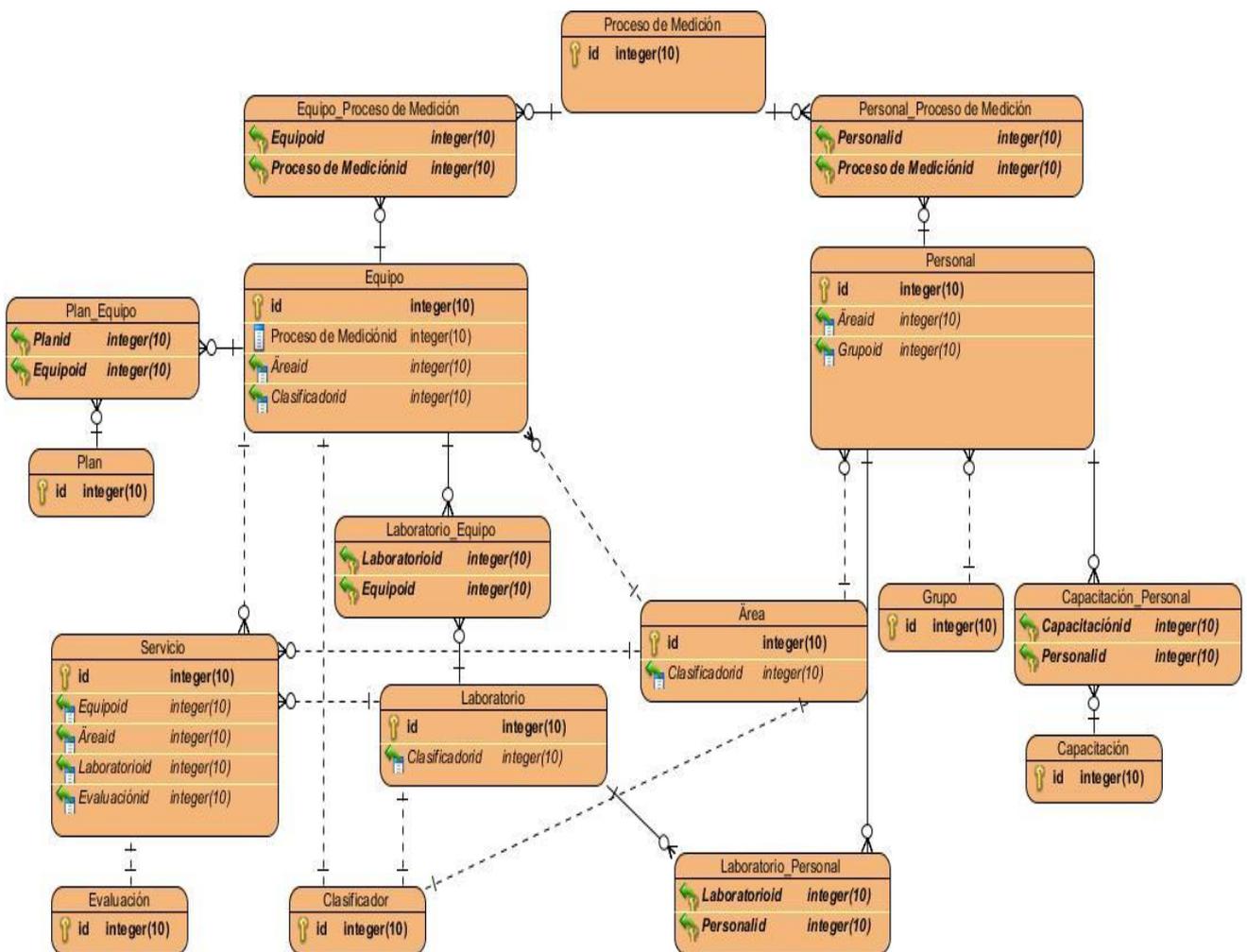


Figura 3.5.1 Diagrama Entidad-Relación

A continuación se describe la tabla equipo, perteneciente al modelo de datos presentado anteriormente. Las descripciones de las demás tablas pueden apreciarse en el [Anexo5](#).

Descripción de la tabla equipo:

Tabla equipo					
Nombre de columna	de	Tipo de Datos	PK / FK	Anulable	Documentación
<u>id</u>		int(11)	PK	No	Almacena el id del equipo, único en toda la tabla.
nombre		varchar(255)	PK	No	Almacena el nombre del equipo.

numero_serie	varchar(255)	PK	No	Almacena la descripción del equipo.
tipo_equipo	int(11)		Sí	Almacena el tipo de equipo.
estado_equipo	int(11)		Sí	Almacena el estado del equipo.
tipo_um	int(11)		Sí	Almacena la unidad de medida del equipo.
tipo_ciclo	int(11)		Sí	Almacena el ciclo del equipo.
tipo_magnitud	int(11)		Sí	Almacena la magnitud del equipo.
tipo_ubicación	int(11)		Sí	Almacena la ubicación del equipo.
laboratorio_id	int(11)		Sí	Almacena el id del laboratorio donde se encuentra el equipo.
area_id	int(11)		Sí	Almacena el id del área a la que pertenece el equipo.

### 3.6 Diagrama de despliegue

“El diagrama de despliegue se utiliza para modelar la disposición física de los componentes de hardware utilizados en la implementación del sistema y la relación entre cada uno de ellos”.

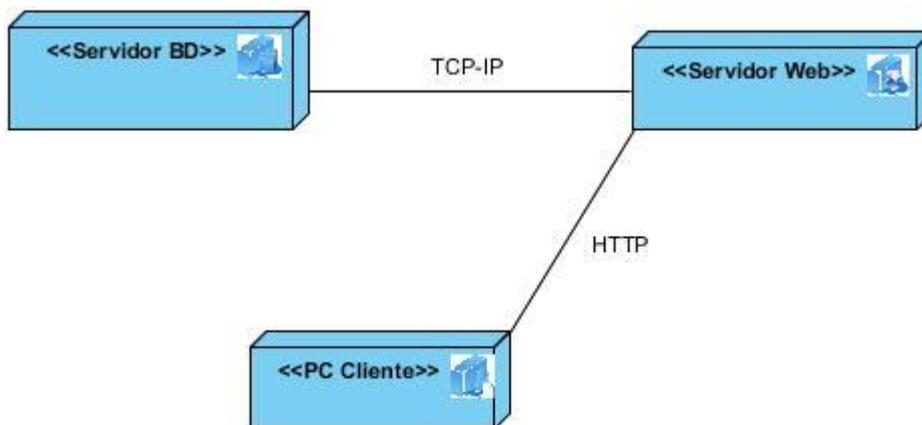


Figura 4.3.1 Diagrama de despliegue del sistema

Para el despliegue del sistema es necesario:

- ❖ Una PC Cliente que puede ser de tipo portátil o de escritorio la cual debe tener instalado un navegador web preferentemente Mozilla Firefox v13.0 o superior.

- ❖ **Un Servidor Web que debe tener instalado el contenedor web Apache que permita las conexiones con el cliente, generando las respuestas a sus peticiones. Las conexiones del cliente con el servidor se realizan mediante el protocolo HTTP.**
- ❖ **Un servidor de base de datos que debe tener instalado el sistema gestor de base de datos es MySQL.**

## **Conclusiones del capítulo**

A lo largo del capítulo fueron descritos todos los aspectos que tuvieron un impacto significativo en el diseño del software en cuestión. Es importante destacar que el framework de desarrollo Symfony propició el uso del patrón arquitectónico MVC y de una serie de patrones de diseño que, en su conjunto, garantizaron la reutilización de las funcionalidades y ponen de manifiesto el uso de buenas prácticas en la codificación del sistema. Además fueron representados en sus respectivos diagramas el modelo de diseño para uno de los casos de uso de la aplicación, el modelo de despliegue y el de base de datos, artefactos que son la entrada al flujo de trabajo implementación.

## Capítulo 4. Implementación y pruebas del sistema

### 4.1 Introducción

Los artefactos generados en el diseño son la principal entrada en el flujo de trabajo implementación, el cual tiene como objetivo implementar las clases identificadas durante el diseño. En el presente capítulo se describen cada uno de dichos componentes y la estrategia de prueba utilizada para verificar el resultado de la implementación.

### 4.2 Modelo de implementación

La organización de los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguajes de programación utilizados es descrita en el modelo de implementación. Este modelo además describe la forma en que los elementos obtenidos del modelo de diseño (como las clases) se implementan en términos de componentes y las dependencias que presentan dichos componentes entre sí. El modelo de implementación es la entrada principal para las pruebas de software que siguen a la etapa de implementación (18).

#### Diagrama de componentes

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño (18). El diagrama de componentes se utiliza para modelar la vista estática de un sistema. Se encarga de mostrar los componentes de software y las relaciones lógicas entre ellos en un sistema, además de estructurar el modelo de implementación en términos de subsistemas de implementación.

A continuación se muestra el diagrama de componentes para la implementación del software:

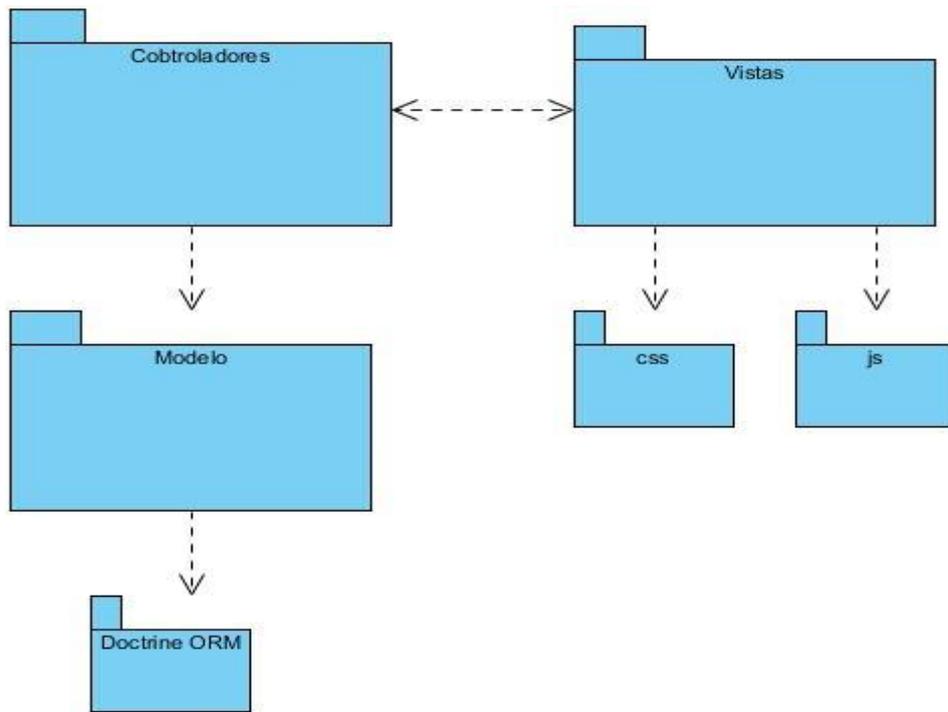


Figura 4.2.1 Diagrama de componente del sistema

## 4.2.1 Descripción de los componentes

- Paquete de componentes "Vistas"

El paquete de componente "Vistas" es una representación de las páginas .twig con las cuales el usuario interactúa, permitiendo la entrada de datos al sistema y realizar consulta de dichos datos.

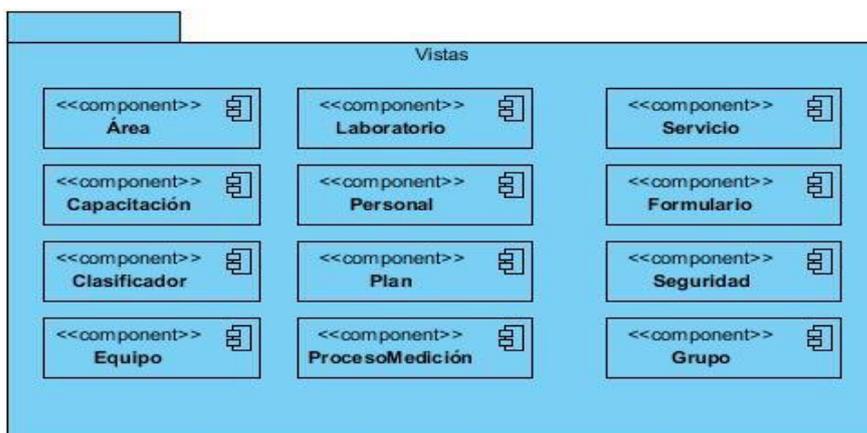


Figura 4.2.1.1 Paquete Vistas

## Paquete de componentes “Controladores”

El paquete de componentes “Controladores” es una representación de las operaciones que permiten el flujo y la comunicación entre las vistas y el modelo, son los encargados de recibir las órdenes por parte del usuario, gestionar la ejecución de la lógica de negocio y posteriormente actualizar la vista para que el usuario pueda ver como ha quedado el modelo de datos tras las actualizaciones.

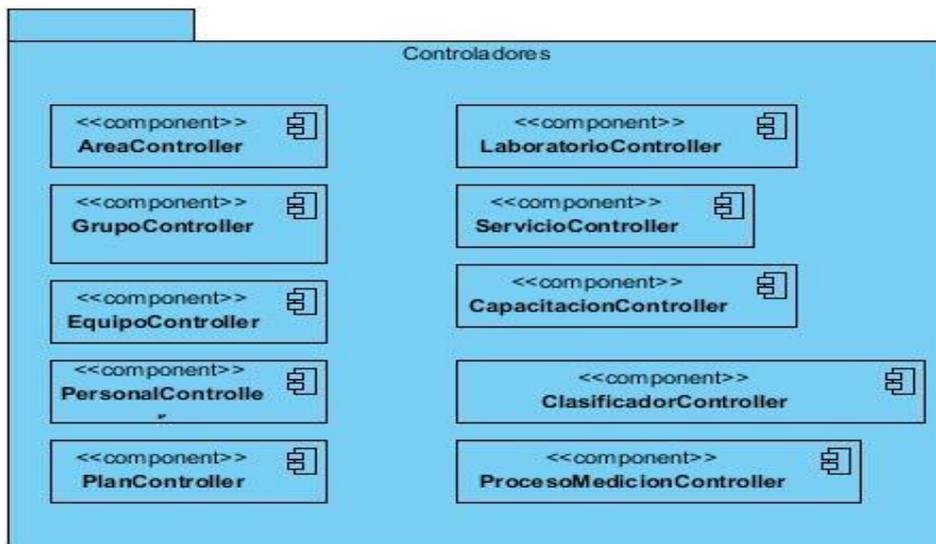


Figura 4.2.1.2 Paquete Controladores

### 4.2.1.1 Paquete de componentes “Modelo”

Son las clases que representan todo el negocio del sistema. Estas clases del dominio constituyen entidades o tablas que son utilizadas en la base de datos mapeadas.

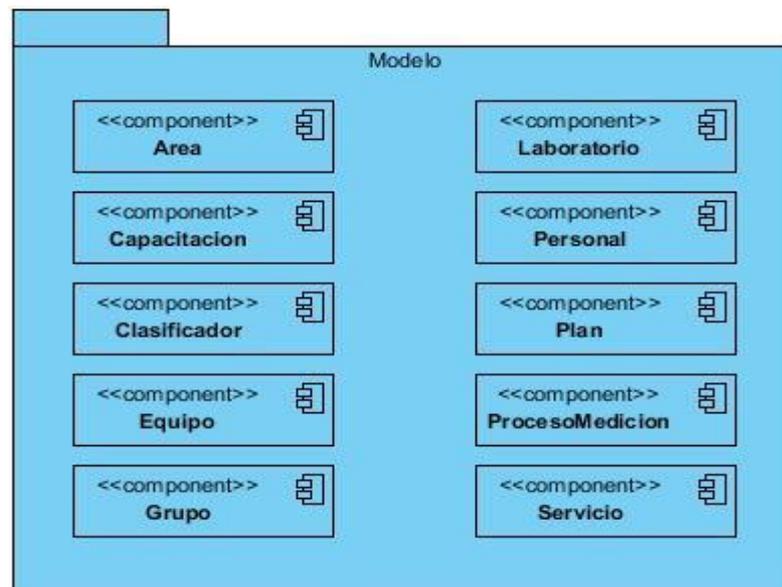


Figura 4.2.1.3 Paquete Modelo

### 4.3 Estándares de codificación

El proceso de codificación contiene el conjunto de reglas o normas usadas para escribir código, lo cual aporta eficiencia al proceso de desarrollo, logrando que los programas sean más comprensibles y de mayor calidad. Los aspectos fundamentales que se deben tener en cuenta a la hora de confeccionar un estilo de código son: notación, comentarios, espacios y líneas en blanco, longitud máxima de las líneas de caracteres, declaración de variables y constantes y parámetros de los métodos.

Los estándares definidos para el desarrollo de la aplicación han sido tomados, en su gran mayoría, de los definidos por la arquitectura de software al cual pertenece el sistema (2). A continuación de enumeran algunos de ellos:

- Se utiliza el tabulador para la alineación, excepto en los archivos YAML donde se utilizan dos espacios.
- Cuando una expresión no cabe en una línea simple debido a su extensión se divide en más de una línea, después de una coma o un operador.
- Los comentarios se definen comenzando con los caracteres “/\*” y terminando con “\*/”.

- Las variables deben ser explícitas y sin abreviaturas.

## 4.4 Pruebas del sistema

**El único instrumento adecuado para determinar el estado de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software en específico o al sistema de software en su totalidad, con el objetivo de medir el grado en que se cumplen los requisitos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de pruebas.**

### Estrategia de prueba

Una vez descrito el software en términos de sus componentes se hizo necesario verificar el funcionamiento del mismo para garantizar calidad en el producto. Los objetivos de las pruebas son (13):

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas de sistema.
- Diseñar e implementar pruebas creando los casos de prueba (especifican qué probar), procedimientos de prueba (especifican cómo realizar las pruebas), creando componentes de prueba para automatizar estos procesos.
- Realizar las pruebas y manejar los resultados de cada una de ellas sistemáticamente. Dentro de este flujo de trabajo en el ciclo de desarrollo se definieron niveles, tipos de pruebas, métodos y técnicas, para verificar que todos los requisitos fuesen implementados. Lo anterior no garantiza la ausencia de defectos en la aplicación pues las pruebas solo se limitan a demostrar la existencia de dichos defectos.

### Niveles, tipos, métodos y técnicas de pruebas

Entre los niveles generales de las pruebas (13) que propone la metodología en uso, el seleccionado fue el de pruebas internas, y dentro de él, las pruebas de aceptación. Los casos de prueba resultantes de esta estrategia están dirigidos a evidenciar la integración de cada componente y el funcionamiento del sistema como un todo. La siguiente tabla relaciona con más detalles la estrategia diseñada.

Tabla 4.4.1 Niveles, tipos, métodos y técnicas para la estrategia de prueba.

Niveles de Pruebas	Tipos de Pruebas	Métodos	Técnicas
Aceptación	Funcionalidad(función)	Caja Negra	Partición de equivalencia

A la aplicación le fueron realizadas pruebas de funcionalidad (función y seguridad) que permitieron validar la implementación de funciones, métodos y casos de uso, así como verificar la integridad y disponibilidad de la información y el acceso a ella por parte de los actores deseados (13).

Las pruebas de caja negra son las que se llevan a cabo sobre la interfaz del software. A partir de los casos de prueba generados bajo este método se demostró que las funcionalidades del software son operativas, que las entradas se aceptan de la forma adecuada y que se produce el resultado correcto; fueron corregidos además algunos errores en la interfaz y en las funciones visibles al usuario. La técnica empleada fue la partición de equivalencia puesto que divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. En el puede ser consultado el caso de prueba para el caso de uso “Gestionar Equipo”.

### Entorno de las pruebas

De acuerdo con lo planteado en los requisitos no funcionales de hardware en el entorno de pruebas las computadoras utilizadas tenían 1 GB de RAM, los sistemas operativos Ubuntu y Windows, todos ellos con el navegador Mozilla Firefox. Para la conexión entre los diferentes nodos el ancho de banda disponible fue de 100 Mbps.

### Análisis de los resultados

Tras aplicar la estrategia de pruebas propuesta, se obtuvieron resultados satisfactorios que validan en cierta medida la calidad de la aplicación desarrollada. La Tabla 4.4.2 Resultados de las pruebas, muestra las no conformidades resultantes de tres iteraciones de pruebas:

Tabla 4.4.2 Resultados de las pruebas

	Iteración 1	Iteración 2	Iteración 3
Significativas	8	4	1
No Significativas	12	7	3

# Capítulo IV

Cantidad	20	11	4
----------	----	----	---

Las no conformidades encontradas en la primera iteración estuvieron estrechamente relacionadas con el sistema de almacenamiento de medias en los directorios físicos, conflictos con los permisos de usuarios. Debido al impacto que tenían en la calidad de los contenidos a gestionar, el equipo de trabajo se enfocó en solucionar dichas no conformidades, logrando en cada iteración de prueba disminuir el número de defectos en el software.

El resto de las no conformidades estuvieron vinculadas a los elementos del diseño de las interfaces y la arquitectura de la información en el software. Ante esta dificultad se decidió replantear estos elementos de acuerdo a las características del sistema al que va ser integrado el software.

A continuación se muestra el diseño del caso de prueba, correspondiente al flujo básico del caso de uso Gestionar Equipo. La versión expandida de algunos de los diseños de casos de prueba de los CU se encuentra en el Anexo6.

Escenario	Descripción	Denominación	No. Serie	Magnitud	Área	Ubicación	Respuesta del sistema	Flujo central
<b>EC 4.1</b> <b>Crear equipo.</b>	El actor selecciona la opción: Crear.	V	V	N A	V	N A	Brinda la posibilidad de introducir los datos del equipo:  <ul style="list-style-type: none"> <li>- Denominación</li> <li>- No. Serie</li> <li>- Magnitud</li> <li>- Área</li> <li>- Ubicación</li> </ul> Y permite:	Equipo/Crear.

# Capítulo IV

							<ul style="list-style-type: none"> <li>- Guardar</li> <li>- Volver al listado</li> </ul>	
<b>EC 4.1.1</b>	El actor selecciona la opción: Guardar.						<p>Valida los datos.</p> <p>Crea el nuevo equipo.</p> <p>Muestra un mensaje de información "El equipo se ha insertado exitosamente".</p>	Equipo/Crear/Guardar.
<b>EC 4.1.2</b>	El actor selecciona la opción: Volver al Listado.						Regresa a la interfaz anterior.	Equipo/Crear/Volver al listado
<b>EC 4.1.3</b>	Existen datos incompletos.						Muestra un indicador sobre los campos vacíos.	Equipo/Crear/Guardar.
<b>EC 4.1.4</b>	Existen datos incorrectos.						Muestra un indicador sobre los campos incorrectos.	Equipo/Crear/Guardar.
<b>EC 4.2 Ver</b>	El actor selecciona la opción: Ver						<p>Muestra los detalles asociados al equipo.</p> <p>Permite:</p> <ul style="list-style-type: none"> <li>- Volver al Listado</li> </ul>	Equipo/Ver.
<b>EC 4.2.1</b>	El actor selecciona la						Muestra la interfaz Equipo.	Equipo/Ver/Volver al Listado.

# Capítulo IV

	opción: Volver al listado.								
<b>EC 4.3</b> <b>Editar</b>	Selecciona la opción: Editar.	N A	N A	N A	N A	N A	Muestra los datos editables del equipo.  Permite:  - Guardar  - Volver al Listado	Equipo/Editar	
<b>EC 4.3.1</b>	El actor selecciona la opción: Guardar.						Valida los datos.  Muestra el listado de los equipos.	Equipo/Editar/Guardar	
<b>EC 4.3.2</b>	El actor selecciona la opción: Volver al Listado.						Regresa a la interfaz anterior.	Equipo/Editar/Volver al Listado.	
<b>EC 4.3.3</b>	Existen datos incorrectos.						Muestra un indicador sobre los datos incorrectos.	Equipo/Editar/Guardar	
<b>EC 4.4</b> <b>Eliminar</b>	El actor selecciona la opción: Eliminar						Muestra el mensaje de confirmación ¿Esta seguro que desea eliminar este producto?  Para confirmarlo selecciona la opción Aceptar.	Equipo/Ver/Eliminar  Equipo/Editar/Eliminar	
<b>EC 4.4.1</b>	Selecciona la opción Aceptar.						Elimina el equipo.  Muestra el listado con el resto de los equipos registrados en el sistema.	Equipo/Ver/Eliminar/A ceptar.  Equipo/Editar/Eliminar/ Aceptar.	

## **Conclusiones del capítulo**

Uno de los resultados de este capítulo es modelo de implementación, donde se definió una estrategia de prueba para verificar el funcionamiento del software. El desarrollo y evaluación de cada caso, componente y procedimiento de prueba estuvo enfocado en los casos de uso que estaban asociados a los requisitos de mayor impacto para el usuario. Las pruebas aplicadas al sistema dieron lugar a realizar mejoras sustanciales en la implementación de las capas de la lógica del negocio y presentación, logrando una apariencia más profesional.

# Conclusiones Generales

---

## Conclusiones Generales

Una vez finalizado el desarrollo del trabajo se llega a las conclusiones siguientes:

- ❖ Al elaborar el marco teórico conceptual a partir del estudio de las principales tendencias, metodologías y herramientas existentes relacionadas con el proceso de confirmación metrológica. Nos llevó a hacer uso de las tecnologías y herramientas más utilizadas en la actualidad.
- ❖ Al no ser fácil un estudio de sistemas similares, se evidenció la importancia del desarrollo del presente trabajo. Teniendo en cuenta algunas de las funcionalidades llevadas a cabo por los sistemas de gestión de la información en la actualidad.
- ❖ Al definir el modelo de negocio, se obtuvieron los requisitos funcionales del sistema. Se generaron los artefactos que dieron paso a la descripción de los casos de uso del proceso de confirmación metrológica.
- ❖ Al diseñar e implementar el sistema se obtuvo la descripción arquitectónica a utilizar en la implementación.
- ❖ Al validar el sistema desarrollado mediante pruebas se pudo verificar que el sistema cumpliera con los requerimientos establecidos por el cliente, validándose así la calidad del mismo.
- ❖ La implementación del sistema facilitó la gestión de la información en el departamento metrológico en el Departamento de Calidad de la Empresa Cubana de Aviación. Garantizando la inmediatez que requiere la toma de decisiones en la actualidad.

# Recomendaciones

---

- ❖ Se recomienda al equipo de desarrollo de la empresa hacer uso del trabajo para la integración del mismo o para el perfeccionamiento del sistema.
- ❖ Incluirle una funcionalidad que permita el manejo de la información interna a través del correo electrónico.

1. otros, Potencier y. 2011.
2. Hernández Leon, Rolando Alfredo y Coello Gonzalez, Aida. *El proceso de la Investigación*. la Habana : Editorial Universitaria del Ministerio de Educación Superior, 2011.
3. Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, Aureliano. *Metodología de Desarrollo del Software*. 2008.
4. Canós, José H., Letelier, Patricio y Penadés, María del Carmen. *Metodologías Ágiles en el Desarrollo de Software*. Valencia : Universidad Politécnica de Valencia.
5. (Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Wesley : Pearson Adisson, 2000.
6. Popkin Software and Systems. Modelado de Sistemas con UML. [En línea]
7. Sierra, María. *Trabajando con Visual Paradigm for UML*. Cantabria : Universidad de Cantabria–Facultad de Ciencias.
8. *Conferencia de Historia de la Informática*. La Habana : s.n., 2008. Historia de los lenguajes de programación.
9. Satrom, Brandon. MSDN Magazine. Creación de aplicaciones con HTML5: lo que necesita saber. [En línea] Agosto de 2011. [Citado el: 23 de Febrero de 2014.] <http://msdn.microsoft.com/es-es/magazine/hh335062.aspx>.
10. Freddy Vega, John y Van Der Henst, Christian. Guía de HTML. El presente de la web. [En línea] [Citado el: 23 de febrero de 2014.]
11. Eguíluz Pérez, Javier. Cascading Style Sheets home page. [En línea] 2 de enero de 2009. [Citado el: 23 de febrero de 2014.] [http://www.librosweb.es/css\\_avanzado](http://www.librosweb.es/css_avanzado).
12. Valdés., Damián Pérez. Los diferentes lenguajes de programación para la web. . [En línea] 2007. [Citado el: 23 de febrero de 2014.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programación-para-la-web>.

13. Potencier, Fabien y Zaninotto, François. *Symfony 2*. 2010.
14. Valdés, Mayliuvis Esquijarrosa y Díaz, Ernesto Vladimir Pereda. Desarrollo del módulo iTopics de la plataforma educativa ZERA. La Habana : s.n., 2011.
15. Sitio Oficial de jQuery. Sitio Oficial de jQuery. [En línea] 2010. [Citado el: 23 de febrero de 2014.] <http://jquery.com>.
16. Symfony. Sitio oficial de Symfony en español. [En línea] 2012. [Citado el: 23 de febrero de 2014.] <http://www.symfony.es/que>.
17. Sesiolabs. Symfony and Doctrine with sfDoctrinePlugin. 2009.
18. Mata, Manel Pérez. TecnoRetales. Qué es Doctrine ORM? . [En línea] 3 de julio de 2009. [Citado el: 23 de febrero de 2014.] <http://www.tecnoretalles.com/programacion/que-es-doctrine-orm>.
19. Entornos de Desarrollo Integrado. Entornos de Desarrollo Integrado. [En línea] [Citado el: 23 de febrero de 2014.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
20. The Eclipse Foundation. Eclipse. [En línea] 2012. [Citado el: 23 de febrero de 2014.] <http://eclipse.org/>.
21. Portal del IDE Java de Código Abierto. Bienvenido a NetBeans. [En línea] 2012. [Citado el: 23 de febrero de 2014.] <http://netbeans.org/index.es.html>.
22. Rrodriguez Almeida, Zoraida, Reyes Perez, Fernando. Proceso de confirmacion metrologica en una organizacion productiva. Universidda de Pereira, 2010.
23. Hernández León, Rolando Alfredo y Coello González, Saida. *El proceso de la Investigación Científica*. La Habana : Editorial Universitaria del Ministerio de Educación Superior, 2011.
24. Pezet, SF and Mendoza, IJ. Vocabulario internacional de términos fundamentales y generales de metrología. Mexico : CENAM, 1997.

25. ISO, UNEEN. *Sistemas de gestión de las mediciones. Requisitos para los procesos y los equipos de medición.* 2003.
26. Paez, U. *¿Qué es la gestión de Información?* 1990.
27. Dans, Enrique. *Information Management. Reflexiones sobre las tecnologías de la información.* [En línea] 28 de noviembre de 2006. [Citado el: 23 de febrero de 2014.] <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>.
28. Díaz Pérez, Maidelyn, Contreras, Yimian de Liz y Rivero Amador, Soleidys. SciELO. *Características de los sistemas de información que permiten la gestión oportuna de la información y el conocimiento institucional.* [En línea] noviembre de 2009. [Citado el: 23 de febrero de 2014.] [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1024-94352009001100006](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352009001100006). ISSN 1561-2880..
29. Amadeus IT Group. *Amadeus IT Group.* [En línea] 2 de noviembre de 2012. [Citado el: 23 de febrero de 2014.] <http://www.amadeus.com/web/amadeus/es>.
30. *Sistemas de gestion. Atenea Air.* [En línea] enero de 2011. [Citado el: 23 de febrero de 2014.] <http://www.sistemasdegestion.com/index.php/entradas/etiqueta/atenea-air/>.
31. AMIS. *AMIS.* [En línea] abril de 2011. [Citado el: 23 de febrero de 2014.] <http://www.tsc-corp.com.html>.
32. Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, Aureliano David. *Metodología de Desarrollo del oftware.* 2008.
33. Canós, José H., Letelier, Patricio y Penadés, María del Carmen. *Metodologías Ágiles en el Desarrollo de Software.* Valencia : Universidad Politécnica de Valencia.
34. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.* Wesley : Pearson Adisson, 2000.
35. Zeldman, Jeffery. *Diseño con estándares web. Diseño con estándares web.*

36. Magazine., Brandon Satrom.MSDN. Creación de aplicaciones con HTML5: lo que necesita saber. [En línea] agosto de 2011. [Citado el: 23 de febrero de 2014.] <http://msdn.microsoft.com/es-es/magazine/hh335062.aspx..>
37. Freddy Vega, John y Van Der Henst, Christian. Guía de HTML. El presente de la web. [En línea] 2011.
38. Valdés, Damián Pérez. Los diferentes lenguajes de programación para la web. [En línea] 2007. [Citado el: 23 de febrero de 2014.] [http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/.](http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/)
39. Garcia, Javier. *Aprenda Java como si estuviera en primero*. San Sebastian : Escuela Superior de Ingenieros Industriales, 2005.
40. Team., jQuery UI. jQuery User Interface. [En línea] 2010. [Citado el: 23 de febrero de 2014.] <http://jqueryui.com/about..>
41. Sesiolabs. *Symfony and Doctrine with sfDoctrinePlugin*. Sesiolabs. Sesiolabs. 2009.
42. Mata, Manel Pérez. Tecno Retales.Qué es Doctrine ORM? [En línea] 3 de julio de 2009. [Citado el: 23 de febrero de 2014.] <http://www.tecnoretails.com/programacion/que>.
43. Integrado., Entornos de Desarrollo. [En línea] [Citado el: 23 de febrero de 2014.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pd>.
44. Pressman, Roger S. Ingeniería del software. Un enfoque practico. *Ingeniería del software. Un enfoque practico*. .
45. Larman, Craig. *UML y Patrones de*.
46. Erich, Gamma, y otros, y otros. *Patrones de diseño*. s.l. : Pearson E, 2003. ISBN 9788478290598.

## **Anexo1 “Entrevista”**

**¿Para usted cuáles son las principales causas en los accidentes aéreos?**

**Los accidentes técnicos que pueden ocurrir en las aeronaves están clasificados según la razón por la cual ocurre la falla, estas pueden estar desde la fabricación de las piezas hasta el momento en el que fallan por el exceso de uso o condiciones climáticas que no ayudan a los materiales de los diferentes elementos de la aeronave.**

**¿Cómo pueden existir errores en la fabricación de algún equipo?**

Los errores en la fabricación de las piezas de la aeronave pueden estar presentes a pesar de la exactitud con las que están hechas, al ser fabricadas con un proceso de automatización y los diferentes controles por las que pasan antes de ser instaladas.

**¿Cuáles pueden ser las fallas técnicas que menciona anteriormente? ¿Qué medidas se adoptan para prevenirlas?**

En las fallas técnicas están: fallas por desgaste, fallas por fatiga superficial, fallas por fracturas, fallas por flujo plástico.

Para determinar estas fallas previamente debe realizarse una inspección de campo, en esta se debe determinar:

- Localización de las piezas rotas respecto a cada una de las otras.
- Identificación del origen de la falla.
- Orientación y magnitud de los esfuerzos.
- Dirección de propagación de la grieta y secuencia de la falla.
- Presencia de oxidación, colores de temperatura o productos de corrosión.
- Presencia de peligros secundarios no relacionados con la falla principal.

Además, es importante hablar con los operarios ya que pueden suministrar datos o pistas importantes para el posterior análisis, indagar por la presencia de ruidos, vibraciones o temperaturas anormales. Para ello es llevado a cabo las medidas de control en los talleres técnicos de la empresa.

**¿Qué tipos de controles son realizados en los talleres técnicos de la empresa?**

En los talleres técnicos se realizan controles a los equipos de medición, los cuales pertenecen al Departamento de Metrología, puesto que la utilización de los mismos es de suma importancia para la rectificación de los instrumentos del avión.

El departamento se ocupa de gestionar todo el proceso de confirmación al que son sometidos los equipos de medición. Para la empresa, la confirmación metrológica de un equipo de medición es una condición indispensable para su uso.

## Anexo2 “Descripción de Casos de Usos del sistema”

Tabla 2.4.4 Descripción del CU del sistema “Gestionar Área”

CU	Gestionar Área	
Actores	Jefe de Área, Técnico (Inicia): Incluye crear, ver, modificar o eliminar equipo.	
Resumen	El caso de uso se inicia cuando el actor Técnico o Jefe de Área selecciona la opción que le permite realizar una acción sobre un área.	
Precondiciones	Los actores deben estar autenticado en el sistema.	
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción “Área” en la página principal.	
2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Crear.</li> <li>• Ver. Ver sección “Ver Área”.</li> <li>• Editar. Ver sección “Editar Área”.</li> <li>• Eliminar. Ver sección “Eliminar Área”</li> </ul>
3.	Selecciona la opción Crear.	
4.		Brinda la posibilidad de introducir los datos del área. <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Descripción</li> </ul> Permite: <ul style="list-style-type: none"> <li>• Guardar.</li> <li>• Volver al Listado.</li> </ul>
5.	Introduce los datos del área.	
6.	Selecciona la opción de guardar los datos.	
7.		Valida los datos.
8.		Crea la nueva área.
9.		Muestra un mensaje de información “El área se ha insertado exitosamente.”
10.		El caso de uso termina.

Flujo Alternos		
4.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
4.a.1		Regresa a la interfaz anterior.
4.a.2		El caso de uso termina.
7.a Existen datos incompletos		
	Actor	Sistema
7.a.1		Muestra un indicador sobre los campos vacíos.
7.a.2		Regresar al paso 5.
7.b Existen datos incorrectos		
	Actor	Sistema
7.b.1		Muestra un indicador sobre los campos incorrectos.
7.b.2		Regresar al paso 5.
Sección "Ver Área"		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción ver asociada a un área.	
2.		Muestra la pantalla principal del producto con los datos asociados al mismo. Permite: <ul style="list-style-type: none"> <li>• Volver al Listado</li> </ul>
3.	Selecciona la opción volver al listado.	
4.		Muestra la interfaz "Área".
5.		El caso de uso termina.
Sección "Editar Área"		
Flujo básico		
	Actor	Sistema

1.	Selecciona la opción modificar asociada a un producto.	
2.		Muestra los datos editables del área. Permite además: <ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Volver al Listado.</li> </ul>
3.	Modifica los datos que necesita y selecciona la opción guardar.	
4.		Valida los datos.
5.		Muestra el listado de las áreas.
6.		El caso de uso termina.
<b>Flujos alternos</b>		
2.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
2.a.1		Regresa a la interfaz anterior.
2.a.2		El caso de uso termina.
4. a Existen datos incorrectos.		
	Actor	Sistema
4.a.1		Muestra un indicador sobre los datos incorrectos.
4.a.2		Regresa al paso 3.
<b>Sección “Eliminar Área”</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1	Selecciona la opción eliminar asociada a un equipo.	
2		Muestra el mensaje de confirmación “¿Está seguro que desea eliminar este producto? Para confirmarlo teclee la palabra ELIMINAR en la caja de texto.”
3.	Selecciona la opción “Eliminar”	
4.		Elimina el producto.

5.		Muestra el listado con el resto de las áreas registradas en el sistema.
6.		El caso de uso termina.

Tabla 2.4.4.4 Descripción del CU del sistema “Gestionar Personal”

CU	Gestionar Personal	
Actores	Jefe de Calidad (Inicia): Incluye crear, ver, modificar o eliminar equipo.	
Resumen	El caso de uso se inicia cuando el actor Jefe de Calidad selecciona la opción que le permite realizar una acción sobre un personal determinado.	
Precondiciones	El actor debe estar autenticado en el sistema.	
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción “Personal” en la página Administración.	
2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Crear.</li> <li>• Ver. Ver sección “Ver Personal”.</li> <li>• Editar. Ver sección “Editar Personal”.</li> <li>• Eliminar. Ver sección “Eliminar Personal”</li> </ul>
3.	Selecciona la opción Crear.	
4.		Brinda la posibilidad de introducir los datos del personal. <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Usuario</li> <li>• Contraseña</li> <li>• Confirmar contraseña</li> <li>• Correo</li> <li>• Teléfono</li> <li>• Ext</li> <li>• Grupo</li> </ul> Permite:

		<ul style="list-style-type: none"> <li>• Guardar.</li> <li>• Volver al Listado.</li> </ul>
5.	Introduce los datos del personal.	
6.	Selecciona la opción de guardar los datos.	
7.		Valida los datos.
8.		Crea el nuevo personal.
9.		Muestra un mensaje de información “El personal se ha insertado exitosamente.”
10.		El caso de uso termina.
<b>Flujo Alternos</b>		
4.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
4.a.1		Regresa a la interfaz anterior.
4.a.2		El caso de uso termina.
7.a Existen datos incompletos		
	Actor	Sistema
7.a.1		Muestra un indicador sobre los campos vacíos.
7.a.2		Regresar al paso 5.
7.b Existen datos incorrectos		
	Actor	Sistema
7.b.1		Muestra un indicador sobre los campos incorrectos.
7.b.2		Regresar al paso 5.
<b>Sección “Ver Personal”</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1.	Selecciona la opción ver asociada a un personal.	

2.		Muestra la pantalla principal del producto con los datos asociados al mismo. Permite: <ul style="list-style-type: none"> <li>• Volver al Listado</li> </ul>
3.	Selecciona la opción volver al listado.	
4.		Muestra la interfaz "Personal".
5.		El caso de uso termina.

## Sección "Editar Personal"

### Flujo básico

	Actor	Sistema
1.	Selecciona la opción modificar asociada a un producto.	
2.		Muestra los datos editables del personal. Permite además: <ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Volver al Listado.</li> </ul>
3.	Modifica los datos que necesita y selecciona la opción guardar.	
4.		Valida los datos.
5.		Muestra el listado del personal.
6.		El caso de uso termina.

### Flujos alternos

#### 2.a El actor selecciona la opción de Volver al Listado

	Actor	Sistema
2.a.1		Regresa a la interfaz anterior.
2.a.2		El caso de uso termina.

#### 4. a Existen datos incorrectos.

	Actor	Sistema
4.a.1		Muestra un indicador sobre los datos incorrectos.
4.a.2		Regresa al paso 3.

Sección "Eliminar Personal"		
Flujo básico		
	Actor	Sistema
1	Selecciona la opción eliminar asociada a un equipo.	
2		Muestra el mensaje de confirmación "¿Está seguro que desea eliminar este producto? Para confirmarlo teclee la palabra ELIMINAR en la caja de texto."
3.	Selecciona la opción "Eliminar"	
4.		Elimina el personal.
5.		Muestra el listado con el resto del personal registrado en el sistema.
6.		El caso de uso termina.

Tabla 2.4.4.5 Descripción del CU del sistema "Gestionar Capacitación"

CU	Gestionar Capacitación	
Actores	Jefe de Área (Inicia): Incluye crear, ver, modificar o eliminar equipo.	
Resumen	El caso de uso se inicia cuando el actor Jefe de Área selecciona la opción que le permite realizar una acción sobre los cursos de capacitación.	
Precondiciones	El actor debe estar autenticado en el sistema.	
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción "Capacitación" en la página Área.	
2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Crear.</li> <li>• Ver. Ver sección "Ver Capacitación".</li> <li>• Editar. Ver sección "Editar Capacitación".</li> <li>• Eliminar. Ver sección "Eliminar Capacitación"</li> </ul>
3.	Selecciona la opción Crear.	
4.		Brinda la posibilidad de introducir los datos del

		<p>área.</p> <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Fecha de inicio</li> <li>• Fecha de fin</li> <li>• Descripción</li> <li>• Personal Capacitado</li> </ul> <p>Permite:</p> <ul style="list-style-type: none"> <li>• Guardar.</li> <li>• Volver al Listado.</li> </ul>
5.	Introduce los datos de la capacitación.	
6.	Selecciona la opción de guardar los datos.	
7.		Valida los datos.
8.		Crea el nuevo curso de capacitación.
9.		Muestra un mensaje de información "El curso de capacitación se ha insertado exitosamente."
10.		El caso de uso termina.
<b>Flujo Alternos</b>		
4.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
4.a.1		Regresa a la interfaz anterior.
4.a.2		El caso de uso termina.
7.a Existen datos incompletos		
	Actor	Sistema
7.a.1		Muestra un indicador sobre los campos vacíos.
7.a.2		Regresar al paso 5.
7.b Existen datos incorrectos		
	Actor	Sistema
7.b.1		Muestra un indicador sobre los campos

		incorrectos.
7.b.2		Regresar al paso 5.
<b>Sección “Ver capacitación”</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción ver asociada a la página Capacitación.	
2.		Muestra la pantalla principal del producto con los datos asociados al mismo. Permite: <ul style="list-style-type: none"> <li>• Volver al Listado</li> </ul>
3.	Selecciona la opción volver al listado.	
4.		Muestra la interfaz “Capacitación”.
5.		El caso de uso termina.
<b>Sección “Editar Capacitación”</b>		
<b>Flujo básico</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción modificar asociada a un producto.	
2.		Muestra los datos editables del curso de capacitación. Permite además: <ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Volver al Listado.</li> </ul>
3.	Modifica los datos que necesita y selecciona la opción guardar.	
4.		Valida los datos.
5.		Muestra el listado de las capacitaciones.
6.		El caso de uso termina.
<b>Flujos alternos</b>		
2.a El actor selecciona la opción de Volver al Listado		

	Actor	Sistema
2.a.1		Regresa a la interfaz anterior.
2.a.2		El caso de uso termina.
4. a Existen datos incorrectos.		
	Actor	Sistema
4.a.1		Muestra un indicador sobre los datos incorrectos.
4.a.2		Regresa al paso 3.
<b>Sección "Eliminar Capacitación"</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1	Selecciona la opción eliminar asociada a una capacitación.	
2		Muestra el mensaje de confirmación "¿Está seguro que desea eliminar este producto? Para confirmarlo teclee la palabra ELIMINAR en la caja de texto."
3.	Selecciona la opción "Eliminar"	
4.		Elimina el producto.
5.		Muestra el listado con el resto de las capacitaciones registradas en el sistema.
6.		El caso de uso termina.

Tabla 2.4.4 Descripción del CU del sistema "Gestionar Plan de Aseguramiento"

CU	Gestionar Plan de Aseguramiento	
Actores	Jefe de Área (Inicia): Incluye crear, ver, modificar o eliminar equipo.	
Resumen	El caso de uso se inicia cuando el actor Jefe de Área selecciona la opción que le permite realizar una acción sobre el plan de aseguramiento.	
Precondiciones	El actor debe estar autenticado en el sistema.	
<b>Flujo básico</b>		
	Actor	Sistema
1.	Selecciona la opción "Plan" en la página Área.	

2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Crear.</li> <li>• Ver. Ver sección "Ver Plan".</li> <li>• Editar. Ver sección "Editar Plan".</li> <li>• Eliminar. Ver sección "Eliminar Plan"</li> </ul>
3.	Selecciona la opción Crear.	
4.		Brinda la posibilidad de introducir los datos del plan. <ul style="list-style-type: none"> <li>• Fecha de inicio</li> <li>• Fecha de cumplimiento</li> <li>• Nombre</li> <li>• Área</li> <li>• Descripción</li> <li>• Lista de equipos</li> </ul> Permite: <ul style="list-style-type: none"> <li>• Guardar.</li> <li>• Volver al Listado.</li> </ul>
5.	Introduce los datos del plan.	
6.	Selecciona la opción de guardar los datos.	
7.		Valida los datos.
8.		Crea el nuevo plan.
9.		Muestra un mensaje de información "El plan se ha insertado exitosamente."
10.		El caso de uso termina.
<b>Flujo Alternos</b>		
4.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
4.a.1		Regresa a la interfaz anterior.
4.a.2		El caso de uso termina.
7.a Existen datos incompletos		
	Actor	Sistema

7.a.1		Muestra un indicador sobre los campos vacíos.
7.a.2		Regresar al paso 5.
7.b Existen datos incorrectos		
	Actor	Sistema
7.b.1		Muestra un indicador sobre los campos incorrectos.
7.b.2		Regresar al paso 5.
Sección "Ver Plan"		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción ver asociada al plan.	
2.		Muestra la pantalla principal del producto con los datos asociados al mismo. Permite: <ul style="list-style-type: none"> <li>• Volver al Listado</li> </ul>
3.	Selecciona la opción volver al listado.	
4.		Muestra la interfaz "Plan".
5.		El caso de uso termina.
Sección "Editar Plan"		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción modificar asociada a un producto.	
2.		Muestra los datos editables del plan. Permite además: <ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Volver al Listado.</li> </ul>
3.	Modifica los datos que necesita y selecciona la opción guardar.	

4.		Valida los datos.
5.		Muestra el listado de los palnes.
6.		El caso de uso termina.
<b>Flujos alternos</b>		
2.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
2.a.1		Regresa a la interfaz anterior.
2.a.2		El caso de uso termina.
4. a Existen datos incorrectos.		
	Actor	Sistema
4.a.1		Muestra un indicador sobre los datos incorrectos.
4.a.2		Regresa al paso 3.
<b>Sección "Eliminar Plan"</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1	Selecciona la opción eliminar asociada al plan.	
2		Muestra el mensaje de confirmación "¿Está seguro que desea eliminar este producto? Para confirmarlo teclee la palabra ELIMINAR en la caja de texto."
3.	Selecciona la opción "Eliminar"	
4.		Elimina el producto.
5.		Muestra el listado con el resto de los planes registrados en el sistema.
6.		El caso de uso termina.

Tabla 2.4.4.7 Descripción del CU del sistema "Gestionar Clasificador"

<b>CU</b>	Gestionar Clasificador
<b>Actores</b>	Jefe de Calidad (Inicia): Incluye crear, ver, modificar o eliminar equipo.
<b>Resumen</b>	El caso de uso se inicia cuando el actor Jefe de Calidad selecciona la

Precondiciones		opción que le permite realizar una acción sobre un área. El actor debe estar autenticado en el sistema.
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción “Clasificador” en la página Administración.	
2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Crear.</li> <li>• Ver. Ver sección “Ver Clasificador”.</li> <li>• Editar. Ver sección “Editar Clasificador”.</li> <li>• Eliminar. Ver sección “Eliminar Clasificador”</li> </ul>
3.	Selecciona la opción Crear.	
4.		Brinda la posibilidad de introducir los datos del área. <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Padre</li> <li>• Descripción</li> </ul> Permite: <ul style="list-style-type: none"> <li>• Guardar.</li> <li>• Volver al Listado.</li> </ul>
5.	Introduce los datos del clasificador.	
6.	Selecciona la opción de guardar los datos.	
7.		Valida los datos.
8.		Crea el nuevo clasificador.
9.		Muestra un mensaje de información “El clasificador se ha insertado exitosamente.”
10.		El caso de uso termina.
Flujo Alternos		
4.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema

4.a.1		Regresa a la interfaz anterior.
4.a.2		El caso de uso termina.
7.a Existen datos incompletos		
	Actor	Sistema
7.a.1		Muestra un indicador sobre los campos vacíos.
7.a.2		Regresar al paso 5.
7.b Existen datos incorrectos		
	Actor	Sistema
7.b.1		Muestra un indicador sobre los campos incorrectos.
7.b.2		Regresar al paso 5.
Sección "Ver Clasificador"		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción ver asociada a un clasificador.	
2.		Muestra la pantalla principal del producto con los datos asociados al mismo. Permite: <ul style="list-style-type: none"> <li>• Volver al Listado</li> </ul>
3.	Selecciona la opción volver al listado.	
4.		Muestra la interfaz "Clasificador".
5.		El caso de uso termina.
Sección "Editar Clasificador"		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción modificar asociada a un producto.	
2.		Muestra los datos editables del clasificador. Permite además:

		<ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Volver al Listado.</li> </ul>
3.	Modifica los datos que necesita y selecciona la opción guardar.	
4.		Valida los datos.
5.		Muestra el listado de los clasificadores.
6.		El caso de uso termina.
<b>Flujos alternos</b>		
2.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
2.a.1		Regresa a la interfaz anterior.
2.a.2		El caso de uso termina.
4. a Existen datos incorrectos.		
	Actor	Sistema
4.a.1		Muestra un indicador sobre los datos incorrectos.
4.a.2		Regresa al paso 3.
<b>Sección “Eliminar Clasificador”</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1	Selecciona la opción eliminar asociada a un clasificador.	
2		Muestra el mensaje de confirmación “¿Está seguro que desea eliminar este producto? Para confirmarlo teclee la palabra ELIMINAR en la caja de texto.”
3.	Selecciona la opción “Eliminar”	
4.		Elimina el producto.
5.		Muestra el listado con el resto de los clasificadores registrados en el sistema.
6.		El caso de uso termina.

Tabla 2.4.4.8 Descripción del CU del sistema “Gestionar Servicio”

CU	Gestionar Servicio	
Actores	Jefe de Área, Técnico (Inicia): Incluye crear, ver, modificar o eliminar equipo.	
Resumen	El caso de uso se inicia cuando el actor Técnico o Jefe de Área selecciona la opción que le permite realizar una acción sobre un área.	
Precondiciones	Los actores deben estar autenticado en el sistema.	
<b>Flujo básico</b>		
	Actor	Sistema
1.	Selecciona la opción "Servicio" en la página principal.	
2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Ver.</li> <li>• Editar. Ver sección "Editar Servicio".</li> <li>• Eliminar. Ver sección "Eliminar Servicio"</li> </ul>
3.	Selecciona la opción Ver.	
4.		Muestra la pantalla principal del producto con los datos asociados al mismo. Permite: Volver al Listado
5.	Selecciona la opción Volver al Listado.	
6.		Muestra la interfaz "Servicio".
7.		El caso de uso termina.
<b>Sección "Editar Servicio"</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1.	Selecciona la opción modificar asociada a un producto.	
2.		Muestra los datos editables del servicio. Permite además: <ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Volver al Listado.</li> </ul>
3.	Modifica los datos que necesita y	

	selecciona la opción guardar.	
4.		Valida los datos.
5.		Muestra el listado de los servicios.
6.		El caso de uso termina.
<b>Flujos alternos</b>		
2.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
2.a.1		Regresa a la interfaz anterior.
2.a.2		El caso de uso termina.
4. a Existen datos incorrectos.		
	Actor	Sistema
4.a.1		Muestra un indicador sobre los datos incorrectos.
4.a.2		Regresa al paso 3.
<b>Sección “Eliminar Servicio”</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1	Selecciona la opción eliminar asociada a un servicio.	
2		Muestra el mensaje de confirmación “¿Está seguro que desea eliminar este producto? Para confirmarlo teclee la palabra ELIMINAR en la caja de texto.”
3.	Selecciona la opción “Eliminar”	
4.		Elimina el producto.
5.		Muestra el listado con el resto de los servicios registrados en el sistema.
6.		El caso de uso termina.

Tabla 2.4.4 Descripción del CU del sistema “Gestionar Laboratorio”

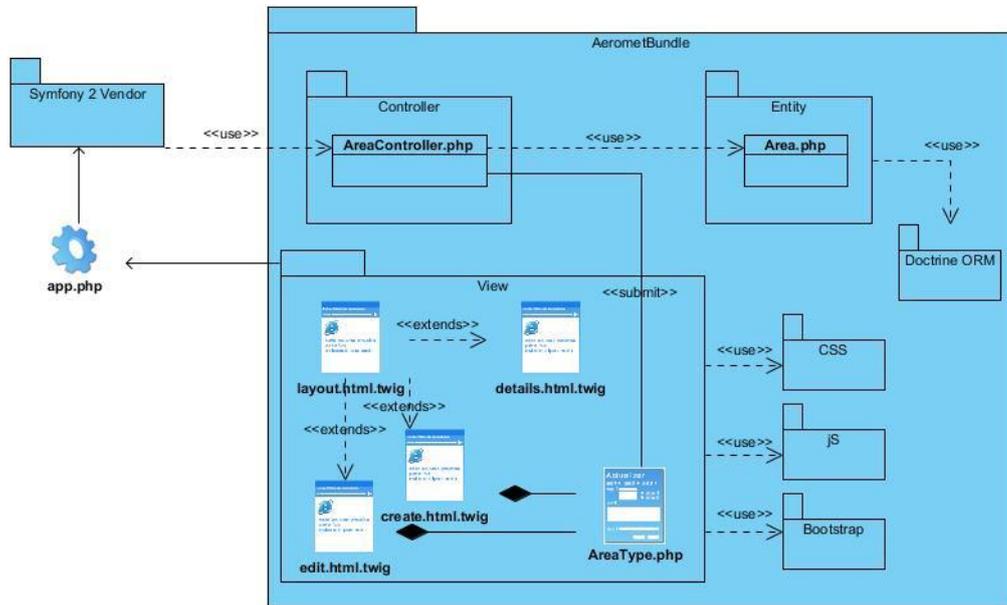
CU	Gestionar Laboratorio
----	-----------------------

Actores	Jefe de Área (Inicia): Incluye crear, ver, modificar o eliminar equipo.	
Resumen	El caso de uso se inicia cuando el actor Jefe de Área selecciona la opción que le permite realizar una acción sobre un laboratorio.	
Precondiciones	El actor debe estar autenticado en el sistema.	
<b>Flujo básico</b>		
	Actor	Sistema
1.	Selecciona la opción "Laboratorio" en la página principal.	
2.		Brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> <li>• Crear.</li> <li>• Ver. Ver sección "Ver Laboratorio".</li> <li>• Editar. Ver sección "Editar Laboratorio".</li> <li>• Eliminar. Ver sección "Eliminar Laboratorio"</li> </ul>
3.	Selecciona la opción Crear.	
4.		Brinda la posibilidad de introducir los datos del área. <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Tipo de laboratorio</li> <li>• Descripción</li> </ul> Permite: <ul style="list-style-type: none"> <li>• Guardar.</li> <li>• Volver al Listado.</li> </ul>
5.	Introduce los datos del laboratorio.	
6.	Selecciona la opción de guardar los datos.	
7.		Valida los datos.
8.		Crea el nuevo laboratorio.
9.		Muestra un mensaje de información "El clasificador se ha insertado exitosamente."
10.		El caso de uso termina.
<b>Flujo Alternos</b>		
4.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema

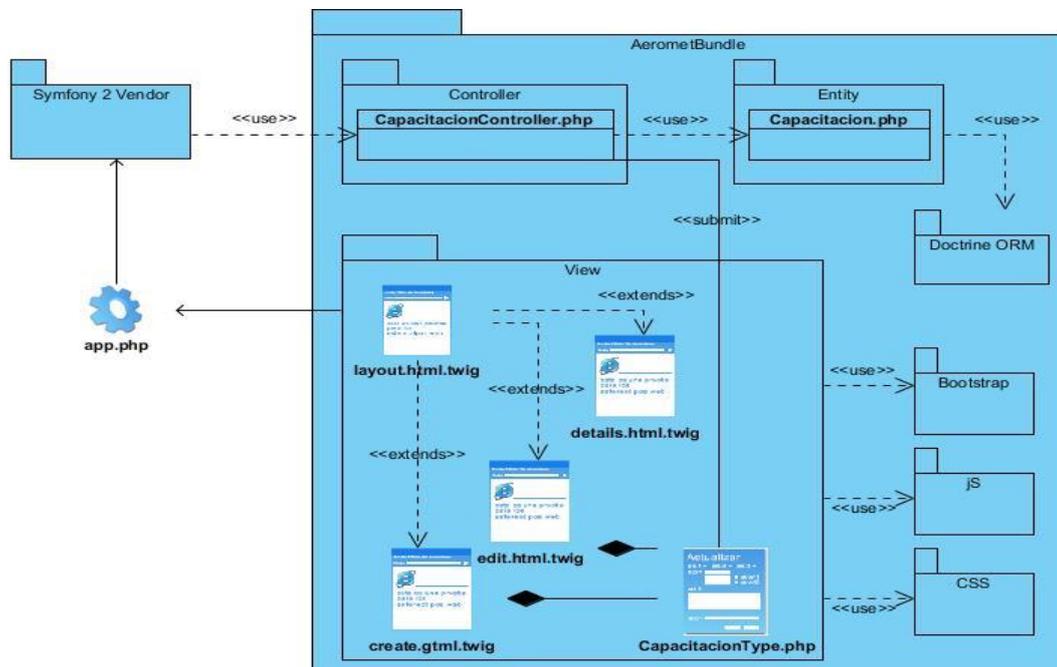
4.a.1		Regresa a la interfaz anterior.
4.a.2		El caso de uso termina.
7.a Existen datos incompletos		
	Actor	Sistema
7.a.1		Muestra un indicador sobre los campos vacíos.
7.a.2		Regresar al paso 5.
7.b Existen datos incorrectos		
	Actor	Sistema
7.b.1		Muestra un indicador sobre los campos incorrectos.
7.b.2		Regresar al paso 5.
Sección "Ver Laboratorio"		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción ver asociada a un laboratorio.	
2.		Muestra la pantalla principal del producto con los datos asociados al mismo. Permite: <ul style="list-style-type: none"> <li>• Volver al Listado</li> </ul>
3.	Selecciona la opción volver al listado.	
4.		Muestra la interfaz "Laboratorio".
5.		El caso de uso termina.
Sección "Editar Laboratorio"		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción modificar asociada a un producto.	
2.		Muestra los datos editables del laboratorio. Permite además:

		<ul style="list-style-type: none"> <li>• Guardar los datos.</li> <li>• Volver al Listado.</li> </ul>
3.	Modifica los datos que necesita y selecciona la opción guardar.	
4.		Valida los datos.
5.		Muestra el listado de los laboratorios.
6.		El caso de uso termina.
<b>Flujos alternos</b>		
2.a El actor selecciona la opción de Volver al Listado		
	Actor	Sistema
2.a.1		Regresa a la interfaz anterior.
2.a.2		El caso de uso termina.
4. a Existen datos incorrectos.		
	Actor	Sistema
4.a.1		Muestra un indicador sobre los datos incorrectos.
4.a.2		Regresa al paso 3.
<b>Sección “Eliminar Laboratorio”</b>		
<b>Flujo básico</b>		
	Actor	Sistema
1	Selecciona la opción eliminar asociada a un laboratorio.	
2		Muestra el mensaje de confirmación “¿Está seguro que desea eliminar este producto? Para confirmarlo teclee la palabra ELIMINAR en la caja de texto.”
3.	Selecciona la opción “Eliminar”	
4.		Elimina el producto.
5.		Muestra el listado con el resto de los laboratorios registrados en el sistema.
6.		El caso de uso termina.

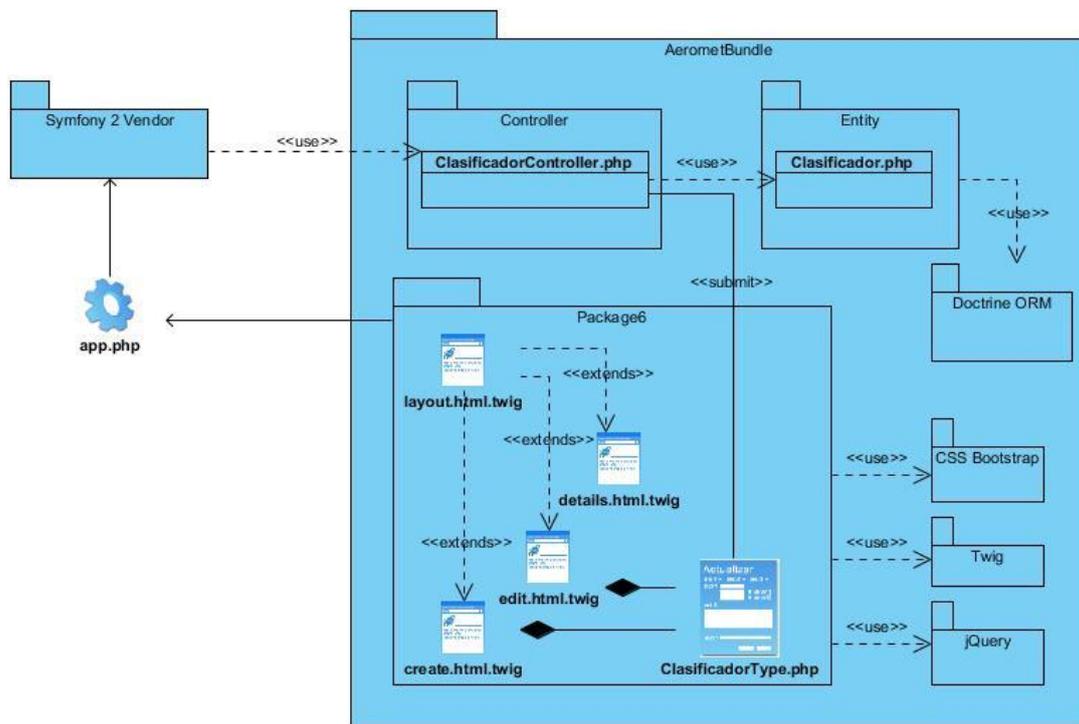
## Anexo3 “Diagramas de clases del diseño”



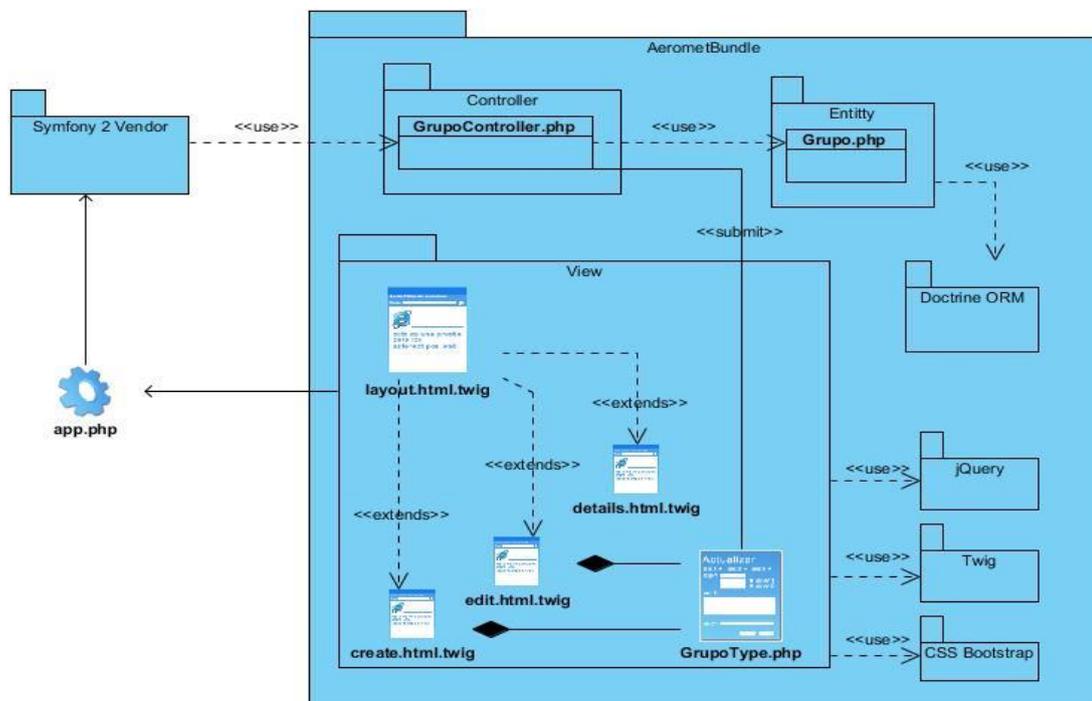
## DCD CU\_Gestionar Area



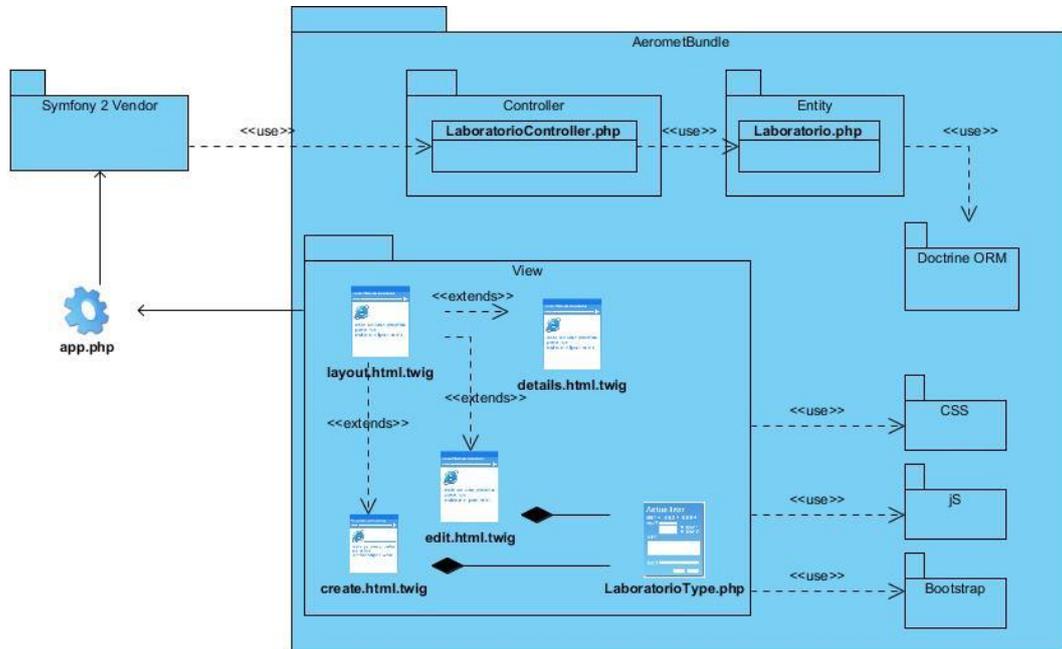
## DCD CU\_Gestionar Capacitacion



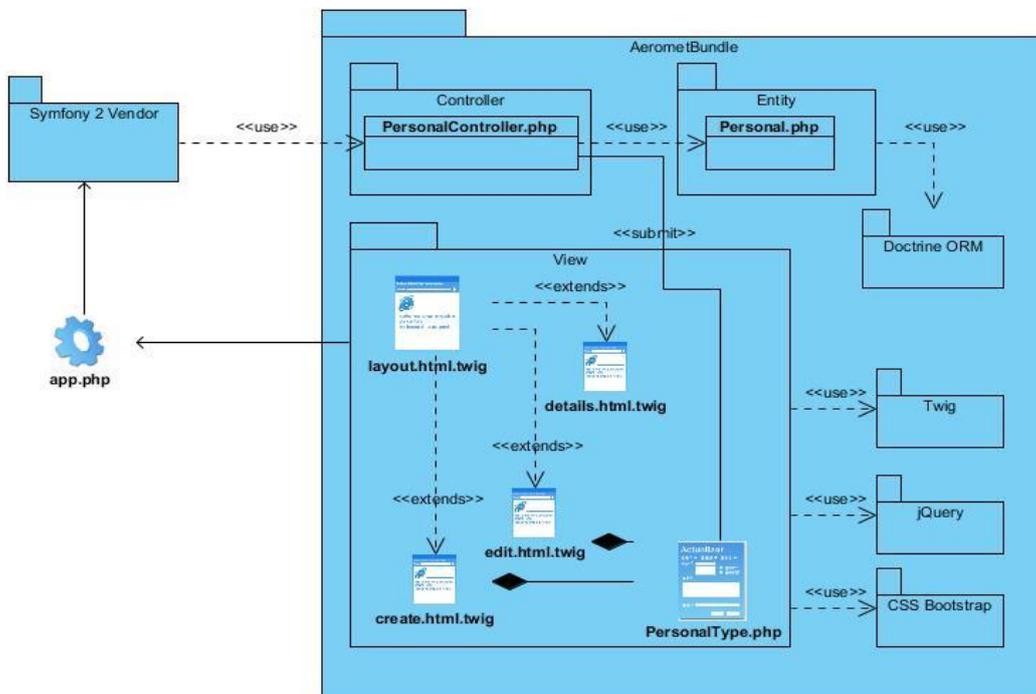
DCD CU\_Gestionar Clasificador



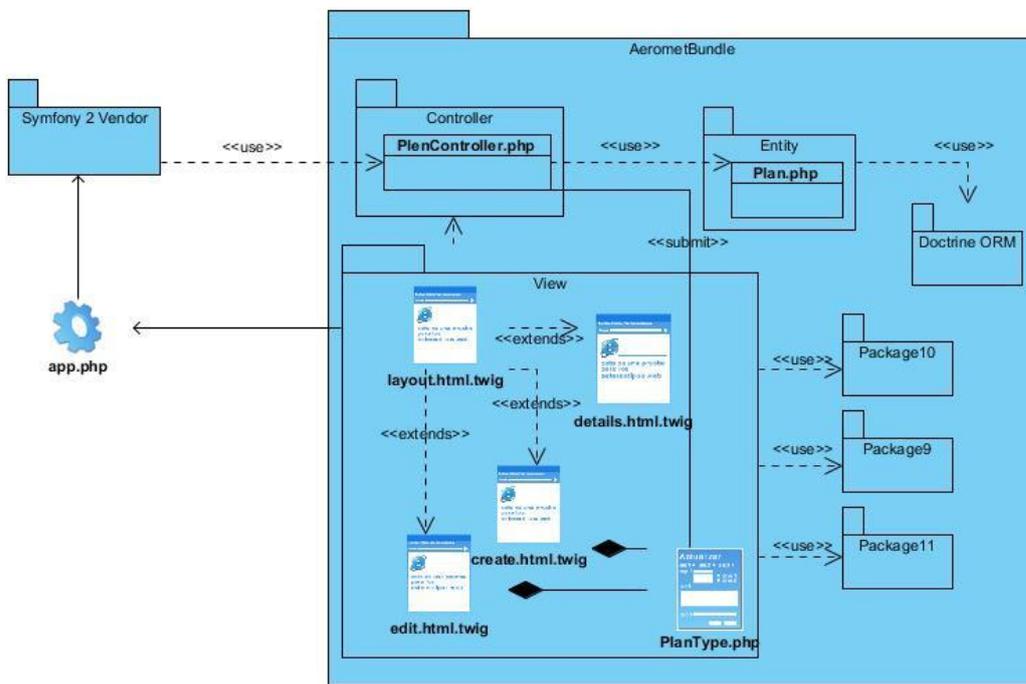
DCD CU\_Gestionar Grupo



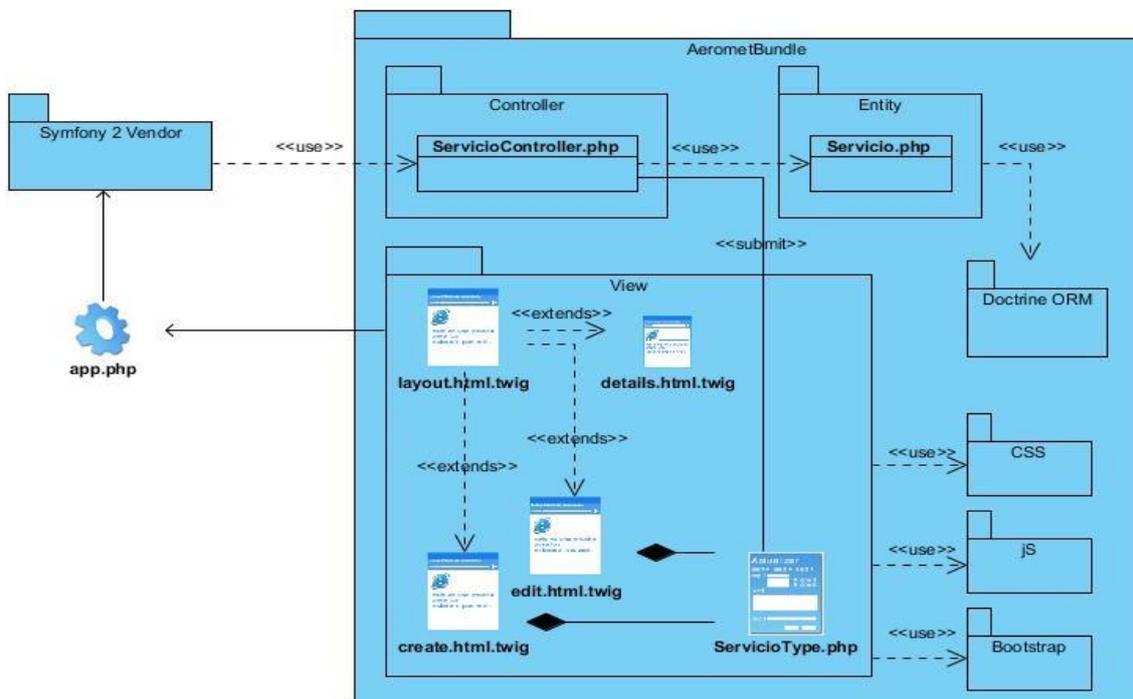
DCD CU\_Gestionar Laboratorio



DCD CU\_Gestionar Personal



DCD CU\_Gestionar Plan



DCD CU\_Gestionar Servicio

## Anexo4 “Descripción de las tablas de la Base de Datos”

Tabla área				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>id</u>	int(11)	PK	No	Almacena el id del área, único en toda la tabla.
nombre	varchar(255)		No	Almacena el nombre del área.
descripción	varchar(255)		No	Almacena la descripción del área.

Tabla capacitación				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>id</u>	int(11)	PK	No	Almacena el id de la capacitación, único en toda la tabla.
nombre	varchar(255)		No	Almacena el nombre de capacitación.
descripción	varchar(255)		No	Almacena la descripción de capacitación.
creador_id	int(11)		Sí	Almacena el id del creador de capacitación.
fecha_inicio	date		No	Almacena la fecha de inicio de la capacitación.
fecha_fin	date		No	Almacena la fecha de fin de la capacitación.

area_id	int(11)	date		Sí	Almacena el id del área de a la que pertenece la capacitación.
---------	---------	------	--	----	--

Tabla capacitación_personal						
Nombre de columna	de	Tipo de Datos	de	PK / FK	Anulable	Documentación
<u>capacitación_id</u>		int(11)		PK	No	Almacena el id de capacitación, único en toda la tabla.
<u>personal_id</u>		int(11)		PK	No	Almacena el id del personal, único en toda la tabla.

Tabla clasificador						
Nombre de columna	de	Tipo de Datos	de	PK / FK	Anulable	Documentación
<u>id</u>		int(11)		PK	No	Almacena el id del clasificador, único en toda la tabla.
tipo_id		int(11)			Sí	Almacena el tipo de clasificador.
nombre		varchar(255)			No	Almacena el nombre del clasificador.
descripción		Varchar(255)			Sí	Almacena la descripción del clasificador.

Tabla evaluación						
Nombre de columna	de	Tipo de Datos	de	PK / FK	Anulable	Documentación
<u>id</u>		int(11)		PK	No	Almacena el id de la evaluación, único en toda la tabla.

equipo_id	int(11)		Sí	Almacena el id del equipo a evaluar.
servicio_id	int(11)		Sí	Almacena el id del servicio a evaluar.
tipo_evaluacion_servicio	int(11)		Sí	Almacena el tipo de servicio a evaluar.
fecha_evaluacion	date		No	Almacena la fecha de evaluación.

Tabla grupo				
Nombre de columna	Tipo de Datos	de PK / FK	Anulable	Documentación
<u>id</u>	int(11)	PK	No	Almacena el id del grupo, único en toda la tabla.
nombre	varchar(30)		No	Almacena el nombre del grupo.
descripción	varchar(250)		No	Almacena la descripción del grupo.
rol	varchar(20)		NO	Almacena el rol del grupo.

Tabla laboratorio				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>id</u>	int(11)	PK	No	Almacena el id del laboratorio, único en toda la tabla.
tipo_laboratorio	int(11)		Sí	Almacena el tipo de laboratorio.
nombre	varchar(255)		No	Almacena el nombre del laboratorio.
descripción	varchar(255)			Almacena la descripción del laboratorio.

Tabla laboratorio_personal				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>laboratrio_id</u>	int(11)	PK	No	Almacena el id del laboratorio, único en toda la tabla.
<u>personal_id</u>	int(11)	PK	No	Almacena el id del personal, único en toda la tabla.

Tabla personal				
Nombre de columna	Tipo de Datos	PK / FK	Anula ble	Documentación
<u>id</u>	int(11)	PK	No	Almacena el id del laboratorio, único en toda la tabla.
usuario	varchar(30)		No	Almacena el nombre de usuario del personal.
contraseña	varchar(40)		No	Almacena la contraseña del personal.
correo	varchar(60)		No	Almacena la dirección de correo del personal.
teléfono	varchar(60)		No	Almacena el teléfono del personal.
ext	varchar(60)		No	Almacena la extensión del personal.
nombre_	varchar(40)		No	Almacena el nombre del personal.

Tabla personal_grupo				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación

<u>grupo_id</u>	int(11)	PK	No	Almacena el id del grupo, único en toda la tabla.
<u>personal_id</u>	int(11)	PK	No	Almacena el id del personal, único en toda la tabla.

Tabla plan				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>id</u>	int(11)	PK	No	Almacena el id del plan, único en toda la tabla.
nombre	varchar(255)		No	Almacena el nombre del plan, único en toda la tabla.
descripción	varchar(255)		No	Almacena la descripción del plan.
fecha_cumplimiento	date		No	Almacena la fecha de cumplimiento del plan.
area_id	int(11)		Sí	Almacena el id del área a la que pertenece el plan.
fecha_inicio	date		No	Almacena la fecha de inicio del plan.

Tabla plan_equipo				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>plan_id</u>	int(11)	PK	No	Almacena el id del plan, único en toda la tabla.
<u>equipo_id</u>	int(11)	PK	No	Almacena el id del equipo, único en toda la tabla.

Tabla procesomedicion_personal				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>procmedicacion_id</u>	int(11)	PK	No	Almacena el id del proceso de medición, único en toda la tabla.
<u>personal_id</u>	int(11)	PK	No	Almacena el id del personal, único en toda la tabla.

Tabla procmedicacion				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>id</u>	int(11)	PK	No	Almacena el id del proceso de medición, único en toda la tabla.

Tabla procmedicacion_equipo				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación
<u>procmedicacion_id</u>	int(11)	PK	No	Almacena el id del proceso de medición, único en toda la tabla.
<u>equipo_id</u>	int(11)	PK	No	Almacena el id del equipo, único en toda la tabla.

Tabla servicio				
Nombre de columna	Tipo de Datos	PK / FK	Anulable	Documentación

<u>id</u>	int(11)	PK	No	Almacena el id del servicio, único en toda la tabla.
tipo_servicio	int(11)		Sí	Almacena el tipo de servicio.
tipo_parametro	int(11)		Sí	Almacena el parámetro del equipo.
equipo_id	int(11)		Sí	Almacena el id del equipo que recibe el servicio.
fecha_ejecución	date		No	Almacena la fecha de ejecución del servicio.
duración	int(11)		No	Almacena el tiempo de duración del equipo.
nombre_proveedor	Varchar(255)		No	Almacena el nombre del proveedor del servicio.
Descripción	Varchar(255)		Sí	Almacena la descripción del servicio.
valor_desvio	Decimal(10,0)		Sí	Almacena el valor de desvío.
area_id	int(11)		No	Almacena el id del área a la que pertenece el equipo que recibe servicio.
laboratorio_id	int(11)		No	Almacena el id del laboratorio que brinda el servicio.
evaluación	Decimal(10,0)		No	Almacena la descripción del servicio.

## Anexo5 “Descripción de las tablas de la Base de Datos”

Escenario	Descripción	Nombre	Descripción	Respuesta del sistema	Flujo central
<b>EC 4.1</b> <b>Crear</b> <b>área.</b>	El actor selecciona la opción: Crear.	V	V	Brinda la posibilidad de introducir los datos del equipo:  - Nombre  - Descripción  Y permite:  - Guardar  - Volver al listado	Área/Crear.
<b>EC 4.1.1</b>	El actor selecciona la opción: Guardar.			Valida los datos.  Crea la nueva área.  Muestra un mensaje de información “El área se ha insertado exitosamente”.	Área/Crear/Guardar.
<b>EC 4.1.2</b>	El actor selecciona la opción: Volver al Listado.			Regresa a la interfaz anterior.	Área/Crear/Volver al listado
<b>EC 4.1.3</b>	Existen datos incompletos.			Muestra un indicador sobre los campos vacíos.	Área/Crear/Guardar.

<b>EC 4.1.4</b>	Existen datos incorrectos.			Muestra un indicador sobre los campos incorrectos.	Área/Crear/Guardar.
<b>EC 4.2 Ver</b>	El actor selecciona la opción: Ver			Muestra los detalles asociados al área. Permite: - Volver al Listado	Área/Ver.
<b>EC 4.2.1</b>	El actor selecciona la opción: Volver al listado.			Muestra la interfaz Área.	Área/Ver/Volver al Listado.
<b>EC 4.3 Editar</b>	Selecciona la opción: Editar.	N A	N A	Muestra los datos editables del área. Permite: - Guardar - Volver al Listado	Área/Editar
<b>EC 4.3.1</b>	El actor selecciona la opción: Guardar.			Valida los datos. Muestra el listado de las áreas.	Área/Editar/Guardar
<b>EC 4.3.2</b>	El actor selecciona la opción: Volver al Listado.			Regresa a la interfaz anterior.	Área/Editar/Volver al Listado.
<b>EC 4.3.3</b>	Existen datos incorrectos.			Muestra un indicador sobre los datos incorrectos.	Área/Editar/Guardar
<b>EC 4.4 Eliminar</b>	El actor selecciona la			Muestra el mensaje de confirmación ¿Esta seguro	Área/Ver/Eliminar

	opción: Eliminar			que desea eliminar este producto?  Para confirmarlo selecciona la opción Aceptar.	Área/Editar/Eliminar
<b>EC 4.4.1</b>	Selecciona la opción Aceptar.			Elimina el área.  Muestra el listado con el resto de las áreas registradas en el sistema.	Área/Ver/Eliminar/Aceptar.  Área/Editar/Eliminar/Aceptar.

## Diseño CP CU Gestionar Personal

Escenario	Descripción	Nombre	Usuario	Contraseña	Correo	Grupo	Respuesta del sistema	Flujo central
<b>EC 4.1 Crear personal.</b>	El actor selecciona la opción: Crear.	V	V	V	V	V	Brinda la posibilidad de introducir los datos del personal:  <ul style="list-style-type: none"> <li>- Nombre</li> <li>- Usuario</li> <li>- Contraseña</li> <li>- Correo</li> <li>- Grupo</li> </ul> Y permite:  <ul style="list-style-type: none"> <li>- Guardar</li> <li>- Volver al</li> </ul>	Administración/Personal/Crear.

							listado	
<b>EC 4.1.1</b>	El actor selecciona la opción: Guardar.						<p>Valida los datos.</p> <p>Crea el nuevo personal.</p> <p>Muestra un mensaje de información "El personal se ha insertado exitosamente".</p>	Administración/Personal/Crear/Guardar.
<b>EC 4.1.2</b>	El actor selecciona la opción: Volver al Listado.						Regresa a la interfaz anterior.	Administración/Personal/Crear/Volver al listado
<b>EC 4.1.3</b>	Existen datos incompletos.						Muestra un indicador sobre los campos vacíos.	Administración/Personal/Crear/Guardar.
<b>EC 4.1.4</b>	Existen datos incorrectos.						Muestra un indicador sobre los campos	Administración/Personal/Crear/Guardar.

	os.						incorrectos.	
<b>EC Ver</b>	<b>4.2</b>	El actor selecciona la opción: Ver					Muestra los detalles asociados al personal. Permite: - Volver al Listado	Administración/Personal/Ver.
<b>EC 4.2.1</b>		El actor selecciona la opción: Volver al listado.					Muestra la interfaz Personal.	Administración/Personal/Ver/Volver al Listado.
<b>EC Editar</b>	<b>4.3</b>	Selecciona la opción: Editar.	N A	N A	N A	N A	Muestra los datos editables del personal. Permite: - Guardar - Volver al Listado	Administración/Personal/Editar
<b>EC 4.3.1</b>		El actor selecciona la opción: Guardar.					Valida los datos. Muestra el listado del personal.	Administración/Personal/Editar/Guardar
<b>EC 4.3.2</b>		El actor selecciona la opción:					Regresa a la interfaz anterior.	Administración/Personal/Editar/Volver al Listado.

	Volver al Listado.							
<b>EC 4.3.3</b>	Existen datos incorrectos.						Muestra un indicador sobre los datos incorrectos.	Administración/Personal/Editar/Guardar
<b>EC 4.4 Eliminar</b>	El actor selecciona la opción: Eliminar						Muestra el mensaje de confirmación ¿Esta seguro que desea eliminar este producto?  Para confirmarlo selecciona la opción Aceptar.	Administración/Personal/Ver/Eliminar  Administración/Personal/Editar/Eliminar
<b>EC 4.4.1</b>	Selecciona la opción Aceptar.						Elimina el personal.  Muestra el listado con el resto del personal registrado en el sistema.	Administración/Personal/Ver/Eliminar/Aceptar.  Administración/Personal/Editar/Eliminar/Aceptar.

## Diseño de CP CU Gestionar Capacitación

Escenario	Descripción	Nombre	Fecha inicio	Fecha fin	Descripción	Personal	Respuesta del sistema	Flujo central
-----------	-------------	--------	--------------	-----------	-------------	----------	-----------------------	---------------

<p><b>EC 4.1</b> <b>Crear capacitación.</b></p>	<p>El actor selecciona la opción: Crear.</p>	V	V	V	V	V	<p>Brinda la posibilidad de introducir los datos de la capacitación:</p> <ul style="list-style-type: none"> <li>- Nombre</li> <li>- Fecha inicio</li> <li>- Fecha fin</li> <li>- Descripción</li> <li>- Personal Capacitado</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>- Guardar</li> <li>- Volver al listado</li> </ul>	<p>Área/Capacitación/Crear.</p>
<p><b>EC 4.1.1</b></p>	<p>El actor selecciona la opción: Guardar.</p>						<p>Valida los datos. Crea la nueva capacitación. Muestra un mensaje de información "El personal se ha insertado exitosamente".</p>	<p>Área/Capacitación/Crear/Guardar.</p>
<p><b>EC 4.1.2</b></p>	<p>El actor selecciona la opción: Volver al Listado.</p>						<p>Regresa a la interfaz anterior.</p>	<p>Área/Capacitación/Crear/Volver al listado</p>

<b>EC 4.1.3</b>	Existen datos incompletos.						Muestra un indicador sobre los campos vacíos.	Área/Capacitación/Crear/Guardar.
<b>EC 4.1.4</b>	Existen datos incorrectos.						Muestra un indicador sobre los campos incorrectos.	Área/Capacitación/Crear/Guardar.
<b>EC 4.2 Ver</b>	El actor selecciona la opción: Ver						Muestra los detalles asociados al personal.  Permite: - Volver al Listado	Área/Capacitación/Ver.
<b>EC 4.2.1</b>	El actor selecciona la opción: Volver al listado.						Muestra la interfaz Capacitación.	Área/Capacitación/Ver/Volver al Listado.
<b>EC 4.3 Editar</b>	Selecciona la opción: Editar.	N A	N A	N A	N A	N A	Muestra los datos editables de la capacitación.  Permite: - Guardar - Volver al Listado	Área/Capacitación/Editar
<b>EC 4.3.1</b>	El actor selecciona la						Valida los datos.  Muestra el listado de	Área/Capacitación/Editar/Guardar

	opción: Guardar.					las capacitaciones.	
<b>EC 4.3.2</b>	El actor selecciona la opción: Volver al Listado.					Regresa a la interfaz anterior.	Área/Capacitación/Editar/Volver al Listado.
<b>EC 4.3.3</b>	Existen datos incorrectos.					Muestra un indicador sobre los datos incorrectos.	Área/Capacitación/Editar/Guardar
<b>EC 4.4 Eliminar</b>	El actor selecciona la opción: Eliminar					Muestra el mensaje de confirmación ¿Esta seguro que desea eliminar este producto?  Para confirmarlo selecciona la opción Aceptar.	Área/Capacitación/Ver/Eliminar Área/Capacitación/Editar/Eliminar
<b>EC 4.4.1</b>	Selecciona la opción Aceptar.					Elimina la capacitación.  Muestra el listado con el resto de las capacitaciones registradas en el sistema.	Área/Capacitación/Ver/Eliminar/Aceptar. Área/Capacitación/Editar/Eliminar/Aceptar.