

**Universidad de las Ciencias Informáticas**  
**Facultad 6**



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

**Módulo de gestión de activos fijos, medios y útiles para el Sistema de Gestión de Expedientes de Control del Capital Humano de la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas.**

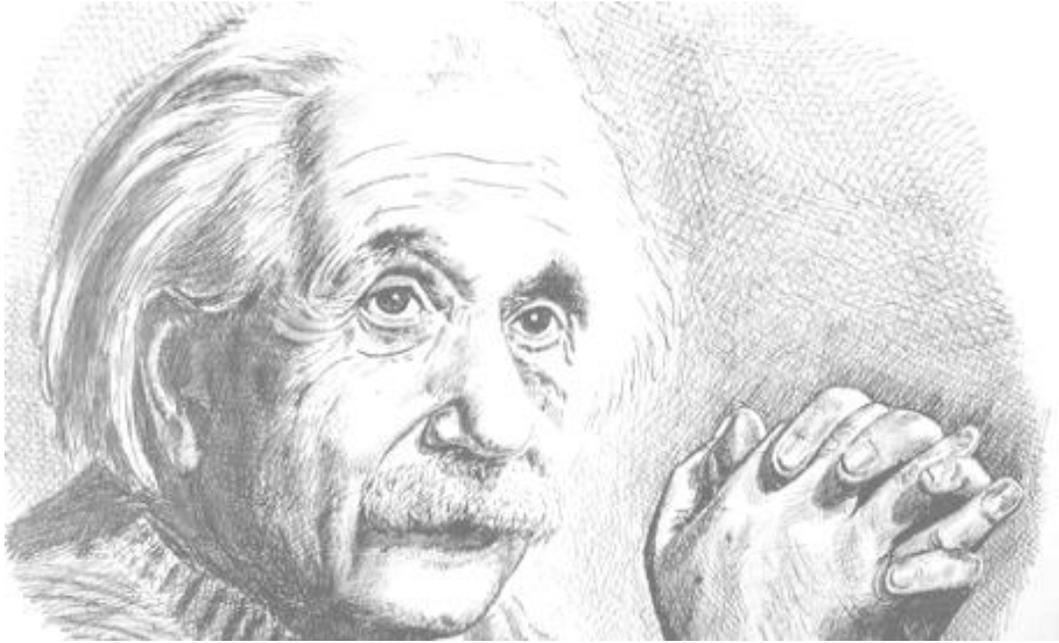
**Autora:** Yaima Hernández de la Oliva

**Tutores:** Ing. Yosvany Marco Pérez

Ing. Luanner Kerton Martínez

**La Habana, 2016**

**“Año 58 de la Revolución”**



*"Si buscas resultados distintos, no hagas siempre lo mismo."*

*A. Einstein*

## DECLARACIÓN DE AUTORÍA

Declaro ser autora del trabajo **“Módulo de gestión de activos fijos, medios y útiles para el Sistema de Gestión de Expedientes de Control del Capital Humano de la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas”** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Yaima Hernández de la Oliva**

\_\_\_\_\_

Firma del Autor

**Ing. Yosvany Marco Pérez**

\_\_\_\_\_

Firma del Tutor

**Ing. Luanner Kerton Martínez**

\_\_\_\_\_

Firma del Tutor

## DATOS DE CONTACTO

Tutor:

- ✓ Ing. Yosvany Marco Pérez
- ✓ Universidad de las Ciencias Informáticas.
- ✓ e-mail: ymarco@uci.cu

Graduado de Ingeniero en Ciencias Informáticas en julio de 2014. Se ha desempeñado como profesor de Teleinformática, Redes y Seguridad Informática en el Departamento de Programación y Señales Digitales de la Facultad 6. Tiene publicaciones y participación en eventos nacionales e internacionales en los que ha presentado trabajos relacionados con el desarrollo de Software y Aplicaciones Informáticas.

Tutor:

- ✓ Ing. Luanner Kerton Martínez
- ✓ Universidad de las Ciencias Informáticas.
- ✓ e-mail: luanner@uci.cu

Graduado de Ingeniero en Ciencias Informáticas en junio de 2012. Se ha desempeñado como programador, analista, arquitecto y líder de proyectos de investigación en proyectos de gestión de información del Centro de Informática Industrial. Ha trabajado en el desarrollo de arquitecturas empresariales y orquestación de servicios. Actualmente se desempeña como desarrollador de interfaces para los sistemas SCADA en el Centro de Informática Industrial de la Universidad de las Ciencias Informáticas.

*A mis abuelos Cati y Cusito.*

## *Agradecimientos*

*A mis tutores Yosvany y Luanner por aceptarme como su estudiante y porque sin ellos esta tesis sería imposible;*

*A mis profesores por su contribución incondicional en mi desarrollo profesional.*

*A Mis colegas, amigos y compañeros de estudio quienes a lo largo de estos años se convirtieron en el soporte indispensable para llegar a la meta.*

*A mi familia, por confiar en que este día llegaría, especialmente a mis tastas Lesther y Jandy.*

*Sinceramente quisiera extender mi más grato agradecimiento a todas las personas que de una u otra forma colaboraron para que pudiera culminar esta meta tan importante en mi vida profesional.*

La gestión del Capital Humano es un recurso importante dentro del entorno corporativo, contribuyendo significativamente al éxito empresarial. Como parte de la estrategia para automatizar esta gestión, en 2012 la Dirección de Gestión Tecnológica (DGT) de la Universidad de las Ciencias Informáticas desarrolló el Sistema de Gestión de Expedientes de Control del Capital Humano (SIGEX), dedicado al control de la documentación de cada trabajador, sus evaluaciones y aprobaciones por parte de superiores. Sin embargo esta herramienta no cuenta con funcionalidades que permita el manejo de la información sobre los medios asignados a los trabajadores de la DGT. La presente investigación tiene como objetivo el desarrollo de un módulo en el sistema SIGEX para la gestión de activos fijos, útiles y otros medios asignados a cada trabajador. Para guiar el desarrollo del módulo se utilizó como metodología OpenUP, PHP y Java Script como lenguajes de programación, Symfony como marco de trabajo, NetBeans como Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) y PostgreSQL como gestor de base de datos. El desarrollo del módulo de gestión de activos fijos, útiles y otros medios, le da la capacidad al Sistema SIGEX de realizar las asignaciones y solicitudes de entrega de los mismos, permitiendo controlar las responsabilidades materiales sobre los medios entregados a los trabajadores y la accesibilidad a dicha información. La DGT cuenta ahora con un sistema más completo que permite una mejor gestión de su Capital Humano.

Palabras claves: Capital Humano, DGT, SIGEX, medio, activo fijo, útil.

**Abstract**

The management of Human Capital is an important resource within the corporative environment, significantly contributing to business success. As part of the strategy to automate such management, in the year 2012 The Technological Management Office (DGT by its acronyms in Spanish) from the University of Informatics Sciences developed The Human Capital Control Files Management System (SIGEX by its acronyms in Spanish), dedicated to control the documentation of every worker, their evaluations and approvals by their superiors. However, this tool has no functionalities that allow handling information about the resources assigned to the workers from the DGT. The present research has as objective to develop a module in SIGEX system to manage fixed assets, resources and other materials given to every worker. To lead the development of the module it was used OpenUP as methodology, PHP and Java Script as programming languages, Symfony as framework, NetBeans as Integrated Development Environment (IDE) and PostgreSQL as databasae management system. The development of the development of the module to manage fixed assets, resources and other materials, provides the SIGEX System the capacity to make assignments and request the delivery of them, allowing to control the material responsibilities over the resources given to the workers and the accesibility to such information. The DGT counts today with a more completed system that allows better management of their Human Capital.

Key Words: Human Capital, DGT, SIGEX, resource, fixed asset, material.

Contenido

**Introducción .....1**

**Capítulo 1 Fundamentos teóricos de la gestión de activos fijos, medios y útiles .....4**

1.1 Marco teórico .....4

1.1.1 Medios empresariales.....4

1.1.2 Activos empresariales.....5

1.1.3 Activos fijos.....5

1.1.4 Gestión administrativa de medios .....6

1.2 Sistemas automatizados para el control de medios .....7

1.2.1 SAP R/3.....7

1.2.2 OpenBravo ERP .....7

1.2.3 SAP One.....8

1.2.4 Versat .....8

1.2.5 ASSETS NS .....8

1.2.6 Conclusión de los sistemas analizados.....9

1.3 Metodología, tecnologías y herramientas de desarrollo .....9

1.3.1 Metodología de desarrollo de software .....10

1.3.2 Lenguaje de modelado unificado (UML 2.0).....12

1.3.3 Lenguaje de programación .....12

1.3.4 Marco de trabajo Symfony 1.4.11 .....13

1.3.5 Sistema Gestor de Base de Datos PostgreSQL 9.1.2.....14

1.3.6 IDE NetBeans 8.0 .....14

1.3.7 Herramienta CASE Visual Paradigm 5.1.....15

1.3.8 Servidor Web Apache 2.0 .....15

1.4 Conclusiones parciales. ....15

**Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles .....16**

2.1 Modelación de dominio .....16

2.1.1	Información a manipular .....	18
2.2	Descripción de la propuesta de solución.....	18
2.2.1	Requisitos funcionales .....	19
2.2.2	Requisitos no funcionales .....	21
2.2.3	Actores del sistema .....	24
2.2.4	Diagrama de casos de uso del sistema .....	24
2.2.5	Especificaciones de caso de uso .....	25
2.3	Arquitectura .....	30
2.4	Patrones de diseño .....	32
2.5	Paquetes y relaciones.....	33
2.6	Diagrama de clases de diseño .....	34
2.7	Diseño físico de la base de datos .....	36
2.8	Diagrama de despliegue .....	39
2.9	Conclusiones parciales .....	40
<b>Capítulo 3 Implementación y pruebas del módulo de gestión de activos fijos, medios y útiles.....</b>		<b>41</b>
3.1	Modelo de implementación .....	41
3.1.1	Interfaz de la aplicación .....	41
3.1.2	Estilo de programación .....	41
3.1.3	Gestión de errores .....	42
3.1.4	Diagrama de componentes.....	43
3.2	Diseño de pruebas.....	44
3.2.1	Pruebas de integración .....	44
3.2.2	Pruebas de aceptación .....	47
3.3	Conclusiones del capítulo. ....	49
<b>Conclusiones generales.....</b>		<b>50</b>
<b>Recomendaciones .....</b>		<b>51</b>
<b>Referencias bibliográficas .....</b>		<b>52</b>

**Índice de Figuras**

Fig 1.	Modelo de dominio .....	18
Fig 2.	Diagrama de casos de uso del sistema .....	25
Fig 3.	Uso del patrón arquitectónico en el módulo propuesto .....	31
Fig 4.	Diagrama de paquetes del módulo propuesto .....	34
Fig 5.	Diagrama de Clases del Diseño caso de uso Gestionar Útil.....	35
Fig 6.	Diagrama de clases del Diseño caso de uso Solicitar asignación de Activo Fijo .....	36
Fig 7.	Diagrama Entidad-Relación del módulo propuesto.....	37
Fig 8.	Diagrama de despliegue .....	39
Fig 9.	Tratamiento de errores en la captura de datos.....	43
Fig 10.	Tratamiento de errores con excepciones en el servidor .....	43
Fig 11.	Diagrama de componentes .....	44
Fig 12.	No Conformidades detectadas .....	46
Fig 13.	Gráfica de iteraciones de pruebas funcionales a nivel de integración.....	49

**Índice de tablas**

Tabla 1.	Criterios de elección entre metodologías tradicionales y ágiles .....	11
Tabla 2.	Clases del Modelo de dominio.....	17
Tabla 3.	Actores del sistema .....	24
Tabla 4.	Especificación caso de uso Gestionar activo fijo .....	26
Tabla 5.	Descripción de entidades .....	38
Tabla 6.	Caso de prueba del caso de uso Gestionar activo fijo .....	46
Tabla 7.	Variables de prueba del caso de uso Gestionar activo fijo.....	47
Tabla 8.	Matriz de prueba caso de uso Gestionar activo fijo .....	47
Tabla 9.	Tiempo medio de respuesta arrojado por la herramienta.....	48

### **Introducción**

Para mediados de los años 90 la gestión empresarial, las habilidades directivas y los procesos de negocios se enfocaron totalmente en la gestión del Capital Humano, como recurso más importante dentro del entorno corporativo. En la actualidad la forma en que se gestione y promueva el personal influye en el éxito que pueda alcanzar una empresa. Las organizaciones del nuevo milenio y aquellas antiguas que se adaptan a las reglas actuales, se han visto forzadas a adoptar la gestión del personal como una tarea de primer orden.

Entre estas organizaciones novedosas en Cuba se encuentra la Universidad de las Ciencias Informáticas, un centro de altos estudios creado en 2002 que combina dentro de su campus centros de desarrollo, áreas docentes, servicios, soportes y terceros, involucrando gran cantidad de personal y tecnología. La estructura encargada de gestionar la tecnología es la Dirección de Gestión Tecnológica (DGT), área que agrupa un gran número de trabajadores que garantizan el soporte de hardware, actualización y reparación de los medios de cómputo. Por tanto, la Dirección tiene definido el proceso de gestión del Capital Humano para asegurar el cumplimiento de sus objetivos.

La DGT cuenta con el Sistema de Gestión de Expedientes de Control del Capital Humano (SIGEX) para la gestión de los expedientes laborales de los trabajadores del área. Esta herramienta informática permite la gestión de las estructuras organizativas de la DGT y la gestión de las plazas y la plantilla obrera. También garantiza el registro de los documentos del expediente, su aprobación por parte de directivos y la gestión de evaluaciones de desempeño.

Las tareas realizadas en la DGT abarcan más que la gestión de expedientes laborales. La propia evaluación de desempeño y muchos de los documentos, se generan a partir de otros procesos que están concertados dentro del quehacer de los trabajadores. La DGT lleva el control de todos los activos fijos y otros medios, así como los útiles que se entregan a los trabajadores, funcionalidad con la que no cuenta el sistema que es utilizado. La Dirección cuenta con un responsable de activos fijos que lleva un registro manual con los medios básicos, el que debe ir actualizando según se procesan los cambios ocurridos en el tiempo. Este proceso al ser manual provoca que se incurra en gasto innecesario de materiales para cada asignación o movimiento y aumenta la carga de trabajo de los especialistas del grupo económico y de Capital Humano. Estos no son notificados sobre las actualizaciones de la información, la que en su mayoría se encuentra redundante.

El encargado de la gestión del Capital Humano debe notificar al grupo económico de todas las actualizaciones en la plantilla obrera para determinar si existen responsabilidades sobre los medios, activos fijos y útiles durante los procesos de baja, licencias y otros estados del personal. La administración

y los responsables de grupos no pueden determinar de forma rápida los estados actuales de los medios, su ubicación e historial de cambios, proceso que es largo y tedioso.

Durante la emisión de evaluaciones en el sistema los directivos no pueden analizar la responsabilidad material de los trabajadores con los medios asignados. Esta tarea se realiza solicitando informes a los responsables de activos fijos o útiles, quienes de forma manual acceden a los archivos en busca de la información solicitada, la que no siempre llega en tiempo por el cúmulo de información archivada.

El trabajador no tiene acceso a revisar frecuentemente el listado de responsabilidad sobre los medios, activos fijos y útiles por los que responde. La administración y jefes de grupos no tienen la forma para responder rápidamente ante solicitudes superiores del estado actual de todos los medios, lo que influye en la mala planificación, control y análisis real del área.

A partir de la situación problemática planteada se tiene el siguiente **problema de investigación**: ¿Cómo contribuir a la gestión de activos fijos, medios y útiles de la Dirección de Gestión Tecnológica en la Universidad de las Ciencias Informáticas (UCI)?

Teniéndose como **objeto de estudio**: Proceso de gestión de activos fijos, medios y útiles; centrándose en el **campo de acción**: Sistema de Gestión de Expedientes de Control del Capital Humano de la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas.

Para solucionar el problema de investigación se define como **objetivo general**: Desarrollar un módulo de gestión de activos fijos, medios y útiles para el Sistema de Gestión de Expedientes de Control del Capital Humano de la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas.

Para guiar la investigación se definen las siguientes **preguntas científicas**:

¿Cuáles son los fundamentos teóricos de la gestión de los activos fijos, medios y útiles?

¿Cuáles son las características que debe cumplir el módulo de gestión de activos fijos, medios y útiles?

¿Cómo estructurar el proceso de desarrollo del módulo de gestión de activos fijos, medios y útiles?

¿El módulo desarrollado contribuye a la gestión de gestión de los activos fijos, medios y útiles dentro de la DGT de la Universidad de las Ciencias Informáticas?

Se establecen las siguientes **tareas de investigación**:

- ✓ Identificación de las características fundamentales del proceso de gestión de activos fijos, medios y útiles en la Dirección de Gestión Tecnológica.
- ✓ Selección de las tareas a automatizar a partir del flujo de negocios identificado, detallando los requisitos a tener en cuenta para cada una de ellas.

- ✓ Realización del análisis y diseño del módulo propuesto y los mecanismos de integración a la herramienta SIGEX a partir del estudio del sistema y el modelo propuesto para el módulo a desarrollar.
- ✓ Implementación de las funcionalidades propuestas siguiendo los principios, acuerdos y métodos definidos en la fase de análisis y diseño.
- ✓ Comprobación del correcto funcionamiento del módulo desarrollado para la gestión de activos fijos, medios y útiles de SIGEX de la Dirección de Gestión Tecnológica.

Para la realización de la investigación se hacen uso de los siguientes métodos científicos:

**Analítico – Sintético:** se utilizó para realizar un análisis y síntesis de los elementos encontrados en las teorías de los diferentes autores sobre el proceso de gestión de responsabilidades materiales, con el objetivo de establecer conclusiones que permitan definir mejor la investigación.

**Monitoreo de proyectos:** utilizado para analizar la inserción de aplicaciones existentes para la gestión de activos fijos, medios y útiles en empresas. Entre estas aplicaciones se incluye ASSETS NS utilizado en la Universidad de las Ciencias Informáticas.

### **Estructura del Documento.**

El documento está organizado en tres capítulos. Con el título de “Fundamentos teóricos de la gestión de activos fijos, medios y útiles”, el primer capítulo contiene el estudio del marco teórico, mostrando las principales características del proceso de gestión de activos fijos, medios y útiles, así como las responsabilidades y los conceptos relacionados. Se analizan algunos sistemas que incluyen la gestión de activos fijos. Por otra parte, se describen y caracterizan las herramientas y lenguaje de programación para el diseño e implementación de la solución, así como la metodología de desarrollo más recomendable. En el capítulo 2 “Análisis y diseño del Módulo de gestión de activos fijos, medios y útiles”, se exponen las principales características y cualidades de la solución a implementar. Además, se identifican los requerimientos, se escoge la arquitectura y los patrones de diseño, se diseñan las clases y se detallan los pasos de la metodología propuesta, incluyendo varios de sus artefactos y diagramas. Concluyendo, el capítulo 3 “Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles”, describe la implementación del módulo a partir de los artefactos y evidencia los resultados de la investigación a partir de las pruebas aplicadas para demostrar el correcto funcionamiento de la solución desarrollada.

## **Capítulo 1 Fundamentos teóricos de la gestión de activos fijos, medios y útiles**

En este capítulo se analiza el marco teórico de la investigación donde se abordarán los conceptos relacionados con la gestión de activos fijos, medios y útiles. También se abordan aspectos importantes de soluciones existentes que se pudieran tener en cuenta para un mejor desarrollo de la solución. Por último, se caracterizan las tecnologías, lenguajes y herramientas definidas en la investigación.

### **1.1 Marco teórico**

Para comprender el proceso enmarcado en el objeto de estudio es necesario analizar los conceptos que serán abordados indistintamente a lo largo de toda la investigación y que se exponen a continuación.

#### **1.1.1 Medios empresariales**

Los medios o recursos empresariales son los bienes tangibles que tiene una empresa a su disposición para alcanzar sus objetivos: dentro de estos recursos se encuentran los siguientes (Andrew, 2014):

- Maquinarias: incluye equipos especializados, máquinas de ensamblaje, tecnologías de montaje entre otras con que cuenta la empresa que están directamente al servicio del proceso de producción.
- Inmuebles: son todos los bienes raíces de la empresa. Entre ellos se encuentran: terrenos, edificios, mesas, sillas, edificios, y otros.
- Insumos: entradas con que cuenta la empresa para convertirlo mediante el proceso de mejora de valor en productos o servicios terminados.
- Productos terminados: producto generado mediante el proceso de producción destinado a la satisfacción de la función económica de la empresa.
- Elementos de Oficina: contiene materiales de oficina y medios de cómputo.
- Instrumentos y herramientas: conjunto de instrumentos, herramientas y útiles destinados a los trabajadores o áreas para facilitar la realización de una tarea mecánica.

Dentro de la administración empresarial, la gestión de los medios materiales constituye un elemento clave para la organización. La dirección siempre debe tener en cuenta todos los elementos para permitir el máximo aprovechamiento de sus recursos materiales, fundamentalmente de las maquinarias, inmuebles, elementos de oficina e instrumentos y materiales, lo que constituyen sus activos fijos.

# *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

## **1.1.2 Activos empresariales**

El activo se define como un conjunto de bienes o derechos que tienen un beneficio económico a futuro. Los activos de las empresas varían de acuerdo con la naturaleza de la actividad desarrollada y a cobrar (Ortega, 2012). La clasificación de los tipos de activo está basada fundamentalmente en su nivel de consistencia y el servicio que pueda aportar a su propietario: circulante, fijo, cargos o gastos diferidos u otros.

Los activos circulantes son aquellos recursos que, aparte del efectivo, sean disponibles en menos de un año, esto es, que sean factibles de convertirse en efectivo en el lapso indicado, por tratarse de recursos que normalmente se cobran, se venden o se consumen en tal periodo (Torres, 1978). Dentro de esta clasificación se encuentran: caja, banco, inversiones temporales, efectos por cobrar, cuentas por cobrar, inventario de mercancías o seguro pagado por anticipado.

Los cargos o gastos diferidos son gastos por monto significativo que generalmente no ocurren de manera repetida y cuya característica principal es que son gastos no recuperables en dinero, pero como el beneficio que estos gastos van a prestar generalmente abarca varios años, esa es la razón por la cual se consideran activos y con el transcurrir del tiempo se van transfiriendo a egresos (Olivo de Latouche, y otros, 1989). Algunos son: gastos de organización, campaña publicitaria y mejoras a propiedad arrendada.

Existen algunos activos que no responden a los lineamientos y características de grupos anteriores ni pueden considerarse como activos fijos, los cuales inclusive son disímiles entre sí de acuerdo a su naturaleza, se agrupan dentro de esta denominación de "Otros Activos" la cual no tiene ninguna característica en común que los defina (Olivo de Latouche, y otros, 1989). Algunos son: depósitos dados en garantía, efectos por cobrar y seguro de vida.

## **1.1.3 Activos fijos**

El activo fijo es el conjunto de bienes de naturaleza relativamente permanente, adquiridos, desarrollados, construidos por administración propia o por contrato, necesarios para el cumplimiento de objetivos institucionales (Ananraya, y otros, 2010).

Un bien para ser considerado como activo fijo debe caracterizarse por ser de naturaleza relativamente duradera y estar destinado al uso y no a la venta (Ananraya, y otros, 2010). Al ser adquiridos en una empresa son destinados a un área, por consecuencia el responsable del bien es el jefe del departamento. Puede ser cualquier instrumento o herramienta: una mesa, una silla, un conector, entre muchos más. Los activos fijos dentro de la empresa representan una de las armas básicas en la creación de aquellos productos o servicios que responden a la actividad fundamental insertada en la organización.

## *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

Es válido destacar que dentro del marco lucrativo que existe en un sistema empresarial, la participación de los distintos tipos de activos (y sus derivados) es relativa, pues estos pueden sufrir cambios según la particularidad de cada empresa. Todo su control de cambios, así como su asignación, responsabilidad y valor, debe ser tenido en cuenta durante la gestión administrativa.

Los activos fijos se clasifican según su naturaleza de duración, materialidad o presencia física. Las clasificaciones son: activos fijos intangibles y activos fijos tangibles. Los activos fijos intangibles están representados por derecho de orden legal o económico, los cuales tienen como característica su inmaterialidad, es decir, carecen de presencia física, lo que implica que su existencia no se puede apreciar con los sentidos. Pueden ser patentes, marcas de fábricas, derechos de autor, plusvalías, entre otros. Los activos fijos tangibles lo constituyen bienes con todas las características anotadas anteriormente para los activos fijos, los cuales están connotados por tener materialidad, presencia física la que se puede apreciar con los sentidos (Planche, y otros, 2004). Ejemplos son: terrenos, maquinarias, vehículos, mobiliarios, edificios, bosques, minas, entre otros. Tanto los activos fijos tangibles como los intangibles son considerados unidades económicas dentro de la empresa.

### **1.1.4 Gestión administrativa de medios**

Generalmente, las empresas tienen un área administrativa que es la encargada de la gestión de sus medios materiales. La administración debe realizar las actividades necesarias para garantizar el control de estado, ubicación, disponibilidad, responsabilidad y mantenimiento de los medios. El control de los activos fijos comienza a partir de un inventario físico, donde se recogen los datos que identifican físicamente cada activo en específico. Luego estos son asignados a un área donde van a prestar servicio, además de la custodia y preservación de los bienes por parte del personal y al centro de costo correspondiente. A partir de estos datos se comienza con la elaboración de un registro económico (gestión de centros de costos) que se comparte con el registro que lleva Capital Humano de las responsabilidades de cada personal incluyendo los activos bajo su custodia (Liu, y otros, 2013).

Por tanto, aunque el área de economía tiene el control de los activos, en su gestión participan las áreas de Capital Humano, logística y la propia dirección general. Los registros son actualizados constantemente; en economía, con los valores cambiantes según la depreciación de los medios y los movimientos y en Capital Humano, según los cambios de responsabilidad material. La realización efectiva de estos procesos depende del grado de gestión de información que se tenga en la empresa, generalmente apoyada por sistemas automatizados para la gestión empresarial.

# *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

## **1.2 Sistemas automatizados para el control de medios**

Los sistemas automatizados para la gestión de activos empresariales datan desde los años 70, cuando la mayoría de las empresas comienzan a adoptar las tecnologías de la información para administrar sus recursos. Desde los sistemas de gestión del conocimiento, sistemas de oficinas, hasta los sistemas inalámbricos de apoyo a la capa estratégica de la administración, propician ambientes para gestionar activos fijos y útiles. La forma de gestionarlos depende en mayor grado, del propósito del software, el tipo de empresa, política empresarial y usuario final. Actualmente, aunque existe la disciplina de Sistemas de Administración de Bienes Empresariales (AMS por las siglas del inglés *Assets Management System*) las soluciones más completas están dentro de los ERP, (siglas del inglés *Enterprise Resources Planning* o Planificación de Recursos Empresariales en español) soluciones integrales destinadas a la gestión de las áreas fundamentales de la empresa (Vladimirui, 2013).

Estas herramientas permiten automatizar la gestión de activos, medios y útiles enlazando en una sola plataforma los procesos relacionados con producción, logística, administración, capital humano, contabilidad y dirección. También existen sistemas que, aunque no son ERP, están destinados a la gestión de activos, tanto del punto de vista contable como desde la administración, aunque ninguno dirigido a la gestión de responsabilidades desde el área de Capital Humano. Algunos de los más utilizados en Cuba son SAP One, Versat y ASSETS NS. A continuación, se caracterizan algunos de estos sistemas utilizados en Cuba y otras partes del mundo.

### **1.2.1 SAP R/3**

SAP R/3 es un ERP comercial líder en el sector empresarial, perteneciente a la compañía alemana SAP. FI-AM es el módulo de Activos Fijos, es un sub-libro del libro de Contabilidad Financiera y es usado para administrar y documentar a detalle las actividades de los activos fijos. El módulo FI-AM permite actualizar la depreciación y los cambios de valor de un activo en el Libro de Contabilidad General del sistema SAP R/3. Pueden realizarse diferentes asignaciones de contabilidad de costos para estas actividades (Arregoces, y otros, 2011). La gestión de los activos fijos tiene por objetivo su valoración y recogida en el inventario de la organización. Desde este módulo se realizan también operaciones de alta, baja y traslado de los mismos, gestionando y controlando aspectos concretos del patrimonio del inmovilizado.

SAP R/3 está desarrollado con el lenguaje de programación Perl. Solo es compatible con bases de datos como Oracle, Microsoft SQL Server y otras de carácter privativo.

### **1.2.2 OpenBravo ERP**

Es el producto ERP de la compañía OpenBravo. Presenta altos costes de licencia. Su módulo de control de activos permite definir medios tangibles de la empresa, definir responsabilidades y gestionar sus

## *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

cambios de valor y responsabilidad. Su sistema de reportes permite tener el control del estado de los medios de forma rápida y segura. Solamente es compatible con los gestores de base de datos Oracle y PostgreSQL.

### **1.2.3 SAP One**

SAP One es un sistema especializado de la empresa SAP, que se integra a la gestión en la nube que realiza la empresa a través de R/3, aunque también puede operar independiente. El módulo de Activos Fijos le permite gestionar y supervisar estos medios en una empresa. Se integra perfectamente dentro del componente estándar *Finanzas* de SAP Business One. Activos fijos calcula amortizaciones previstas en función de los costes de adquisición y producción, así como la vida útil, y realiza las contabilizaciones relevantes en el libro mayor (AX Consulting, 2003). Algunas de sus características son la facilidad de parametrización, la importación de datos, realización de distintos tipos de transacciones, previsión de la amortización y cuadro de activos fijos. Como sistema, su instalación es costosa y compleja, prácticamente obliga a adquirir todo el paquete de software, los que presentan costos elevados de licencia, capacitación y soporte.

### **1.2.4 Versat**

El Versat Sarasola lo definen sus desarrolladores como un paquete integrado para la gestión económica financiera que permite enviar información eficaz, de forma inmediata, desde lugares apartados, a la vez que ofrece mayor organización, control y disciplina en cada gestión. Está realizado por la empresa Datazucar, entidad cubana que inicialmente desarrolló la herramienta específicamente para gestionar operaciones relacionadas con la industria azucarera. Es un sistema económico actualmente integrado por 10 módulos o subsistemas que incluyen configuración y seguridad, contabilidad general y de gastos, costos y procesos, finanzas y caja. Además, en el proceso intervienen activos fijos, planificación y presupuestos, control de inventarios, pago de salario (nómina), facturación y generador de reportes (Sosa, y otros, 2010). Su principal problema con respecto al campo de acción es que está dedicado totalmente a un ambiente contable y no orienta sus funciones a la gestión de actividades o procesos propiamente de Capital Humano.

### **1.2.5 ASSETS NS**

ASSETS NS es un sistema de gestión integral estándar y parametrizado que permite el control de los procesos de compras, ventas, producción, taller, inventario, finanzas, contabilidad, presupuesto, activos fijos, útiles y capital humano. Como sistema integral todos sus módulos trabajan en estrecha relación, generando, automáticamente, al módulo de contabilidad los comprobantes de operaciones por cada una

## *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

de las transacciones efectuadas, esto permite que se pueda trabajar bajo el principio de contabilidad al día (D´MARCO S.A, 2015).

Es un sistema que funciona en ambiente multiusuario incluidas estaciones remotas. Asimismo, proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario. En ASSETS NS se facilita el uso de la parametrización para adaptarse a las exigencias de cada entidad en particular, garantizando que sus reportes tengan la forma y el contenido que el usuario les defina. Actualmente, el sistema ASSETS es el sistema utilizado por la Universidad de las Ciencias Informáticas para la gestión de sus activos fijos. Como mismo refleja la herramienta, está orientada a la gestión contable de estos activos, lo que no permite gestionarlos desde un enfoque de Capital Humano, a partir de su estado, asignaciones y responsabilidades.

Este sistema, que tiene una estructura cerrada, no permite la adición de funcionalidades ni la extensión de las actuales, además de que su enfoque multiterminal no es web, teniendo dependencia del sistema operativo del cliente.

### **1.2.6 Conclusión de los sistemas analizados**

A pesar de las funcionalidades orientadas a la gestión de medios empresariales, los sistemas analizados presentan limitaciones que no propician su elección como solución al problema planteado. SAP R/3 y SAP One son las herramientas más completas, pero se descartan debido a su orientación a Software como Servicio (SaaS), su interdependencia modular y su elevado costo de licencia (17000 USD para 10 usuarios utilizando las funcionalidades de la versión estándar (Taktik Consulting, 2010)). OpenBravo requiere el rediseño de sus módulos de gestión para adaptarlos a las necesidades de la DGT, lo que implica tiempo y costos de desarrollo fuera del alcance de la DGT. Versat y ASSET NS están orientadas totalmente a la gestión contable y requieren que la información manejada desde Capital Humano se relacione con procesos fuera del campo de acción, complejizando el trabajo de los especialistas de Capital Humano e involucrando actividades innecesarias. Por tanto se hace necesario desarrollar una nueva solución que permita gestionar los activos fijos, medios y útiles de la DGT, posibilitando además gestionar las asignaciones y responsabilidades materiales desde un ámbito de Capital Humano, facilitando el acceso a la información y su historial de cambios.

### **1.3 Metodología, tecnologías y herramientas de desarrollo**

Para cubrir las necesidades de la Dirección de Gestión Tecnológica, en el año 2015 se creó el Sistema de Gestión de Expedientes de Control del Capital Humano (SIGEX) que permite a los especialistas de Capital Humano, automatizar la generación, almacenamiento, búsqueda y recuperación de evaluaciones de los trabajadores de la DGT. Este sistema, realizado con una arquitectura extensible, de forma modular y con

## *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

un ambiente de seguridad adecuado, realiza funciones bases necesarias para el desarrollo de la solución propuesta. Por tanto, se procede a diseñar e implementar un módulo para este sistema que permita la gestión de activos fijos, medios y útiles. Debido a esto, las tecnologías y herramientas a exponer, son aquellas que utiliza el sistema SIGEX y que constituyen las que serán utilizadas en el desarrollo del módulo.

### **1.3.1 Metodología de desarrollo de software**

Una metodología de desarrollo de software es un concepto de la ingeniería que define al conjunto de técnicas y métodos que se utilizan, de forma estandarizada, para gestionar las actividades de ciclo de vida de un proyecto de software. En general, una metodología facilita las tareas de planificación, control y seguimiento de un proyecto, mejora la relación coste/beneficio, optimiza los recursos disponibles, provee la evaluación de resultados y cumplimiento de los objetivos y la comunicación efectiva entre usuarios y desarrolladores (Brooks, 2011).

Actualmente no se puede decir que existe una metodología única para garantizar el éxito de cualquier proyecto de desarrollo de software, debido a que estas deben ajustarse al tipo de proyecto que se desarrolle. Existen metodologías ágiles y metodologías tradicionales. Las ágiles se caracterizan por enfocarse solo en la documentación necesaria y proveer un desarrollo acelerado, mientras que las tradicionales aseguran toda la documentación y métodos más formales y detallados. Entre las metodologías tradicionales se encuentran RUP (*Rational Unified Process*), MSF (*Microsoft Solution Framework*), *Win-Win Spiral Model* e *Iconix*. Entre las ágiles se pueden citar varias propuestas: XP (*eXtreme Programming*), SCRUM, OpenUp, DSDM (*Dynamic Systems Development Method*) entre otras (Wallace, y otros, 2002).

Desde el punto de vista de Bohem y Turner (Jacobson, y otros, 2000), a partir de su matriz sobre el uso de metodologías ágiles o clásicas para el desarrollo de software, se puede fundamentar la metodología adecuada para la solución propuesta.

Teniendo en cuenta las características del proyecto descritas en la Tabla 1, se puede concluir que es necesario utilizar una metodología ágil. Esto garantiza que se mantenga un control cualitativo durante el desarrollo del software, agilizándolo y midiendo el resultado mediante casos de prueba.

## Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles

Tabla 1. Criterios de elección entre metodologías tradicionales y ágiles

Criterio	Ágil	Clásica	Solución
<b>Objetivos primarios</b>	Rápidos, respuesta al cambio.	Estabilidad, alta garantía	La aplicación debe ser adaptable a posibles cambios que aparezcan según los procesos de la DGT.
<b>Tamaño</b>	Pequeños proyectos y equipos de desarrollo	Grandes equipos y proyectos	Proyecto de pequeño tamaño.
<b>Entorno</b>	Cambiante	Estable	El entorno es estable
<b>Planeación y control</b>	Planes y control cualitativo	Planes documentados, control cuantitativo	Control cualitativo a partir de resultado de las fases.
<b>Comunicación</b>	Conocimiento interpersonal	Conocimiento explícito documentado	Conocimiento implícito y personal.
<b>Requerimientos</b>	Pequeños y medibles mediante casos de pruebas	Complejos, Formalizados, medibles y evolutivos	Pequeños basados en pruebas y experiencias de usuario.
<b>Desarrollo</b>	Diseño simple, incrementos cortos	Diseño extensivo, incrementos largos	Desarrollo simple
<b>Prueba</b>	Casos de prueba	Pruebas documentadas	Casos de prueba para los requisitos.

Para el desarrollo rápido de aplicaciones bajo un enfoque ágil, existe la metodología OpenUP, una versión ligera de RUP que simplifica los artefactos generados (Brooks, 2011). Mantiene los mismos principios de RUP: iterativo e incremental, centrado en la arquitectura y dirigido por casos de uso, pero incluye cambios en las fases para permitir agilizar el desarrollo de las aplicaciones. Por tanto, es apropiado para proyectos pequeños o de bajos recursos, aplicable a un conjunto amplio de plataformas de desarrollo y su punto de vista pragmático le permite centrarse en una naturaleza colaborativa dentro del equipo de desarrollo. Una de sus principales características es su alto grado de adaptabilidad a las necesidades de un proyecto en particular (Addison Wesley, 2004). Es el proceso unificado que aplica acercamientos iterativos e incrementales dentro de un ciclo vital estructurado.

El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y dirección para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos. OpenUp estructura el ciclo de vida de un proyecto en cuatro fases: inicio, elaboración, construcción y transición.

## *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

La elección de una metodología ágil se realiza teniendo en cuenta el tiempo de desarrollo del proyecto, la cantidad de participantes y su alcance. Aunque SIGEX se desarrolló con la metodología *Extreme Programming* (XP), al utilizarse OpenUp como metodología, se garantiza que el módulo pueda ser comprendido y mejorado a partir de los artefactos generados.

### **1.3.2 Lenguaje de modelado unificado (UML 2.0)**

En las ciencias informáticas un lenguaje de modelado es un recurso utilizado para especificar, visualizar, construir y detallar un artefacto generado durante el desarrollo de un software. El lenguaje incluye las notaciones y representaciones necesarias para que el producto sea comprendido en su totalidad, desde sus componentes persistentes hasta sus relaciones con otros sistemas (Duggan, y otros, 2012). El lenguaje más estandarizado actualmente es Lenguaje de modelado unificado (UML por sus siglas en inglés) (Fowler, 2007).

UML se caracteriza por incluir partes estáticas, dinámicas, de entorno y organizativas. Es independiente de todos los métodos para el análisis y diseño de aplicaciones. Está diseñado para usarse desde herramientas interactivas de modelado que permiten a través de UML generar el código representado. UML no define un proceso estándar, pero sirve para representar cualquier artefacto de un sistema de software. Consolida varias notaciones y conceptos principalmente de la programación orientada a objetos (OMG, 2001). Permite la modelación del ciclo completo de desarrollo de software y contiene: los diagramas de casos de uso, clases, objetos, secuencia, colaboración, estado, actividades, despliegue y componentes. Crea un lenguaje de modelado utilizado tanto por humanos como por máquinas. Muestra mejor soporte a la planeación y al control de proyectos y una alta reutilización y minimización de costos.

### **1.3.3 Lenguaje de programación**

Se denomina lenguaje de programación a la representación diseñada para expresar un proceso o algoritmo que pueden ser interpretados por máquinas y ejecutarse mediante bloques de instrucción. Están compuestos por símbolos y reglas que definen la estructura de cada lenguaje (Meyer, 2009).

Para el desarrollo de aplicaciones web los sistemas más relevantes actualmente son JavaScript, HTML, CSS y PHP. Estos son los utilizados en el sistema SIGEX.

#### **PHP 5.4.12**

PHP (siglas del inglés *Hypertext Preprocessor*) es un lenguaje etiquetado diseñado para incrementar el dinamismo con las páginas web, aprovechando los beneficios de la red. Es un lenguaje muy difundido, flexible, de alto nivel y embebido dentro de las páginas HTML. Se ejecuta del lado del servidor y permite insertar contenidos dinámicamente. El principal objetivo de PHP 5 ha sido mejorar los mecanismos de

## *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

programación orientada a objetos. Un paso necesario para lograr que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos (Merley, 2012).

### **JavaScript**

JavaScript es uno de los recursos que han surgido para incorporar dinamismo y capacidades al lenguaje HTML. Actualmente es la tecnología más extendida en el enriquecimiento de páginas web del lado del cliente. JavaScript no es un lenguaje orientado a objetos es más bien un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto y plantillas de cálculo. Entre las acciones típicas que se pueden realizar en JavaScript se tienen dos vertientes. Por un lado los efectos especiales sobre páginas web y por el otro, permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas. Es un lenguaje interpretado con la filosofía orientada a objetos (W3C, 2015).

### **CSS 3**

CSS (siglas en inglés de *Cascading Style Sheets*) es un lenguaje diseñado para controlar el estilo de los documentos definidos con HTML o XHTML (W3C, 2015). Permite que se separe el contenido de la presentación creando interfaces enriquecidas visualmente.

### **HTML 5**

HTML (siglas en inglés de *Hyper Text Markup Language*) es el lenguaje de marcado más utilizado para la construcción de páginas web. Es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la síntesis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. Su versión 5 brinda más facilidades que sus antecesores, lo que permite el desarrollo de la aplicación con una mayor vistosidad, calidad e integración (W3C, 2015).

#### **1.3.4 Marco de trabajo Symfony 1.4.11**

Es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que se le añaden las últimas piezas para construir una aplicación concreta (De los Ángeles, 2012).

SIGEX está desarrollado con Symfony 1.4.11. Es un completo marco de trabajo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (Potencier, 2011).

## *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

Symfony se diseñó para ser independiente del sistema gestor de bases de datos. Es sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos. Está basado en la premisa de "convenir en vez de configurar", en la que el desarrollador sólo debe configurar aquello que no es convencional. Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares) (Potencier, 2011).

Para el uso de JavaScript y trabajo con las vistas se utiliza el marco de trabajo Ext JS 3.2.1. ExtJS es una librería de JavaScript ligera, de alto rendimiento para el desarrollo rápido de aplicaciones web cross-browser. Presenta una interfaz de usuario personalizable "*widgets*", bien diseñada y contiene un modelo de componentes extensibles. Tiene disponibles licencias comerciales y de código abierto. Es compatible con la mayoría de los navegadores (Ext JS Inc, 2015).

### **1.3.5 Sistema Gestor de Base de Datos PostgreSQL 9.1.2**

Los sistemas de gestión de bases de datos (*en inglés Database Management System, abreviado DBMS*) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización (Arregoces, y otros, 2011). SIGEX utiliza como motor de base de datos PostgreSQL 9.1.2.

PostgreSQL es un servidor de base de datos relacional libre. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. Presenta como características principales claves ajenas también denominadas Llaves Ajenas o Llaves Foráneas, disparadores, vistas, implementación del estándar SQL92/SQL99, integridad transaccional, acceso concurrente multiversión, capacidad de albergar programas en el servidor en varios lenguajes, herencia de tablas y tipos de datos (PostgreSQL Development Group, 2009).

### **1.3.6 IDE NetBeans 8.0**

Es un entorno de desarrollo integrado libre, extensible para el desarrollo sobre muchos lenguajes de programación, aunque se realizó fundamentalmente para el lenguaje Java. Como ventajas cuenta con administración de interfaces de usuario y de almacenamiento, integración a múltiples marcos de trabajo. Muestra gran cantidad de módulos y extensiones para múltiples lenguajes y tecnologías. Fuerte comunidad de respaldo (Domínguez, 2005).

# *Capítulo 1. Fundamentos Teóricos de la gestión de activos fijos, medios y útiles*

## **1.3.7 Herramienta CASE Visual Paradigm 5.1**

Es un producto de alta calidad donde el modelado de sistemas y soluciones de software se realiza utilizando UML. Permite crear diagramas de clases y artefactos que tributan a la documentación de un software a lo largo del proceso de desarrollo. Se integra fácilmente con varios entornos de desarrollo y permite la generación semiautomática de código a partir de los diagramas construidos y también se puede llevar a cabo la ingeniería inversa para refinar los modelos. Soporta aplicaciones web y existe compatibilidad entre sus ediciones (Visual Paradigm Group, 2011).

## **1.3.8 Servidor Web Apache 2.0**

Apache es una plataforma de servidores web de código abierto. Su alta configuración, robustez y estabilidad hacen que más del 64% de toda web reiteren su confianza en este programa. Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Apache es una de las plataformas de servidores web más destacadas dentro de la gran cantidad existentes, de código fuente abierto para plataformas Unix, Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache es personalizable, su arquitectura modular permite construir un servidor hecho a la medida y posibilita la implementación de los últimos y nuevos protocolos (ASF, 2015).

## **1.4 Conclusiones parciales.**

En este capítulo se abordaron los conceptos de la investigación que serán el soporte teórico de la propuesta de solución, además del motivo y la necesidad de la realización del presente trabajo. Los sistemas analizados, aunque presentan funcionalidades esenciales para la gestión de activos fijos, medios y útiles, no se adaptan a las necesidades de la DGT, por lo que es necesario la realización de una aplicación informática que permita definir, gestionar y evaluar las asignaciones y responsabilidades sobre los medios desde un ambiente de Capital Humano. Se decidió utilizar como metodología de desarrollo OpenUp y mantener como marco de trabajo Symfony 1.4.11, el IDE NetBeans 8.0 y PHP 5.4.12 como lenguaje de programación del lado del servidor y JavaScript como lenguaje de programación del lado del cliente, además de Visual Paradigm 5.1 como herramienta CASE. Se empleará la librería Ext JS 3.2.1, como Sistema Gestor de Bases de Datos PostgreSQL 9.1.2 y como servidor web Apache 2.0. El empleo de las herramientas y tecnologías seleccionadas aporta eficacia y agilidad durante el proceso de desarrollo, de igual manera impulsan la soberanía tecnológica propuesta por el país y la UCI. Además aseguran una total compatibilidad con el sistema SIGEX ya que fueron las empleadas en el desarrollo del mismo.

## **Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles**

En el presente capítulo se exponen las principales características y cualidades de la propuesta de solución, mediante la identificación de los requisitos funcionales y no funcionales con los que debe cumplir. Se describen los conceptos relacionados con las definiciones de estilo y patrones de diseño de software empleados. Se incluye la elaboración del modelo de dominio, la definición de actores y relaciones entre ellos, además de los diagramas de casos de uso del sistema y las descripciones textuales de los casos de uso. Definiendo finalmente los diagramas de clases del diseño propuestos, la arquitectura y modelo de datos.

### **2.1 Modelación de dominio**

El modelo de dominio o modelo conceptual captura los tipos más importantes de objetos en el contexto de un negocio a automatizar. Estos objetos representan los responsables y activos relevantes en el negocio que se traducirán en roles o entidades en un sistema. Esta modelación debe permitir establecer claramente la relación entre los objetos, extraer reglas específicas, así como la responsabilidad sobre cada una de las entidades (McConell, 2012).

Este modelo se realiza cuando no se logra determinar el proceso del negocio con fronteras bien establecidas y donde los flujos de información son difusos (múltiples orígenes, sólo eventos, sucesos), cuando existe solapamiento o múltiples responsabilidades (Larman, 2003). El modelo de dominio se describe mediante diagramas UML (diagrama de clases). Estos diagramas muestran las clases del dominio y cómo se relacionan unas con otras mediante asociaciones.

Para la presente investigación se plantea un modelo de dominio porque no se tiene una estructura definida de los procesos del negocio y existe solapamiento y múltiples responsabilidades por cada uno de los actores del negocio.

El modelo de dominio contribuye posteriormente en el proceso de desarrollo del software. Mediante este se identifican las clases que se utilizarán para modelar el sistema, las que se describen a continuación en la Tabla 2.

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

Tabla 2. Clases del Modelo de dominio

Clases	Descripción
Útil	Referencia al tipo de medio considerado como útil que es necesario para que un trabajador pueda realizar sus tareas laborales.
Activo fijo	Hace referencia a los activos fijos tangibles de la DGT.
Medio	El concepto hace referencia a otros medios que son manipulados por los trabajadores de la dirección como sellos y órdenes de servicio. Las clases Útil y Activo fijo se excluyen de este término, dado su importancia en la investigación se tratarán como términos independientes a la clase Medio.
Especialista Capital Humano	Hace referencia al especialista encargado de las solicitudes de activos fijos, medios y útiles necesarios para cada trabajador de la DGT.
Grupo económico	Refiere al área económica de la dirección encargada del control de útiles, medios y activos fijos.
Registro	El concepto hace referencia al registro de responsabilidad material sobre las asignaciones realizadas a los trabajadores.
Responsable de medio	Refiere a la persona encargada de la asignación y entrega de medios dentro de la DGT y que integra el grupo económico de la DGT.
Jefe inmediato superior	Hace referencia al jefe establecido por cada área de mando de la dirección.

El negocio está representado por las actividades de supervisión, control y actualización de asignaciones y responsabilidades materiales, las que se recogen dentro de las tareas del especialista de Capital Humano de la DGT. Además, involucra a los trabajadores del área y al grupo económico que tiene el control de todos los medios.

El especialista de Capital Humano es el encargado de determinar las necesidades de activos fijos, medios y útiles para cada puesto de trabajo y solicita las asignaciones para los trabajadores. El responsable de medio pertenece al grupo económico y tiene la responsabilidad de llevar a cabo la asignación de activos fijos y útiles, actividad que en el caso de otros medios es ejecutada por el jefe inmediato superior. Como constancia de estas asignaciones se tiene el registro de responsabilidad material de cada empleado. Dicho registro se lleva en un archivo único que es controlado por el especialista de Capital Humano y que a su vez supervisa el jefe inmediato superior. Cada trabajador debe firmar las asignaciones de medios que están bajo su responsabilidad, dejando constancia de la tenencia física de los medios. Tanto el



## *Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles*

un cambio de estado de su plantilla sin antes tener actualizado el registro de responsabilidad material de activos fijos, medios y útiles según lo establecido para el estado que opta.

Durante los cambios de estados (bajas, traslados, licencias) de cada trabajador, el especialista de Capital Humano debe verificar y actualizar las asignaciones de estos medios sobre este. Para esto es posible generar una solicitud de entrega de medios, la que deriva en un cambio de estado del trabajador. Esta actividad permite que exista una constancia de la entrega total de todos los activos fijos, medios y útiles asignados a un trabajador. Para esto los responsables de cierre de cada medio son establecidos por el especialista de Capital Humano. Una entrega de medios será efectiva cuando todos los responsables den su aceptación y finalmente se emita la aprobación por el especialista de Capital Humano. Para optimizar estas funciones se emitirán notificaciones a los actores involucrados en cada acción, garantizando que en el momento que interactúen con la aplicación puedan conocer sobre las actividades realizadas o pendientes de aprobación.

Las responsabilidades materiales, reportes y listados generales de activos fijos, medios y útiles, podrán exportarse a formato PDF e imprimirse para uso de la DGT. Se mostrarán gráficos que posibiliten el análisis de las distribuciones de estos medios por áreas o grupos de la dirección.

Para dar cumplimiento a estas funcionalidades, el módulo utilizará el registro de trabajadores de SIGEX, la taxonomía de estructuras laborales y la definición de plazas. Esto permitirá aumentar el grado de automatización del flujo de información operado dentro de la DGT. Sus funcionalidades se expresarán mediante requisitos de software.

### **2.2.1 Requisitos funcionales**

Los requisitos se dividen en funcionales y no funcionales. Los funcionales son las capacidades y acciones con que el sistema debe cumplir, mientras los no funcionales constituyen cualidades o condiciones que el software debe tener (Krcuten, 2004).

Los requisitos funcionales (RF) son declaraciones de los servicios que deben proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005).

Para el correcto funcionamiento del módulo propuesto fueron identificados 54 requisitos funcionales.

- RF 1. Adicionar nuevo activo fijo
- RF 2. Modificar activo fijo existente
- RF 3. Eliminar activo fijo existente

## *Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles*

- RF 4. Listar activos fijos
- RF 5. Realizar búsqueda avanzada de activo fijo
- RF 6. Exportar listado de activos fijos en formato PDF
- RF 7. Imprimir listado de activos fijos
- RF 8. Adicionar unidad de medida de útil
- RF 9. Modificar unidad de medida de útil
- RF 10. Eliminar unidad de medida de útil
- RF 11. Listar unidades de medidas existentes
- RF 12. Adicionar motivo de entrega de medio
- RF 13. Eliminar motivo de entrega de medio
- RF 14. Modificar motivo de entrega de medio
- RF 15. Listar motivos de entrega de medio
- RF 16. Adicionar nuevo útil
- RF 17. Modificar útil existente
- RF 18. Eliminar útil existe
- RF 19. Listar útiles en el sistema
- RF 20. Exportar listado de útiles en formato PDF
- RF 21. Realizar búsqueda avanzada de útil
- RF 22. Imprimir listado de útiles
- RF 23. Definir responsable de activos fijos
- RF 24. Eliminar responsable de activos fijos
- RF 25. Solicitar asignación de activos fijos
- RF 26. Definir responsable de útiles
- RF 27. Eliminar responsable de útiles
- RF 28. Solicitar asignación de útil
- RF 29. Solicitar asignación de otros medios
- RF 30. Listar historial de movimientos de útiles
- RF 31. Solicitar movimiento de útiles
- RF 32. Aprobar movimiento de útiles a partir de una solicitud existente
- RF 33. Listar responsables de entrega de medio
- RF 34. Adicionar responsable de entrega de medio
- RF 35. Eliminar responsable de entrega de medio
- RF 36. Modificar responsable de entrega de medio
- RF 37. Listar solicitudes de entrega de medio

## *Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles*

- RF 38. Realizar búsqueda avanzada de solicitud de entrega aprobada
- RF 39. Solicitar entrega de medio
- RF 40. Cerrar entrega de medio correspondiente a su responsable
- RF 41. Aprobar entrega de medio a partir de una solicitud existente
- RF 42. Generar notificación de solicitud de entrega de medio a jefe inmediato
- RF 43. Generar notificación de cierre de entrega de medio
- RF 44. Generar notificación de solicitud de movimiento
- RF 45. Generar notificación de cambios en responsabilidad material existente
- RF 46. Generar modelo de responsabilidad material de activos fijos
- RF 47. Generar modelo de responsabilidad material de útiles
- RF 48. Aprobar entrega de otros medios a partir de una solicitud existente
- RF 49. Exportar listado de otros medios en formato PDF
- RF 50. Consultar listado individual de útiles asignados
- RF 51. Consultar listado individual de activos fijos asignados
- RF 52. Consultar listado individual de otros medios asignados
- RF 53. Visualizar reporte gráfico de activos fijos por locales de la DGT
- RF 54. Visualizar reporte gráfico de útiles por locales de la DGT

### **2.2.2 Requisitos no funcionales**

Los requisitos no funcionales (RNF) son las capacidades o cualidades que el sistema debe tener. Representan las cualidades que ayudan y aportan a la experiencia de los usuarios finales. Aunque no definen la funcionalidad, influyen en gran medida en el éxito del producto (Fowler, y otros, 2002). Para el desarrollo del módulo se definieron los siguientes requisitos no funcionales teniendo en cuenta el estándar IEEE Std 830-1993 (IEEE, 1993).

#### **RNF de Apariencia o interfaz externa.**

- El sistema tendrá colores grises, tipografía Arial 10 y con los íconos de colores que resalten su funcionalidad.
- La interfaz será tabulada, con acceso mediante menú lateral visible en todo momento, utilizará una barra superior con migas de página para indicar al usuario donde se encuentra en cada momento.
- Los accesos y funciones comunes como los datos de usuario y cambio de contraseña estarán visibles en un panel inferior y todos los elementos estarán etiquetados con una tipografía uniforme y notable.

## *Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles*

- Las áreas estarán divisibles en el árbol de funcionalidades y panel de trabajo. El árbol podrá esconderse o no a disposición del usuario.
- La visualización de datos debe estar limitada a no más de 30 filas por página con paginación en la parte inferior. Durante la carga de datos se debe mostrar un mensaje al usuario indicando que se están cargando datos del servidor. Ante fallas de la comunicación se le debe informar al usuario las posibles causas.

### **RNF de Usabilidad**

La aplicación debe permitir a los nuevos usuarios familiarizarse rápidamente con el sistema. La utilización de menús y pestañas debe permitir que el sistema se comporte de forma similar a las aplicaciones de escritorios. Deberá hacerse uso apropiado de las ventanas y mensajes.

### **RNF de Rendimiento**

La aplicación debe ejecutarse de forma rápida sobre navegadores web que soporten JavaScript. Los tiempos para las operaciones de visualización de datos debe ser menor a seis segundos para 80 usuarios conectados concurrentemente.

### **RNF de Portabilidad**

El sistema estará basado en herramientas multiplataforma y libres. La utilización de datos centralizados y el acceso mediante los navegadores especificados en el RNF de Software, debe permitir que los usuarios puedan operar de forma similar desde cualquier estación de trabajo. Los informes exportados deberán ser en ficheros PDF para aumentar la portabilidad de la información sin importar la tecnología desde la cual se accede a la aplicación.

### **RNF de Seguridad**

Se utilizará el método de acceso basado en roles y usuarios (RBAC), cada usuario estará asociado a uno o varios roles que tienen a su vez, permisos sobre la aplicación. El sistema debe ser accedido solo mediante comunicación segura a través de la red, haciendo uso del protocolo https. Las credenciales de acceso de los usuarios deben ser comprobadas mediante el directorio LDAP de la UCI y las mismas no deben persistirse de ninguna forma dentro de la aplicación.

## *Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles*

### **RNF de Software**

Los requisitos de software son:

En el nodo servidor de aplicación:

- Un sistema operativo con arquitectura de 32 u 64 bits con soporte para Apache.
- Apache Server versión 2 o superior.

En el nodo servidor de base de datos:

- Un sistema operativo con arquitectura de 32 u 64 bits con soporte para PostgreSQL.
- PostgreSQL versión 9.1.2 o superior.

En el nodo cliente:

- Un navegador web con soporte JavaScript (Opera, Mozilla Firefox, Google Chrome, Internet Explorer 8 o superior).
- Un lector de ficheros PDF con extensión para el navegador (Adobe Acrobat o Foxit Reader).

### **RNF de Hardware**

Para los nodos servidores:

- Procesador dual core a 2.3 GHz o superior.
- 2 GB de RAM o superior.
- 80 GB de HDD o superior.

Para el nodo cliente:

- Procesador a más de 700 MHz o superior.
- 512 Mb de RAM o superior.

### **Restricciones en el diseño y la implementación**

- El lenguaje de programación será PHP versión 5.4.12
- El marco de trabajo será Symfony versión 1.4.11
- Se utilizará en la vista el marco de trabajo Ext JS 3.2.1
- El modelado se realizará con UML 2 con la herramienta Visual Paradigm 5.1

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

- El IDE de programación será NetBeans 8.0
- La metodología de desarrollo será OpenUP

### 2.2.3 Actores del sistema

Se le llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos, pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo (Rumbaugh, y otros, 1999). En la Tabla 3 se describen los actores del sistema identificados.

Tabla 3. Actores del sistema

Actor del sistema	Descripción
<b>Administrador</b>	Usuario con acceso a todas las interfaces del sistema. Es el encargado de realizar las configuraciones iniciales, definir unidades de medida y los tipos de motivos de entregas.
<b>Especialista Capital Humano</b>	Usuario principal del sistema, realiza la gestión de solicitudes de entregas, asignaciones e informes.
<b>Responsable de medio</b>	Encargado de validar el cierre de la entrega correspondiente a la responsabilidad de cada trabajador y que se asocia con los activos fijos o útiles. Se determina específicamente para cada solicitud de entrega de medios.
<b>Responsable de Grupo Económico</b>	Trabajador del grupo económico responsable de la gestión de los activos fijos, medios y útiles.
<b>Trabajador</b>	Persona a la que se le asignan los activos fijos, medios y útiles siendo responsable de su cuidado.
<b>Jefe superior</b>	Directivo de DGT u otro superior que visualiza los reportes generados por el sistema.

### 2.2.4 Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema, documenta el comportamiento de un software desde el punto de vista del usuario. Por tanto, los casos de uso determinan los requisitos funcionales del sistema, los cuales representan las funcionalidades que un sistema puede ejecutar (Addison Wesley, 2004). Para el establecimiento de los casos de uso del sistema, se tuvo en cuenta el actor Usuario con el caso de uso Autenticar usuario ya que, aunque no pertenece al módulo a desarrollar, agrupa cada uno de los actores del sistema. La Figura 2 presenta el diagrama de casos de uso del sistema:

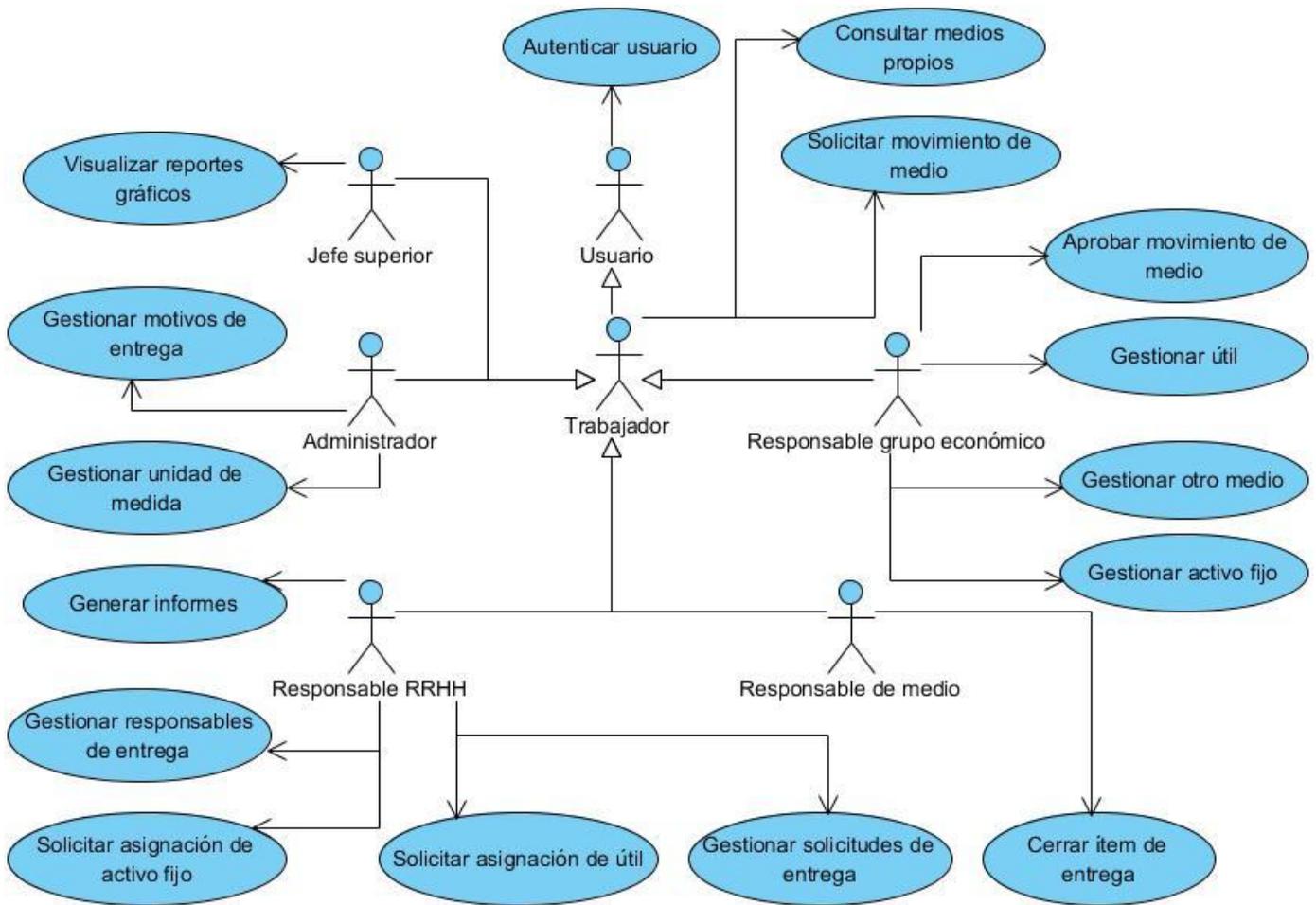


Fig 2. Diagrama de casos de uso del sistema

### 2.2.5 Especificaciones de caso de uso

A continuación, se especifica el caso de uso Gestionar activo fijo, detallando en la Tabla 4 el comportamiento del sistema ante las posibles acciones del actor. Las especificaciones del resto de los casos de uso se muestran en los Anexos de la investigación.

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

Tabla 4. Especificación caso de uso Gestionar activo fijo

<b>Caso de uso</b>	Gestionar activo fijo	
<b>Objetivo</b>	Adicionar, modificar, eliminar, listar, buscar y exportar un listado de activos fijos.	
<b>Actores</b>	Responsable de medio	
<b>Resumen</b>	El responsable de medio en el módulo adiciona, modifica, elimina, lista, realiza búsqueda avanzada sobre los activos fijos y exporta un listado de estos.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El usuario se ha autenticado. El usuario tiene el rol de Responsable de medio.	
<b>Postcondiciones</b>	Se gestiona uno o más activos fijos.	
<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar activo fijo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Accede a la funcionalidad Activos fijos de las operaciones del árbol de funcionalidades.	
2.		<p>Muestra una lista de los activos fijos existentes que estarán ordenados descendientemente según la fecha de creación. En caso de no existir activos fijos ocurre el evento No 1 del flujo alternativo.</p> <p>El sistema permitirá al usuario:</p> <ul style="list-style-type: none"> <li>- Adicionar un nuevo activo fijo (Ver sección Adicionar activo fijo)</li> <li>- Modificar activo fijo (Ver sección Modificar activo fijo)</li> <li>- Eliminar activo fijo (Ver sección Eliminar activo fijo)</li> <li>- Listar activo fijo (Ver sección Listar activo fijo)</li> <li>- Realizar búsqueda avanzada de activo fijo (Ver sección</li> </ul>

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

		<p>búsqueda avanzada de activo fijo)</p> <ul style="list-style-type: none"> <li>- Exportar listado de activos fijos en formato PDF (Ver sección Exportar listado de activos fijos)</li> </ul>
<b>Flujos alternos</b>		
<b>No 1 No existen activos fijos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra la lista activos fijos vacía con el mensaje: "No existen activos fijos definidos".
<b>Sección 1: "Adicionar activo fijo"</b>		
<b>Flujo básico Adicionar activo fijo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Escoge la opción adicionar activo fijo.	
2.		Muestra un formulario para el registro del activo fijo. El formulario contiene los campos de: nombre, número de inventario, estado y descripción.
3.	Introduce los datos en los campos del formulario.	
4.		<p>Valida los datos introducidos por el usuario. Si algunos de los datos están sin llenar se produce el flujo alternativo No. 2.</p> <p>Se registran los datos del activo fijo en el sistema. Concluye el caso de uso.</p>
<b>Flujos alternos</b>		
<b>Nº 2 Campos sin llenar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Por cada campo vacío muestra el mensaje: "Este campo es obligatorio".
2.	Comienza el paso 2 del flujo básico.	
<b>Sección 2: "Modificar activo fijo"</b>		
<b>Flujo básico Modificar activo fijo</b>		

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

1.	Escoge la opción “Modificar” en la barra de herramientas.	Si no se ha seleccionado un activo pasa al flujo alternativo No 3. Muestra un formulario con todos los datos del activo para ser modificados.
2.	Modifica los datos del formulario.	
3.		Registra los nuevos datos en el sistema. Concluye el caso de uso.
<b>Flujos alternos</b>		
<b>Nº 3 No se ha seleccionado ningún activo fijo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje que indica que no se ha seleccionado ningún activo fijo.
2.	Selecciona un activo fijo de la lista de activos e inicia nuevamente el flujo básico.	
<b>Sección 3: “Eliminar activo fijo”</b>		
<b>Flujo básico Eliminar activo fijo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Escoge la opción “Eliminar” en la barra de herramientas.	Si no se ha seleccionado un activo fijo pasa al flujo de eventos No 4. Muestra el mensaje: “¿Está seguro que desea eliminar el activo fijo? Todos los datos dependientes serán eliminados.”
2	Escoge la opción “Si”	
3		Elimina el activo fijo del sistema. Concluye el caso de uso.
4	Escoge la opción “No”	Concluye el caso de uso.
<b>Flujos alternos</b>		
<b>Nº 4 No se selecciona un activo fijo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje que indica que no se ha seleccionado ningún activo fijo.
2.	Selecciona un activo fijo de la lista de activos e inicia nuevamente el flujo básico.	
<b>Sección 4: “Listar activo fijo”</b>		
<b>Flujo básico Listar activo fijo</b>		

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

	Actor	Sistema
1	Desplegar los estados de medios en la sección de Activos fijos.	
2		Muestra el listado de medios agrupados por estados. Concluye el caso de uso.
<b>Sección 5: “Búsqueda avanzada de activo fijo”</b>		
<b>Flujo básico Filtrar activo fijo</b>		
1	En la barra de herramientas escoge la característica por la que se realizará la búsqueda.	
2		Si no se ha seleccionado un estado pasa al flujo de eventos No 5. Muestra un listado con todos los activos fijos coincidentes con la característica seleccionada. Concluye el caso de uso.
<b>Flujos alternos</b>		
<b>Nº 5 No se selecciona un estado</b>		
1		Muestra el listado de todos los activos fijos agrupados por estados. Concluye el caso de uso.
<b>Sección 6: “Exportar listado de activo fijo”</b>		
<b>Flujo básico Exportar listado de activo fijo</b>		
1	Seleccionar opción generar reporte en barra de herramientas	
2		Muestra en formato PDF el listado general de todos los activos fijos especificando nombre, número de inventario, estado y descripción.
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos funcionales</b>	<b>no</b>	RNF de Apariencia o interfaz externa, RNF de Usabilidad, RNF de Rendimiento, RNF de Portabilidad, RNF de Seguridad.
<b>Asuntos pendientes</b>		No aplica

### **2.3 Arquitectura**

La arquitectura de software permite representar de forma concreta la estructura y funcionamiento interno de un sistema. Autores como (Clements, y otros, 2003) la definen como una estructura o estructuras del sistema, que abarca componentes de software, las propiedades visibles externamente de esos componentes, y las relaciones entre ellas.

Al diseñar una arquitectura de software se debe crear y representar componentes que interactúen entre ellos y tengan asignadas tareas específicas, además de organizarlos de forma tal que se logren los requerimientos establecidos. Se puede iniciar con patrones de soluciones ya probados, y utilizar modelos que han funcionado. Estas soluciones probadas se conocen como estilos arquitectónicos o patrones arquitectónicos, que van de lo general a lo particular.

Un patrón arquitectónico se enfoca en dar solución a un problema en específico, de un atributo de calidad, y puede abarcar solo parte de la arquitectura (Bosh, 2000).

La arquitectura de la solución propuesta es bajo el patrón Modelo-Vista-Controlador (MVC). El MVC separa el sistema en tres bloques, donde cada uno es responsable de una parte del sistema. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. De esta forma, las capas del modelo, la vista y el controlador se pueden subdividir en más capas (Potencier, y otros, 2008). Las ventajas de usar el patrón MVC son diversas, al proyecto aporta que: separará en capas la aplicación haciendo más fácil su desarrollo, compresión y soporte, permite establecer un solo modelo estandarizado de desarrollo y aplicarlo a cada capa del producto, garantiza aplicar el desarrollo ágil de aplicación web y asegura que las herramientas que se utilicen se acoplen al estándar, se comuniquen y colaboren.

En la solución, la aplicación de este patrón arquitectónico se puede observar de forma general en la Figura 3. El funcionamiento de cada capa puede verse en el diagrama con la distribución de los principales directorios de la aplicación.

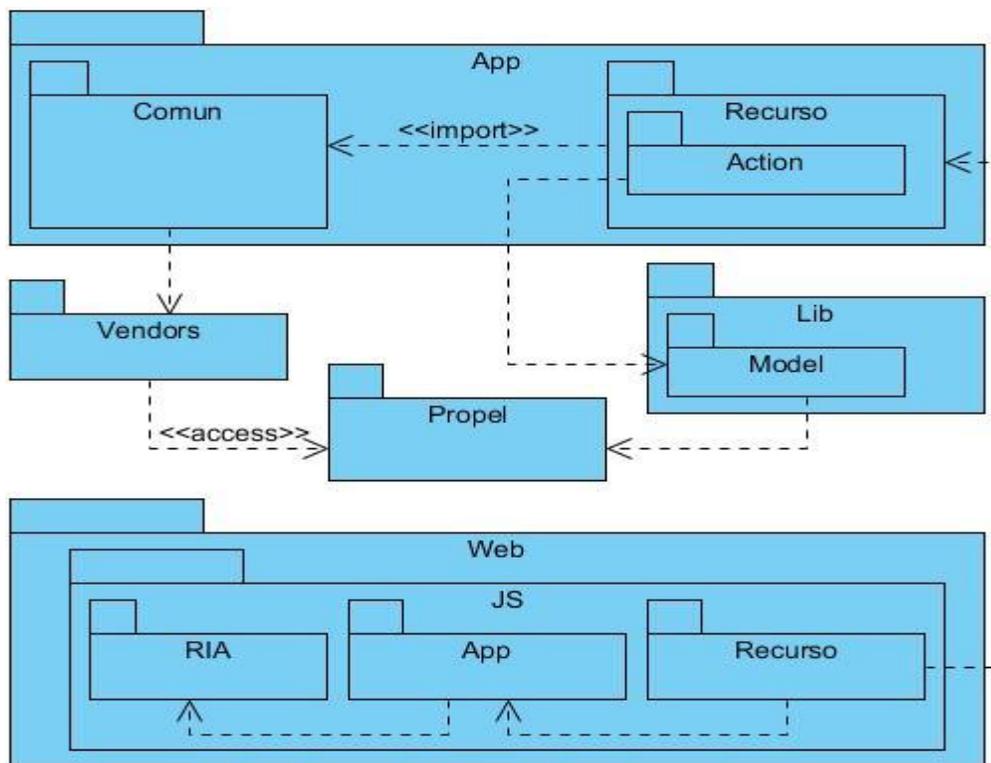


Fig 3. Uso del patrón arquitectónico en el módulo propuesto

El paquete Recurso dentro del directorio Web contiene todas las vistas JavaScript que se muestran al usuario, que heredan directamente del controlador central de la aplicación, que a su vez utiliza el directorio RIA donde se encuentra el núcleo de EXT JS. Recurso depende de las acciones que se encuentran en el directorio APP, donde se especifica la lógica del negocio, haciendo uso del directorio Vendors que incluye los archivos de Symfony y que se comunica con la base de datos a través el paquete Propel.

**Vistas:** las vistas están desarrolladas con JavaScript, utilizando el marco de trabajo Ext JS. De esta forma la aplicación es más ligera ya que ejecuta dicho código del lado del cliente. En la vista se tiene el archivo *dserp.js* que contiene todas las configuraciones globales y actúa como instancia única, permitiendo la herencia de sus configuraciones hacia los archivos del paquete módulo, donde se definen las vistas propias de la aplicación. Dentro de cada módulo existe un paquete *browsers*, donde se guardan los archivos que permiten listar datos, y un paquete *forms* donde se definen los formularios.

**Controlador:** las clases controladoras se definen dentro del paquete *apps* de la aplicación. Todas las clases controladoras se implementan dentro del paquete *actions* de cada módulo. Estas clases heredan del controlador frontal de *symfonysfAction*, que actúa como instancia única para procesar las peticiones de los usuarios y manejar los recursos de Symfony.

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

**Modelo:** el modelo está descrito en el paquete *libs*, donde se crean las interfaces de las clases persistentes, que se encuentran en el directorio *model*, a partir del Mapeo de Objetos Relacional que realiza Propel dentro del directorio OM.

### Estilo arquitectónico

El estilo arquitectónico es Cliente-Servidor, que define la relación entre dos o más aplicaciones donde una de ellas es el servidor y otras actúan como clientes consumiendo datos de dicho servidor. Permite distribuir sistemas de software a través de una red de conexión que garantiza que múltiples clientes accedan incluso a múltiples servidores. Este estilo permite la multiconexión a la aplicación y la distribución de responsabilidades de procesamiento, representación y almacenamiento (Gerner, 2011).

### 2.4 Patrones de diseño

Para el desarrollo de software un elemento fundamental y de buena práctica, lo constituye el uso de patrones, estos representan una excelente manera de trabajar en base a la calidad. Además un patrón de diseño es una buena práctica documentada o solución, que se ha aplicado con éxito en múltiples ambientes para erradicar problemas comunes de diseño de software, con una probada efectividad y con características de reutilización (Fowler, y otros, 2002). El uso del marco de trabajo Symfony como escenario de desarrollo, por las características que tiene, brinda una gama de patrones que permiten garantizar que el software que se desarrolla tenga una línea arquitectónica establecida, robusta e íntegra, y que pueda establecer una correcta comunicación con los elementos que puedan interactuar internamente.

#### Patrones GOF

**Singleton:** este patrón se implementa mediante la clase *sfRouting* una variable global definida en el sistema. El *sfRouting* contiene el método *getInstance ()* que ejecuta todas las peticiones que se realizan a la aplicación a través del controlador *sfWebFrontController*. Mientras que la variable global *dserp*, se comporta como instancia única del sistema que se encarga de construir todas las clases, las funciones y acciones dentro de las vistas generadas por Ext JS.

**Comando:** este patrón se manifiesta a través de la clase *sfWebController*, encargada de establecer el módulo y la acción que se usará por cada petición del usuario.

**Decorador:** utilizado en la clase *sfView*, padre de todas las vistas y que permite agregar funcionalidades dinámicamente.

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

**Registro:** este patrón se aplica en la clase *sfConfig*, que es la encargada de acumular todas las variables de uso global en el sistema, es una manera fácil y rápida de compartir datos y objetos en la aplicación, permitiendo que no sea necesario conservar muchos parámetros y reduce el uso de variables globales.

### Patrones generales de software para asignación de responsabilidades

**Experto:** este patrón se introduce mediante el uso de la librería *Propel* para mapear la Base de Datos. Esta librería, es usada en la capa del modelo, maneja toda la lógica de los datos, generando clases a partir del modelo racional con todas las funcionalidades necesarias para el desarrollo orientado a objetos.

**Creador:** en la clase *Actions* se evidencia dicho patrón; en la misma se encuentran las acciones que crean los objetos de las clases que representan las entidades.

**Controlador:** todas las peticiones desde la web son manipuladas por un solo controlador: *sfActions*, que es el punto de entrada único para la aplicación dentro de un determinado entorno. Este patrón se evidencia en las clases *sfFrontController*, *sfWebFrontController* y *sfContext*.

**Alta cohesión:** el uso del marco de trabajo Symfony provee una alta cohesión para la asignación de responsabilidades y el empaquetamiento de archivos. Este patrón persigue la idea de que los elementos de un componente colaboran para producir algún comportamiento bien definido, preferentemente único en el sistema. Cada uno de los casos de uso contiene los componentes modelo, vista, controlador, los mismos trabajan en alta cohesión colaborando entre sí, al mismo tiempo tratan de inmiscuirse lo menos posible en el comportamiento de los componentes de otros casos de uso. Este patrón plantea que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase.

**Bajo acoplamiento:** este patrón permite fomentar la reutilización de código que indica una menor dependencia entre clases. De tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases. Se evidencia durante la creación de objetos de las entidades con la clase *Actions*, que hereda únicamente del controlador *sfActions*, alcanzando un bajo acoplamiento de las clases.

Estos dos últimos patrones no deben tratarse de forma independiente pues poseen relaciones que mantienen el equilibrio entre clases, equilibrando las responsabilidades y garantizando que las clases sean creadas con un buen diseño donde los objetos sean capaces de interactuar.

### 2.5 Paquetes y relaciones

La aplicación tiene tres paquetes que agrupan sus tres módulos principales. Estos tienen relación entre ellos, los paquetes de Organización Empresarial y Capital humano acceden al paquete de Seguridad y los

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

tres dependen de Ext JS. A la vez, acceden a App, donde se implementa la lógica de la aplicación haciendo uso de Propel para el acceso a datos. El diagrama de paquetes mostrado en la Figura 4, muestra la relación.

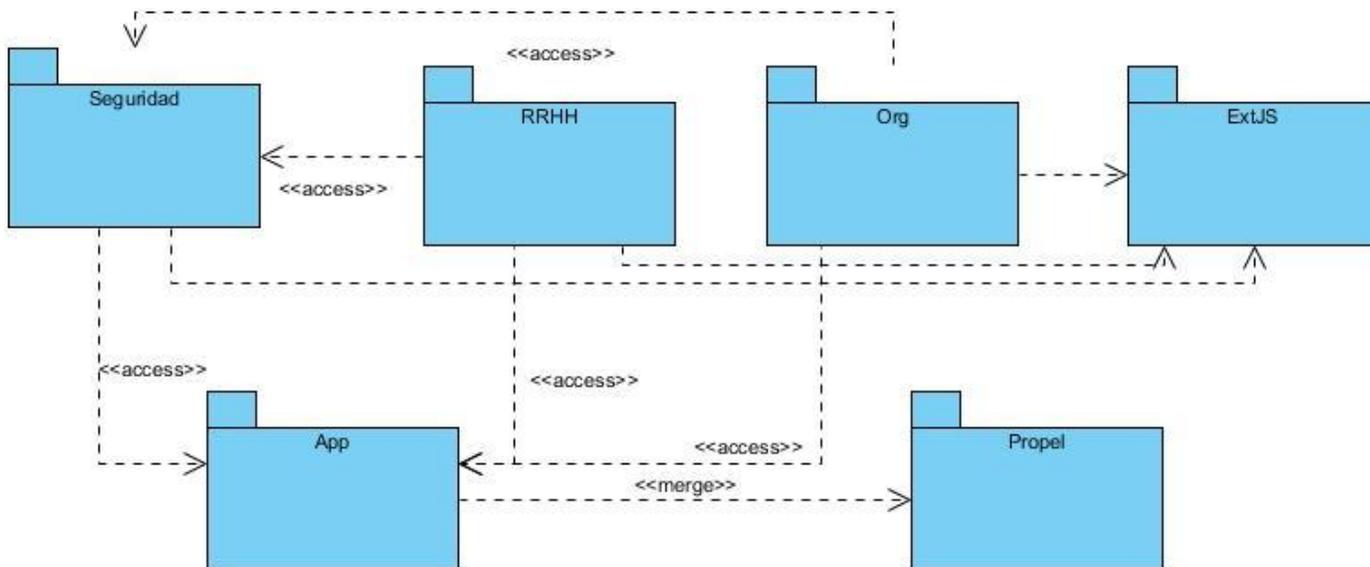


Fig 4. Diagrama de paquetes del módulo propuesto

### 2.6 Diagrama de clases de diseño

El Diagrama de Clases es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama y se modifica para satisfacer los detalles de las implementaciones. Describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Además de los componentes que se encargarán del funcionamiento y la relación entre uno y otro (Fowler, 2007).

Los diagramas de clases de la aplicación pueden comprenderse a través de la siguiente estructura. Cuando se realiza una petición desde el controlador frontal de Symfony `sfController`, se genera una vista definida en `dserp.js`, archivo que constituye la variable global encargada de la presentación al usuario. Por cada caso de uso se generan los contenedores `config`, que agrupan un caso de uso en la vista e importan listas o árboles que son presentados al usuario. Además, importan formularios de inserción de datos. En la aplicación se tiene una clase controladora en cada una de las acciones, lo que permite utilizar el patrón comando para apuntar cada vista a acciones concretas y especializar cada una de las clases controladoras. Por tanto, cada formulario o tabla se comunica con una acción determinada, evidenciando el bajo acoplamiento. Cada clase controladora utiliza el modelo según la lógica definida y este a su vez, se comunica con la base de datos.

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

Basándose en los casos de uso Gestionar útil y Solicitar asignación de activo fijo, se modelaron los diagramas de clase del diseño correspondientes, los que se muestran en la Figura 5 y Figura 6 respectivamente. Nótese que se tienen establecidas, según los patrones de responsabilidades elegidos, las relaciones entre la vista, el controlador y el modelo.

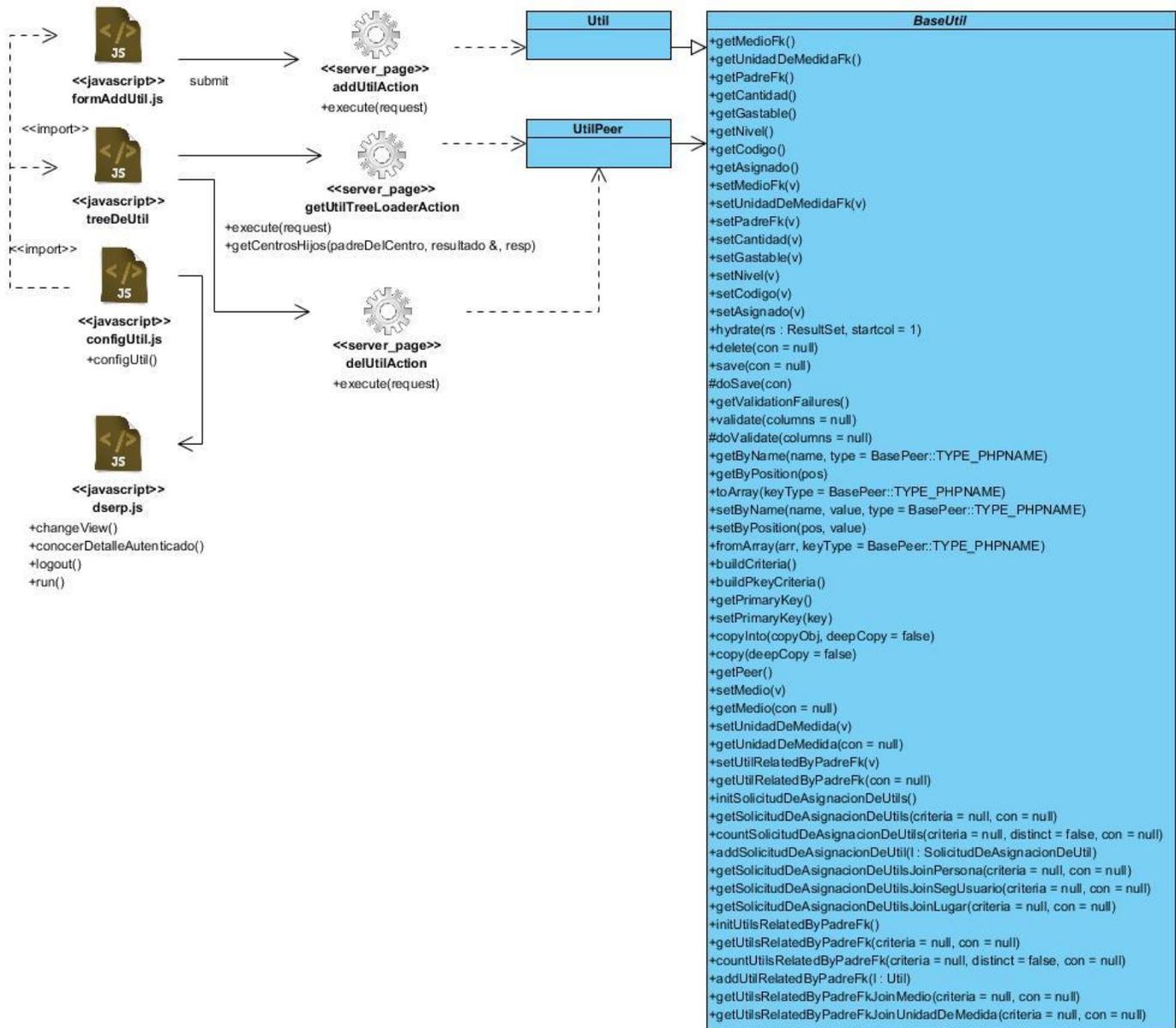


Fig 5. Diagrama de Clases del Diseño caso de uso Gestionar Útil

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

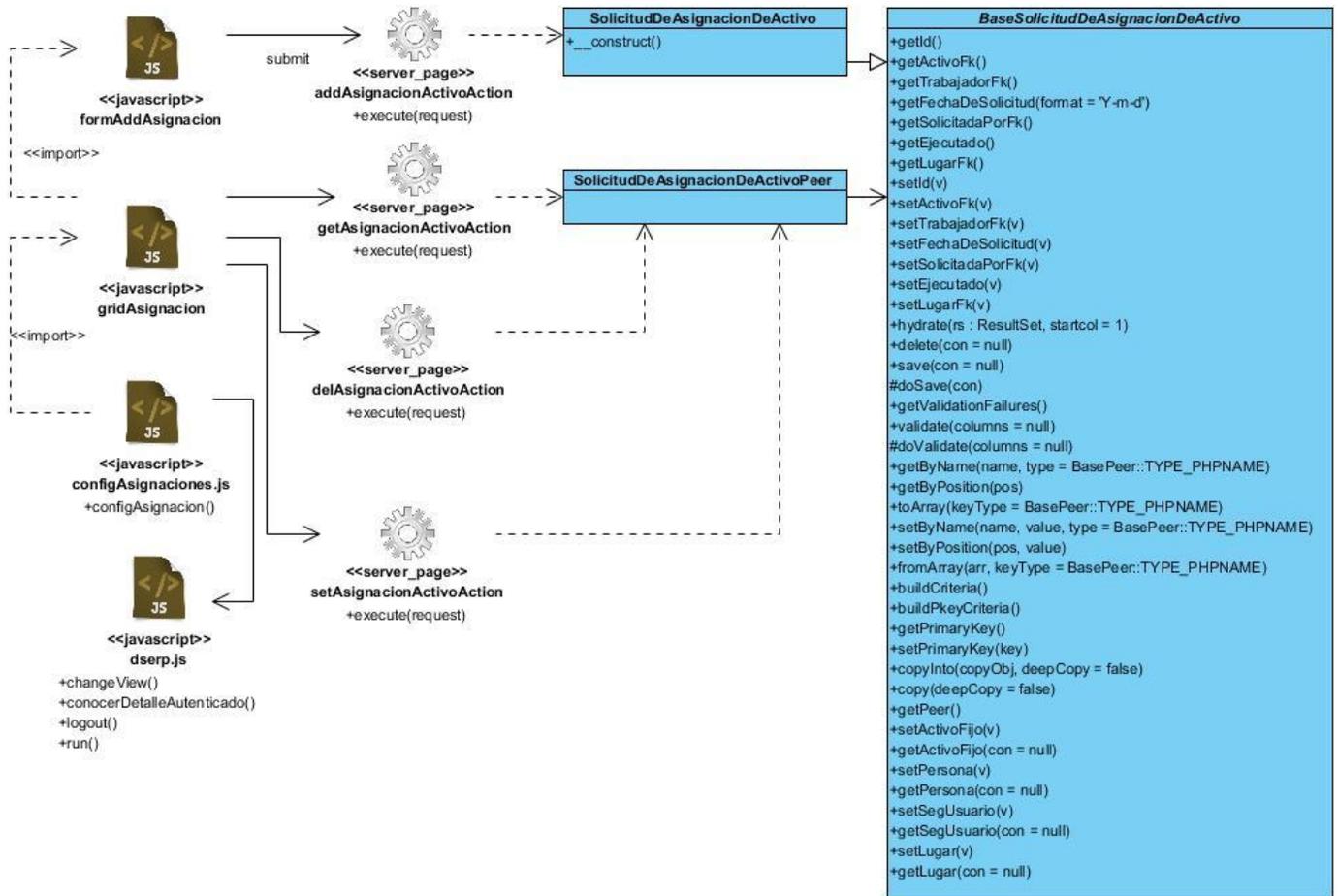


Fig 6. Diagrama de clases del Diseño caso de uso Solicitar asignación de Activo Fijo

### 2.7 Diseño físico de la base de datos

El modelo de datos está descrito por el diagrama entidad-relación, que denota el modo físico en que está dispuesta la información dentro de la base de datos. La Figura 7 muestra el modelo entidad-relación del módulo propuesto.

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

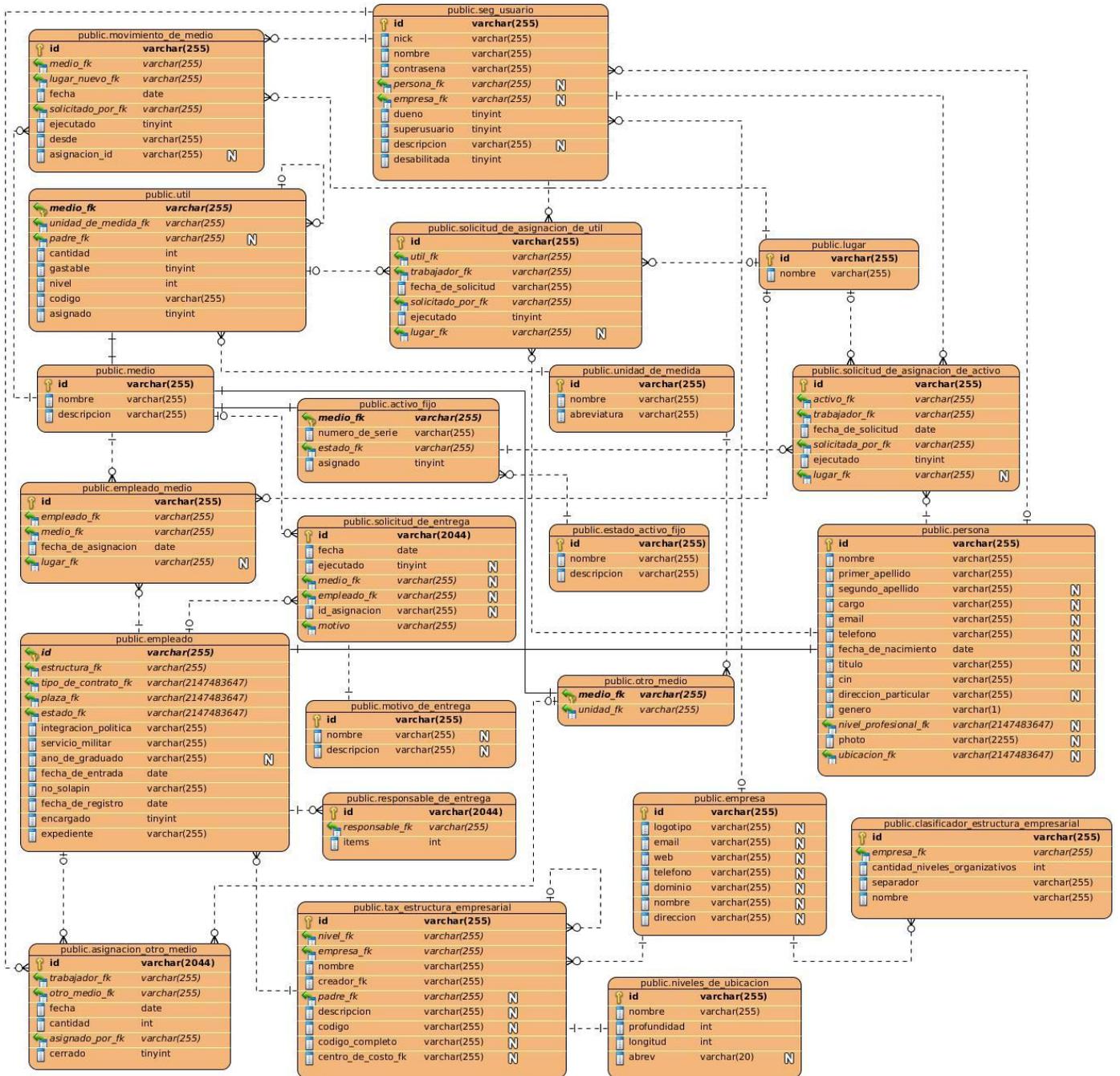


Fig 7. Diagrama Entidad-Relación del módulo propuesto

## Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles

La descripción de cada entidad identificada en el diagrama entidad-relación se muestra en la Tabla 5, todas pertenecientes al esquema public.

Tabla 5. Descripción de entidades

Tabla	Descripción
Movimiento_de_medio	Registra los movimientos ejecutados desde un lugar a otro.
Útil	Entidad especializada que registra los útiles heredando el nombre y la descripción de la entidad general Medio.
Empleado_medio	Registra las relaciones entre los empleados y los medios de cualquier tipo.
Empleado	Registra los empleados de la DGT, hereda directamente de la entidad Persona.
asignacion_otro_medio	Registra las asignaciones de otros medios a los empleados.
solicitud_de_asignacion_de_util	Registra las solicitudes de asignaciones de útiles a los empleados.
solicitud_de_asignacion_de_activo	Registra las solicitudes de asignaciones de activos fijos a los empleados.
Persona	Entidad general que tiene los datos de la persona.
solicitud_de_entrega	Contiene las solicitudes de entrega de medios realizadas.
responsable_de_entrega	Registra los responsables de asignaciones y entregas de medios.
empresa	Registra los datos de la DGT
seg_usuario	Guarda los datos de los usuarios, pertenece al módulo de Seguridad del sistema.
unidad_de_medida	Registra las unidades de medida con las que trabaja la DGT.
Lugar	Registra los lugares físicos donde se encuentran los activos fijos.
activo_fijo	Entidad especializada que guarda los activos fijos obteniendo el nombre y la descripción de la entidad Medio.
Medio	Entidad general para guardar todos los medios de la DGT.
tax_estructura_empresarial	Contiene las estructuras organizativas de la DGT.
otro_medio	Entidad especializada en el registro de otros medios que tiene asignados la dirección (órdenes de servicio, sellos y otros). Hereda de la entidad medio.
niveles_de_ubicación	Define los distintos niveles de la estructura empresarial de la dirección (áreas, brigadas, departamentos y otros).
estado_activo_fijo	Contiene los diferentes estados en lo que puede encontrarse un

	activo fijo.
motivo_de_entrega	Contiene los distintos motivos de entrega de los medios
clasificador_estructura_empresarial	Define el nomenclador a utilizar para la estructura organizativa de la DGT.

## 2.8 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. El modelo de despliegue suministra la base para la comprensión de la distribución física de un sistema a través de nodos, indicando de qué forma se sitúa el software en el hardware que lo contiene.

Para una mejor comprensión del ambiente donde se desplegará la solución se propone en la Figura 8 el diagrama de despliegue.

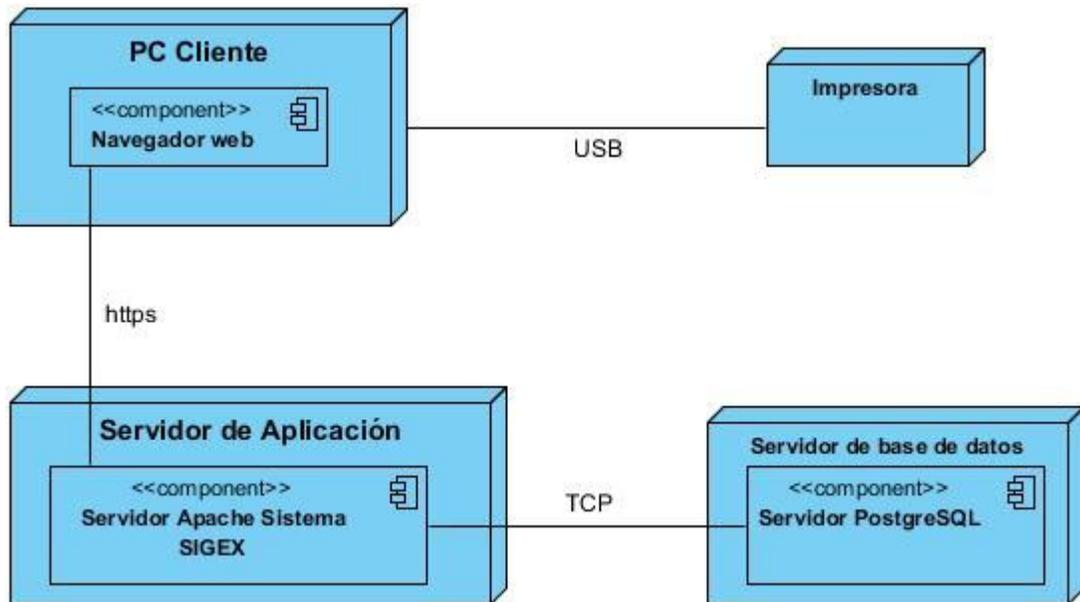


Fig 8. Diagrama de despliegue

Los cuatros nodos representados en el diagrama de la figura 8 son: la PC cliente equipada con un navegador web que soporte JavaScript, una impresora necesaria para la impresión de los reportes emitidos por el sistema, un servidor de aplicaciones en el que estará montado el sistema SIGEX y el servidor de base de datos en el que estará instalado PostgreSQL. La PC cliente y el servidor de

## *Capítulo 2 Descripción del Módulo de gestión de activos fijos, medios y útiles*

aplicaciones se conectarán a través del protocolo HTTPS y la conexión entre servidores se realiza utilizando sockets enviados mediante TCP.

### **2.9 Conclusiones parciales**

Con el modelado del dominio y su descripción, la obtención de los 7 requisitos no funcionales y los 54 funcionales y el diseño del diagrama de casos de uso del sistema, se propicia el entendimiento entre el equipo de desarrollo y el cliente, en función de lo que el sistema debe realizar y las características que debe poseer. Se define la propuesta de solución en la que se describen los principales procesos a automatizar en la gestión de activos fijos, medios y útiles de la DGT. El diseño de los diagramas de casos de uso del sistema brindó la posibilidad de mostrar gráficamente los procesos y su interacción con los actores. Se realizó la representación de las clases del diseño y la relación que se presentan entre ellas, mostrando la estructura interna del sistema y enumerando los métodos que se deben implementar y dónde deben estar posicionados. A su vez se obtuvo la estructura de la base de datos a partir de la confección del diagrama de entidad-relación. Se confeccionó además el diagrama de despliegue propiciando una visión de cómo debe quedar distribuido el sistema físicamente. La obtención de los artefactos correspondientes al flujo de trabajo de diseño, sirve como material de referencia para futuras ampliaciones y modificaciones del sistema y como guía certera para comenzar su implementación.

## **Capítulo 3 Implementación y pruebas del módulo de gestión de activos fijos, medios y útiles**

En el presente capítulo se describe la fase de implementación del módulo a partir de los elementos de interfaz de usuario, la gestión de errores y el diagrama de componentes. Además, se evidencian los resultados de las pruebas realizadas al software en los niveles de integración y aceptación.

### **3.1 Modelo de implementación**

Un modelo de implementación es un conjunto de artefactos que permiten documentar la construcción de un software en su etapa de codificación, a partir de la colección de componentes y subsistemas de implementación que lo conforman (McConell, 2012). Describe tanto los elementos del modelo de diseño, como clases y otros artefactos que se comportan como componentes dentro del sistema, además de descripciones útiles para comprender partes o funciones de la aplicación.

#### **3.1.1 Interfaz de la aplicación**

La interfaz de usuario ha sido diseñada con colores en escalas grises, usando patrones uniformes en la distribución de los principales elementos de la pantalla, sus principales características son:

1. La interfaz es tabulada, cada pestaña podrá cerrarse de forma independiente o en conjunto.
2. El árbol de funcionalidades estará en la parte izquierda de la pantalla.
3. Los botones de acceso rápido del usuario están en la parte inferior derecha de la pantalla.
4. Todas las pestañas refrescarán los datos cada vez que estén activas.

#### **3.1.2 Estilo de programación**

Un estilo de programación se refiere a la forma en que se da formato al código fuente (Calcifer, 2012). A diferencia de la sintaxis de un programa que son reglas estáticas que obligatoriamente hay que seguir, un estilo de programación está constituido por directrices que ayudan a obtener programas más legibles. Si bien no existen estilos de programación correctos o incorrectos es aconsejable la adopción de un conjunto de normas para la escritura de programas. En la aplicación se utiliza el estilo propuesto por Symfony PSR-2 (Potencier, y otros, 2008), que tiene las características de:

Estructura

- Añade un solo espacio después de cada delimitador coma.
- Añade un solo espacio alrededor de los operadores (==, &&, ...).

## Capítulo 3 Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles

- Añade una coma después de cada elemento del arreglo en un arreglo multilínea, incluso después del último.
- Añade una línea en blanco antes de las declaraciones *return*, a menos que el valor devuelto solo sea dentro de un grupo de declaraciones (tal como una declaración *if*).
- Usa llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga.
- Define una clase por archivo — esto no se aplica a las clases ayudante privadas, de las cuales no se tiene la intención de crear una instancia desde el exterior y por lo tanto no les preocupa la norma PSR-0.
- Declara las propiedades de clase antes que los métodos.

### Convenciones de nomenclatura

- Utiliza mayúsculas intercaladas —sin guiones bajos— en nombres de variable, función, método o argumentos.
- Usa guiones bajos para nombres de opción y nombres de parámetro.
- Utiliza espacios de nombres para todas las clases.
- Prefija las clases abstractas con *Abstract*.
- Sufija las interfaces con *Interface*.
- Sufija las características con *Trait*.
- Sufija las excepciones con *Exception*.
- Utiliza caracteres alfanuméricos y guiones bajos para los nombres de archivo.

### Documentación

- Añade bloques *PHPDoc* a todas las clases, métodos y funciones.
- Omite la etiqueta *@return* si el método no devuelve nada.
- Las anotaciones *@package* y *@subpackage* no se utilizan.

#### 3.1.3 Gestión de errores

El tratamiento de errores se realiza a dos niveles, en las vistas y en el servidor. En las vistas se utiliza las propias funciones del marco de trabajo Ext JS, mostrándose en la Figura 9. Esto se realiza para tratar los errores durante la inserción de datos con el objetivo de evitar realizar peticiones con errores al servidor y para cumplir con el requisito de robustez del sistema. Los errores de datos son comunicados inmediatamente al usuario.

## Capítulo 3 Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles

```
xtype: 'textfield',
anchor: '95%',
allowBlank: 'false',
name: 'nombrePrioridad',
emptyText: 'Nombre de la prioridad',
blankText: 'Debe introducir un nombre para la prioridad',
fieldLabel: 'Nombre',
```

Fig 9. Tratamiento de errores en la captura de datos

Además, durante los errores excepcionales generados por el servidor, el cliente muestra al usuario las posibles causas evitando que la aplicación incumpla con los requisitos de corrección.

En la Figura 10 se ejemplifica como en el servidor los errores de manipulación de datos y acceso a recursos del sistema son capturados a través de excepciones y mostrados a los usuarios con mensajes coherentes.

```
try {
    $Prioridad->delete();
    $con->commit();
    SegTraza::crearTraza('Eliminación satisfactoria de la prioridad ' .
        $nombre, 3, 'Eliminación de prioridad de mensajes');
    return $this->renderText("{success : true}");
} catch (Exception $e) {
    $con->rollback();
    return $this->renderText("{success:false, msg:'el objeto no ha podido
        ser eliminado, por favor intente;ntelo otra vez.'}");
}
```

Fig 10. Tratamiento de errores con excepciones en el servidor

### 3.1.4 Diagrama de componentes

Los diagramas de componentes ilustran las piezas del software que conformarán un sistema. Un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre dichos elementos. Es un grafo de componentes unidos a través de relaciones que pueden ser de compilación o de ejecución, y además se pueden representar las interfaces de esos componentes (Pressman, 2008).

## Capítulo 3 Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles

Es otra forma de representar una vista estática del sistema, que muestra la organización y dependencia que existe entre los componentes físicos que se necesitan para ejecutar la aplicación, sean estos componentes de código fuente, librerías, binarios o ejecutables.

El módulo de gestión de activos fijos, medios y útiles de SIGEX se refleja en la Figura 11 mediante el diagrama de componentes.

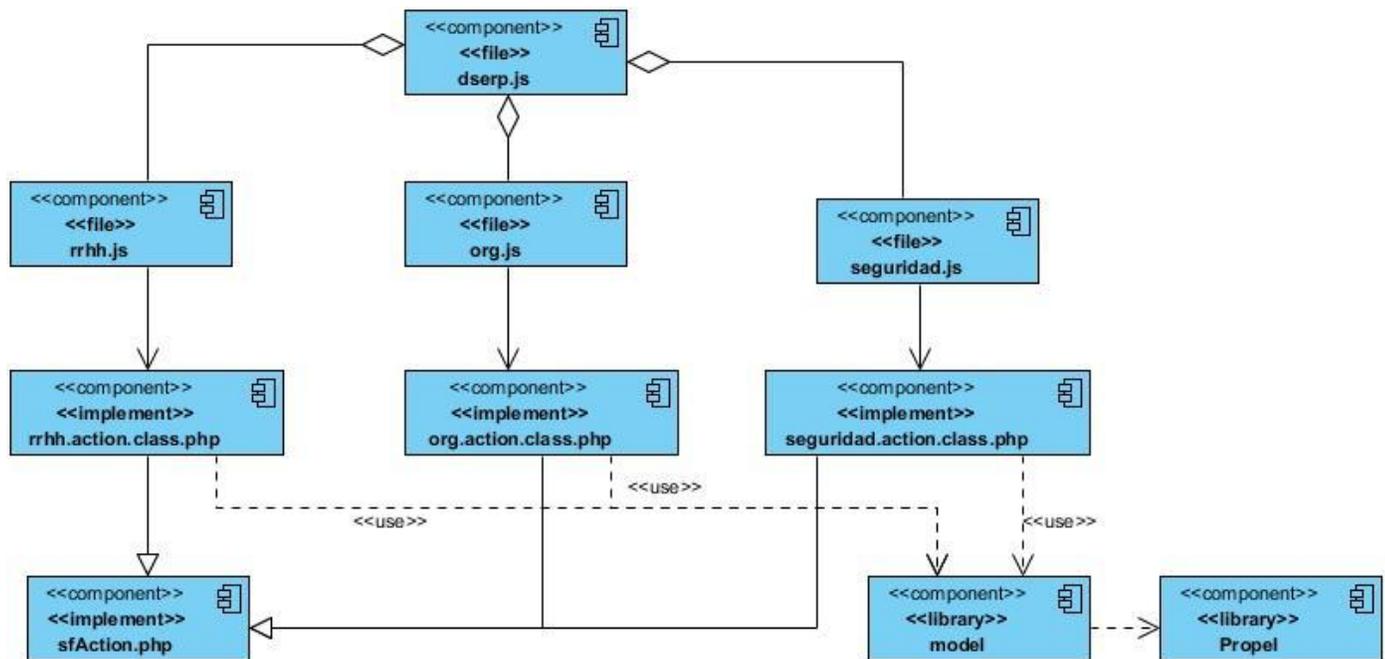


Fig 11. Diagrama de componentes

### 3.2 Diseño de pruebas

Las pruebas son actividades que se realizan para garantizar la calidad de las aplicaciones de software a partir de comprobaciones del funcionamiento de la misma o sus componentes. Las pruebas representan la revisión final de las especificaciones del diseño y la codificación. El resultado de las pruebas determina si el software cumplió o no los requisitos establecidos durante su definición (Addison Wesley, 2004).

Existen 4 niveles de pruebas, estos son: pruebas de componente (o pruebas unitarias), pruebas de integración, pruebas de sistema y pruebas de aceptación (Juristo, y otros, 2005). Los niveles seleccionados para las pruebas son integración y aceptación.

#### 3.2.1 Pruebas de integración

Las pruebas de integración son aquellas que permiten probar en conjunto distintos subsistemas funcionales o componentes del sistema para verificar que interactúan de manera correcta y que se ajustan a los requisitos especificados (sean estos funcionales o no). Para la realización de estas pruebas existen

## *Capítulo 3 Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles*

dos enfoques: incremental (ascendente o descendente) y Big Bang (Rodríguez Tello, 2012). Se realizará la integración incremental ascendente para ir incorporando cada una de las funcionalidades desarrolladas hasta conformar el módulo. Una vez terminado el módulo se integrará con el sistema evidenciándose la integración Big Bang.

La integración incremental se aplica de forma que el programa se construye y prueba en pequeños incrementos, lo cual permite aislar y corregir los errores que surjan. La integración incremental ascendente comienza con la construcción y prueba de módulos atómicos. Los componentes se integran desde los niveles más bajos hacia arriba, de forma que los módulos de bajo nivel se combinen en grupos que realicen una función del software y se prueba el grupo (Quesada, 2009).

La integración Big Bang permite combinar todos los componentes y probar el sistema como un todo. De esta forma se integrará el módulo desarrollado al sistema realizando comprobaciones de forma íntegra a la aplicación.

Las pruebas en este nivel tienen como propósito la detección de defectos en las interfaces de usuarios. Para ello se aplicará como tipo de prueba las pruebas funcionales ya que son las encargadas de verificar que cada funcionalidad cumple con lo que se ha especificado. Para la ejecución de las pruebas funcionales se utilizará como herramienta de apoyo los casos de prueba basados en CU. Estos serán diseñados teniendo en cuenta la validación de entradas y salidas de cada una de las funcionalidades. Por cada CU se debe realizar un caso de prueba dividido en secciones y escenarios, detallando las funcionalidades descritas en él y describiendo cada variable que recoge el CU en cuestión. Para cada escenario se conformarán matrices de datos con el uso de valores válidos e inválidos, evidenciándose el uso de la técnica partición de equivalencia del método de prueba caja negra.

Las pruebas de caja negra son aquellas que se realizan sobre la interfaz del software concentrándose solamente en los RF. Pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. A la vez permiten derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los RF de un sistema (Pressman, 2008).

### **Ejecución de las pruebas de integración**

La aplicación de las pruebas durante la integración incremental ascendente estuvo reflejada durante el desarrollo del módulo; pues a medida que se iban realizando o modificando cada uno de los componentes estos se iban probando y corrigiendo los errores detectados.

Las pruebas de integración Big Bang se aplicaron una vez integrado todo el módulo dentro del sistema para probar de forma íntegra la aplicación. Para ello se realizaron tres iteraciones de pruebas funcionales haciendo uso de los 16 casos de pruebas diseñados (uno por cada caso de uso). En la primera iteración

## *Capítulo 3 Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles*

se detectaron un total de 11 No Conformidades (NC) desglosadas en dos de funcionalidad, una de acceso a datos, tres de validación y cinco de ortografía. En la segunda iteración se detectaron cuatro NC, de ellas una de funcionalidad, una de validación y dos de ortografía. En la tercera iteración no se detectaron nuevas NC y se comprobó que fueron corregidas las NC identificadas en las iteraciones anteriores. Estas pruebas permitieron comprobar el correcto funcionamiento del módulo una vez integrado al sistema y que se cumplieron todos los requisitos definidos. En la Figura 12 se muestran las NC detectadas en cada una de las iteraciones realizadas de acuerdo a su clasificación (funcionalidad, acceso a datos, validación y ortografía). En las Tablas 6, 7 y 8 se evidencia el diseño del caso de prueba correspondiente al caso de uso Gestionar activo fijo; el resto de los casos de prueba se encuentran en los Anexos de la investigación.

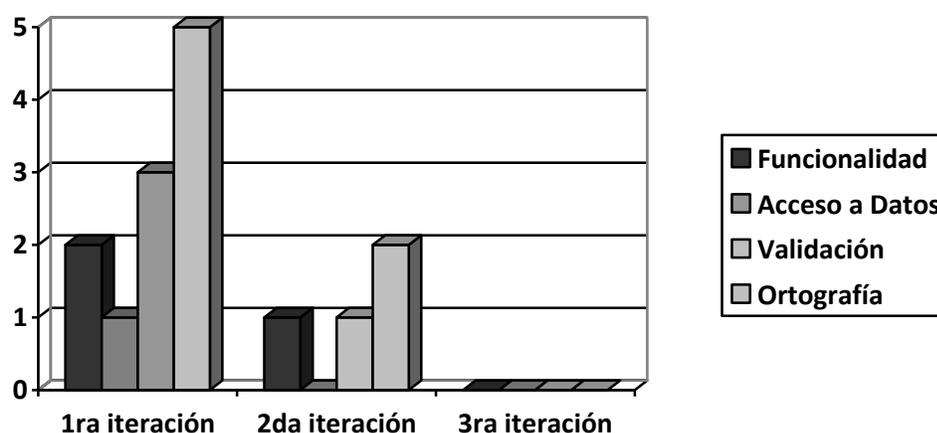


Fig 12. No Conformidades detectadas

En la realización de los casos pruebas se utilizan los siguientes nomencladores en el juego de variables: V (válido), I (Inválido), N (Campo vacío), N/A (Sin utilización de variables de entrada).

Tabla 6. Caso de prueba del caso de uso Gestionar activo fijo

Sección de prueba	Escenarios a probar	Descripción
<b>Adicionar activo fijo</b>	Datos de activo fijo correctos	Se introducen correctamente todos los datos del activo fijo
	Campos vacíos	Se dejan sin llenar los campos de No. de Inventario y estado.
<b>Modificar activo fijo</b>	Activo fijo sin seleccionar	Se procese a realizar la modificación sin seleccionar un activo fijo
<b>Eliminar activo fijo</b>	Eliminar activo fijo	Se procede a eliminar un activo fijo.

## *Capítulo 3 Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles*

Tabla 7. Variables de prueba del caso de uso Gestionar activo fijo

No	Nombre del campo	Clasificación	Nulo	Descripción
1	Nombre	Texto	No	Nombre del activo fijo
2	No. de Inventario	Texto	No	Número de inventario asignado por el grupo económico.
3	Estado	Lista desplegable	No	Estado actual en que se encuentra el activo fijo.
4	Descripción	Texto	No	Descripción completa del activo fijo.

Tabla 8. Matriz de prueba caso de uso Gestionar activo fijo

Escenario	Juego de variables	Respuesta del sistema
<b>Datos del activo fijo correctos</b>	V,V,V,V	Registra un nuevo activo fijo en el sistema
<b>Campos vacíos</b>	V,N,N,V	Muestra un mensaje sobre los campos vacíos indicando que estos son obligatorios.
<b>Activo fijo sin seleccionar</b>	N/A	El sistema muestra un mensaje: "Debe seleccionar un activo fijo"
<b>Eliminar activo fijo</b>	N/A	El activo fijo queda eliminado del sistema.

### **3.2.2 Pruebas de aceptación**

La prueba de aceptación es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. En estas pruebas se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique (Pruebas de Software, 2016).

Serán aplicadas en este nivel las pruebas de rendimiento (Carga y Estrés) para comprobar la velocidad con la que el sistema ejecuta una tarea. Además, permite verificar el comportamiento del módulo bajo una demanda excesiva. Para la ejecución de dichas pruebas se utilizará la herramienta de código abierto JMeter, desarrollada para realizar pruebas de rendimiento en aplicaciones web.

También serán realizadas pruebas al módulo por los usuarios finales en su lugar de trabajo, por lo que se les entregará la aplicación en su totalidad. De esta forma se podrá comprobar el funcionamiento de la misma en un entorno real que no puede ser controlado por el desarrollador. Los errores encontrados

## *Capítulo 3 Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles*

durante estas pruebas serán informados periódicamente al desarrollador para su posterior corrección. Estas pruebas son conocidas como pruebas beta.

### **Ejecución de las pruebas de aceptación**

Se realizaron las pruebas de rendimiento utilizando la herramienta JMeter 2.9, simulando peticiones a la aplicación de varios usuarios conectados concurrentemente. Para ello se pobló la base de datos y se midió el tiempo de respuesta de todas las funcionalidades haciendo énfasis en las de mayor nivel de complejidad. Estas son las asociadas a los casos de uso Gestionar activos fijos, Gestionar útiles y Gestionar medios debido al gran volumen de información que manejan sobre la base de datos; ocasionando que el tiempo de respuesta de la aplicación aumente. Estas pruebas fueron realizadas teniendo en cuenta las características establecidas en los requisitos no funcionales de hardware y software para el nodo cliente, el servidor de aplicaciones y el de base de datos.

En la Tabla 9 se muestran los tiempos medios de respuesta arrojados por la herramienta en segundos para distintas cantidades de usuarios conectados concurrentemente.

Tabla 9. Tiempo medio de respuesta arrojado por la herramienta.

<b>Cantidad de usuarios concurrentes</b>	<b>Tiempo medio de respuesta en segundos</b>
80	1.8
160	3.4
240	5.6
320	7.5

Como resultado se obtuvo que el módulo responde en 1.8 segundos a las peticiones de los 80 usuarios concurrentes definidos en los RNF; siendo este óptimo ya que los tiempos de respuesta deben estar por debajo de los 6 segundos establecidos en los RNF. Además, se midió el tiempo de respuesta de la aplicación para una mayor cantidad de usuarios; obteniéndose que responde adecuadamente a una demanda excesiva de hasta 240 usuarios conectados concurrentemente, con las prestaciones de hardware mínimas establecidas en los RNF. Para obtener tiempos de respuesta adecuados para más de 240 usuarios es necesario mejorar las prestaciones de hardware.

Se realizaron tres iteraciones de pruebas beta arrojando un total de nueve NC. En la primera iteración se detectaron siete NC, de ellas una de funcionalidad, una de acceso a datos, tres de validación y dos de ortografía. En la segunda iteración se detectaron tres NC, de ellas una de validación y dos de ortografía. En la revisión realizada en la tercera iteración se verificó que las NC de las etapas anteriores estuvieran

## Capítulo 3 Implementación y pruebas del Módulo de gestión de activos fijos, medios y útiles

resueltas y no se detectaron nuevas NC (ver Figura 13). La ejecución de las pruebas de aceptación culminó con la emisión, por parte del cliente, de la carta de aceptación que se muestra en los Anexos.

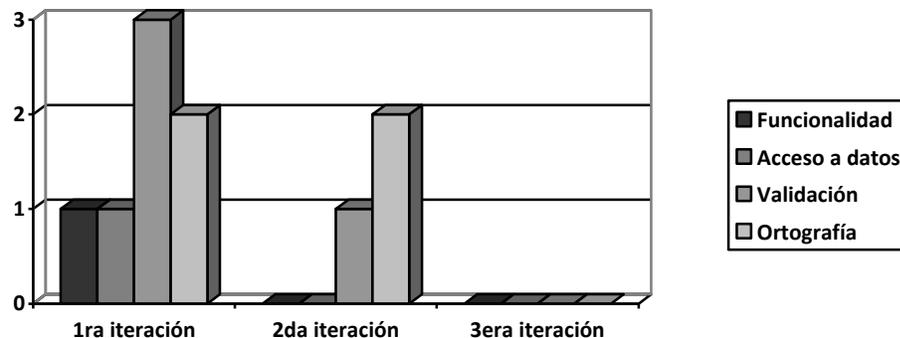


Fig 13. Gráfica de iteraciones de pruebas funcionales a nivel de integración

### 3.3 Conclusiones del capítulo.

Con el desarrollo de este capítulo se ha completado el desarrollo del Módulo de gestión de activos fijos, medios y útiles para el Sistema de Gestión de Expedientes de Control del Capital Humano de la Dirección de Gestión Tecnológica de la Universidad de las Ciencias Informáticas. La programación de los componentes del módulo de acuerdo a los requisitos establecidos en la fase de análisis y según los requerimientos establecidos en el diseño se ha finalizado totalmente. Durante la realización del sistema se ejecutaron tres ciclos de pruebas que arrojaron diversas no conformidades que fueron erradicadas totalmente en la tercera iteración. Las pruebas del sistema permitieron garantizar que se completaron el 100% de los requisitos establecidos, cumpliéndose los objetivos planteados para la investigación.

## **Conclusiones generales**

Una vez culminada la investigación se puede afirmar que se dio cumplimiento a los objetivos planteados, arribando a las siguientes conclusiones:

- El estudio de los principales conceptos relacionados con la gestión de activos fijos, medios y útiles permitió sentar las bases para el desarrollo de la solución propuesta al problema de investigación planteado.
- La realización de un Módulo para la gestión de activos fijos, medios y útiles para el sistema SIGEX, permitió utilizar las funcionalidades básicas de esta herramienta y extenderlas para garantizar la automatización de varias actividades de gestión en la DGT.
- A partir de la realización del análisis y diseño del Módulo de gestión de activos fijos, medios y útiles para la herramienta SIGEX, se obtuvo como resultado los diagramas y artefactos necesarios para guiar su desarrollo. Sirviendo estos también como material de referencia para futuras ampliaciones o modificaciones del sistema.
- La implementación del módulo propuesto dio cumplimiento a los 54 requisitos funcionales identificados en las fases de análisis y diseño.
- El diseño y ejecución de las pruebas a nivel de integración y aceptación arrojó No Conformidades las cuales fueron corregidas asegurando el correcto funcionamiento del Módulo de gestión de activos fijos, medios y útiles.
- La herramienta obtenida contribuye a la gestión del Capital Humano de la Dirección de Gestión Tecnológica en la Universidad de las Ciencias Informáticas (UCI). Mediante esta y desde un ambiente de Capital Humano se logra realizar la gestión de asignaciones y responsabilidades materiales sobre los medios de la DGT.

## **Recomendaciones**

1. Incluir al sistema un mecanismo de integración con terceras herramientas de gestión utilizadas en la Universidad para mejorar la automatización de tareas y reducir la posibilidad de errores humanos.
2. Incluir nuevos módulos a la herramienta SIGEX que permita extender sus funcionalidades a otras actividades realizadas dentro de la DGT como la formación y capacitación de los trabajadores.

## Referencias bibliográficas

1. **Addison Wesley object technology series. 2004.** *Software development for small team: A RUP - centric approach.* s.l. : Addison Wesley, 2004. ISBN: 978-0321-199-50-8.
2. **Ananraya, K y Ammarapala, V. 2010.** *The development of highway assets management system.* s.l. : IEEE Conference Publications, 2010. 17th International Conference on Service System and Service Management.
3. **Andrew , Ang. 2014.** *Asset Management: A Systematic Approach to Factor Investing.* s.l. : ISBN-13: 978-0199959327, 2014.
4. **Arregoces, Mauricio y Portolani, Maurizio. 2011.** *Data Center Fundamentals.* Indiana : Cisco Press, 2011. ISBN: 1-58705-023-4.
5. **ASF. 2015.** Apache Software Foundation. [En línea] Apache Software Foundations. [Citado el: 22 de Noviembre de 2015.] <http://www.apache.org/>.
6. **AX Consulting; SAP. 2003.** AXalpha Consulting The best-run business run. *AXalpha Consulting The best-run business run.* [En línea] AX Consulting, 2003. [Citado el: 20 de Noviembre de 2015.] [www.axconsulting.com/activos\\_fijos.html](http://www.axconsulting.com/activos_fijos.html).
7. **Bosh, Jan. 2000.** *Design and Use of Software Architectures.* London: Addison Wesley, 2000.
8. **Brooks, Frederick Phillips. 2011.** *Prácticas de Software.* [En línea] 2011. [Citado el: 11 de Agosto de 2014.] <http://www.practicadesoftware.com.ar>.
9. **Calcifer. 2012.** Guía de programación de GNOME. [En línea] 2012. [Citado el: 24 de 04 de 2014.] <http://calcifer.org/documentos/programming-guidelines/code-style.html>.
10. **Clements, P, Bass, L y Kazman, R. 2003.** *Software Architecture in Practice.* s.l.: SEI Series in Software Engineering: Addison Wesley, 2003.
11. **D´MARCO S.A.** Assets, Sistema de gestión integral. [En línea] INFOMASTER. [Citado el: 21 de Noviembre de 2015.] <http://www.assets.co.cu/>.
12. **De los Ángeles, O. 2012.** *Marcos de Trabajo para el desarrollo de Aplicaciones Web.* 2012.
13. **Domínguez, M. Dorado. 2005.** *NetBeans IDE 4.1, la alternativa a Eclipse.* Madrid : Editorial Iberprensa, 2005. págs. 32-34.
14. **Duggan, Evan y Reichgelt, Han. 2012.** *Measuring Information Systems Delivery Quality.* Hershey. s.l. : Idea Group Inc., 2012. ISBN 1-59140-859-8.

15. **Ext JS Inc. 2015.** Ext JS. [En línea] [Citado el: 11 de noviembre de 2015.] [www.extjs.com](http://www.extjs.com).
16. **Fowler, Martin, Rice, David y Foemmel, Matthew. 2002.** *Patterns of Enterprise Application Architecture*. s.l. : Addison Wesley, 2002. ISBN : 0-321-12742-0 .
17. **Fowler. 2007.** *UML distilled, A brief guide to standard object modeling lenguaje*. Tercera. s.l. : Pearson Education, 2007. ISBN: 978-8131-715-65-9.
18. **Gerner, Jason. 2011.** *Professional LAMP: Linux, Apache, MySQL and PHP 5 Web Development*. Indianapolis : Wiley Publishing, 2011. ISBN: 978-0-7645-9723-7.
19. **IEEE. 1993.** *IEEE Recommended Practice for Software Requirements Specifications*. s.l. : Software Engineering Standards Committee of the IEEE Computer Society, 1993. ISBN: 1-55937-395-4.
20. **Jacobson, Ivar, Booch, G. y Rumbaugh, J. 2000.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación, 2000. ISBN: 84-7829-036-2.
21. **Juristo, Natalia, Moreno, Ana M. y Vegas, Sira. 2005.** *Técnicas de evaluación de software. v12.0. 2005*.
22. **Krcuten, P. 2004.** *The Rational Unified Process, An Intruduction*. s.l. : Addison Wesly, 2004. ISBN: 978-032-1197-70-2.
23. **Larman, Craig. 2003.** *UML y Patrones. 2da Edición*. s.l. : Prentice Hall, Prentice Hall, 2003.
24. **Liu, Jun y Miao, Fend. 2013.** *The desing of assets management service platform in universities based on computer cloud model*. s.l. : IEEE Conferecen Publications, 2013. 2013 International Conference on Mechatronic Sciences, Electrical Engineering and Computer.
25. **McConell, S. 2012.** *Professional Software Development*. s.l. : Addison Wesley, 2012. ISBN : 0-321-19367-9.
26. **Merley, Jimi. 2012.** *¿Por qué elegir PHP?* 2012.
27. **Meyer, B. 2009.** *Construcción de Software Orientado a Objetos*. La Habana : Félix Varela, 2009.
28. **Olivo de Latouche, Marfa y Maldonado, Ricardo. 1989.** *Estudio de la Contabilidad General*. Venezuela : Valencia, Tatum, 1989.
29. **OMG. 2001.** *Unified Modeling Language Especification*. s.l. : Pearson Education,, 2001.
30. **Ortega Paredes, José Gabriel. 2012.** *Contabilidad. Paradigma de reconstrucción a través del giro informático* . Madrid,España : EAE, 2012.

31. **Planche El, Odalis y León Domínguez, Bernardo. 2004.** *Manual "Temas de contabilidad "*. La Habana : s.n., 2004.
32. **PostgreSQL Development Group. 2009.** *PostgreSQL 8.4.0 Documentation*. s.l. : PostgreSQL Global Development Group, 2009.
33. **Potencier, Fabien y Zaninotto, François. 2008.** *Symfony, la guía definitiva*. EE.UU : libroweb, 2008.
34. **Potencier, Fabien. 2011.** [www.symfony-project.org](http://www.symfony-project.org). [En línea] Symfony Project, 20 de Noviembre de 2011.
35. **Pressman, Roger. 2008.** *Ingeniería de Software: un enfoque práctico*. s.l. : Mc Graw Hill, 2008. ISBN: 970-10-5473-3.
36. **Pruebas de Software. 2016.** [En línea] [Citado el: 6 de mayo de 2016.] <http://www.pruebasdesoftware.com/pruebadeaceptacion.htm>.
37. **Quesada, Antonio. 2009.** *Pruebas de Software*. 2009.
38. **Rodríguez Tello, Eduardo A. 2012.** *Estrategias y técnicas de pruebas del software*. Tamaulipas: CINVESTAV, 2012.
39. **Rumbaugh, J, Jacobson, I y Booch, G. 1999.** *UML Reference Manual*. 1999.
40. **Sommerville, I. 2005.** *Ingeniería del Software*. 2005.
41. **Sosa Porteiro, Msc. Marisel y Cobo Morales, Lic. Pedro. 2010.** *El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero*. La Habana : s.n., 2010.
42. **Taktik Consulting. 2010.** *Taktik Consulting SAP Business One*. [En línea] Taktik Consulting, 2010. [Citado el: 5 de Febrero de 2016.] <http://www.taktik.com.mx/>.
43. **Torres Tovar, Juan Carlos. 1978.** *Introducción a la Contabilidad I*. 1978.
44. **Torres Tovar, Juan Carlos. 1978.** *Introducción a la Contabilidad I*. 1978.
45. **Visual Paradim Group. 2011.** *Reasons to Choose Visual Paradigm*. s.l. : Visual Paradim Group, 2011.
46. **Vladivimirui, A. 2013.** *Desing and development of school assets management system*. s.l. : IEEE Computer Publications, 2013. 36th International Convention on Information & Communication Technology.
47. **W3C. 2015.** [En línea] [Citado el: 15 de 2 de 2015.] <http://www.w3.org/TR/websockets/>.

## *Referencias bibliográficas*

48. **Wallace, Doug y Ragget, Isobel. 2002.** *Extreme Programming for Web Projects*. s.l. : Addison Wesley, 2002. ISBN : 0-201-79427-6.
49. **WebDesignish.** WebDesignish. [En línea] WebDesignish. [Citado el: 21 de 11 de 2015.] <http://www.webdesignish.com/>.

Anexo I. Acta de aceptación del cliente.



**Dirección de Gestión Tecnológica**

**Acta de aceptación**

La Dirección de Gestión Tecnológica (DGT) como cliente, luego de haber revisado el producto determina que el Módulo de gestión de activos fijos, medios y útiles para el Sistema de gestión de expedientes de control de Capital Humano, cumple con los requisitos preestablecidos ya que, abarca el flujo de procesos necesarios para realizar una adecuada gestión de los medios asignados a los trabajadores y las solicitudes de entrega de estos. Además, aumentó el grado de automatización del flujo de información operado dentro de la DGT y permite el almacenamiento digital de la información para su posterior análisis. Cuenta con un entorno de fácil uso, posee funcionalidades que le permiten al usuario interactuar fácilmente con el sistema y obtener en menor tiempo resultados que le permitan tomar pertinentes decisiones. Se garantiza la seguridad mediante la gestión de roles y usuarios lo que permite tener un mejor control de la información, garantizando que solo el personal autorizado tenga acceso a esta.

<b>Entrega:</b> Dirección de Gestión Tecnológica	<b>Recibe:</b> Dirección de Gestión Tecnológica
<b>Nombre y apellidos:</b> Yaima Hernández de la Oliva	<b>Nombre y apellidos:</b> José Antonio Castaño Guevara <b>Cargo:</b> Director de Gestión Tecnológica
<b>Cargo:</b> Técnica en Ciencias Informáticas	<b>Firma:</b>
<b>Firma:</b>	<b>Nombre y apellidos:</b> Nayadmi Acosta Piloto <b>Cargo:</b> Especialista General
	<b>Firma:</b>

Anexo II. Tabla de especificación de caso de uso Gestionar útil.

<b>Caso de uso</b>	Gestionar útil	
<b>Objetivo</b>	Listar, adicionar, modificar o eliminar útiles en el sistema.	
<b>Actores</b>	Técnico general: Responsable de medio	
<b>Resumen</b>	El Responsable de medio lista, adiciona, modifica o elimina activos fijos del sistema.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El usuario se ha autenticado. El usuario tiene el rol de Responsable de medio.	
<b>Postcondiciones</b>	Se gestiona uno o más útiles del sistema.	
<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar útil</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Accede a la funcionalidad Útiles de las operaciones del árbol de funcionalidades	
2.		Muestra un árbol de los útiles, así como sus componentes internos con un formulario con los detalles de cada útil que se seleccione. En caso de no existir útiles ocurre el evento No 1 del flujo alternativo.  El usuario podrá <ul style="list-style-type: none"> <li>- Adicionar un nuevo útil (Ver sección Adicionar útil)</li> <li>- Eliminar útil (Ver sección Eliminar útil)</li> <li>- Exportar listado de útiles (Ver sección Exportar útil)</li> </ul>
<b>Flujos alternos</b>		
<b>No 1 No existen útiles</b>		
	<b>Actor</b>	<b>Sistema</b>
2.		Muestra el árbol de útiles vacío con el mensaje: "No existen útiles definidos".
<b>Sección 1: "Adicionar útil"</b>		
<b>Flujo básico Adicionar útil</b>		
	<b>Actor</b>	<b>Sistema</b>

5.	Escoge la opción adicionar útil	
6.		Muestra un formulario para el registro del útil. El formulario contiene los campos de: código, nombre, unidad de medida, cantidad, gastable y descripción.
7.	Introduce los datos en los campos del formulario	
8.		Valida los datos introducidos por el usuario. Si algunos de los datos están sin llenar se produce el flujo alternativo No. 2.  Se registran los datos del útil en el sistema. Concluye el caso de uso.
<b>Flujos alternos</b>		
<b>Nº 2 Campos sin llenar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Por cada campo vacío muestra el mensaje: "Este campo es obligatorio"
2.	Comienza el paso 2 del flujo básico	
<b>Sección 2: "Modificar útil"</b>		
<b>Flujo básico Exportar útiles</b>		
1.	Escoge la opción "Exportar" en la barra de herramientas.	Si no se puede exportar el archivo pasa al flujo de eventos No 3.  Muestra un mensaje de "Exportando" mientras genera un archivo PDF con el listado de útiles y sus responsables.
2.	El usuario escoge visualizar el PDF.	
3.		Muestra el archivo PDF en otra pestaña del navegador. Concluye el caso de uso.
<b>Flujos alternos</b>		
<b>Nº 3 No se ha podido exportar el listado</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje en rojo que incida "Ha ocurrido un error al intentar exportar el listado de útiles"
<b>Sección 3: "Eliminar útiles"</b>		
<b>Flujo básico Eliminar útil</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Escoge la opción "Eliminar" en la barra de	Si no se ha seleccionado un útil pasa al

	herramientas.	flujo de eventos No 4. Muestra el mensaje: "¿Está seguro que desea eliminar el útil? Todos los datos dependientes serán eliminados."
	Escoge la opción "Si"	
		Elimina el útil y sus dependencias del sistema. Concluye el caso de uso.
<b>Flujos alternos</b>		
<b>Nº 4 No se selecciona un útil</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje que indica que no se ha seleccionado ningún útil.
2.	Selecciona un útil del árbol e inicia nuevamente el flujo básico	
<b>Relaciones</b>	<b>CU Incluidos</b>	No aplica
	<b>CU Extendidos</b>	No aplica
<b>Requisitos funcionales</b>	<b>no</b>	RNF de Apariencia o interfaz externa, RNF de Usabilidad, RNF de Rendimiento, RNF de Portabilidad, RNF de Seguridad.
<b>Asuntos pendientes</b>		No aplica

Anexo III. Caso de prueba del caso de uso Solicitar asignación de activo fijo

Sección de prueba	Escenarios a probar	Descripción
<b>Adicionar una solicitud</b>	Datos de correctos de la solicitud	Se introducen correctamente todos los datos de la solicitud
	Campos vacíos	Se deja sin llenar el campo de Activo Fijo
<b>Aprobar solicitud</b>	Solicitud sin seleccionar	Se marca la acción Aprobar sin seleccionar una solicitud
<b>Eliminar solicitud</b>	Eliminar solicitud	Se procede a eliminar una solicitud luego de seleccionarla

## Anexo IV. Variables de prueba del caso de uso Solicitar asignación de activo fijo

No	Nombre del campo	Clasificación	Nulo	Descripción
1	Activo fijo	Lista con filtro de texto	No	Nombre del activo fijo a asignar
2	Trabajador	Lista con filtro de texto	No	Nombre del trabajador para el cual se solicita el activo fijo.
3	Lugar	Lista desplegable	No	Lugar a donde se solicita el activo fijo.

## Anexo V. Matriz de prueba del caso de uso Solicitar asignación de activo fijo

Escenario	Juego de variables	Respuesta del sistema	Resultado
<b>Datos de la solicitud correctos</b>	V,V,V	Registra una nueva solicitud en el sistema y notifica al responsable de activos fijos del grupo económico	Satisfactorio
<b>Campos vacíos</b>	N,V,V	Muestra un mensaje sobre los campos vacíos indicando que estos son obligatorios.	Satisfactorio
<b>Solicitud sin seleccionar</b>	N/A	El sistema muestra un mensaje: "Debe seleccionar una solicitud"	Satisfactorio
<b>Eliminar solicitud</b>	N/A	La solicitud queda eliminada del sistema.	Satisfactorio.

## Anexo VI. Caso de prueba caso de uso Aprobar movimiento de medio

Sección de prueba	Escenarios a probar	Descripción
<b>Aprobar movimiento</b>	Movimiento aprobado correctamente	Se escoge una solicitud de movimiento de medio y se aprueba.
<b>Eliminar solicitud de movimiento</b>	Eliminar solicitud	Se procede a eliminar una solicitud de movimiento luego de seleccionarla

## Anexo VII. Matriz de prueba caso de uso Aprobar movimiento de medio

Escenario	Juego de variables	Respuesta del sistema	Resultado
<b>Aprobar movimiento</b>	N/A	Se aprueba el movimiento. El activo cambia su lugar de asignación y se notifica al responsable del activo sobre el movimiento	Satisfactorio
<b>Eliminar solicitud de movimiento</b>	N/A	La solicitud queda eliminada del sistema.	Satisfactorio