

Universidad de las Ciencias Informáticas
Facultad de Ciencias y Tecnologías Computacionales



***Módulo para la gestión del proceso de baja de Activos Fijos
Tangibles de la Universidad de las Ciencias Informáticas***

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

Autores:

Ariadna Falcón Novelo

Tutores:

Ing. Alexander Rodríguez Mompíe

Ing. Enmanuel Azahares Reyes

La Habana, Cuba

Junio, 2017

Declaración de autoría

Declaro ser autora de la presente tesis que tiene por título: Módulo para la gestión del proceso de baja de Activos Fijos Tangibles de la Universidad de las Ciencias Informáticas. Reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Ariadna Falcón Novelo

Firma del Autor

Ing. Alexander Rodríguez Mompíe

Firma del Tutor

Ing. Emmanuel Azahares Reyes

Firma del Tutor

Datos del contacto

- **Autores**

Nombre y apellidos: Ariadna Falcón Novelo

Dirección: calle 405 #19001 e/ 190 y 192 Santiago de las Vegas, Boyeros, La Habana

E-mail: ariadnafn@uci.cu

- **Tutores**

Ing. Alexander Rodríguez Mompié (arodriguezm@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) el año 2007. Profesor Asistente. Director de la Dirección de Informatización.

Ing. Enmanuel Azahares Reyes (eazhares@uci.cu)

Graduado con título de oro de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2009. Especialista general con 7 años de experiencia con la categoría docente de Instructor. Ha trabajado en proyectos productivos en el centro CENIA de la Facultad 1. Se ha desempeñado como desarrollador en el proyecto de Fuerza de Trabajo Calificada, jefe de proyecto del proyecto Kaínos para la UJC y líder del proyecto de Portadores Energéticos. Actualmente trabaja en la Dirección de Informatización como desarrollador, atendiendo los sistemas de Personal, Organizaciones y Activos del Sistema de Gestión Universitaria.

Agradecimientos

Hoy se cumple uno de los sueños que debía realizar, el graduarme y convertirme en una profesional. Más saldo una deuda inmensa con dos seres especiales que me sería imposible hablar por separado, pues han dedicado toda su vida para darme una formación de la cual me siento orgullosa, dos seres que han sido mis pilares y ejemplo a seguir, a mis padres.

Agradezco a todas las personas que de una forma u otra han hecho posible este sueño...

A ustedes, mamá y papí, por apoyarme incondicionalmente en todas las decisiones que he tomado en mi vida.

A mi hermana y mi Sofí, la que en sus cuentos de chiquita cuatro de los siete enanitos estudiaban informática en la UCI.

A mis tíos, tías y primos, siempre con un apoyo incondicional no solo conmigo, sino con mis padres, para todos mis agradecimientos eternos.

A mi novio bello por soportarme todos los días, por estar ahí cuando más lo necesitaba, por ayudarme a enfrentar y darme sustento en cada momento difícil.

A mis vecinos, más que vecinos mi otra familia, siempre siguiendo y apoyando cada avance de mi vida y mi carrera.

A mis tutores, por guiarme, por tener tanta paciencia, por toda la ayuda prestada y el tiempo dedicado. Sin ustedes este sueño no se hubiera logrado hoy, a ustedes mis más gratos agradecimientos.

A todos mis amigos y compañeros de aula por estar siempre en los momentos buenos y malos, por apoyarme en mis locuras, por aceptar mi terquedad y por cuidarme cuando más lo necesité.

A cada uno de mis compañeros de trabajo, mencionarlos a todos sería no terminar, pues para cada uno de ellos existió una pregunta, una ayuda, una opinión, mil gracias por su tiempo y dedicación.

A cada uno de mis profesores durante toda mi carrera, cada uno diferente, pero todos especiales.

En fin, quiero agradecer a cada persona que de una forma u otra han colaborado con este logro alcanzado, muchas gracias.

Resumen

El Sistema de Gestión Universitaria es una herramienta de apoyo que facilita la informatización de varios procesos que se desarrollan en la Universidad de las Ciencias Informáticas. Está compuesta por un conjunto de sistemas que garantizan que se cumpla este objetivo, donde el Núcleo, es la base para que exista un correcto funcionamiento en las operaciones que se realizan entre ellos. El Núcleo cuenta con un módulo Activos, que permite obtener los submayores actualizados de cada área administrativa, generar las actas de responsabilidad material individual y colectiva, gestionar los movimientos temporales y definitivos y reportes en ASSETS de movimientos y bajas. Estas funciones que realiza el módulo no son suficientes para satisfacer todos los procesos que se llevan a cabo con los Activos Fijos Tangibles en la universidad. Las solicitudes de baja técnica de los Activos Fijos Tangibles son realizadas a diario por las áreas administrativas. Es un proceso que se torna engorroso y se emplea gran holgura de tiempo debido a que es efectuado manualmente en la actualidad. Por lo que se propone realizar una versión al módulo, con nuevas funcionalidades que permitan toda la gestión del proceso de baja de los Activos Fijos Tangibles y facilite generar todo los documentos asociados a dicho proceso. También controlar y monitorear el estado de las solicitudes.

Como resultado se obtuvo una solución informática que gestiona el proceso de baja de los Activos Fijos Tangibles no tecnológicos de la Universidad de las Ciencias Informáticas.

Palabras clave: Activo Fijo Tangible, Comisión de baja técnica, Baja técnica, Dictamen de comisión.

Abstract

The University Management System is a support tool that facilitates the computerization of several processes that are developed at the University of Informatics Sciences. It is composed of a set of systems that guarantee that this objective is fulfilled. In this tool, the Nuclous is the base that ensures that the operations that take place in it function correctly. The Nuclous has an ASSETS module, which allows to obtain the updated subsidiary ledger of each administrative area, to generate the individual and collective material responsibility acts, to manage the temporary and definitive movements and reports in ASSETS of movements and write-offs. These functions performed by the module are not sufficient to satisfy all the processes that are carried out with Tangible Fixed Assets at the university. The requests for write-offs of Tangible Fixed Assets are made daily by the administrative areas. It is a process that becomes awkward and consumes much time because it is done manually today. Therefore it is proposed to make a version to the module, with new functionalities that allows the entire management of the process of write-off of Tangible Fixed Assets and facilitates the generation of all the documents associated with this process.

Also, to monitor and control the status of requests.

As a result, a computer solution was obtained that manages the process of write-off of non-technological Tangible Fixed Assets of the University of Informatics Sciences.

Keywords: Tangible Fixed Assets, Write-off Commission, Write-off, Commission dictum.

Índice general

Introducción	1
Capítulo 1: Fundamentos teóricos	6
1.1. Introducción	6
1.2. Marco conceptual	6
1.3. Análisis de las soluciones existentes	7
1.3.1. MINCOM ELLIPSE 8	7
1.3.2. SAP R/3.....	7
1.3.3. Microsoft Dynamics NAV	8
1.3.4. Versat-Sarasola.....	9
1.3.5. Rodas XXI Versión 3.0.....	10
1.3.6. XAUCE AKADEMOS - Sistema de Gestión Universitaria.....	10
1.3.7. Sistema de Gestión Integral (ASSETS)	11
1.3.8. Sistema de Auditoría de Activos Fijos.....	11
1.3.9. Sistema de Conciliación de Activos Fijos.....	11
1.3.10. Resultado del análisis de las soluciones existentes	12
1.4. Metodología a utilizar para la implementación de la solución	12
1.4.1. Desarrollo Ágil con Calidad (DAC).....	12
1.5. Marco de trabajo a utilizar para el desarrollo de la solución	14
1.5.1. GUUD 2.0.....	14
1.6. Lenguajes a utilizar para la implementación de la solución	14
1.6.1. Lenguaje del lado del servidor: PHP versión 5.6.7.....	14
1.6.2. Lenguaje del lado del cliente: JavaScript versión 1.8.....	15
1.6.3. Lenguaje de marcas de hipertexto: HTML versión 4	15
1.6.4. Hojas de estilo en cascada: CSS versión 3.....	15
1.6.5. Lenguaje de Consultas Estructurado: SQL versión 2012	16
1.6.6. Modelo y Notación de Proceso de Negocio: BPMN versión 2.0.....	16
1.6.7. Lenguaje Unificado de Modelado: UML versión 2.0	16
1.6.8. Lenguaje de Marcas Extensibles: XML versión 1.0.....	17
1.7. Herramientas a utilizar para el desarrollo de la solución	17
1.7.1. Entorno de Desarrollo Integrado: NetBeans versión 8.0	18
1.7.2. Administrador de base de datos: PgAdmin III versión 1.14.0.....	18
1.7.3. Herramienta para prototipos: Evolus Pencil versión 2.0.5.....	18
1.7.4. Herramienta de modelado: Visual Paradigm para UML versión 8.0.....	19
1.7.5. Sistema gestor de base de datos: PostgreSQL versión 9.4.1	19
1.7.6. Servidor web: Apache versión 2.4.7.....	19
1.7.7. Herramienta de pruebas: Apache JMeter versión 2.3.1	20
1.8. Conclusiones parciales	20
Capítulo 2: Concepción de la propuesta de solución	21
2.1. Modelado del proceso de negocio	21
2.1.1. Descripción del proceso de negocio	21
2.1.2. Reglas del negocio	25
2.2. Especificación de requisitos de software.....	27
2.2.1. Técnicas de obtención de requisitos.....	27
2.2.2. Definición de Requisitos funcionales.....	28
2.2.3. Definición de Requisitos no Funcionales	30

2.2.4. Descripción de requisitos.....	32
2.2.5. Validación de los requisitos	35
2.3. Descripción de la propuesta de solución.....	36
2.4. Descripción de la arquitectura.....	38
2.4.1. Estilo arquitectónico.....	39
2.4.2. Patrón de arquitectura	39
2.4.3. Patrones de diseño.....	41
2.4.4. Diagramación de las vistas	44
2.4.5. Sistema de mensajes de la propuesta	46
2.4.6. Plantillas de documentos generados	47
2.4.7. Patrones de base de datos.....	48
2.4.8. Modelo de datos.....	50
2.4.9. Modelo de despliegue.....	51
2.5. Conclusiones parciales	52
Capítulo 3. Construcción y validación de la propuesta de solución.....	53
3.1 Estándares de codificación.....	53
3.1.1 Indentación, llaves de apertura y cierre y tamaño de las líneas	53
3.1.2 Conversión de nomenclatura	54
3.1.3 Estructura de control.....	55
3.1.4 Documentación.....	55
3.2 Técnicas de programación	56
3.3 Integración con otros módulos y sistemas	57
3.5 Estrategia de pruebas	58
3.5.1 Prueba de unidad.....	59
3.5.2 Prueba de integración.....	60
3.5.3 Pruebas de sistema	61
3.5.4 Pruebas de rendimiento.....	66
3.6 Conclusiones parciales.....	67
Conclusiones	68
Recomendaciones	69
Referencias bibliográficas	70

Índice de figuras

Figura 1. Modelo del proceso DAC.	13
Figura 2. Diagrama de proceso de negocio de Aprobación de baja de AFT.	24
Figura 3. Mapa de navegación.	38
Figura 4. Representación de la arquitectura Cliente-Servidor.	39
Figura 5. Patrón Modelo-Vista-Controlador en GUUD.	41
Figura 6. Áreas de la vista de presentación.	44
Figura 7. Áreas de la vista de escritorio.	45
Figura 8. Áreas de la vista de gestión de procesos.	45
Figura 9. Mensaje de confirmación en ventana emergente.	46
Figura 10. Mensaje de información en ventana emergente.	46
Figura 11. Mensaje incrustado en formularios.	46
Figura 12. Plantilla estandarizada de Solicitud de baja.	47
Figura 13. Documento de Solicitud de baja no estandarizado.	47
Figura 14. Plantilla estandarizada de Dictamen técnico.	48
Figura 15. Documento de Dictamen técnico no estandarizado.	48
Figura 16. Patrón de diseño de base de datos árbol fuertemente codificado.	49
Figura 17. Patrón de diseño de base de datos descripción de elemento.	49
Figura 18. Modelo físico de datos.	51
Figura 19. Representación del modelo de despliegue.	52
Figura 20. Indentación, llaves de apertura y cierre y tamaño de las líneas.	53
Figura 21. Conversión de nomenclatura de variables.	54
Figura 22. Conversión de nomenclatura de clases.	54
Figura 23. Conversión de nomenclatura de funciones.	54
Figura 24. Conversión de nomenclatura de estructuras de control.	55
Figura 25. Documentación de clases.	56
Figura 26. Documentación de funciones.	56

Índice de tablas

Tabla 1. Listado de documentos generados en Implementación de baja de AFT.....	24
Tabla 2. Listado de Requisitos funcionales.	28
Tabla 3. Listado de Requisitos no Funcionales.	30
Tabla 4. Descripción de Requisito funcional.....	32
Tabla 5. Estrategia de pruebas de software.	59
Tabla 6. Caso de prueba unitaria de la funcionalidad Crear solicitud primaria de baja AFT.	59
Tabla 7. Caso de prueba de integración con el módulo Estructura y Composición.	61
Tabla 8. Diseño de caso de prueba del requisito funcional Listar comisiones. Parte 1.	62
Tabla 9. Diseño de caso de prueba del requisito funcional Listar comisiones. Parte 2.	63
Tabla 10. Resultados obtenidos de las pruebas de rendimiento.	66

Introducción

Cuba hoy en día se encuentra inmersa en convertir las Tecnologías de la Información y Comunicación (TICs) en un sector de desarrollo estratégico para el país [1]. Las bases de la política para el perfeccionamiento de la informatización segura del país tienen entre sus prioridades nacionales la mejora de los procesos de gestión para el control de los recursos materiales y financieros. Las herramientas utilizadas en estos procesos permiten fortalecer la economía, al facilitar la gestión y asegurar la exactitud de los registros de la información económica y contable [2].

El sistema empresarial cubano no está libre de los problemas que afectan la economía, razón que progresivamente obliga al país a emplear sistemas informáticos de gestión empresarial que entre sus procesos beneficie al Sistema de Control Interno establecido en la Resolución 60 del 1ero de marzo del 2011 de la Contraloría General de la República [3]. El Sistema de Control Interno implanta como norma que el 10% de los Activos Fijos Tangibles (AFT¹) sean auditados cada mes. Tarea crucial en cada entidad, donde se verifica la exactitud del registro en los libros contables contra los medios existentes y confirma su coincidencia. De esta forma se estrecha la posibilidad de hurto, despilfarro, uso indebido u otras irregularidades. Actividad que ha de ejecutarse con un carácter exhaustivo pero que se les dificultan a las entidades que realizan el control de forma manual y presentan una gran cantidad de AFT.

El Ministerio de Educación Superior como órgano rector establece que todas sus instituciones subordinadas utilicen el sistema informático ASSETS para la gestión de los procesos contables, financieros y de recursos humanos. Dentro de los procesos contables se encuentra el control de los Activos Fijos Tangibles y útiles. El sistema ASSETS es comercializado por la firma panameña D`MARCO S.A y distribuido por la empresa cubana INFOMASTER, entidad perteneciente a la Empresa Nacional de Producción y Servicios a la Educación Superior del Ministerio de Educación Superior. Dicho sistema es una solución de escritorio, su licencia se paga por módulos y por cantidad de usuarios que utilicen el mismo.

La Universidad de las Ciencias Informáticas (UCI) tiene un contrato para su utilización, por el cual paga un monto elevado anualmente. El sistema solo es utilizado en la Dirección de Contabilidad y Finanzas por cada trabajador para el cumplimiento de sus funciones. Pero en la universidad se evidencia una amplia estructura organizacional la cual está compuesta por el Rectorado, Vicerrectorías, Facultades, Secretaría General, Direcciones y Centros de desarrollo. Todos los directivos de estas áreas administrativas

¹ **AFT**: Activos Fijos Tangibles, representan propiedades físicamente tangibles que han de utilizarse por un período largo en las operaciones regulares de la entidad y que normalmente no se destinan a la venta.

necesitan tener acceso a la información real de los diferentes AFT asociados a los centros de costo² que tiene definido la universidad. Esto requeriría instalar el sistema de escritorio en cada uno de sus puestos de trabajo, además de pagar la licencia por cada nuevo usuario. Acción que encarecería el pago realizado a la empresa INFOMASTER por concepto de licencia.

Uno de los trámites que se lleva en el control de los AFT en la universidad por parte de los directivos de las áreas administrativas, es el proceso de baja técnica para aquellos activos que han finalizado su vida útil. Proceso engorroso que conlleva conformar un expediente de baja técnica, que documente oficialmente la ejecución del mismo. Dichos documentos son elaborados manualmente, imprimiendo para ello una plantilla previamente. Estas plantillas pueden ser elaboradas por diferentes personas, incurriendo en una diferencia de información y formatos y dificultando el trabajo de evaluación de los medios por parte de las comisiones que dictaminan los Activos Fijos Tangibles a los que se les solicita la baja técnica.

Con este estilo de trabajo, la consolidación de la información para la obtención de reportes se ve limitada debido a que todo el proceso de baja técnica se documenta solo en formato duro, y para lograr un reporte se debe consultar la información en cada expediente de baja.

El sistema ASSETS, como se enunció anteriormente, incluye entre sus funcionalidades el registro de baja para los AFT, pero no soporta el proceso previo a este registro. Por tanto, no cuenta con la gestión de los documentos, expedientes y reportes que la Dirección de la universidad necesita como parte de la documentación del proceso. Todo lo antes expuesto evidencia la necesidad de solicitar al proveedor del producto una personalización del sistema. De esta forma la universidad incurriría en un gasto adicional del ya abonado por concepto de soporte y accesos.

Debido a lo antes expuesto, se evidencia la necesidad del desarrollo de una solución informática capaz de soportar el proceso de gestión de baja de los activos fuera del ASSETS, así como dar cumplimiento a las necesidades vigentes que no satisface este sistema, realizando un análisis previo para identificar las necesidades reales de los clientes y lograr un eficaz desarrollo de la solución.

A partir de la situación problemática descrita se plantea el siguiente **problema de investigación**:

¿Cómo mejorar la gestión del proceso de baja de Activos Fijos Tangibles en las áreas administrativas de la Universidad de las Ciencias Informáticas?

² **Centro de costos:** es una división que genera costos para la organización, pero sólo indirectamente le añaden beneficio o utilidad.

Para dar solución al problema en cuestión se traza como **objeto de estudio** el proceso de baja de los Activos Fijos Tangibles, centrando su **campo de acción** en el proceso de baja de los Activos Fijos Tangibles en la Universidad de las Ciencias Informáticas.

En consecuencia, se define como **objetivo general** desarrollar un módulo integrado al Sistema de Gestión Universitaria – Akademos de la Universidad de las Ciencias Informáticas, para mejorar el control de la información en el proceso de baja de los Activos Fijos Tangibles en sus áreas administrativas.

Para guiar la investigación se plantea la siguiente **idea a defender**: la inclusión del módulo en el Sistema de Gestión Universitaria – Akademos para la gestión del proceso de baja de Activos Fijos Tangibles, contribuirá a la mejora de dicho proceso en las áreas administrativas de la Universidad de las Ciencias Informáticas.

Para la realización del presente trabajo se definen las siguientes **tareas de investigación**:

1. Desarrollo del diseño teórico-metodológico de los sistemas informáticos de gestión de baja de Activos Fijos Tangibles.
2. Elaboración de los fundamentos teóricos en el desarrollo del módulo para la gestión del proceso de baja de Activos Fijos Tangibles.
3. Fundamentación del uso de la metodología, tecnologías, lenguajes y herramientas a utilizar en el desarrollo de la solución.
4. Identificación de las funcionalidades del módulo para la gestión del proceso de baja de Activos Fijos Tangibles.
5. Descripción de las funcionalidades del módulo para la gestión del proceso de baja de Activos Fijos Tangibles.
6. Implementación de las funcionalidades del módulo para la gestión del proceso de baja de Activos Fijos Tangibles.
7. Realización de pruebas al módulo para la gestión del proceso de baja de Activos Fijos Tangibles, basado en los diseños de casos de prueba elaborados.

Con el desarrollo de la investigación se persigue como **resultado**:

1. Desarrollar un módulo dentro del Sistema de Gestión Universitaria – Akademos para la gestión del proceso de baja de Activos Fijos Tangibles.
2. Contribuir con la mejora de la gestión del proceso y centralización de la información relacionado con el proceso de baja de los Activos Fijos Tangibles en las áreas administrativas de la UCI.

3. Rediseñar las plantillas de los documentos que se generan como parte del proceso de baja, ajustadas al trabajo de estandarización de la papelería administrativa en la que se encuentra enfocada la universidad.
4. Obtener reportes relacionados con el proceso de baja de Activos Fijos Tangibles ajustados a las necesidades de información de las áreas administrativas de la universidad.

Métodos de investigación

Los **métodos científicos de investigación** utilizados para la realización de la investigación son los siguientes:

- **Métodos teóricos**
 - **Analítico-sintético:** permitirá el análisis de teorías y documentos referentes a los conceptos, herramientas, lenguajes, tecnologías y metodología asociados a la investigación, extrayendo los elementos más importantes y necesarios para el apoyo en el desarrollo de la solución.
 - **Histórico-lógico:** se aplicará en el estudio de las herramientas existentes de los procesos de control de Activos Fijos Tangibles, con el objetivo de obtener características de su funcionamiento y determinar aspectos de utilidad para dar solución al problema existente.
 - **Modelación:** se utilizará para crear diagramas y modelos con el objetivo de explicar el funcionamiento de los procesos de negocio en el módulo para la gestión del proceso de baja de Activos Fijos Tangibles.
- **Métodos empíricos**
 - **Entrevista:** se utilizará para lograr un mejor entendimiento sobre las características y distintas etapas del proceso de gestión baja de los Activos Fijos Tangibles en las áreas administrativas de la Universidad de las Ciencias Informáticas. Sus diseños están accesibles en el Anexo 1.

El documento está estructurado por los siguientes capítulos:

Capítulo1. Fundamentos teóricos: incluye el estudio del estado del arte del tema tratado, donde se exponen los principales conceptos empleados en el transcurso de la investigación. Se realiza la descripción y análisis de los sistemas informáticos de gestión de Activos Fijos Tangibles existentes y una caracterización de las herramientas, técnicas, lenguajes, metodología y el proceso de desarrollo a utilizar en la construcción de la propuesta de solución.

Capítulo 2. Concepción de la propuesta de solución: describe la propuesta de solución enmarcada en el proceso actual de gestión de baja de los Activos Fijos Tangibles en las áreas administrativas de la universidad donde permita construir el diagrama del proceso de negocio, identificar las reglas del negocio, y definir mediante las técnicas de obtención de requisitos, los requisitos funcionales y no funcionales. Además de describir la arquitectura, patrones de diseño y de base de datos, que se utilizan en la investigación.

Capítulo 3. Construcción y validación de la propuesta de solución: detalla la construcción de la solución propuesta. Se efectúa el proceso de evaluación del producto desarrollado especificando las evaluaciones y validaciones realizadas antes, durante, y después del proceso de creación del código.

Capítulo 1: Fundamentos teóricos

1.1. Introducción

En el presente capítulo se definen los principales conceptos asociados a la gestión de activos fijos, así como el estudio a diversos sistemas informáticos referentes al tema, con el objetivo de estudiar sus funcionalidades y características dando lugar a una visión general del estado del arte. Para guiar la implementación de la propuesta de solución, se caracterizan las tecnologías y la metodología de desarrollo de software a utilizar, definidas por el Departamento de Desarrollo de la Dirección de Informatización (DIN).

1.2. Marco conceptual

Como base para el conocimiento de diferentes aspectos a tratar en la investigación, se hace necesario abordar un conjunto de conceptos principales relacionados con el dominio del problema.

Activo Fijo Tangible

Constituyen los bienes materiales que una entidad utiliza de manera continua en sus operaciones, con un carácter permanente, los cuales sufren una depreciación durante su vida útil que afectará al final el nivel de utilidades [4]. Representan propiedades físicamente tangibles que han de utilizarse por un período largo en las operaciones regulares de la entidad y que normalmente no se destinan a la venta.

Recursos a favor de la entidad, representados por todos aquellos bienes y derechos adquiridos con el propósito de utilizarlos, para la realización de sus operaciones; es decir se mantienen en la empresa con el propósito de ser usados y no como objeto de su comercialización, estos activos tienen cierta permanencia en la empresa y solo se enajenan cuando dejan de ser útiles [5].

Comisión de baja técnica

Conjunto de personas elegidas a nivel superior y designadas por la entidad mediante el respaldo de una Resolución, con la responsabilidad integral de dar curso a la gestión del proceso de baja de los Activos Fijos Tangibles de la Universidad de las Ciencias Informáticas.

Baja técnica

Es aquella que se produce cuando un activo fijo ha llegado al término de su vida útil, en los casos en que no posee un estado técnico aceptable para continuar su explotación, causados por roturas definitivas, destrucción o en aquellos que se consideren inservibles, obsoletos o irreparables [6].

Dictamen de comisión

Opinión y juicio que se forma y emite la comisión una vez verificado el estado del activo fijo, dando paso a la confirmación o no del proceso de baja técnica.

1.3. Análisis de las soluciones existentes

Hoy en día las empresas utilizan como buena práctica sistemas informáticos para la optimización de sus procesos. En el ámbito nacional e internacional existe gran variedad de programas encargados de llevar el control de los Activos Fijos Tangibles. Es objetivo de la investigación realizar el estudio a sistemas de características similares a las necesidades planteadas en el problema a resolver, que sirvan de apoyo al desarrollo de la investigación que se desea implementar.

1.3.1. MINCOM ELLIPSE 8

Es un sistema móvil de gestión de activos empresariales y de planificación de recursos empresariales. Permite a las empresas planificar, medir, monitorear y mejorar el rendimiento de activos durante todo el ciclo de vida de un activo [7]. Programa informático que presenta una arquitectura abierta y posibilita la participación más amplia en la gestión de activos de todas las funciones en la empresa. Admite la integración completa de solución y proceso para respaldar las mejores prácticas en la gestión de activos.

Esta plataforma móvil integrada proporciona la aplicación para automatización del proceso, análisis de activos en tiempo real, planificación de trabajo a corto y largo plazo, junto con la administración de pedidos de trabajo.

1.3.2. SAP R/3

Es un sistema de tipo ERP (*Enterprise Resource Planning*)³ de origen alemán, que ha sido diseñado en base a una plataforma que ofrece una gran versatilidad de programación, facilidad de uso, y precisión en el manejo total de los datos recolectados [8]. En este sistema de gestión es posible realizar un control exhaustivo de todos los procesos y operaciones que se realizan dentro de una compañía, almacenando la información que puede ser utilizada por cualquier sector de la organización.

Es una solución estándar, que presenta módulos preprogramados con procesos básicos de cada empresa. Generalmente estos módulos y los procesos básicos deben ser adaptados para cada compañía. Los módulos más populares son [9]:

- Contabilidad financiera (FICO)
- Recursos humanos o gestión de personal (HR)
- Gestión de materiales (MM)
- Ventas y distribución (SD)
- Gestión de producción (PP)

³ **ERP**: siglas en inglés de *Enterprise Resource Planning*, Planificación de Recursos Empresariales; son sistemas informáticos destinados a la administración de recursos en una organización.

Las funcionalidades del módulo de Gestión de Activos sistema SAP /R3, permiten al usuario la contabilización de todos los datos y movimientos contables relacionados con los activos fijos, donde se encuentran:

ACTIVOS FIJOS – DATOS MAESTROS.

- Crear Activo Fijo.
- Modificar Activo Fijo.
- Visualizar Activo Fijo.
- Bloquear Activo Fijo.
- Borrar Activo Fijo.

ACTIVOS FIJOS – PROCESO DE PETICIÓN DE BAJA

- Proceso de Petición de Baja.
- Solicitud Departamento

ACTIVOS FIJOS – LIQUIDAR ALTAS MASIVAS PRESUPUESTARIAS.

- Liquidar altas masivas presupuestarias-distribuir.

1.3.3. Microsoft Dynamics NAV

Es un software ERP (por sus siglas en inglés: *enterprise resource planning*). Microsoft Dynamics NAV forma parte de la familia de productos Microsoft Dynamics. Software de gestión empresarial global, que recopila todos los procesos organizativos en un sistema totalmente integrado, que ayuda a mejorar la eficiencia y proporciona un servicio diferenciado a precios competitivos. Microsoft Dynamics NAV es un ERP orientado a roles. El sistema está basando en las personas dentro de una organización, su rol, y las tareas que llevan a cabo. Esta aplicación de gestión está dotada de estrechos lazos funcionales y visuales con el resto de productos Microsoft, proporcionando una interfaz familiar [10].

Cubre diferentes áreas funcionales dentro de una organización donde se encuentra Gestión Financiera que abarca funcionalidades de: contabilidad, presupuestos, esquemas de cuentas, informes financieros, gestión de tesorería, efectos a cobrar y a pagar, activos fijos, declaraciones de IVA⁴, registros entre empresas, contabilidad de costes, consolidación y divisas.

Dentro del módulo de contabilidad de Dynamics NAV, existen varias opciones en lo referente al control de los activos fijos:

- Realizar un seguimiento de los bienes inmovilizados de la empresa (equipos, maquinaria, edificios)

⁴ **IVA**: sigla de impuesto sobre el valor añadido o de impuesto sobre el valor agregado, impuesto que grava el valor añadido o agregado de un producto en las distintas fases de su producción.

- Registro de transacciones con los activos fijos: adquisiciones, depreciaciones, amortizaciones y ventas/bajas.
- Puede asignar diferentes porcentajes de transacciones de activos fijos, como coste de adquisición y amortización, a departamentos o proyectos distintos mediante claves de asignación. Siendo útil cuando varios departamentos comparten un activo fijo.

1.3.4. Versat-Sarasola

Sistema integrado de gestión económica, diseñado para ser utilizado por el sector empresarial cubano, que se adecua a las características de cada entidad, ya que es configurable por cada una de ellas en el momento de su instalación y tiene como objetivo fundamental ofrecerles a los usuarios la posibilidad de contar con un instrumento seguro, rápido, eficaz y de fácil manejo para la Planificación, Control y el Análisis de la Gestión Económica [11].

Se estructura en un grupo de módulos, donde se procesan y contabilizan documentos primarios y se anotan los movimientos de los recursos materiales, financieros y laborales que se utilizan en una Entidad a partir de una configuración previa de los comprobantes que se originan.

Este Sistema contempla los requerimientos necesarios para lograr el control efectivo de los Activos Fijos Tangibles e intangibles a partir de las siguientes funciones:

- Control por áreas y centros de costos.
- Posibilidad de calcular y registrar diariamente la depreciación de los Activos Fijos Tangibles y la amortización de los intangibles.
- Brinda la posibilidad de calcular la depreciación por 4 métodos.
- Conceptualiza los diferentes movimientos y posibilita configurar la contabilización de los mismos.
- Realizar inventarios a partir de diferentes selecciones, determinar las posibles diferencias y contabilizar las mismas.
- Permite definir diferentes ciclos de depreciación de los Activos Fijos Tangibles teniendo en cuenta las características de la entidad.
- Permite el control de los activos en diferentes monedas.
- Ofrece variedad de reportes sobre las existencias y movimientos de los activos.

1.3.5. Rodas XXI Versión 3.0

Sistema Integral Económico Administrativo desarrollado por la empresa CITMATEL⁵ que posibilita automatizar el funcionamiento de cualquier empresa o unidad presupuestada [12].

Es un sistema multiempresa y en constante desarrollo. Actualmente cuenta con ocho módulos: finanzas, contabilidad, activos fijos, nóminas, inventario, facturación, recursos humanos y telecobranzas. Estos módulos pueden ser empleados e integrados en su totalidad, y se puede formar cualquier subconjunto entre ellos, o cada uno de forma independiente [12].

El módulo de activos fijos de RODAS XXI le permite tener un control detallado de los activos fijos de su entidad, realizando en el mismo momento que se registra un movimiento, su contabilización. Este módulo da la posibilidad de realizar la depreciación de forma automática mediante opción y no permite el cierre de períodos contables si no es realizada la depreciación del período a cerrar. El módulo cuenta con una gran variedad de opciones de informes, entre ellos podrá obtener el submayor de activos fijos, listados y localización de los medios de transporte de la entidad, la depreciación mensual y acumulada de uno o de los activos fijos que desee, el acta de responsabilidad material de cada una de las áreas y muchos otros, cubriendo con seguridad todas las necesidades de información que sobre los activos fijos la entidad requiera.

RODAS XXI ha sido diseñado para trabajar en los sistemas operativos de Microsoft: Windows XP, 2000 ó 2003, con un servidor que cuente con: sistema operativo Microsoft Windows XP, 2000 ó 2003, Microsoft SQL Server 2000, Seagate Crystal Report Web Component Server 8.0. Por otra parte los requerimientos de hardware son mínimos.

1.3.6. XAUCE AKADEMOS - Sistema de Gestión Universitaria

Sistema de gestión universitaria (SGU) desarrollado en la UCI. Aplicación web que se adecúa a las necesidades de la Universidad de las Ciencias Informáticas, siendo un sistema modular que cuenta con 12 áreas de procesos identificadas (Pregrado, Posgrado, Cooperación, Residencia, Teleformación, Biblioteca, Desarrollo, Tecnologías, Investigación, Extensión, Organizaciones y Egreso). Estas aplicaciones interactúan con un núcleo común que agrupa todos los procesos horizontales (Configuración, Seguridad, Estructura y Composición, Traza, Inmuebles, Personal, Documentos Acreditativos, Archivos y Activos y Útiles) permitiendo estandarizar las mismas para ser usadas por cada una de las áreas de procesos del Sistema de Gestión Universitaria. Apoyado en los módulos de seguridad y configuración, se permite personalizar la pantalla de entrada al sistema para el usuario autenticado, mostrándole las áreas

⁵ **CITMATEL:** empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados que desarrolla, produce y comercializa aplicaciones informáticas, proyectos, equipamiento y asistencia técnica, producciones multimedia, audiovisuales y ediciones electrónicas.

de procesos a las que el usuario tiene acceso y construyendo el escritorio del sistema, en el que se incluyen algunos elementos como identidad visual, sistema de íconos y traza en línea que potencian su usabilidad.

Integrado a un archivo universitario, donde se almacenan las evidencias documentales de cada área de proceso.

El módulo de Activos y Útiles del SGU presenta en la actualidad las siguientes funciones para el control efectivo de los mismos:

- Generación de los submayores de los centros de costo y áreas de responsabilidades asociadas.
- Generación de los rótulos de los números de inventario de los activos.
- Generación de las actas de responsabilidad individual, colectiva y por áreas.
- Gestión de los movimientos temporales y definitivos.

1.3.7. Sistema de Gestión Integral (ASSETS)

El sistema ASSETS es comercializado por la firma panameña D`MARCO S.A y distribuido por la empresa cubana INFOMASTER, entidad perteneciente a la Empresa Nacional de Producción y Servicios a la Educación Superior del Ministerio de Educación Superior, es un sistema de gestión integral estándar y parametrizado concebido para el control de la actividad económica empresarial, que es actualmente utilizado en la UCI. Es una solución de escritorio, su licencia se paga por módulos y cantidad de usuarios que utilicen el mismo.

El módulo de Activos Fijos de ASSETS permite controlar, por centros de costos, sus medios básicos. Define la ubicación física de los mismos, controla los procesos de Compras, Alquiler, Altas, Bajas, Préstamo, Traspasos hacia otras áreas dentro y fuera de la entidad, realiza reparaciones, controla ociosos, realiza ajustes y controla la depreciación acumulada de cada activo [13].

Aplicación cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000, utilizando adicionalmente *Crystal Reports 7.0* para la generación de reportes de salidas [14].

1.3.8. Sistema de Auditoría de Activos Fijos

Sistema desarrollado y concebido solo para la UCI, realiza el levantamiento de la información actual de los activos existentes en las distintas áreas de responsabilidad de la universidad según los centros de costos vigentes en la misma [15]. Donde la información es protegida por claves y ofrece algunos reportes de los AFT.

1.3.9. Sistema de Conciliación de Activos Fijos

Sistema desarrollado en la UCI capaz de generar los inventarios de AFT de forma automática y la gestión documental de los procesos que se realizan con ellos. Se implementa con el objetivo de realizar la

conciliación de los activos de la universidad con vista a las auditorías. Se desarrolló completamente con software libre y además es una plataforma web [16].

1.3.10. Resultado del análisis de las soluciones existentes

Durante el estudio de las diversas soluciones existentes que manejan información asociada a la gestión empresarial y que entre sus procesos contienen la gestión de los Activos Fijos Tangibles se evidencia que, a pesar de presentar características semejantes a las necesidades planteadas, los sistemas desarrollados y propios de la universidad solo satisfacen parte del problema de la entidad.

Las soluciones nacionales están desarrolladas sobre plataformas envejecidas y propietarias, además que son aplicaciones de escritorio, por lo que es necesario para su uso, la instalación de la misma en cada estación de trabajo. En cambio, con una aplicación web solo bastaría tener acceso a la red para acceder a ella. No todos ellos se adaptan a las características de las empresas, esto trae consigo que existan empresas con más de un sistema en uso.

Los sistemas internacionales a pesar de la generalidad de sus procesos y contar con elementos positivos para su implantación, existiría la necesidad de invertir por concepto de soporte y licenciamiento.

De forma general ninguno de los sistemas estudiados ofrece funcionalidades que gestionen y documenten un proceso de baja, solo brindan la opción de dar baja al activo en el sistema.

1.4. Metodología a utilizar para la implementación de la solución

Una metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Cada metodología de desarrollo de software presenta relativamente su propio enfoque. Para el desarrollo de la solución el Departamento de Desarrollo de la DIN de la UCI define la metodología DAC⁶.

1.4.1. Desarrollo Ágil con Calidad (DAC)

Esta metodología combina las buenas prácticas del nivel 2 de Integración de Modelos de Madurez de Capacidades (CMMI⁷ por sus siglas en inglés) y los conceptos vinculados al enfoque ágil,

⁶ **DAC**: por sus siglas Desarrollo Ágil con Calidad, metodología diseñada por el Departamento de Desarrollo de la Dirección de Informatización de la UCI.

⁷ **CMMI**: *Capability Maturity Model Integration* por sus siglas en inglés - Integración de modelos de madurez de capacidades - es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de *software*.

estableciéndose un marco de trabajo del proceso común, adaptable para los proyectos del Departamento de Desarrollo de la DIN, a partir de la integración de los modelos: lineal, desarrollo concurrente y programación extrema. Está enfocado a proyectos pequeños o proyectos grandes divididos en sub-proyectos que desarrollan *software* de gestión basado en componentes.

DAC descompone los problemas identificados en sub-problema el cual será resuelto mediante un componente, siendo la solución del problema el *software* o producto final. Las entregas en DAC son a nivel de iteración, en la que habrá obligatoriamente un incremento del producto a partir de la solución de un componente del mismo. Además, en cada iteración se define como mínimo un hito a cumplir por cada fase [17].

La Figura 1 muestra el modelo de proceso de la metodología DAC.

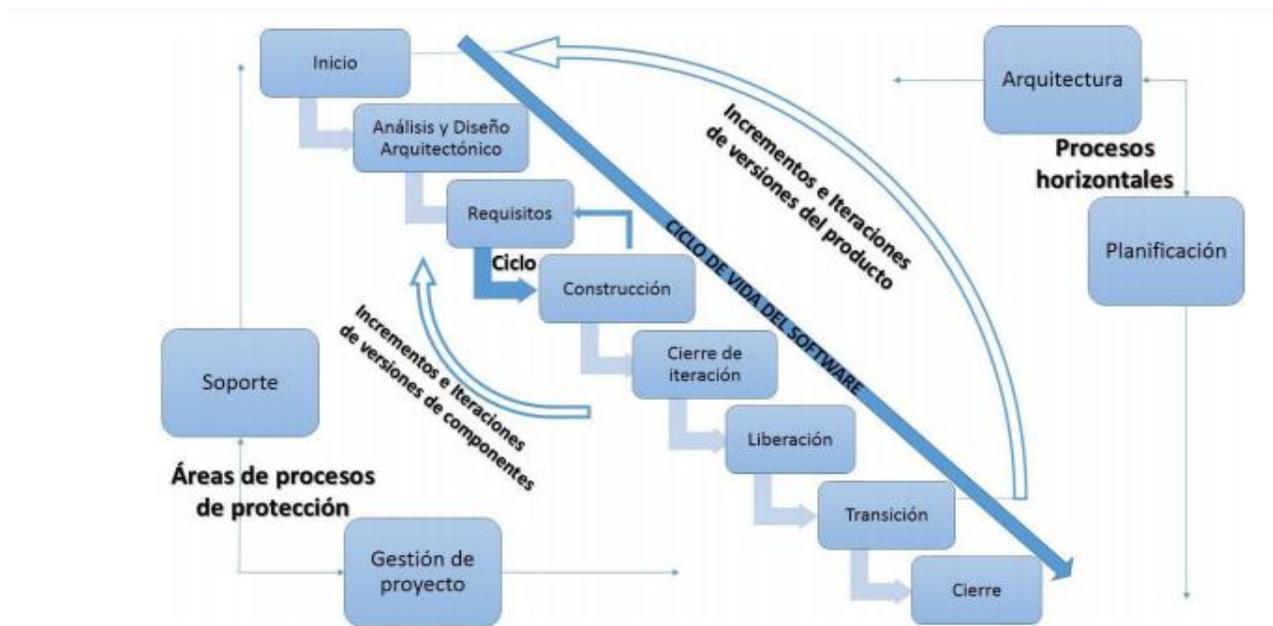


Figura 1. Modelo del proceso DAC [17].

El proceso presenta 8 fases o Procesos del ciclo de vida, el cual contempla: Inicio, Análisis y diseño arquitectónico, Requisitos, Construcción, Cierre de iteración (opcional), Liberación, Transición y Cierre. Entre el Inicio y Cierre del proyecto se realizan iteraciones por entregas de versiones del producto. Entre las fases de Desarrollo de Requisitos, Construcción del Producto y Cierre de Iteración ocurren iteraciones concurrentes a nivel de componentes. Y antes de un Cierre de Iteración también puede ocurrir un ciclo pues a medida que los requisitos son identificados y descritos pueden entrar en la fase de Construcción.

1.5. Marco de trabajo a utilizar para el desarrollo de la solución

Un marco de trabajo en el desarrollo de *software* es una estructura conceptual y tecnológica de soporte definido, que ofrece un conjunto de bondades con el que se obtiene un ágil desarrollo y mayor calidad del producto. Permite compartir en un grupo de desarrollo una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación que se desea construir.

1.5.1. GUUD 2.0

GUUD (siglas creadas con las iniciales de los departamentos del antiguo centro CENIA: Gestión Universitaria, Universidad Digital y Gestión Documental). Es definido por el equipo de arquitectura del Departamento de Desarrollo de la DIN donde se incorpora los marcos de trabajo PHP⁸ CodeIgniter en su versión 1.7.3 y la librería de JavaScript, jQuery en su versión 1.9.2 en una sola infraestructura, que permite la creación de aplicaciones web escritas en PHP. Realiza un manejo de excepciones y mensajes. Modulariza el marco de trabajo CodeIgniter e implementa como estrategia de comunicación entre módulos la Inversión de Control (IoC). La programación es orientada a aspectos, más contiene adjunto los plugins `export_pi` e `import_pi` que permite la exportación e importación de ficheros. Contiene una serie de componentes visuales que permiten la fácil interacción del usuario con la aplicación tales como el calendario y las tablas [18].

1.6. Lenguajes a utilizar para la implementación de la solución

Un lenguaje de programación está diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar, se basa en sintaxis codificadas dígame conjunto de instrucciones, órdenes, comandos, operadores y reglas de sintaxis empleadas por el programador para darle solución al proceso a través del ordenador, estableciendo una comunicación con los dispositivos de *software* y *hardware*. Existe amplia gama de lenguajes en el desarrollo de aplicaciones web, diferenciándose entre sí y utilizadas según las características específicas del producto final que se quiera alcanzar.

1.6.1. Lenguaje del lado del servidor: PHP versión 5.6.7

PHP (*Hypertext Preprocessor*), traducido al español como Preprocesador de hipertexto es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Con PHP no se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF (*Portable Document Format*, "Formato de Documento Portátil") y películas

⁸ **PHP**: es un lenguaje de programación de uso general, de código del lado del servidor, originalmente diseñado para el desarrollo web de contenido dinámico.

Flash (usando *libswf* y *Ming*) sobre la marcha. También puede presentar otros resultados, como XHTML⁹ y archivos XML¹⁰ [19]. Así mismo, puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla. Quizás la característica más potente y destacable es su soporte para una gran cantidad de base de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP.

1.6.2. Lenguaje del lado del cliente: JavaScript versión 1.8

JavaScript es un lenguaje interpretado, es decir, no requiere compilación [20]. Se define como orientado a objetos y basado en prototipos. El modelo de objetos de JavaScript está reducido y simplificado, pero incluye los elementos necesarios para que los scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor. Es el complemento ideal del HTML y una excelente solución para poner en práctica la validación de datos de un formulario en el lado del cliente.

1.6.3. Lenguaje de marcas de hipertexto: HTML versión 4

Lenguaje de Marcas de Hipertexto (HTML por sus siglas en inglés), hace referencia al lenguaje de marcado para la elaboración de páginas web que determina el contenido de la misma pero no su funcionalidad. Es el lenguaje que describe la estructura y el contenido semántico de un documento web que soporta imágenes y otro tipo de elementos multimedia para el desarrollo de sitios estáticos y dinámicos. Se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>), también puede describir hasta un cierto punto, la apariencia de un documento y es el lenguaje de publicación del *World Wide Web* (WWW¹¹) [21].

1.6.4. Hojas de estilo en cascada: CSS versión 3

Las hojas de Estilo en Cascada (*Cascading Style Sheets*), es un lenguaje formal usado para dar estilo a documentos HTML y XML, (y por extensión en XHTML) separando el contenido de la presentación. Los

⁹ **XHTML**: siglas del inglés *eXtensible HyperText Markup Language*. XHTML es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.

¹⁰ **XML**: por sus siglas en inglés *eXtensible Markup Language* ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el *World Wide Web Consortium* (W3C) utilizado para almacenar datos en forma legible.

¹¹ **WWW**: en informática, la *World Wide Web* (WWW) o Red informática mundial comúnmente conocida como la web, es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Con un navegador web, un usuario visualiza sitios web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia y navegar a través de esas páginas usando hiperenlaces.

Estilos definen la forma de mostrar los elementos HTML y XML. CSS¹² permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. El W3C¹³ es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario [22].

1.6.5. Lenguaje de Consultas Estructurado: SQL versión 2012

Lenguaje de Consultas Estructurado (SQL por sus siglas en inglés), es un lenguaje declarativo de alto nivel vinculado con la gestión de bases de datos de carácter relacional, que permite especificar diversos tipos de operaciones en ellas. En esencia, el SQL es un lenguaje declarativo de alto nivel ya que, explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones. Una sentencia de SQL puede resultar equivalente a más de un programa que emplee un lenguaje de bajo nivel. Se habla por tanto de un lenguaje normalizado que permite trabajar con (PHP) en combinación con cualquier tipo de base de datos (MS Access, SQL Server, MySQL) [23].

1.6.6. Modelo y Notación de Proceso de Negocio: BPMN versión 2.0

Modelo y Notación de Procesos de Negocio (BPMN por sus siglas en inglés), es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. Este estándar agrupa la planificación y gestión del flujo de trabajo, así como el modelado y la arquitectura [24]. El modelado en BPMN se realiza mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos. Presenta cuatro categorías básicas de elementos, como son:

- Objetos de Flujo: Eventos, Actividades, Rombo de control de flujo.
- Objetos de Conexión: Flujo de Secuencia, Flujo de Mensaje, Asociación.
- Carriles de nado: Piscina, Carril.
- Artefactos: Objetos de Datos, Grupo, Anotación.

1.6.7. Lenguaje Unificado de Modelado: UML versión 2.0

Lenguaje de Modelado Unificado (UML por sus siglas en inglés), es el lenguaje de modelado más conocido y utilizado en la actualidad para especificar o para describir métodos o procesos. Es un lenguaje

¹² **CSS**: por sus siglas en inglés *Cascading Style Sheets* - Hojas de Estilo en Cascada - es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora o por voz.

¹³ **W3C**: el Consorcio World Wide Web (W3C) es una comunidad internacional donde las organizaciones a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web, la misión del W3C es guiar la Web hacia su máximo potencial.

gráfico para visualizar, especificar, construir y documentar los artefactos de los sistemas de software, así como para el modelado del negocio y otros sistemas. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables [25]. Diseñado para aplicarse con todos los métodos de desarrollo, etapas de ciclo de vida, dominios de aplicación y medios. Se puede emplear en el desarrollo de software brindando gran variedad de formas para dar soporte a una metodología de desarrollo, pero no especifica en sí mismo que metodología o proceso usar.

UML consta de tres clases de bloques de construcción:

- Elementos: son abstracciones de cosas reales o ficticias (objetos, acciones).
- Relaciones: relacionan los elementos entre sí.
- Diagramas: son colecciones de elementos con sus relaciones.

De esta forma posibilita la construcción de una serie de requerimientos y estructuras necesarias para originar un sistema de software previo al proceso intensivo de escribir código.

1.6.8. Lenguaje de Marcas Extensibles: XML versión 1.0

El Lenguaje de Marcas Extensibles (XML por sus siglas en inglés) es un lenguaje de marcas desarrollado por el W3C utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML¹⁴ y permite definir la gramática de lenguajes específicos para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. El XML a diferencia del HTML, separa el contenido de la presentación. Tiene un papel muy importante ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Se puede comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos [26].

1.7. Herramientas a utilizar para el desarrollo de la solución

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo. Cada herramienta se crea y diseña para una o varias funciones determinadas, por lo que existe diversos tipos de herramientas informáticas según el campo al que se dediquen.

¹⁴ **SGML** :lenguaje de marcado generalizado estándar (por sus siglas en inglés de Standard Generalized Markup Language) (SGML; ISO 8879: 1986) es un estándar para definir lenguajes de marcado generalizadas para documentos.

1.7.1. Entorno de Desarrollo Integrado: NetBeans versión 8.0

NetBeans es un IDE escrito en Java, de código abierto (OpenSource), gratuito para desarrolladores de *software*. Ofrece todas las herramientas necesarias para crear aplicaciones profesionales, empresariales, web y móviles con el lenguaje Java, JavaFX, C/C ++ y lenguajes dinámicos como PHP, JavaScript, Groovy y Ruby, es fácil de instalar y se puede ejecutar tanto en *Windows*, como en Linux, Mac OS X y Solaris. NetBeans ofrece una versión del IDE hecho a la medida para el desarrollo de sitios web en PHP que comprenden una variedad de secuencias de comandos y lenguajes de marcado; se integra dinámicamente con HTML, JavaScript y CSS; cuando se configura el proyecto, se ejecuta en el servidor web incorporado a PHP; es totalmente compatible con el desarrollo iterativo, por lo que las pruebas proyectos PHP sigue los patrones clásicos familiares para los desarrolladores web [27]. Las aplicaciones implementadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de *software*.

1.7.2. Administrador de base de datos: PgAdmin III versión 1.14.0

PgAdmin III es una herramienta de código abierto para administrar el sistema gestor de base de datos PostgreSQL. Capáz de gestionar versiones de PostgreSQL 7 y superiores a ella, ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar base de datos complejas. La aplicación incluye:

- Herramienta de consulta SQL¹⁵.
- Interfaz administrativa gráfica.
- Editor de código procedural.
- Agente de planificación SQL/shell/batch.

La conexión al servidor puede hacerse mediante la familia de protocolos de internet (TCP/IP) y ser cifrado mediante el protocolo de Capa de Conexión Segura (SSL) para mayor seguridad [28].

1.7.3. Herramienta para prototipos: Evolus Pencil versión 2.0.5

Es una herramienta para elaborar prototipos libres y de código abierto con interfaz gráfica de usuario, disponible para todas las plataformas, de fácil instalación y manejo para crear maquetas de plataformas de escritorio populares. Los proyectos se pueden exportar en los formatos HTML, PNG (*Portable Network Graphics*, “Gráficos de Red Portátiles”), documento Open office, documento de Word y PDF (*Portable Document Format*, “Formato de Documento Portátil”). Permite instalar plantillas definidas por el usuario,

¹⁵SQL: por sus siglas en inglés significa Lenguaje de Consultas Estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

operaciones de dibujo estándar: alinea, escala y rota, y la adición de los objetos externos [29]. Se decide utilizar la herramienta para el desarrollo de los prototipos de interfaz de la propuesta de solución.

1.7.4. Herramienta de modelado: Visual Paradigm para UML versión 8.0

Es una herramienta profesional que soporta el ciclo completo de desarrollo del *software*: análisis y diseño orientados a objetos, construcción y despliegue a través de la representación de varios tipos de diagramas. Es una herramienta UML, multiplataforma y cuenta con una versión libre para la comunidad. Genera la documentación del sistema en los formatos PDF, HTML y el formato de documentos de Microsoft Word. Permite importar proyectos de otras herramientas de modelado como Rational Rose, Erwin y Microsoft Visio. Soporta la revisión ortográfica, brindando sugerencias para los idiomas: inglés, español, francés, alemán y portugués. Ayuda a los clientes a transformar con precisión sus requisitos del sistema en soluciones de *software* de alta calidad. Todos los productos de Visual Paradigm están diseñados y desarrollados para eliminar la complejidad, mejorar la productividad y comprimir el desarrollo de *software* según plazos emitidos por los clientes [30].

1.7.5. Sistema gestor de base de datos: PostgreSQL versión 9.4.1

Es un Sistema Gestor de Bases de Datos Objeto-Relacional (de sus siglas en inglés ORDBMS) basado en el proyecto Postgres, de la universidad de Berkeley, que contiene características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. También, PostgreSQL se puede extender por el usuario en muchas maneras, por ejemplo, mediante la adición de nuevos tipos de datos, funciones, operadores, funciones de agregado, métodos de índice y lenguas de procedimiento. Debido a la licencia liberal, PostgreSQL puede ser utilizado, modificado y distribuido por cualquier persona de forma gratuita para cualquier propósito, ya sea privada, comercial o académico [31].

1.7.6. Servidor web: Apache versión 2.4.7

Apache es un servidor web altamente configurable, robusto y estable. Es un sistema de código abierto para plataformas Windows, Unix, Macintosh y otras que implementa el protocolo HTTP¹⁶. Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP. Es multiplataforma y modular, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con la API¹⁷ de programación de módulos, para el

¹⁶ **HTTP**: por sus siglas en inglés significan Protocolo de Transferencia de Hipertexto, es el protocolo usado en cada transacción de la *World Wide Web*.

¹⁷ **API**: *Application Programming Interface*, es un conjunto de subrutinas, funciones y procedimientos o métodos, en la programación orientada a objetos, que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.

desarrollo de módulos específicos [32]. También permite personalizar la respuesta ante posibles errores y posibilita configurarlo para que ejecute un determinado *script* cuando ocurra un error en concreto. Permite la creación de ficheros de *log* facilitando, de este modo, el control de las acciones realizadas en el servidor.

1.7.7. Herramienta de pruebas: Apache JMeter versión 2.3.1

Apache JMeter, es una herramienta de código abierto que se utiliza para realizar pruebas de rendimiento y resistencia, habitualmente a aplicaciones web. JMeter permite realizar simulaciones de gran carga en el servidor, red o aplicación para comprobar su “fuerza” y para analizar el rendimiento ante diferentes tipos de sobrecarga. Permite cargar y realizar pruebas sobre distintos tipos de servidores mediante un sistema de pruebas distribuido. JMeter es multiplataforma y permite realizar pruebas de forma concurrente. A esto se añade su interfaz gráfica permitiendo realizar las operaciones a una mayor velocidad [33].

1.8. Conclusiones parciales

El argumento de los conceptos asociados al objeto de estudio sirve de base y guía para el entendimiento de los procesos de gestión de activos logrando así, fundamentar las bases teóricas de la investigación. El análisis a sistemas de similares propósitos facilita concluir que, como no satisfacen las necesidades y condiciones requeridas, se hace necesario desarrollar una solución informática para mejorar la gestión y el control de la información en el proceso de baja de los Activos Fijos Tangibles de las áreas administrativas de la UCI. Se utiliza la metodología DAC para estructurar, planear y guiar el proceso de desarrollo y se analizaron las herramientas, lenguajes y proceso de desarrollo de software definido por el Departamento de Desarrollo de la DIN, que permite profundizar los conocimientos necesarios para el desarrollo de la solución informática.

Capítulo 2: Concepción de la propuesta de solución

En el capítulo se presentan, las características de la propuesta de solución, iniciándose con la descripción de los procesos de negocio asociados a la gestión de baja de los AFT. Se definen las reglas de negocio y se representa una descripción detallada de los requisitos funcionales y no funcionales con los que debe cumplir la propuesta de solución, así como las técnicas empleadas para su obtención. Se realiza un estudio de la arquitectura y los patrones de diseño y de base de datos utilizados. Se representa el modelo de base de datos y modelo de despliegue.

2.1. Modelado del proceso de negocio

Un modelado de procesos de negocio es un conjunto estructurado y medible de actividades diseñadas para producir un producto especificado para un cliente o mercado específico. Implica un fuerte énfasis en cómo se ejecuta el trabajo dentro de la organización [34].

2.1.1. Descripción del proceso de negocio

Un proceso de gestión de baja de AFT en la Universidad de las Ciencias Informáticas consta de todo un trámite de solicitudes, verificaciones y aceptación de los responsables vinculados en dicho proceso, para ello se identifican tres subprocesos que comprende toda la gestión desde el inicio de una solicitud de baja hasta el cierre de la misma: **Aprobación de baja de AFT** inicia con una solicitud realizada por el área administrativa que concluye con la conformación de un Expediente de baja, compuesto por diferentes documentos generados a partir de dictámenes emitidos por una comisión base y confirmados por una comisión central, que bajo el respaldo de la Resolución No.52/2016 y No.175/2016 designa a sus miembros a decidir sobre el mal estado de los activos y proceder con la baja técnica de los mismos. Cada uno de estos documentos son avalados por rector(a) de la universidad. **Implementación de baja de AFT** se enfoca en la gestión por parte del área administrativa en conjunto con la Dirección de Servicios Generales, al destino final de dichos activos, donde se adjuntan otro conjunto de documentos que evidencian la finalidad del proceso y **Cierre de baja de AFT** donde una vez conformado el Expediente de baja con todos los documentos que se requieren, se prosigue a dar de baja del sistema ASSETS por parte de la Dirección de Contabilidad y Finanzas de la universidad.

Roles que intervienen en el proceso:

Especialista del área: persona asignada para la gestión de los activos del centro de costo correspondiente al área asociada. Asignándole por sistema el rol de Asesor(s) económico, donde se le otorgan los permisos para realizar las solicitudes de bajas a Comisión base y Dirección de contabilidad, y generar los movimientos temporales sin destinos a los activos dictaminados como baja por la Comisión.

Comisión base: grupo de personas que bajo el respaldo de la Resolución No.52/2016 designa a sus miembros a decidir sobre el mal estado de los activos y proceder con la baja técnica de los mismos, que posterior a la verificación realizada, son encargados de generar el dictamen técnico para cada activo y un informe general de la solicitud de baja atendida.

Especialista de Contabilidad y Finanzas: persona asignada por la Dirección de Contabilidad y Finanzas para llevar a cabo el proceso de baja de activos. Responsable de recibir y evaluar el expediente de baja emitido por un área y generar un acta de Comisión central para posterior entrega del expediente a la misma. Confirma al área la aceptación de la solicitud realizada y procede a dar baja en el sistema ASSETS, una vez terminado el proceso y conformado en su totalidad el expediente de baja.

Comisión central: grupo de personas que bajo el respaldo de la Resolución No.175/2016 designa a sus miembros a reevaluar el estado de los activos solicitados, como estado de baja.

Rector(a): máxima autoridad de la universidad que avala con su firma cada uno de los documentos adjuntos al expediente de baja.

Sistema ASSETS: sistema utilizado en la universidad para la gestión de los procesos contables, financieros y de recursos humanos. Dentro de los procesos contables atiende el control de los Activos Fijos Tangibles y Útiles, registrando todas las bajas una vez aceptado y gestionado todo el proceso. Generando los documentos de baja de activos y comprobante de operaciones, necesarios para ser adjuntados al expediente de baja.

Los subprocesos se desarrollan de la siguiente manera:

Aprobación de baja de AFT

En un área administrativa de la universidad al identificar un activo deteriorado e inprovechable, el responsable designado al control y gestión de los mismos en dicha área comienza realizando una solicitud de baja a la comisión definida a nivel rectoral que bajo el respaldo de la Resolución No.52/2016 designa a sus miembros a decidir sobre el mal estado de los activos y proceder con la baja técnica de los mismos. Una vez realizada la verificación por dicha comisión, se genera un dictamen técnico por cada uno de ellos que expone la necesidad de ser reparados o dados de baja, especificando si son desechados completamente o provechosos para materias primas, en caso de que todos los activos de la solicitud se encontrase en estado de reparación, es informado al área que ninguno será procesado como baja, de lo contrario se comienza la conformación de un Expediente de baja inicialmente con los Dictámenes técnicos de baja para cada activo y un Informe General de la solicitud, que es entregado al responsable del área, encargado de generar un acta de solicitud de baja oficial dirigida a la Dirección de Contabilidad y

Finanzas, donde refleja solo los activos que fueron dictaminados como baja y un movimiento temporal sin destino para cada uno de ellos.

Este Expediente es entregado a la Dirección de Contabilidad y Finanzas, siendo revisado por el especialista del área, detectando cualquier error de datos en los documentos, de ser confirmado algún posible error es devuelto al área para su rectificación y continuar el proceso, en caso de encontrarse en orden, enumera y actualiza el Expediente con el Acta de Comisión de baja y pasa a manos de la Comisión Central que bajo el respaldo de la Resolución No.175/2016 es encargada de reevaluar el estado de los activos y emitir otro documento de Acta de comprobación. Una vez concluida todas las verificaciones se prosigue a firmar el Acta de comisión de baja por todos sus integrantes y rector(a) de la universidad, que certifica cada uno de los documentos del expediente con su firma.

El Expediente va siendo devuelto a las diferentes áreas responsables de evaluar el proceso de forma descendiente hasta llegar a la Dirección de Contabilidad y Finanzas, que archiva el expediente y confirma al área administrativa la aprobación de la solicitud de baja realizada.

Para una mayor comprensión del subproceso Aprobación de baja de AFT, se muestra en la Figura 2 el diagrama del proceso de negocio.

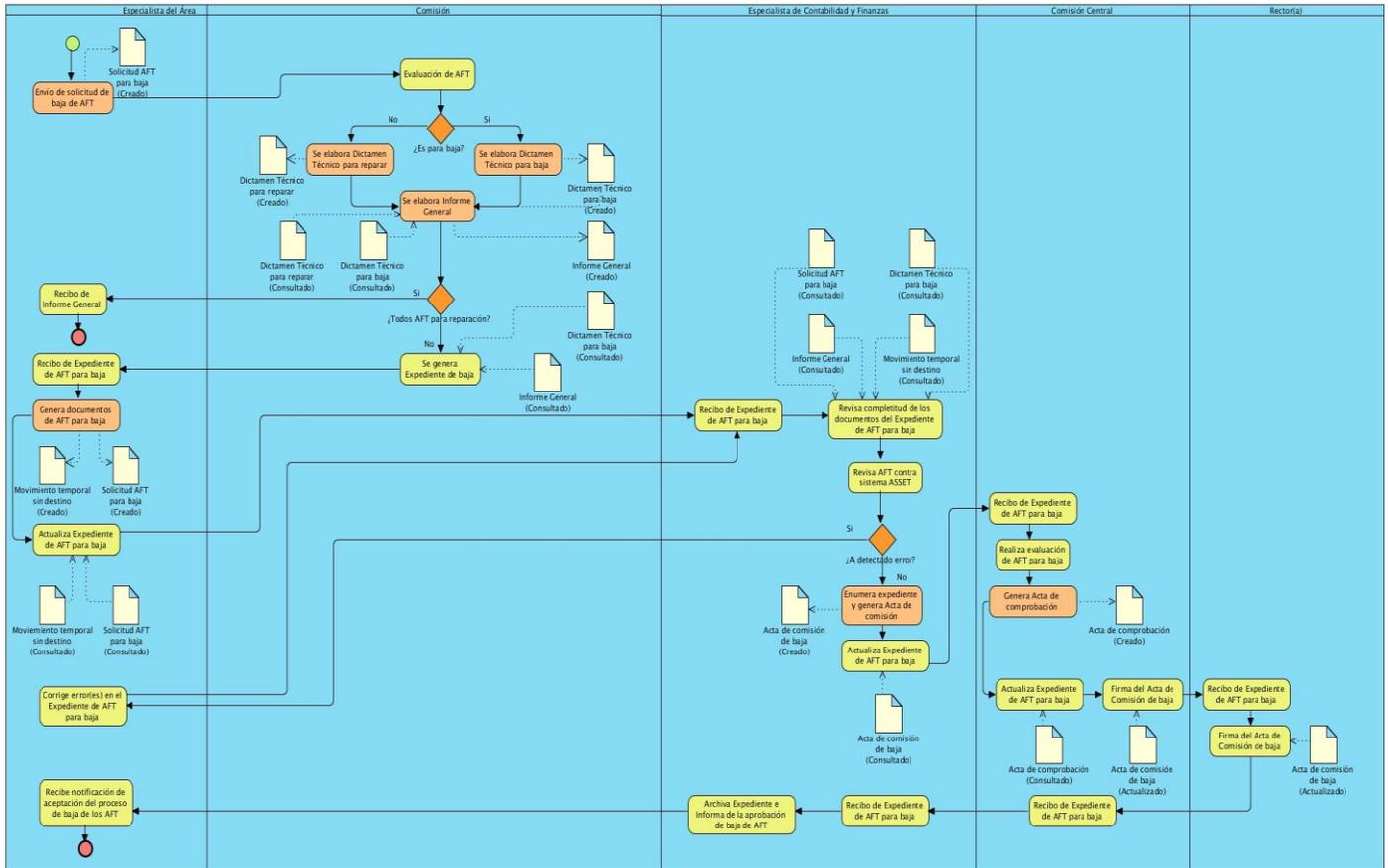


Figura 2. Diagrama de proceso de negocio de Aprobación de baja de AFT.

Implementación de baja de AFT

El área administrativa al ser informada de la aceptación de las solicitudes de bajas realizadas, se comunica con la Dirección de Servicios Generales encargada de la gestión para el destino final de dichos activos, proceso en el que se generan documentos, preceptivo en el Expediente de baja:

Tabla 1. Listado de documentos generados en Implementación de baja de AFT.

Documentos	Emite	Recibe
Carta con relación de los activos que van para materia prima, con pie de firma del área administrativa y Dirección de Servicios Generales.	Área administrativa	Dirección de Servicios Generales
Carta con relación de los activos que van para basurero, con pie de firma del área administrativa y Dirección de Servicios Generales.	Área administrativa	Dirección de Servicios Generales

Acta de entrega de activos a materia prima, con pie de firma de especialista y director de Servicios Generales y especialista de la Dirección de Contabilidad y Finanzas.	Dirección de Servicios Generales	Dirección de Contabilidad y Finanzas
Acta de entrega de activos al basurero, con pie de firma de especialista y director de Servicios Generales y especialista de la Dirección de Contabilidad y Finanzas.	Dirección de Servicios Generales	Dirección de Contabilidad y Finanzas
Carta de evidencia del desecho de los activos al basurero, con pie de firma de una comisión de tres testigos propios del área administrativa.	Área administrativa	Dirección de Contabilidad y Finanzas
Foto de evidencia del desecho de los activos en el basurero.	Área administrativa	Dirección de Contabilidad y Finanzas
Factura que emite materia prima a la entrega de los activos.	Dirección de Servicios Generales	Dirección de Contabilidad y Finanzas

Cierre de baja de AFT

Una vez conformado el Expediente de baja con todos los documentos anexos a él, el especialista de Contabilidad y Finanza realiza una revisión confirmando la integridad del mismo, luego prosigue efectuando la baja de los activos solicitados, en el sistema ASSETS, donde se generan dos nuevos documentos también adjuntos al Expediente: Baja de activos y Comprobante de operaciones. Concluye informando la confirmación de cierre del proceso de baja a la especialista del área administrativa.

Para una mayor comprensión del proceso se muestra el diagrama del proceso de negocio, accesible en el Anexo 2.

2.1.2. Reglas del negocio

Las reglas de negocio son condiciones, estándar o reglas que deben ser cumplidas y controladas por la organización durante el flujo del proceso [35]. El proceso de gestión de baja de AFT presenta diversas reglas que permiten su ejecución con la mayor calidad posible.

- ✓ Aprobación de baja de AFT

- El especialista del área solo podrá efectuar solicitudes de baja de su centro de costo y áreas de responsabilidades correspondiente.
- Un jefe de comisión tendrá permisos para generar un proceso de baja desde el comienzo de una solicitud.
- El dictamen técnico no se generará para ningún activo reportado como reparable.
- El dictamen técnico se generará para cada uno de los activos dictaminados como baja.
- Cada dictamen técnico es avalado con la firma de todos los participantes de la comisión que verificaron el estado de los activos.
- El movimiento temporal sin destino se generará para cada uno de los activos dictaminados como baja.
- Se desactivará la opción del registrar movimiento temporal sin destino para los activos dictaminados para reparar.
- Cada solicitud de baja responde a un informe general, este último recoge el estado de los activos dictaminados como baja y reparación.
- Un informe general no será concebido hasta no haber sido dictaminado cada uno de los activos de una misma solicitud de baja.
- La cantidad de AFT a procesar del informe general, responde a un total de activos tramitados, desglosados posteriormente en cantidad de activos dictaminados como baja y para reparar.
- El documento generado de solicitud de baja a la Dirección de Contabilidad y Finanzas, reflejará una relación de los activos dictaminados solo como baja.
- Cada expediente de baja posee un único número, con formato #/año.
- Los activos reportados en la solicitud de baja a la Dirección de Contabilidad y Finanzas, deben estar en correspondencia con los activos del área registrados en el sistema ASSETS.
- El acta de Comisión de baja se generará para cada solicitud de baja.
- El acta de Comisión de baja es avalada por la firma de todos los integrantes de la comisión central y rector(a) de la universidad.

✓ Cierre de baja de AFT

- El expediente de baja debe poseer todos los documentos de registro obligatorio para el proceso de baja.

2.2. Especificación de requisitos de software

Los requisitos para un sistema de software determinan lo que debe hacer el sistema y definen las restricciones en su funcionamiento e implementación, es decir, lo que el software debe hacer y bajo qué circunstancias debe hacerlo [36]. Para el desarrollo de la propuesta de solución se identificaron requisitos funcionales y no funcionales, siendo los primeros guiados mediante técnicas de obtención que permiten establecer una comunicación más específica entre el cliente y el equipo de desarrollo.

2.2.1. Técnicas de obtención de requisitos

La obtención de requerimientos es una de las cuatro actividades que define la Ingeniería de Requisitos, para ayudar a reconocer la importancia que tiene una especificación y administración adecuada de los requerimientos de los usuarios. La obtención de requisitos es el proceso donde los interesados en un sistema de software descubren, revelan y entienden sus requerimientos. Existen diferentes técnicas para identificar los requisitos, dentro de las que se encuentran: sesiones de tormentas de ideas, entrevistas, observación de campo, revisión de la documentación técnica, análisis de sistemas existentes, ingeniería inversa, simulaciones y prototipado [37]. Para la captura de requerimientos de la solución propuesta se utilizaron las siguientes técnicas:

- **Sesiones de tormentas de ideas:** es una técnica de reuniones en grupo que permite generar ideas originales en un ambiente libre de críticas. Se realizaron reuniones con la participación de clientes e integrantes del proyecto de SGU, con el objetivo de definir las funcionalidades y esclarecer las dudas en el proceso de gestión de baja de AFT. A partir de las reuniones se encontraron algunas funcionalidades e interfaces que sirvieron de base para la propuesta de solución.
- **Entrevistas:** se realizaron reuniones con las partes interesadas, donde se efectuaron una serie de preguntas para lograr una mayor comprensión de la problemática existente. Las entrevistas permitieron identificar varios de los requerimientos de la solución. Sus diseños están accesibles en el Anexo 1.
- **Prototipado:** un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y conocer más acerca del problema. Esta técnica permite formalizar la aceptación previa de los requisitos por los clientes.

2.2.2. Definición de Requisitos funcionales

Los Requisitos funcionales de un sistema describen lo que el sistema debe hacer. Dependen del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general de la organización tomado al redactar los mismos [38]. En la siguiente tabla se muestran los Requisitos funcionales definidos para la propuesta de solución con la complejidad y la prioridad para el cliente.

Tabla 2. Listado de Requisitos funcionales.

No.	Requisitos	Complejidad	Prioridad para el cliente
RFAct1	Listar solicitudes primarias de baja AFT	Media	Alta
RFAct2	Crear solicitud primaria de baja AFT	Media	Alta
RFAct3	Modificar solicitud primaria de baja AFT	Media	Alta
RFAct4	Exportar documento de solicitud primaria de baja AFT	Baja	Alta
RFAct5	Eliminar solicitud primaria de baja AFT	Media	Alta
RFAct6	Enviar a dictamen técnico AFT	Baja	Alta
RFAct7	Confirmar recepción de dictamen técnico AFT	Media	Alta
RFAct8	Exportar en formato .pdf informe final de evaluación de AFT	Baja	Alta
RFAct9	Confirmar realización de evaluación de AFT	Baja	Alta
RFAct10	Listar AFT a dictaminar	Media	Alta
RFAct11	Registrar dictamen técnico de AFT	Alta	Alta
RFAct12	Modificar dictamen técnico de AFT	Alta	Alta
RFAct13	Exportar en formato .pdf documento de dictamen técnico de AFT	Media	Alta

RFAct14	Listar solicitudes oficiales de baja de AFT	Media	Alta
RFAct15	Exportar solicitud oficial de baja de AFT	Media	Alta
RFAct16	Registrar reevaluación de AFT	Media	Alta
RFAct17	Exportar en formato .pdf acta de comprobación de AFT	Baja	Media
RFAct18	Exportar en formato .pdf acta de comisión central de AFT	Baja	Alta
RFAct19	Exportar en formato .pdf documento de movimiento temporal sin destino	Baja	Alta
RFAct20	Modificar evaluación de AFT	Media	Alta
RFAct21	Registrar baja de AFT	Media	Alta
RFAct22	Listar comisiones	Media	Media
RFAct23	Crear comisión	Alta	Media
RFAct24	Modificar comisión	Alta	Media
RFAct25	Ver detalles de comisión	Media	Media
RFAct26	Eliminar comisión	Baja	Media
RFAct27	Listar asesores económicos	Media	Media
RFAct28	Registrar asesor económico	Media	Media
RFAct29	Modificar asesor económico	Media	Media
RFAct30	Eliminar asesor económico	Baja	Media
RFAct31	Listar subclasificaciones asociadas a tipo de medio de AFT	Baja	Baja
RFAct32	Asociar subclasificaciones a tipo de medio de AFT	Baja	Baja
RFAct33	Desasociar subclasificaciones asociadas a tipo de medio de AFT	Baja	Baja
RFAct34	Mostrar solicitudes de baja de AFT por área	Media	Alta
RFAct35	Detalles estadísticos del dictamen por solicitud de baja	Media	Alta
RFAct36	Totales estadísticos de solicitudes por área	Media	Alta

2.2.3. Definición de Requisitos no Funcionales

Los Requisitos no Funcionales imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe tener [38]. En la Tabla 3 se muestra los Requisitos no Funcionales definidos para la propuesta de solución.

Tabla 3. Listado de Requisitos no Funcionales.

Usabilidad	
RNFAct1	La aplicación debe estar dirigida a registrar y gestionar la información referente al proceso de baja de AFT y Útiles.
RNFAct2	Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan acceder mostrando en la vista mediante iconos y menús el acceso.
RNFAct3	La información que se registra estará protegida de modificaciones no deseadas de acuerdo a la estrategia de seguridad definida.
RNFAct4	Las vistas del sistema deben indicar en cada momento la acción que se está realizando así como lo iconos deben estar representados por una imagen acorde a la acción que se realiza.
RNFAct5	El sistema debe presentar un menú lateral y una barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.
Software	
RNFAct6	Para el despliegue del sistema se contará en el servidor de base de datos con PostgreSQL 9.4 o superior bajo el sistema operativo CentOS 7.0 o superior.
RNFAct7	Para el despliegue del sistema estará instalado en el servidor de aplicaciones web con el sistema operativo CentOS 7.0 donde se encuentre instalado PHP v5.6 con las librerías php5-ldap, php5-gd, php5-mcrypt, php5-pgsql, php5-xsl, php5-openssl, Apache 2.2 con el módulo <i>rewrite</i> activado.
RNFAct8	Para el uso del sistema se requiere una PC cliente con el navegador web Mozilla Firefox 3.6 o superior.
Hardware	
RNFAct9	El servidor de aplicación y el de base de datos deberán tener los siguientes componentes de hardware para su funcionamiento en el ambiente en que se encuentra desplegado el módulo Activos: Microprocesador de 8 núcleos, 4GB de memoria RAM, 250 GB disco duro y una fuente de 800 W como mínimo.
RNFAct10	Para la ejecución del sistema se requiere que la PC cliente tenga los siguientes componentes de hardware: Pentium 4 o superior, 512 MB RAM como mínimo.
Confiabilidad	
RNFAct11	El sistema puede permanecer inactivo durante 10 minutos. Al cumplirse el término se cerrará la sesión

	teniendo que autenticarse el usuario nuevamente.
RNFAct12	El sistema de autenticación deberá recoger datos necesarios de los usuarios y tendrá control de identidades.
RNFAct13	El tratamiento de las excepciones permitirá un seguimiento hasta guardar información acerca del lugar dónde se produjo el error y de los parámetros de configuración del sistema que lo provocaron.
RNFAct14	Cuando ocurre una excepción el sistema mostrará un mensaje explicativo del error ocurrido y permanecerá en el mismo estado sin realizar ninguna otra operación.
RNFAct15	El sistema de autenticación debe permitir auditoría de trazas de las acciones de los usuarios para lo que se proveerá algún mecanismo para la configuración dinámica y centralizada de la gestión de historiales.
RNFAct16	El servidor de aplicaciones y de base de datos deberá mantener una seguridad mediante firewall para proteger el código y la información.
Eficiencia	
RNFAct17	El sistema soportará la conexión simultánea de un promedio de 50 y un máximo de 500.
RNFAct18	El tiempo promedio de respuesta del sistema no excederá los 8 segundos de manera general.
Soporte	
RNFAct19	El sistema se registrará por un estándar de código debidamente documentado en el expediente de proyecto.
RNFAct20	Los productos de trabajo generados en el desarrollo del software se registrarán por las pautas de configuración debidamente documentadas en el expediente de proyecto.
Restricciones de diseño	
RNFAct21	El software deberá ser desarrollado en su totalidad con tecnologías y componentes de código abierto.
Interfaz	
RNFAct22	La comunicación entre el cliente y el servidor de aplicaciones se lleva a través del protocolo HTTPS.
RNFAct23	La comunicación entre el servidor de aplicaciones y la base de datos se lleva a través del protocolo TCP/IP.
RNFAct24	Se tendrán centralizados los archivos de configuración. Para ellos se utilizarán archivos XML que permitirán el fácil mantenimiento del sistema si cambia alguna de las configuraciones que posee.
Requisitos de licencia	
RNFAct25	El sistema se caracterizará por un licenciamiento cerrado. El producto está basado en código abierto pero no está disponible bajo licenciamiento de código abierto.
Estándares aplicables	
RNFAct26	La gestión de la Comisión base se registrará por la Resolución No. 52/2016 emitido a los 8 días del mes de marzo del 2016 por el rector(a) de la Universidad de las Ciencias Informáticas.
RNFAct27	La gestión de la Comisión central se registrará por la Resolución No. 175/2016 emitido por el rector(a) de

	la Universidad de las Ciencias Informáticas.
RNFAct28	Para exportar datos del sistema se usa el Formato de Documento Portátil (.pdf): norma ISO 32000.

2.2.4. Descripción de requisitos

Con el objetivo de presentar los Requisitos funcionales de un modo más sólido y por ende más entendible, se realizó la descripción de cada Requisito funcional. Se definieron secciones para describir detalladamente cada requisito y las acciones que se deben llevar a cabo por el usuario para cumplir con esa funcionalidad en el sistema. A continuación, se muestra en la Tabla 4 una de las descripciones de requisitos diseñadas, para consultar otras descripciones remitirse al Anexo 3.

Tabla 4. Descripción de Requisito funcional.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFAct2	Crear solicitud primaria de baja AFT	<ul style="list-style-type: none"> El requisito permite crear una solicitud de baja de un AFT. El usuario se dirige al encabezado de la página y selecciona de la iconografía para el acceso a los componentes del Núcleo del SGU, el ícono correspondiente al módulo “Activos”, luego en la agrupación funcional Proceso de bajas selecciona Solicitud de bajas y dentro de ella en el área de íconos flotantes la opción Crear solicitud de baja AFT. En la interfaz se muestra la lista de AFT (sin asociar) para que el usuario a través del botón Asociar incorpore en la lista de AFT (asociados) los AFT que desea incorporar a la solicitud de baja. Al seleccionar el botón Aceptar la solicitud de baja queda registrada en el sistema. 	Media	Alta
Prototipo				

☰

Registrar solicitud de baja

Centro de costo:

Área de responsabilidad:

Subclasificación:

Cantidad por página

<input type="checkbox"/>	Nombre	Rótulo	Centro de costo	Área de responsabilidad	Subclasificación
<input type="checkbox"/>	BURO EJECUTIVO CON ANGULO DE CONEXION REDONDO	MB0075828	Facultad 6	DECANO	BURO

Resultados encontrados: 1 Página 1 de 1

Activos asociados

Nombre	Rótulo	Subclasificación	Centro de costo	Área de responsabilidad	Causa de baja	Opciones
BURO EJECUTIVO CON ANGULO DE CONEXION REDONDO	MB0075828	BURO	Facultad 6	DECANO	<input type="text" value="Roto"/>	<input type="button" value="🗑️"/>
GAVETERO MOVIL	MB0075910	GAVETERO	Facultad 6	DECANO	<input type="text" value="Roto"/>	<input type="button" value="🗑️"/>

Campos	Tipos de Datos	Reglas o Restricciones
Tabla	Listado de AFT (sin asociar)	<p>Muestra la lista de todos los AFT existentes en el sistema que estén en estado activo, que no estén en trámite de baja y que no sean tecnológicos.</p> <p>Deben mostrarse ordenados alfabéticamente por el campo Nombre del AFT. Y de cada AFT los datos: Nombre, Rótulo, Centro de costo, Área de responsabilidad y Subclasificación.</p>
Botón Asociar	Botón	Permite incorporar a la tabla de Listado de AFT (asociados) los activos seleccionados en la tabla Listado de AFT (sin asociar).
Tabla	Listado de AFT (asociados)	<p>Muestra la lista de todos los AFT que formarán parte de la solicitud de baja.</p> <p>Deben mostrarse ordenados alfabéticamente por el campo Nombre del AFT. Y de cada AFT los datos: Nombre, Rótulo, Centro de costo, Área de responsabilidad y Subclasificación.</p>
Buscar	Campo de texto	Permite introducir números y letras.

Botón Buscar	Botón	Filtra la lista de AFT (sin asociar) con los AFT que su rotulo se corresponda en total o parcial medida con el introducido en el campo Buscar.
Centro de costo	Campo de selección	<p>Debe mostrar todos los centros de costo existentes en el sistema ordenados ascendentemente por su identificador.</p> <p>Filtra la lista de AFT (sin asociar) mostrando los AFT que correspondan con el centro de costo seleccionado.</p>
Área de responsabilidad	Campo de selección	<p>Si no existe un Centro de costo seleccionado se muestra sin elementos de selección.</p> <p>Debe mostrar todas las áreas de responsabilidad existentes en el sistema asociadas al Centro de costo seleccionado en orden ascendentemente por su identificador.</p>
Subclasificación	Campo de selección	<p>Muestra los tipos de AFT no tecnológicos existentes en el sistema.</p> <p>Filtra la lista de AFT (no asociados) donde su subclasificación corresponda con la seleccionada.</p>
Cantidad por página	Campo de selección	<p>Muestra números (5, 10, 15, 20 y 120).</p> <p>Muestra en la lista de AFT (no asociados) la cantidad de activos que corresponda con el número seleccionado.</p>
Número de página	Campo de texto	<p>Permite introducir números.</p> <p>Muestra la página correspondiente al número introducido.</p> <p>En caso de que se introduzca un número de página mayor a la cantidad de páginas del listado, el mismo mostrará la última página del listado.</p>
Causa de baja	Campo de selección	Muestra las causas de baja existentes en sistema: Roto y Deterioro.
Botón Eliminar	Botón	<p>Si se desea eliminar un ATF del listado de ATF (asociado) se presiona el icono interno Eliminar y se muestra el mensaje de confirmación: “¿Está seguro de realizar la acción?”.</p> <p>Si selecciona Aceptar se elimina el ATF de la lista de ATF (asociado) y si selecciona Cancelar se muestra la interfaz de Crear solicitud de baja</p>

			AFT.
	Observaciones	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema con el rol de asesor económico y/o administrador del sistema. 2. Al seleccionar el botón Aceptar: <ul style="list-style-type: none"> • Si no existen elementos en la lista de AFT (asociados) se muestra el mensaje de validación: “Debe seleccionar al menos un activo.” • En caso de no haberse seleccionado causa de baja para un AFT se muestra el mensaje de validación: “Debe seleccionar una causa de baja para el activo.” • En caso de validaciones correctas se muestra un mensaje de información: “El elemento ha sido creado satisfactoriamente” y se muestra la interfaz Listar solicitud de baja AFT. 3. Al seleccionar el botón cancelar se muestra un mensaje de confirmación: “¿Está seguro de realizar la acción? “. Si selecciona Aceptar se muestra la interfaz Listar solicitud de baja AFT, si selecciona Cancelar se muestra la interfaz Crear solicitud de baja AFT. 4. En caso de que el usuario seleccione la acción de Listar en el área de iconos flotantes el sistema mostrará la interfaz de Listar solicitud de baja AFT. 	

2.2.5. Validación de los requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto [39]. Existen varias técnicas para la validación de requisitos como son: revisiones formales de requerimientos, construcción de prototipos y generación de casos de pruebas. Para la validación de los requisitos de la solución se utiliza la técnica de revisión formal la que involucra al usuario del sistema y al equipo de desarrollo. Este último debe explicar al cliente las implicaciones de cada requisito y verificar su consistencia individualmente y como un todo.

A continuación se muestra las preguntas definidas para verificar la consistencia de cada requisitos y definir si los mismos pueden ser aprobados o no.

- ¿El requisito tiene un identificador único?

- ¿El requisito es modificable?
- ¿El requisito no es ambiguo?
- ¿El requisito es congruente con otros requisitos relacionados?
- ¿El requisito está completo?
- ¿El requisito puede ser implementado?
- ¿El requisito puede ser probado?
- ¿El requisito es correcto?
- ¿El proveedor del requisito es un proveedor válido?

Luego de aplicar el conjunto de interrogantes para validar los requisitos definidos para el desarrollo del módulo, se obtuvo el 100 % de aprobación por parte del cliente.

2.3. Descripción de la propuesta de solución

La propuesta de solución tiene entre sus principales objetivos además de agilizar y garantizar la gestión del proceso de baja de los Activos Fijos Tangibles, la generación de todos los documentos que se originan en dicho proceso.

El módulo Activos cuenta con tres agrupaciones funcionales macro: Proceso de baja, Configuración y Reportes. Cada una de ellas poseen a su vez un conjunto de agrupaciones funcionales, que se describen a continuación.

Proceso de baja presenta cuatro agrupaciones funcionales: Solicitud de baja, Dictamen de la comisión, Trámite de baja técnica y Movimiento temporal/Reevaluación.

Solicitud de baja con un conjunto de funcionalidades como Crear solicitud, permitirá siguiendo criterios de búsqueda, el acceso a la lista de activos o activo deseado de forma rápida y exacta, que serán seleccionados y asociados a una lista de activos que incluirá la solicitud. El documento de la solicitud podrá ser generado por las partes implicadas mediante el Exportar documento de solicitud. La funcionalidad Listar solicitud mostrará todas aquellas solicitudes realizadas que se encuentran en curso, las cuales podrán ser modificadas y eliminadas en caso de introducción de errores o por si así lo requiera el usuario y solo antes de que sean enviadas a dictamen técnico. Acción que dará paso a que sea visible dicha solicitud por la comisión base, que mediante el Confirmar recepción ratificará que la misma está siendo atendida. Una vez que la comisión base genere todos los dictámenes técnicos de cada uno de los activos de una solicitud, automáticamente se activará el Exportar informe final y el Confirmar realización de la evaluación, este último será ejecutado por el solicitante una vez que le haya sido entregado todos los

documentos generados por la comisión base. Dando continuidad de esta forma al proceso, ya que se generan todos los movimientos temporales sin destino de cada activo sumado a la solicitud oficial de baja.

La agrupación funcional Dictamen de la comisión contará con cuatro funcionalidades: Listar AFT a dictaminar, reflejando por separado cada uno de los activos, que estarán identificados por un mismo número de solicitud. Registrar dictamen técnico dará paso a la evaluación del activo donde se registrarán sus características, estado y causa de baja, generando de esta forma un documento mediante el Exportar dictamen técnico. El dictamen generado también podrá ser modificado en caso de introducción de errores y solo hasta que la comisión base entregue en formato duro los documentos generados al solicitante.

En Trámite de baja técnica se listan todas las solicitudes oficiales de baja, en las cuales solo se contemplan los activos que fueron dictaminados como baja, dicho documento es generado mediante el Exportar solicitud oficial de baja. Ya conformado y entregado el expediente de baja a comisión central, esta actualiza mediante el Registrar reevaluación el estado de los activos de una solicitud y origina mediante el Exportar acta de comprobación y Exportar acta de comisión central los documentos generados en el proceso de reevaluación de los activos.

En Movimiento temporal/Reevaluación se listan por separados todos los activos de la solicitud que está siendo atendida y fueron dictaminados con estado de baja, permitiendo generar los movimientos temporales sin destino para cada uno de ellos mediante la funcionalidad Exportar movimiento temporal sin destino. También se encuentra el Modificar evaluación, que se ejecutará en caso que la comisión central determine un cambio de estado del activo. Siempre se eliminan automáticamente del listado los movimientos temporales de los activos que fueron reevaluados y actualizado su estado como reparables. El registro de la baja representa una funcionalidad no visible, pues una vez terminado el proceso y dado de baja en el sistema ASSETS los activos que correspondan a una solicitud de baja que se encuentra activa en el sistema, instantáneamente desaparecerá de la lista de solicitudes.

Configuración contará con tres agrupaciones funcionales: Asesores económicos, Subclasificaciones de baja técnica y Comisión de dictamen técnico. Donde la funcionalidad Asesores económicos asignará el rol a la persona seleccionada, asociándole las áreas a las que tendrá los permisos para realizar un proceso de baja. Además, se encargará de poder modificar la asignación del área o las áreas asignadas al asesor económico, en caso de introducción de errores. Muestra la lista de los asesores activos en el sistema y brinda la opción de poder eliminarlos. En Subclasificaciones de baja técnica se listan las subclasificaciones asociadas a un tipo de medio (tecnológico o no tecnológico), brindando la posibilidad de asociar y desasociar una subclasificación del tipo de medio seleccionado. Comisión de dictamen técnico, listará las comisiones registradas en el sistema, mostrando mediante el Ver detalles las personas que las

integran. La funcionalidad también permitirá el crear una comisión especificando su tipo y asociando a ella las personas correspondientes. De igual forma permite modificar y eliminar una comisión en caso que se desee.

Los reportes a gestionar se clasifican en varios tipos dependiendo de las necesidades del usuario, brindando la posibilidad de exportarlos en formato pdf. La agrupación funcional Reportes facilitará la gestión de los mismos, entre los que se encuentran: Mostrar solicitudes de baja de AFT por área, Detalles estadísticos del dictamen por solicitud de baja y Totales estadísticos de solicitudes por área.

La Figura 3 muestra cómo se distribuyeron las funcionalidades de la propuesta de solución, una distribución más detallada se encuentra accesible en el Anexo 4.

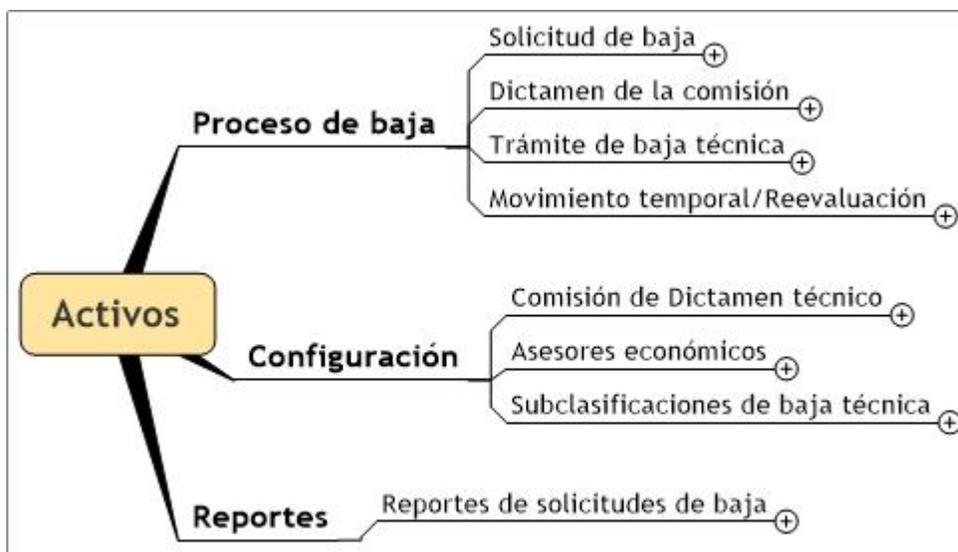


Figura 3. Mapa de navegación.

2.4. Descripción de la arquitectura

La arquitectura de un software es la estructura u organización de un sistema que incluye los componentes de este, las propiedades visibles externas de esos componentes y las relaciones que existen entre ellos [40]. Es una forma coherente de establecer patrones y abstracciones para que los analistas y desarrolladores trabajen en una línea común hacia la implementación del sistema.

La propuesta de solución a desarrollar está integrada al Núcleo del Sistema de Gestión Universitaria y debe cumplir con las pautas especificadas para el desarrollo de dicho sistema. La propuesta se ajustará a la arquitectura ya definida, proponiendo como estilo arquitectónico Cliente-Servidor haciendo uso del patrón Modelo-Vista-Controlador (MVC), que a su vez es el patrón de arquitectura base que utiliza el marco de trabajo GUUD.

2.4.1. Estilo arquitectónico

El estilo arquitectónico Cliente-Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos, servidor de aplicaciones o brindan determinado servicio. Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.

Una representación gráfica de este tipo de arquitectura sería la siguiente.

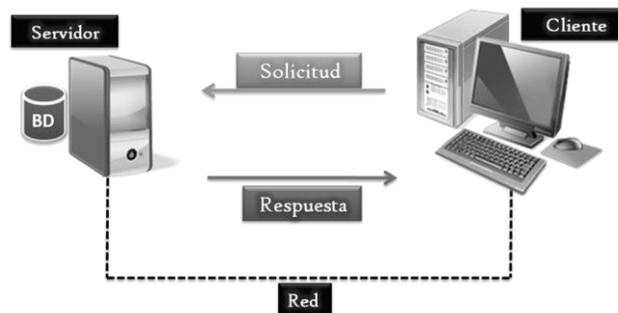


Figura 4. Representación de la arquitectura Cliente-Servidor.

El sistema se hospedará en el servidor web de aplicaciones Apache 2.4.7 ubicado en uno de los servidores del nodo central de la universidad, siendo de acceso mediante la dirección URL <https://activos.akademos.uci.cu/>. Esta arquitectura proporciona que los usuarios puedan acceder al sistema en cualquier dispositivo conectado a la red UCI logrando centralizar la información y facilitando el acceso a esta, sin necesidad de que todos tengan que utilizar el mismo sistema operativo.

2.4.2. Patrón de arquitectura

El patrón arquitectónico Modelo-Vista-Controlador separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. En el patrón Modelo Vista Controlador, la vista es la página HTML y el código provee los datos dinámicos de la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Las principales ventajas que se presentan al utilizar el patrón Modelo Vista Controlador radican en que se consiguen múltiples vistas del modelo, todas las vistas están sincronizadas, no acoplamiento y facilidad de

evolución para cambiar las vistas y los controladores y que la aplicación puede soportar un tipo de interfaz para cada rol del usuario.

El Modelo: contiene una representación de los datos que maneja el sistema, por lo que gestiona los accesos, las consultas y actualizaciones de los datos del sistema. Interactúa con los controladores de la aplicación, respondiendo sus peticiones de acceso al sistema de persistencia de información.

La Vista: o interfaz de usuario, se comunica con el controlador del que recibe los datos para su visualización y captura los eventos que desencadenan acciones de acceso a datos.

El Controlador: actúa como intermediario entre el Modelo (proveedor de la información) y la Vista (forma en que se presenta la información), gestionando el flujo de información entre ellos. Reacciona a los eventos desencadenados por el cliente, ejecutando la acción adecuada e invoca peticiones a los modelos que necesite cuando se hace alguna solicitud sobre la información.

Implementación del MVC que realiza GUUD

El marco de trabajo GUUD cuenta con un controlador frontal que inicializa los recursos básicos necesarios para usar el CodeIgniter (parte estructural del GUUD), cuando un cliente realiza una petición HTTP, este analiza y determina cuál controlador de aplicación (controlador de una determinada funcionalidad de un módulo) debe ser cargado para atender la petición realizada. Cada controlador de aplicación tiene asociada una o varias librerías y esta a su vez tiene asociado uno o varios modelos. Las librerías son responsables de procesar los datos e implementar la lógica del negocio y la clase modelo es encargada del acceso a los datos.

Luego de ser cargado el controlador de aplicaciones, este analiza la petición para determinar si necesita interactuar con la base de datos o solo cargar una vista determinada. En caso que necesite interactuar con la base de datos envía los datos recibidos a las librerías y estas cargan los modelos necesarios para obtener, registrar o actualizar en la base de datos la información solicitada. Cuando los datos son obtenidos, se retornan al controlador de aplicación en un proceso inverso al descrito anteriormente. Posteriormente, el controlador carga estos datos a archivos escritos en HTML los cuales pueden incluir llamadas a archivos escritos en JavaScript para manejar dinámicamente su contenido. Finalmente, el resultado obtenido de todo este proceso es enviado al navegador web como respuesta a la petición inicial.

La Figura 5 muestra el funcionamiento del patrón MVC en el marco de trabajo GUUD.

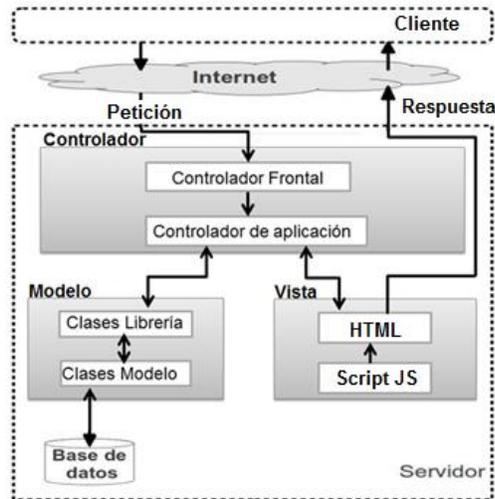


Figura 5. Patrón Modelo-Vista-Controlador en GUUD.

2.4.3. Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Cada patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema. Esta solución provee una forma confiable, segura y probada para resolver problemas recurrentes en el diseño de software. Los patrones de diseño tienen como finalidad precisar en detalle los subsistemas y componentes de la aplicación [41].

Dentro de los más conocidos se encuentran los Patrones de Asignación de Responsabilidades (GRASP) y los patrones de la Banda de los Cuatro (GoF).

Patrones para asignar responsabilidades

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Las responsabilidades están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento [41]. Para el diseño de la propuesta de solución se tuvo en cuenta los patrones GRASP: Experto, Creador, Bajo acoplamiento, Alta cohesión y Controlador. El marco de trabajo GUUD busca un máximo rendimiento y flexibilidad en sus soluciones y pone en práctica estos patrones para lograr un sistema reusable y flexible.

Experto: asigna una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Indica que la responsabilidad de la creación de un objeto debe

recaer sobre la clase que conoce toda la información necesaria para crearlo. El uso de este patrón se evidencia en las clases librerías, que son las que cuentan con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio. Un ejemplo del uso del patrón en la propuesta de solución es la librería encargada de la información de los dictámenes de la comisión, la que posee las responsabilidades relacionadas con la actualización de los dictámenes de cada uno de los activos contemplados en una solicitud de baja.

Creador: expresa la asignación a una clase la responsabilidad de crear una instancia de otra. El uso del patrón se evidencia en la clase *loader* que se encarga de cargar los elementos del marco de trabajo dígase, librerías, modelos. En la propuesta de solución cada clase controladora crea una instancia de las librerías que necesita para su funcionamiento a partir de la clase *loader*, que es la responsable de la creación (o instanciación) de nuevos objetos o clases. En la clase controladora de comisión de los dictámenes se utiliza esta clase *loader* para cargar las librerías necesarias para su correcto funcionamiento.

Controlador: define que se le debe asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. El uso del patrón se evidencia en las clases controladoras que se encargan de obtener los datos y enviarlos a las librerías y las vistas, así como de manejar los posibles errores o mensajes que se muestran. Un ejemplo de esto es la controladora de expediente de baja que recibe las peticiones realizadas a la aplicación sobre la conformación de los documentos relacionados con la formación del expediente de baja. También encontramos, la controladora solicitud de baja que recibe las peticiones que se hacen a la aplicación sobre la creación de las solicitudes de los activos que se le realizará el proceso de baja.

Alta cohesión: la cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una clase con baja cohesión hace muchas tareas no afines o un trabajo excesivo. La librería *solicitud_baja_lib.php* es ella y solo ella la única encargada de la lógica de esa parte del negocio referente a las solicitudes de baja.

Bajo acoplamiento: el acoplamiento es una medida de la fuerza con que una clase está conectada otras clases. El patrón propone que cada clase debe tener un bajo grado de dependencia con otras clases en la medida de lo posible. Un ejemplo es la librería *comision_dictamen_lib.php* que en su implementación solo contiene la lógica de su negocio y el acceso a datos. Esta clase no tiene instancia de más ninguna clase que refleje dependencias.

Patrones de la “Banda de los cuatro”

Los patrones GoF (Gang of Four), describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos que ayudan a realizar tareas complejas. Estos patrones pueden ser de tres tipos: de creación, estructurales y de comportamiento [42].

Los patrones GoF que se utilizaron en el desarrollo del componente son:

Los **patrones de creación** abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas [43].

Instancia única (Singleton): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto [44]. En la propuesta este patrón se evidencia en la clase librería `comision_dictamen_lib.php` la cual tiene en su constructor la instancia de la modelo `tb_dpersona_comision_tecnica_mdl.php`, de esta forma no existe la necesidad de instanciar esta modelo en cualquier otra parte de la clase. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Para asegurar que la clase no pueda ser instanciada nuevamente se regula el alcance del constructor.

Los **patrones de comportamiento** estudian las relaciones entre los diferentes objetos, normalmente ligados con la dimensión temporal.

Mediador (Mediator): define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto. Los tipos de clases librerías implementadas en GUUD funcionan como mediadoras entre las clases controladoras y las modelos o acceso a datos. Un ejemplo de la implementación del patrón es la librería asociada a las solicitudes de baja de los activos, que sirve de mediadora entre la modelo encargada de registrar en la base de datos, la información y la controladora de dicha información.

Observador (Observer): define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. El objeto `load` de las clases controladoras, se encarga de cargar los elementos del marco de trabajo dígame, librerías, modelos y actualiza la controladora instanciada. En la propuesta de solución en la controladora `solicitud_baja`, mediante el objeto `load`, se llama a la librería especializada para gestionar las solicitudes de bajas actualizando el estado de la controladora.

2.4.4. Diagramación de las vistas

El componente núcleo, sistema al que se integrará la propuesta de solución, cuenta con 3 tipos de vistas principales en las que se distribuyen las funcionalidades de acuerdo con su objetivo. A continuación, se describen las áreas de contenido de las vistas del sistema.

Vista de presentación: es la primera pantalla que se visualiza al acceder al sistema. Brinda la facilidad de autenticarse. Está compuesta por 2 áreas de contenido, el área 1 que incluye el logotipo de la UCI a la izquierda y el nombre del sistema a la derecha y el área 2 donde se encuentra el formulario de autenticación del sistema.

La Figura 6 detalla la vista de presentación.

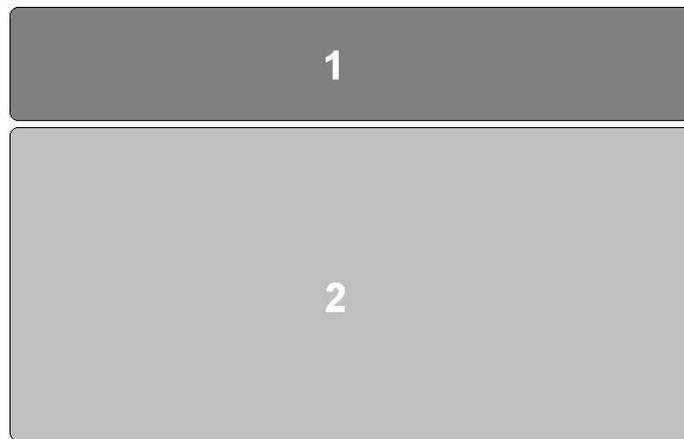


Figura 6. Áreas de la vista de presentación.

Vista de escritorio: es la pantalla mediante la que se accede a subsistemas o módulos. Está formada por 4 áreas. La primera constituye el encabezado de la página, incluye la iconografía para el acceso a los componentes del Núcleo del SGU, así como su ayuda general, el nombre del sistema, los datos del usuario autenticado y la opción de cierre de sesión. La segunda constituye el área de contenido que incluye los vínculos de acceso a los diferentes sistemas que están embebidos en el SGU. El área 3 se conforma por el menú lateral correspondiente para la vista de escritorio. La última de las áreas tiene el pie de página incluyendo logotipo de la UCI y el tipo de diseño XAUCE-SGU.

La Figura 7 detalla la vista de escritorio.

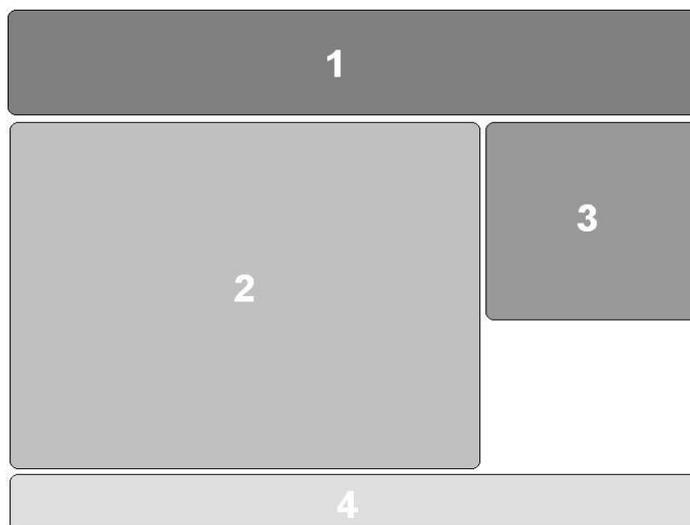


Figura 7. Áreas de la vista de escritorio.

Vista de gestión de procesos: es la pantalla que permite la gestión de la información en los diferentes componentes del sistema. La vista tiene 6 áreas de contenido basadas en la vista de escritorio donde la segunda área (3), contiene la barra de iconos flotantes para una determinada funcionalidad y el área (4) donde se realiza la gestión de dicha funcionalidad. Las demás áreas de contenido coinciden con las definidas en la vista de escritorio.

La Figura 8 detalla la vista de gestión de procesos.

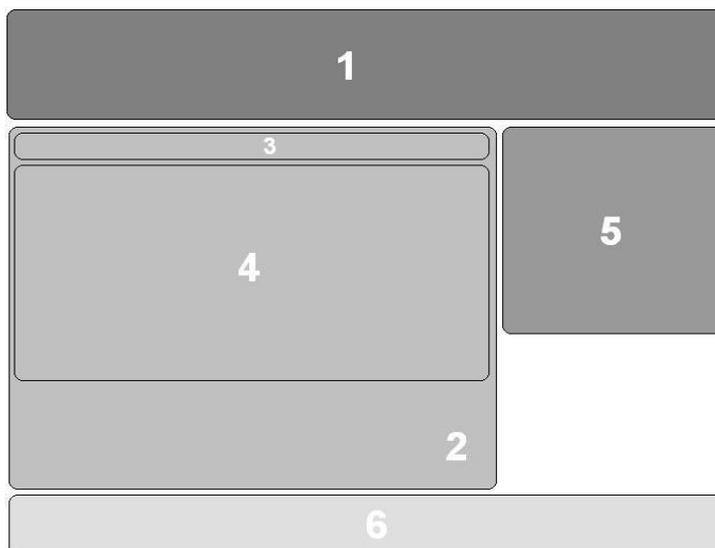


Figura 8. Áreas de la vista de gestión de procesos

2.4.5. Sistema de mensajes de la propuesta

El componente Núcleo cuenta con 2 tipos de mensajes que posibilitan la interacción con el usuario en correspondencia con las acciones realizadas en el sistema. A continuación, se describen los tipos de mensajes de la propuesta de solución.

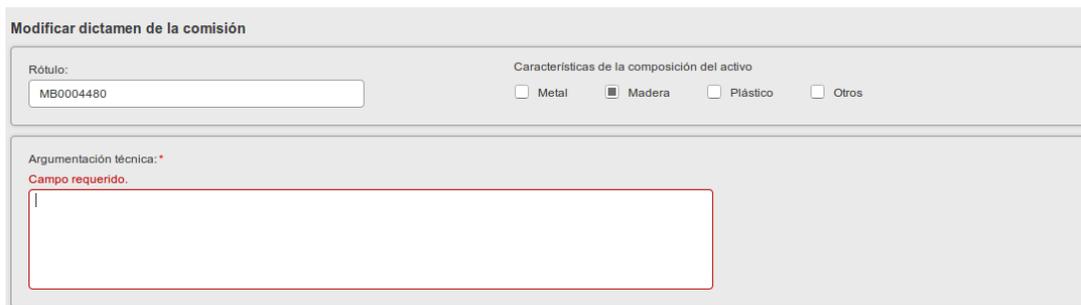
Mensajes en ventanas emergentes: representan mensajes en ventanas emergentes para informarle al usuario un determinado suceso en el sistema. Pueden ser de información y de confirmación. Los mensajes de información se muestran como resultado de una acción realizada satisfactoriamente en el sistema y los mensajes de confirmación verifican si el usuario está seguro de realizar la acción que pretende ejecutar.

Las Figuras 9 y 10 muestran los distintos tipos de mensajes en ventanas emergentes.

 <p>CONFIRMACIÓN</p> <p>¿Está seguro de realizar la acción?</p> <p>Aceptar Cancelar</p>	 <p>INFORMACIÓN</p> <p>El elemento ha sido modificado satisfactoriamente.</p> <p>Aceptar</p>
<p>Figura 9. Mensaje de confirmación en ventana emergente.</p>	<p>Figura 10. Mensaje de información en ventana emergente.</p>

Mensajes incrustados en formularios: representan mensajes de error y generalmente se muestran sobre un componente de la vista cuando falla su validación determinada.

La Figura 11 muestra un ejemplo de validación de formulario que informa al usuario el error de campos obligatorios vacíos.



Modificar dictamen de la comisión

Rótulo: MB0004480

Características de la composición del activo

Metal
 Madera
 Plástico
 Otros

Argumentación técnica: *

Campo requerido.

[Campo de texto vacío con borde rojo]

Figura 11. Mensaje incrustado en formularios.

2.4.6. Plantillas de documentos generados

La estandarización es una actividad que implica ajustar algo a un tipo de patrón o modelo, para que sea uniforme. Las normas de estandarización pueden detallar características de los modelos con la finalidad de la simplificación de los mismos.

A partir de la diversidad de formatos en los informes que se generan como parte de los procesos administrativos en la universidad, la Dirección de Informatización en conjunto con la Dirección de la universidad deciden llevar a cabo la estandarización de las plantillas, buscando uniformidad en la información centralizada.

Como parte de la solución se establece aplicar dicho proceso a todos los documentos generados por el sistema. Las Figuras 12,13,14 y 15 muestran ejemplos de documentos que se originan en un proceso de baja de AFT. Para consultar las restantes plantillas de los documentos remitirse al Anexo 5.



UCI Universidad de las Ciencias Informáticas
RECTORADO/FACULTAD/DIRECCIÓN/CENTRO DEPARTAMENTO / GRUPO DE TRABAJO

SOLICITUD PARA PROCESAR BAJA TÉCNICA DE ACTIVOS FIJOS TANGIBLES

NO. SOLICITUD: DIA: MES: AÑO:

A: DIRECCIÓN DE CONTABILIDAD Y FINANZA
Por este medio se solicita sea procesada la siguiente relación de Activos Fijos Tangibles para baja técnica, estos ya fueron dictaminados por la Comisión creada para tal efecto.

RELACION DE ACTIVOS:

Centro de Costo	Área de Responsabilidad

No.	No. Dictamen	No. de inventario	Descripción
1.			

CENTRO DE COSTO: ÁREA DE RESPONSABILIDAD:

No.	No. Dictamen	No. de inventario	Descripción
1.			

TOTAL DE ACTIVOS FIJOS TANGIBLES A PROCESAR: 0000000

SOLICITADO POR:
Nombre y Apellidos:
Cargo que ocupa:

Página 1 de 1
Universidad de las Ciencias Informáticas
Carretera 6 San Antonio de los Baños Km 2 1/2, Torrelago, Boyeros, La Habana, Cuba.
Teléfonos: +53 7 6060000 y +53 7 060 0000

Figura 12. Plantilla estandarizada de Solicitud de baja.



 FACULTAD 6

La Habana, 14 de Julio de 2016
"Año 58 de la Revolución"

A: Dirección de Contabilidad y Finanzas

Asunto: Solicitud para procesar baja técnica de AFT de la Facultad 6

Por este medio se solicita sea procesado los Activos Fijos Tangibles que se relacionan como baja técnica los cuales fueron dictaminados por la comisión creado para tal efecto.

Relación de Activos

No.	Dic.	No Inventario	Descripción
1	0981	MB00098100	SILLA GIRATORIA
2	0982	MB0106879	SILLAS GIRATORIAS C/B
3	0983	MB0080981	SILLA GR. RESP. ALTO C/B VINIL NEGRO
4	0984	MB0087360	Silla Fija S/B Tapiz Vinyl Negro
5	0985	MB0084504	Silla Fija S/B Tapiz Vinyl Negro
6	0986	MB0085536	Silla Gir. Resp. Alto C/B Vinyl Negro
7	0987	MB0000319B	Silla Fija
8	0988	MB0085458	BIURO RECTANGULAR
9	0989	MB0076547	BIURO RECTANGULAR
10	0990	MB00919319	BIURO 1200
11	0992	MB009221456	MESA ESCOLARES
12	0993	MB0092211661	MESA ESCOLARES
13	0994	MB0108587	MESA ESCOLAR 120*45*75
14	0995	MB0108078	MESA ESCOLAR 120*45*75
15	0996	MB0092211260	MESA ESCOLAR
16	0997	MB0092211600	MESA ESCOLARES
17	0998	MB0086743	ARMARIO PERSONAS VERTICALES 1800
18	0999	MB0009553	Silla fija plastica
19	1000	MB0090325	SILLA GRANDE
20	1001	MB0090310	SILLA GRANDE

Atentamente,
Dra. C. Ailee Granda Dihigo
Decana Facultad 6

Figura 13. Documento de Solicitud de baja no estandarizado.

DICTAMEN TÉCNICO DE ACTIVOS FIJOS TANGIBLES

DÍA			MES			AÑO		
0, 0			0, 0			0, 0		
INTRODUCCIÓN								
Mediante la Resolución Rectoral No: 0000/0000, la Rectora de la Universidad de las Ciencias Informáticas, dispuso designar la comisión encargada de dictaminar el estado de los muebles y proceder con las Bajas Técnicas para los Activos Fijos Tangibles no tecnológicos que procedan.								
DATOS DE LA SOLICITUD								
NO. SOLICITUD			ESTRUCTURA A LA QUE PERTENECE EL ÁREA SOLICITANTE					
CENTRO DE COSTO			ÁREA DE RESPONSABILIDAD					
DIRECTIVO SOLICITANTE			CARGO			TELÉFONO		
DATOS DEL MEDIO								
NO. INVENTARIO			NO. DICTAMEN			DESCRIPCIÓN		
Características <input type="checkbox"/> METAL <input type="checkbox"/> MADERA <input type="checkbox"/> PLÁSTICO <input type="checkbox"/> OTROS								
ARGUMENTACIÓN TÉCNICA								
<input type="checkbox"/> EL ACTIVO FIJO TANGIBLE NO ES REPARABLE Y DEBE SER DADO DE BAJA. <input type="checkbox"/> EL ACTIVO FIJO TANGIBLE ES REPARABLE Y DEBE SER GESTIONADA SU REPARACIÓN.								
Destino propuesto <input type="checkbox"/> DESTRUIR Y BOTAR. <input type="checkbox"/> MATERIAS PRIMAS.								
DISPOSICIÓN FINAL								
El AFT debe mantenerse íntegramente en el área actual donde fue dictaminado, hasta el final del proceso.								
COMISIÓN DE BAJA								
No.	Nombres y apellidos	Cargo	Firma					
1.								
2.								
3.								

Figura 14. Plantilla estandarizada de Dictamen técnico.

DICTAMEN TÉCNICO DE ACTIVOS FIJOS TANGIBLES

Mediante la Resolución Rectoral No: 52 de 2016, la Rectora de la Universidad de las Ciencias Informáticas, dispuso designar la comisión encargada de dictaminar el estado de los muebles y proceder con las Bajas Técnicas para los Activos Fijos Tangibles no tecnológicos que procedan.

DATOS DE SOLICITUD					
Estructura a la que pertenece el área solicitante.		Centro de Costo		Área de Responsabilidad	
Vicerrectoría de Tecnología		4123- Dirección de Informatización		060407 Oficina Jefe Depto de Tecnología 4to piso D-4	
Directivo que solicita		Cargo		Teléfono	
Denys Buedo Hidalgo		Vicerrector			
DATOS DEL MEDIO					
Descripción		Silla Fija de Vinil		Fecha	
Número de Inventario		MB0000730		D	M A
No. Dictamen		0793		8	06 2016
Características <input checked="" type="checkbox"/> Metal <input type="checkbox"/> Madera <input checked="" type="checkbox"/> Plástico <input type="checkbox"/> otros (Vinil)					
ARGUMENTACION TECNICA					
Dictamen del estado técnico: (Describir el estado técnico de las partes del medio) El medio está compuesto por una estructura de metal y plástico, actualmente posee la estructura con plástica partida, por lo que el medio no es posible recuperar.					
CONCLUSION DEL DICTAMEN					
<input checked="" type="checkbox"/> El Activo Fijo Tangible no es reparable y debe ser dado de baja. <input type="checkbox"/> El Activo Fijo Tangible es reparable y debe ser gestionada su reparación.					
DESTINO PROPUESTO					
<input type="checkbox"/> Destruir y Botar <input checked="" type="checkbox"/> Materias primas					
COMISION DE BAJA					
Nombre y Apellido		Cargo		Firma	
MSc. Omar Mar Cornelio		Jefe de la Comisión			
Ing. Yunior Mesa Reyes		Miembro			
Ing. Lester González López		Miembro			
DATOS DEL SOLICITANTE					
Nombre Apellido				Firma	
Denys Buedo Hidalgo					
DISPOSICION FINAL					
El AFT debe mantenerse íntegramente en el área actual donde fue dictaminado, hasta el final del proceso.					

Figura 15. Documento de Dictamen técnico no estandarizado.

2.4.7. Patrones de base de datos

Los patrones de diseño de una Base de Datos permiten al usuario crear una base de datos más fortalecida ya que constituyen una guía que especifica cómo debe ser la misma. El diseño y construcción de una base de datos requiere del mayor esfuerzo y análisis posible, ya que a partir de este diseño es que

esta se crea y la calidad con que se obtenga determinará su comportamiento futuro [45]. A continuación, se describen los patrones de diseño de base de datos que se utilizan en la propuesta de solución.

Árbol fuertemente codificado: a cada nivel de árbol se le asocia una entidad. Normalmente constituyen relaciones de 1 a muchos (n). Utilizado para representar jerarquías donde es bien conocida la estructura y es importante representar la correspondencia como son las estructuras organizacionales. Además de admitir tantos niveles como requiera la jerarquía que se vaya a representar. El patrón se utiliza en la jerarquía de la estructura de las áreas en el negocio de gestión de activos de la universidad, donde un área de responsabilidad pertenece a un centro de costo y un centro de costo puede tener varias áreas de responsabilidad asociadas a él. La Figura 16 describe una representación del uso del patrón.



Figura 16. Patrón de diseño de base de datos árbol fuertemente codificado.

Descripción de elemento: surge cuando el modelo relaciona los datos y metadatos. Es necesario definir los datos y los que son metadatos en el modelo para su utilización. En el modelo físico se manifiesta el patrón entre las entidades solicitud de baja y estado de solicitud de baja siendo el elemento y su descripción respectivamente. La Figura 17 muestra una representación del uso del patrón.

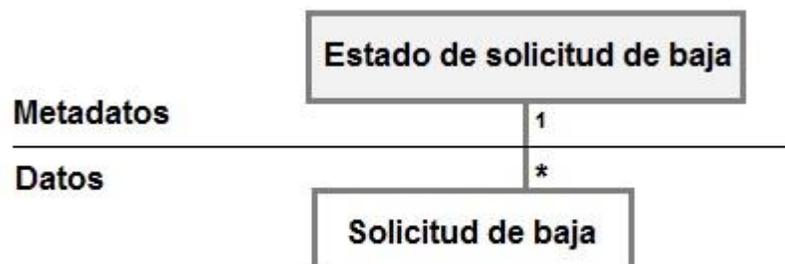


Figura 17. Patrón de diseño de base de datos descripción de elemento.

Llaves subrogadas: este patrón es muy utilizado porque facilita la interacción con la base de datos en un futuro. El mismo plantea que se genere una llave primaria única para cada entidad, en vez de usar un

atributo identificador en el contexto dado. Su uso permite que las tablas sean más fáciles de consultar a partir del identificador, pues todos tienen el mismo tipo en cada una de las tablas. La utilización del patrón está presente en la definición de la llave primaria de todas las tablas del modelo físico porque se generan llaves primarias independientes a los atributos de las entidades.

2.4.8. Modelo de datos

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre ellos y las restricciones que deben cumplirse sobre los mismos. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas y actualizaciones de datos [46].

El modelo de la base de datos consta de tres fases: diseño conceptual, lógico y físico de la base de datos. En la primera fase se obtiene una estructura de la información de la futura base de datos, que es independiente de las consideraciones físicas. Luego el modelo se refina en un esquema lógico eliminando las construcciones que no se puede representar en el modelo de base de datos escogido. En la fase de diseño físico se transforma la estructura obtenida en la etapa del diseño lógico, con el objetivo de conseguir una mayor eficiencia; además, se completa con aspectos de implementación física que dependerán del *SGBD*¹⁸. La fase de diseño físico considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente a la base de datos.

En la Figura 18 se muestra el modelo físico de la propuesta de solución.

¹⁸ **SGBD**: (sistema de gestión de base de datos) o en inglés *Database management system* (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

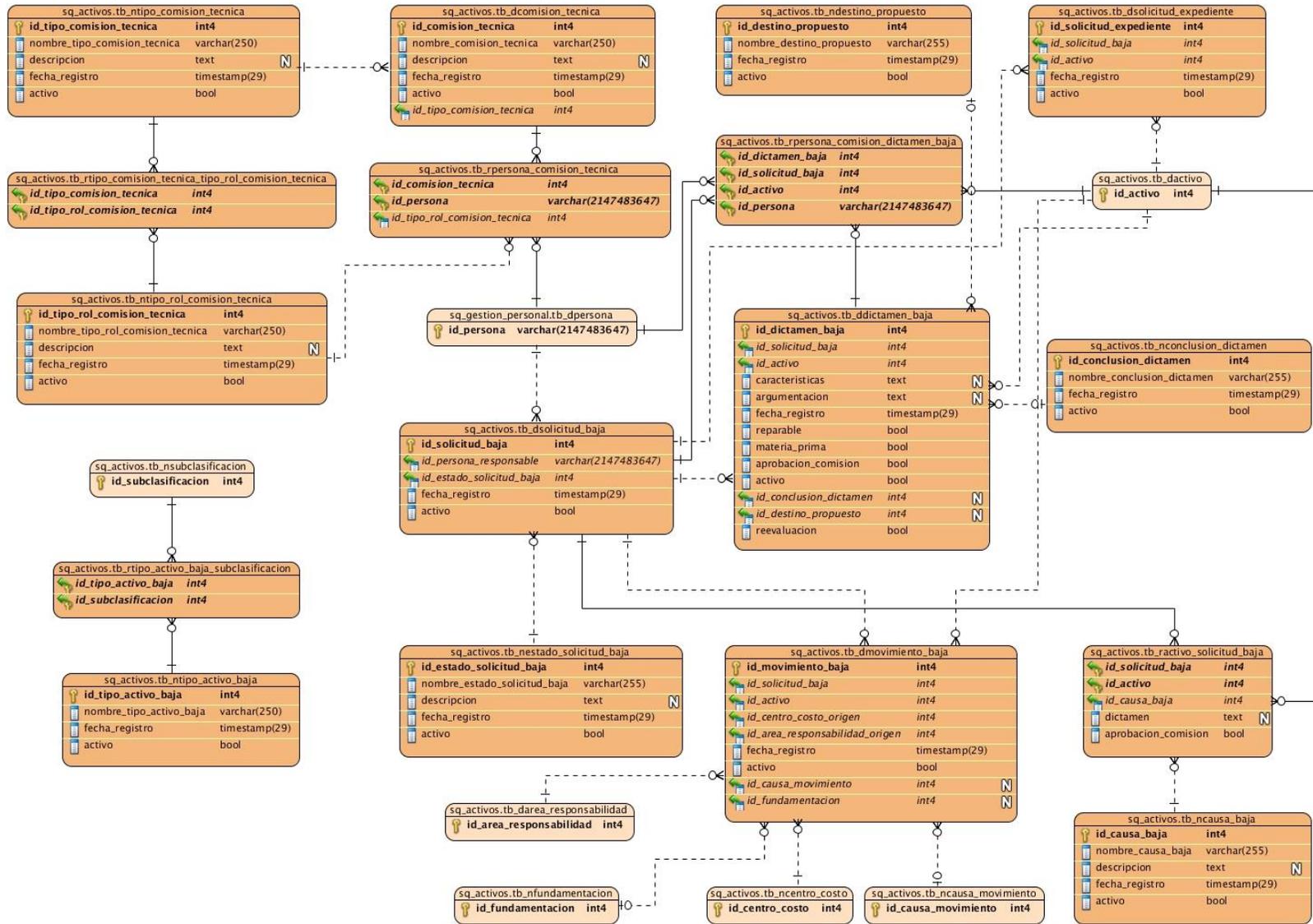


Figura 18. Modelo físico de datos.

2.4.9. Modelo de despliegue

El diagrama de despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Describe la topología del sistema: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos [47].

La Figura 19 muestra una representación de dicha estructura.

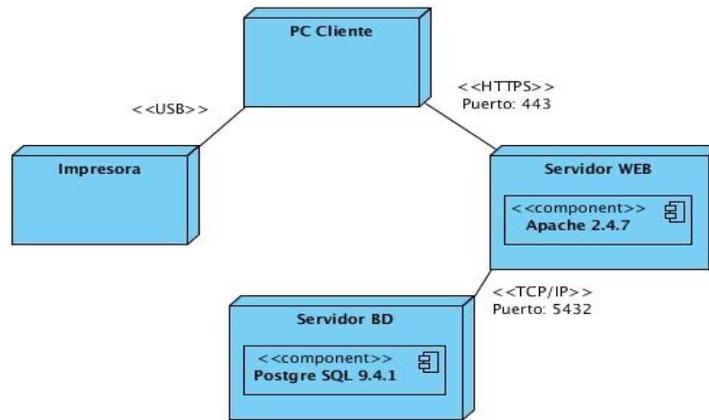


Figura 19. Representación del modelo de despliegue.

Descripción de elementos e interfaces de comunicación:

- **PC cliente:** su función es acceder e interactuar con el sistema según sus necesidades. Al estar la aplicación desarrollada sobre la web, la máquina cliente necesita de muy pocas prestaciones puesto que solo necesita un navegador web para poder acceder al sistema y realizar las operaciones necesarias.
- **Servidor aplicaciones:** aquí reside la capa de presentación del sistema, la que es accedida por las máquinas clientes a través de un navegador web. Contiene las funcionalidades del sistema.
- **Servidor de base de datos:** es el encargado de almacenar la información generada del sistema.
- **<<HTTPS>>:** el protocolo seguro de transferencia de hipertexto es un estándar de red que sigue el esquema petición-respuesta entre un cliente y un servidor. Está destinado a la transferencia segura de datos de hipertexto.
- **<<TCP/IP>>:** la familia de protocolos de Internet es un conjunto de estándares de red en la que se basa la red global y que permite la transmisión de datos entre redes de computadoras sin importar el tamaño.

2.5. Conclusiones parciales

La descripción del proceso de baja de AFT en la UCI permitió su mejor comprensión, así como definir los principales involucrados en dicho proceso, sumado a que propició un mejor entendimiento con el cliente. El estudio de las técnicas de obtención de requisitos, permitió comprender identificar y describir los requisitos funcionales y no funcionales. La utilización de patrones de diseño y arquitectura permitió diseñar una solución robusta, flexible y escalable. A partir del análisis, el diseño realizado y de los artefactos generados quedan sentadas las bases para la implementación y validación de la solución propuesta.

Capítulo 3. Construcción y validación de la propuesta de solución

En el presente capítulo se describe la implementación de la propuesta de solución descrita en el capítulo anterior, teniendo en cuenta las técnicas de programación, estrategias de codificación y la integración con otras aplicaciones. Además, se define y aplica la estrategia de pruebas del producto, especificando las evaluaciones realizadas al *software* y los resultados, para validar de esta forma que dicha solución cumpla con todos los requisitos exigidos por el cliente y posea la calidad requerida.

3.1 Estándares de codificación

Los estándares de codificación son reglas de codificación que permiten tener una programación homogénea, abarcando todos los aspectos de la generalización del código. Un código fuente completo debe estar organizado, como si un único programador hubiera escrito todo el código de una sola vez. Son de vital importancia durante la etapa de construcción del *software* ya que permite que el personal del proyecto pueda entender de forma fácil el código, garantizándose la organización y estructura del mismo [48].

Para el desarrollo de la solución se utilizan los estándares de codificación establecidos por el grupo de arquitectura del Departamento de Desarrollo de la DIN, con el propósito de estandarizar las nomenclaturas en la implementación del sistema y obtener un producto estable eficiente.

3.1.1 Indentación, llaves de apertura y cierre y tamaño de las líneas

Se usa una indentación¹⁹ sin tabulaciones, con un equivalente a 4 espacios, para mantener integridad en las revisiones. El uso de las llaves “{}” será en una nueva línea y la longitud de las líneas de código es aproximadamente de 75-80 caracteres para mantener la legibilidad del código. La Figura 20 muestra un ejemplo del uso de los estándares de codificación descritos.

```
public function index()  
{  
    echo $this->template->render('solicitud/solicitud_view');  
}
```

Figura 20. Indentación, llaves de apertura y cierre y tamaño de las líneas.

¹⁹ **Indentación:** es un anglicismo de la palabra inglesa *indentation*, de uso común en la informática. No es un término reconocido por la Real Academia Española. En los lenguajes de programación es un tipo de notación secundaria utilizado para mejorar la legibilidad del código fuente por parte de los programadores.

3.1.2 Conversión de nomenclatura

Variables: se rigen por la nomenclatura *camelCase*²⁰. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. La Figura 21 muestra el uso de la notación.

```
$params = new stdClass();
$params->activo = 't';
$centro_costo = $this->centro_costo_lib->obtenerCentrosCostoSubmayor($params);
```

Figura 21. Conversión de nomenclatura de variables.

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con guión bajo “_” y el resto en minúscula, como se muestra a continuación en la Figura 22

```
class Dictamen_comision extends MY_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->library('dictamen_comision_lib');
        $this->load->library('responsabilidad_colectiva_lib');
        $this->load->library('responsabilidad_individual_lib');
        $this->load->library('comision_dictamen_lib');
        $this->load->library('pie_firma_lib');
    }
}
```

Figura 22. Conversión de nomenclatura de clases.

Funciones: se rigen por la nomenclatura *camelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa. La Figura 23 ejemplifica el uso de la nomenclatura.

```
public function obtenerActividadesConSolicitudes($inicio = null, $limite = null, $elementoOrdenar = '', $dirOrdenar = 'asc', $params = array())
{
    $token = $this->_ci->session->userdata('token');
```

Figura 23. Conversión de nomenclatura de funciones.

Ficheros: los nombres se escriben siempre en minúscula y en caso de nombres compuestos se usa el guión bajo, seguido del sufijo definido para cada tipo de fichero.

²⁰ **camelCase:** es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en *camelCase* se asemejan a las jorbas de un camello. El término case se traduce como "caja tipográfica", que a su vez implica si una letra es mayúscula o minúscula.

- **Vista:** intuitivo y relacionado con el formulario y/o vista que representa, con el sufijo “_view”.

Ejemplo: listar_comision_dictamen_view.php

- **Modelo:** mismo nombre de la tabla de la base de datos para la cual se crea, con el sufijo “_mdl”.

Ejemplo: tb_ddictamen_baja_mdl.php

- **Librería:** mismo nombre de la clase que representa, con el sufijo “_lib”.

Ejemplo: solicitud_baja_lib.php

- **Controladora:** mismo nombre de la clase que representa.

Ejemplo: Expediente_baja.php

3.1.3 Estructura de control

Se incluye un espacio entre las estructuras de control (if, for, foreach, while, switch) y los paréntesis. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos. Si las condiciones son muy largas que sobrepasan el tamaño de la línea, estas se dividen en varias líneas. La Figura 24 ejemplifica el uso de las estructura de control en la solución.

```

if ($es_comision){
    $id_conclusion_dictamen_desconocido = (int) $this->_ci->xml->configuracion->sistema->activos->conclusion_dictamen->desconocido;
    $id_destino_propuesto_desconocido = (int) $this->_ci->xml->configuracion->sistema->activos->destino_propuesto->desconocido;
    $activos = $this->_ci->tb_ractivo_solicitud_baja_mdl->obtenerDadosAtributos(array('id_solicitud_baja' => $id_solicitud_baja));
    foreach ($activos as $activo) {
        $datos1 = array();
        $datos1['id_solicitud_baja'] = $id_solicitud_baja;
        $datos1['id_activo'] = $activo->id_activo;
        $datos1['id_conclusion_dictamen'] = $id_conclusion_dictamen_desconocido;
        $datos1['id_destino_propuesto'] = $id_destino_propuesto_desconocido;
        $this->_ci->tb_ddictamen_baja_mdl->registrar($datos1);
    }
}

```

Figura 24. Conversión de nomenclatura de estructuras de control.

3.1.4 Documentación

Todos los archivos deben de tener la documentación asociada al mismo. Cada clase y función se deben describir con el autor, una breve descripción, la categoría en caso de las clases y los parámetros en caso de las funciones. Las Figura 25 y 26 ejemplifica la documentación de una clase y una función en la solución.

```

/**
 *
 * Clase libreria solicitud_baja_lib
 *
 * Esta clase funciona como intermediaria entre la controladora y la modelo
 * @package Activos
 * @subpackage Libraries
 * @category Libraries
 *
 * @author Ariadna Falcon Novelo <ariadnafn@uci.cu>
 *
 * @property solicitud_baja $_ci
 */
class solicitud_baja_lib {

    public function __construct() {
        $this->_ci = & get_instance();
    }
  
```

Figura 25. Documentación de clases.

```

/**
 * Devuelve arreglo de las solicitudes
 * @author Ariadna Falcon Novelo <ariadnafn@uci.cu>
 * @param integer $inicio
 * @param integer $limite
 * @param string $elementoOrdenar
 * @param integer $dirOrdenar
 * @return array stdClass
 */
public function obtenerTodos($inicio = NULL, $limite = NULL, $elementoOrdenar = NULL, $dirOrdenar = NULL, $param = array()) {
    $tipo_acta_colectiva = (int) $this->_ci->xml->configuracion->sistema->activos->tipo_acta->colectiva;
  
```

Figura 26. Documentación de funciones.

Buenas prácticas

Los valores booleanos y nulos siempre se escriben con mayúscula, para facilitar la legibilidad del código. Se debe usar un salto de línea antes de las estructuras de control y definición de las funciones.

3.2 Técnicas de programación

Las técnicas de programación contribuyen a la mejora del proceso de desarrollo de software y a su mantenimiento. Tienen como objetivo principal facilitar la comprensión del programa, además permitir de forma rápida, las ampliaciones y posibles modificaciones del software.

Programación orientada a objetos (POO)

La programación orientada a objetos (OOP por sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento [49].

El marco de trabajo GUUD facilita la forma de programar en PHP y permite reutilizar el código con el paradigma de programación orientado a objetos que ya posee implementado. En él, cada objeto es

responsable de inicializarse y destruirse de forma correcta, no existe la necesidad de llamar explícitamente al procedimiento de creación o de terminación debido a que el marco de trabajo se encarga de esto. Una de las principales características de esta programación es que proporciona la herencia lo que permite la reutilización del código, por lo que GUUD la utiliza entre los distintos tipos de clases como las modelos y las controladoras. Un ejemplo del uso de la herencia en la solución, es la definición de la clase controladora Dictamen_comision.php que hereda de la clase MY_Controller.

Programación dirigida por eventos (PDE)

La programación dirigida por eventos (EDP por sus siglas en inglés) es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos provoquen. En el paradigma será el propio usuario el que dirija el flujo del programa [50].

GUUD implementa este paradigma de programación debido a que el marco de trabajo JQuery lo utiliza. Esta librería es usada para el manejo de las interfaces de usuario y utiliza los principales eventos que provee el navegador para la interacción de sus componentes. Estos eventos pueden ser desencadenados por el usuario o por otros eventos en el sistema, permitiendo una mayor interoperabilidad usuario-sistema. En la solución se ejemplifica el uso del paradigma cuando en la funcionalidad Trámite de baja técnica se cargan las solicitudes que están tramitadas para baja.

3.3 Integración con otros módulos y sistemas

El módulo Activos forma parte del Núcleo del Sistema de Gestión Universitaria, lo que permite el uso de las facilidades provistas por dicha integración. El Núcleo provee funcionalidades horizontales a las demás aplicaciones del SGU con el objetivo de centralizar la información común de los sistemas integrados. Entre los módulos del Núcleo con los que se integra Activos se encuentra:

- **Estructura y composición:** administra la información referente a la estructura administrativa y la jerarquía de la universidad. Se emplea para la obtención de la jerarquía de las áreas administrativas posibilitándole asociar personas y activos a las mismas.
- **Personal:** se ocupa de la información referente a las personas, asociándola a una responsabilidad, área, estructura, etc. En el módulo Activos, Personal posibilita el acceso a las mismas mediante criterios de búsqueda, empleados para gestionar permisos y comisiones en el sistema.
- **Seguridad:** garantiza el acceso a la información dados los niveles de privilegio de cada usuario, haciendo uso de la arquitectura sobre la que está desarrollado el sistema. En el módulo Activos, la seguridad permitiría que los jefes administrativos solo puedan acceder y tener privilegios de gestión

sobre la información de los activos que tienen asociados a su centro de costo y sus áreas de responsabilidad.

- **Trazas:** permite almacenar las incidencias de un usuario sobre el sistema. El módulo Trazas recoge de cada aplicación la información del acceso de los usuarios, acciones realizadas, fecha y hora de la ejecución, por lo que es utilizado de forma global para todos los sistemas del SGU.
- **Configuración:** la configuración del sistema es gestionada de forma centralizada facilitando un mejor control sobre los posibles cambios que pueden ocurrir en el sistema, lo que provee robustez y adaptabilidad. Dicho módulo permite gestionar la información de cada sistema, sus módulos y las funcionalidades asociadas.

La integración de la solución con las funcionalidades horizontales del Núcleo, se gestiona a través de la variable *ioc* y el esquema de base de datos *master_integracion*. Con los mecanismos de integración del SGU se puede acceder a las funcionalidades de los diferentes módulos, ya sean las que se encuentren compartidas en el fichero *ioc.xml* o las funciones de base de datos definidas en el esquema *sq_master_integracion* evitando redundancias en el código.

El sistema ASSETS, utilizado en la gestión de los procesos contables de la universidad, presenta la información actualizada de los activos pertenecientes a un centro de costo o área de responsabilidad. Su integración con el módulo Activos permite el poder asociar centros de costos a las áreas administrativas del SGU, de esta forma se actualizan diariamente todos los activos pertenecientes a las diferentes áreas. De igual forma, permite obtener diferentes datos de los activos como clasificación y subclasificación.

3.5 Estrategia de pruebas

La prueba de software es una actividad donde un sistema o uno de sus componentes se ejecutan en circunstancias previamente descritas, registrándose los resultados obtenidos. Son realizadas con el objetivo de obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos de un software.

Para evaluar la calidad del módulo que se está desarrollando y verificar el cumplimiento de los objetivos trazados, se aplicaron un conjunto de pruebas definidas por Pressman en su libro de Ingeniería del software “Un enfoque práctico”, en su quinta edición [51]. En la siguiente tabla se muestra la estrategia de pruebas diseñadas para aplicar en la solución desarrollada:

Tabla 5. Estrategia de pruebas de software.

Nivel	Tipo	Método	Técnica
Unidad	Funcional	Caja blanca	Camino básico
Integración	Funcional	Caja negra	Incremental
Sistema	Funcional	Caja negra	Particiones de equivalencia
	Rendimiento	Caja negra	Automática

3.5.1 Prueba de unidad

Las pruebas de unidad tienen como objetivo verificar la unidad más pequeña del diseño del software. Estas pruebas se concentran en la lógica del procesamiento interno y en las estructuras de datos tales como: código fuente, archivos binarios, archivos de datos, entre otros. Para la prueba se utiliza el método de caja blanca o estructural que se basa en un minucioso examen de los detalles procedimentales del código a evaluar. Existen varias técnicas para el diseño de las pruebas de unidad que permiten decidir qué sentencias o caminos se deben examinar con los casos de prueba. La técnica del camino básico se basa en obtener una medida de la complejidad del diseño procedimental de un programa. La medida de complejidad del procedimiento es la complejidad ciclomática y representa un límite inferior para el número de casos de pruebas que se deben realizar para asegurar que se ejecuta cada camino del programa [52].

Esta prueba en el sistema se realiza de forma automática. El marco de trabajo cuenta con una librería nombrada `unit_test`, especializada en la ejecución de pruebas estructurales. Aunque es bastante sencilla, cuenta con una sola función de evaluación y dos funciones de resultados, permite determinar con certeza si un código específico produce el tipo de dato y resultado esperado. Para ejecutar la prueba se utiliza la línea de código: `$this->unit->run(código, resultado esperado, 'nombre de prueba');` donde “código” es el segmento de código que se desea probar, “resultado esperado” es lo que debe devolver la evaluación del código que puede ser un tipo de dato o un valor literal y “nombre de prueba” es el nombre que se le puede dar a la prueba.

La Tabla 6 muestra el diseño de una prueba unitaria y su resultado suministrado por GUUD, los diseños restantes están accesibles en el Anexo 6.

Tabla 6. Caso de prueba unitaria de la funcionalidad Crear solicitud primaria de baja AFT.

Caso de prueba unitaria de la funcionalidad Crear solicitud primaria de baja AFT
Probador: Ariadna Falcón Novelo
Código al que se aplica:

```
public function registrar($all_post) {
    $this->_ci->db->trans_start();
    $estado_solicitud_baja_iniciada = (int) $this->_ci->xml->configuracion->sistema->activos->estado_solicitud_baja->iniciada;
    $elemento = $this->_ci->ioc->seguridad->obtenerElementosDadoToken($this->_ci->session->userdata('token'));
    $datos_registrar['id_persona_responsable'] = $elemento['id_persona'];
    $datos_registrar['id_estado_solicitud_baja'] = $estado_solicitud_baja_iniciada;
    $id_solicitud_baja = $this->_ci->tb_dsolicitud_baja_activo_md5->registrar($datos_registrar, TRUE);

    $activos = json_decode($all_post['activos_asociados']);
    foreach ($activos as $activo) {
        $datos_activo['id_solicitud_baja'] = $id_solicitud_baja;
        $datos_activo['id_activo'] = $activo->id_activo;
        $datos_activo['id_causa_baja'] = $activo->id_causa_baja;
        $this->_ci->tb_ractivo_solicitud_baja_md5->registrar($datos_activo);
    }
    $this->_ci->db->trans_complete();
    return $this->_ci->db->trans_status();
}
```

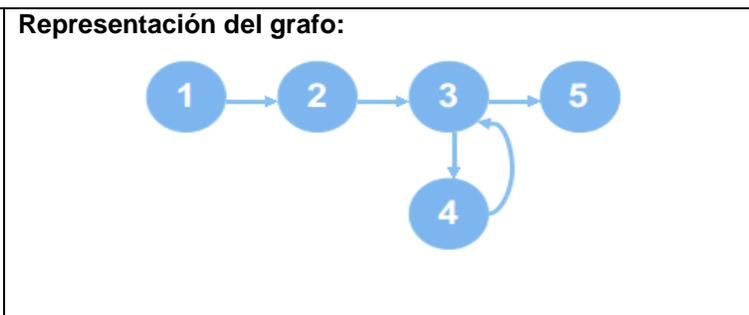
Complejidad ciclométrica:

$V(G) = (A - N) + 2$

$V(G) = (5 - 5) + 2 = 2$

Caminos independientes:

- 1 → 2 → 3 → 5
- 1 → 2 → 3 → 4 → 3 → 5



Caso de prueba para los caminos básicos

Tipo de dato esperado: Boolean

Función de evaluación:

```
public function pruebaRegistrar() {
    $all_post = array();
    $a1 = '{"id_activo": "1179589", "id_centro_costo": "277", "id_area_responsabilidad": "13792", "id_causa_baja": "1"}';
    $a2 = '{"id_activo": "1201585", "id_centro_costo": "277", "id_area_responsabilidad": "13788", "id_causa_baja": "1"}';
    $a3 = '{"id_activo": "1176194", "id_centro_costo": "277", "id_area_responsabilidad": "13787", "id_causa_baja": "1"}';
    $t = '['.$a1.', '.$a2.', '.$a3.'];
    $all_post['activos_asociados'] = $t;
    return $this->_ci->unit->run($this->registrar($all_post), true, 'Crear solicitud primaria de baja AFT');
}
```

Resultado de la prueba:

Test Name	Crear solicitud primaria de baja AFT
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	/var/www/html/sgu/base/application/sistema/activos/libraries/solicitud_baja_lib.php
Line Number	490

Evaluación del caso de prueba: Satisfactoria

3.5.2 Prueba de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura de un software a la vez que se aplican las pruebas para encontrar errores asociados a las interfaces [52]. Además, verifica sus adecuadas operaciones conjuntas.

Existen dos tipos de integración: no incremental e incremental. En la integración no incremental se combinan todos los módulos y se prueba el programa en su conjunto, lo que arroja un gran número de errores, y la consiguiente dificultad para identificar el módulo que los provocó. Por su parte, la integración incremental se fundamenta en dividir la integración en pequeños incrementos (módulos) y realizar la integración por separado. De esta forma, los errores son más fáciles de aislar y corregir.

Se empleó el método incremental ascendente que comienza la construcción y las pruebas en los niveles más bajos de la estructura y combina el siguiente que se debe probar con el conjunto que ya ha sido probado. En la validación de la solución se utiliza el método de caja negra para probar su integración con los componentes del Núcleo a los que se integra y al sistema ASSETS. La Tabla 7 muestra un caso de prueba de integración, los demás están accesibles en el Anexo 7. Las pruebas de integración realizadas con los distintos módulos con los que se relaciona Activos, arrojaron resultados satisfactorios en todos los casos.

Tabla 7. Caso de prueba de integración con el módulo Estructura y Composición.

Caso de prueba de integración con el módulo Estructura y Composición	
Componente al que se integra	Estructura y Composición.
Condiciones de ejecución	El módulo de Estructura y Composición haya introducido los datos en la base de datos central y exista conexión con la misma.
Descripción de la prueba	Comprobar que el módulo proporcione la información referente al árbol de estructura, de las áreas administrativas de la universidad.
Entradas/Pasos de ejecución	El módulo de Estructura y Composición introduce los datos en la base de datos central y en el módulo Activos se consultan los mismos y se gestionan las estructuras, asociándole los centros de costo y áreas de responsabilidades correspondientes, así como, las involucradas con las nuevas funcionalidades.
Resultado esperado	El módulo de Estructura y Composición brinda toda la información solicitada.
Evaluación	Prueba satisfactoria.

3.5.3 Pruebas de sistema

Las pruebas funcionales tienen por objetivo probar que los sistemas desarrollados cumplen con las funciones específicas para los que han sido creados. Se utiliza el método de caja negra para evaluar funcionalmente la solución donde los probadores se enfocan en el funcionamiento de la interfaz del sistema a partir del estudio de sus entradas y salidas [52].

Para confeccionar los casos de prueba de caja negra existen distintas técnicas entre las que se encuentra la técnica de particiones equivalentes. Se basa en identificar las particiones para un sistema o componente. Las particiones de equivalencia son un conjunto de datos segregados por su validez en el sistema, donde cada uno de los miembros de los conjuntos debería ser procesado equivalentemente. Por cada requerimiento funcional del sistema se generó un caso de prueba donde se recogen los datos necesarios para probarlo. La Tabla 8 y 9 describen el diseño de un caso de prueba, otro ejemplo se encuentra accesible en el Anexo 8.

En el diseño de caso de prueba hay que tener en cuenta que **V** indica Válido, **I** Inválido y **NA** que no aplica por lo que no es necesario proporcionar un valor del dato en ese caso. Las variables Buscar, Cantidad por página y Número de página significan los valores de entrada de datos para el presente caso de prueba en cada escenario; la respuesta del sistema indica su comportamiento, el flujo central y los pasos que se deben ejecutar para completar cada escenario. La condición para ejecutar la prueba es que el usuario debe estar autenticado en el sistema con el rol de administrador del sistema.

Tabla 8. Diseño de caso de prueba del requisito funcional Listar comisiones. Parte 1.

Escenario	Descripción	Buscar	Cantidad por página	Número de página
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se muestra el listado de las comisiones existentes en el sistema. En el listado se muestra un conjunto de íconos internos con las acciones a desarrollar sobre cada elemento: Detalles de la comisión, Modificar comisión y Eliminar comisión. Además se muestra un área de íconos flotantes con la acción: Crear.	NA	NA	NA
		V	NA	NA
		Comisión base 1		
EC 1.2 Buscar.	Mediante este escenario se muestra en el listado de comisiones los elementos que coincidan en total o parcial medida con los criterios de búsqueda especificados.	V	NA	NA
		Comisión		
		V	NA	NA
		Vacío		
		I	NA	NA
EC 1.3 Seleccionar correctamente la cantidad de elementos por página.	Mediante este escenario el usuario escoge la cantidad de elementos a mostrar por página (los valores a escoger son 5, 10, 15 y 20).	\$253nhg		
		NA	V	NA
			El usuario escoge la opción de 5	
		NA	V	NA
			El usuario escoge la opción de 10	
		NA	V	NA
			El usuario escoge la opción de 15	
NA	V	NA		
	El usuario escoge la opción de			

			20	
EC 1.4	Mediante este escenario el usuario selecciona una cantidad de elementos por página mayor que la cantidad de elementos que existen en el sistema.	NA	V	NA
			El usuario escoge los valores 10, 15, 20.	
EC 1.5	Mediante este escenario el usuario escribe debajo del listado de comisiones, el número de la página a la que desea acceder.	NA	NA	V
				2
EC 1.6	Mediante este escenario el usuario escribe debajo del listado de comisiones, el número de la página a la que desea acceder.	NA	NA	V
				0
EC 1.7	Mediante este escenario el usuario deja vacío el número de página que se encuentra debajo del listado de las comisiones.	NA	NA	V
				Vacío
EC 1.8	Mediante este escenario el usuario escribe debajo del listado de las comisiones, un número de página mayor que la cantidad existente.	NA	NA	I
				120
EC 1.9	Mediante este escenario el usuario escribe en el campo de texto, que se encuentra debajo del listado de las comisiones, algún carácter que sea diferente a un valor numérico (carácter extraño, letras, etc.).	NA	NA	I
				' (comillas simples) Nota: teclado en distribución español.
EC 1.10	Mediante este escenario en caso de que no exista creada ninguna comisión, se muestra un listado vacío.	NA	NA	NA

Tabla 9. Diseño de caso de prueba del requisito funcional Listar comisiones. Parte 2.

Respuesta del sistema	Flujo central
El sistema muestra el listado de las comisiones existentes.	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra las comisiones de bajas existentes.
El sistema muestra el listado de los elementos que coincidan con el nombre especificado.	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos".
El sistema muestra un listado de todos los elementos existentes.	<ul style="list-style-type: none"> -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional

<p>El sistema muestra el listado vacío porque no encontró coincidencias.</p>	<p>"Configuración" la funcionalidad "Comisión de dictamen técnico".</p> <ul style="list-style-type: none"> -El sistema muestra las comisiones de bajas existentes. -El usuario introduce los datos por los que desea filtrar la búsqueda y presiona el botón Buscar. -El sistema muestra las comisiones de baja que se corresponden.
<p>El sistema muestra la cantidad de elementos según la opción escogida (Se puede escoger 5, 10, 15 y 20).</p>	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra las comisiones de bajas existentes. -El usuario selecciona la cantidad de elementos que desea mostrar.
<p>El sistema muestra en el listado la cantidad de elementos que tiene, indicando en la parte inferior del listado el número de elementos que existen. Muestra la cantidad de elementos que están registrados, que en este caso es menor que la cantidad seleccionada.</p>	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra las comisiones de bajas existentes. -El usuario selecciona una cantidad de elementos por páginas mayor que la cantidad de elementos que existen en el sistema.
<p>El sistema muestra la página solicitada por el usuario.</p>	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra las comisiones de bajas existentes. -El usuario escribe correctamente el número de la página a la que desea acceder y presiona Enter en el teclado.
<p>El sistema muestra la página en la que se encontraba el usuario.</p>	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra las comisiones de bajas existentes. -El usuario escribe correctamente el número de la página a la

	que desea acceder y presiona Enter en el teclado.
El sistema muestra la primera página.	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra las comisiones de bajas existentes. -El usuario deja vacío el número de la página y presiona Enter en el teclado.
El sistema muestra la última página del listado.	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra las comisiones de bajas existentes. -El usuario escribe un número de página mayor que la cantidad de páginas que contienen los elementos registrados en el sistema y presiona Enter en el teclado.
El sistema muestra la primera página.	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra las comisiones de bajas existentes. -El usuario inserta en el número de la página, caracteres diferentes a un valor numérico y presiona Enter en el teclado.
El sistema muestra el listado vacío.	<ul style="list-style-type: none"> -El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona de la iconografía para el acceso a los componentes del Núcleo, el ícono correspondiente al módulo "Activos". -El sistema muestra las opciones de menú. -El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Comisión de dictamen técnico". -El sistema muestra el listado de las comisiones de bajas vacío.

Resultados de las pruebas aplicadas al sistema

Se llevaron a cabo tres iteraciones que permitieron detectar un grupo de no conformidades que fueron totalmente erradicadas. Se hallaron errores de interfaz, de ortografía y de validación. En la primera iteración fueron detectadas 20 no conformidades para un total de 36 requisitos revisados, en la segunda 7 no conformidades de 36 requerimientos. No encontrando ninguna no conformidad en la última iteración y cumpliendo con los requisitos funcionales expuestos, se demuestra la calidad de la solución implementada.

3.5.4 Pruebas de rendimiento

Las pruebas de rendimiento son realizadas luego de que el sistema está completamente integrado, con el objetivo de observar el comportamiento del mismo cuando se ve sometido a una carga que actúa de manera concurrente. También se encargan de demostrar que el sistema satisface sus requerimientos y monitorizar los comportamientos en cuanto a tiempo de respuesta de la petición.

Las pruebas de rendimiento realizadas a la solución se aplicaron con la herramienta Apache JMeter para realizar mediciones exactas y evaluar el rendimiento. Las pruebas realizadas consistieron en ensayos de 50 a 500 accesos de usuarios respectivamente. Las cantidades de usuarios seleccionadas para la prueba tienen como objetivo dar cumplimiento al requisito no funcional de eficiencia RNFAct17, que especifica que la solución soportará la conexión simultánea promedio de 50 y un máximo de 500 usuarios conectados concurrentemente.

Se definió una lista de enlaces a los que se simuló el acceso aleatorio y se recolectaron los datos necesarios para su interpretación. La Tabla 10 muestra un resumen del resultado de las pruebas y en el Anexo 9 se encuentran accesibles un resultado detallado de las pruebas de rendimiento.

Tabla 10. Resultados obtenidos de las pruebas de rendimiento.

Funcionalidad	Usuarios	Tiempo de ejecución				%	Rendimiento	
		Mínimo	Máximo	Media	Mediana		Error	Peticiones/seg.
Crear solicitud primaria de baja AFT	50	0	7700	1130	647	0.00%	39.8	529.0
	500	0	8000	2722	626	0.00%	78.5	787.7
Confirmar recepción de dictamen técnico AFT	50	98	1020	246	140	0.00%	57.8	441.6
	500	28	7738	3385	1684	0.00%	59.1	309.5
Registrar reevaluación de AFT	50	0	1417	170	111	0.00%	113.7	1947.0
	500	0	7654	1560	148	0.00%	148.4	2156.7

➤ **Usuarios:** indica la cantidad de usuarios haciendo peticiones de manera concurrente.

- **Mínimo:** indica el mínimo de tiempo de ejecución invertido para una petición.
- **Máximo:** indica el máximo de tiempo de ejecución invertido para una petición.
- **Media:** representa el tiempo de ejecución promedio de una petición.
- **Mediana:** significa que el 50% de las peticiones realizadas tardaron menos del valor reflejado.
- **% Error:** indica la relación entre el total de peticiones y el número de peticiones que originaron errores.
- **Rendimiento (Peticiones/seg.):** hace referencia al número de peticiones que el servidor puede procesar en un segundo.
- **Rendimiento (Kb/seg.):** hace referencia a la cantidad de datos que el servidor puede procesar en un segundo.

Resultados de las pruebas de rendimiento aplicadas al sistema

En la Tabla 10 se puede percibir que el tiempo de respuesta del componente no excede a los 8 segundos en la mayoría de los casos, de esta forma se valida el cumplimiento del requisito no funcional de eficiencia RNFAct18, que especifica que los tiempos de ejecución no excederán los 8 segundos de manera general. Para todas las muestras de usuarios la ocurrencia de errores se mantiene en 0.0%, lo que evidencia que las peticiones hechas se ejecutan satisfactoriamente.

3.6 Conclusiones parciales

La definición de los estándares de codificación y de las técnicas de programación empleadas para la implementación permitió describir el proceso de creación del código de la solución. El manejo de los paradigmas de programación establecidos en el marco de trabajo GUUD, facilitó establecer un método por el cual guiarse para programar y organizar el desarrollo de la solución. Mediante la integración con otros módulos del SGU, se erradicaron posibles redundancias de código. Tras efectuar todas las pruebas previstas y haber alcanzado el 100% de las pruebas con resultado satisfactorio, se concluyó que la solución cumple con las especificaciones dadas por el cliente, pues se logró los objetivos propuestos.

Conclusiones

Con la investigación realizada, el diseño y la implementación de una solución informática para la gestión del proceso de baja de activos en la UCI, se obtuvieron resultados que permiten arribar a las siguientes conclusiones:

- El estudio a diferentes sistemas que entre sus procesos cuentan con la gestión de activos, demostró que no se ajustan a las necesidades y especificaciones de la universidad.
- Con la aplicación del proceso de desarrollo de software, se logró obtener y especificar los requisitos y se describieron los procesos que conllevan las actividades relacionadas a la gestión del proceso de baja de los Activos Fijos Tangibles no tecnológicos de la UCI.
- Una vez realizadas las pruebas de software se logró garantizar el cumplimiento de los requisitos establecidos y satisfacer las necesidades del cliente.
- Con la solución informática desarrollada se logró la gestión de los procesos de baja de los Activos Fijos Tangibles no tecnológicos de la UCI, elevando la eficiencia, calidad y control del proceso en la institución.

Recomendaciones

Tras haber finalizado la investigación y desarrollado la propuesta de solución surgen ideas que podrían ser implementadas en futuras versiones del módulo. Se recomienda:

- Extender la solución para los procesos de bajas de los diferentes Activos Fijos Tangibles existentes en la universidad.
- Implementar nuevos reportes estadísticos sobre el proceso de baja de los Activos Fijos Tangibles que permitan gráficar los resultados para facilitar la toma de decisiones de los directivos de las diferentes áreas de la universidad.

Referencias bibliográficas

1. BATISTA, Anisbert Suárez, ESTRADA, Ailyn Febles AND CASAÑOLA, Yaimí Trujillo Software como servicio: necesidades y retos en los sistemas de servicio de la Industria Cubana del Software. Revista Cubana de Ciencias Informáticas, 2016, 10, 31-45.
2. SOPEÑA, Gladys Ma. Rodríguez, LAMAZARES, Margarita Méndez AND CABRERA, Janette Lozano. El control y la confiabilidad de la información contable. In *Betsime*. 2016.
3. ELIADES FUENTES ALCALDE, Lic; BECHARA, Adela Fernández; PÉREZ, Mailín. Metodología para la implementación del Sistema de Control Interno.
4. CLEMENTE, Alexis López, BELTRÁN, Dauvildo Hurtado AND GONZÁLEZ, Nancy Jiménez PROPUESTA DE ACCIONES PARA SOLUCIONAR LAS DEFICIENCIAS EN CUANTO AL CONTROL DE LOS ACTIVOS FIJOS TANGIBLES EN CUBA. Observatorio de la Economía Latinoamericana, 2012, (169).
5. Diccionario Economía - Administración - Finanzas - Marketing. [En línea] 25 de 11 de 2016. [Citado el: 25 de 11 de 2016.] <http://www.eco-finanzas.com/diccionario/A/ACTIVO.htm>.
6. Glosarios. *Glosarios*. [En línea] 25 de 11 de 2016. <http://glosarios.servidor-alicante.com/contabilidad/baja-tecnica-activos-fijos>.
7. Saez, Enrique. Minería Pan-Americana. *Sistema de gestión informática para mejorar la productividad minera*. [En línea] 26 de 11 de 2016. <http://www.minera-pa.com/productos-y-tecnologia/sistema-de-gestion-informatica-para-mejor-la-productividad-minera/>.
8. INFORMATICAHOY. *SAP R/3: Uno de los mejores ERP del mercado*. [En línea] 2016.
9. TuERP-SOFTWARE. [En línea] SAP R/3, 26 de 11 de 2016.
10. SISTEMAS DE GESTION. *Microsoft Dynamics NAV*. [En línea] 2016.
11. Cabrera González, Lic.Miguel P. XV Fórum de Ciencia y Técnica-Sistema Económico Integrado. *Versat-Sarasola*.
12. RODAS XXI. [En línea] [Citado el: 26 de 11 de 2016.] www.rodasxxi.cu.
13. SISTEMA DE GESTION INTEGRAL. *ASSETS*. [En línea] 2016. <http://www.assets.co.cu/fijos.asp>
14. Caballero, Yanisleidi González. *Sistema para auditoría y Control de Activos Fijos Tangibles*. La Habana: Universidad de las Ciencias Informáticas, 2013.

15. Pérez, Ariannis Martínez. *Propuesta de solución para la conciliación de Activos Fijos Tangibles en la Universidad de las Ciencias Informáticas*. La Habana: Universidad de las Ciencias Informáticas, 2012.
16. *Sistema de Gestión para el Control de Activos de la Universidad de las Ciencias Informáticas*. Reyes, Enmanuel Azahares. La Habana: Universidad de las Ciencias Informáticas, 2016.
17. Méndez, Alelí Sánchez. Proceso de desarrollo de software DAC. La Lisa: Universidad de las Ciencias Informáticas: Departamento de Desarrollo de la Dirección de Informatización, 2013. <https://dialnet.unirioja.es/servlet/articulo?codigo=5235925>
18. Collazo, Ing. Yasmany Tellez. *Arquitectura GUUD*. La Lisa: Universidad de las Ciencias Informáticas: Ministerio de Educación Superior, 2013.
19. PHP. [En línea] [Citado el: 30 de 11 de 2016.] <http://us3.php.net/manual/es/preface.php>.
20. Alberto Gimenez. *El lenguaje JavaScript*. [En línea] 30 de 11 de 2016. <http://albertogimenez.com/el-lenguaje-javascript/>.
21. W3C. [En línea] [Citado el: 2 de 12 de 2016.] <http://www.w3.org/html>.
22. W3C. [En línea] [Citado el: 2 de 12 de 2016.] <http://www.w3.org/TR/CSS/>.
23. Mastermagazine. [En línea] 2016. [Citado el: 2 de 12 de 2016.] <http://www.mastermagazine.info/termino/5234.php#ixzz2qa0JtGOq>.
24. OMG. [En línea] [Citado el: 1 de 6 de 2017.] <http://www.omg.org/spec/BPMN/2.0>
25. UNAD Universidad Nacional Abierta y a Distancia. [En línea] [Citado el: 2 de 12 de 2016.] http://datateca.unad.edu.co/contenidos/201493/CONTENIDO%20DIDACTICO%20EXE1/leccin_80_u ml.html.
26. W3C. [En línea] [Citado el: 2 de 12 de 2016.] <http://www.w3.org/TR/REC-xml/>.
27. NetBeans IDE Features. [En línea] [Citado el: 2 de 12 de 2016.] <https://netbeans.org/features/php/>.
28. pgAdmin. [En línea] [Citado el: 2 de 12 de 2016.] <http://www.pgadmin.org/>.
29. Pencil Project. [En línea] [Citado el: 2 de 12 de 2016.] <http://pencil.evolus.vn/wiki/devguide/Introduction.html>.
30. Visual Paradigm. [En línea] [Citado el: 5 de 12 de 2016.] <http://www.visual-paradigm.com/aboutus/newsreleases/vp122.jsp>.

31. PostgreSQL. [En línea] [Citado el: 5 de 12 de 2016.] http://www.postgresql.org/es/sobre_postgresql#intro
32. openSUSE. [En línea] [Citado el: 5 de 12 de 2016.] <https://es.opensuse.org/Apache>.
33. Apache JMeter. [En línea] [Citado el: 5 de 12 de 2016.] <http://jmeter.apache.org/>.
34. Davenport, Tomas Hill. Process innovation: reengineering work through information. Boston: s.n., 2010. ISBN: 0875843662.
35. Sommerville, Ian. Ingenieria de Software, Octava Edición. New York: Pearson Education, 2002. ISBN 978-0-321-31379-9.
36. Bizagi. Guía de usuario aplica para la Versión 10.3.0 de Bizagi BMP Suite. [En línea] 2016. [Citado el: 3 de 2 de 2017.] http://help.bizagi.com/bpmsuite/es/index.html?bienvenido_a_bizagi_bpm_suite.htm.
37. Pressman, Roger. Ingeniería de Software, un enfoque práctico. Séptima Edición. New York: McGraw-Hill Companies, 2007. ISBN 978-0-07-337597.
38. Sommerville, Ian. Ingenieria de Software. Octava Edición. New York: Pearson Education, 2002. ISBN 978-0-321-31379-9.
39. Academic. 2012. Programación dirigida por eventos. [En línea] 2017. [Citado el: 27 de marzo de 2017.] http://enciclopedia_universal.esacademic.com/72980/Programacion_dirigida_por_eventos.
40. Pressman, Roger S. Pressman_Cap_10_Disenio_Arquitectonico_Parte_1. MEXICO: s.n., 2005. ISBN: 9789701054734.
41. Pressman, Roger. Ingeniería de Software, un enfoque práctico. Séptima Edición. New York: McGraw-Hill Companies, 2007. ISBN 978-0-07-337597.
42. —. UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado, Segunda edición. s.l.: Prentice Hall, 2003. ISBN:84-205-3438-2.
43. EcuRed. Conocimiento con todos y para todos. [En línea] [Citado el:10 de 2 de 2017.] http://www.ecured.cu/Patrones_Gof.
44. Larman, Craig. UML Y PATRONES.Introducción al análisis y diseño orientado a objetos. New York: Prentice Hall. ISBN 970-17-0261-1.

45. ROMERO, G. M. P. Metodología ágil para proyectos de software libre. Junio 2008, Ciudad de La Habana 2008, no.
46. Ehow. Tres fases del diseño tradicional de bases de datos. [En línea] [Citado el: 22 de 2 de 2017.] http://www.ehowenespanol.com/tres-fases-del-diseno-tradicional-bases-datosinfo_197446/.
47. Booch, Grady. El proceso unificado de desarrollo de software. Madrid: Pearson Educación, 2000. ISBN 894-36-5624-3.
48. MICROSOFT. Revisiones de código y estándares de codificación. [En línea] [Citado el: 27 de 3 de 2017.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx..>
49. Wang., Paul S. Java con Programación orientada a Objetos. 2000.
50. Academic. 2012. Programación dirigida por eventos. [En línea] 2017. [Citado el: 27 de marzo de 2017.] http://enciclopedia_universal.esacademic.com/72980/Programacion_dirigida_por_eventos.
51. Pressman, Roger S. Ingeniería de software: Un enfoque práctico, Quinta edición. Madrid : McGraw-Hill, 2002. ISBN:84-481-3214-9.
52. —. Ingeniería de Software, un enfoque práctico, Séptima Edición. New York : McGraw-Hill Companies, 2007. ISBN 978-0-07-337597.