



Universidad de las Ciencias Informáticas

Facultad 6

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

**Sistema para Auditoría y Control de
Activos Fijos Tangibles.**

Autora: Yanisleidi González Caballero

Tutor: Lic. Omar Mar Cornelio

La Habana, 2013



La revolución no se lleva en los labios para vivir de ella, se lleva en el corazón para morir por ella.

Ernesto Che Guevara

Dedicatoria

A mi madre querida, por darme la vida, por sus sabios consejos, por estar a mi lado siempre que lo he necesitado, y darme todo su amor.

A mi abuelita querida, por dedicarme todo su tiempo y brindarme siempre todo su amor mientras estuvo presente en mi vida.

Agradecimientos

A mi tutor Omar, por su constante apoyo, paciencia, dedicación y colaboración en el desarrollo de este trabajo.

A Dinorah y a Nohemy por su ayuda incondicional y dedicarme siempre todo su tiempo.

A mis compañeros de trabajo, a Osmany por darme todo el tiempo que necesitaba.

A mi esposo Lorenzo, por brindarme todo su amor y comprensión en todo momento.

A mi familia, a mi hermano Orlendys que siempre confió en mí.

A mi madre, por el apoyo incondicional que me brindó siempre para poder salir adelante.

Declaración Jurada de Autoría

Yo, Yanisleidi González Caballero, con carné de identidad 86062604258, declaro que soy la autora del resultado que expongo en el presente trabajo titulado “Sistema para Auditoría y Control de Activos Fijos Tangibles”, optando por el título de Ingeniero en Ciencias Informáticas y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

El presente trabajo fue desarrollado en el periodo comprendido entre el mes de septiembre del 2012 y junio del 2013.

Finalmente, declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional. Para que así conste, firmo la presente declaración jurada de autoría en La Habana a 8 del mes de junio del año 2013.

Yanisleidi González Caballero

Autora

Lic. Omar Mar Cornelio

Tutor

Datos de Contactos

Tutor: Lic. Omar Mar Cornelio

Profesor Asistente. Licenciado en Educación en la Especialidad de Informática 2006. Actualmente se desempeña como Vicedecano Administrativo Facultad 6, vinculado a la docencia en el curso regular diurno en el Dpto de Programación. Certificado como especialista en calidad para revisiones de Software, Centros de Datos por empresa española EQUINSA, Diplomado como Investigador.

E-mail: omarmar@uci.cu

Resumen

Como parte del sistema de control interno, se establece como norma que el 10% de los Activos Fijos Tangible AFT sean auditados cada mes como una tarea primordial en las organizaciones. En la Facultad 6 de la Universidad de las Ciencias Informática UCI, se realiza a través de la comprobación manual de los medios contra un inventario previamente impreso, lo que genera un proceso engorroso e ineficiente con altos gastos de recursos humanos y materiales. El presente trabajo describe una solución a dicha problemática con la implementación de una aplicación Web codificada con PHP sobre Entorno de Desarrollo Integrado NetBeans, con framework de desarrollo Symfony, gestor de base de datos PostgreSQL y un sistema de adquisición de datos a través de un cliente desktop que permite obtener los medios físicos de un local mediante etiquetas en código barras. El sistema genera además un conjunto de reportes que contribuyen a elevar el control de los AFT y su proceso de identificación.

PALABRAS CLAVES: Activos Fijos Tangibles, Auditoría, Control, Códigos Barra.

Introducción	1
Capítulo 1: Fundamento Teórico	6
<i>Introducción</i>	6
<i>1.1. Activos Fijos Tangibles</i>	6
<i>1.2. Auditoría y Control</i>	7
<i>1.4. Sistemas para la Gestión de Activos Fijos Tangibles</i>	10
<i>1.5. Análisis de los Sistemas Estudiados</i>	12
<i>1.6. Estudio de la Tecnología</i>	13
<i>1.6.1. Metodologías de Desarrollo de Software</i>	13
<i>1.6.2. Lenguajes de Modelado</i>	17
<i>1.6.3. Herramientas de Modelado</i>	18
<i>1.6.4. Lenguajes de Programación</i>	19
<i>1.6.5. Framework de Desarrollo</i>	22
<i>1.6.6. Herramientas de Desarrollo</i>	25
<i>1.6.7. Sistemas Gestores de Base de Datos</i>	27
<i>Conclusiones</i>	29
Capítulo 2: Características del Sistema	30
<i>Introducción</i>	30
<i>2.1. Descripción del Sistema Propuesto</i>	30
<i>2.2. Definir Modelo del Dominio</i>	31
<i>2.3. Descripción de los Objetos</i>	31
<i>2.4. Especificación de los Requisitos del Sistema</i>	32
<i>2.4.1. Requisitos Funcionales</i>	32
<i>2.4.2. Requisitos no Funcionales</i>	33
<i>2.5. Actores y Casos de Uso del Sistema</i>	36
<i>2.5.1. Actores del Sistema</i>	36
<i>2.5.2. Casos de Uso del Sistema</i>	37
<i>2.5.3. Patrón de Casos de Uso (CRUD)</i>	38
<i>2.5.4. Diagrama de Casos de Uso del Sistema</i>	38
<i>2.5.5. Descripciones de los Casos de Uso del Sistema</i>	39
<i>Conclusiones</i>	39
Capítulo 3: Análisis y Diseño del Sistema	40
<i>Introducción</i>	40
<i>3.1. Modelo del Análisis</i>	40

Índice

3.1.1. Diagrama de Clases del Análisis.....	40
3.2. Modelo del Diseño	42
3.2.1. Patrón y Estilo de Arquitectura	42
3.3. Patrones de Diseño (GRASP).....	45
3.4 Diagramas de Clases del Diseño	46
3.5. Diagramas de Interacción	50
3.6. Diseño de la Base de Datos.....	53
3.7. Diagrama de Despliegue.....	54
Conclusiones	55
Capítulo 4: Implementación y Análisis de los Resultados.....	56
Introducción	56
4.1. Modelo de Implementación	56
4.1.1. Diagrama de Componentes	56
4.2. Pantallas del Sistema.....	58
4.3. Determinación de Posibilidad de Éxito Utilizando Criterio de Experto	61
4.4. Prueba de Interfaz de Usuario	69
Conclusiones	74
Conclusiones	75
Recomendaciones	76
Referencias Bibliográficas	77
Bibliografía	79
Glosario de Términos y Siglas.....	80
Anexos.....	81

Índice de Figura

Figura 1.1: Capas de OpenUp	16
Figura 1.2: Fases de OpenUp	16
Figura 2.1: Diagrama del Modelo de Dominio creado a partir del problema planteado.....	31
Figura 2.2: DCU del sistema definido a partir del problema planteado.....	39
Figura 3.1: Diagrama de Clase del Análisis CU Autenticar Usuario	41
Figura 3.2: Diagrama de Clase del Análisis CU Cargar Documento Excel	41
Figura 3.3: Diagrama de Clase del Análisis CU Gestionar Usuario	41
Figura 3.4: Diagrama de Clase del Análisis CU Solicitar Realizar Auditoría	42
Figura 3.5: Diagrama de Clase del Análisis CU Realizar Auditoría	42
Figura 3.6: Representación gráfica del flujo Cliente-Servidor	44
Figura 3.7: Representación del Patrón Arquitectónico Modelo Vista Controlador.....	45
Figura 3.8: Diagrama de Clase de Diseño CU Autenticar Usuario	48
Figura 3.9: Diagrama de Clase de Diseño CU Cargar Documento Excel	48
Figura 3.10: Diagrama de Clase de Diseño CU Registrar Datos de Auditoría	49
Figura 3.11: Diagrama de Clase de Diseño CU Solicitar Realizar Auditoría.....	49
Figura 3.12: Diagrama de Clase de Diseño CU Realizar Auditoría	50
Figura 3.13: Diagrama de Secuencia CU Autenticar Usuario	51
Figura 3.14: Diagrama de Secuencia CU Cargar Documento Excel	51
Figura 3.15: Diagrama de Secuencia CU Gestionar Usuario	52
Figura 3.16: Diagrama de Secuencia CU Realizar Auditoría	52
Figura 3.17: Diagrama de Secuencia CU Solicitar Realizar Auditoría	53
Figura 3.18: Representación Gráfica de la Base de Datos.....	53
Figura 3.19: Diagrama de Despliegue.....	54
Figura 4.1: Diagrama de Componentes	57
Figura 4.2: Componentes Vistas	58
Figura 4.3: Componentes Controladores	58
Figura 4.4: Componentes Modelos	58
Figura 4.5: Componentes Escritorio.....	58
Figura 4.6: Interfaz del Módulo Cargar Documento Excel	59
Figura 4.7: Interfaz del Módulo Gestionar Áreas de la Facultad	59
Figura 4.8: Interfaz del Módulo Gestionar Usuario	60
Figura 4.9: Interfaz del Módulo Solicitar Realizar Auditoría	60
Figura 4.10: Interfaz del Módulo Realizar Auditoría.....	61
Figura 4.11: Escenario Campos Vacíos	70
Figura 4.12: Escenario Campos Vacíos	71
Figura 4.13: Escenario Cargar Excel.....	71
Figura 4.14: Escenario Movimiento de Medios.....	72
Figura 4.15: Escenario Realizar Auditoría.....	73
Figura 4.16: Escenario Solicitar Datos	73

Índice de Tablas

Tabla 2.1: Descripción de las entidades que interactúan en el negocio	31
Tabla 2.2: Descripción de los actores que interactúan en el sistema	36
Tabla 2.3: Casos de Uso del Sistema organizados por orden de prioridad	37
Tabla 3.1: Representación de los estereotipos a utilizar en el diagrama de clases del análisis.....	40
Tabla 3.2: Representación de los Estereotipos a Utilizar en el Diagrama de Clases del Diseño	47
Tabla 3.3: Representación de las relaciones de clases del diseño con estereotipos Web.....	47
Tabla 4.1: Descripción de los Componentes del Sistema.....	57
Tabla 4.2: Asignación de Peso a los de Indicadores	63
Tabla 4.3: Cálculo de Concordancia de Kendall.....	63
Tabla 4.4: Escala Valorativa de 0 a 10 para Expresar el Nivel de Satisfacción	64
Tabla 4.5: Peso Otorgado por los Expertos.....	65
Tabla 4.6: Cálculo de la Dispersión (S) para hallar la Concordancia entre Expertos	65
Tabla 4.7: Cálculo de Concordancia de Kendall.....	66
Tabla 4.8: Definición del Peso Relativo de cada Criterio	67
Tabla 4.9: Calificación de cada Criterio e Índice de Aceptación	68
Tabla 4.10: Rangos Predefinidos para el Índice de Aceptación.....	68

Introducción

Las grandes evoluciones económicas y sociales de los últimos tiempos han conformado un nuevo modelo socio técnico, impulsado principalmente por un sorprendente desarrollo tecnológico. Estos eventos han provocado un obligatorio cambio en el propio contenido de la organización de la información en la gestión empresarial y su control.

El control es un plan en donde se preverán todas las medidas administrativas dentro de la entidad para el logro de los objetivos; de ahí que el control interno sea un plan de organización donde se establecen las políticas y procedimientos que persigue la entidad con el fin de salvaguardar los recursos con que cuenta. La efectividad del control interno depende de gran medida de la integridad y de los valores éticos del personal que diseña, administra y vigila el proceso de control interno de la entidad. Es de suma importancia controlar ya que es la única manera de proteger los recursos contra pérdida, fraude o ineficiencia. Además de medir el cumplimiento de la organización cerciorándose si ha habido adhesión a las políticas generales de la organización. De igual forma promover la exactitud o confiabilidad de los informes contables y administrativos.

El General de Ejército Raúl Castro Ruz, Presidente de los Consejos de Estado y de Ministros, en el Tercer Período Ordinario de Sesiones de la VII Legislatura de la Asamblea Nacional del Poder Popular, en el Palacio de Convenciones, el 1º de agosto de 2009 hizo un llamado a la sistematicidad, la organización, planificación y control efectivos en todas las esferas del país. La experiencia nos ha enseñado que aquello que no se controla con efectividad, no se cumple o se ejecuta superficialmente[1].

Teniendo en cuenta la situación económica por la que atraviesa Cuba, se hace necesario insistir en la observación de todas aquellas normas y regulaciones que permitan un mejor control de los recursos como vía de organización de los Activos Fijos Tangibles (AFT).

Debido a la importancia que se le amerita a estos, en agosto de 2009 la Asamblea Nacional del Poder Popular, crea la Contraloría General de la República, la que tiene entre sus funciones establecidas, según lo señalado en el Artículo 31 inciso I), normar, supervisar, evaluar los sistemas de control interno; manifestar las recomendaciones necesarias para su mejoramiento y perfeccionamiento continuo [2] .

El sistema de control interno establecido en la resolución 60 el 1ero de marzo del 2011 de la Contraloría General de la República, para organizar las normas y principios básicos de un modelo estándar del Sistema de Control Interno, establece que se deben auditar el 10 % de los AFT cada

mes. Esta tarea de control es necesario realizarla con un carácter inmensamente detallado. Durante esta etapa se comparan las existencias físicas con los registros contables para verificar su coincidencia y prevenir riesgos de sustracción, despilfarro, uso indebido u otras irregularidades; con el método tradicional es muy difícil que se detecten faltantes y sobrantes, lo cual es un gran problema controlar una inmensa cantidad de AFT.

La Universidad de las Ciencias Informáticas (UCI) como institución se conformó en el 2002, proyecto que se creó con el objetivo de aportar al avance económico y social del país, mediante el desarrollo productivo e investigativo en el área de las ciencias informáticas. A partir de entonces, sus propósitos se orientan a obtener su posicionamiento, tanto a nivel nacional como internacional; así como a elevar sus indicadores de salida en materia de contribuciones a la investigación, producción, comercialización de productos y servicios informáticos. Es por ello que la universidad aplica dichas normas y regulación en el Departamento de Economía.

Dada la importancia que representa el control de AFT atribuido a la tecnología en la universidad, se hace necesario establecer nuevos mecanismos que faciliten el control de AFT en la Facultad 6. Hasta el momento llevar el control de AFT es bastante complicado, pues mediante la información que brinda el departamento de economía, se conoce que la Facultad 6 cuenta con 1490 estaciones de trabajo destinadas a los procesos docentes productivos, para una aproximación de 6000 medios distribuidos en las diferentes áreas, ya sea departamentos, aulas, laboratorios y oficinas.

Toda esta situación trae consigo dificultades tales como:

- ✓ Se hace muy difícil mantener un control de los constantes cambios entre locales por razones operativas o en muchas ocasiones los movimientos generados para taller por concepto de roturas.
- ✓ Actualmente este control lo llevan a cabo las personas destinadas en cada área, que realizan la revisión con documentos impresos donde efectúan la comparación.
- ✓ En la práctica esta labor no es efectiva porque un solo local puede tener aproximadamente 130 medios y se hace muy engorroso revisarlos comparando un papel con el código que se le ha rotulado manualmente al AFT.

Dada la problemática anteriormente planteada se tiene como **problema a resolver** ¿Cómo contribuir a la gestión del proceso de auditoría y control de los Activos Fijos Tangibles de la Facultad 6 de la UCI?

La investigación tiene como **objeto de estudio** el proceso de gestión de la auditoría y control de los

Activos Fijos Tangibles.

Para la solución del problema se plantea como **objetivo general** desarrollar un sistema informático que contribuya con la gestión del proceso de auditoría y control de Activos Fijos Tangibles de la Facultad 6 de la UCI.

Objetivos específicos:

- ✓ Elaborar el marco teórico de la investigación.
- ✓ Realizar el análisis y diseño del sistema para auditoría y control de Activos Fijos Tangibles.
- ✓ Implementar sistema para auditoría y control de Activos Fijos Tangibles de la Facultad 6 de la UCI.
- ✓ Evaluar el sistema para auditoría y control de Activos Fijos Tangibles.

Se delimita como **campo de acción** los sistemas informáticos para la auditoría y control de los Activos Fijos Tangibles.

Con el propósito de optimizar el proceso de auditoría y control sobre los AFT de la Facultad 6 de la UCI se define como **hipótesis**: El desarrollo de un sistema informático para la auditoría y control de los Activos Fijos Tangibles contribuirá a mejorar la gestión de los Activos Fijos Tangibles.

Partiendo de la hipótesis planteada, se detectaron las siguientes variables independientes y dependientes:

Variable independiente: Sistema informático.

Variable dependiente: Gestión en el proceso de auditoría y control.

Para darle cumplimiento al objetivo general se definieron las siguientes **tareas de la investigación**:

- ✓ Análisis bibliográficos sobre las soluciones existentes.
- ✓ Levantamiento de los requisitos funcionales y no funcionales del sistema.
- ✓ Realizar diseño de los diagramas de clases del sistema para auditoría y control de Activos Fijos Tangibles.
- ✓ Realizar diseño de los diagramas de secuencia del sistema para auditoría y control de Activos Fijos Tangibles.
- ✓ Realizar diseño del modelo de Despliegue del sistema para auditoría y control de Activos Fijos Tangibles.

- ✓ Realizar bases de datos del sistema para auditoría y control de Activos Fijos Tangibles.
- ✓ Realizar diagrama de componentes del sistema para auditoría y control de Activos Fijos Tangibles.
- ✓ Implementación de las clases del sistema para auditoría y control de Activos Fijos Tangibles.
- ✓ Elaboración de los casos de pruebas del sistema para auditoría y control de Activos Fijos Tangibles.
- ✓ Aplicación de los casos de pruebas del sistema para auditoría y control de Activos Fijos Tangibles.
- ✓ Determinación del índice de gestión del proceso.

Para obtener los conocimientos necesarios con la finalidad de hacer posible el cumplimiento del objetivo trazado en el trabajo, se llevó a cabo una investigación en la que se utilizaron algunos de los métodos científicos existentes, como son los **teóricos** y los **empíricos**.

En el método científico de investigación **teórico** se utilizó:

Análítico-Sintético: Permite descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos de forma independiente, para luego sintetizarlos en la solución de la propuesta.

Hipotético Deductivo: Este método posibilita establecer predicciones a partir del sistema de conocimiento existente, se deduce y explican hipótesis de menor nivel de generalización.

Histórico – lógico: Permite el estudio crítico de soluciones anteriores referidas en la bibliografía científica, para extraer aspectos positivos de ellos, así como el uso de trabajos como punto de referencia para la definición de requisitos preliminares.

En el caso de los métodos científicos de investigación **empíricos** se utilizó:

Entrevistas: Permite diagnosticar el estado actual del proceso de auditoría y control. En este caso se efectuarán entrevistas al personal encargado de realizar auditorías para conocer cuáles serán los aspectos que necesitan para llevar a cabo dicho proceso. [\(Ver Anexo 1\)](#)

Observación: Permite constatar el estado de la gestión del proceso a partir de las prácticas cotidianas y la percepción directa en el panorama real. [\(Ver Anexo 2\)](#)

La **Significación Práctica** es que el conjunto de trabajadores responsables de realizar las auditorías sobre los AFT de la Facultad 6 de la UCI cuente con un sistema informático en su labor diaria para la

gestión del proceso.

El **Aporte Económico** está dado en el ahorro y control significativo de los recursos humanos y materiales. Facilita y mejora la gestión de los recursos materiales, lo que favorece a una disminución considerable en el consumo de estos.

El presente documento está estructurado en cuatro capítulos. A continuación se expone una breve descripción:

Capítulo 1: Fundamentación Teórica

Se recogen los principales conceptos que serán tratados durante el desarrollo de la investigación. Se hace referencia a las tendencias actuales, así como las tecnologías, herramientas y la metodología que se aplican durante el proceso de investigación y desarrollo del sistema.

Capítulo 2: Características del Sistema

Se describe las características del sistema a desarrollar. Se define el modelo de dominio, los requisitos funcionales y no funcionales del sistema para comprender mejor su funcionamiento, así como, los actores, diagrama de casos de uso del sistema y el patrón de casos de uso utilizado.

Capítulo 3: Análisis y Diseño del Sistema

Está relacionado con el diseño del sistema; se describen los patrones de diseño empleados. Se presentan los diagramas de clases del diseño, diagramas de interacción y diagrama de despliegue del sistema con el objetivo de tener una idea de cómo quedará.

Capítulo 4: Implementación y Análisis de los Resultados

Está enfocado a la fase de implementación para dar solución a los requerimientos funcionales identificados. Se modela los diagramas de componentes; se describen los estándares, codificación y el tratamiento de errores para darle solución al sistema propuesto.

Capítulo 1: Fundamento Teórico

Introducción

En el presente capítulo se exponen los diferentes conceptos de los cuales tratará la investigación, se muestra también un análisis sobre diferentes sistemas que permiten la gestión de AFT en el ámbito nacional e internacional que pueden ser de ayuda para el desarrollo del sistema que se desea implementar.

Además se realiza un estudio de las tendencias, tecnologías y metodologías más usadas en la actualidad, así como las herramientas a utilizar en el desarrollo del sistema para la gestión de los Activos Fijos Tangibles.

1.1. Activos Fijos Tangibles

Los activos fijos se definen como los bienes que una empresa utiliza de manera continua en el curso normal de sus operaciones; representan al conjunto de servicios que se recibirán en el futuro a lo largo de la vida útil de un bien adquirido.

Características:

- ✓ Ser físicamente tangible.
- ✓ Tener una vida útil relativamente larga (por lo menos mayor a un año o a un ciclo normal de operaciones, el que sea mayor).
- ✓ Sus beneficios deben extenderse, por lo menos, más de un año o un ciclo normal de operaciones, el que sea mayor. En este sentido, el activo fijo se distingue de otros activos (útiles de escritorio, por ejemplo) que son consumidos dentro del año o ciclo operativo de la empresa.
- ✓ Ser utilizado en la producción o comercialización de bienes y servicios, para ser alquilado a terceros, o para fines administrativos. En otras palabras, el bien existe con la intención de ser usado en las operaciones de la empresa de manera continua y no para ser destinado a la venta en el curso normal del negocio [3].

Área de Responsabilidad

Es el departamento, centro o nivel jerárquico de administración, dirección y ejecución del trabajo, organizado para intervenir directamente en la formulación de objetivos y en la comparación de los

mismos con los resultados obtenidos, con el objeto de establecer su responsabilidad en las desviaciones habidas por distintos conceptos y causas. Para el control de los AFT el área debe coincidir con el ámbito de responsabilidad o custodia de un jefe, pudiendo éste subdividirla y en cada subdivisión designar un subjefe o responsable de la custodia de los bienes ubicados en la misma. Cada trabajador, funcionario o dirigente tiene la obligación de responder por la custodia y conservación de los recursos que se le ha entregado mediante documento firmado. La responsabilidad material se fija mediante un acta que firma el responsable y quien asigna la responsabilidad, y se exige de acuerdo con lo establecido. El acta puede ser tanto individual, como colectiva. [\(Ver Anexo 3\)](#) y [\(Ver Anexo 4\)](#).

Movimientos de Medios

En cada área se controlarán los AFT ubicados en la misma, mediante un modelo de control de estos bienes, en que conste la identificación, descripción y valor de cada activo. Toda alta, baja, venta, traslado interno o remisión a reparación de un AFT, debe ampararse mediante el modelo de movimiento correspondiente, donde conste la firma del receptor y del que entregue el bien.

Todos los movimientos de AFT estarán amparados por la emisión de los documentos primarios que para cada tipo de movimiento se especifican en este procedimiento. El modelo [\(Ver Anexo 5\)](#) nos permite formalizar y dejar evidencia de todos los movimientos de los AFT que impliquen o no su movimiento físico. Estos sirven para llevar un control adecuado del AFT desde su primera ubicación hasta el destino final donde se encuentra actualmente. El modelo se distribuye de la siguiente manera:

Original: Se remite al área de contabilidad para servir de base al registro de estas operaciones, actualizar los controles y su posterior archivo.

Duplicado: Se retiene o envía por el área de responsabilidad para actualizar los controles y su posterior archivo. En el caso de traslados se remite con el activo fijo y se archiva en el área receptora.

Cuando el movimiento abarque más de un activo fijo tangible se utilizará en forma colectiva, detallándose al dorso el detalle de los bienes trasladados.

1.2. Auditoría y Control

Comprende el plan de organización, todos los métodos coordinados y medidas adoptadas dentro de una empresa para salvaguardar sus medios. El conjunto de áreas funcionales en una Empresa y de

acciones especializadas en la comunicación y control al interior de la misma. Verifica la exactitud, veracidad de sus datos contables, promueve la eficiencia de operación y fomenta la adhesión a las políticas administrativas prescritas.

Auditoría

Es el examen objetivo y sistemático de las operaciones financieras y administrativas de una entidad, practicado con posterioridad a su ejecución y para su evaluación. Revisión o análisis periódico que se efectúa a los libros de contabilidad, sistemas y mecanismos administrativos, así como a los métodos de control interno de una organización administrativa, con el objeto de emitir opiniones y recomendaciones con respecto a su funcionamiento.

Las auditorías pueden clasificarse en:

- ✓ **Auditoría externa:** es realizada por auditores totalmente ajenos a la empresa, esto permite que el auditor externo utilice su libre albedrío en la aplicación de los métodos, técnicas y herramientas con las cuales hará la evaluación de las actividades y operaciones de la empresa que audita.

Ventaja

Al no tener ninguna dependencia de la empresa, el trabajo de los auditores es totalmente independiente y libre de cualquier injerencia por parte de las autoridades de la empresa auditada.

Desventaja

Al auditor conocer poco la empresa, su evaluación puede estar limitada a la información que pueda recopilar.

- ✓ **Auditoría interna:** es realizada por un auditor que labora en la empresa donde se realiza la misma.

Ventaja

Debido a que el auditor pertenece a la empresa, casi siempre conoce integralmente sus actividades, operaciones y áreas; por lo tanto su revisión puede ser más profunda y con mayor conocimiento de las actividades, funciones y problemas de la institución.

Desventaja

Su veracidad, alcance y confiabilidad pueden ser limitados, debido a que puede haber cierta

injerencia por parte de las autoridades de la institución sobre la forma de evaluar y emitir el informe.

Objetivos generales de la auditoría

- ✓ Realizar una revisión independiente de las actividades, áreas o funciones especiales de una institución, a fin de emitir un dictamen profesional sobre la razonabilidad de sus operaciones y resultados.
- ✓ Hacer una revisión especializada, desde un punto de vista profesional y autónomo, del aspecto contable, financiero y operacional de las áreas de una empresa.
- ✓ Evaluar el cumplimiento de los planes, programas, políticas, normas y lineamientos que regulan la actuación de los empleados y funcionarios de una institución.
- ✓ Dictaminar de manera profesional e independiente sobre los resultados obtenidos por una empresa y sus áreas, así como sobre el desarrollo de sus funciones y el cumplimiento de sus objetivos y operaciones [4].

Control

Es un mecanismo preventivo y correctivo adoptado por la administración de una dependencia o entidad que permite la oportuna detección y corrección de desviaciones, ineficiencias o incongruencias en el curso de la formulación, instrumentación, ejecución y evaluación de las acciones, con el propósito de procurar el cumplimiento de la normatividad que las rige, las estrategias, políticas, objetivos, metas y asignación de recursos.

Objetivos generales del control

Atendiendo los principios constitucionales que debe caracterizar la administración pública, el diseño y el desarrollo del Sistema de Control Interno se orientarán al logro de los siguientes objetivos fundamentales:

- ✓ Proteger los recursos de la organización, buscando su adecuada administración ante posibles riesgos que los afecten.
- ✓ Garantizar la eficacia, la eficiencia y economía en todas las operaciones. Promover, facilitar la ejecución de las funciones y actividades definidas para el logro de la misión institucional.
- ✓ Velar porque todas las actividades y recursos de la organización estén dirigidos al cumplimiento de los objetivos de la entidad.
- ✓ Garantizar la correcta evaluación y seguimiento de la gestión organizacional.
- ✓ Asegurar la oportunidad, confiabilidad de la información y de sus registros.

- ✓ Definir, aplicar medidas para prevenir los riesgos, detectar y corregir las desviaciones que se presenten en la organización y que puedan afectar el logro de sus objetivos.
- ✓ Garantizar que el sistema de control disponga de sus propios mecanismos de verificación y evaluación.
- ✓ Velar porque la entidad disponga de procesos de planeación y mecanismos adecuados para el diseño y desarrollo organizacional [5].

1.4. Sistemas para la Gestión de Activos Fijos Tangibles

Llevar el control de los AFT de cualquier institución es de gran importancia para garantizar su subsistencia. A continuación se muestra un estudio realizado a sistemas utilizados para el control de AFT que puede ser de ayuda para la implementación de la solución propuesta.

Sistema Automatizado de Control de los Activos Fijos Tangibles para el Sector de Salud (MEBUS)

El MEBUS forma parte del Proyecto SEUS (Sistema Económico para Unidades de Salud) que se está empleando en las entidades del sector de la Salud. Permite obtener una serie de informes significativos para el control de los Activos Fijos Tangibles y el proceso de informaciones esenciales que se brinda a niveles superiores. Encierra la información de los Activos Fijos Tangibles, ofrece la desvalorización por cada medio, brinda la posibilidad de revalorizar el medio cuya depreciación haya caducado y, en caso de no revalorizar, admite que el medio no siga depreciándose, informa cuándo la depreciación de cada medio básico este alcanzando su fin y posee una base de datos pasiva con los medios básico que se le haya dado de baja por 5 años. Las informaciones que se han procesado y son precisas para otros subsistemas (contabilidad general, costos, cobros y pagos, así como presupuesto), se envían de forma automática, mediante la red o vía correo electrónico [6].

VERSAT Sarasola

La aplicación VERSAT Sarasola cuenta una serie de funcionalidades que permite la planificación y control de los recursos humanos, materiales y financieros de cualquier entidad, tanto del sistema empresarial como presupuestado. Asimismo posee una versión educativa para su utilización en escuelas de comercio y en universidades. Se comenzó a trabajar en el sistema en julio de 2001 en el complejo agro industrial George Washington, del municipio de Santo Domingo, en la provincia de Villa Clara. El sistema informático VERSAT posee 12 módulos o subsistemas de configuración, pero aún le faltan varios subsistemas por desarrollar completamente. Otra virtud de esta aplicación es que fue el primer sistema de este tipo realizado en la isla certificado por los ministerios de Finanzas y Precios y

de Informática y las Comunicaciones. La Contraloría General de la República de Cuba, solicitó que se le adjunte un subsistema para posibilitar a los auditores la observación de las operaciones contables y financieras de las entidades en las cuales se hace uso de este [7].

Desventajas:

- ✓ Los datos introducidos en el programa pueden ser modificados o eliminados por cualquier persona con acceso al sistema.
- ✓ Antes que hacer el cierre de cuentas nominales hay que hacer manualmente el asiento correspondiente de existencias iniciales y finales a variación de existencias.
- ✓ Faltan subsistemas, pues sólo cuenta con facturación, tesorería y utilidad, faltaría el subsistema de inventario.

Sistema Computarizado para el Manejo de Inventario y Control de Pedidos en una Industria de Elaboración de Perfume

Este sistema se desarrolló en la Universidad de San Carlos en Guatemala en el año 2004 como un trabajo de graduación. Este software trata de optimizar los recursos con los que un trabajador cuenta a la hora de hacer las labores, a la vez que se reducen tiempo y costos.

El sistema computarizado de control de inventario logra dar la información de un modo rápido y exacto al haber definido la forma de controlar tangiblemente las mercancías en la bodega y de ahí localizarlo fácilmente en la aplicación. La organización del sistema permite controlar la cantidad en existencia de cierto producto, así como su localización física a través de los diferentes estantes mediante el código. Se logra disminuir el porcentaje de error en la realización de pedidos, ya que éste calcula automáticamente las cargas y descargas del producto, obteniendo un valor más exacto.

Sistema de Gestión Integral (ASSETS)

Este es el sistema que se utiliza actualmente en la UCI, permite controlar las operaciones de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos. Fue desarrollado en Visual Basic 6.0 y Microsoft SQL Server 2000, utilizando adicionalmente Crystal Reports 7.0 para la generación de reportes. El Módulo de Inventario permite observar las existencias, disponibilidad y reserva de los diferentes elementos en múltiples almacenes al realizar conteos físicos de los productos y hacer ajustes tanto de existencia, como de precios de costo de los artículos almacenados.

Como Sistema Integral todos sus módulos trabajan en estrecha relación, generando automáticamente al módulo de contabilidad los comprobantes de operaciones por cada una de las transacciones efectuadas, esto permite que se pueda trabajar bajo el principio de contabilidad al día. Es flexible, amigable, con ayuda en línea que puede ser instalado en una microcomputadora o sobre varias, funcionando en ambiente multiusuario incluidas estaciones remotas. Asimismo, proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario. En ASSETS se facilita el uso de la parametrización para adaptarse a las exigencias de cada entidad en particular, garantizando que sus reportes tengan la forma y el contenido que el usuario les defina.

1.5. Análisis de los Sistemas Estudiados

Después de realizar un análisis de los sistemas expuestos con anterioridad, se puede decir que en su generalidad no cumplen con la necesidad que se plantea en la problemática. Es decir, la información que gestionan estos sistemas, no coincide con la información que se desea controlar y no manejan los mismos tipos de datos en cada uno de ellos.

El MEBUS gestiona los medios básicos de las unidades de salud que más bien se encarga de realizar diferentes informes de los AFT en cuanto a su depreciación; la aplicación VERSAT Sarasola cuenta una serie de funcionalidades que permiten la planificación y control de los recursos humanos, materiales y financieros de cualquier entidad, aun así faltan subsistemas, pues sólo cuenta con facturación, tesorería y utilidad, faltaría el subsistema de inventario, en este caso se analizó la posibilidad de incluir los módulos a realizar pero no se llevó a cabo puesto que la aplicación a pesar de ser desarrollada en Cuba posee licencia por tiempo determinado, lo cual incluiría un gasto para quien desee utilizarla. El Sistema Computarizado para el Manejo de Inventario y Control de Pedidos en una Industria de Elaboración de Perfume se enfoca más bien en optimizar los recursos con los que un trabajador cuenta a la hora de hacer las labores. El Sistema de Gestión Integral (ASSETS), contiene varios módulos bien definidos, pero no es adaptable a la necesidad presente en la Facultad 6 de la UCI, pues con este sistema no se puede realizar auditorías.

En su mayoría no cumplen con la problemática planteada, si bien unos realizan la gestión de los AFT de manera eficiente, la forma en que los controla no es la más adaptable para el trabajador que realiza las auditorías. Tanto en unos como en otros hay que verificar la existencia de los AFT de forma manual, revisando el rótulo de cada medio previamente escrito, contra la lista del local donde deben estar. Algunos de los sistemas son del tipo de programas instaladores lo que equivale a pérdida en tiempo y recursos en la labor de despliegue al tener que instalarlo en cada una de las PC

donde se desea utilizar.

1.6. Estudio de la Tecnología

Para desarrollar el sistema toda la tecnología fue seleccionada cautelosamente, ya que pueden llevar al fracaso o el éxito de éste, para lo cual se estudiaron las diferentes opciones y necesidades del proyecto. El principal elemento que se tuvo en cuenta fue que las herramientas debían ser compatibles.

A continuación se describen las herramientas seleccionadas:

1.6.1. Metodologías de Desarrollo de Software

Actualmente el desarrollo de software ha alcanzado un alto nivel debido a la competencia que existe, por lo que los desarrolladores se han visto en la necesidad de buscar técnicas mediante las cuales se logren estandarizar el trabajo de las aplicaciones que se desarrollan. Es por ello que se ha creado un conjunto de metodologías para el desarrollo del software; que hace posible que todo el personal de un proyecto se vincule y pueda entenderse.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayuda a la documentación para el desarrollo de productos software, en la que se indican, paso a paso, todas las actividades a realizar para lograr informatizar el proceso deseado; muestran qué personas deben participar en el desarrollo de las actividades y el papel que desempeñan. Además, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Las metodologías se pueden agrupar en tradicionales o formales y ágiles. Las tradicionales centran su atención principalmente en desarrollar una documentación completa de los proyectos, así como el cumplimiento de un plan de proyecto, el cual debe ser definido al comienzo del mismo. Mientras que para las metodologías ágiles es de vital importancia la respuesta a los cambios y mantener una buena relación con el cliente.

Existen diversos tipos de metodologías a continuación se nombran las más utilizadas, como por ejemplo: RUP, XP, Scrum, OpenUp y MSF. La selección depende de cuál se adapta más a lo que se quiere desarrollar.

Programación Extrema (XP)

XP es una de las metodologías de desarrollo de software más exitosas. Se utiliza para proyectos de corto plazo y equipo de desarrollo pequeño. Es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software; promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define especialmente para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Se considera como una adopción de las mejores metodologías de desarrollo a partir de lo que se pretende realizar con el proyecto, y de manera dinámica aplicarlo durante el ciclo de vida del software.

SCRUM

Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza ejecutable que incorpora una nueva funcionalidad. Es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

El equipo, se enfoca solamente en construir software de calidad. La gestión de proyecto se focaliza en definir las características del software a construir y remover cualquier obstáculo que pudiera entorpecer el trabajo del equipo de desarrollo. Se busca que los equipos de desarrollo, sean lo más productivos posible.

Se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos y donde la innovación, la competitividad y la productividad son fundamentales.

SCRUM también se utiliza para resolver situaciones en que no se entrega al cliente lo que necesita, cuando las entregas se alargan demasiado, los costos se disparan o la calidad no es aceptable; cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar con un proceso especializado en el desarrollo del producto.

OpenUp

Open Up es un proceso modelo, extensible, dirigido a gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños o de bajos recursos, aplicable a un conjunto amplio de plataformas de desarrollo. Basada en la metodología de Proceso Unificado de Rational (Rational Unified Process) RUP. Es el subconjunto de esta última que contiene el conjunto mínimo de prácticas que ayudan a un equipo de desarrollo de software a realizar un producto de alta calidad. Utiliza un punto de vista pragmático, una filosofía ágil que se centraliza en la naturaleza colaborativa del proceso de desarrollo del software. Una de sus principales características es su alto grado de adaptabilidad a las necesidades de un proyecto en particular.

OpenUp es un proceso de desarrollo iterativo del software que es mínimo, completo y extensible. Es mínimo en que solamente el contenido fundamental es incluido; es completo en que puede ser manifestado como todo el proceso para construir un sistema; extensible en que puede ser utilizado como fundamento sobre el cual el contenido de proceso se pueda agregar o adaptar según lo necesitado [8].

Como metodología de desarrollo es conducida por el principio de colaboración para alinear intereses y para compartir su comprensión. Es el proceso unificado que aplica acercamientos iterativos e incrementales dentro de un ciclo vital estructurado [9]. En la Figura 1.1 se muestran las capas de OpenUp.

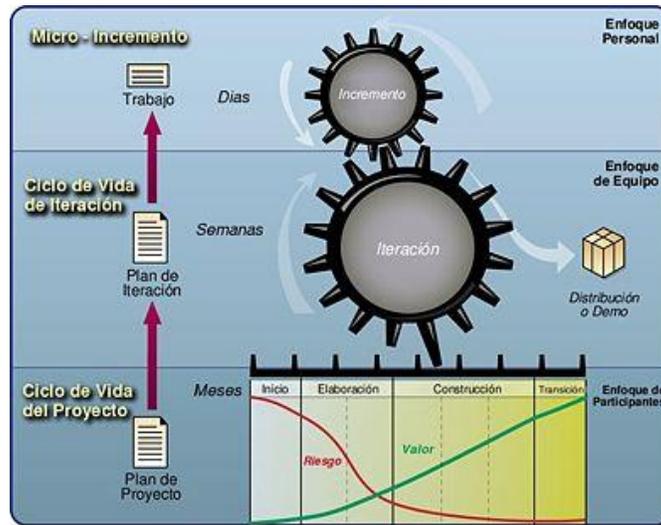


Figura 1.1: Capas de OpenUp

El OpenUp estructura el ciclo de vida de un proyecto en cuatro fases: inicio, elaboración, construcción y transición. El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y dirección para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos. A continuación en la Figura 1.2 se muestran las fases de OpenUp.

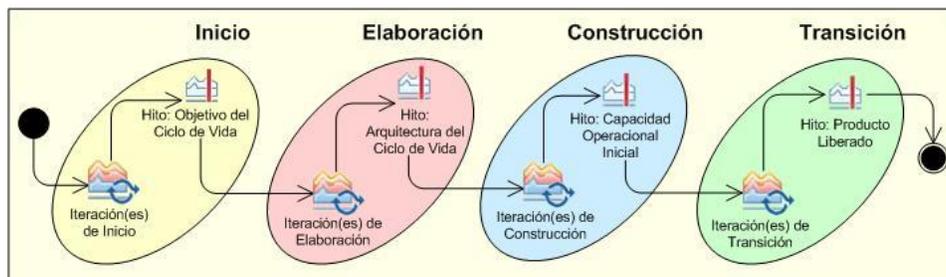


Figura 1.2: Fases de OpenUp

Después del análisis de las metodologías anteriormente mencionadas se decidió escoger para el proceso de desarrollo OpenUp, ya que es iterativo e incremental. Es apropiado para proyectos pequeños y de bajos recursos. Permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito. Permite detectar errores tempranos a través de un ciclo iterativo. Evita la elaboración de documentación, diagramas e iteraciones innecesarias. Por ser una metodología ágil tiene un enfoque centrado al cliente con iteraciones cortas. Es ligero y proporciona una comprensión detallada del proyecto, beneficia a clientes y desarrolladores sobre el producto a entregar y su formalidad. Es extensible ya que el proceso se puede agregar o adaptar según lo vayan requiriendo los sistemas.

1.6.2. Lenguajes de Modelado

El lenguaje de modelación es independiente del lenguaje de programación que se desea emplear en el desarrollo del software. Con el transcurso de los años se ha producido un auge en la Programación Orientada a Objeto (POO), lo que ha conllevado a la creación de diferentes lenguajes de modelado orientado a objetos.

Lenguaje de modelado OO i*

La notación i* fue creada en la primera mitad de la década de los 90. Permite expresar de forma clara y sencilla los objetivos de los actores que aparecen en los modelos y la dependencia entre ellos. Consta con una notación gráfica que permite tener una visión intuitiva y unificada del entorno modelado que muestra tales actores y dependencias.

Tiene la desventaja de no tener una definición única del lenguaje. Además, las definiciones existentes no son tan claras como se desearía ya que contienen ambigüedades y contradicciones.

Lenguaje Unificado de Modelado UML

Es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software[10].

Lenguaje Unificado de Modelado (Unified Modeling Language) UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el (Object Management Group) OMG. Incluye aspectos conceptuales tales como procesos de negocio, funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Es un lenguaje gráfico para:

- ✓ Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
- ✓ Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Se utiliza para detallar los artefactos en el sistema y para documentar y construir. UML intenta

solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos, se crea una documentación también común, que cualquiera con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado.

Después del estudio realizado a los diferentes lenguajes de modelado se propone usar UML en su versión 2.1, pues proporciona a los desarrolladores un mejor entendimiento y asimilación del proyecto, minimiza el tiempo invertido en el desarrollo de la arquitectura, además agiliza la detección y resolución de errores.

1.6.3. Herramientas de Modelado

Las herramientas de modelado o CASE se pueden definir como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Rational Rose

Rational Rose es una herramienta de modelado visual con plataforma independiente, que permite la comunicación entre los miembros del equipo. Tiene la ventaja que utiliza la notación estándar en la arquitectura de software (UML), posibilita a los arquitectos y desarrolladores de software visualizar todo el sistema mediante el uso de un lenguaje común. Mantiene la consistencia de los modelos de sistemas de software.

Permite chequear las sintaxis UML y realizar ingeniería inversa. Posibilita la generación de documentación automáticamente y la generación de código a partir de los modelos. A través de él los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con los demás componentes del proyecto.

Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: Análisis y Diseño orientados a objetos, Construcción, Pruebas y Despliegue.

El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML[10].

Posibilita la representación gráfica de los diagramas. Esto propicia ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información. Se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo a través de un lenguaje gráfico. Tiene disponible, distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Facilita licencias especiales para fines académicos.

Entre sus características más significativas se puede resaltar:

- ✓ Su licencia es gratuita y comercial.
- ✓ Posee varios idiomas.
- ✓ Sus ediciones son compatibles.
- ✓ Es de fácil uso para la creación de aplicaciones Web.
- ✓ Disponibilidad en múltiples plataformas (Windows, Linux).
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Fácil de instalar y actualizar.
- ✓ Soporte de UML versión 2.1.

Después del análisis realizado a las diferentes herramientas de modelado se escogió Visual Paradigm UML en su versión 8.0. La misma soporta el ciclo de vida completo del desarrollo de software y es de fácil uso para la creación de aplicaciones Web.

1.6.4. Lenguajes de Programación

En la actualidad existen diferentes lenguajes de programación, estos han surgido debido a las tendencias y necesidades de las plataformas existentes.

Desde el comienzo de Internet, surgieron diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. Con el transcurso del tiempo, las tecnologías se desarrollaron y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la Web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de bases de datos.

ASP.NET

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML. Forma parte de la plataforma .NET y es la tecnología sucesora de Active Server Pages (ASP).

ASP.NET es una nueva estructura de programación y fue diseñado con el objetivo de que dichas aplicaciones respondan rápidamente a las solicitudes de los usuarios, sin importar la cantidad de datos que se estén procesando en el servidor, debido a que en una página ASP se pueden incluir: HTML plano, código de scripting y texto; no hay una distinción formal entre el contenido de una página y su comportamiento. Impone un cierto orden sobre el modelo de programación estándar ASP y proporciona diversas mejoras en las cuales se destacan:

- ✓ **Rendimiento:** se compila desde el código nativo, lo que permite mucho mejor rendimiento y un almacenamiento de la caché en el servidor.
- ✓ **Rapidez en programación:** mediante diversos controles, se logra con unas pocas líneas y en menos de 5 (cinco) minutos, mostrar toda una base de datos y hacer rutinas complejas.
- ✓ **Servicios Web:** posee herramientas para compartir datos e información entre distintos sitios.
- ✓ **Seguridad:** posee diversas herramientas que garantizan la seguridad de nuestras aplicaciones[11].

Java

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido. Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma, esto en alguna medida es una ventaja significativa para los desarrolladores de software pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux y Apple.

Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente. Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador, como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria[12].

PHP

Es un lenguaje de programación del lado del servidor, tiene como característica que es fácil de aprender, y está pensado para desarrollar páginas Web eficientes. Se utiliza para escribir el contenido dinámico de las páginas. Soportado por la mayoría de los servidores Web entre ellos Apache. Se utiliza en cualquier sistema operativo. Permite manipular la información almacenada en la base de datos como PostgreSQL [13].

Es gratuito e independiente de plataforma, rápido, con una amplia librería de funciones y mucha documentación. Fue creado en 1994 por Rasmus Lerdorf.

Es un lenguaje orientado a objeto, lo que permite la reutilización de código y un rápido desarrollo. Posee compatibilidad con las bases de datos más comunes, como MySQL, Oracle y PostgreSQL, entre otros. Proporciona soporte para diferentes protocolos de comunicación conocidos entre los cuales se tienen:

- ✓ HTTP (Protocolo de Transferencia de Hipertexto).
- ✓ IMAP (Protocolo de Acceso a Mensajes de Internet).
- ✓ FTP (Protocolo de Transferencia de Archivos).
- ✓ LDAP (Protocolo Ligero de Acceso a Directorios), entre otros.

Es publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Csharp

C# (Csharp) es un lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes.

Es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. Su compilador es el más depurado y optimizado de los incluidos en el *.NET Framework*.

Entre sus características más significativas se puede resaltar:

- ✓ **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.

- ✓ **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular.
- ✓ **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código. C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo[14].

Después de analizar los lenguajes de programación anteriormente mencionados, para la aplicación Web se escogió PHP en su versión 5.2.3, pues es un lenguaje de los más utilizados en el mundo del desarrollo Web, presenta una amplia documentación, es gratuito e independiente de la plataforma donde se quiera utilizar.

Para la aplicación de escritorio se escogió Csharp porque es un lenguaje orientado a objetos simple, moderno y de propósito general. Es adecuado para escribir aplicaciones de cualquier tamaño, desde las más grandes y sofisticadas como sistemas operativos, hasta las más pequeñas funciones.

1.6.5. Framework de Desarrollo

Un framework de desarrollo es una estructura de trabajo y soporte definida, mediante la cual otro software puede utilizar para llevar un trabajo más organizado y desarrollado en la realización de software. Con los framework de desarrollo, se pueden encontrar soportes de programas, bibliotecas, un lenguaje interpretado, entre otros software, para facilitar el desarrollo, unir las diferentes estructuras y componentes de un proyecto.

Symfony

Es un framework para aplicaciones Web desarrollado en PHP5 por Fabien Potencier y patrocinado por los laboratorios de Sensio, Agencia francesa Web muy conocida por sus puntos de vista innovador en el desarrollo Web. Acelera la creación, mantenimiento de las aplicaciones Web, y permite la reutilización de código, lo cual agiliza el desarrollo de la aplicación.

Su diseño permite optimizar las aplicaciones y mantener una organización de la misma. Permite reducir el tiempo de desarrollo de una aplicación pues proporciona varias herramientas y clases las cuales pueden ser reutilizadas por los desarrolladores. Permite el desarrollo por capas, ya que se

basa en el patrón arquitectónico Modelo Vista Controlador (MVC). Es compatible con la mayoría de los gestores de base de datos como PostgreSQL [13].

Symfony fue diseñado para ajustarse a los siguientes requisitos:

- ✓ Fácil de instalar y configurar en la mayoría de plataformas.
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Utiliza programación orientada a objetos.
- ✓ Sencillo de usar en la mayoría de casos.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Fácil de extender.

Kumbia

Es un framework para desarrollo de aplicaciones Web escrito en PHP5. Se basa en las prácticas de desarrollo Web como “Don't Repeat Yourself” (DRY) y el Principio “Keep It Simple, Stupid” (KISS). Fomenta la eficiencia y velocidad en la creación y mantenimiento de aplicaciones Web mediante el remplazo de códigos repetitivos. Intenta proporcionar un medio fácil para el desarrollo de aplicaciones robustas para entornos educativos y comerciales, gracias a que es flexible y configurable.

.Net

El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables.

.NET Framework se compone de cuatro partes, el entorno común de ejecución, un conjunto de bibliotecas de clases, un grupo de lenguajes de programación y el entorno ASP.NET. El Framework .NET fue diseñado con tres objetivos en mente. Primero, debía lograr aplicaciones Windows mucho más estables, aunque también debía proporcionar una aplicación con un mayor grado de seguridad. En segundo lugar, debía simplificar el desarrollo de aplicaciones y servicios Web que no sólo funcionen en plataformas tradicionales, sino también en dispositivos móviles. Por último, el entorno fue diseñado para proporcionar un solo grupo de bibliotecas que pudieran trabajar con varios

lenguajes.

.NET Framework se creó pensando en la seguridad, lo que se refleja en cada aspecto del entorno. El marco de trabajo .NET es capaz de ejecutarse de manera remota, realizar descargas dinámicas de nuevos componentes e incluso de ejecución dinámica. Con este tipo de entorno, si un programador debe crear el modelo de seguridad, probablemente tarde más en codificarlo que en crear el propio programa.

CakePHP

Es un framework para aplicaciones Web desarrollado en PHP. Empezó su desarrollo en el 2005 cuando Ruby On Rails¹ estaba ganando popularidad y desde entonces ha generado muchos proyectos y su comunidad ha crecido. Facilita el usuario la interacción con las Bases de Datos mediante el uso de Active Record, se basa en el patrón arquitectónico Modelo Vista Controlador (MVC), lo que le permite dividir la aplicación en capas.

Entre sus características fundamentales se encuentran:

- ✓ Es compatible con PHP4 y PHP5.
- ✓ Presenta URLs amigables.
- ✓ Sistema de plantillas rápido y flexible.
- ✓ Trabaja en cualquier subdirectorio del sitio.
- ✓ Validación integrada.
- ✓ Componentes de seguridad y sesión.

Después de analizar los Framework existentes, para la aplicación Web se selecciona Symfony en su versión 1.4.10, por ser un framework gratuito y de código abierto y por poseer una gran variedad de paquetes, que permite agilizar el trabajo. Es compatible con el Sistema Gestor de Base de Datos (SGBD) que se va a utilizar. Symfony es independiente del SGBD, prácticamente debido a que internamente usa Propel o Doctrine. Además tiene una amplia comunidad a nivel mundial y mucha documentación en inglés y español.

Para la aplicación de escritorio se escogió .Net pues es una plataforma para diferentes lenguajes de programación entre los que se encuentra Csharp. Fue basado pensando en la seguridad, es capaz de ejecutarse de manera remoto. Además de ser escalable, adaptable para aplicaciones desde las más

¹ **Ruby On Rails:** Es un framework de aplicaciones Web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC).

sencillas hasta las más complejas. Es capaz de interactuar con los múltiples SGBD existentes. Hay que destacar que aunque .Net es de licencia privativa, hay colaboración de otras empresas con Microsoft para el desarrollo y fortalecimiento de .Net ya sea por medio de la implementación de la plataforma para otros sistemas operativos aparte de Windows (*Proyecto Mono de Ximian/Novell para GNU/Linux/macOS X/BSD/Solaris*), el desarrollo de lenguajes de programación adicionales para la plataforma (*Lexico para hispanoparlantes, ANSI C de la Universidad de Princeton, NetCOBOL de Fujitsu, Delphi de Borland, PowerBuilder de Sybase entre otros*) o la creación de bloques adicionales para la plataforma (*como controles, componentes y bibliotecas de clases adicionales*); siendo algunas de ellas software libre, distribuibles bajo la licencia GPL.

1.6.6. Herramientas de Desarrollo

Para desarrollar, normalmente solo es necesario un editor de texto, un intérprete o compilador y una terminal de líneas de comando, pero siempre es más rápido y fácil si se usan algunas herramientas o un Entorno de Desarrollo Integrado (IDE). Esto simplifica el trabajo y ahorra tiempo de desarrollo.

NetBeans

NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java; pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso.

Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones Web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles y resaltados de sintaxis.

Entre las características de la plataforma están:

- ✓ Administración de las interfaces de usuario (ejemplo: menús y barras de herramientas).
- ✓ Administración de las configuraciones del usuario.
- ✓ Administración del almacenamiento (guarda y carga cualquier tipo de dato).
- ✓ Administración de ventanas.
- ✓ Framework basado en asistentes (diálogos pasos a paso).

Zend Studio

Zend Studio es uno de los ambientes de desarrollo integrado, disponible para desarrolladores

profesionales que agrupa todos los componentes de desarrollo necesarios para el ciclo de desarrollo de aplicaciones PHP. Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos. Proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. La del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP, o en caso de que estén instalados, los configura para trabajar juntos en depuración.

Visual Studio

Es una herramienta para los desarrolladores, ya que presenta funcionalidades como el IntelliSense para JavaScript, que ayudará a los programadores de Aplicaciones Web a realizar sus Proyectos mucho más rápido; también el hecho de tener la posibilidad de cambiar entre versiones del Framework .NET ayuda a las empresas a elegir el tipo de estructura que ocuparán sus nuevos Sistemas

Es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta varios lenguajes de programación, al igual que entornos de desarrollo Web como ASP.NET. Aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones Web, así como servicios Web en cualquier entorno que soporte la plataforma .NET. Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas Web y dispositivos móviles.

Después del análisis de los diferentes entornos de desarrollo integrados mencionados anteriormente, para la aplicación Web se escogió NetBeans IDE, para llevar a cabo el desarrollo en PHP, en su versión 6.8, ya que esta versión incorpora un módulo para el trabajo con Symfony, muy amigable y fácil de utilizar.

Para la aplicación de escritorio se escogió Visual Studio por ser un entorno de desarrollo para el framework .Net, posee completamiento de código, es fácil de usar, además por poseer inclusión de tipos genéricos en sus lenguajes de programación, lo que lleva a identificar rápidamente los errores en el momento de compilar el proyecto.

1.6.7. Sistemas Gestores de Base de Datos

Una base de datos (BD) o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Existen unos programas denominados Sistemas Gestores de Bases de Datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las características de un Sistema Gestor de Base de Datos SGBD son:

- ✓ Abstracción de la información.
- ✓ Independencia.
- ✓ Redundancia mínima.
- ✓ Consistencia.
- ✓ Seguridad.
- ✓ Integridad.
- ✓ Respaldo y recuperación.
- ✓ Control de la concurrencia.

Poseen grandes ventajas entre las que se destacan:

- ✓ Facilidad de manejo de grandes volúmenes de información.
- ✓ Gran velocidad de ejecución de las consultas.
- ✓ Independencia del tratamiento de información.
- ✓ Seguridad de la información (acceso a usuarios autorizados).
- ✓ Protección de información, de modificaciones, inclusiones, consulta.

PostgreSQL

Es un sistema de gestión de base de datos relacional orientada a objetos, de software libre, publicado bajo la licencia de Distribución de Software Berkeley (Berkeley Software Distribution) BSD, con características de los mejores sistemas de bases de datos comerciales. PostgreSQL es libre y su código fuente completo está disponible.

Es un motor de base de datos surgido en 1986 con el lanzamiento de su primera versión. Este gestor es altamente potente y posee prestaciones y funcionalidades equivalentes a otros de carácter comercial. Es más completo que MySQL ya que permite métodos almacenados, restricciones de

integridad y vistas. Algunas de sus características principales:

- ✓ Alta concurrencia.
- ✓ Amplia variedad de tipos nativos.
- ✓ Uso de disparadores.
- ✓ Funciones de ventanas.
- ✓ Expresiones de tablas comunes y consultas recursivas.
- ✓ Instalaciones ilimitadas.
- ✓ Estabilidad y confiabilidad.
- ✓ Extensible.
- ✓ Multiplataforma.
- ✓ Diseño para ambientes de amplio volumen.

MySQL

MySQL es un SGBD SQL de código abierto, lo que significa que cualquier programador puede remodelar el código de la aplicación para mejorarlo. Existen versiones de MySQL para Linux y Windows, esto lo hace que sea multiplataforma.

MySQL es multihilo y multiusuario licenciado bajo la GNU GPL. MySQL fue creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Es una popular herramienta de código abierto del mundo de software de base de datos, con más de 100 millones de copias de su software descargado o distribuidos a lo largo de su historia. Con su velocidad, fiabilidad y facilidad de uso. Se ha convertido en la opción preferida para la Web, Web 2.0, SaaS y empresas de telecomunicaciones, debido a que elimina los problemas más importantes asociados con el tiempo de inactividad, mantenimiento y administración y de aplicaciones en línea.

Las principales características de este gestor de bases de datos son las siguientes:

- ✓ Aprovecha la potencia de sistemas multiprocesador a la hora de realiza las búsquedas de datos.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.

- ✓ Posee API's² que permiten el desarrollo en gran cantidad de lenguajes (C, C++, Java, PHP, entre otros.).
- ✓ Gestión de usuarios y contraseñas, que mantiene buen nivel de seguridad en los datos[15].

Realizado el estudio de los gestores de bases de datos anteriores se escogió para el desarrollo del sistema PostgreSQL en su versión 9.1, pues es muy sencillo de utilizar, se integra perfectamente con el framework Symfony y es una de las más utilizadas en sistemas que no requieran de mucho procesamiento de información por parte del gestor de bases de datos.

Para el desarrollo del sistema se propone como servidor Web Apache en su versión 2.2.4 ya que es multiplataforma; es modular pues puede ser adaptado a diferentes entornos y necesidades, es extensible. Es también el servidor Web más utilizado hoy en día y acapara casi el 60 % de la cuota del mercado.

Para el diseño de las interfaces del sistema se propone Ext JS en su versión 3.1 por ser un framework JavaScript del lado del cliente, que permite crear aplicaciones Web con interfaces muy similares a la de una aplicación de escritorio. Entre sus características se pueden encontrar que es multiplataforma, completamente orientado a objetos, y posee múltiples posibilidades para el trabajo con las validaciones y manejo de errores en el cliente.

Conclusiones

Del análisis sobre la forma de gestionar y controlar los AFT se evidenció que el trabajo realizado con estos se ejecuta de forma manual o semi-automatizada lo que provoca la existencia de diferentes problemas.

Las aplicaciones estudiadas para la gestión de los AFT a los niveles internacional, nacional e institucional, no satisfacen las necesidades para garantizar los objetivos de control de la institución por lo que se decide implementar una desde un nivel básico.

² API por sus siglas en inglés (Application Programming Interface) o **interfaz de programación de aplicaciones** es conjunto de funciones y procedimientos (métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas.

Capítulo 2: Características del Sistema

Introducción

El presente capítulo abordará las características de la aplicación que se pretende desarrollar con el objetivo de dar una solución factible al problema que se investiga. Se realiza el levantamiento de los requisitos, definiendo los requisitos funcionales y no funcionales. Se detalla cada uno de los casos de uso del sistema. Así como modelo de dominio, entidades, actores que intervienen y diagrama de caso de uso del sistema.

2.1. Descripción del Sistema Propuesto

El sistema informático que se propone constituye un factor fundamental para el proceso de auditoría y control sobre los AFT de la Facultad 6 de la UCI. La propuesta será una herramienta de gestión que contará con dos módulos; el primero una aplicación Web, esta es la encargada de realizar la gestión de activos fijos, usuarios, generar etiquetas de códigos de barra y demás entidades que se relacionan con el sistema. El segundo módulo consiste en una aplicación de escritorio que estará montada en una PC cliente (laptops) con un lector de código de barras y será la encargada de realizar las auditorías y mostrar los reportes. El sistema informático que se propone permitirá:

- ✓ Realizar auditorías sobre los AFT a mucho más del 10% mensual, incluso se realizaría en un lapso de tiempo relativamente corto si se desea.
- ✓ Generación y recuperación de códigos de barra en formato PDF con la información de los AFT para ser adherido a estos y evitar la equivocación en la escritura del rótulo o corroborarlos con los existentes.
- ✓ Evitará introducir errores de la información de los AFT, puesto que cargará el documento realizado por la dirección de economía y planificación física.
- ✓ Permitirá ahorro de recursos en despliegue puesto que se necesitará solamente de un servidor de aplicación y un servidor de base de datos que pudieran compartirse entre sí, y el sistema estaría disponible desde cualquier punto de la red UCI.

Será un sistema de alto grado de fiabilidad que emplea el principio de seguridad informática control de acceso basado en roles (RBAC). Para la generación de etiquetas y acta de responsabilidad material utiliza el tipo de documentos portable (PDF), que es poco probable la modificación de los datos que se hayan escrito en el.

2.2. Definir Modelo del Dominio

Debido a las características, grado de complejidad del problema que se plantea y el conocimiento que se posee sobre el mismo, se decide realizar Modelo de Dominio o Modelo Conceptual como también se le conoce. Se considera suficiente el empleo de esta variante de modelo de negocio, dado que se centra en una parte del negocio, la relacionada con el ámbito del proyecto.

El modelo de dominio puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. Es utilizado como un medio para comprender el sector de negocios al cual el sistema va a servir. Puede ser tomado como el punto de partida para el diseño del sistema. Cuando se realiza la programación orientada a objetos, el funcionamiento interno del software va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de dominio constituye una primera versión del sistema. El modelo conceptual de la problemática a resolver se muestra en la Figura 2.1.

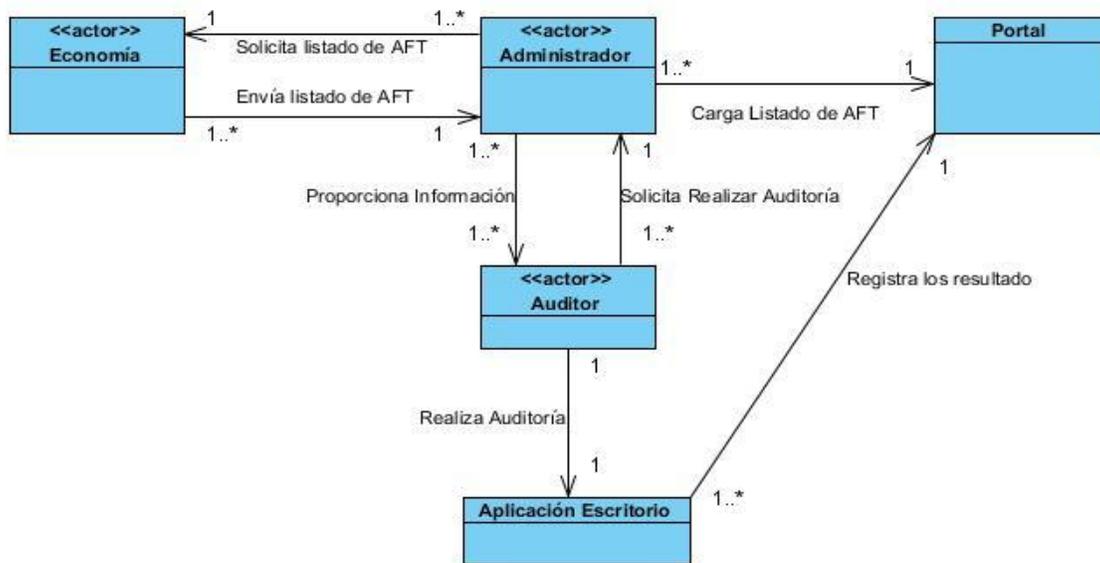


Figura 2.1: Diagrama del Modelo de Dominio creado a partir del problema planteado.

2.3. Descripción de los Objetos

A continuación se muestra la Tabla 2.1 con la definición de las entidades que interactúan en el negocio y los principales conceptos que se tratan en el problema que se analiza.

Tabla 2.1: Descripción de las entidades que interactúan en el negocio

Entidad	Descripción
---------	-------------

WSDL	Servicios Web que pueden ser consumidos.
Reporte	Contiene la información de los resultados obtenidos.
Acta de Responsabilidad Material	Una persona se hace responsable de los medios de un local.
Usuario	Representa la persona que interactúa con el sistema solicitando una respuesta.
Planilla de AFT	Contiene la toda la información de los Activos Fijos Tangibles.
Modelo de Movimientos de Medios	Contiene los datos de un movimiento realizado a un AFT.

2.4. Especificación de los Requisitos del Sistema

Para que un esfuerzo de desarrollo de software tenga éxito, es esencial comprender perfectamente los requisitos del software. Independientemente de lo bien diseñado o codificado que esté un programa, si se ha analizado y especificado pobremente, decepcionará al usuario y desprestigiará al que lo ha desarrollado. La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan ardua como establecer los requisitos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas.

La calidad con que se realice la captura de los requisitos va a influenciar en todo el proceso de desarrollo del software repercutiendo en el resto de las fases del mismo. Una definición eficiente de los requisitos permite mostrar un nivel de disciplina en el proceso de desarrollo, dar un mejor soporte a la Gestión de Cambios y ganar una mayor eficiencia en las pruebas reduciendo el riesgo, mejorando la calidad y permitiendo la automatización. Además contribuye a tomar mejores decisiones de diseño y de arquitectura. También le permite al equipo de desarrollo reducir los problemas de mantenimiento.

2.4.1. Requisitos Funcionales

Los requisitos funcionales son aquellos que describen qué debe hacer el sistema o sea especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto, desde el punto de vista de las necesidades del usuario, son capacidades o condiciones que debe cumplir el sistema y que están fuertemente ligados a las opciones del programa. A continuación se muestran los requisitos funcionales del sistema.

Requisitos Funcionales para la aplicación Web

RF 1. Autenticar Usuario

RF 2. Gestionar Usuario

- 2.1 - Insertar un nuevo usuario.
- 2.2 - Buscar usuario partiendo de un criterio especificado.
- 2.3 - Modificar datos de un usuario existente.
- 2.4 - Eliminar un usuario del sistema.

RF 3. Gestionar Centro de Costo

- 3.1 - Insertar un nuevo centro de costo.
- 3.2 - Buscar un centro de costo.
- 3.3 - Modificar datos de un centro de costo existente.
- 3.4 - Eliminar un centro de costo del sistema.

RF 4. Gestionar Área

- 4.1 - Insertar una nueva área.
- 4.2 - Buscar un área.
- 4.3 - Modificar datos de un área existente.
- 4.4 - Eliminar un área del sistema.

RF 5. Gestionar Activo Fijo

- 5.1 - Insertar un activo fijo.
- 5.2 - Buscar un activo fijo.
- 5.3 - Modificar datos de un activo fijo existente.
- 5.4 - Eliminar un activo fijo del sistema.

RF 6. Cargar Documento Excel

RF 7. Generar Código de Barras

RF 8. Generar Acta de Responsabilidad Material

RF 9. Generar Modelo de Movimiento de Medios

Requisitos Funcionales para la aplicación de escritorio

RF 10. Solicitar Realizar Auditoría

RF 11. Realizar Auditoría

RF 12. Ver Resultados de Auditoría

RF 13. Registrar Datos de Auditoría

2.4.2. Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, forman una parte significativa de la especificación. Tienen la característica de marcar la diferencia, pues una vez

comprobado que el producto cumple con lo requerido por el cliente, las propiedades no funcionales; como cuán usable, seguro, conveniente y agradable, distinguen a un producto bien aceptado de uno con poca aceptación.

Para un correcto funcionamiento el sistema debe tener como características principales para su funcionamiento los siguientes requisitos no funcionales:

Usabilidad: El sistema puede ser utilizado por cualquier persona que tenga al menos conocimientos básicos en el manejo de la computadora, navegación y exploración de los sitios Web en sentido general, las operaciones se realizan con bajo nivel de complejidad.

Seguridad: Es el requerimiento más complejo y difícil de garantizar, a su vez permitirá garantizar que la aplicación será utilizada correctamente por cada usuario según sus niveles permitidos.

- ✓ **Confidencialidad:** Se debe tratar el manejo de permisos de forma que solamente se acceda a la información autorizada de acuerdo a los niveles de permisos que debe tener cada usuario del sistema. De esta forma se garantiza que la información no sea expuesta a personal indebido.
- ✓ **Integridad:** Se debe tratar el manejo de la información de forma tal, que la información no sea modificada por personal ajeno. Esto evitará alteraciones en los resultados planteados en la documentación.
- ✓ **Disponibilidad:** Se deberá garantizar el acceso pleno de cada usuario con facultades para el uso de la aplicación las 24 horas del día.

Copia de Respaldo: El sistema debe permitir la realización de copias de respaldo hacia otros dispositivos de almacenamiento para prevenir pérdida de información. Esta base de datos debe ser posible de recuperar mediante las copias de respaldo previamente creadas.

Soporte: Debe presentar un diseño que sea capaz de la realización de pruebas que permitan la explotación de la aplicación de forma eficiente. Se incluye la posibilidad de brindar asistencia técnica que permita la solución de problemas en tiempo real de ejecución que garantice la solución de fallas que pueda presentar. Debe ser garantizada su adaptabilidad y compatibilidad con los distintos sistemas operativos.

Interfaz: El diseño de la interfaz visual debe ser amigable para los usuarios que interactúan con la aplicación, de forma tal que permita el fácil entendimiento de las funcionalidades que brinda, además de poseer colores amigables y refrescantes para una mejor interacción entre el usuario y la

aplicación.

El diseño de la interfaz permitirá mostrar mensajes para la guía de usuarios en caso de errores en entradas inválidas de los datos o de confirmación de realización de actividades.

Software:

Para el modulo Web:

Cliente:

- ✓ Sistema operativo Linux o Windows.
- ✓ Navegadores Internet Explorer (versión 8 o superior) y Mozilla (versión 8 o superior).

Servidor:

- ✓ Sistema Operativo Linux, Windows o Unix.
- ✓ Servidor Web Apache 2.2.4 o superior.
- ✓ Servidor de Bases de Datos PostgreSQL 9.1 o superior.
- ✓ PHP 5.2.3.

Para el módulo de escritorio:

- ✓ .Net framework 2.0 o superior.
- ✓ Lector de código de barras.

Hardware:

Para los dos módulos:

Cliente:

- ✓ Se necesitan como requerimientos mínimos una Computadora Personal (*Personal Computer*) PC con procesador Pentium IV o superior.
- ✓ Microprocesador con una velocidad a 2.0 GHz o superior.
- ✓ Memoria RAM de 256 MB o superior.
- ✓ Tarjeta de red de 10 MB o superior.

Servidor:

- ✓ Se necesitan como requerimientos mínimos una PC con procesador Pentium IV o superior.
- ✓ Microprocesador con una velocidad de 3.0 GHz o superior.
- ✓ Memoria RAM de 1GB o superior.
- ✓ HDD de 80 GB o superior.
- ✓ Tarjeta de red de 100 MB o superior.

Rendimiento: La disponibilidad de trabajo en red contra el servidor es constante. Se garantiza que la respuesta a solicitudes de los usuarios, sea en un período de tiempo breve para evitar la acumulación de trabajo por parte de los responsables. El sistema deberá ser lo más estable y confiable posible.

Portabilidad: El producto corre sobre una plataforma Web, los servicios en PHP y su sistema de bases de datos en PostgreSQL. Es de fácil mantenimiento.

Restricciones en el diseño y la implementación: Es un sistema Web desarrollada con la tecnología para creación de páginas Web dinámicas PHP y base de datos en PostgreSQL.

2.5. Actores y Casos de Uso del Sistema

Un actor del sistema es una entidad externa, que representa el rol de una o varias personas, un equipo o un procedimiento automatizado que interactúa con el sistema, por lo que no forma parte del mismo. Puede intercambiar información o ser un recipiente pasivo de información.

Un caso de uso del sistema es una técnica para la captura de requisitos potenciales y a su vez una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

2.5.1. Actores del Sistema

En la Tabla 2.2 se describen los actores que interactúan en el sistema:

Tabla 2.2: Descripción de los actores que interactúan en el sistema

Actores	Descripción
Administrador	Es el encargado de la gestión de usuarios, de activos fijos, áreas, centros de costo y organizaciones.
Auditor	Es el encargado de realizar las auditorías, capturar los datos de estas.

Usuario	Este puede visualizar los datos existentes.
---------	---

2.5.2. Casos de Uso del Sistema

A continuación en la Tabla 2.3 se describen los casos de usos del sistema:

Tabla 2.3: Casos de Uso del Sistema organizados por orden de prioridad

Orden	Nombre del Caso de Uso	Prioridad	Descripción
2	Gestionar Usuario	ALTO	Solo podrá ser accedido por el administrador del sistema. Permitiéndole crear, eliminar y modificar los usuarios que accederán al sistema, así como definir para cada uno los roles o permisos que tendrán dentro del mismo.
1	Autenticar Usuario	ALTO	Constituye un elemento de seguridad dentro del sistema, admite que solo los usuarios registrados puedan acceder y navegar dentro del sistema, de acuerdo con el permiso (rol) asignado.
6	Cargar Documento Excel	ALTO	Funcionalidad dentro del sistema considerada de vital importancia para la existencia del mismo. Se encarga de importar todos los datos referentes a centros de costos, áreas y activos fijos hacia la aplicación.
10	Solicitar Realizar Auditoría	ALTO	Se encarga de traer los datos hacia la aplicación de escritorio para realizar la posterior auditoría.
11	Realizar Auditoría	ALTO	Es el que comprueba los AFT faltantes y sobrantes de un local.
7	Generar Código de Barras	MEDIO	Funcionalidad establecida para el administrador del sistema mediante la cual se generan el/los códigos de barras a través del rótulo de los AFT y se muestra a través de un archivo .pdf.
13	Registrar Datos de Auditoría	MEDIO	Es el encargado de mantener una actualización constante de la base de datos.
8	Generar Acta de Responsabilidad Material	MEDIO	Funcionalidad que permite realizar las actas de responsabilidad material, ya sea individual como colectiva. Donde se visualiza el responsable de cada área.
9	Generar Modelo Movimiento de Medios	MEDIO	Se efectúa un movimiento de AFT de local y permite visualizar un archivo .pdf con la información del movimiento en forma de reporte.
12	Ver Resultados de Auditoría	MEDIO	A través de esta funcionalidad se puede ver los AFT faltantes y sobrantes dentro del local auditado.

3	Gestionar Centro de Costo	BAJO	Funcionalidad mediante la cual se puede realizar las operaciones de insertar, modificar, eliminar y buscar los datos de un centro de costo en el sistema.
4	Gestionar Área	BAJO	Funcionalidad mediante la cual se puede realizar las operaciones de insertar, modificar, eliminar y buscar los datos de un área en el sistema.
5	Gestionar Activo Fijo	BAJO	Funcionalidad mediante la cual se puede realizar las operaciones de insertar, modificar, eliminar y buscar los datos de un AFT en el sistema.

2.5.3 Patrón de Casos de Uso (CRUD)

Los patrones de casos de uso son comportamientos que deben existir en el sistema, ayuda a describir qué es lo que el sistema debe hacer, es decir, describe el uso del sistema y cómo este interactúa con los usuarios. Estos son utilizados generalmente como plantillas que especifican como deberían ser estructurados y organizados los casos de uso y capturan mejores prácticas para modelar casos de uso.

Los patrones de casos de uso brindan los siguientes beneficios:

- ✓ Aumentar la productividad.
- ✓ Reutilizar elementos existentes.
- ✓ Evitar el re trabajo por errores.
- ✓ No invertir tiempo en resolver problemas ya resueltos.
- ✓ Aplicar la teoría al trabajo práctico.
- ✓ Habilitar las herramientas de soporte para modelar el desarrollo.

Durante el diseño de los casos de uso del sistema se utilizó el patrón CRUD (acrónimo de Crear, Obtener, Actualizar y Borrar del original en inglés: Create, Read, Update and Delete). El mismo permite modelar todas las operaciones que se pueden realizar sobre una misma entidad, tales como crearla, leerla, actualizarla y eliminarla. Este patrón deberá ser usado cuando todas las operaciones contribuyen al mismo valor de negocio y todas son cortas y simples.

2.5.4. Diagrama de Casos de Uso del Sistema

Gracias a las facilidades que ofrece el UML, se procede a capturar los requisitos funcionales del sistema y a representar a los mismos mediante un diagrama de casos de uso (DCU). La Figura.2.2

muestra los actores y casos de uso definidos para el sistema propuesto.

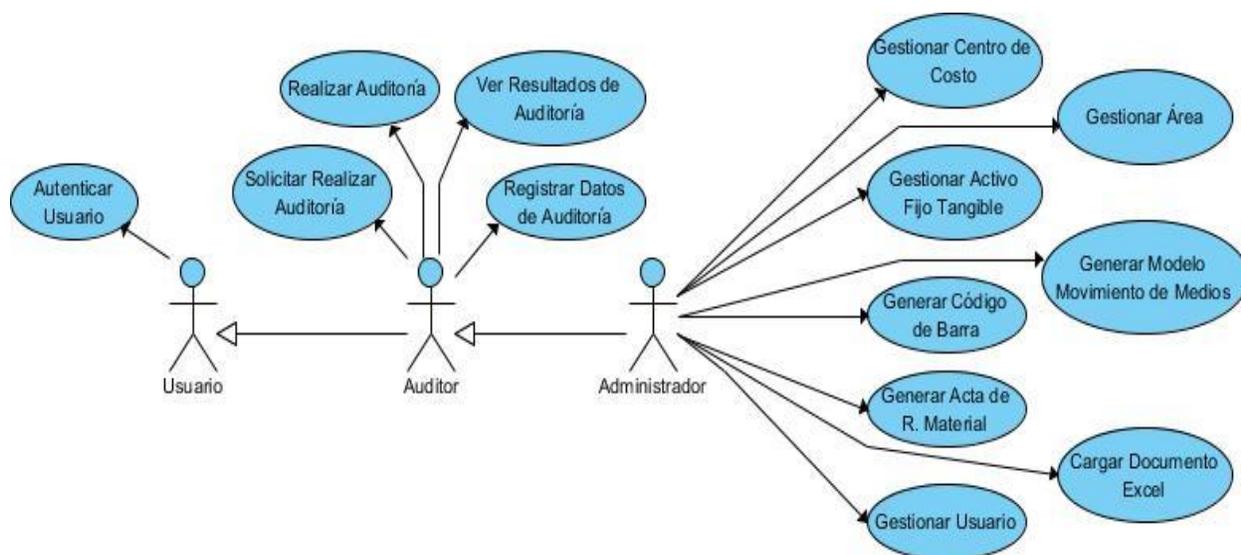


Figura 2.2: DCU del sistema definido a partir del problema planteado.

2.5.5. Descripciones de los Casos de Uso del Sistema

Las descripciones textuales de los principales casos de uso se pueden encontrar en el [\(Anexo 6\)](#).

Conclusiones

En aras de satisfacer las necesidades del cliente, se realizó un estudio detallado correspondiente al problema que se analiza, se modelaron los artefactos que genera la metodología OpenUp: modelo de dominio, actores y casos de uso del sistema, descripciones de los mismos, así como el levantamiento de los requisitos funcionales y no funcionales. Se identificaron los casos de uso del sistema en correspondencia con las funcionalidades recogidas en los requisitos funcionales. Se utilizó el patrón CRUD para estructurar y organizar los casos de uso, lo que definió el diagrama de casos de uso del sistema y la descripción detallada de cada uno. Luego de haber quedado definido el diseño del sistema y las relaciones entre sus elementos, es posible dar comienzo el proceso de implementación del “Sistema para Auditoría y Control de Activos Fijos Tangibles de la Facultad 6 de la UCI”.

Capítulo 3: Análisis y Diseño del Sistema

Introducción

En este capítulo se realiza la descripción del diseño de la aplicación mediante el uso de los diferentes diagramas afines para dicha actividad. Se analizan cada uno de los casos de usos y se describe, mediante el uso de estereotipos Web, la relación entre las distintas páginas que conforman la aplicación que se pretende desarrollar. Se obtiene además el diagrama de clases persistentes para la generación de la base de datos. Se definen los principios de diseño y los patrones y estándares a utilizar y se define el diagrama de despliegue del sistema propuesto.

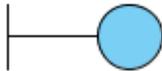
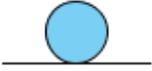
3.1. Modelo del Análisis

El análisis consiste en obtener una visión detallada del sistema, de modo que solo se interesa por los requisitos funcionales, para refinarlos y estructurarlos, además es el principal eslabón para el comienzo de las actividades de diseño e implementación. El objetivo del análisis, se centra en comprender perfectamente los requisitos del software y no en precisar cómo se implementa el producto.

3.1.1. Diagrama de Clases del Análisis

El diagrama de clase del análisis se realiza para cada caso de uso del sistema. Muestra las clases participantes en dichos casos de uso, así como la relación entre ellas. En los diagramas de clases se identifican tres tipos de clases: Interfaz, Controladora y Entidad. En la Tabla 3.1 se representan los estereotipos a utilizar en el diagrama de clases del análisis.

Tabla 3.1: Representación de los estereotipos a utilizar en el diagrama de clases del análisis.

Clases	Descripción	Representación
Interfaz	Las clases <i>interfaz</i> se utilizan para modelar la interacción entre el sistema y sus actores.	 Interfaz
Entidad	Las clases <i>entidad</i> se utilizan para modelar información que poseen larga vida y que es a menudo persistente.	 Entidad
Control	Las clases <i>control</i> representan coordinación, secuencia, transacciones y control de objetos y son utilizadas para encapsular el control de un caso de uso.	 Control

Diagramas de Clases del Análisis

A continuación se muestran los diagramas de clases del análisis de los casos de uso más significativos.

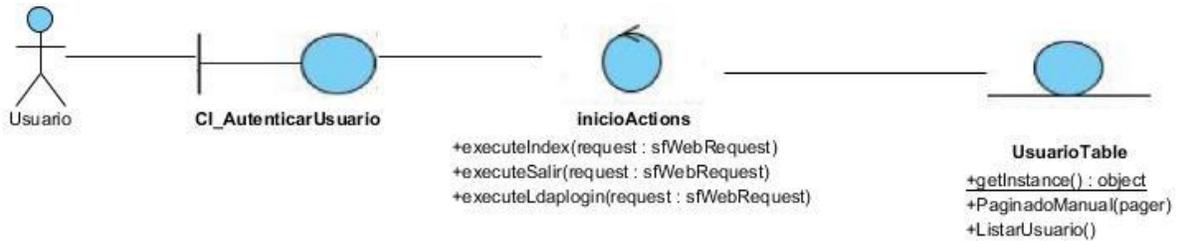


Figura 3.1: Diagrama de Clase del Análisis CU Autenticar Usuario

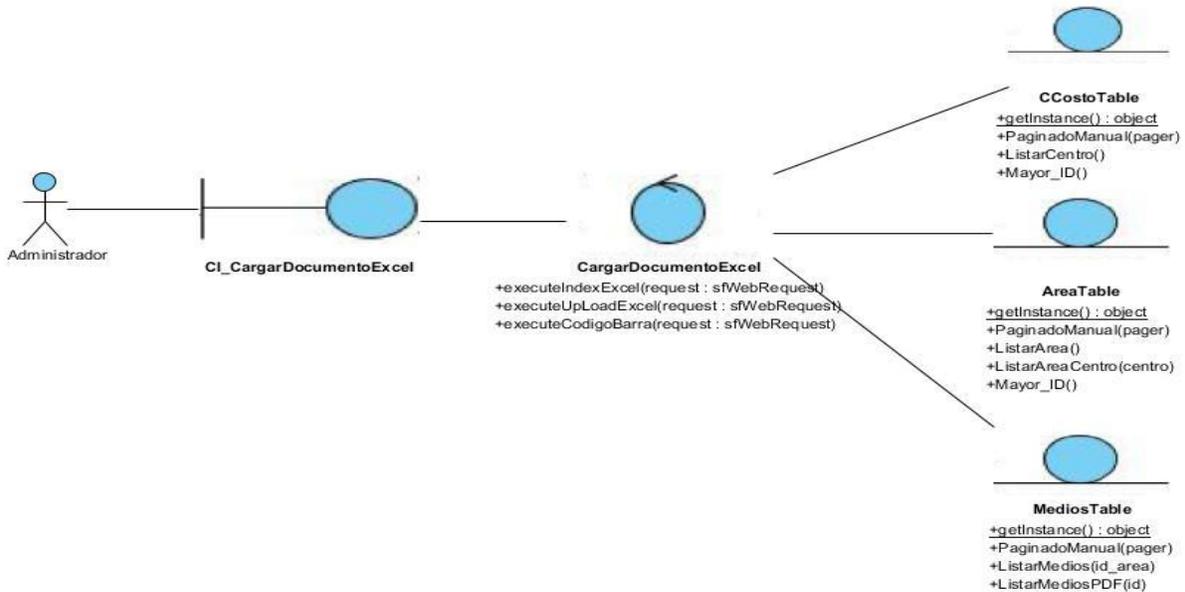


Figura 3.2: Diagrama de Clase del Análisis CU Cargar Documento Excel

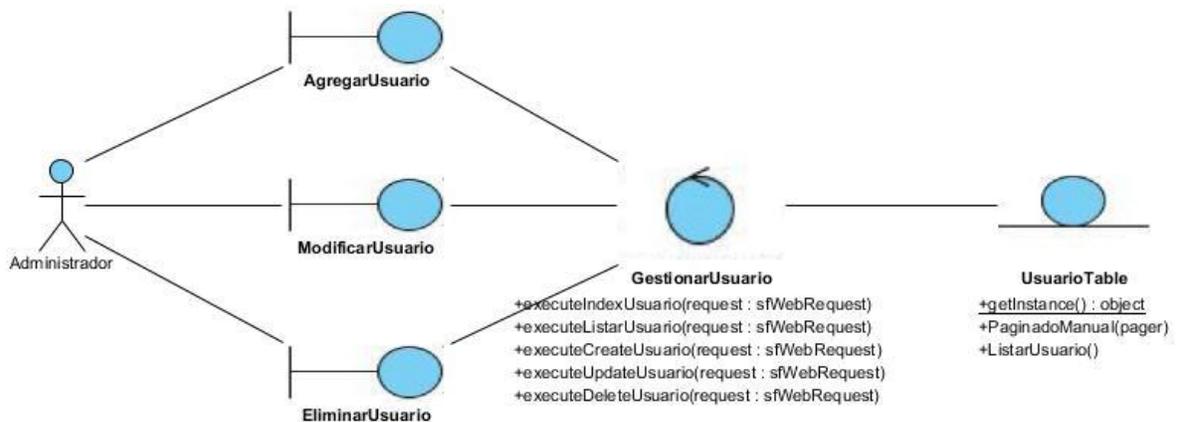


Figura 3.3: Diagrama de Clase del Análisis CU Gestionar Usuario



Figura 3.4: Diagrama de Clase del Análisis CU Solicitar Realizar Auditoría

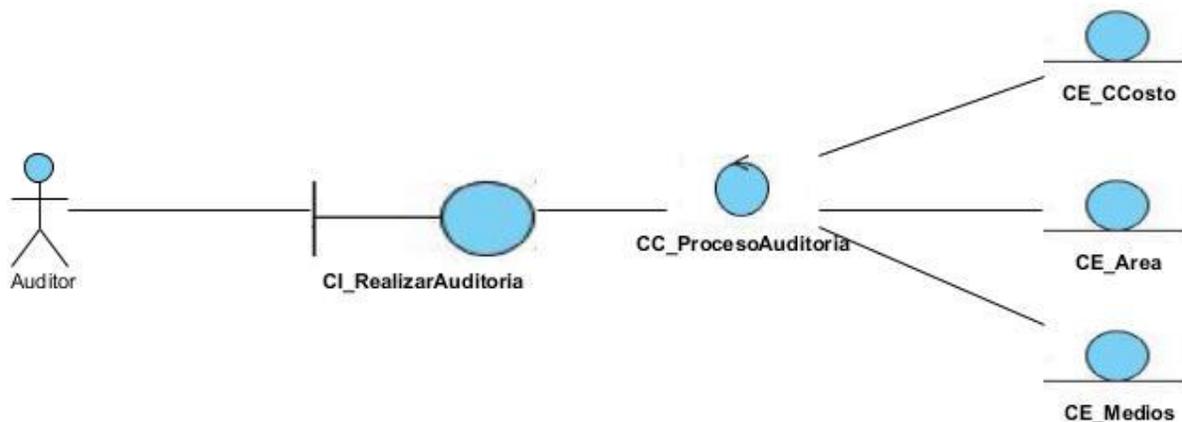


Figura 3.5: Diagrama de Clase del Análisis CU Realizar Auditoría

3.2. Modelo del Diseño

El propósito del diseño es modelar el sistema y encontrar la forma para que soporte todos los requisitos definidos. Se planean todos los aspectos relacionados con las restricciones y características del sistema como lo son el lenguaje de programación a utilizar, el sistema operativo donde se podrá ejecutar la aplicación y las tecnologías de interfaz de usuario. El diseño tiene en cuenta los requisitos funcionales y no funcionales para obtener una visión detallada de la implementación futura para que el sistema sea realizado sin errores.

3.2.1. Patrón y Estilo de Arquitectura

Los patrones de arquitectura son patrones de diseño de software que brindan soluciones a problemas de arquitectura en ingeniería de software. Describen los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Expresa un esquema de organización estructural esencial para un sistema, que consta de subsistemas, sus responsabilidades e interrelaciones. Tienen una escala más grande en comparación con los patrones de diseño.

Son utilizados generalmente como plantillas que describen la estructura y organización de los casos de uso. Son patrones que capturan mejores prácticas para modelar casos de uso.

Los dominios en el diseño de Patrones son:

- ✓ Control de acceso: Hay muchas situaciones en las cuales el acceso a datos, características y funcionalidad son limitadas a la definición de los usuarios. Desde un punto de vista arquitectónico, acceder a determinadas partes del software debe tener un riguroso control.
- ✓ Concurrencia: Muchas aplicaciones deben manejar múltiples tareas de forma que simule el paralelismo. Hay muchas formas de manejar esta concurrencia, y cada una puede ser presentada por un patrón arquitectónico diferente.
- ✓ Distribución: El problema de distribución dirige el problema de forma en que los sistemas o componentes se comunican con otros en un entorno distribuido.
- ✓ Persistencia: Los datos persistentes son almacenados en bases de datos o archivos y pueden ser leídos o modificados por otros procesos más adelante. En los entornos orientados a objetos esto va más allá, y lo que puede ser accedido o modificable son las propiedades de los objetos.

Cliente-Servidor

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Las principales características de la arquitectura cliente-servidor son:

- ✓ Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras y módems.
- ✓ Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco.
- ✓ Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- ✓ Existe una clara distinción de funciones basada en el concepto de servicio, que se establece entre clientes y servidores.

- ✓ La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- ✓ Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- ✓ No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- ✓ El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
- ✓ El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente-Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

En la Figura 3.6 de muestra el flujo Cliente- Servidor:

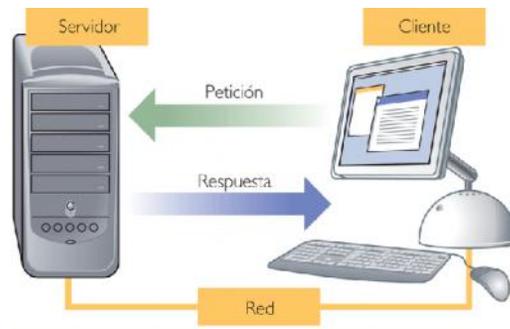


Figura 3.6: Representación gráfica del flujo Cliente-Servidor

Una vez realizado el análisis al modelo cliente – servidor y teniendo en cuenta la lógica del sistema que se desea desarrollar. Se decidió utilizar este estilo arquitectónico ya que provee flexibilidad, usabilidad, escalabilidad en las comunicaciones y uso de entornos multiplataforma.

Modelo Vista Controlador

El Modelo Vista Controlador (MVC) es un patrón para el desarrollo del software que se basa en separar los datos (por un lado), la interfaz del usuario (por otro) y la lógica interna (por un último lado).

Es mayormente usado en aplicaciones Web, dónde la vista es la página HTML, el modelo es el Sistema de Gestión de Base de Datos y la lógica interna, y el controlador es el responsable de recibir los eventos y darles solución.

- ✓ **El Modelo:** Es la representación de la información en el sistema. Trabaja junto a la vista para mostrar la información al usuario y es accedido por el controlador para añadir, eliminar, consultar o actualizar datos.
- ✓ **La Vista:** Es el objeto que representa al modelo en un formato adecuado para que el usuario pueda interactuar con él, casi siempre es la interfaz de usuario.
- ✓ **El Controlador:** Es el elemento más abstracto. Recibe, trata y responde los eventos enviados por el usuario o por la propia aplicación. Interactúa tanto con el modelo como con la vista.

En la Figura 3.7 se puede observar el funcionamiento de un sistema implementado que usa el patrón arquitectónico Modelo Vista Controlador (MVC).

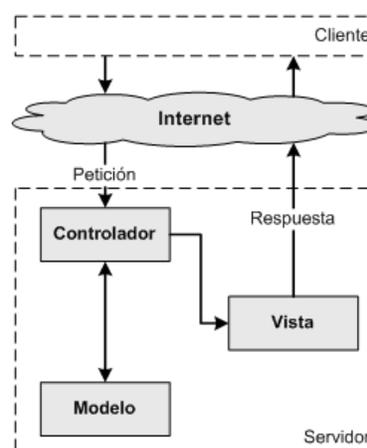


Figura 3.7: Representación del Patrón Arquitectónico Modelo Vista Controlador.

3.3. Patrones de Diseño (GRASP)

Un patrón de diseño es la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Este resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Durante el desarrollo del sistema se utilizaron patrones básicos de asignación de responsabilidades

como Patrones Generales de Software para Asignar Responsabilidades (General Responsibility Assignment Software Patterns) GRASP.

Experto: Se encarga de asignar la responsabilidad a la clase que cuenta con la información necesaria para efectuar la tarea que tiene encomendada. Las clases que brinda el framework Ext JS se encargarán de visualizar las interfaces ya que cuentan con la información para crear los diferentes componentes visuales.

Creador: Tiene en cuenta para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto solo pueda ser creada por el objeto que contiene la información necesaria. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo cual favorece al mantenimiento del sistema.

Bajo Acoplamiento: Brinda como solución asignar responsabilidades de manera que las clases no dependan fuertemente de otras. Ofrece como beneficio que son fáciles de entender por separadas, fáciles de reutilizar y no se afectan por cambios de otros componentes. Dicho patrón se tiene en cuenta debido a la importancia de realizar un diseño de clases independientes que soporten los cambios.

Alta cohesión: Propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables, para evadir un trabajo excesivo. Su utilización mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, generan un bajo acoplamiento, soporta mayor capacidad de reutilización.

3.4 Diagramas de Clases del Diseño

Para la realización de los diagramas de clases del diseño se presentan como elementos significativos a tres clases UML fundamentales: server page, client page y form, empleadas para el código servidor, código cliente y formularios respectivamente. Esto permite representar ficheros contenedores de sentencias script.

Descripción de los elementos del diseño.

En la elaboración de los diagramas de clases del diseño se utilizarán los elementos que a continuación se muestran en la Tabla 3.2:

Tabla 3.2: Representación de los Estereotipos a Utilizar en el Diagrama de Clases del Diseño

Clases	Descripción	Representación
Server Page	Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor. Solo puede mantener relaciones con objetos que se encuentren en el servidor.	
Client Page	Representan páginas que son dibujadas por el navegador Web y pueden ser una combinación de algún o algunos lenguajes, scripts del lado del cliente.	
Form	Representa una colección de campos de entrada que forman parte con una página del lado cliente	

Relaciones entre las clases:

En la Tabla 3.3 se muestran las relaciones entre las clases.

Tabla 3.3: Representación de las relaciones de clases del diseño con estereotipos Web

Clases/Relaciones	Server Page	Client Page	Form
Server Page	Redirect	Build/Redirect	-
Client Page	Link	Link/Redirect	Agregación
Form	Submit	-	-

A continuación se muestra una representación gráfica de la estructura de los principales casos de uso.

CU Autenticar Usuario

La Figura 3.8 muestra las clases utilizadas para la autenticación de los usuarios en el sistema. En este caso la página cliente CI_Autenticar muestra el formulario Form_Autenticar a través del cual se introduce las credenciales del usuario, para la autenticación utiliza la clase controladora AutenticarUsuario la cual consume el servicio Web WSDL y se comunica con la clase entidad Usuario.

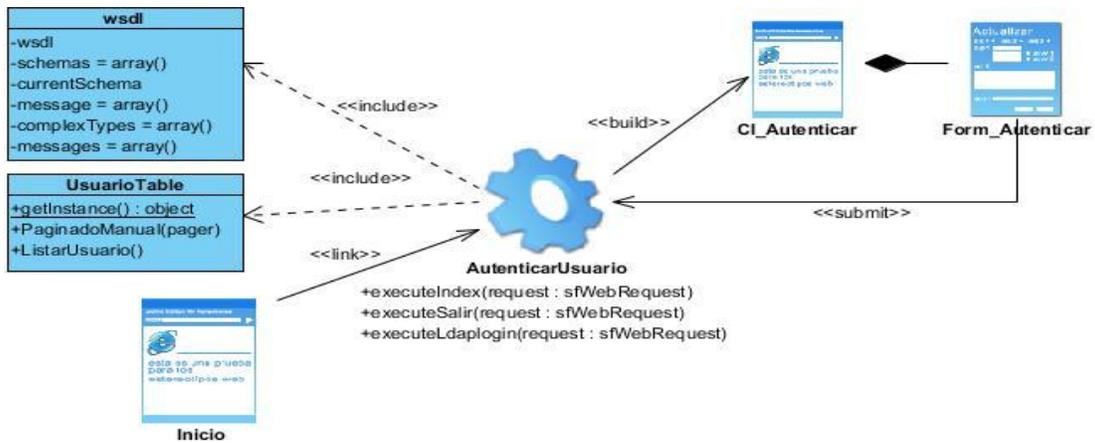


Figura 3.8: Diagrama de Clase de Diseño CU Autenticar Usuario

CU Cargar Documento Excel

La Figura 3.9 muestra las clases utilizadas para el proceso de captura de los datos a partir de un documento Excel. En este caso la página cliente CP_IndexExcel incluye el formulario Form_IndexExcel a través del cual se buscará el documento Excel, para el proceso se utiliza la clase controladora CargarDocumentoExcel la misma incluye la librería sfExcelReader para el proceso de captura de los datos y las clases entidades CCosto, Área y Medios para la persistencia de los mismos.

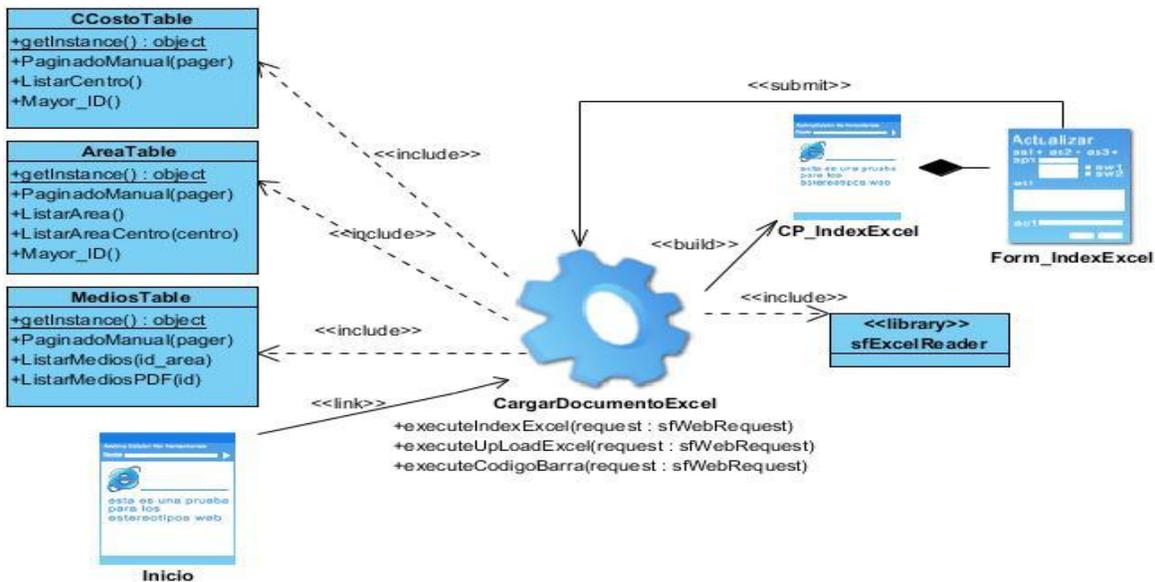


Figura 3.9: Diagrama de Clase de Diseño CU Cargar Documento Excel

CU Gestionar Usuario

La Figura 3.10 muestra las clases utilizadas para el proceso de gestionar usuario. En este caso la página cliente CP_IndexUsuario incluye el formulario UsuarioForm a través del cual se agregará un nuevo usuario al sistema, así como las opciones de buscar usuario, modificar usuario y eliminar usuario, para el proceso se utiliza la clase controladora GestionarUsuario la misma la cual utiliza la clase entidad Usuario para la persistencia de los mismos.

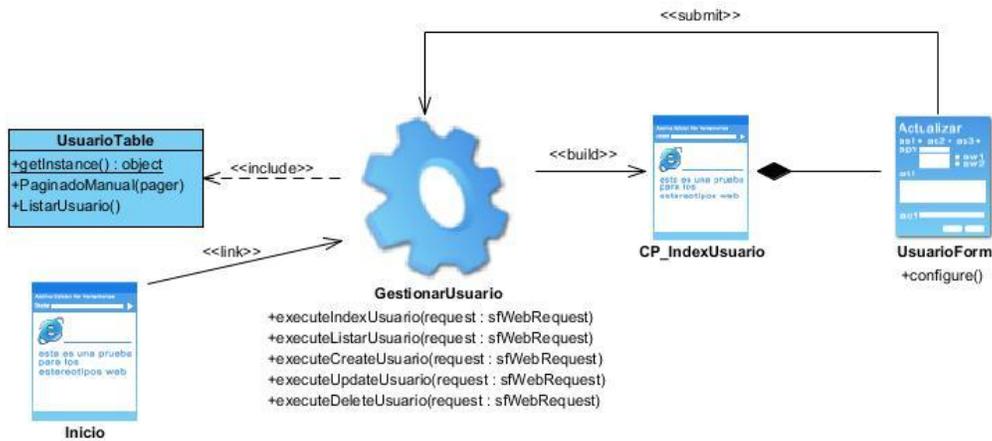


Figura 3.10: Diagrama de Clase de Diseño CU Registrar Datos de Auditoría

CU Solicitar Realizar Auditoría

La Figura 3.11 muestra las clases utilizadas para el proceso de solicitud de auditoría. En este caso se muestra la interfaz principal que contiene la opción seleccionar datos de auditoría, la cual contiene el formulario solicitar datos, a través del cual se realiza el pedido de los datos, para el proceso se utiliza la clase controladora ProcesoAuditorías la cual se comunica con las clases entidades CCosto, Área y Medios.

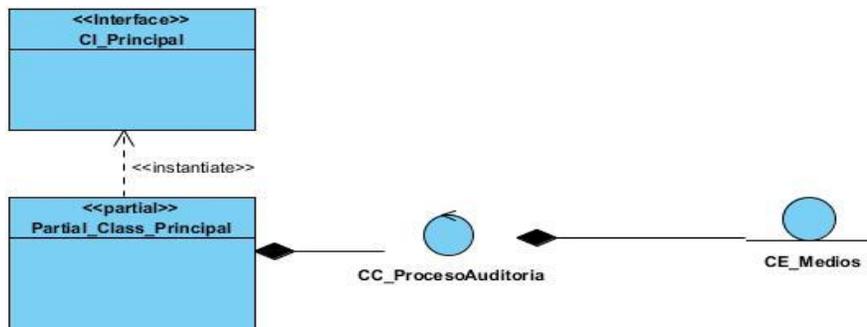


Figura 3.11: Diagrama de Clase de Diseño CU Solicitar Realizar Auditoría

CU Realizar Auditoría

La Figura 3.12 muestra las clases utilizadas para el proceso de realizar auditoría. En este caso se muestra la interfaz principal que contiene la opción realizar auditoría, la cual contiene el formulario para introducir el rótulo a través del lector de código de barras, para el proceso se utiliza la clase controladora *ProcesoAuditorías* la cual interactúa con la clase entidad *Medios*.

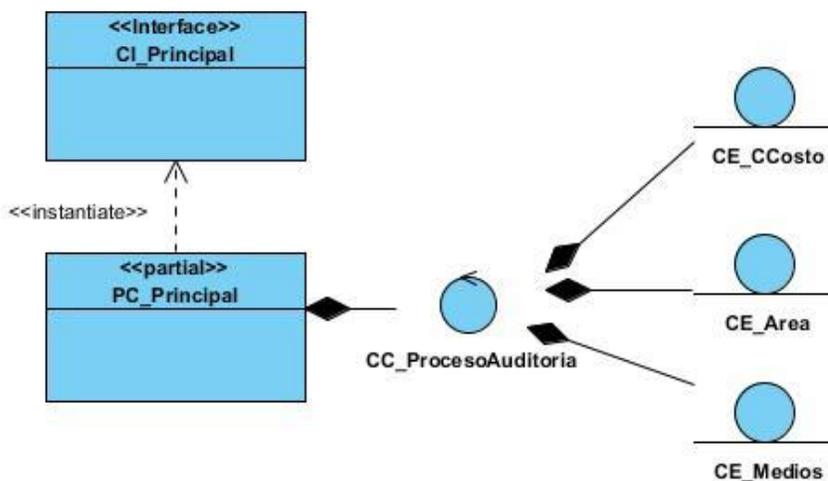


Figura 3.12: Diagrama de Clase de Diseño CU Realizar Auditoría

3.5. Diagramas de Interacción

Son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado. Existen dos tipos de diagramas de interacción: diagramas de colaboración y diagramas de secuencias. Durante el análisis de la aplicación será utilizado este último dado que es un tipo de diagrama usado para modelar interacción entre objetos. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

A continuación se muestran los diagramas de secuencia de los principales casos de uso.

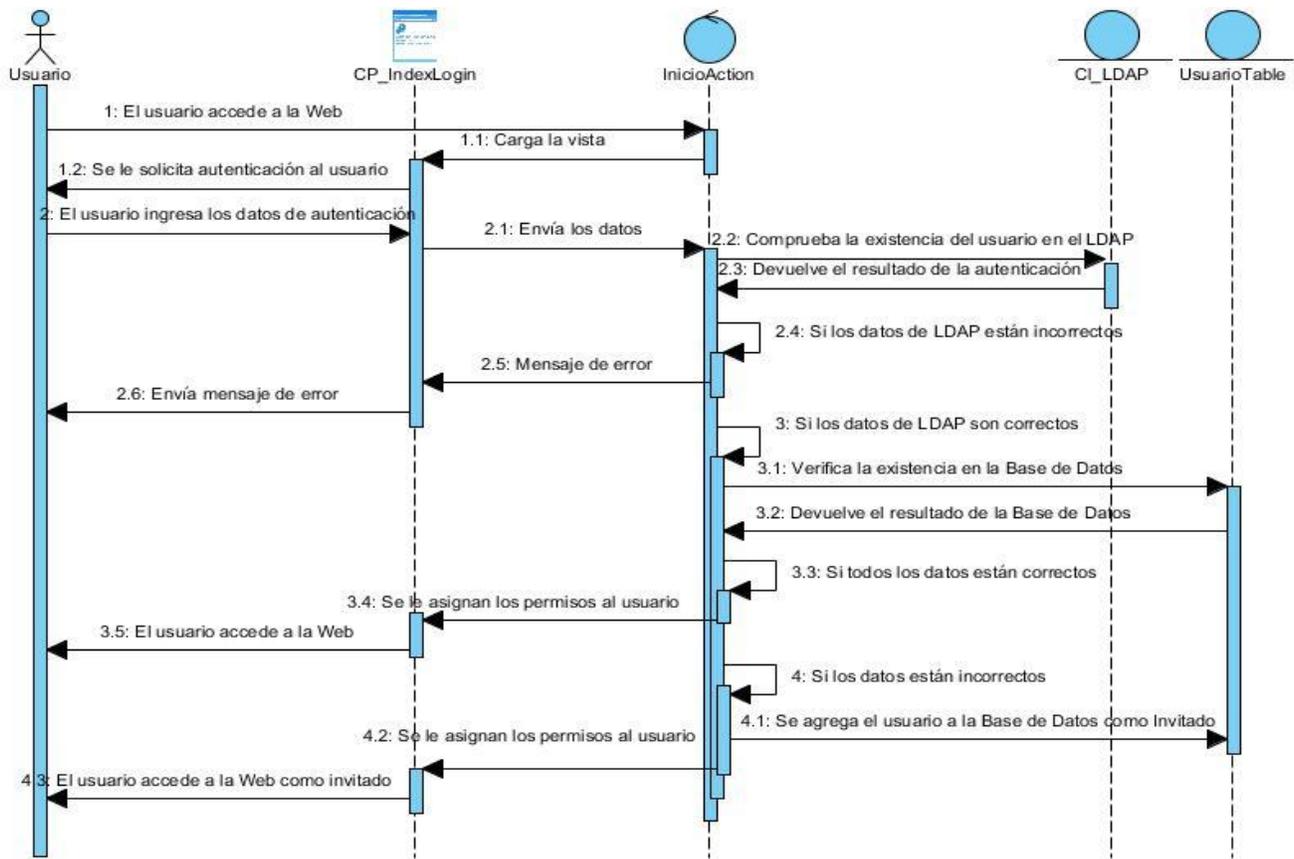


Figura 3.13: Diagrama de Secuencia CU Autenticar Usuario

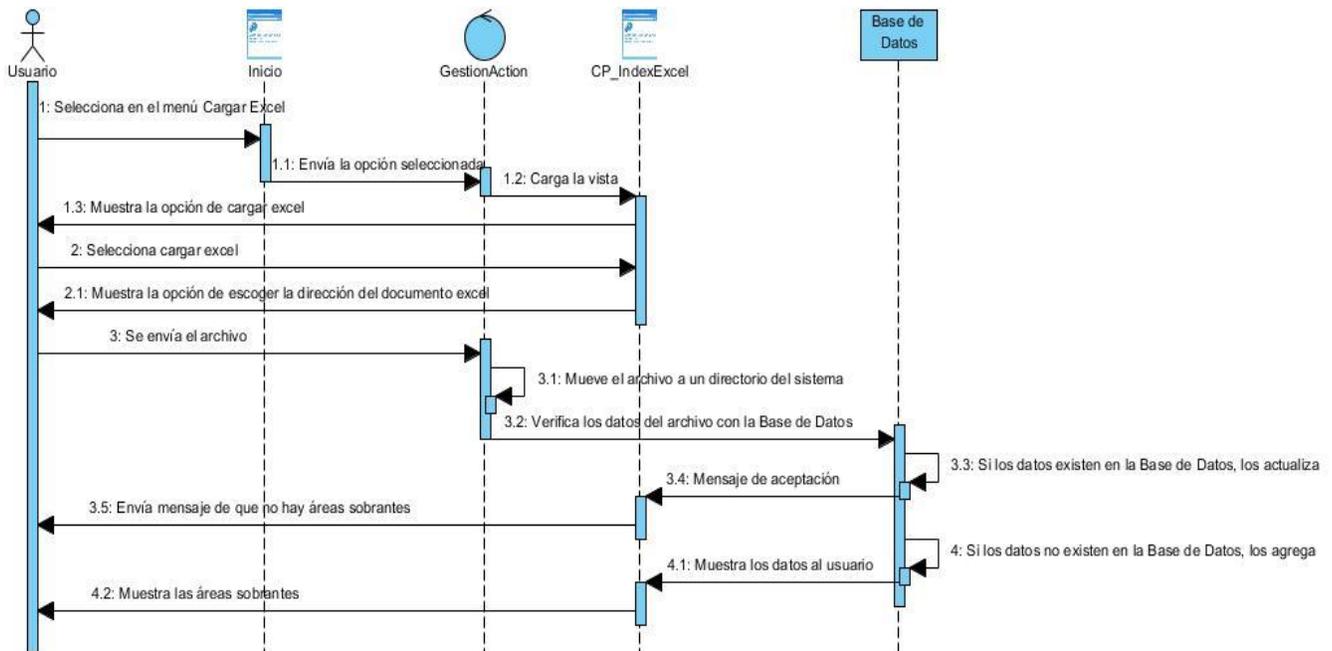


Figura 3.14: Diagrama de Secuencia CU Cargar Documento Excel

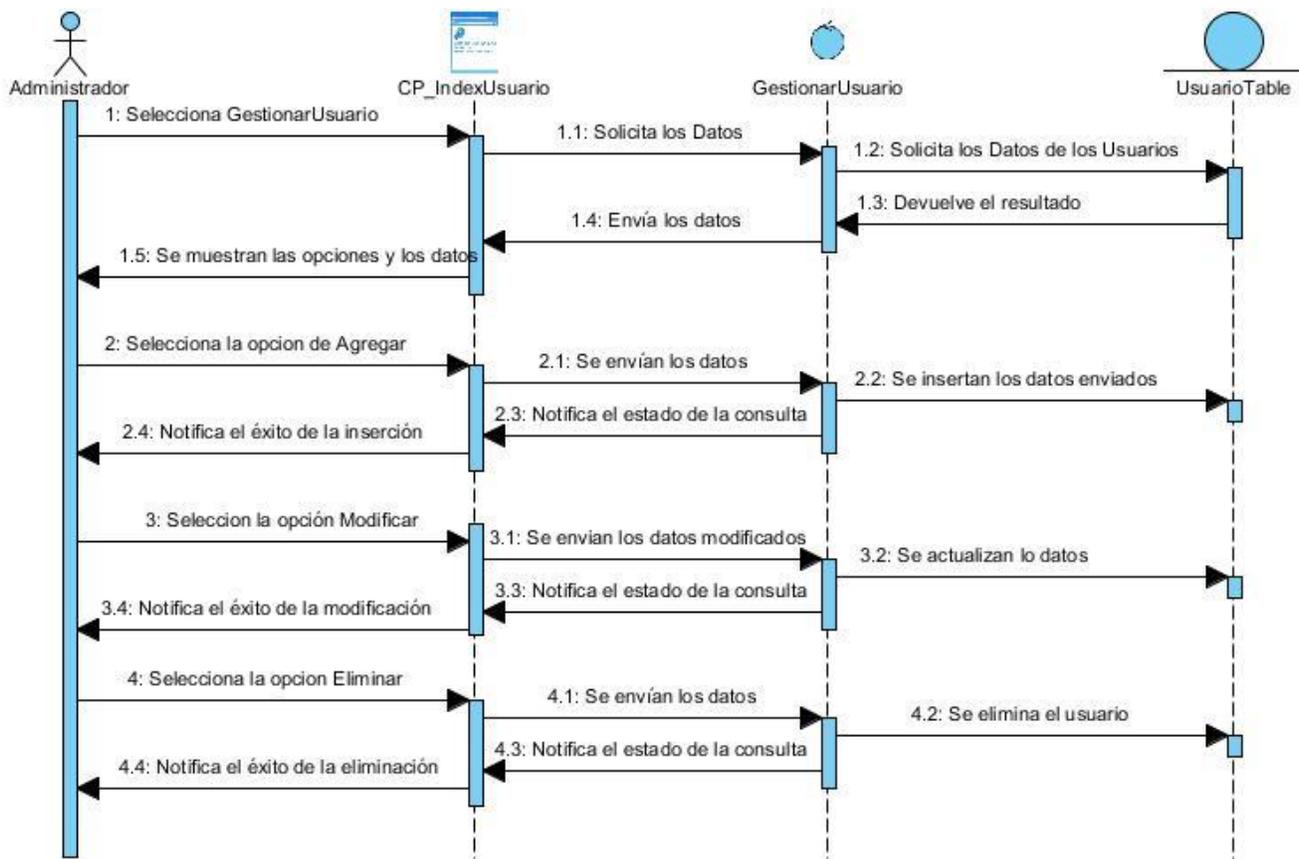


Figura 3.15: Diagrama de Secuencia CU Gestionar Usuario

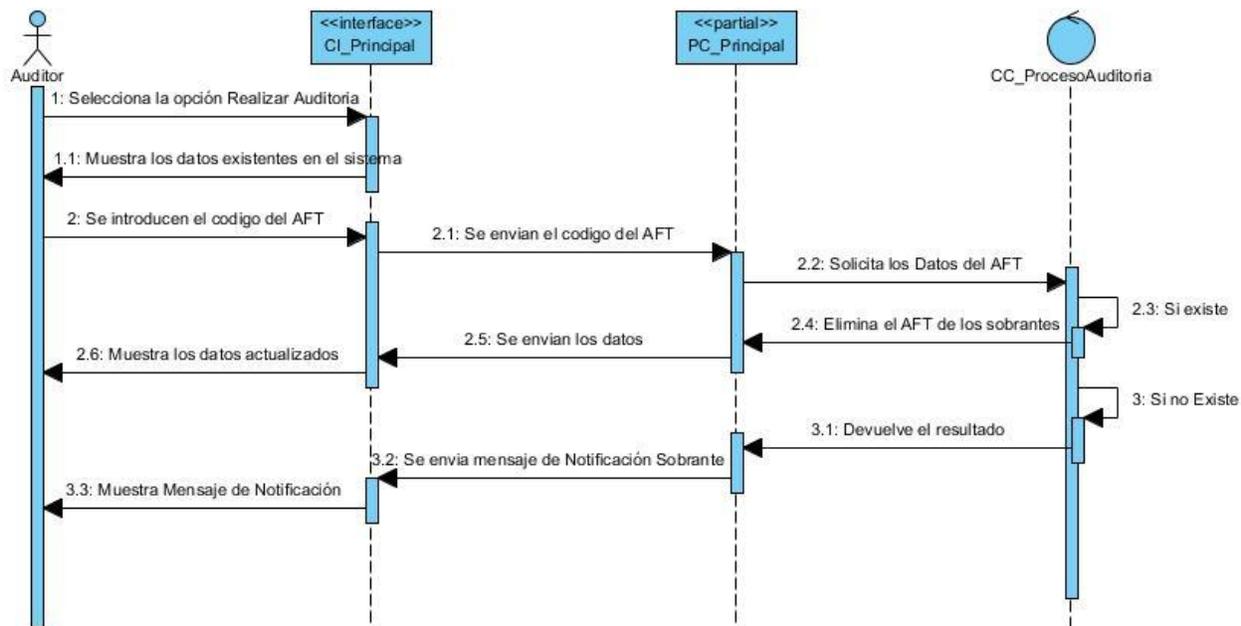


Figura 3.16: Diagrama de Secuencia CU Realizar Auditoría

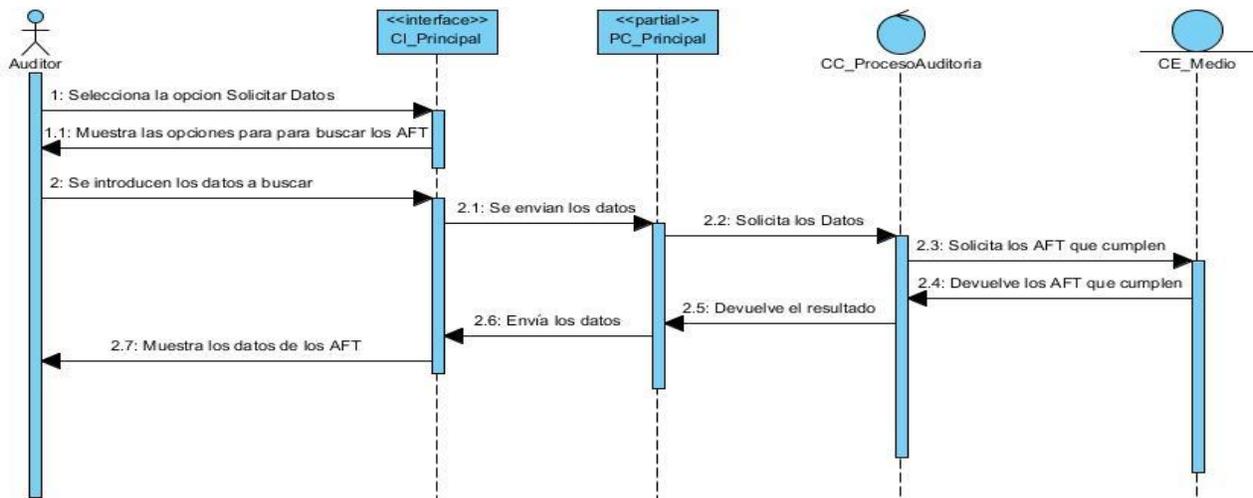


Figura 3.17: Diagrama de Secuencia CU Solicitar Realizar Auditoría

3.6. Diseño de la Base de Datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos correctamente diseñada permite obtener acceso a la información exacta y actualizada.

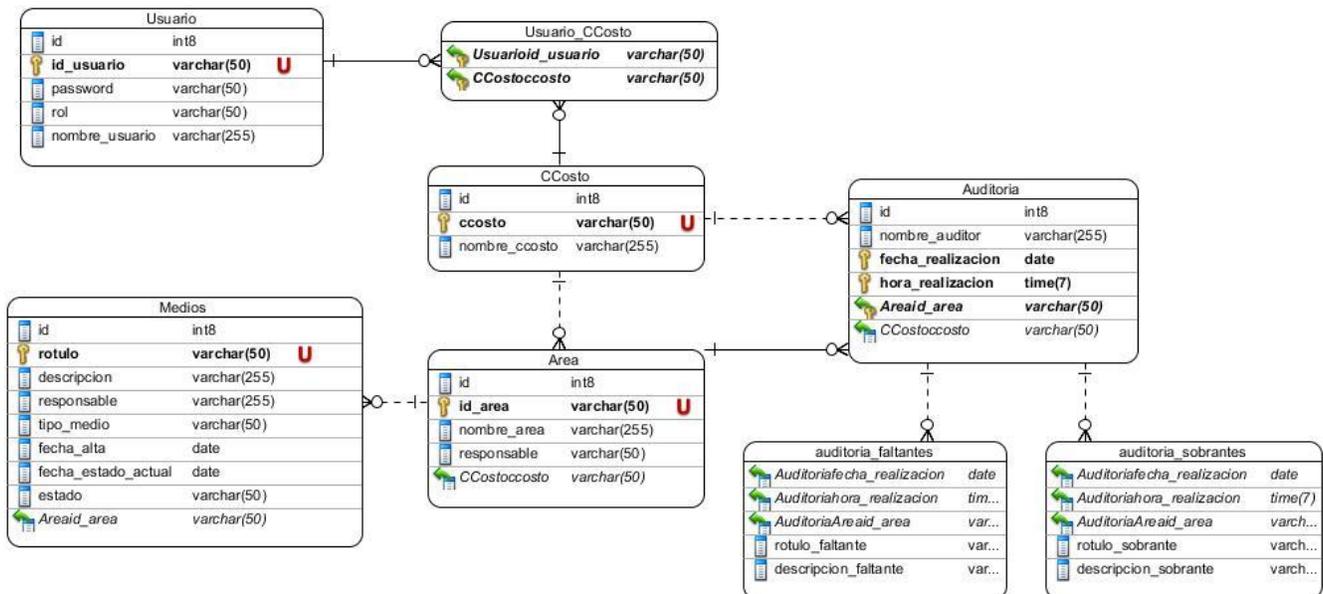


Figura 3.18: Representación Gráfica de la Base de Datos

3.7. Diagrama de Despliegue

Los diagramas de despliegue muestran la configuración en funcionamiento del sistema incluyendo su software y su hardware. Para cada componente de un diagrama es necesario que se deba documentar las características técnicas requeridas, el tráfico de la red, el tiempo de respuesta. Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Los diagramas de despliegue representan a los nodos y sus relaciones. En la Figura 3.19 se muestra el diagrama de despliegue del sistema.



Figura 3.19: Diagrama de Despliegue

Descripción de los nodos físicos del sistema:

- ✓ **PC Cliente:** El nodo representa una PC cliente desde la cual se podrá acceder al sistema por medio de un navegador Web e interactuar con todas las funcionalidades que este brinda.
- ✓ **Servidor Web:** El nodo representa el servidor Web donde estará alojado el sistema así como los componentes almacenados en el mismo.
- ✓ **Servidor Web (WSDL):** El nodo representa el servidor de servicios Web que ofrece la universidad (UCI). Entre los servicios que ofrece se utiliza el servicio de Protocolo Ligerero de Acceso a Directorios LDAP para la autenticación de los usuarios en el sistema.
- ✓ **Servidor de Base de Datos:** El nodo representa el servidor de Base de Datos PostgreSQL en el que estará alojada la base de datos del sistema.
- ✓ **Laptop:** El nodo representa la PC cliente en la que estará el módulo de escritorio a través del cual se realizará las auditorías.
- ✓ **Escáner de Código de Barras:** El nodo representa el escáner conectado a la laptop para realizar la lectura de los códigos de barras situados en cada AFT.

- ✓ **Impresora:** El nodo representa la impresora a través de la cual se imprimen los códigos de barras para hacer adheridos sobre los AFT y los reportes generados por el sistema.

Descripción de los protocolos utilizados:

- ✓ **HTTP:** Protocolo para Transferencia de Hipertexto (*HyperText Transfer Protocol*) Protocolo utilizado para transmitir hipertextos en la Web.
- ✓ **HTTPS:** Protocolo para Transferencia de Hipertexto seguro. Extensión del HTTP para la autenticación y encriptación de datos entre un servidor Web y un navegador Web.
- ✓ **TCP/IP:** Proviene de dos protocolos, Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP) (*Transmission Control Protocol y Internet Protocol*). Forma de comunicación básica que usa el Internet.
- ✓ **USB:** Bus Universal en Serie (*Universal Serial Bus*) Interfaz entre la PC y ciertos dispositivos.
- ✓ **LPT:** Línea terminal de impresión/línea de la impresora (*Line Print Terminal / Line PrinTer*). Es un conector que permite la transmisión de datos desde un dispositivo externo hacia la computadora.

Conclusiones

En este capítulo se concretaron los estilos y arquitectura de la aplicación: Cliente-Servidor. Se describió el patrón arquitectónico Modelo Vista Controlador (MVC). Se elaboraron los diagramas de clases de diseño e interacción de los principales casos de uso, así como el diseño de la BD y el diagrama de despliegue. Una vez modelado los artefactos que genera la metodología OpenUp durante la etapa de desarrollo Análisis y Diseño que dan solución a la situación problemática existente es posible pasar a las siguientes fases: Implementación y Análisis de los Resultados.

Introducción

En el presente capítulo se especifica el modelo de implementación con el objetivo de precisar la estructura y organización del sistema. Se implementan las clases y subsistemas hallados durante el diseño representado en los diagramas de componentes. Además se realiza un análisis de los resultados obtenidos durante el desarrollo del sistema. Así como una ejemplificación de las principales pantallas del sistema con las que interactúa el usuario.

4.1. Modelo de Implementación

El modelo de implementación es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros de código fuente, y otros tipos de ficheros necesarios para la implantación y despliegue del sistema. Describe también, como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación en el lenguaje o lenguajes de programación utilizados, y como dependen de los componentes unos de otros [16].

4.1.1. Diagrama de Componentes

Un diagrama de componentes muestra las dependencias entre los componentes de software. Incluye los clasificadores que los especifican (por ejemplo, clases de implementación) y los artefactos que los implementan, tales como, archivos de código fuente, archivos de código binario, archivos ejecutables, scripts [17].

Los diagramas de componentes describen los elementos físicos del sistema ya sean ejecutables, librerías, clases y las relaciones entre ellos. Para la modelación del diagrama que a continuación se propone primeramente se identificaron los componentes, luego se agruparon los mismos por la arquitectura, en este caso Modelo Vista Controlador (MVC) y se representaron las relaciones entre ellos.

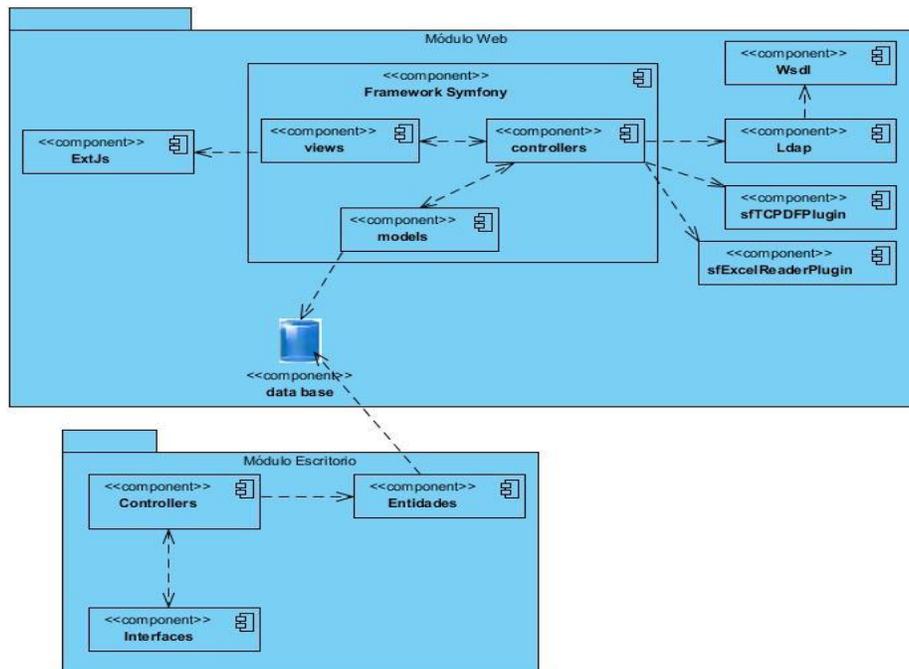


Figura 4.1: Diagrama de Componentes

Descripción de cada componente representado en el diagrama anterior:

Tabla 4.1: Descripción de los Componentes del Sistema

Componente	Propósito
sfTCPDFPlugin	Librerías externas de Symfony que se utilizan para la generación de pdf que incluyen códigos de barras.
sfExcelReaderPlugin	Librerías externas de Symfony que se utilizan para la lectura de documentos Excel.
LDAP	Clase PHP que permite la autenticación LDAP y la integración con Active Directory.
ExtJs	Framework de javascript que se utiliza para la interfaz de usuario.
WsdI	Interfaz pública a los servicios Web de la UCI.
View	Contiene las páginas Web que son vistas por el usuario.
Models	Son clases de PHP que se diseñan para trabajar con información de la Base de Datos.
Controllers	Contiene las clases que determinan cómo se manejan las solicitudes HTTP, y permiten la lógica del negocio.
Base de Datos	Representa la Base de Datos.

A continuación se muestra en cuatro diagramas cada uno de los componentes generados que se incluyen en las capas definidas del patrón MVC.

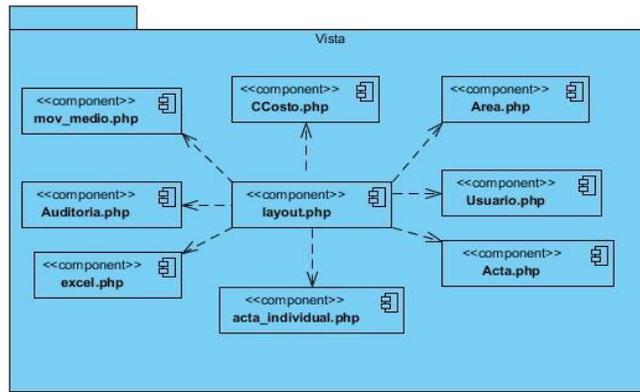


Figura 4.2: Componentes Vistas

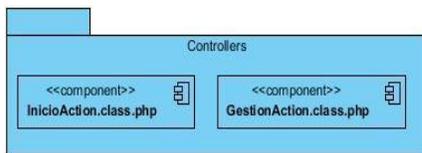


Figura 4.3: Componentes Controladores

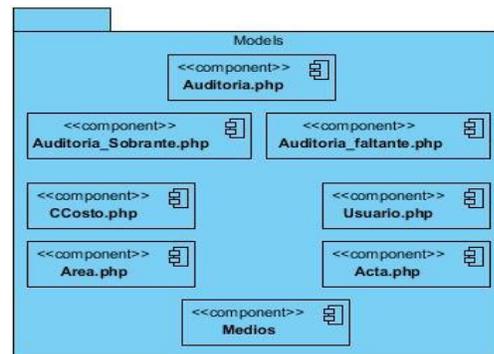


Figura 4.4: Componentes Modelos

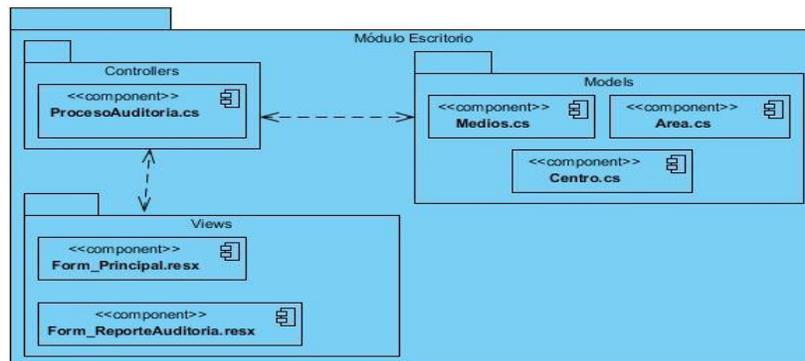


Figura 4.5: Componentes Escritorio

4.2. Pantallas del Sistema

Las pantallas de la aplicación son fotos tomadas durante su funcionamiento. Estas imágenes muestran parte de los resultados obtenidos con el desarrollo de esta investigación:

Módulo Web

En la Figura 4.6 se muestra la interfaz cargar documento excel que a través del rol Administrador se puede cargar el documento con los datos pertinentes.

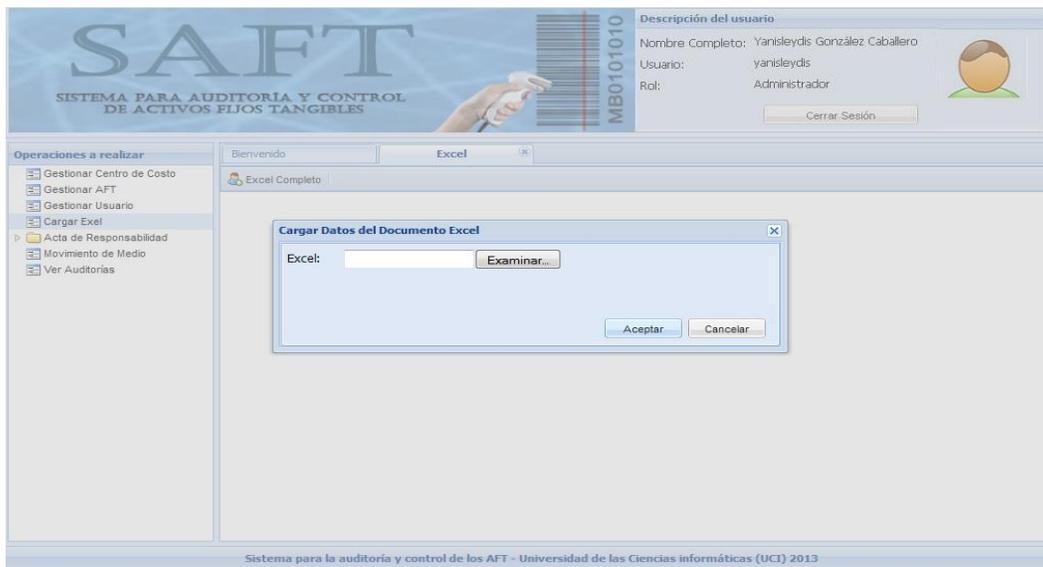


Figura 4.6: Interfaz del Módulo Cargar Documento Excel

En la Figura 4.7 se muestra la interfaz gestionar activo fijo tangible que a través del rol Administrador se gestionan todos los medios que pertenezcan a un área determinada. Esta vista tiene la opción de generar un documento pdf con los códigos de barras a través del rótulo de todos los AFT o de un AFT determinado.

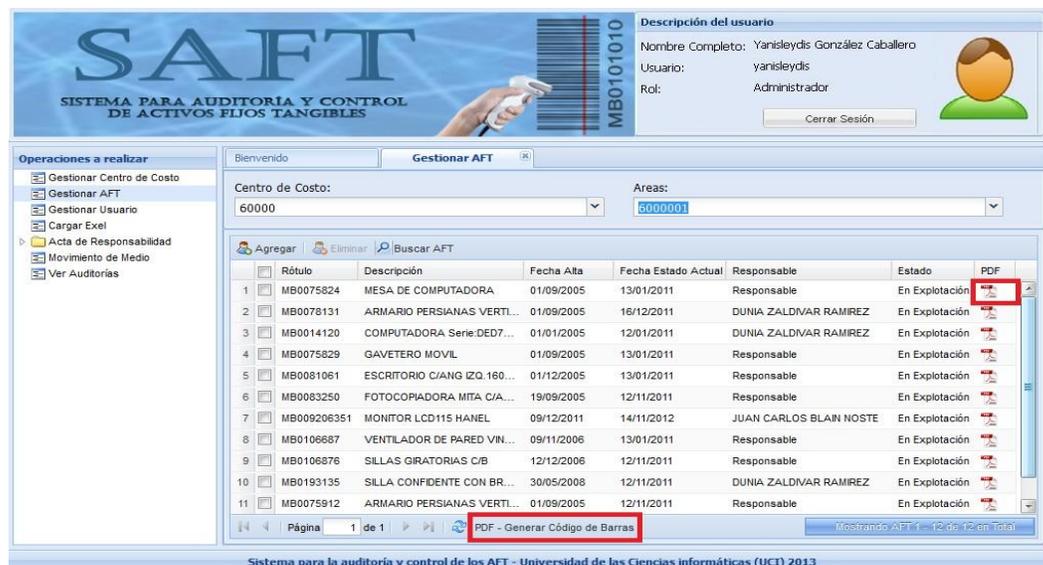


Figura 4.7: Interfaz del Módulo Gestionar Activo Fijo Tangible

En la Figura 4.8 se muestra la interfaz gestionar usuario que a través del rol Administrador permite gestionar los usuarios con su rol y a los centros de costos que ellos podrán visualizar.

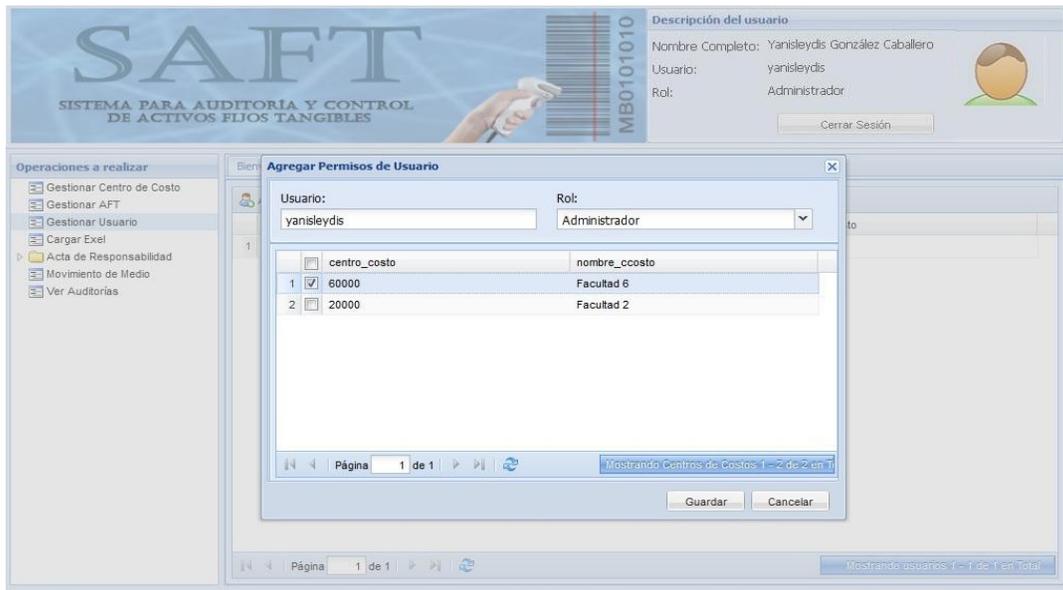


Figura 4.8: Interfaz del Módulo Gestionar Usuario

Módulo Escritorio

En la Figura 4.9 se muestra la interfaz solicitar realizar auditoría, que a través del rol Auditor realiza la solicitud del centro de costo y el área que desea auditar. Al hacer clic sobre el botón solicitar datos se muestra la información de los AFT.



Figura 4.9: Interfaz del Módulo Solicitar Realizar Auditoría

En la Figura 4.10 se muestra la interfaz realizar auditoría, que a través del rol Auditor se obtienen los rótulos de todos los AFT de un área seleccionada anteriormente. En esta vista se realiza la lectura de

todos los rótulos y realiza la comparación con el objetivo de obtener los AFT faltantes y sobrantes.

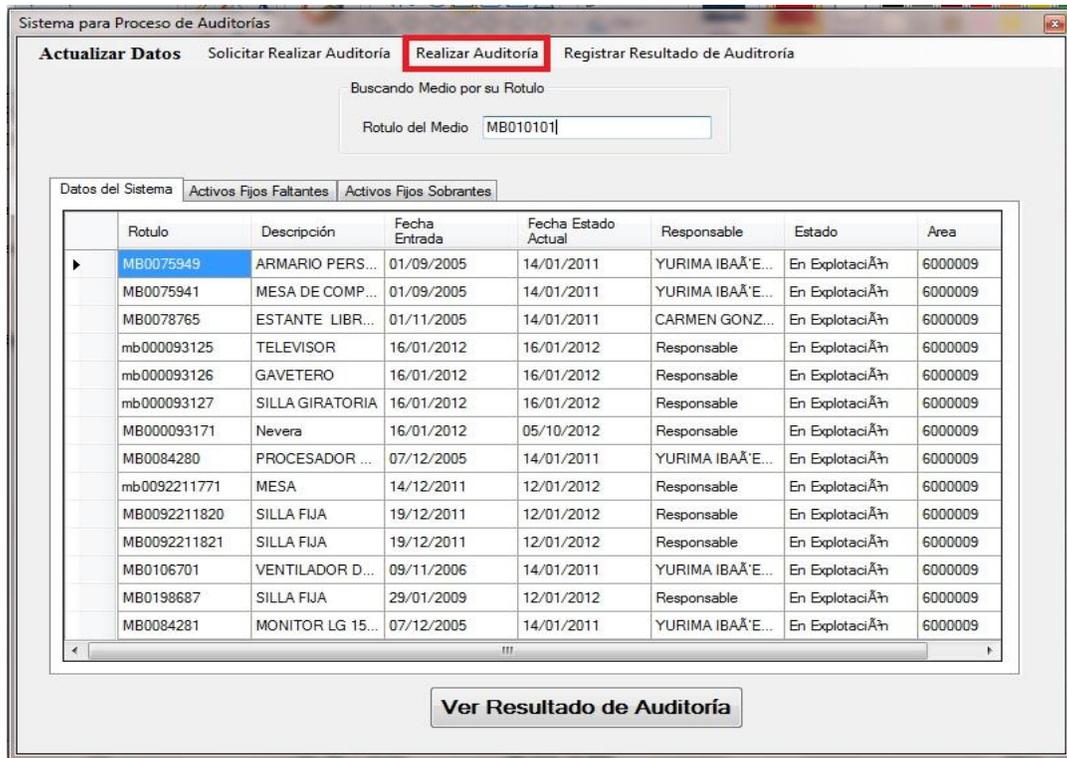


Figura 4.10: Interfaz del Módulo Realizar Auditoría

4.3. Determinación de Posibilidad de Éxito Utilizando Criterio de Experto

Se utilizó Delphi, para realizar la valoración técnica de la solución. Este método permite medir el índice de gestión en el proceso de auditoría y control según la experiencia de los expertos de acuerdo a criterios definidos dando una posible validez de la propuesta después de su aplicación.

Se realizaron los siguientes pasos:[18]

Paso 1: Selección de los Expertos

La selección de los expertos se basó en los siguientes criterios:

Que el área del conocimiento en la que se mueva esté enmarcada en “Ingeniería en Informática” o sus carreras afines con dominio en los siguientes temas.

- ✓ Especialista en el trabajo con sistemas de gestión.
- ✓ Especialista en el trabajo de gestión económica.

- ✓ Especialista en el control de la calidad.

Se cumplió con los requisitos definidos, se seleccionaron 9 expertos, los cuales se relacionan en la tabla. [\(Ver anexo 7\)](#).

Paso 2: Elaboración de los Criterios a Evaluar

Se agrupan de acuerdo a las características de la propuesta.

Grupo 1: Criterios de mérito científico.

1. Contribución científica de la solución.

Grupo 2: Criterios de creación.

2. Correspondencia con el negocio de estudio.
3. Ahorro de recursos materiales.
4. Legalidad en el proceso.

Grupo 3: Criterios de seguridad.

5. Rendimiento.
6. Integridad.
7. Fiabilidad.
8. Disponibilidad

Grupo 4: Criterios de impacto.

9. Impacto en el área para la cual está destinada la propuesta de solución.
10. Gestión del proceso.
11. Tiempo de ejecución.

Grupo 5: Criterios de usabilidad.

12. Facilidad del usuario para realizar las operaciones.
13. Facilidad de administración.
14. Trazabilidad de los datos con respecto a estados anteriores.
15. Adquisición de los datos.

Paso 3: Asignación de Valor Relativo a Cada Criterio

Se entenderá por valor relativo a la importancia que se le atribuye a la evaluación de un criterio con respecto al resto de los criterios. Para ello se le pide a cada experto que emita su opinión acerca de la importancia que tiene cada indicador con relación a los demás para el proyecto a evaluar. La sumatoria de estos valores relativos debe ser igual a 100. [\(Ver anexo 8\)](#)

EP: Valor relativo de cada criterio

$$EP = (\sum V_{ECn} / E)$$

Tabla 4.2: Asignación de Peso a los de Indicadores

C/E	E1	E2	E3	E4	E5	E6	E7	E8	E9	ΣE	EP	ΣE-ΣΣE/C	(ΣE-ΣΣE/C) ²
C1	8	8	7	8	8	7	8	8	8	70	7,7778	70	4900
C2	6	4	6	4	4	6	7	5	4	46	5,1111	46	2116
C3	7	9	7	9	9	6	8	8	9	72	8,0000	72	5184
C4	5	4	6	5	4	6	11	5	4	50	5,5556	50	2500
C5	6	4	6	5	4	6	6	5	4	46	5,1111	46	2116
C6	5	4	6	4	8	6	6	5	4	48	5,3333	48	2304
C7	8	9	8	9	8	8	6	6	9	71	7,8889	71	5041
C8	8	9	7	9	7	6	5	8	9	68	7,5556	68	4624
C9	5	4	6	5	4	6	11	6	4	51	5,6667	51	2601
C10	6	5	6	5	4	6	6	6	4	48	5,3333	48	2304
C11	5	5	6	5	8	6	6	6	4	51	5,6667	51	2601
C12	7	8	7	8	9	8	5	7	9	68	7,5556	68	4624
C13	8	10	7	8	7	7	5	8	9	69	7,6667	69	4761
C14	8	8	7	8	9	8	5	8	10	71	7,8889	71	5041
C15	8	9	8	8	7	8	5	9	9	71	7,8889	71	5041
ΣΣE/C										900	100,0000	9	
S = Σ (ΣE - ΣΣE/C)²													55758

Verificación de la concordancia de los expertos.

- ✓ Se calcula el coeficiente de concordancia de Kendall, se conoce la dispersión:

$$W = S / (E^2(C^3 - C) / C)$$

- ✓ Se calcula el Chi cuadrado real, se sabe el valor del coeficiente de concordancia de Kendall:

$$X^2 = E * (C - 1) * W$$

Tabla 4.3: Cálculo de Concordancia de Kendall

S	E ²	C ³ - C	E ² (C ³ - C)	E ² (C ³ - C) / C	W = S / (E ² (C ³ - C) / C)	X ² = E * (C - 1) * W
55758	81	3360	272160	4082400	0,0137	0,9834

Se compara el Chi cuadrado real calculado con el que se obtiene de la Tabla de Distribución Chi Cuadrado ([Ver Anexo 9](#)), se toma $1 - \alpha = 0.99$ donde $\alpha = 0.01$ es el error permisible.

Si se cumple que el X^2 real $< X^2$ (α , c-1) puede decirse que existe concordancia en el trabajo de los expertos.

Según el resultado del cálculo en la tabla anterior: $0.98 < 29.14$ por tanto, existe concordancia entre los expertos.

Paso 4: Habilita Acceso a los Expertos a la Aplicación y su Documentación

Se entrega a los expertos el acceso al sistema y una documentación breve sobre la solución, para que se documenten sobre el tema y expresen sus criterios. Los expertos conceden pesos de cero (valor mínimo) a diez (valor máximo) a cada uno de los criterios establecidos y una apreciación cualitativa con una clasificación final del proyecto en alta, media, baja, y fracaso.

Tabla 4.4: Escala Valorativa de 0 a 10 para Expresar el Nivel de Satisfacción

Valor	Impacto
0	Fracaso
1	Muy Pésimo
2	Pésimo
3	Muy Mal
4	Mal
5	Regular
6	Adecuado
7	Optimo
8	Bien
9	Muy Bien
10	Excelente

Paso 5: Construye la Tabla Criterio Contra Evaluación

Después de recibir los valores del peso relativo de cada criterio se construye la tabla correspondiente, donde $C_{1...15}$ son los criterios a evaluar y $E_{1...9}$ la evaluación de los expertos emitidos mediante encuestas. ([Ver Anexo 10](#))

Tabla 4.5: Peso Otorgado por los Expertos

C/E	E1	E2	E3	E4	E5	E6	E7	E8	E9	ΣE	$E_p = \Sigma E/9$
C1	9	10	8	9	8	7	8	9	10	78	8,6667
C2	4	4	5	4	5	6	7	5	4	44	4,8889
C3	9	10	8	9	8	6	7	9	10	76	8,4444
C4	5	4	6	5	5	7	11	4	4	51	5,6667
C5	4	4	5	5	5	6	6	5	4	44	4,8889
C6	5	4	6	4	8	6	6	4	4	47	5,2222
C7	9	9	8	9	9	9	5	9	10	77	8,5556
C8	9	10	8	9	6	7	4	9	10	72	8,0000
C9	4	4	5	4	5	6	7	5	4	44	4,8889
C10	9	9	8	9	8	6	7	9	10	75	8,3333
C11	5	4	6	5	5	7	11	5	4	52	5,7778
C12	5	4	6	5	5	6	6	4	4	45	5,0000
C13	5	4	6	5	8	6	5	5	4	48	5,3333
C14	9	10	7	9	9	9	5	9	9	76	8,4444
C15	9	10	8	9	6	6	5	9	9	71	7,8889
Total										100	1

Verificación de la concordancia de los expertos

Se verifica el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X^2). A continuación el procedimiento:

- ✓ Sea C el número de criterios que van a evaluarse y E el número de expertos que realizan la evaluación.
- ✓ Para cada criterio se determina la ΣE que representa la sumatoria del peso dado por cada experto, E_p que es la puntuación promedio de los pesos correspondientes a cada criterio.
- ✓ Se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión S por la expresión: $S = \Sigma (\Sigma E - \Sigma \Sigma E/C)^2$.

Tabla 4.6: Cálculo de la Dispersión (S) para hallar la Concordancia entre Expertos

E/ C	ΣE	$\Sigma E/C$	$\Sigma E - \Sigma \Sigma E/C$	$(\Sigma E - \Sigma \Sigma E/C)^2$
C ₁	78	5.20	18.00	324.00
C ₂	44	2.93	-16.00	256.00
C ₃	76	5.07	16.00	256.00
C ₄	51	3.40	-9.00	81.00

C ₅	44	2.93	-16.00	256.00
C ₆	47	3.13	-13.00	169.00
C ₇	77	5.13	17.00	289.00
C ₈	72	4.80	12.00	144.00
C ₉	44	2.93	-16.00	256.00
C ₁₀	75	5.00	15.00	225.00
C ₁₁	52	3.47	-8.00	64.00
C ₁₂	45	3.00	-15.00	225.00
C ₁₃	48	3.20	-12.00	144.00
C ₁₄	76	5.07	16.00	256.00
C ₁₅	71	4.73	11.00	121.00
ΣΣE/C	900	60.0		
S = Σ (ΣE - ΣΣE/C) ²				3066

Se calcula el coeficiente de concordancia de Kendall, se conoce la dispersión:

$$W = S / (E^2 (C^3 - C) / C)$$

✓ Se calcula el Chi cuadrado real, se sabe el valor del coeficiente de concordancia de Kendall:

$$X^2 = E * (C - 1) * W$$

Tabla 4.7: Cálculo de Concordancia de Kendall

S	E ²	C ³ - C	E ² (C ³ - C)	E ² (C ³ - C)/C	W = S / (E ² (C ³ - C) / C)	X ² = E * (C - 1) * W
3066	81	3360	272160	4082400	0.0008	0.0541

Se compara el Chi cuadrado real calculado con el que se obtiene de la Tabla de Distribución Chi Cuadrado ([Ver Anexo 10](#)), se toma 1 - α = 0.99 dónde α = 0.01 es el error permisible.

Si se cumple que el X² real < X² (α, c-1) puede decirse que existe concordancia en el trabajo de los expertos.

Según el resultado del cálculo en la tabla anterior: 0.054 < 29.141 por tanto, existe concordancia entre los expertos.

Paso 6: Definición del Peso Relativo de los Criterios

Posteriormente se identifica el peso (P) relativo de cada criterio y se calcula el índice del peso relativo

de cada indicador se define a partir de la evaluación anterior a través de la siguiente expresión:

$$P = (EP_n / 100)$$

P: peso relativo

EP_n: Media aritmética de los valores por cada criterio

Tabla 4.8: Definición del Peso Relativo de cada Criterio

C/E	E1	E2	E3	E4	E5	E6	E7	E8	E9	ΣE	EP	P
C1	9	10	8	9	8	7	8	9	10	78	8.6667	0.09
C2	4	4	5	4	5	6	7	5	4	44	4.8889	0.05
C3	9	10	8	9	8	6	7	9	10	76	8.4444	0.08
C4	5	4	6	5	5	7	11	4	4	51	5.6667	0.06
C5	4	4	5	5	5	6	6	5	4	44	4.8889	0.05
C6	5	4	6	4	8	6	6	4	4	47	5.2222	0.05
C7	9	9	8	9	9	9	5	9	10	77	8.5556	0.09
C8	9	10	8	9	6	7	4	9	10	72	8.0000	0.08
C9	4	4	5	4	5	6	7	5	4	44	4.8889	0.05
C10	9	9	8	9	8	6	7	9	10	75	8.3333	0.08
C11	5	4	6	5	5	7	11	5	4	52	5.7778	0.06
C12	5	4	6	5	5	6	6	4	4	45	5.0000	0.05
C13	5	4	6	5	8	6	5	5	4	48	5.3333	0.05
C14	9	10	7	9	9	9	5	9	9	76	8.4444	0.08
C15	9	10	8	9	6	6	5	9	9	71	7.8889	0.08

Paso 7: Índice de Gestión

Posteriormente se identifica el peso (P) relativo de cada criterio y se calcula el Índice de Gestión (IG) de la solución integral[19].

- ✓ Se calcula el peso de cada criterio (P), se conoce el número de expertos que realizan la evaluación E y la sumatoria de las puntuaciones de cada criterio (C).
- ✓ Se conoce el peso de cada criterio y la cantidad de expertos se puede obtener el valor de $P \times C_p$, donde C_p , es el criterio promedio concebido por los expertos en escala de 5.
- ✓ Con el valor anterior se calcula el Índice de Gestión (IG). $IG = \Sigma (P \times C_p) / 10$.

Tabla 4.9: Calificación de cada Criterio e Índice de Gestión

Criterio	C _p	P	P x C _p
C ₁	8.67	0.0867	0.7511
C ₂	4.89	0.0489	0.2390
C ₃	8.44	0.0844	0.7131
C ₄	5.67	0.0567	0.3211
C ₅	4.89	0.0489	0.2390
C ₆	5.22	0.0522	0.2727
C ₇	8.56	0.0856	0.7320
C ₈	8.00	0.0800	0.6400
C ₉	4.89	0.0489	0.2390
C ₁₀	8.33	0.0833	0.6944
C ₁₁	5.78	0.0578	0.3338
C ₁₂	5.00	0.0500	0.2500
C ₁₃	5.33	0.0533	0.2844
C ₁₄	8.44	0.0844	0.7131
C ₁₅	7.89	0.0789	0.6223
Σ (P x C_p)	100.0000	1	7.0452
IG			0.7045

Paso 8: Determina la Probabilidad de Éxito

Por último se determina la probabilidad de éxito de la propuesta, donde se ubica el IG calculado anteriormente en rangos que están predefinidos en la Tabla 4.10, en dependencia de su ubicación será la probabilidad de éxito que tenga la propuesta.

Tabla 4.10: Rangos Predefinidos para el Índice de Gestión

0.7 < IG	Alto Índice de Gestión
0.5 < IG < 0.7	Medio Índice de Gestión
0.3 < IG < 0.5	Bajo Índice de Gestión
IG < 0.3	Insuficiente Índice de Gestión

El Índice de Gestión calculado es 0.71 lo que significa que existe alto índice de gestión de la propuesta después de su aplicación.

4.4. Prueba de Interfaz de Usuario

Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente.

En todo proceso de desarrollo de aplicaciones es indispensable la presencia de un proceso de Pruebas de Software para garantizar así el buen funcionamiento y la calidad del producto final. Estas tienen como objetivo general verificar y validar un software, independientemente de las características y el entorno donde se desarrollen, además de los recursos y los factores vinculados al proceso de desarrollo.

Pruebas de Caja Negra

Son las pruebas que se realizan de forma que se compruebe que cada función es operativa. Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Se intenta encontrar con estas pruebas:

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores en estructuras de datos o en accesos a las Bases de Datos externas.
4. Errores de rendimiento.
5. Errores de inicialización y terminación.

Se decide realizar este método de prueba ya que se llevan a cabo sobre la interfaz del software. O sea, pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Caso de Prueba – Autenticar Usuario

ID	Escenario	Variable	Respuesta del Sistema	Resultado de la Prueba
----	-----------	----------	-----------------------	------------------------

C.P 1.1	Campos vacíos	usuario="	El sistema debe mostrar un mensaje de error: “Debe introducir el nombre de usuario” , según la variable que esté vacía.	El sistema muestra el mensaje de error: “Debe introducir el nombre de usuario” .
C.P 1.2	Campos vacíos	contraseña="	El sistema debe mostrar un mensaje de error: “Debe introducir la contraseña para este usuario” , según la variable que esté vacía.	El sistema muestra el mensaje de error: “Debe introducir la contraseña para este usuario” .

Pantallas de Casos de Prueba – Autenticar Usuario



Figura 4.11: Escenario Campos Vacíos

Caso de Prueba – Gestionar Usuario

ID	Escenario	Variable	Respuesta del Sistema	Resultado de la Prueba
C.P 2.1	Campos vacíos	usuario="	El sistema debe mostrar un mensaje de error: “Debe llenar los campos obligatorios” , según la variable que esté vacía.	El sistema muestra el mensaje de error: “Debe llenar los campos obligatorios” .

Pantalla del Caso de Prueba – Gestionar Usuario

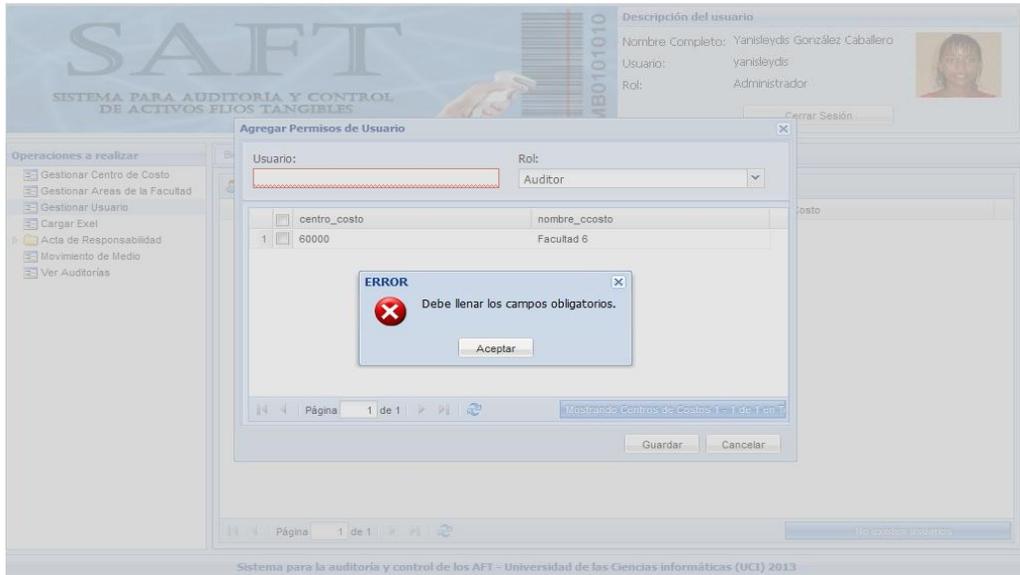


Figura 4.12: Escenario Campos Vacíos

Caso de Prueba – Cargar Documento Excel

ID	Escenario	Variable	Respuesta del Sistema	Resultado de la Prueba
C.P 3.1	Cargar Excel	Extensión del archivo incorrecto	El sistema debe mostrar el mensaje: “El archivo que intenta cargar no es un documento Excel” .	El sistema muestra el mensaje: “El archivo que intenta cargar no es un documento Excel” .

Pantalla del Caso de Prueba – Cargar Documento Excel

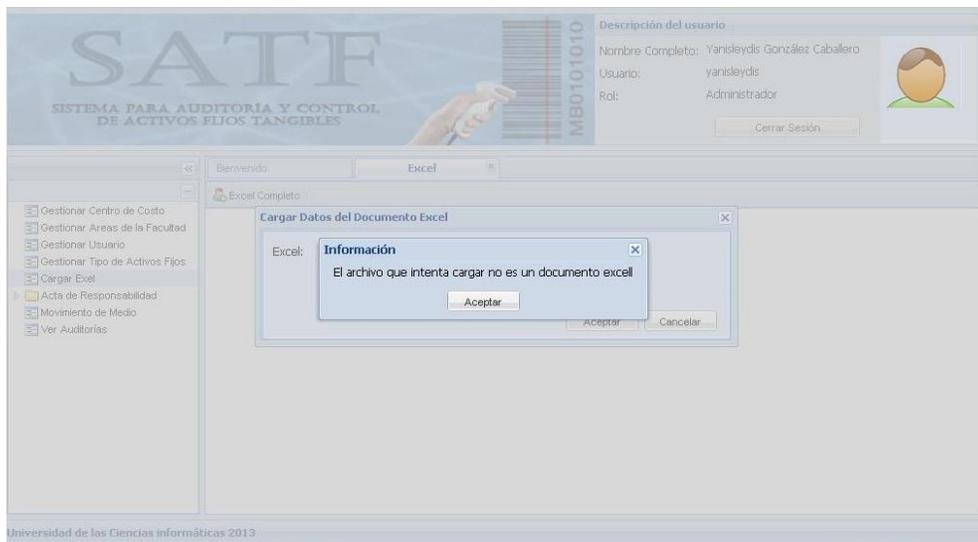


Figura 4.13: Escenario Cargar Excel

Caso de Prueba – Generar Modelo de Movimiento de Medios

ID	Escenario	Variable	Respuesta del Sistema	Resultado de la Prueba
C.P 4.1	Movimiento de Medios	Área de entrega y recibo igual	El sistema debe mostrar el mensaje: “El área que entrega y recibe no puede ser la misma. Seleccione otra área” .	El sistema debe mostrar el mensaje: “El área que entrega y recibe no puede ser la misma. Seleccione otra área” .

Pantallas de Casos de Prueba – Generar Modelo de Movimiento de Medios

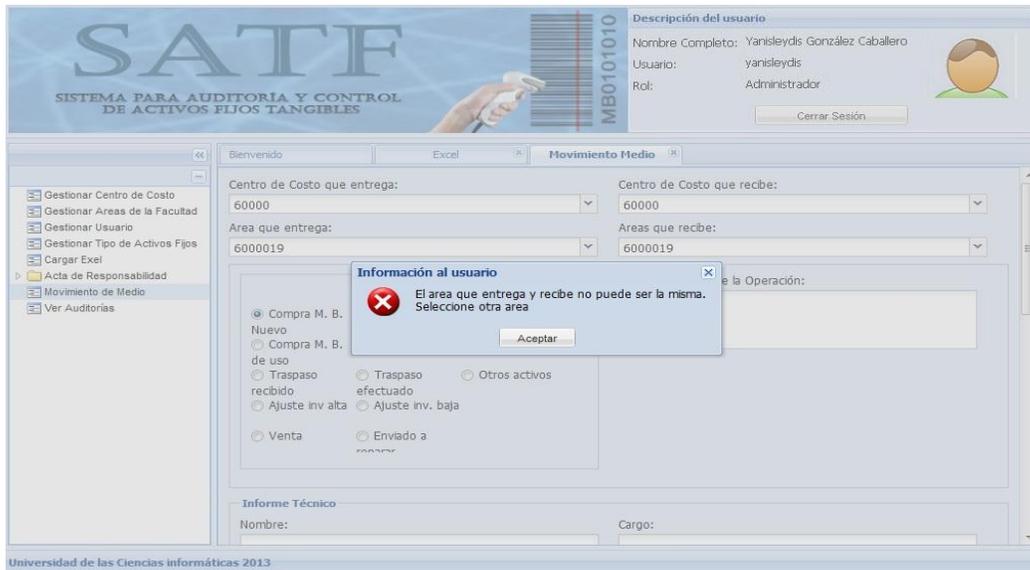


Figura 4.14: Escenario Movimiento de Medios

Caso de Prueba – Realizar Auditoría

ID	Escenario	Variable	Respuesta del Sistema	Resultado de la Prueba
C.P 5.1	Realizar Auditoría	Formato de número de inventario no válido.	El sistema debe mostrar el mensaje: “El número de inventario que se introdujo no posee un formato válido” .	El sistema muestra el mensaje: “El número de inventario que se introdujo no posee un formato válido” .

Pantalla del Caso de Prueba – Realizar Auditoría

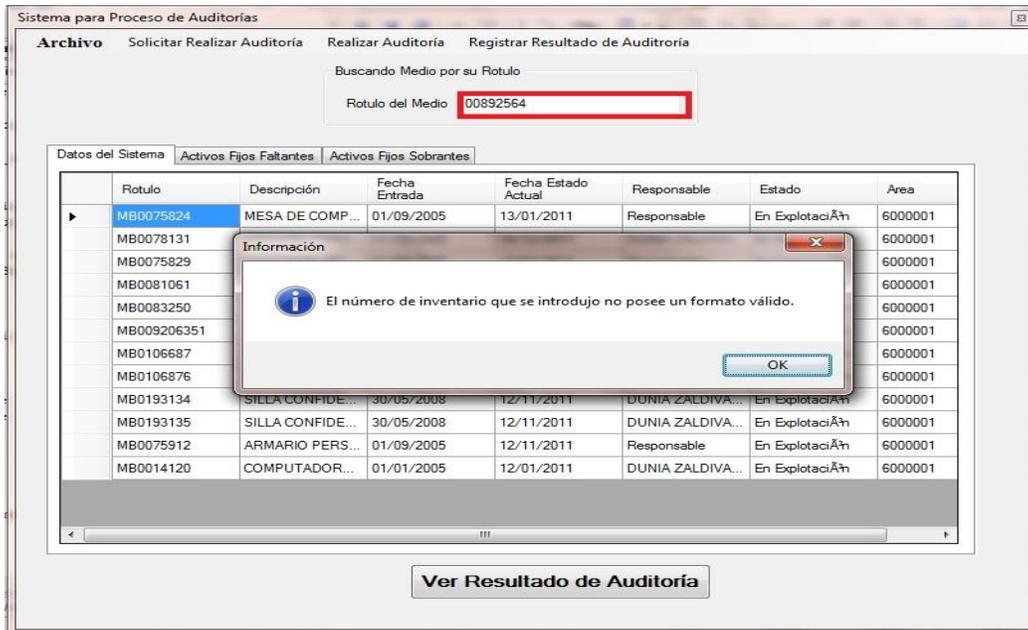


Figura 4.15: Escenario Realizar Auditoría

Caso de Prueba – Solicitar Datos

ID	Escenario	Variable	Respuesta del Sistema	Resultado de la Prueba
C.P 6.1	Solicitar Datos	Área de Responsabilidad="	El sistema debe mostrar el mensaje “Debe seleccionar su Centro de Costo y el Área” .	El sistema muestra el mensaje “Debe seleccionar su Centro de Costo y el Área” .

Pantalla del Caso de Prueba – Solicitar Datos



Figura 4.16: Escenario Solicitar Datos

Conclusiones

Al concluir el presente capítulo se logró completar la fase de implementación y fueron concretados los diagramas de componentes. Se mostraron algunas imágenes o pantallas principales del sistema durante su funcionamiento. Fue determinado la posibilidad de éxito del sistema según el criterio de expertos, que arrojó resultados satisfactorios. Se constituyeron escenarios de prueba a la interfaz de usuario con el objetivo de corroborar el correcto funcionamiento del sistema.

Conclusiones

Dada la necesidad de contribuir a la gestión del proceso de auditoría y control de los Activos Fijos Tangibles de la Facultad 6 de la UCI, se realizó un estudio de las herramientas para su informatización. Esto arrojó como resultado la inexistencia de una aplicación para la gestión del proceso de auditoría y control de los Activos Fijos Tangibles. Por lo que se decide crear un sistema para obtener mejores resultados y garantizar la eficiencia laboral.

Para satisfacer las necesidades del cliente se realiza el análisis y diseño de la aplicación, basada en la metodología de desarrollo OpenUp. Con lo que se logró un enfoque centralizado en la arquitectura, requisitos del sistema, casos de uso, escenarios, patrones y estilos.

Con la utilización del método Delphi se pudo demostrar el cumplimiento de los objetivos trazados, y se determinó el índice de gestión de la propuesta donde se obtuvo un alto nivel según el consenso de los expertos.

Con la implementación de un sistema informático que utiliza la generación y recuperación de códigos de barra se logró contribuir a la gestión del proceso de auditoría y control de los Activos Fijos Tangibles. Se alcanzó una disminución del consumo de materiales y se digitalizó la gestión de los recursos humanos. Cumpliéndose de esta forma el objetivo general trazado desde inicio de la investigación.

Recomendaciones

Al finalizar este trabajo se recomienda:

- ✓ Podría prolongarse hacia otras áreas de la Universidad y así apoyar la mejora del proceso en la Institución.
- ✓ Continuar con el proceso de desarrollo con el fin de agregarle nuevas funcionalidades y reportes según las necesidades que surjan.
- ✓ Publicar la documentación para que futuros investigadores puedan tomarla como referencia para nuevos desarrollos.

Referencias Bibliográficas

1. RUZ, R. *Discurso pronunciado por el General de Ejército Raúl Castro Ruz*. 2009, [En línea]. [Consultado el: 17 de octubre de 2012], Disponible en: <http://www.cuba.cu/gobierno/rauldiscursos/2009/esp/r010809e.html>
2. NACIONAL, A. *Contraloría General de la República de Cuba*. 2009, [En línea]. [Consultado el: 17 de octubre de 2012], Disponible en: http://www.contraloria.cu/index.php?option=com_content&view=article&id=84&Itemid=22
3. MILLONES, L. *Los Activos Fijos*. 2012, [En línea]. [Consultado el: 22 de octubre de 2012], Disponible en: <http://www.bibliomaster.com/pdf/2363.pdf>
4. AUDITORÍA DE INFORMATICA. *Conceptos generales de Auditoría*. 2010, [En línea]. [Consultado el: 22 de octubre de 2012], Disponible en: <http://auditoriadeinformatica.blogspot.es/>
5. RESTREPO, C. *Manual de Control Interno E.S.E. Carisma*. 2011, [En línea]. [Consultado el: 23 de octubre de 2012], Disponible en: <http://www.esecarisma.gov.co/esecarisma/downloads/procesos/juridica%20y%20control%20interno/MANUAL%20DE%20CONTROL%20INTERNO%20Y%20DE%20CALIDAD%20DE%20LA%20ESE%20CARISMA.pdf>
6. GONZÁLEZ, I. *MEBUS*. 2003, vol 7, [En línea]. [Consultado el: 23 de octubre de 2012] Disponible en: http://bvs.sld.cu/revistas/san/vol7_1_03/san53103.htm
7. GARCIA, N. *Destacan aplicación de software cubano para el control contable*. 2012, [En línea]. [Consultado el: 23 de octubre de 2012], Disponible en: <http://www.juventudrebelde.cu/ciencia-tecnica/2012-08-20/destacan-aplicacion-de-software-cubano-para-el-control-contable/>
8. EPF. *OpenUp Basic. Eclipse Foundation*. 2011, [En línea]. [Consultado el: 24 de octubre de 2012] Disponible en: <http://epf.eclipse.org/wikis/openupsp/>
9. LARMAN, C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Third Edition. 2004, ISBN: 0131489062
10. JACOBSON, I. *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson Educación S.A. 2000, ISBN: 84-7829-036-2
11. FAHNLE, P. *ASP.NET*. 2012, [En línea]. [Consultado el: 24 de octubre de 2012] Disponible en: www.programacion.com/asp/articulo/aspnet_quees/
12. ÁLVAREZ, Á. *JAVA*. 2012, [En línea]. [Consultado el: 24 de octubre de 2012] Disponible en: www.desarrolloWeb.com/articulos/497.php
13. POTENCIER, F. *Symfony 1.2, la guía definitiva*. ISBN-13: 978-1590597866.
14. SECO, J. *El lenguaje de programación C#*, [En línea]. [Consultado el: 24 de octubre de 2012] Disponible en: <http://dis.um.es/~bmoros/privado/bibliografia/LibroCsharp.pdf>

15. SÁNCHEZ, J. *MySQL guía rápida*. 2004, [En línea]. [Consultado el: 24 de octubre de 2012] Disponible en: <http://www.olajedatos.com/documentos/mysql.pdf>
16. CARABALLO, S. *Prolegómenos Sobre el Lenguaje de Modelado Unificado (UML)*. 2003, [En línea]. [Consultado el: 24 de octubre de 2012] Disponible en: <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/prolego.pdf>
17. PATRICIO, L. *Rational Unified Process (RUP)*. Universidad Politécnica de Valencia, [En línea]. [Consultado el: 8 de enero de 2013] Disponible en: <http://es.scribd.com/doc/51852648/Introduccion-a-RUP>
18. HERNANDEZ, R. *Una Introducción a la Gestión de Proyectos*, 2009, ISBN:13-978-970-10-4219-9
19. SÁNCHEZ, K. *Método para evaluar proyectos informáticos y establecer un orden de prioridad que ayude a la toma de decisiones*. Tesis de maestría, UCI. 2010.

Bibliografía

1. ELÍAS, O. *Manual Temas de Contabilidad*, segunda edición. La Habana 2004, [En línea]. [Consultado el: 18 de octubre de 2012] Disponible en: <http://biblioteca.ihatuey.cu/links/otros/mc.pdf>
2. MINSAP. *Vice-Ministerio de Economía. Cuba. Dirección Nacional de Finanzas y Contabilidad*. 2011, [En línea]. [Consultado el: 18 de octubre de 2012] Disponible en: <http://www.dncontabilidad.sld.cu/Temas/activos.html>
3. RAE. *Diccionario de la Real Academia Española*. 2012 [En línea], Disponible en: <http://www.rae.es/rae.html>
4. BEJERANO, G. *Ministerio de Auditoría y Control. Resolución No.306/08.2008*, [En línea]. [Consultado el: 18 de octubre de 2012] Disponible en: <http://correo.servisa.tur.cu/Capacitacion/Consultor/09%20Legislacion/MAC/RES-MAC-2008-306.htm>
5. BEJERANO, G. *Contraloría General de la República de Cuba. Resolución NO. 60/11*. 2011, [En línea]. [Consultado el: 18 de octubre de 2012] Disponible en: <http://www.contraloria.cu/documentos/Res.60-11.pdf>
6. MINISTERIO DE AUDITORÍA Y CONTROL MINISTERIO DE FINANZAS Y PRECIOS. *COMPROBACIÓN NACIONAL AL CONTROL INTERNO*.2008, [En línea] [Consultado el: 18 de octubre de 2012] Disponible en: <http://correo.servisa.tur.cu/Capacitacion/Consultor/03%20Control%20Interno/7%20Control%20Nacional/PROGRAMA%20COMP%20NAC%20CI%202008%20.pdf>
7. BELTRÁN, H. *Propuesta de acciones para solucionar las deficiencias en cuanto al control de los activos fijos tangibles en Cuba en Observatorio de la Economía Latinoamericana*, 2012, N° 169, [En línea]. [Consultado el: 18 de octubre de 2012] Disponible en: <http://www.eumed.net/cursecon/ecolat/cu/2012>
8. RUMBAUGH, I. *El Lenguaje Unificado de Modelo, Manual de Referencias*. 1998, [En línea]. [Consultado el: 1 de noviembre de 2012] Disponible en: <http://bibliodoc.uci.cu/pdf/reg03050.pdf>
9. SAMPIERI, R. *Metodología de la Investigación*. Cuarta edición, 2006, ISBN: 9701057538.
10. *Tutorial de Java. Arquitectura Modelo-Vista-Controlador*. 2012, [En línea]. [Consultado el: 17 de octubre de 2012] Disponible en: http://sunsite.dcc.uchile.cl/java/docs/JavaTut/Apendice/arg_mvc.html
11. BREA, F. *Web Services con PHP*. 2012, [En línea]. [Consultado el: 29 de octubre de 2012] Disponible en: <http://www.desarrolloWeb.com/manuales/61>
12. *PostgreSQL 9.0.1 Documentation*.1996-2010. The PostgreSQL Global Development Group, [En línea]. [Consultado el: 1 de noviembre de 2012] Disponible en: <http://www.network-theory.co.uk/docs/postgresql9/vol2/>
13. MORÁN, M. *Metodología para el Control Interno de los Recursos Físicos Informáticos de la Corporación Copextel S.A*. 2010, [En línea]. [Consultado el: 1 de noviembre de 2012] Disponible en: <http://www.fec.uh.cu/CUGIO/1%20acciones/Proyectos-Protocolos/15/Mireya%20Bencharidl.pdf>

Glosario de Términos y Siglas

Active Record - Es la principal clase para la administración y funcionamiento de modelos.

Don't Repeat Yourself (DRY) - Se trata de una filosofía de desarrollo de software que promueve que las piezas de información no estén duplicadas.

Framework - Abstracción de código común que provee funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo.

IntelliSense - Es la aplicación de autocompletar, mejor conocido por su utilización en Microsoft Visual Studio entorno de desarrollo integrado.

Keep It Simple, Stupid (KISS) - Es la filosofía tradicional de Unix para mantener las cosas simples. Es un concepto que explica muchas de las opciones en el diseño.

LDAP - Protocolo Ligero de Acceso a Directorios (*Lightweight Directory Access Protocol*) referencia a un protocolo a nivel de aplicación el cual permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

WSDL – Lenguaje de Descripción de Servicios Web (*Web Services Description Language*).

Anexo 1: Entrevista a responsables de AFT en la Facultad6.

Estudiante de la carrera de Ingeniería en Ciencias Informáticas de la UCI, realizo un proyecto de investigación del que usted forma parte del proceso objeto de estudio, considero que tiene informaciones importantes para el desarrollo de este trabajo, por lo que solicito su colaboración y se le informa el carácter confidencial de sus respuestas.

Gracias de antemano.

Para que la Facultad 6 logre cumplir con los objetivos estratégicos trazados, requiere de un correcto control de los AFT donde el proceso de auditoría y control de los mismos juega un papel protagónico para este fin.

Objetivo: Diagnosticar el estado actual de la gestión en el proceso de auditoría y control de los AFT para la Facultad 6 de la UCI.

1. ¿A través de qué soporte se realiza la adquisición de los datos obtenidos en el proceso de auditoría y control de los AFT?

Sistema Informático.

Modelos impresos.

Modelos digitales.

Otros.

Especificar para otros _____.

2. Existe algún mecanismo que permita procesar la información recogida.

Existe un sistema informático que filtra la información obtenida y la almacena en una base de datos.

Existe una persona encargada de filtrar los datos obtenidos y crear hojas de cálculos que se guardan en: la PC, correo electrónico, memorias flash, CD u otro.

Existe una persona encargada de filtrar los datos obtenidos y archivarlos en hojas y carpetas.

Otros.

Especificar para otros _____.

3. A través de qué recurso se reporta la información previamente procesada sobre los AFT.

Sistema Informático.

Reportes digitales.

Reportes impresos.

Otros.

Especificar para otros _____.

4. De existir estados no deseados en la información recopilada existe algún mecanismo que detecte y alerte a los directivos sobre una posible afectación.

Existe un sistema informático que procesa los datos recogidos y genera un informe con el posible problema que puede estar ocurriendo.

Existe una persona que durante el proceso de filtrar los datos, detecta que el problema y genera un informe manualmente.

Otros.

Especificar para otros _____

5. Considera, usted, que el % de control establecido actualmente para el proceso de auditoría y control de los AFT es suficiente para obtener buenos resultados.

Suficiente: SI NO

Justifique: _____

6. Según su experiencia. ¿Considera, usted, que existe fiabilidad en la forma que es recogida y almacenada la información?

Es fiable el proceso: SI NO

Justifique: _____

Anexo 2: Guía de Observaciones

Soy estudiante de la carrera de Ingeniería en Ciencias Informáticas de la UCI, realizo un proyecto de investigación del que usted forma parte del proceso objeto de estudio. Considero que tiene información importante para el desarrollo de este trabajo, por lo que solicito su colaboración para observar los principales elementos que intervienen en el proceso de auditoría y control de los AFT de la Facultad 6, correspondiente al mes en curso. Le garantizo el carácter confidencial de la observación.

Gracias de antemano.

Objetivo: Diagnosticar el estado actual de la gestión en el proceso de auditoría y control de los AFT para la Facultad 6 de la UCI, según la observación realizada por el investigador durante el proceso.

1. ¿A través de qué soporte se evidencia el proceso de adquisición de los datos obtenidos por el departamento de economía?

Sistema Informático.

Modelos impresos.

Modelos digitales.

Otros.

Especificar para otros _____.

2. ¿Cada qué periodo de tiempo puede realizar las auditorías sobre los AFT al 10%?

Trimestral

Mensual.

Semanal.

Otros.

Especificar para otros _____.

3. Existe algún mecanismo para notificar a la persona encargada del control de AFT si ha ocurrido algún cambio en un área determinada. ¿Cuál?

Comunicación personal.

Modelos digitales.

Correo electrónico.

Modelos impresos.

Otros.

Especificar para otros _____.

4. ¿Qué mecanismos se utiliza para identificar los AFT dentro de los locales?

Rotulado manual.

Información impresa.

Otros.

Especificar para otros _____.

5. ¿Existe algún mecanismo que permita procesar el comportamiento histórico de las auditorías realizadas a los diferentes locales?

___ Mediante documentos impresos.

___ Mediante tablas digitales en Excel.

___ Preguntar a los responsables de las áreas.

___ Otros.

Especificar para otros _____

Anexo 3: Acta de Responsabilidad Material Individual.



FECHA ____ / ____ / ____

Acta de Responsabilidad Material Individual.

Por este medio se notifica que los activos fijos que a continuación se relacionan, se encuentran ubicados en el local de _____

Bajo la responsabilidad del trabajador _____

EL cual ocupa el cargo de _____

Nro.	Descripción del medio.	Nro Inventario.	Estado Técnico	Observaciones

Firma del Responsable

Anexo 4: Acta de Responsabilidad Material Colectiva.



FECHA ____ / ____ / ____

Acta de Responsabilidad Material Colectiva.

Por este medio se notifica que los activos fijos que a continuación se relacionan, se encuentran ubicados en el local de _____

Bajo la responsabilidad de los trabajadores que a continuación se relacionan:

Nº	Nombre y Apellidos	Firma

Anexo 5: Modelo de Movimiento de Medios.

Entidad:		Código		Movimiento de Medios Básicos						
Dirección:										
Área:										
Descripción:										
CNMB		D	M	A	Alquiler					
Inventario No:					Tiempo	D	M	A		
Sub-Cuenta		Valor								
Entidad:										
Dirección:										
Área:				Firma:						
<input type="checkbox"/> Compra M. B. Nuevo <input type="checkbox"/> Retiro <input type="checkbox"/> Traslado interno <input type="checkbox"/> Compra M. B. de uso <input type="checkbox"/> Perdida <input type="checkbox"/> Activo ocioso <input type="checkbox"/> Traspaso recibido <input type="checkbox"/> Traspaso efectuado <input type="checkbox"/> Ajuste inv alta <input type="checkbox"/> Ajuste inv. baja <input type="checkbox"/> Otros activos <input type="checkbox"/> Venta <input type="checkbox"/> Enviado a reparar				Fundamentación de la Operación						
Informe técnico										
Nombre		Cargo				Firma				
Hecho:		D	M	A	Aprobado:			D	M	A
Autorizado:		D	M	A	Cargo:			D	M	A
Cargo:					Transportador o Receptor					
Firma _____					Firma _____					
Anotando		No _____			D	M	A	No		

Anexo 6: Especificación y Descripción ampliada de los casos de uso del sistema.

CU Autenticar Usuario

Caso de Uso	Autenticar Usuario	
Actores	Usuario, Auditor y Administrador	
Resumen	<p>Se inicia cuando el usuario ingresa sus credenciales para acceder al sistema. Los datos serán verificados a través del servicio Web de autenticación de la UCI; de ser correctos, se habilita el acceso al sistema en dependencia de los permisos que tenga habilitado su rol.</p> <p>El caso de uso termina cuando el usuario accede al sistema.</p>	
Precondiciones		
Referencias	RF 1	
Prioridad	Alto	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El caso de uso inicia cuando los usuarios acceden al sistema.</p> <p>3. El usuario introduce sus credenciales.</p>	<p>2. El sistema brinda la posibilidad de introducir usuario y contraseña.</p> <p>4. El sistema verifica que los campos del formulario no estén vacíos.</p> <p>5. Se verifica que el usuario exista en la BD y que la contraseña especificada sea correcta.</p> <p>6. Si los datos son correctos, la aplicación muestra la interfaz principal, donde aparecen las funcionalidades según el rol que tenga.</p>	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	<p>4.1. De existir un campo vacío el sistema muestra un mensaje de error y regresa a la actividad 1 del flujo normal de los eventos.</p> <p>5.1. Si los datos no son correctos, la aplicación muestra un mensaje de error, y le permite al usuario volver a introducir las credenciales.</p>	
Poscondiciones	El usuario queda autenticado.	

CU Cargar Documento Excel

Caso de Uso	Cargar Documento Excel	
Actores	Administrador	
Resumen	El caso de uso inicia cuando el administrador selecciona en el menú la opción cargar Excel, el sistema muestra la interfaz para seleccionar el documento en cuestión. Los datos son verificados para ver si existen la base de datos; de ser así el sistema los actualiza y el caso de uso termina, en el caso de no existir dichos datos el sistema los muestra y da la posibilidad de añadirlos a un centro de costo.	
Precondiciones	El administrador tiene que estar autenticado.	
Referencias	RF 7	
Prioridad	Alto	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario desde la interfaz principal, selecciona la opción "Cargar Excel".	2. El sistema muestra la interfaz cargar Excel con un botón para seleccionar donde se encuentra el documento.	
3. El usuario selecciona el documento a cargar.	4. El sistema verifica si los datos de las áreas están o no en la base de datos.	
	5. Si todos los datos existen en la base de datos, el sistema los actualiza con la información del documento recién cargado y muestra el mensaje "No hay áreas sobrantes".	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	4.1 Si todos los datos no se encuentran en la base de datos. El sistema muestra las áreas sobrantes, dando la posibilidad de añadirlos a un centro de costo. El caso de uso termina.	
Poscondiciones	Los datos del documento Excel son agregados o actualizados en la base de datos.	

CU Gestionar Usuario

Caso de Uso	Gestionar Usuario	
Actores	Administrador	
Resumen	El caso de uso inicia cuando el administrador selecciona en el menú la opción Gestionar Usuario. El sistema muestra las opciones de agregar, buscar, modificar y eliminar usuario. El administrador realiza las operaciones que desee y de esta forma se gestionan los usuarios. El caso de uso termina.	
Precondiciones	El administrador tiene que estar autenticado.	
Referencias	RF 2	
Prioridad	Alto	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador selecciona la opción gestionar usuario	2. El sistema muestra el listado de usuarios existentes brinda la posibilidad de Agregar, Buscar, Modificar o Eliminar usuario.	
Sección "Agregar Usuario"		
3. El administrador Selecciona la opción agregar usuario.	4. El sistema muestra los datos correspondientes: <ul style="list-style-type: none"> ➤ nombre de usuario ➤ rol ➤ centro de costo al que pertenece. <p>Y permite guardar usuario se guardan los datos del nuevo usuario</p> <p>Cancelar ver Alternativa 1: "Cancelar Operación "</p>	
5. El administrador introduce los datos y selecciona la opción guardar.	6. Valida los datos, si hay campos incompletos ver Alternativa 2: "Hay campos vacíos", si hay campos incorrectos ver Alternativa 3: "Hay campos incorrectos".	
	7. El caso de uso termina.	
Flujos Alternos "Cancelar Operación"		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción de Cancelar operación.	2. Regresa a la vista anterior. 3. El caso de uso termina.	
Flujos Alternos "Hay campos vacíos"		
Acción del Actor	Respuesta del Sistema	
	1. Muestra un indicador al lado de los campos vacíos.	
Flujos Alternos "Hay campos incorrectos"		
Acción del Actor	Respuesta del Sistema	

	<ol style="list-style-type: none"> 1. Muestra el mensaje de error "Existen campos incorrectos" 2. Muestra un indicador sobre los campos incorrectos.
Sección "Buscar Usuario"	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El Administrador selecciona la opción Buscar Usuario. 3. El administrador introduce los criterios de búsquedas. 	<ol style="list-style-type: none"> 2. El sistema muestra los datos correspondientes: <ul style="list-style-type: none"> ➤ nombre de usuario ➤ rol ➤ centro de costo al que pertenece. 4. El sistema muestra el resultado de acuerdo con los criterios de búsquedas. 5. El caso de uso termina.
Sección "Modificar Usuario"	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El Administrador selecciona el campo que desea modificar haciendo doble clic sobre él. 	<ol style="list-style-type: none"> 2. Muestra un cuadro de texto para introducir el nuevo valor que tendrá el usuario en ese campo. 3. El sistema envía el nuevo valor para ser procesado. 4. El caso de uso termina.
Sección "Eliminar Usuario"	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El Administrador selecciona un usuario de la lista y luego la opción Eliminar Usuario. 6. En el mensaje Presiona el botón Sí. 	<ol style="list-style-type: none"> 2. Muestra un mensaje de confirmación de eliminación. 7. Envía la información del usuario y se elimina. 8. El caso de uso termina.
Flujos Alternos: "Cancelar operación"	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. En el mensaje Presiona el botón No. 	<ol style="list-style-type: none"> 2. Regresa a la vista anterior. 3. El caso de uso termina.
Poscondiciones	Se gestionaron los datos de los usuarios.

CU Solicitar Realizar Auditoría

Caso de Uso	Solicitar Realizar Auditoría
Actores	Auditor y Administrador
Resumen	Comienza cuando los actores, seleccionan en la interfaz principal la opción Seleccionar Realizar Auditoría, el sistema muestra las opciones correspondientes, los actores seleccionan los datos correspondientes, el sistema muestra los AFT según los datos introducidos. El caso de uso termina.
Precondiciones	El actor debe estar autenticado.
Referencias	RF 11
Prioridad	Alto
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Solicitar realizar auditoría	2. El sistema muestra la interfaz en la cual se seleccionan los siguientes datos: <ul style="list-style-type: none"> ➤ Centro de costo. ➤ Área de Responsabilidad.
3. El actor introduce los datos y presiona el botón solicitar datos.	4. El sistema muestra los datos de los AFT correspondientes a los datos introducidos.
	5. El caso de uso termina.
Poscondiciones	Queda solicitada la auditoría con la información necesaria.

CU Realizar Auditoría

Caso de Uso	Realizar Auditoría
Actores	Auditor y Administrador
Resumen	El caso de uso inicia cuando el actor selecciona en la interfaz principal la opción Realizar Auditoría. El sistema toma por defecto que todos los AFT son faltantes y lo muestra, así como los datos solicitados y el campo para introducir el MB de los AFT, verifica si los MB corresponden a las áreas auditadas. De ser el caso lo borra de los faltantes, de no ser así se le informa al actor que se ha encontrado un AFT sobrante del área y se le pregunta que desea hacer con él.
Precondiciones	El actor debe estar autenticado.
Referencias	RF 12
Prioridad	Alto
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

<p>1. El actor selecciona la opción Realizar Auditoría.</p> <p>3. El actor introduce el código de cada AFT.</p>	<p>2. El sistema muestra la interfaz con la información de los AFT seleccionados y brinda la posibilidad de introducir el código de cada uno de ellos para comprobar si existen.</p> <p>4. El sistema realiza la comprobación, si existe en el área, lo elimina de los sobrantes.</p> <p>5. Finaliza el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<p>4.2. En el mensaje el actor Presiona el botón Sí.</p> <p>4.2.1. En el mensaje el actor Presiona el botón No.</p>	<p>4.1. Si no existe en el área, lo busca en las otras áreas, de encontrarlo le muestra un mensaje al usuario que ese AFT no es del área, que si desea declararlo como sobrante.</p> <p>4.3. El sistema lo agrega a la lista de sobrantes del área.</p> <p>4.4 EL caso de uso termina.</p> <p>4.5 Regresa a la vista anterior</p> <p>4.3.El caso de uso termina</p> <p>4.4 Regresa al punto 3.</p>
Poscondiciones	Queda realizada la auditoría

Anexo 7: Relación de expertos seleccionados para la aplicación del método Delphi.

No	Nombre y Apellidos	Centro y Competencias
1	Msc. Maidel Beatriz Ginarte Duran	UCI- Especialista en calidad y sistema de gestión
2	Dr. Sergio L. Reyes Lorente	UCI – Especialista en Calidad de Software.
3	Msc. Yenisleidy Piloto Loste	UCI- Especialista en calidad y sistema de gestión
4	Msc. Oneyvis Chávez Robayna	UCI- Especialista en calidad y sistema de gestión
5	Msc. Zobeida R. Pérez López-Chávez	UCI- Especialista en calidad y sistema de gestión
6	Msc. Iraida Paredes Sánchez	ISPETP – Gestión de la Información.
7	Lic. Omar Mar Cornelio	UCI – Especialista en Auditoría de Software.
8	Ing. Victor F. Mestre Ruano	UCI – Especialista en Gestión de Proyecto.
9	Jenisley Verde Acosta	UCI – Especialista en Gestión de Proyecto.

Anexo 8: Guía de evaluación a los expertos. Determinación del peso de los criterios de prioridad.

Se entenderá por peso a la importancia que se le atribuye a la evaluación de ese criterio para el proyecto con respecto al resto de los criterios. Para ello se le pide a cada experto que emita su opinión acerca de la importancia (en base a 100) que tiene cada criterio con relación a los demás para el proyecto a evaluar. La sumatoria de estos pesos debe ser igual a 100.

<i>Determinación del peso de los criterios de prioridad.</i>		
No	Criterio	Peso
1	Contribución científica de la solución.	
2	Correspondencia con el negocio de estudio	
3	Ahorro de recursos materiales	
4	Legalidad en el proceso	
5	Rendimiento	
6	Integridad	
7	Fiabilidad	
8	Disponibilidad	
9	Impacto en el área para la cual está destinada la propuesta de solución	
10	Gestión del proceso	
11	Tiempo de ejecución	
12	Facilidad del usuario para realizar las operaciones.	
13	Fácil administración	
14	Trazabilidad de los datos con respecto a estados anteriores	
15	Adquisición de los datos	
		100

Anexo 9: Tabla de Distribución Chi-cuadrado (X^2). P ($X \geq x$). Donde “P” es la probabilidad de encontrar un valor mayor o igual que el Chi cuadrado tabulado, y “v” los grados de libertad.

V	0,005	0,01	0,025	0,05	0,95	0,975	0,99	0,995
1	0,00003935	0,000157	0,000982	0,00393	3,841	5,024	6,635	7,879
2	0,010	0,020	0,051	0,103	5,991	7,378	9,210	10,597
3	0,072	0,115	0,216	0,352	7,815	9,348	11,345	12,838
4	0,207	0,297	0,484	0,711	9,488	11,143	13,277	14,860
5	0,412	0,554	0,831	1,145	11,070	12,832	15,086	16,750
6	0,676	0,872	1,237	1,635	12,592	14,449	16,812	18,548
7	0,989	1,239	1,690	2,167	14,067	16,013	18,475	20,278
8	1,344	1,647	2,180	2,733	15,507	17,535	20,090	21,955
9	1,735	2,088	2,700	3,325	16,919	19,023	21,666	23,589
10	2,156	2,558	3,247	3,940	18,307	20,483	23,209	25,188
11	2,603	3,053	3,816	4,575	19,675	21,920	24,725	26,757
12	3,074	3,571	4,404	5,226	21,026	23,337	26,217	28,300
13	3,565	4,107	5,009	5,892	22,362	24,736	27,688	29,819
14	4,075	4,660	5,629	6,571	23,685	26,119	29,141	31,319
15	4,601	5,229	6,262	7,261	24,996	27,488	30,578	32,801
16	5,142	5,812	6,908	7,962	26,296	28,845	32,000	34,267
17	5,697	6,408	7,564	8,672	27,587	30,191	33,409	35,718
18	6,265	7,015	8,231	9,390	28,869	31,526	34,805	37,156
19	6,844	7,633	8,907	10,117	30,144	32,852	36,191	38,582
20	7,434	8,260	9,591	10,851	31,410	34,170	37,566	39,997
21	8,034	8,897	10,283	11,591	32,671	35,479	38,932	41,401
22	8,643	9,542	10,982	12,338	33,924	36,781	40,289	42,796
23	9,260	10,196	11,689	13,091	35,172	38,076	41,638	44,181
24	9,886	10,856	12,401	13,848	36,415	39,364	42,980	45,558
25	10,520	11,524	13,120	14,611	37,652	40,646	44,314	46,928
26	11,160	12,198	13,844	15,379	38,885	41,923	45,642	48,290
27	11,808	12,878	14,573	16,151	40,113	43,195	46,963	49,645
28	12,461	13,565	15,308	16,928	41,337	44,461	48,278	50,994
29	13,121	14,256	16,047	17,708	42,557	45,722	49,588	52,335
30	13,787	14,953	16,791	18,493	43,773	46,979	50,892	53,672

Anexo 10: Guía de evaluación a los expertos. Calificación de los criterios de evaluación.

El paso de calificación de cada criterio consiste en estudiar y evaluar el comportamiento de cada uno de los criterios; es por ello que luego de determinado el peso de cada criterio, se pasa a un segundo momento de interacción con los expertos, en el que se les pide que califiquen cada criterio en una escala de 1-10 según su opinión y conocimiento acerca del comportamiento que tiene del caso de estudio.

Indicador	No	Criterio	Peso
Criterios de mérito científico	1	Contribución científica de la solución.	
Criterios de creación	2	Correspondencia con el negocio de estudio	
	3	Ahorro de recursos materiales	
	4	Legalidad en el proceso	
Criterios de seguridad	5	Rendimiento	
	6	Integridad	
	7	Fiabilidad	
	8	Disponibilidad	
Criterios de impacto	9	Impacto en el área para la cual está destinada la propuesta de solución.	
	10	Gestión del proceso	
	11	Tiempo de ejecución	
Criterios de usabilidad	12	Facilidad del usuario para realizar las operaciones.	
	13	Fácil administración	
	14	Trazabilidad de los datos con respecto a estados anteriores.	
	15	Adquisición de los datos	