

Universidad de las Ciencias Informáticas
Facultad 15



**“Informatización del proceso modificación de Activo Fijo
Tangible del sistema integral de gestión de entidades,
Cedrux”**

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

Autores: Alberto Pérez González.
Lisbet Milagro González Batista.

Tutor: Henry Ernesto Bermúdez Pérez

La Habana, 2010
“Año 52 de la Revolución”

Declaración de autoría.

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Lisbet Milagro González Batista

Alberto Pérez González

Ing. Henry Ernesto Bermúdez Pérez

Agradecimientos

Agradecemos a todas las personas que han colaborado para con la realización de este trabajo.

A mis abuelos Elsa Polanco y Fermín González son a los que más les agradezco, por privarse en muchas ocasiones de muchas cosas para tuviera lo necesario para seguir en la escuela. Mis tíos Diana González y Luis González, por haber sido incondicionales, en todos los sentidos, realmente les agradezco por haber estado ahí siempre, cada vez que he necesitado de su apoyo. A mi abuela María del Carmen Calzadilla, por todo su cariño y por ayudarme en lo que ha podido aunque sé que ella hubiese querido hacer más por mí, por apoyarme y confiar en mí. Mi tía Idalia Calzadilla, que se desvive por mí y me cuida y ceba como un preciado tesoro. A mi tía Mileidis y Dania, a mi prima Sulema, por todo su esfuerzo para conmigo. Héctor Barban (mi cosita) que ha sido mi amor, mi amigo y mi compañero, que ha sabido comprenderme y entenderme con todos mis defectos y malacrianzas, por estar a mi lado siempre, por cuidarme cuando mi familia ha estado lejos y no lo ha podido hacer, por todos esos momentos a mi lado tanto buenos o malos. A toda mi familia en general, es que son muchos. Mis queridos suegros teresita y Héctor que mejor no los quiero que me han apoyado muchísimo, y me han dado mucho cariño. A mis compañeros: María, Mairelys, Rolando y Yaimara que han sido como unos hermanos para mí. En fin, a todos aquellos que de una forma u otra me ayudaron a que me hiciera Ingeniera Informática en la UCI.

Lisbet Milagro González Batista.

A mi madre Celia María González Novo, que ha sabido ayudarme en todos los momentos, brindándome siempre su apoyo incondicional y su aliento en todo momento, para seguir mi carrera y poderme formar como profesional. A mi padre Alberto Luis Pérez Zabadi por inculcar en mí los valores necesarios para lograr que mis sueños se conviertan en realidad. A mi padrino Justo Pelladito por alentarme, y darme fe y esperanza en los momentos que más lo he necesitado. A mi amigo Raudel que ha sido como un hermano para mí, A Rubén Núñez que ha sido siempre mi ejemplo a seguir como profesional por su laboriosidad y perseverancia. Al profesor Pascual que siempre ha tenido un consejo para mí cuando más lo he necesitado. A todos aquellos que de manera directa o aún, sin saberlo, han hecho su aporte a mi formación como ser humano y como profesional.

Alberto Pérez González.

Dedicatoria

A la Revolución y al Comandante en Jefe Fidel Castro, artífices de este genial proyecto que es la Universidad de las Ciencias Informáticas; institución que nos brinda la posibilidad de optar por el título que persigue el presente trabajo.

Primeramente decir que no existe nada en el mundo, que puede llenar el amor, la comprensión y dedicación de una madre, por eso dedico mi tesis a la mujer que luchó incansablemente por darme la vida y que después siguió luchando por que fuera una buena persona y que casi con su sangre logró que llegara a esta universidad, aunque ella no podrá ver que su hija alcanzó sus sueños y logró hacerse una ingeniera gracias a su gran empeño y amor. A tu memoria mamita.

A mis abuelos que han sido como unos padres para mí. Por estar ahí siempre, apoyándome cuando más falta me ha hecho, en todos los momentos tristes y en los momentos felices también; por sus sabios consejos constructivos, así como todo el amor que han sabido darme en todos estos años.

Lisbet Milagro González Batista

Dedico este Trabajo de Diploma a mi madre, que durante toda mi vida ha sido para mí no tan solo una madre sino también un padre, una amiga y un ejemplo a seguir. Aunque no siempre he estado a su altura, trato en todos los momentos de mi vida de serle fiel y de lograr ser cada día mejor para que nunca se sienta defraudada. Si a alguien le debo el estar aquí, es a ti mami.

A mi padre, que durante todo el tiempo que estuvo a mi lado me enseñó la importancia de ser un buen profesional, me inculcó los valores que hoy tengo como ser humano y me apoyó en todo cuanto pudo sin importar cuán difícil fuera para él. Hoy, doy por seguro que estarías orgulloso de vivir este día junto a mí.

Alberto Pérez González

Resumen

El país avanza con amplios pasos dentro del proceso de utilización de las tecnologías de la información y las comunicaciones, basándose en los principios de independencia tecnológica y principalmente en las particularidades de la economía cubana; factor decisivo para el desarrollo de las empresas, economía y de la sociedad cubana.

Debido a lo anteriormente expuesto, Cuba se encuentra inmersa en la creación del producto Cedrux. Esta solución tiene incluido un subsistema para el control de los *activos fijos tangibles* que pretende servir de apoyo a la toma de decisiones en las entidades. Para lograrlo, la prioridad se centra en la necesidad de proteger los medios y productos, manejados durante la realización de un proceso determinado de tal forma que ayude y agilice el trabajo.

Este trabajo abarca el análisis, diseño, implementación y validación del componente modificaciones de los *activos fijos tangibles*, que permite mejorar la eficiencia, confiabilidad y auditoría de estos medios en la gestión de estos medios en cualquier entidad.

Palabra clave: *activo fijo tangible.*

Índice de contenido:

Introducción.....	9
Capítulo 1: Fundamentación teórica.	12
1.1 Introducción.....	12
1.2 Conceptos básicos asociados al dominio del problema a resolver.	12
1.3 Modelo de desarrollo.....	13
1.4 Modelo Vista Controlador (Model-View-Controller).....	14
1.5 Sistemas vinculados al proceso de modificación de los AFT.	15
1.5.1 Sistemas nacionales.....	15
1.5.2 Sistemas internacionales.....	18
1.5.3 Limitaciones de los sistemas existentes.	20
1.6 Notación para el modelado de procesos de negocio (BPMN)	20
1.7 Lenguajes, herramientas, marcos de trabajo y tecnologías utilizadas.	21
1.7.1 Lenguajes.....	21
1.7.2 Herramientas.....	23
1.7.3 Marcos de trabajo (Frameworks).....	26
1.7.4 Tecnologías.....	28
1.8 Navegador web	30
Conclusiones parciales.....	31
Capítulo 2: Análisis y diseño del módulo de modificaciones de AFT.	32
2.1 Introducción.....	32
2.2 Reglas del negocio.....	32
2.3 Mapa de procesos y descripción del negocio.	32
2.3.1 Descripción de procesos de negocio.	33
2.3.2 Mapa de procesos.	36

2.4	Descripción de los requisitos.	37
2.5	Diccionario de datos.....	41
2.6	Prototipo de interfaz de usuario.	45
2.7	Arquitectura base.	47
2.8	Patrones de diseño utilizados.....	48
2.8.1	Patrones de asignación de responsabilidades.....	48
2.8.2	Patrón Gang of Four (GoF) Cuadrilla de Cuatro.	49
2.8.3	Patrón Modelo Vista Controlador.	50
2.8.4	Patrón de integración entre componentes (Inversor de control).	51
2.9	Diagrama de clases.	51
2.10	Modelo de datos.....	56
	Conclusiones parciales.....	56
Capítulo 3: Construcción y validación de la solución propuesta.		57
3.1	Introducción.....	57
3.2	Estrategias de integración.	57
3.2.1	Servicios consumidos.....	58
3.2.2	Servicios Brindados.	59
3.3	Estándares de código.	59
3.4	Estructura de datos apropiada para la implementación de los algoritmos.	60
3.5	Clases del componente: Modificaciones.	60
3.6	Diagrama de componentes.....	61
3.7	Modelo de despliegue.	63
3.8	Pruebas de software.	63
3.8.1	Prueba de caja blanca.	63
3.8.2	Prueba de caja negra.....	67

Conclusiones parciales.....	71
Conclusiones.....	72
Recomendaciones.....	73
Bibliografía.....	74
Glosario de Términos.....	75

Introducción.

En los inicios de un nuevo milenio marcado por un entorno agresivo, globalizado y dinámicamente cambiante, y cuando impera la era del conocimiento y la información, las empresas cubanas buscan y experimentan soluciones que puedan brindar alcance, dentro del imprescindible desarrollo sostenible al que aspiran los pueblos del tercer mundo.

El país ejecuta actualmente una serie de cambios, todos dirigidos a reestructurar el proceso de gestión de entidades¹ basadas en el uso de las nuevas tecnologías de la información y las comunicaciones. Cuba se encuentra inmersa en la creación del producto Cedrux basándose en los principios de independencia tecnológica y principalmente en las particularidades de la economía cubana. Esta solución tiene incluido un subsistema² para el control de los **Activos Fijos Tangibles (AFT)**, que pretende servir de apoyo a la toma de decisiones en las entidades. Para lograr esto, la prioridad se centra en la necesidad de proteger los medios y productos, manejados durante la realización de un proceso³ determinado de tal forma que ayude y agilice el trabajo.

El sistema de planificación de recursos de la empresa (ERP), es un sistema económico integrado, el cual está dividido en módulos⁴, para su mejor entendimiento, uno de los módulos es el de Control de AFT. El módulo de AFT de la Línea Logística incluye temas como la multimonedas⁵, la agrupación de activos, la generación de documentos y la vista de AFT. Este último garantiza la búsqueda de activos y las modificaciones de propiedad de dichos activos, actualmente no satisfechas de manera conjunta por los sistemas existentes.

Existen dos tipos de modificaciones de AFT las modificaciones de propiedad de AFT y las modificaciones de valor de los AFT. Para realizar modificaciones de valor en las entidades no se utiliza ningún documento primario, por lo que atendiendo a la solución de este sistema, basado en documentos para registrar las operaciones, se incluirá la realización de un documento para ejecutar estas operaciones. Las modificaciones de valor que se le pueden realizar a un AFT se pueden dividir en: Adiciones, adaptaciones y reconstrucciones, revalorizaciones o avalúos. Para realizar modificaciones de propiedad solo se cambia los datos de los campos que corresponden a las propiedades específicas de cada AFT (son los atributos dinámicos de los grupos que heredan los AFT que corresponden a él).

Estas actividades en las empresas o entidades no automatizadas se llevan a cabo en papeles de forma manual, lo que provoca que el proceso de modificación de un AFT se torne lento y complejo. En muchos casos se deterioran estos papeles o son dañados por desastres naturales provocando así la pérdida de la información. Cuando se desea modificar un AFT, la búsqueda de este se hace un

proceso muy engorroso, por causa de los grandes volúmenes de información existentes en las empresas.

Por otro lado, están las entidades que usan sistemas nacionales que se caracterizan por abordar solamente parte del problema de la gestión de la empresa o la unidad presupuestada⁶ y no soportan mecanismos estándares de integración con otras aplicaciones. Además, estos sistemas no realizan el proceso de modificación de los AFT de una forma estandarizada, dificultando así el trabajo de los auditores y entidades relacionadas con la revisión de estos.

Problema científico.

La inexistencia de una herramienta que gestione la información, referente al proceso de modificación de los AFT para el entorno empresarial cubano.

Objetivo general.

Desarrollar una herramienta que gestione la información, referente al proceso de modificación de los AFT para el entorno empresarial cubano, en correspondencia con las nuevas concepciones de informatización.

Objetivos específicos.

1. Analizar los procesos de modificación en los AFT así como las herramientas que se utilizarán para el desarrollo de la solución.
2. Realizar análisis y diseño del módulo de modificación del subsistema AFT según la metodología de desarrollo⁷ escogida.
3. Implementar el módulo de modificación del subsistema AFT según la metodología de desarrollo escogida.
4. Validar el resultado obtenido.

Objeto de estudio.

Procesos de gestión de AFT.

Campo de acción.

Procesos de modificación de AFT.

Idea a defender.

Con el desarrollo del módulo modificaciones, perteneciente al subsistema AFT, se contará con una herramienta que informatice dicho proceso, para el entorno empresarial cubano.

Tareas de la investigación.

- Realizar un estudio del estado del arte a nivel nacional e internacional referente al proceso de modificación de los AFT.
- Realizar un análisis de los requisitos⁸ definidos para el proceso de modificación de los AFT del producto Cedrux.
- Describir las herramientas y lenguajes utilizados.
- Realizar entrevistas a los especialistas funcionales para comprender el negocio.
- Realizar el análisis y diseño del módulo de modificación del subsistema AFT según los requerimientos definidos por los analistas.
- Implementar el componente de modificación de los AFT.
- Identificar e implementar las necesidades de integración con otros componentes¹⁰.
- Realizar prueba de caja blanca al código del componente.
- Realizar prueba de caja negra a la aplicación.
- Escribir en formato digital y copia dura todo el proceso investigativo del desarrollo del trabajo como resultado de la experiencia, recogido en las especificaciones de la guía para la presentación del trabajo de diploma.

Capítulo 1: Fundamentación teórica.

1.1 Introducción.

En este capítulo se abordan temas y conceptos considerados de importancia para el desarrollo y la comprensión del proceso de modificación de los AFT. Se hace referencia a tecnologías y herramientas para el desarrollo web, definidas por el grupo de desarrollo del producto Cedrux. También se realiza un análisis de las herramientas existentes para este proceso, tanto nacionales como internacionales, enfocado en el marco de la economía de Cuba.

1.2 Conceptos básicos asociados al dominio del problema a resolver.

AFT.

Son bienes y derechos adquiridos, recibidos en donación o elaborados por la empresa, que tienen una larga vida útil y su adquisición se realiza con el propósito de utilizarlos en la actividad que desarrolla la entidad y no para ser vendidos.

Los AFT conservan su vida útil por un tiempo superior a un año, no se agotan en el primer uso, transfieren gradualmente su valor a la producción o servicio prestado, están sujetos a desgaste por el uso o condiciones climáticas y tienen forma corpórea, es decir tienen presencia física.

Dichos medios, de acuerdo con sus características, pueden considerarse individualmente o como un conjunto productivo o de servicio integral, compuesto por una suma de medios para el desarrollo de actividades administrativas o de servicio.

Algunos de los AFT son: terrenos, edificios, mobiliarios, maquinarias, minas de carbón.

Modificaciones de propiedad.

Son las modificaciones que se le realiza a las propiedades específicas de cada AFT que son los atributos dinámicos de los grupos que heredan los AFT que corresponden a él.

Modificaciones de valor.

A un AFT se le realizan además de los movimientos, modificaciones a su valor. Para realizar esta operación en las entidades, no se utiliza ningún documento primario, por lo que atendiendo a la solución de este sistema basado en documentos para registrar las operaciones, se incluirá la realización de un documento para ejecutar estas operaciones. Las modificaciones de valor que se le pueden realizar a un AFT se pueden dividir en adiciones, adaptaciones o avalúo.

Avalúos.

El avalúo es el ajuste de los saldos de las cuentas de AFT que resulten de los certificados periciales de valores emitidos por las entidades autorizadas a emitirlos. La correcta realización de los avalúos trae el beneficio de una estimación precisa, detallada e informativa del valor de los bienes muebles e inmuebles.

Los avalúos pueden implicar un aumento o disminución en los valores actuales de los AFT registrados en empresas estatales. Cuando la reevaluación implica una disminución del valor del AFT, se registra como *gastos por pérdidas*, afectando el resultado del período.

Adaptaciones.

En los casos en que al ejecutarse la adaptación o mejora, sea necesaria la sustitución de partes, estas se dan de baja, bien por su valor en libros si existiese o por una estimación efectuada por peritos.

Ejemplo: Adicionarle 2 memorias RAM (512 Mb -20 CUC c/u) y quitarle la memoria que tiene actualmente (128 Mb- 10 CUC) a una PC (Core 2 duo 500 CUC). El valor actual del AFT es de 530 CUC.

Adiciones.

Es el desembolso en que se incurre para agregarle algo nuevo a un AFT y que aumenta su valor en la misma cantidad del desembolso realizado para la adición.

Ejemplo: Adicionarle una memoria RAM (512 Mb 10 CUC) a una PC (Core 2 duo 500 CUC). El nuevo valor del AFT es de 510 CUC.

1.3 Modelo de desarrollo.

El modelo de desarrollo del software utilizado fue definido por la línea de arquitectura del proyecto ERP-Cuba para el producto Cedrux. Este describe la secuencia de actividades para la construcción y desarrollo de soluciones. Se logra con la combinación entre los modelos basados en componentes y el iterativo incremental. Se emplearán las técnicas de prototipado¹⁵, si son requeridas, para los requerimientos del usuario de los que no existe una visión clara por parte de estos, con el objetivo de desarrollar una definición mejorada de los requisitos del usuario para el sistema.

Centrado en la arquitectura:

La arquitectura determina la línea base¹⁶, los elementos de software estructurales a partir de los elementos de la arquitectura de negocio. Interviene en la gestión de cambios y diseña la evolución e

integración del producto. La arquitectura orienta las prioridades del desarrollo y resuelve las necesidades tecnológicas y de soporte para el desarrollo.

Orientado a componentes:

Las iteraciones son orientadas por el nivel de significancia arquitectónica de los componentes. Los mismos son abstracciones¹⁷ arquitectónicas de los procesos de negocio y requisitos asociados que modelan el componente que es la unidad de medición y ordenamiento de las iteraciones.

Iterativo e incremental:

Las iteraciones son planificadas y coordinadas con el equipo de arquitectura, los clientes y la alta gerencia. Cada iteración constituye el desarrollo de componentes, los cuales son integrados al término de la integración, permitiendo de esta manera la evolución incremental del producto.

Ágil y adaptable al cambio:

El desarrollo de las partes formaliza solamente las características principales de la solución, priorizando los talleres y las comunicaciones entre las personas. Los clientes y funcionales están involucrados en el proyecto y poseen parte de las responsabilidad del éxito del mismo. Los cambios son conciliados semanalmente, discutidos y aprobados.

1.4 Modelo Vista Controlador (Model-View-Controller).

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

Modelo: administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: maneja la visualización de la información.

Controlador: controla el flujo entre la vista y el modelo (los datos).



Figura 1. Modelo Vista Controlador.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

Soporte de múltiples vistas: dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.

Adaptación al cambio: los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocio. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares o PDAs. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Una desventaja que tiene este modelo es el costo de actualizaciones frecuentes: Si el modelo experimenta cambios frecuentes, por ejemplo, podría desbordar las vistas con una lluvia de requerimientos de actualización.

1.5 Sistemas vinculados al proceso de modificación de los AFT .

Se investigó sobre aplicaciones existentes en nuestro país y en el ámbito internacional similares al proceso que se desea desarrollar, haciendo una caracterización y una descripción de sus principales funcionalidades, enfocando el análisis en cómo estos sistemas conciben o no, funcionalidades para la gestión de los AFT y si realizan o no el proceso de modificación de los AFT, de una forma que cumpla con las normativas y resoluciones actuales emitidas por el Ministerio de Finanzas y Precios, y se puedan estandarizar para las diferentes entidades del país. A continuación, se exponen algunos de estos sistemas ya existentes.

1.5.1 Sistemas nacionales.

Versat-Sarasola.

Permite enviar información eficaz, de forma inmediata, desde lugares apartados a la vez que ofrece mayor organización, control y disciplina en cada gestión. Constituido por 12 módulos que incluyen configuración, seguridad, contabilidad general y de gastos, costos y procesos, análisis económico empresarial y control de activos fijos. Además, interviene finanzas y cajas, planificación y presupuestos, control de inventarios, de productos terminados, pago de salario, paquete de gestión, contratación y facturación.

Dentro de los subsistemas que posee el Versat se encuentra activos fijos, el cual recoge las operaciones normales que en esta actividad se realizan, como son: Altas, bajas y modificaciones de los activos, las cuales son consideradas como movimientos. También posee un asistente para la configuración del subsistema.

Algunas de las funciones de este subsistema para el control de AFT son las siguientes:

- Conceptualiza los diferentes movimientos y posibilita configurar la contabilización de los mismos.
- Realizar inventarios a partir de diferentes selecciones, determinar las posibles diferencias y contabilizar las mismas.
- Permite definir diferentes ciclos de depreciación de los activos fijos teniendo en cuenta las características de la entidad.
- Permite el control de los activos en diferentes monedas.
- Ofrece variedad de reportes sobre las existencias y movimientos de los activos.

Inventuris-Fijos.

Este programa le permite administrar su asignación al personal de su empresa a través de la emisión de resguardos, registrar la ubicación física del bien en sus instalaciones, administrar su almacén de mobiliario sin asignar, generar el reporte de depreciación acumulada con actualización fiscal y muchas otras características que le son necesarias a todo administrador de bienes materiales. El sistema está preparado para generar las etiquetas de activo fijo y para trabajar con colectoras de datos portátiles.

Rodas XXI versión 3.0.

Sistema multiempresa y multiusuario desarrollado en Cuba por CITMATEL para la automatización de la gestión empresarial. Rodas XXI 3.0 contiene un módulo de Finanzas que contiene el control de la caja. Como aspecto a señalar Rodas XXI trabaja con doble moneda así como permite la realización de las operaciones con múltiples monedas.

Además permite tener un control detallado de los activos fijos de su entidad, realizando en el mismo momento que se registra un movimiento su contabilización. Se pueden realizar varios tipos de operaciones a activos fijos con facilidad en el momento que se desee, generando el documento asociado al movimiento de que se trate de forma automática con previa configuración del sistema para ello. Permite el control por separado de los activos fijos que se encuentran en almacén de los que se encuentran en explotación. Es posible además realizar ajustes a los activos fijos con facilidad mediante una opción que brinda el sistema.

Este módulo da la posibilidad de realizar la depreciación de forma automática mediante opción y no permite el cierre de períodos contables si no se realizó la depreciación del período a cerrar. El comprobante de depreciación se genera, al igual que con los movimientos de activos fijos, de forma automática. Todos los comprobantes que se generan por las distintas operaciones pueden ser visualizados antes de ser exportados al módulo de Contabilidad de RODAS XXI. Tanto los asociados a movimientos de activos fijos como el de la depreciación mensual. Los comprobantes pueden ser exportados cada vez que se desee siempre que se haya realizado alguna operación, lo que genera la consiguiente contabilización.

Cuenta con una gran variedad de opciones de informes, entre ellos podrá obtener el submayor de activos fijos, listados y localización de los medios de transporte de la entidad, la depreciación mensual y acumulada de uno o de los activos fijos que desee, el acta de responsabilidad material de cada una de las áreas y muchos otros, cubriendo con seguridad todas las necesidades de información que sobre los activos fijos su entidad requiere. Garantiza la mayor seguridad y trazabilidad de las operaciones que se realizan en el mismo. Cada usuario accederá al sistema con una clave individual y las operaciones que se realicen, quedarán automáticamente registradas en un sistema de bitácora que registra quien la realizó, cuando la realizó y que operación realizó.

Permite además visualizar información correspondiente a períodos anteriores, tan sólo con cambiar de período contable a períodos anteriores ya cerrados, aunque en dichos períodos no podrá realizar ninguna operación.

Redbeam.

RedBeam es utilizado para rastrear activos fijos e inventario en menores cantidades. Este sistema basado en Windows viene listo y fácil de instalar en una computadora.

Rastreo de activos fijos

La edición estándar del sistema es una base de datos completa de activos fijos con la habilidad de utilizar lectores de códigos de barras enlazados a terminales de PC (Computadora Personal). Este sistema es utilizado para mantener registro de sus activos al día y tomar un inventario físico periódicamente. Los campos utilizados incluyen ID (identificador) de código de barras, ubicación, departamento, mantenimiento, compra, garantía, préstamo, depreciación, información sobre tecnología y otros 15 campos definidos por el usuario.

La edición móvil del sistema le permite realizar las mismas actividades que la edición estándar utilizando computadoras móviles con capacidad de escáner. Esta edición, le elimina la restricción de realizar las operaciones en un solo lugar. El sistema incluye licencias ilimitadas para computadoras móviles compatibles.

1.5.2 Sistemas internacionales.

Advance Assets.

Software para control interno y administración de activos fijos. Sus características le ofrecen la posibilidad de hacer el levantamiento de inventario periódicamente con una terminal portátil según sus necesidades. De acuerdo a las ubicaciones de los activos, puede llevarse a cabo el levantamiento para detectar los activos que están en su lugar o fuera de él.

El software se encarga de almacenar una base de datos con toda la información de los bienes de la empresa. Sus características permiten tener informes inmediatos en cualquier momento y obtener datos históricos sobre las asignaciones de cada activo, además de llevar a cabo de ser necesario, la depreciación fiscal de cada uno. Este proceso puede ser mensual, trimestral, semestral o anual. La clasificación de sus activos en el sistema puede ser tan detallada como sea necesario, haciendo más fácil el control en diversos tipos de compañías. El proceso completo de controlar bienes y activos, solo se reducirá a poner una etiqueta de identificación y periódicamente leer un código de barras, una tarea que le ahorrará tiempo y recursos valiosos a su compañía.

SAP N72 – Gestión de activos fijos.

Este building block provee una evaluación, implementación y soporte para los procesos de Gestión de activos fijos en SAP ECC 6.0. El propósito es reducir el esfuerzo de implementación y encontrar los requerimientos exactos de la empresa de la mejor manera posible. El building block de gestión de activos fijos para cada país puede ser integrado con los sistemas existentes y también puede ser usado junto a nuevos sistemas que se estén implementando.

La gestión de activos fijos se utiliza para administrar y documentar en detalle las transacciones de activos fijos. El sistema le permite actualizar amortizaciones y los cambios en el valor de activos en el balance en la contabilidad del libro mayor del sistema SAP ECC 6.0. Se pueden realizar diferentes asignaciones de cuentas en contabilidad de costos para estas actividades.

Lista de funcionalidades

Las siguientes funcionalidades son provistas por el building block de gestión de activos fijos:

- Datos maestros
- Transacciones de activos
- Capitalización directa
- Procesos periódicos

Software para la administración del activo fijo (SAAF).

Identifica cada activo de acuerdo a sus características, especificaciones, ubicación, persona responsable del mismo, centro de costo y cuenta contable, de forma tal que siempre cuente con información confiable respecto a sus activos. Como apoyo se utiliza un software para la administración de activos fijos, que ha sido diseñado por personal de mucha experiencia en el manejo del activo fijo y que se ha ido perfeccionando a lo largo del tiempo, siendo una alternativa buena y económica para la administración del patrimonio.

SAAF ofrece las siguientes ventajas:

- Control del Inventario detallado de activos fijos, incluyendo descripciones detalladas y relación de padres e hijos para cada activo.
- Actualización de valores de activos fijos por índices específicos.
- Carga de los valores de avalúo.
- Selección entre activos fijos capitalizables y controlables.
- Multimoneda.
- Consulta de imágenes digitalizadas para cada activo (se pueden incluir facturas, pedimentos, fotografías, póliza de seguro, etc.)
- Opera en red (es multiempresa y multiusuario).
- Compara inventarios físicos contra inventario del software.
- Control de cambios (registra cualquier movimiento y lo guarda en memoria).
- Generación automática de resguardos por persona o centro de costo.

Gestión de activos empresariales InforEAM Enterprise Asset Management.

InforEAM permite a fabricantes, distribuidores y organizaciones de servicio, ahorrar tiempo y dinero con la optimización de recursos de mantenimiento, la mejora de la productividad de equipos y personal, una gestión eficaz de su inventario y el aumento de posibilidades de ingresos por reclamaciones de garantías. Las herramientas de generación de informes de InforEAM facilitan la toma de decisiones con vistas a la mejora de la gestión y rentabilidad del rendimiento de activos.

El software de gestión de activos empresariales de InforEAM puede ayudar a:

- Aumentar la eficacia de la mano de obra ya sea propia o subcontratada.
- Implementar eficaces programas de mantenimiento a fin de reducir los tiempos de parada.
- Gestionar procesos de órdenes de trabajo para garantizar entregas puntuales.
- Reducir los niveles de stocks del Inventario para un mayor ahorro.

- Analizar distintos escenarios para determinar los niveles óptimos de stocks y facilitar la toma de decisiones.
- Hacer un seguimiento exhaustivo de contratos para aumentar los ingresos por reclamaciones de garantías.

1.5.3 Limitaciones de los sistemas existentes.

- No abarcan en su totalidad las operaciones económicas que se llevan a cabo en las entidades cubanas.
- Son soluciones de escritorio, lo que requiere que el programa este instalado en cada PC, ocupando así muchos más recursos del sistema y espacio en el disco duro que una aplicación web.
- Ninguno de estos sistemas se acoge a las nuevas normas emitidas a finales del 2008 por el Ministerio de Finanzas y Precios de Cuba.
- Las soluciones están desarrolladas sobre plataformas propietarias, o sea, no cumplen con el principio de independencia tecnológica.
- No todos se adaptan a las características de las empresas. Esto trae consigo que existan empresas con más de un sistema en uso para diversas funciones.
- Algunos no soportan mecanismos estándares de integración con otras aplicaciones.

1.6 Notación para el modelado de procesos de negocio (BPMN)

La notación para el modelado de procesos de negocio -business process modeling notation (BPMN), es una notación gráfica normalizada que consiente en el modelado de los procesos de negocio, en un formato de flujo de trabajo (workflow).

Su principal objetivo es suministrar una notación estándar que sea sencillamente legible y entendible por parte de todos los implicados e interesados del negocio (stakeholders). Entre estos interesados están los analistas de negocio, los gerentes, los desarrolladores técnicos y los administradores del negocio. En síntesis, BPMN tiene el propósito de utilizarse como lenguaje común para cerrar la brecha de comunicación que frecuentemente se crea entre el diseño de los procesos de negocio y su implementación.

Proceso de negocio

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas.

Las entradas son requisitos que deben tenerse antes de que una función pueda ser aplicada. Cuando una función es aplicada a las entradas de un método, tendremos ciertas salidas resultantes.

Los procesos describen como es realizado el trabajo en la organización y se caracterizan por ser observables, medibles, mejorables y repetitivos.

1.7 Lenguajes, herramientas, marcos de trabajo y tecnologías utilizadas.

El lenguaje, las herramientas y las tecnologías que se describen en este capítulo son las utilizadas en la implementación de los componentes del software. Fue una decisión tomada por el equipo de arquitectura del software del Centro de Informatización para la Gestión de Entidades (CEIGE).

1.7.1 Lenguajes.

Lenguaje unificado de modelado (Unified Modeling Language, UML).

UML es ante todo un lenguaje. Un lenguaje que proporciona un vocabulario y reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones principales:

- Visualizar: UML permite expresar de una forma gráfica un sistema de manera que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.

Javascript.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, con tecnología orientada a objetos, con funciones y estructuras de datos complejos. Además, Javascript pone a disposición del programador todos los elementos que forman la página web, para que este pueda acceder a ellos y modificarlos dinámicamente.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo.

Por otro lado, Javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Comparte muchos elementos con otros lenguajes. Hay que tener en cuenta que este lenguaje es muy semejante a otros como C, Java o PHP, tanto en su formato como en su sintaxis, aunque por supuesto tiene sus propias características definitorias.

PHP 5.2.4.

Acrónimo de hypertext preprocessor, es un lenguaje interpretado “del lado del servidor” (esto significa que PHP funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas web dinámicas. Puede ser incluido con facilidad dentro del código HTML, y permite una serie de funcionalidades tan extraordinarias que se ha convertido en el favorito de millones de programadores en todo el mundo.

A continuación se presentan algunas características del lenguaje de programación PHP:

- Permite las técnicas de programación orientada a objetos; no requiere definición de tipos de variables y tiene manejo de excepciones.
- Presenta la capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.

HTML.

Es un lenguaje de marcado orientado a la publicación de documentos en Internet. La mayoría de las marcas son semánticas, debido a la amplia variedad de dispositivos donde se va a mostrar la información, aunque también existen algunas marcas físicas.

HTML es un lenguaje extensible, al que se le pueden añadir nuevas características, marcas y funciones. Desde su nacimiento HTML ha ido evolucionando, incorporando elementos introducidos por distintos fabricantes como Netscape y Microsoft y definiendo sucesivos estándares⁹ (como el HTML 6, el estándar actual). (Valle, pp. 20-21)

XML.

XML es un subconjunto de SGML (standard generalised mark-up language⁴¹), simplificado y adaptado a Internet. Este metalenguaje nos permite definir lenguajes de marcado adecuados a usos

determinados. Es un estándar internacionalmente reconocido y no pertenece a ninguna compañía por lo que su utilización es libre.

Características de XML:

- Diseño de lenguaje marcado en cualquier dominio específico.
- Auto describe sus datos.
- Intercambio de datos entre aplicaciones.
- Datos estructurados e integrados.

Ventajas de XML:

- Separa radicalmente la información o el contenido de su presentación o formato.
- Diseñado para ser utilizado en cualquier lenguaje o alfabeto.
- Permite poderosas técnicas de extracción de información y minería de datos.
- XML + validación = datos autodefinidos.

Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato.

1.7.2 Herramientas.

Visual Paradigm 6.3.

Visual Paradigm para UML 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 mejor calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta case UML también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos de UML.

Lista de características:

- Diagramas de procesos de negocio - proceso, decisión, actor de negocio, modelado colaborativo con CVS y Subversion (nueva característica).
- Ingeniería inversa-código a modelo, código a diagrama Ingeniería inversa Java, C++, esquemas XML, XML, NET exe/dll, CORBA IDL.
- Diagramas EJB-visualización de sistemas EJB³⁸.
- Generación de bases de datos - transformación de diagramas de entidad-relación en tablas de base de datos.

- Ingeniería inversa de bases de datos - desde sistemas gestores de bases de datos (DBMS) existentes a diagramas de entidad-relación.
- Distribución automática de diagramas - reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XML.
- Editor de figuras.

Zend Studio para Eclipse 6.0.

Es la última versión del popular entorno de programación integrado (IDE³⁹). Diseñado para desarrolladores profesionales de PHP, esta nueva versión combina un IDE versátil y potente con las capacidades de expansión del ecosistema del proyecto Eclipse.

Algunas de las nuevas funcionalidades de Zend Studio para Eclipse ayudarán a todas las compañías que desarrollan sus sistemas en PHP a disponer de un entorno mucho más flexible y profesional para controlar todo el ciclo de vida de un desarrollo.

Desde el punto de vista de un IDE completo, disponer de un buen debugger local con la conexión a los servidores de desarrollo, junto a una política de trabajo en equipo y un sistema de control de versiones es posible manejar sin problemas proyectos complejos en PHP.

PgAdmin III.

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3, ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración.

La aplicación también incluye un editor SQL¹⁹ con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets, y puede encriptarse²⁰ mediante SSL para mayor seguridad.

En Linux (Ubuntu y derivados de Debian), instalar el paquete pgadminII:

sedo apt-get install pgadmin3

En Windows, pgAdmin viene incorporado con el instalador de PostgreSQL, para instalarlo por separado descargar el zip y ejecutar pgadmin3.msi .

Los instaladores, si bien son multilinguaje, generalmente hay que establecer que lenguaje usará la interfaz.

TortoiseSVN 1.6.2.

Es un cliente gratuito de código abierto para el sistema de control de versiones²¹ *Subversion*. TortoiseSVN maneja ficheros²² y directorios²³ a lo largo del tiempo. Los ficheros se almacenan en un repositorio²⁴ central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. Esta es la razón por la que mucha gente piensa que Subversion, y los sistemas de control de versiones en general, son una especie de “máquinas del tiempo”.

Subversion, es un sistema general que puede ser utilizado para manejar colecciones de ficheros, incluyendo código fuente²⁵.

Características de TortoiseSVN:

- Integración con el shell de Windows: TortoiseSVN se integra perfectamente en el shell de Windows⁴² (por ejemplo, el explorador). Esto significa que puede seguir trabajando con las herramientas que ya se conocen.
No está obligado a usar el explorador de Windows. Los menús contextuales de TortoiseSVN también funcionan en otros administradores de archivos y en el diálogo Fichero/Abrir que es común a la mayoría de aplicaciones estándar de Windows. Sin embargo, debe tener en cuenta que TortoiseSVN está desarrollado con la mirada puesta en hacerle extensión del Explorador de Windows. Por este motivo, puede que en otras aplicaciones la integración no sea tan completa y que, por ejemplo, los íconos sobreimpresionados en las carpetas no se muestren.
- Iconos sobreimpresionados: El estado de cada carpeta y fichero versionado se indica por pequeños íconos sobreimpresionados. De esta forma, puede ver fácilmente el estado en el que se encuentra su copia de trabajo.
- Fácil acceso a los comandos de Subversion: Todos los comandos de Subversion están disponibles desde el menú contextual del explorador. TortoiseSVN añade su propio submenú allí.

PostgreSQL 8.3.

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos²⁶ y libre, publicado bajo la licencia BSD.

Sus principales características son:

- Alta concurrencia: Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para: Números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP (IPv4 e IPv6), arrays.

Otras características:

- Claves ajenas también denominadas llaves ajenas o claves foráneas²⁷ (foreign keys).
- PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query²⁸ en inglés).
- Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS²⁹, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés).

1.7.3 Marcos de trabajo (Frameworks).

Los marcos de trabajo, no son más que arquitecturas definidas para un determinado dominio de la aplicación que contiene un conjunto de componentes implementados y sus interfaces bien definidas, estos componentes se pueden utilizar, redefinir y crear nuevos componentes.

El marco de trabajo utilizado para la creación del producto Cedrux es Sauxe el cual está compuesto por ExtJs, Zend, Doctrine y ZendExtJs estructurados en niveles o capas.

ExtJs 2.2.

ExtJS es una librería Javascript para la creación de aplicaciones enriquecidas del lado del cliente. Sus características principales son: Gran desempeño, componentes de interfaz de usuario personalizables, un buen diseño y documentación.

Dispone de un conjunto de componentes (widgets⁴⁰) para incluir dentro de una aplicación web, como cuadros y áreas de texto, campos para fechas, campos numéricos, combos, radiobuttons y checkboxes, editor HTML, árbol de datos, pestañas, barra de herramientas, menús al estilo de Windows y paneles divisibles en secciones.

Varios de estos componentes están capacitados para comunicarse con el servidor usando AJAX³².

Doctrine Framework.

Doctrine es un mapeador de objetos relacionales (ORM) para PHP 5.2.3 que contiene una potente capa de abstracción de bases de datos (dbal). Uno de sus principales características es la opción de escribir las consultas de base de datos en un objeto de propiedad orientada a SQL llamada Doctrine Query Language (DQL). Esto proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad, sin necesidad de la duplicación de código innecesaria.

Entre muchas otras cosas tienes la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases (convenientemente creadas siguiendo las pautas del ORM), a tablas de una base de datos.

Por otro lado, como la librería es bastante grande ésta tiene un método para ser 'compilada' al pasar a producción. Uno de sus rasgos importantes es la habilidad de escribir opcionalmente las preguntas de la base de datos orientado a objeto. Esto les proporciona una alternativa poderosa a diseñadores de SQL manteniendo un máximo de flexibilidad sin requerir la duplicación del código innecesario.

Zend Framework.

Es un framework para tratamiento de aplicaciones Web y servicios Web con PHP. Brinda soluciones para construir sitios web modernos, resistentes y seguros. Además es código abierto y trabaja con PHP5.

Características principales: proporciona los componentes que forma la infraestructura del patrón MVC, proporciona una capa de acceso a base de datos construida sobre PDO(PHP Data Object) pero ampliándola con diferentes características, posee mecanismos de filtrado y validación de entradas de datos, permite la conversión de estructuras de datos PHP a JSON y viceversa para su utilización en

aplicaciones AJAX, brinda las características necesarias para proveer y consumir servicios web vía REST, además de permitir consumir y proveer servicios web.

ZendExt Framework.

Es un framework código abierto, que está diseñado para PHP 5 y buenas capacidades de ampliación. Es elaborado a partir de Zend Framework desempeñando todas sus características. Este trae de novedoso un controlador vertical para el control de las acciones realizada por las vistas hacia el controlador, un motor de reglas para las validaciones en el servidor. Se le incluyó el IoC para la comunicación entre los módulos o componentes, la integración con el ORM Doctrine Framework para trabajo en la capa de abstracción a base de datos, el ExtJs Framework para el desarrollo de las vistas y un controlador de trazas para controlar las acciones del sistema (acción, excepciones, rendimiento, integración, y excepción de integración).

1.7.4 Tecnologías.

Cliente/Servidor.

La tecnología llamada cliente/servidor es actualmente utilizada en casi todas las aplicaciones administrativas e internet/intranet. Bajo este esquema, un servidor es un ordenador remoto, en algún lugar de una red, que proporciona información según se le solicite. Mientras que un cliente funciona en su computadora local, se comunica con el servidor remoto y pide a éste información.

Cliente web: El cliente web es un programa con el que el usuario interactúa para solicitar a un servidor web el envío de páginas de información. Estas páginas se transfieren mediante el protocolo HTTP.

Servidor web: El servidor web es un programa que está permanentemente escuchando las peticiones de conexión de los clientes mediante el protocolo HTTP. El servidor web también se ocupa de controlar los aspectos de seguridad, comprobando si el usuario tiene acceso a los documentos.

AJAX.

El término AJAX es un acrónimo de Asynchronous JavaScript and XML, que se puede traducir como "JavaScript asíncrono and XML".

El artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005 define AJAX de la siguiente forma:

“Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

Desde su aparición, se han creado cientos de aplicaciones web basadas en AJAX. En la mayoría de los casos, AJAX puede sustituir completamente a otras técnicas como flash. Además, en el caso de las aplicaciones web más avanzadas, pueden llegar a sustituir a las aplicaciones de escritorio.

(Pérez, 2008)

Apache HTTP Server 2.0

Es un servidor web HTTP de código abierto para plataformas³³ Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP³⁴.

Apache presenta entre otras características:

- Apache funciona en Linux y en otros sistemas de Unix. Linux y Apache van de la mano en el mundo empresarial de hoy en día. Considero que la aceptación de Linux en el mundo de los negocios ha hecho sencilla la entrada de Apache en ese territorio, por lo que Apache y Linux constituyen una poderosa combinación.
- Apache también funciona en Windows. Aunque Apache va a funcionar mucho mejor en Windows con la versión 2.0, ya se encontraba en el mercado de Windows con la versión 1.3.x.
- Soporte del protocolo HTTP. Apache es uno de los primeros servidores Web en integrar el protocolo HTTP. Es totalmente compatible con el estándar HTTP 1.1 y al mismo tiempo sigue siendo compatible con HTTP 1.0.
- Sencilla configuración basada en un poderoso archivo. El servidor Apache no posee una interfaz de usuario gráfica para su administración. Se trata de un sencillo archivo de configuración llamado httpd.conf que se puede utilizar para configurar Apache. Únicamente necesita su editor de texto favorito.

- Soporte de FastCGI. No todo el mundo escribe sus CGI; **Error! No se encuentra el origen de la referencia.** en Perl, ¿cómo pueden hacer sus aplicaciones CGI más rápidas? Apache también tiene una solución para esto. Utilice el módulo mod FCGI para implementar un entorno FastCGI dentro de Apache y haga sus aplicaciones FastCGI arranquen rápidamente.
- Soporte de host virtuales. Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales.
- Soporte de autenticación HTTP. Está preparado para autenticación basada en la digestión de mensajes, que es algo que los navegadores Web populares ya han implementado. Apache puede implementar autenticación básica utilizando tanto archivos estándar de contraseña como los DBM, llamadas a SQL o llamadas a programas externos de autenticación.

1.8 Navegador web

Un navegador, navegador red o navegador web (del inglés, *web browser*) es un programa que permite visualizar la información que contiene una página web (ya esté alojada en un servidor dentro de la World Wide Web o en uno local).

Mozilla Firefox 2.0.0.17

Mozilla Firefox es un navegador web libre descendiente de Mozilla Application Suite, desarrollado por la Corporación Mozilla, la Fundación Mozilla y un gran número de voluntarios externos.

Firefox es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL.

Para visualizar páginas web, Firefox usa el motor de renderizado³⁷ Gecko, que implementa algunos estándares web actuales además de otras funciones, algunas de las cuales están destinadas a anticipar probables adiciones a los estándares web.

Presenta entre otras las siguientes ventajas:

- Antimalware: Firefox protege el ordenador de virus, gusanos, troyanos y spyware. Si de forma accidental se entra en una web atacante, este notificará que el sitio es sospechoso y el por qué no es seguro.
- Opciones personalizadas de seguridad: controla el nivel de cuidado que debe tener Firefox con un sitio e introduce excepciones-sitios que no necesitan un tercer grado. Personaliza las opciones para las contraseñas, cookies, carga de imágenes y la instalación de complementos para una mejor experiencia de la Web.

- Menor impacto de memoria: liviano y dulce, Firefox usa menos memoria de tu ordenador mientras se está ejecutando.
- Gestión de memoria: con todas las nuevas funciones de gestión implementadas, Firefox 3 mantiene bajo control el uso de memoria. El recolector de ciclos XPCOM limpia de forma continua la memoria no usada. Además, se han arreglado cientos de fugas de memoria. (Navegador web Firefox)

Conclusiones parciales.

Se pudo desarrollar el diseño teórico de la investigación dejando claro los conceptos de objeto de estudio y campo de acción. Se demuestra que los sistemas informáticos que actualmente se utilizan para el proceso de modificación de AFT, son ineficientes para el país e incluso algunos con altos costos por sus licencias, por lo que se hace necesario la construcción de un sistema informático que se adapte a todas las políticas económicas actuales del país y sea capaz de reducir en tiempo y costo el control de los recursos financieros de todas las entidades.

Capítulo 2: Análisis y diseño del módulo de modificaciones de AFT.

2.1 Introducción.

Este capítulo aborda todo el análisis realizado para comprender el negocio, el diseño de la solución propuesta por los autores para satisfacer los requerimientos, así como los artefactos generados.

Como artefactos de análisis se exponen el mapa de procesos, la descripción de los procesos, la descripción textual de los requisitos, los prototipos de interfaz de usuario, el modelo conceptual y el diccionario de datos.

El diseño de la solución está compuesta por la arquitectura base definida por el grupo de arquitectura del proyecto ERP Cuba, el modelo de datos, el diagrama de clases del diseño y los patrones de diseño empleados.

2.2 Reglas del negocio

Las reglas de negocio se refieren a las políticas que deben efectuarse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

Para realizar las modificaciones a los AFT tanto de propiedad como de valor hay que crear un documento para realizar las modificaciones correspondientes y en el que deben existir AFT en el área para poder realizar dicha operación.

2.3 Mapa de procesos y descripción del negocio.

Para comprender mejor el sistema que se desea desarrollar es necesario iniciar la modelación del negocio, en este caso el modelado se realizará por procesos. El modelado de procesos del negocio radica en figurar la realidad de forma tal que esta pueda ser entendida y de ser necesario modificada con el fin de juntarle mejoras. Es muy importante contar con una notación que admita modelar con la mayor claridad posible la particularidad del negocio, concediendo obtener una visión del funcionamiento de una empresa u organización determinada, para ello se utiliza la notación BPMN.

2.3.1 Descripción de procesos de negocio.

Con el objetivo de entender los procesos de negocio y equilibrar las necesidades reales de los clientes y usuarios, se ejecutaron varias entrevistas, tanto individuales como colectivas, las cuales fueron preparadas con antelación para garantizar la participación del personal necesario.

Como resultado de dichas entrevistas se obtuvo la descripción de los procesos. Estas descripciones de procesos fueron validadas posteriormente por los funcionales entrevistados.

Descripción de procesos.

Descripción del proceso adiciones, adaptaciones o mejoras, reconstrucciones y avalúos de AFT.

Objetivos	Definir el tratamiento a seguir en los casos de reparaciones, mejoras, reconstrucciones y avalúos de AFT.
Eventos que lo generan	Es el desembolso en que se incurre para adicionarle algo nuevo a un AFT y que aumenta su valor, adaptaciones o mejoras efectuadas a los AFT, que incrementen la capacidad de producción, la eficiencia, prolonguen la vida útil o ayuden a reducir los costos operacionales futuros, reconstrucciones o reparaciones extraordinarias que aumentan el valor del AFT original, el ajuste de los saldos de las cuentas que resulten de los avalúos realizados por entidades autorizadas.
Precondiciones	Se debe de contar con el Modelo "Movimiento de AFT".
Pos condiciones	Se crean el modelo "Movimiento de AFT" y el comprobante de operaciones.
Marco jurídico	Resolución 9 del 2007 del Ministerio de Finanzas y Precios. Resolución 87 del 2007 del Ministerio de Finanzas y Precios. Resolución 10 del 2007 del Ministerio de Finanzas y Precios. Resolución 399 del 2002 del Ministerio de Finanzas y Precios. Resolución No. 235 del 2005, del Ministerio de Finanzas y Precios, en su Anexo Único sobre las Normas de Valoración y Exposición
Clientes internos	Activos fijos.
Clientes externos	Contabilidad general.
Entradas	Modelo "Movimiento de AFT".

Salidas

Modelo "Movimiento de AFT".
Comprobante de operaciones.

Descripción del flujo básico de adiciones de AFT .**1. Aumentar valor de la adición:**

Se aumenta el valor del activo por el valor de la adición.

2. Estimar la Tasa de Depreciación:

Sobre este nuevo valor del activo y en base al número de años que le falten por depreciar, estimar la tasa de depreciación.

3. Concluye el proceso

Para visualizar el diagrama del proceso consulte el Anexo 1.

Descripción del flujo básico de adaptaciones o mejoras de AFT .**1. Costo de las mejoras realizadas:**

Se debe cargar al valor del activo, el costo de las mejoras realizadas.

2. Cancelar el valor del activo la parte sustituida:

Se debe cancelar del Valor del activo la parte del mismo que ha sido sustituida.

Restricciones: Esta actividad solo se realizará si se cambiaron partes.

3. Cancelar la depreciación acumulada de la parte reemplazada:

Se debe cancelar la depreciación acumulada correspondiente a la parte reemplazada.

4. Estimar la nueva depreciación anual:

Sobre este nuevo valor del activo y en base al número de años que le falten por depreciar, estimar la nueva depreciación anual.

5. Concluye el proceso**Descripción de las extensiones****1. a ¿Cambiaron partes?****1. a. 1 Estimar la nueva depreciación anual:**

Sobre este nuevo valor del activo y en base al número de años que le falten por depreciar, estimar la nueva depreciación anual.

Para visualizar el diagrama del proceso consulte el Anexo 2.

Descripción del flujo básico de reconstrucciones de AFT.

1. Dar de baja las partes sustituidas:

Si la reconstrucción ha sido parcial, se dan de baja las partes sustituidas, bien por su valor en Libros si existiese o por una estimación efectuada por peritos.

Restricciones: Esta actividad solo se realizará si la reconstrucción ha sido parcial.

2. Concluye el proceso.

Descripción de las extensiones

1. a ¿Reconstrucción parcial?

1. a. 1 Considerar como una nueva unidad de activo:

Si ha sido prácticamente total, su costo se considera como una nueva unidad de activo, dando consecuentemente de baja a la anterior.

Para visualizar el diagrama del proceso consulte el Anexo 3.

Descripción del flujo básico de avalúos de AFT.

1. Registrar como Gastos por Pérdidas:

Cuando la revaluación implica una disminución del valor del activo, se registra como gastos por pérdidas, afectando el resultado del período.

Restricciones: Esta actividad solo se realizará si la revaluación implica una disminución del valor del activo.

2. Concluye el proceso

Descripción de las extensiones

1. a ¿Si es incremento?

1. a. 1 Registrar la revaluación:

Cuando por parte de la entidad se ejecuta una revaluación de activos fijos, el importe que representa la revaluación se registra como un incremento del capital de la entidad por este concepto.

2. a ¿Hay depreciación de AFT revaluados?

2. a.1 Afecta resultado de los períodos económicos:

En los casos que exista capital por revaluación, este no se minorará por cargos directos que correspondan a la depreciación de los AFT revaluados. El importe de la depreciación afecta el resultado de los períodos económicos a los cuales se imputa.

Restricciones: En caso de que no haya depreciación de los AFT revaluados se terminaría el proceso.

Para visualizar el diagrama del proceso consulte el Anexo 4.

Para visualizar el diagrama de interacción con otros procesos consulte el Anexo 5.

2.3.2 Mapa de procesos.

Después de haber sido validadas las descripciones de los procesos, para un mejor entendimiento de la relación existente entre cada uno de los procesos, se procedió a la creación del siguiente mapa de procesos.

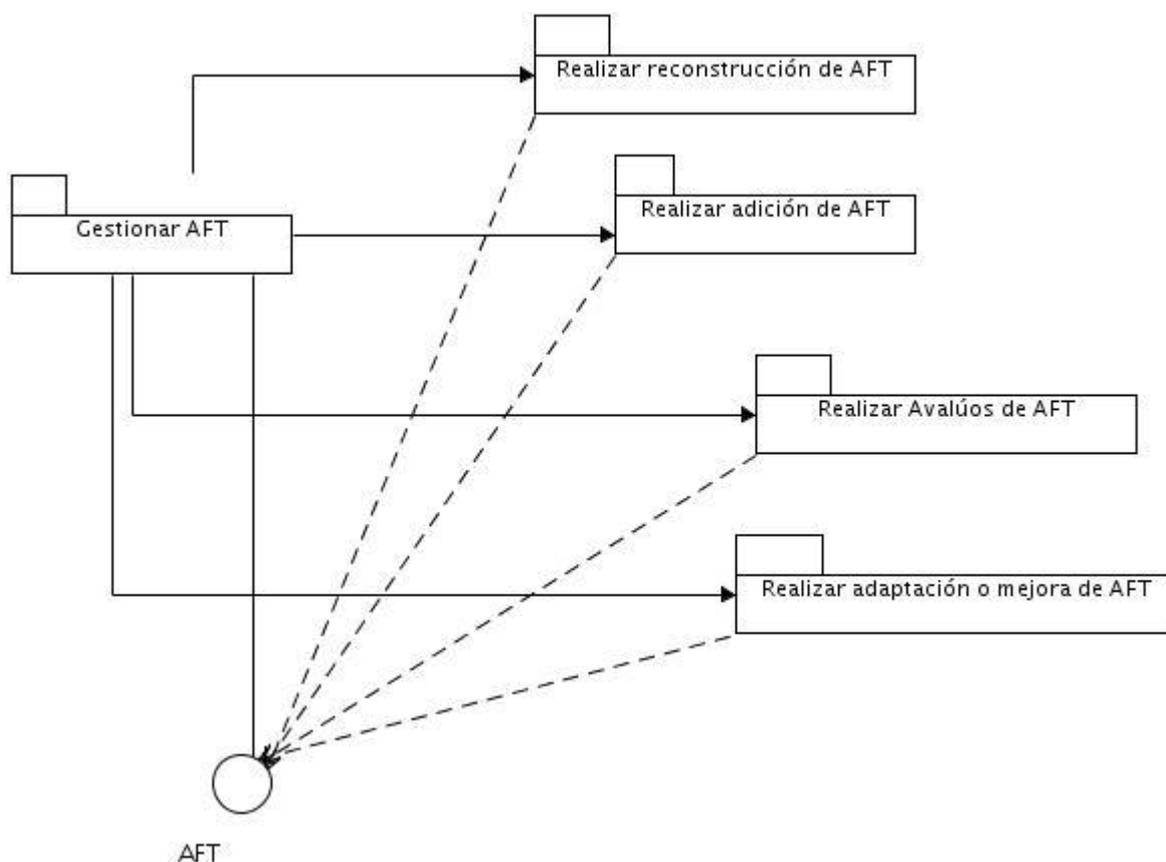


Figura 2. Descripción de Procesos de Negocio.

Gestionar AFT: en este proceso se llevan a cabo todas las actividades necesarias para la gestión de los AFT en la entidad, como son: crear, modificar y eliminar un AFT.

Realizar adición de AFT: en este proceso se llevan a cabo todas las actividades necesarias para aumentar el valor del AFT por el valor de la adición. Una adición no es más que el desembolso en que se incurre para agregarle algo nuevo a un AFT y que aumenta su valor, en la misma cantidad del desembolso realizado para la adición.

Realizar adaptación y mejoras de AFT: en este proceso se realizan mejoras a los AFT en la entidad. Si al ejecutarse la adaptación o mejora es necesaria la sustitución de partes, estas se dan de baja, bien por su valor en Libros si existiese o por una estimación efectuada por peritos.

Realizar avalúos de AFT: en este proceso se realizan ajustes a los saldos de las cuentas de AFT que resulten de los certificados periciales de valores emitidos por las entidades autorizadas a emitirlos.

Realizar reconstrucción de AFT: en este proceso se realiza una reconstrucción del AFT. Si la reconstrucción ha sido parcial, se dan de baja las partes sustituidas, bien por su valor en libros si existiese o por una estimación efectuada por peritos.

2.4 Descripción de los requisitos.

Requerimiento funcional adiciones, adaptaciones o mejoras, reconstrucciones y avalúos.

Las adiciones son el desembolso en que se incurre para adicionarle algo nuevo a un AFT y que aumenta su valor.

Igualmente son los gastos por adaptaciones o mejoras efectuadas a los AFT, que incrementen la capacidad de producción, la eficiencia, prolonguen la vida útil o ayuden a reducir los costos operacionales futuros, aumentan el valor del AFT existente y por tanto se capitalizan en forma conjunta con el bien existente, o por separado en dependencia con la naturaleza de la operación realizada y del bien incorporado.

Las reconstrucciones o reparaciones extraordinarias que aumentan el valor del AFT original, constituyen partidas capitalizables y también son consideradas como revalorizaciones. El avalúo es el ajuste de los saldos de las cuentas de AFT que resulten de los certificados periciales de valores emitidos por las entidades autorizadas a emitirlos.

Especificación del requisito: registrar una adición, adaptación, reconstrucción y avalúo.

Conceptos tratados	Conceptos	Atributos
	Depreciación	Tasa depreciación, depreciación acumulada, método depreciación, depreciación mensual.
	AFT	Descripción, CAFT, fecha alta, cuenta, subcuenta, valor inicial, No. inventario, vida útil, módulo, unidad básica o funcional, centro de costo, fecha explotación.
	Entidad	Entidad, dirección de la entidad, entidad receptora, Dirección receptora.
	Área	Área, área receptora.
	Submayores Específicos	No. submayor.
Precondiciones	Precondiciones	Pre-requisito
	El contador y especialista en contabilidad se han autenticado en el sistema.	Autenticar usuario.
Descripción	<p>Registrar los datos del documento que aprueba la realización de la modificación al valor del AFT.</p> <p>Seleccionar el AFT al cual le voy a modificar el valor.</p> <p>Registrar nuevo valor, ya sea por una adición, adaptación, reconstrucción o avalúos.</p> <p>Si se cambiaron partes se cancela la parte sustituida por su valor en Libros si existiese o por una estimación efectuada por peritos y la depreciación acumulada.</p> <p>Si es una reconstrucción total, el AFT se considera como uno nuevo.</p> <p>Estimar la nueva depreciación anual.</p> <p>Registrar la modificación.</p>	
Validaciones	El sistema valida los datos según lo descrito en <u>ERP-LOG-AFT-Modelo Conceptual-(avalúos, adaptaciones o mejoras, adiciones y reconstrucciones) v1.0</u>	

Post-condiciones	Se ha registrado una modificación.
Post-requisito	Generar el Comprobante de Operaciones.

Especificación del requisito: Modificar una adición, adaptación, reconstrucción y avalúo.

Conceptos tratados	Conceptos	Atributos
	Depreciación	Tasa depreciación, depreciación acumulada, método depreciación, depreciación mensual.
	AFT	Descripción, CAFT, fecha alta, cuenta, subcuenta, valor inicial, No. inventario, vida útil, módulo, unidad básica o funcional, centro de costo, fecha explotación.
	Entidad	Entidad, dirección de la entidad, entidad receptora, dirección receptora.
	Área	Área, área receptora.
	Submayores Específicos	No. submayor.
Precondiciones	Precondiciones	Pre-requisito
	El especialista encargado del control de los AFT en el área de contabilidad se ha identificado y autenticado ante el sistema.	Autenticar usuario
	Se ha seleccionado AFT con adaptación o mejora.	Registrar adaptación o mejora
Descripción	Seleccionar el AFT que se desea modificar. Modificar datos de la una adición, adaptación, reconstrucción o avalúo. Registrar modificación.	
Validaciones	El sistema valida los datos según lo descrito en <u>ERP-LOG-AFT-Modelo Conceptual-(avalúos, adaptaciones o mejoras, adiciones y reconstrucciones)</u>	

	<u>v1.0</u>
Post-condiciones	Queda actualizado el valor del AFT.
Post-requisito	Generar el Comprobante de Operaciones.

Especificación del requisito: consultar una adición, adaptación, reconstrucción y avalúo.

Conceptos	Conceptos	Atributos
tratados	Depreciación	Tasa depreciación, depreciación acumulada, método depreciación, depreciación mensual.
	AFT	Descripción, CAFT, fecha alta, cuenta, subcuenta, valor inicial, No. inventario, vida útil, módulo, unidad básica o funcional, centro de costo, fecha explotación.
	Entidad	Entidad, dirección de la entidad, entidad receptora, dirección receptora.
	Área	Área, área receptora.
	Submayores Específicos	No. submayor.
Precondiciones	Precondiciones	Pre-requisito
	El especialista encargado del control de los AFT en el área de contabilidad se ha identificado y autenticado ante el sistema.	Autenticar usuario
Descripción	Permitir registrar adaptación o mejora, modificar adaptación o mejora, eliminar adaptación o mejora.	
Validaciones	El sistema valida los datos según lo descrito en <u>ERP-LOG-AFT-Modelo Conceptual-(avalúos, adaptaciones o mejoras, adiciones y reconstrucciones)</u>	
	<u>v1.0</u>	
Post-condiciones	No procede	

Post-requisito

No procede

2.5 Diccionario de datos.

El diccionario de datos tiene incluido las características lógicas de las entidades y sus atributos que van a ser utilizados en el sistema y que fueron contenidos en el modelo conceptual, incluyendo nombre de la entidad, descripción de la entidad, nombre del atributo, descripción del atributo, tipo del atributo, restricciones del atributo, entre otras. A continuación se presenta el diccionario de datos con los conceptos que tienen mayor relevancia en el modelo.

Depreciación.

Nombre de la entidad	Depreciación					
Descripción de la entidad	Es el desgaste que se produce en los AFT.					
Nombre del atributo	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clase válida	Clases no válidas
Depreciación acumulada	Depreciación acumulada del AFT	Numérico	No	No	6 enteros 2 decim.	No numér.

Vida útil	Vida estimada del AFT	Numérico	No	No	1 a 99	No numér.
Tasa	Porciento de depreciación anual	Numérico.	No	No	1 a 99	No numér.
Depreciación Mensual	Importe depreciación mensual	Numérico.	No	No	4 enteros 2 decim.	No numér.

Submayores específicos.

Nombre de la entidad	Submayores Específicos					
Descripción de la entidad	Es el documento donde se puede encontrar las características de cada AFT en específico.					
Nombre del atributo	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clase válidas	Clases no válidas
Nro. submayor	Número consecutivo del submayor específico del AFT	Numérico	No	No	1 a 9999	No numér.

AFT.

Nombre de la entidad	AFT					
Descripción de la entidad	Son todos los AFT con sus atributos					
Nombre del atributo	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clase válidas	Clases no válidas
CAFT	Código del clasificador de AFT (antes CNMB)	Numérico	No	Sí	Según codificador Ent.	No numér.
Valor Inicial	Valor adquisición del AFT	Numérico	No	No	6 enteros 2 decim.	No numér.
Módulo	Módulo al que pertenece el AFT	Numérico	No	Sí	1 a 9999	No numér.
Unidad básica funcional	Unidad básica funcional	Numérico	No	Sí	1 a 9999	No numér.
Descripción	Descripción del AFT	Alfanumérico	No	No	Número/ caracter	Ninguno
Fecha alta	Fecha de alta del AFT día/mes/año	Numérico	No	No	Fecha	No numér.
Centro de costo	C .Costo al que pertenece el	Numérico	No	Sí	1 a 999	No numér.

	AFT					
Fecha explotación	Fecha de puesto en explotación el AFT	Numérico	No	No	Fecha	No numér.
Cuenta	Cuenta control del AFT	Numérico	No	No	Según codificador Ent.	No numér.
Nro. inventario	Número asignado al AFT	Numérico	No	Sí	1 a 9999	No numér.
Centro de costo	C. Costo al que pertenece el AFT	Numérico	No	Sí	1 a 999	No numér.
Vida útil	Vida estimada del AFT	Numérico	No	No	1 a 99	No numér.
Subcuenta	Subcuenta del AFT	Numérico	No	No	Según codificador Ent.	No numér.

Área.

Nombre de la entidad	Áreas					
Descripción de la entidad	Son todos los datos que poseen los documentos.					
Nombre del atributo	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clase válidas	Clases no válidas
Área	Área de la entidad	Alfanumérico	No	No	Número/ carácter	Ninguno

Área receptora	Área de la entidad receptora	Alfanumérico	No	No	Número/ carácter	Ninguno
----------------	------------------------------	--------------	----	----	------------------	---------

Entidad.

Nombre de la entidad	Entidad					
Descripción de la entidad	Son todos los datos que poseen los documentos.					
Nombre del atributo	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clase válidas	Clases no válidas
Entidad	Nombre de la entidad	Alfanumérico	No	No	caracteres	Ninguno
Dirección entidad	Dirección de la entidad	Alfanumérico	No	No	caracteres	Ninguno
Entidad receptora	Nombre de la entidad receptora	Alfanumérico	No	No	caracteres	Ninguno

2.6 Prototipo de interfaz de usuario.

La interfaz de usuario es la relación entre el usuario y el programa de procesamiento de datos. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso.

Los prototipos de interfaz de usuario son una propuesta que muestra el equipo de desarrollo a los especialistas funcionales para que los mismos validen que la interfaz examina las necesidades reales de los clientes y usuarios finales. Además de ser la base para que los desarrolladores implementen las interfaces de usuario finales.

Los prototipos de interfaces de usuario expuestos en el presente trabajo fueron validados por los especialistas funcionales, lo que compone una validación para los requisitos funcionales.

La elaboración de estos prototipos de interfaces se realizó con la herramienta Visual Paradigm 6.1.

Adiciones, adaptaciones o mejoras, reconstrucciones, a valúos.



Figura 3. Prototipo de interfaz de usuario para adicionar documento.

1. La funcionalidad señalada se encarga de crear un nuevo documento especificándole el tipo de documento a ser creado, en nuestro caso un nuevo documento de modificación.
2. La funcionalidad señalada gestiona los AFT del documento actualmente seleccionado.



Figura 4. Prototipo de interfaz de usuario para modificar AFT.

3. La funcionalidad "AFT" se encarga de modificar los AFT previamente seleccionados de un documento determinado. Los AFT adicionados se observan en la tabla "Datos de los AFT".

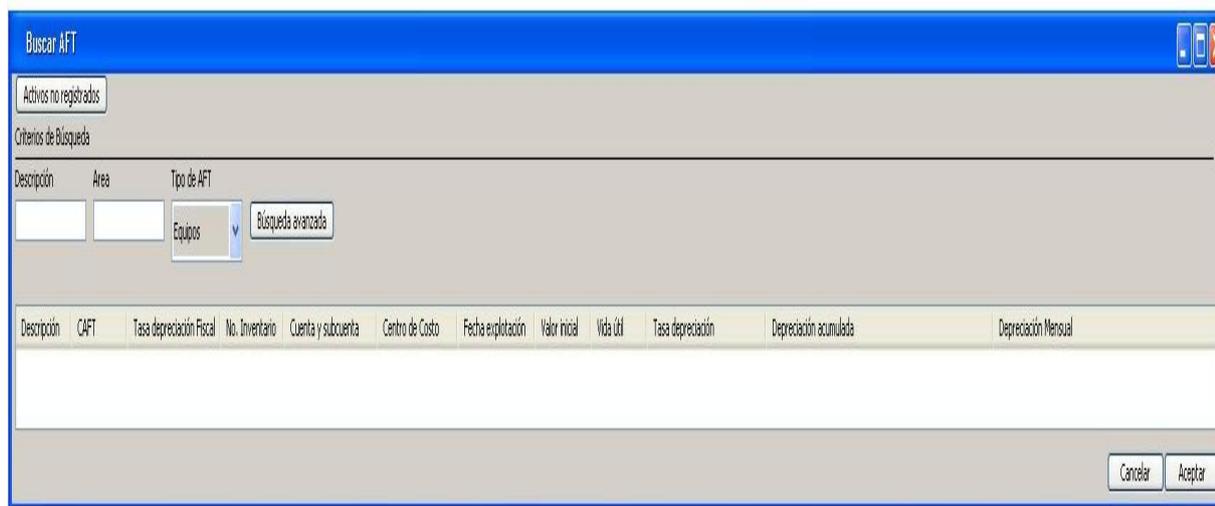


Figura 5. Prototipo de interfaz de usuario para adicionar un AFT a un documento.

Esta interfaz muestra los AFT existentes en la entidad, dichos activos son los que podrán adicionarse al documento de modificación.

2.7 Arquitectura base.

La arquitectura de software es la organización fundamental de un sistema, representada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. Establece los fundamentos para que analistas, diseñadores y programadores, trabajen en conjunto y así permitan el alcance de los objetivos del sistema, cubriendo todas las necesidades y expectativas.

Para el desarrollo de la solución se adoptó la propuesta de arquitectura base definida por la línea de arquitectura del proyecto ERP-Cuba para el producto Cedrux.

Vista vertical de la propuesta de arquitectura.

El desarrollo de la arquitectura está basado en componentes. La arquitectura basada en componentes tiene como objetivo hacer un uso correcto de software reutilizable, para la construcción de aplicaciones de software mediante la unión de partes ya existentes.

Un componente es un fragmento cambiante de un sistema de software, una unidad de composición con interfaces especificadas contractualmente, que satisface una o varias funcionalidades dentro del contexto de una arquitectura bien definida y puede ser ensamblado con otros fragmentos por medio de una interfaz. Un componente puede contener múltiples objetos, clases y otros componentes.

A continuación se expone la estructura propuesta por la línea de arquitectura para los componentes.

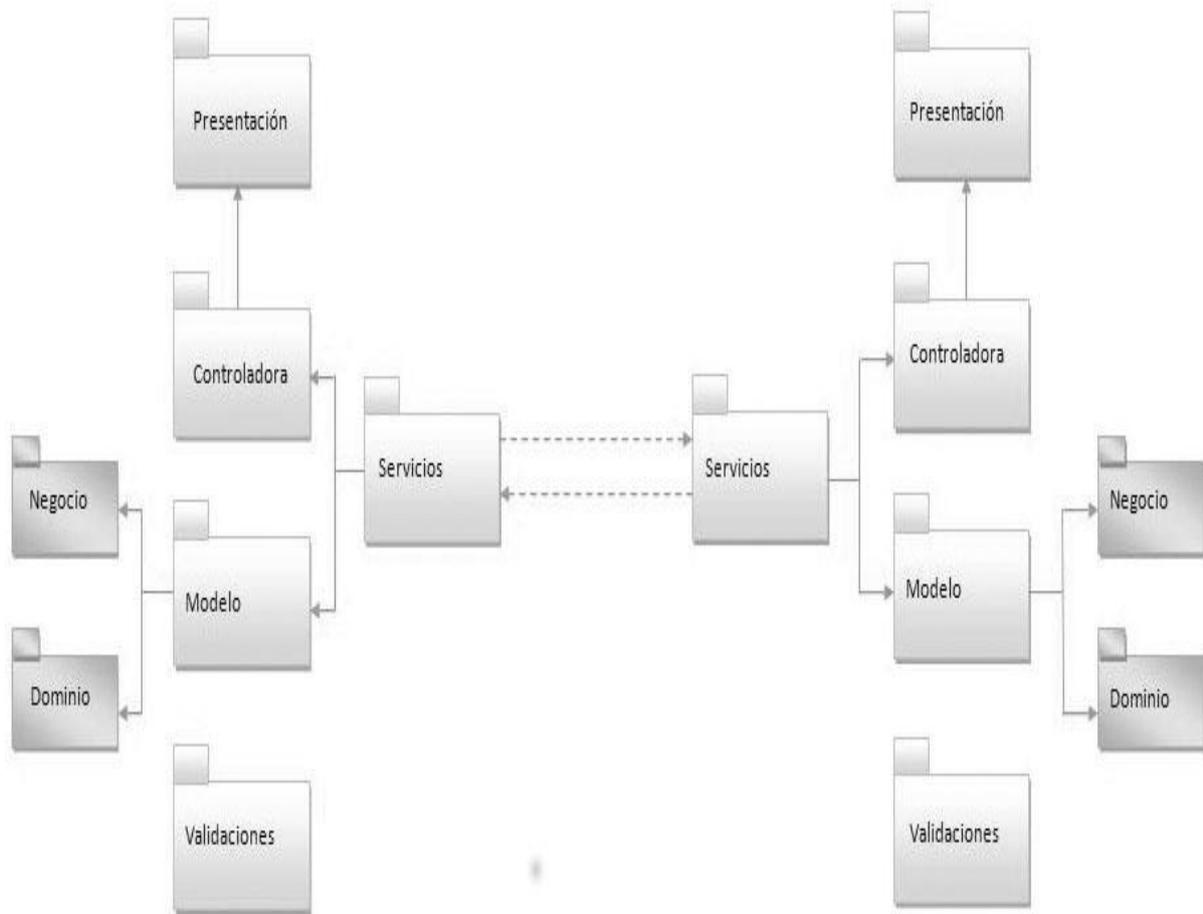


Figura 6. Estructura de los componentes.

2.8 Patrones de diseño utilizados.

Un patrón de diseño es una representación de clases y objetos notificándose entre sí conformada para dar solución a un problema de diseño general en un contexto particular. Identifica: Instancias, clases, colaboraciones, roles y la distribución de responsabilidades.

2.8.1 Patrones de asignación de responsabilidades.

Para la construcción del diseño, frecuentemente se emplean un grupo de patrones o modelos, que no son más que soluciones concretas y técnicas para lograr objetivos específicos.

Dentro de los más conocidos y usados se encuentran los GRASP (General Responsibility Assignment Software Patterns, Patrones Generales de Software para Asignación de Responsabilidades), que

describen los principios fundamentales de la asignación de responsabilidades a objetos. El empleo de este patrón tiene como objetivo la representación de los principios fundamentales del diseño de objetos para la asignación general de responsabilidades.

Los utilizados fueron:

- **Alta cohesión.**

Una clase con numerosa cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Su valioso grado de funcionalidad, ajustada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un acrecentamiento de la capacidad de reutilización. El patrón alta cohesión presenta semejanzas con el mundo real. Se sabe que si alguien asume excesivas responsabilidades, sobre todo las que debería delegar, no sería eficiente.

Ejemplo de uso de este patrón se aprecia en la clase `modificacionAftService.php`, encargada de mantener la comunicación de los datos con los diferentes componentes del subsistema AFT.

- **Bajo acoplamiento.**

Estimula establecer una responsabilidad de modo que su colocación no aumente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que agrandan la oportunidad de una productividad superior.

Ejemplo de uso de este patrón se evidencia en la clase `GestmodificacionesController.php`, encargada de acceder a los valores y métodos de las clases `Model`.

2.8.2 Patrón Gang of Four (GoF) Cuadrilla de Cuatro.

El patrón fachada clasificado como estructural dentro de los patrones GoF proporciona una interfaz sencilla unificada para un conjunto de clases o subsistemas, siendo más fácil de usar. Permite reducir la complejidad y disminuir las dependencias, los clientes acceden a los subsistemas por medio de la clase fachada, ella es la encomendada de reenviar las peticiones a los objetos de los subsistemas, por lo que no se accede directamente a los mismos, ocultando la complejidad de ellos.

Este patrón beneficia un Bajo Acoplamiento entre los clientes y los subsistemas, reconociendo a uno de los patrones GRASP, permiten alternar las clases internas, de manera transparente a los clientes que las usan.

Se logró proporcionar mediante el patrón fachada una interfaz de servicios simple para un subsistema tan complicado como lo es el de AFT, al cual pertenece el componente a diseñar. Su uso en el nivel más alto de abstracción, se somete al diseño general de la arquitectura del sistema, puesto que la existencia de dependencias entre el subsistema AFT y otros como por ejemplo: Inventario y Facturación, solicita la implementación de una clase fachada donde se publiquen los servicios necesarios para los otros subsistemas que proporcione la interacción. A nivel interno también se emplea este patrón para disminuir el grado de dependencia entre los componentes.

2.8.3 Patrón Modelo Vista Controlador.

En la propuesta, los componentes implementan el patrón Modelo Vista Controlador (MVC). El paquete de Servicios incluirá todas las clases y funcionalidades incluidas en los paquetes Controladora, Modelo, y Validaciones. Los paquetes Presentación, Controladora y Modelo forman los elementos del MVC y el paquete Validaciones define las reglas del negocio.

Dentro del paquete Modelo se encuentran las funcionalidades y clases contenidas en los paquetes Negocio y Dominio. El paquete Negocio implementa las funcionalidades del negocio y el paquete Dominio incluye las clases entidad. Cuando se desea solicitar un servicio a otro componente se accede a través del paquete Servicios quien analizará la solicitud e irá a la clase que tiene dicha funcionalidad recuperará el resultado y lo devolverá al solicitante.

Modelo: esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (frecuentemente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: esta capa maneja la visualización de la información, es decir, presenta el modelo en un formato apropiado para interactuar, que normalmente es la interfaz de usuario.

Controlador: esta capa controla el flujo de datos entre la vista y el modelo; es el que responde a eventos, frecuentemente acciones del usuario e invoca cambios en el modelo y probablemente en la vista; dependiendo la vista y el controlador del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual. En el caso de los diagramas de clases realizados para el diseño la página phtml representa la vista, las clases controladoras (controller), y en el modelo se representan las clases de la lógica del negocio y las de dominio que no son más que las clases de acceso a datos, quedando así lista la representación de este patrón y logrando que cualquier cambio que se realice en la vista no afecte a la lógica del negocio ni al dominio.

2.8.4 Patrón de integración entre componentes (Inversor de control).

El patrón Inversión de Control o como también se conoce Inyección de Dependencia (*Dependency Injection*) es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos (procedure calls) o funciones. Tradicionalmente el programador especifica la secuencia de decisiones y procedimientos que pueden darse durante el ciclo de vida de un programa mediante llamadas a funciones. En su lugar, en la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir. Se refiere también a la forma en que un objeto usa otro objeto.

2.9 Diagrama de clases.

A continuación se muestra el diagrama de clases correspondiente a la gestión de modificaciones de un AFT. Se puede observar el diseño teniendo en cuenta la estructura propuesta por la línea de arquitectura para los componentes.

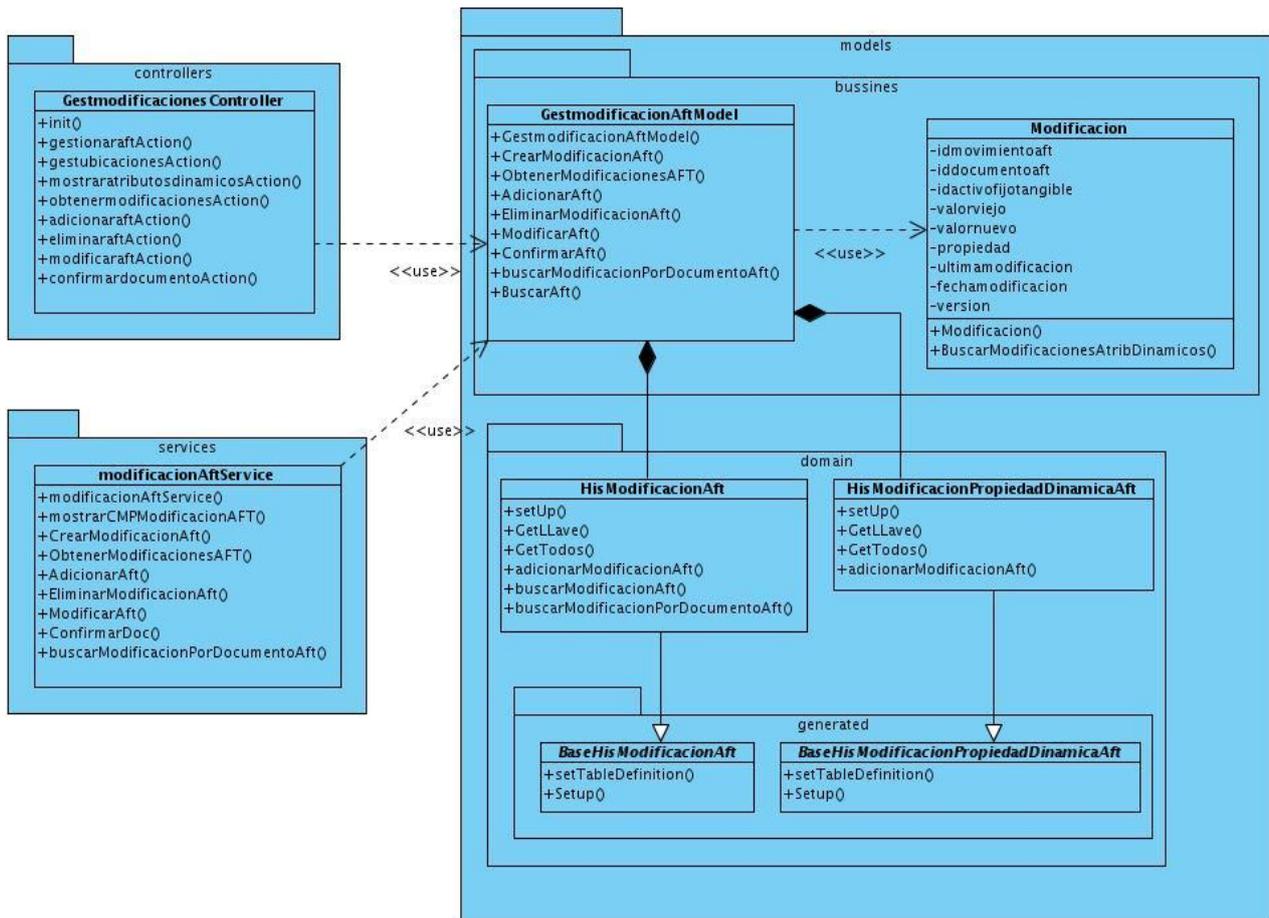


Figura 7. Diagrama de clases.

Clases controladoras.

Las clases de control coordinan el trabajo de uno o unos pocos casos de uso, coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso, por lo que definen el flujo de control y las transacciones dentro de un caso de uso delegando el trabajo a otros objetos.

Nombre: GestmodificacionesController	
Tipo de clase: Controladora	
Atributo	
Para cada responsabilidad	
Nombre	Descripción
gestionaraftAction	servicio
gestubicacionesAction	servicio
mostraratributosdinamicosAction	servicio
obtenermodificacionesAction	Obtiene las modificaciones de un documento
adicionaraftAction	Adiciona AFT a un documento
eliminarraftAction	Elimina AFT de un documento
modificaraftAction	Modifica los atributos de un AFT
confirmardocumentoAction	Confirma las modificaciones de los AFT a un documento

Clases auxiliares.

Son las que guardan poca o ninguna información del estado por sí misma, pero asisten en la ejecución de tareas complejas.

Nombre: GestmodificacionAftModel	
Tipo de clase: Auxiliar	
Atributo	
Para cada responsabilidad	
Nombre	Descripción
GestmodificacionAftModel	Constructor de la clase
CrearModificacionAft	Crea un objeto "Modificacion"

ObtenerModificacionesAFT	Obtener las modificaciones de un documento
AdicionarAft	Adicionar AFT a un documento
EliminarModificacionAft	Eliminar AFT a un documento
ModificarAft	Modificar los atributos de un AFT
ConfirmarAft	Confirmar las modificaciones de los AFT a un documento
buscarModificacionPorDocumentoAft	Busca las modificaciones de los AFT de un documento

Nombre: modificacionAftService	
Tipo de clase: Auxiliar	
Atributo	
Para cada responsabilidad	
Nombre	Descripción
CrearModificacionAft	Crea un objeto "Modificacion"
ObtenerModificacionesAFT	Obtener las modificaciones de un documento
AdicionarAft	Adicionar AFT a un documento
EliminarModificacionAft	Eliminar AFT a un documento
ModificarAft	Modificar los atributos de un AFT
ConfirmarAft	Confirmar las modificaciones de los AFT a un documento
buscarModificacionPorDocumentoAft	Busca las modificaciones de los AFT de un documento

Clases entidad.

Estas clases modelan información que poseen una larga vida y que a menudo son conceptos y sucesos que ocurren en el mundo real. La fuente principal de obtención son las clases entidades del negocio y el glosario de términos que se ha ido elaborando. Se encargan de modelar la información del sistema y el comportamiento asociado a una información.

Nombre: HisModificacionAft	
Tipo de clase: Entidad	
Atributo	
Para cada responsabilidad	
Nombre	Descripción
setUp	Crea un objeto "Modificacion"
GetLlave	Obtiene la llave o id representativo de la clase
GetTodos	Obtiene todos los atributos
adicionarModificacionAft	Adiciona/modifica valores de la modificación
buscarModificacionAft	Busca las modificaciones
buscarModificacionPorDocumentoAft	Busca las modificaciones de los AFT de un documento

Nombre: HisModificacionPropiedadDinamicaAft	
Tipo de clase: Entidad	
Atributo	
Para cada responsabilidad	
Nombre	Descripción
setUp	Crea un objeto "Modificacion"
GetLlave	Obtiene la llave o id representativo de la clase
GetTodos	Obtiene todos los atributos

adicionarModificacionAft	Adiciona/modifica valores de la modificación
--------------------------	--

2.10 Modelo de datos.

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información.

Un modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

En el siguiente modelo de datos se muestran las tablas que están directamente relacionadas con el proceso de Modificación de los AFT.

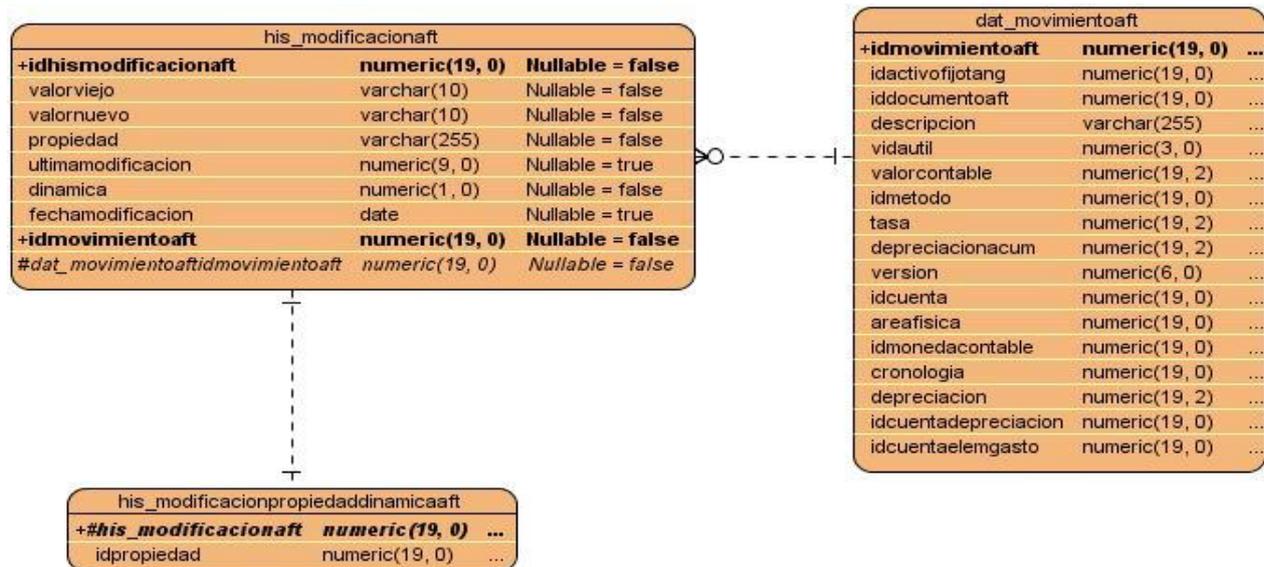


Figura 8. Modelo de datos del componente modificaciones del subsistema AFT.

Conclusiones parciales.

Mediante el desarrollo de este capítulo se logró realizar el análisis para comprender mejor el negocio, efectuar el diseño de la solución propuesta por los autores para satisfacer los requerimientos, así como los artefactos generados, para dar paso a la realización posterior de la implementación y validación de la solución propuesta.

Capítulo 3: Construcción y validación de la solución propuesta.

3.1 Introducción.

Basados en todo el análisis y diseño realizado se procede la construcción de la herramienta. En este capítulo se expone el diagrama de componentes, los estándares de codificación utilizados y la estructura de datos apropiada para la implementación de los algoritmos. Esta etapa incluye también la creación de interfaces, lógica de negocio, acceso a datos e integración con otros componentes.

Durante esta etapa también se realizan pruebas de caja blanca y de caja negra al sistema, para que asegure que “todas las piezas encajen”, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

3.2 Estrategias de integración.

La aplicación está especificada por tres capas. Capa de presentación (view), negocio (controller) y acceso a datos (models). Esta arquitectura facilita un trabajo seguro, rápido y eficiente. La integración vertical o llamada arquitectura en 3 capas consiste en el flujo de los datos desde la vista hacia la capa de datos y viceversa, pasando por los diferentes elementos que componen la arquitectura. Esta consta de cuatro nodos de integración, nodo situado entre la vista y el controlador, nodo entre el controlador y el modelo, nodo que relaciona el modelo con el framework doctrine y el que se encuentra entre el doctrine y la base de datos. Todo el código dentro de un mismo componente manipula llamadas a métodos o eventos de forma directa. La comunicación entre diferentes módulos y componentes se realiza mediante llamadas a la inversión de control. El IoC especifica respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que otro módulo o componente lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir.

Cada componente tiene su registro de los datos de los módulos en un fichero XML que será mapeado por el framework para el funcionamiento del mismo, dicho fichero tiene por nombre IoC y registra las funcionalidades que ofrecen los métodos de las clases control de los componentes del sistema. La base de datos es accedida de forma directa mediante controladoras y los componentes reusados son integrados mediante interfaces sencillas, garantizando así una total integración de las capas en el sistema.

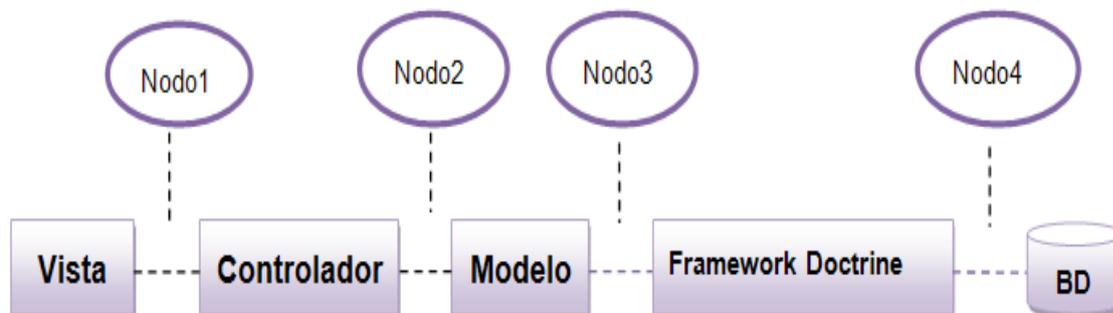


Figura 9. Estrategia de integración.

3.2.1 Servicios consumidos.

Servicio	Descripción
<i>ConfirmarDocAFT</i>	Permite la confirmación de un documento para salvar una modificación determinada.
<i>mostrarCMPAtributosDinamicos</i>	Devuelve los atributos dinámicos de un AFT.
<i>existeMovimientoAft</i>	Se utiliza con el objetivo de conocer si existe un movimiento, que tenga asociado un determinado documento con un AFT y un número de inventario dado.
<i>crearMovimientoAft</i>	Permite asociar documento con un AFT y un número de inventario en un movimiento.
<i>ObtenerGrupoPerteneceAft</i>	Se usa para saber a qué grupo pertenece un AFT determinado
<i>ObtenerAtributos</i>	Permite obtener los atributos dinámicos de un AFT
<i>buscarValorAtributoActivo</i>	Permite obtener el valor de un atributo de un AFT determinado.

3.2.2 Servicios Brindados.

Servicio	Descripción
AdicionarAft	Se brinda para posibilitar la adición de un AFT a una modificación determinada.
EliminarModificacionAft	Sirve para posibilitar la eliminación de un AFT a una modificación determinada.
ModificarAft	Permite que desde otro componente se pueda modificar un AFT de una modificación determinada.
ConfirmarDoc	Sirve para que el documento pueda confirmar la modificación de un AFT determinado.
buscarModificacionPorDocumentoAft	Se brinda para poder obtener las modificaciones de un documento.
CrearModificacionAft	Sirve para crear una nueva modificación.

3.3 Estándares de código.

Los estándares de codificación son pautas de programación las cuales no enfatizan la lógica del programa, sino su estructura y apariencia física para proporcionar la lectura, comprensión y mantenimiento del código. Los estándares de codificación permiten una mejor integración entre las líneas de producción y establece modelos que conlleven a lograr un código más legible y reutilizable, de tal forma que se pueda extender su existencia a lo largo del tiempo.

Notación Camello:

Se usa para denotar variables, parámetros y funciones. En esta notación, si el identificador es una palabra simple se escribe todo con minúscula, pero si es compuesta, la primera letra de todas las palabras que vienen a continuación de la primera comienza con mayúscula. La primera palabra debe ser un sustantivo que describa claramente al identificador y las otras palabras a continuación deberán ser adjetivos.

Ejemplo de función: `adicionarAftAction()`.

Ejemplo de variable: `$arrAft`.

Notación Pascal:

Se usa para denotar clases. En esta notación, si el identificador es simple, el primer carácter se escribe con mayúscula y el resto con minúscula, si el identificador es una palabra compuesta, la segunda palabra debe comenzar con mayúscula también.

Ejemplo: GestModificaciones.

3.4 Estructura de datos apropiada para la implementación de los algoritmos.

Para lograr una mayor organización cuando se desea almacenar elementos como parte del lenguaje de programación que necesita el programador se han utilizado las estructuras de datos que nos brinda esa posibilidad. Existen varias estructuras de datos, por lo que hay que tener en cuenta varios aspectos para escoger la que se utilizará: la continuidad con la que se usarán los datos que en ella se guarden, es lo más importante, pues la simplicidad o la complejidad que las mismas poseen, pueden hacer la ejecución del sistema mucho más lento o de mayor rendimiento.

En este caso la estructura de datos a utilizar se denomina Arreglos o Arrays, los mismos son variables complejas, que almacenan gran cantidad de valores dentro de una sola variable, la capacidad del mismo puede estar concretada por el programador en caso de ser necesario.

Su información puede ser cambiada y manejada de manera fácil. La ventaja de usar los arreglos de PHP es que dicho lenguaje tiene una gran variedad de funciones para el uso de los mismos, entre ellas están las funciones de ordenamiento. Además este lenguaje nos facilita mucho el recorrido los arreglos.

3.5 Clases del componente: Modificaciones.

Descripción de la clase controladora.

Nombre: GestmodificacionesController.	
Tipo de clase: Controladora.	
Para cada responsabilidad	
Nombre.	Descripción.
init()	Constructor de la clase.
gestmodificacionesAction()	Método encargado de cargar la interfaz de usuario.
adicionarraftAction()	Método para adicionar un AFT a un documento específico.

eliminarmodificacionAction()	Método para la eliminación de una modificación.
modificaractivoAction()	Método encargado de la modificación de los AFT ya sea por valor o por propiedad.
confirmardocumentoAction()	Método para guardar el nuevo valor del AFT.
mostraratributosdinamicosAction()	Método para obtener los atributos dinámicos de un AFT

Clases modelo del Componente: Modificaciones de AFT.

Nombre: GestmodificacionAftModel.	
Tipo de clase: modelo.	
Para cada responsabilidad	
Nombre.	Descripción.
AdicionarAft ModificarAft EliminarModificacionAft	Responsabilidades encomendadas de la realización del CRUD.
ConfirmarAft	Responsabilidad que se encarga de confirmar que se guardó un comprobante de operaciones a la modificación realizada.

3.6 Diagrama de componentes.

Los diagramas de componentes se representan como un grafo de componentes de software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. Son usados para estructurar el modelo de implementación en conjuntos como relaciones de componentes y subsistemas de implementación. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, ya sean estos componentes de código fuente, binarios o ejecutables. (IBM Corporation, 2003)

El subsistema para el control de los AFT en su primera versión permite gestionar todos los procesos desde el punto de vista contable de cualquier entidad empresarial o presupuestada del país. Puede trabajar integrado o no con el subsistema de contabilidad que brinda Cedrux y puede perfectamente

integrarse con cualquier otro. Se permite acceder solo a las funcionalidades que el usuario tenga privilegios, pudiendo crear los roles y establecerle los permisos que decida la entidad mediante el Sistema de Gestión Integral de Seguridad (SIGIS).

A continuación se mostrará el diagrama de integración de los componentes para los movimientos⁴³ y modificaciones de AFT, diagrama que forma parte de la solución arquitectónica del subsistema de AFT.

Aquí se aprecia la interacción del componente modificaciones con otros componentes. Dicho componente permite la gestión de los AFT al documento cuando la operación que se desea realizar es de modificación ya sea por valor, de propiedad. En el caso de las modificaciones de propiedad se permite cambiar el valor de cualquiera de las propiedades dinámicas que posee el AFT.

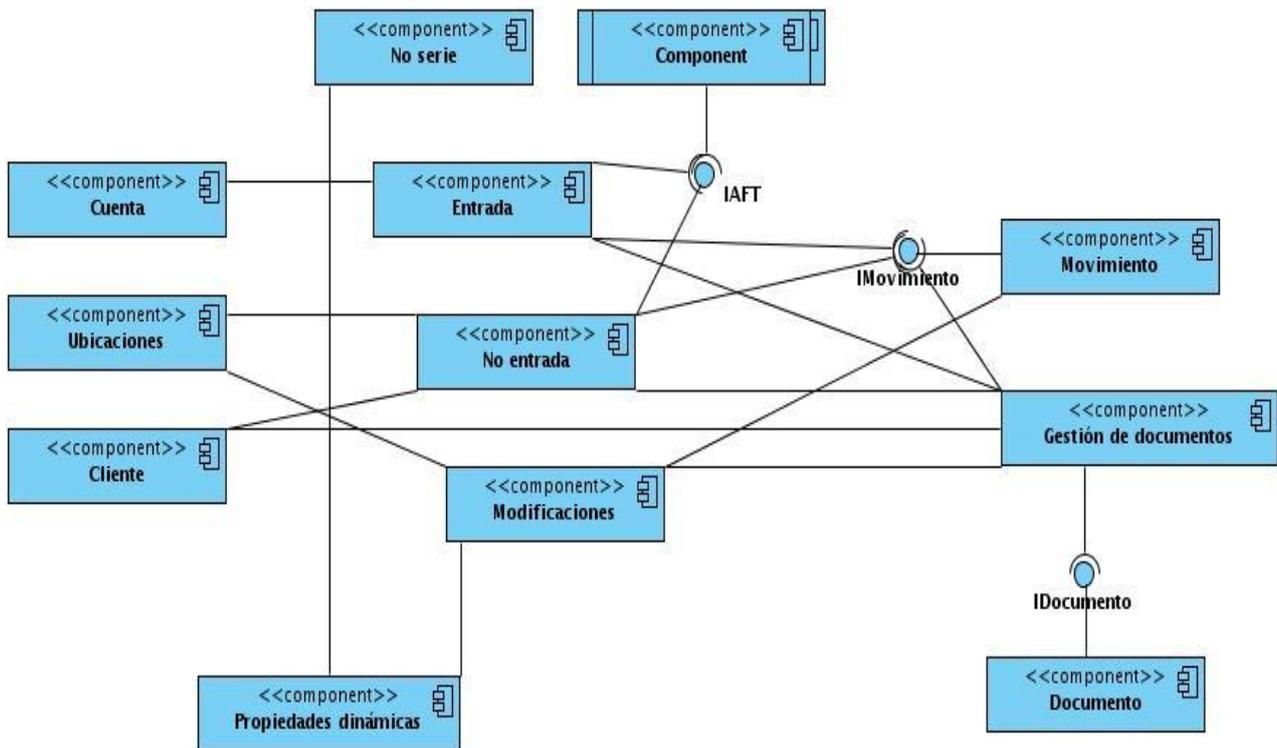


Figura 10. Diagrama de componentes del subsistema AFT.

3.7 Modelo de despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.

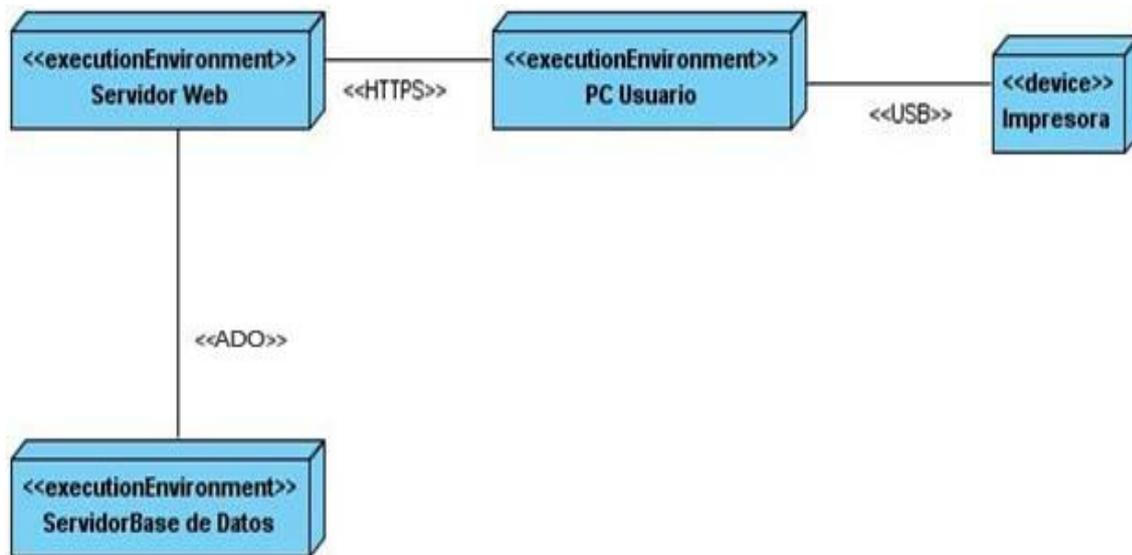


Figura 11. Modelo de despliegue del producto Cedrux.

3.8 Pruebas de software.

Las pruebas de software son las técnicas experimentales que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa informático. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Las pruebas de software se integran dentro de las diferentes fases del ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

3.8.1 Prueba de caja blanca.

Permiten inspeccionar la estructura interna del programa. Se diseñan casos de prueba para examinar la lógica del programa. Es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar casos de prueba que garanticen que:

- Se ejercitan todos los caminos independientes de cada módulo.
- Se ejercitan todas las decisiones lógicas.
- Se ejecutan todos los bucles.
- Se ejecutan las estructuras de datos internas.

Prueba del camino básico.

La prueba del camino básico es una técnica de prueba de la caja blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico.

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguieron para aplicar esta técnica son:

1. Enumerar las sentencias de código del procedimiento.

```

public function modificaractivoAction()
{
    $datos->dinamica = ($this->_request->getPost ('dinamica'))?1:0; #1
    $datos->idmovimientoaft = $this->_request->getPost ('idmovimientoaft'); #1
    $idactivo fijotang = $this->_request->getPost ('idactivo fijotang'); #1
    $idactivo fijotangible = $this->_request->getPost ('idactivo fijotangible'); #1
    $datos->fechamodificacion = $this->global->FechaContable->fecha; #1

    if($datos->dinamica == 1) #2
    {
        $grupo = $this->pIntegrator->activosFT->ObtenerGrupoPerteneceAFT($idactivo fijotang); #
        $arrAtributos = $this->pIntegrator->nomencldoresaft->ObtenerAtributos
        ($grupo[0]->DatActivoentidad[0]['idgruposubgrupo']); #2

        foreach ( $arrAtributos as $index => $atributo ) #3
        {
            $buscado = $this->pIntegrator->nomencldoresaft->
            buscarValorAtributoActivo($idactivo fijotangible,$atributo->idatributo);#3
            if ($buscado) #5
                $valorviejo = $buscado[0][valor];#6
            else
                $valorviejo = ''; #7
            $datos->idpropiedad = $atributo->idatributo; #8
            $datos->propiedad = $atributo->nombremostrar;#8
            $datos->valornuevo = $this->_request->getPost ($atributo->nombremostrar);#8
            $datos->valorviejo = $valorviejo;#8
            $this->modificacion->ModificarAft ($datos);#8
        }
    }
    else
    {
        $datos->propiedad = $this->_request->getPost ('descripcion'); #4
        $datos->valorviejo = $this->_request->getPost ('valorviejo'); #4
        $datos->valornuevo = $this->_request->getPost ('valornuevo'); #4

        $this->modificacion->ModificarAft ($datos); #4
    }

    echo ("{'codMsg':1,'mensaje':'Activo modificado'}");#9
}

```

Figura 12. Método modificaractivo.

2. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.

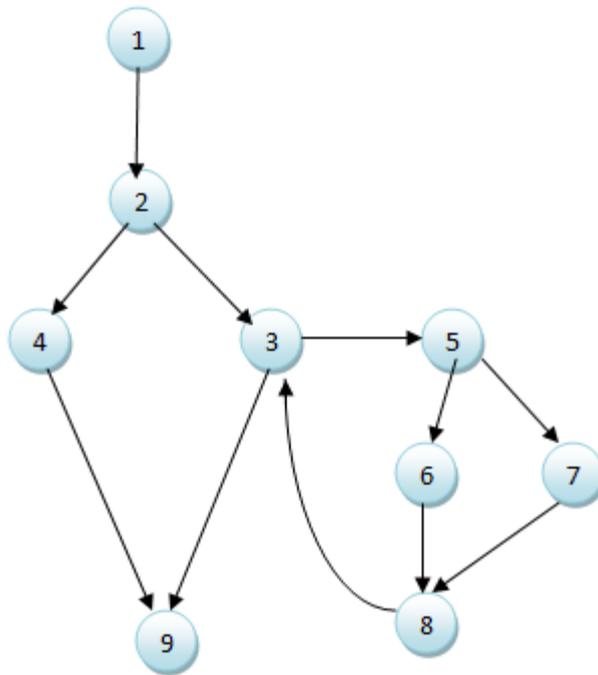


Figura 13. Grafo del método modificaractivo.

3. Se calcula la complejidad ciclomática del grafo.

$$\text{Complejidad} = V(G) = A - N + 2$$

$$V(G) = 10 - 9 + 2 = 3$$

Comprobando resultado por otra vía.

$$V(G) = P + 1$$

$$V(G) = 2 + 1 = 3$$

4. Se determinan un conjunto básico de caminos independientes (en este caso tres).

Camino #1: 1-2-4-9

Camino#2:1-2-3-5-6-8-3-9

Camino#2:1-2-3-5-7-8-3-9

5. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Caso de prueba para el camino básico # 1:

Descripción: los datos de entrada cumplirán con los siguientes requisitos. El valor de “dinamica” es un número entero entre 0 y 1.

Condición de ejecución: el valor de “dinamica” será igual a 0.

Entrada: \$datos->dinamica = 0

Resultados esperados: se espera que se actualicen los valores de descripción, valor nuevo y valor viejo.

Se modifica satisfactoriamente los valores del AFT.

Caso de prueba para el camino básico # 2:

Descripción: los datos de entrada cumplirán con los siguientes requisitos. El valor de “dinamica” es un número entero entre 0 y 1.

Condición de ejecución: el valor de “dinamica” será igual a 1 y el atributo a modificar del AFT debe poseer un valor.

Entrada: \$datos->dinamica = 1, \$idactivofijotangible = 9200, \$atributo->idatributo = 101

Resultados esperados: se espera que se actualicen los valores de descripción, valor nuevo y valor viejo. Valor viejo debe tomar el valor actual del atributo y valor nuevo el nuevo valor a modificar.

Se modifica satisfactoriamente los valores del AFT.

Caso de prueba para el camino básico # 3:

Descripción: los datos de entrada cumplirán con los siguientes requisitos. El valor de “dinamica” es un número entero entre 0 y 1.

Condición de ejecución: el valor de “dinamica” será igual a 1 y el atributo a modificar del AFT no debe poseer un valor.

Entrada: \$datos->dinamica = 1, \$idactivofijotangible = 9350, \$atributo->idatributo = 105

Resultados esperados: se espera que se actualicen los valores de descripción, valor nuevo y valor viejo. Valor viejo debe tomar el valor de cadena vacía (“”) valor nuevo el valor a modificar.

Se modifican satisfactoriamente los valores del AFT.

Luego de aplicar los distintos casos de pruebas, se pudo comprobar que el flujo de trabajo de la función está correcto ya que cumple con las condiciones necesarias que se habían planteado.

3.8.2 Prueba de caja negra.

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa.

Los casos de prueba de la caja negra procura señalar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.

Se derivan conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales del programa.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Aplicación de pruebas de caja negra.

Para la aplicación de este tipo de prueba se usará el requisito Modificar de propiedad. El objetivo de este requisito son las modificaciones de los atributos dinámicos de los grupos que heredan los AFT que corresponden a él.

Requisito a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1:Modificar propiedad.	Se modifica la propiedad del AFT (son los atributos dinámicos de los grupos que heredan los AFT que corresponden a él).	EP 1.1: Modificar la propiedad introduciendo datos válidos.	<ul style="list-style-type: none"> – Se debe presionar el botón Modificar. – Se debe introducir los datos de la propiedad a modificar. – Se presiona el botón Aceptar. – Se muestra un mensaje de información: “La propiedad ha sido modificada satisfactoriamente.”.

	<p>EP 1.2: Modificar una propiedad introduciendo datos inválidos.</p>	<ul style="list-style-type: none"> - Se debe presionar el botón Modificar. - Se introducen datos inválidos. - Se presiona el botón Aceptar. - Se muestra un mensaje de información: "Los datos introducidos no son correctos, intente nuevamente."
	<p>EP 1.3: Modificar la propiedad dejando campos requeridos en blanco.</p>	<ul style="list-style-type: none"> - Se debe presionar el botón Modificar. - Se introducen los datos de la propiedad dejando algún campo requerido en blanco. - Se presiona el botón Aceptar. - El sistema muestra una ventana de información: "Ha dejado datos requeridos en blanco, intente nuevamente."
	<p>EP 1.4: Cancelar.</p>	<ul style="list-style-type: none"> - Se modifican o no los datos de la propiedad. - Se presiona el botón Cancelar.

Descripción de variable

No	Nombre de campo	Tipo	Válido	Inválido
1	Propiedad	Nomenclado	N/A	N/A

Juegos de datos a probar

Id del escenario	Escenario	Propiedad	Respuesta del sistema	Resultado de la prueba
EP 1.1	Modificar la propiedad introduciendo datos válidos.	N/A	Se muestra un mensaje de información: "La propiedad ha sido modificada satisfactoriamente."	
EP 1.2	Modificar una propiedad introduciendo datos inválidos.	N/A	Se muestra un mensaje de información: "Los datos introducidos no son correctos, intente nuevamente."	

EP 1.3	Modificar la propiedad dejando campos requeridos en blanco.	N/A	El sistema muestra una ventana de información: "Ha dejado datos requeridos en blanco, intente nuevamente."
EP 1.4	Cancelar.	N/A	Se cancela la operación y se cierra la ventana.

Conclusiones parciales.

Con el desarrollo de este capítulo se logró diseñar e implementar las interfaces de usuario, implementar las clases necesarias para desarrollo de la solución propuesta, implementar los servicios brindados, realizar el diagrama de componentes para comprender la interacción entre estos y conocer las interfaces que estos soportan, crear el modelo de despliegue para visualizar la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo , realizar pruebas de caja blanca y caja negra para la validación de la solución propuesta.

Conclusiones.

El presente trabajo demostró la vital importancia para el país una herramienta para el proceso de modificación de los AFT en las entidades y empresas presupuestadas.

Se detallaron las herramientas elegidas por el proyecto para la implementación de los componentes. Para identificar e implementar las necesidades de integración con otros componentes dentro del subsistema Logística, se realizó un estudio de aquellos que podían ser reutilizados, lo que viabilizó acceder a funcionalidades que hubieran sido necesarias implementar, lo que hubiese provocado un esfuerzo mucho mayor para lograr la solución.

El desarrollo de este componente constituye un aporte muy importante, debido a que es una novedad con respecto a la informatización de procesos en nuestro país, que marca un proceso de avance en el desarrollo del software interoperable dentro de la entidad, que incrementará la eficiencia del proceso de modificación de los AFT. Al finalizar el presente trabajo de diploma se dan por cumplidos los objetivos planteados en sus inicios pues se ha logrado una realización eficiente de los procesos involucrados, obteniendo la implementación de una herramienta en el que se aplican los resultados de la investigación llevada a cabo.

Recomendaciones.

Al concluir el presente trabajo de diploma y al cumplir los objetivos trazados en el mismo, se recomienda:

- Realizar el despliegue de la aplicación en varias entidades para comprobar que cumple las expectativas del cliente.
- Seguir profundizando en el tema abordado para ampliar los conocimientos con vistas a detectar nuevos procesos a informatizar.

Bibliografía.

1. *Assets Sistema de Gestión Integral*. (2910). Obtenido de <http://assets.co.cu/fijos.asp>.
2. Bradenbaugh, J. *Aplicaciones JavaScript*. Anaya Multimedia.
3. Castro, E. *HTML CON XHTML y CSS*. Anaya Multimedia.
4. *EXT Js*. (s.f.). Recuperado el 20 de Enero de 2010, de <http://www.extjs.com>
5. IBM Corporation. (2003). *IBM Rational Unified Process*. New York.
6. Ivar Jacobson, G. B. (200). *El Lenguaje Unificado de Modelado*. Madrid.
7. Jeffrey, Z. *Diseño con estandares web*. Anaya Multimedia.
8. Kabir, M. J. *La Biblia Servidor Apache 2*. Anaya.
9. Kiccillof, C. R.–N. (Marzo de 2004). Estilos y Patrones en la Estrategia de. UNIVERSIDAD DE BUENOS AIRES.
10. Larman, C. *UML y Patrones* (Segunda ed.). Prentice Hall.
11. Maldonado, R. (2006). *Estudio de la Contabilidad General*. Felix Varela.
12. MÖHRKE, C. Exploración del Entorno de Desarrollo Zend Studio 4.
13. *Navegador web Firefox*. (s.f.). Recuperado el 15 de 1 de 2010, de <http://www.mozilla-europe.org>
14. *NetCraft*. (s.f.). Recuperado el 15 de 12 de 2009, de <http://www.netcraft.co.uk/Survey/>
15. Pérez, J. E. (2008). *Introducción a AJAX*.
16. Pressman, R. (2005). *Ingeniería del Software. Un enfoque práctico*.
17. *Rodas XXI*. (s.f.). Obtenido de <http://www.rodasxxi.cu/>
18. Sobotka, J. F. *Superutilidades para JavaScript*. McGRAW-HIL.
19. Valle, J. M. *Diseño de Páginas Web*. McGRAW-HIL.
20. *Zend Technologies*. (2009). Obtenido de <http://framework.zend.com/>

Glosario de Términos.

1. **Entidad:** empresa, unidad presupuestada u otro tipo de organización similar con una gestión económica, financiera, organizativa, técnica, productiva, comercial, laboral y contractual, con autonomía controlada, en cumplimiento de lo establecido por el Gobierno.
2. **Subsistema:** es un sistema que se ejecuta sobre un sistema operativo.
3. **Proceso:** (del latín processus) es un conjunto de actividades o eventos (coordinados u organizados) que se realizan o suceden (alternativa o simultáneamente) con un fin determinado.
4. **Módulo:** software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.
5. **Multimoneda:** puede trabajar con todas las monedas que el cliente desee y realiza la conversión y comparación a una moneda base.
6. **Unidades presupuestadas:** entidades que administran los recursos que el presupuesto del Estado asigna para cubrir los gastos sociales, como la educación, la salud, los servicios comunales, la defensa y el orden interior y otros. Las unidades presupuestadas pueden estar subordinadas a los órganos locales del Poder Popular (municipales o provinciales) o a los organismos de la administración central del estado.
7. **Metodología de desarrollo de software:** en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.
8. **Requisitos:** circunstancia o condición necesaria para algo, necesidad o expectativa establecida, generalmente implícita u obligatoria.
9. **Estándares web:** es un término muy general utilizado para referirse a estándares y otras especificaciones técnicas que definen y describen aspectos de la World Wide Web. En años recientes, el término ha sido frecuentemente asociado con la tendencia de aprobar un conjunto de mejores prácticas estandarizadas para construir sitios web y a la filosofía del diseño y desarrollo web que incluye esos métodos.
10. **Componente:** es una clase de uso específico, lista para usar, que puede ser configurada o utilizada de forma visual, desde el entorno de desarrollo.

11. **Patrones de diseño:** (design patterns) son 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. Un patrón de diseño es 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. Lo otro, es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.
12. **Herramientas CASE:** (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.
13. **Artefacto:** en conexión con el desarrollo de software, está mayormente asociado a métodos o procesos de desarrollo específicos. El uso del término pudo haberse originado en estos métodos.
14. **Algoritmo:** Conjunto de procedimientos mediante los que se consigue un efecto. Suelen expresarse a través de letras, cifras y símbolos, que forman un algoritmo determinado. Dícese del procedimiento para resolver problemas en términos de las acciones a ejecutar o el orden en que se ejecutarán dichas acciones en un problema dado. Conjunto de Instrucciones que especifican la secuencia de operaciones a realizar, en orden, para resolver un sistema específico o clase de problema.
15. **Técnicas de prototipado:** la aplicación de la técnica de prototipado es fundamental en el desarrollo e implementación de los métodos para la inspección y test de un producto, dado que, habitualmente, no será el producto final lo que se someta a los diversos experimentos, sino un prototipo del mismo con unas determinadas características, en virtud de las cuales se enfocarán los métodos en una dirección concreta. En definitiva, se trata de una cuestión indispensable, presente incluso en ciertos métodos contextuales.
16. **Línea Base:** es un concepto de gestión de la configuración del software que lleva a controlar los cambios sin impedir seriamente los cambios justificados. El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) 610.12/1990, define una línea base como: una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en

adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios.

17. **Abstracción:** consiste en aislar un elemento de su contexto o del resto de los elementos que lo acompañan. En programación, el término se refiere al énfasis en el "¿qué hace?" más que en el "¿cómo lo hace?" (característica de caja negra). El común denominador en la evolución de los lenguajes de programación, desde los clásicos o imperativos hasta los orientados a objetos, ha sido el nivel de abstracción del que cada uno de ellos hace uso.
18. **Antagónica:** manifiesta o implica antagonismo.
Antagonismo: situación en la cual, el efecto combinado de dos o más factores es menor que el efecto aislado de uno de ellos por la interferencia que ejerce. Se usa para señalar la acción conjunta de dos materias perjudiciales, cuando dicha acción es menor que la producida por separado.
19. **Editor de SQL:** es un programa que permite crear y modificar archivos digitales compuestos únicamente por sentencias SQL.
20. **Encriptar:** es la acción de usar la criptografía (acrónimo de escritura oculta). Es el arte o ciencia de cifrar y descifrar información mediante técnicas especiales y se emplea frecuentemente para permitir un intercambio de mensajes que sólo puedan ser leídos por personas a las que van dirigidos y que poseen los medios para descifrarlos.
21. **Sistema de Control de Revisiones o versiones (*Revision Control System - RCS*):** es una implementación de software para el control de versiones que automatiza las tareas de guardar, recuperar, registrar, identificar y mezclar versiones de archivos. RCS es útil para archivos que son modificados frecuentemente, por ejemplo programas informáticos, documentación, gráficos de procedimientos, monografías y cartas.
22. **Fichero:** es un conjunto de bits almacenado en un dispositivo periférico.
23. **Directorio:** es una agrupación de archivos de datos, atendiendo a su contenido, a su propósito o a cualquier criterio que decida el usuario. Técnicamente el directorio almacena información acerca de los archivos que contiene los atributos de los archivos o dónde se encuentran físicamente en el dispositivo de almacenamiento.
24. **Repositorio:** es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.
25. **Código fuente** (de un programa informático o software): es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está descrito por completo su funcionamiento.

26. **Gestión de base de datos relacional orientada a objetos:**

Base de datos orientada a objetos: la información se representa mediante objetos como los presentes en la programación orientada a objetos. Cuando se integra las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos (ODBMS, object database management system).

Base de datos relacional: es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene su nombre: "Modelo Relacional".

27. **Clave foránea:** en el contexto de bases de datos relacionales, una clave foránea (o Foreign Key FK) es una limitación referencial entre dos tablas. La clave foránea identifica una columna o grupo de columnas en una tabla (tabla hija o referendo) que se refiere a una columna o grupo de columnas en otra tabla (tabla maestra o referenciada). Las columnas en la tabla referendo deben ser la clave primaria u otra clave candidata en la tabla referenciada.
28. **Query:** cadena de consulta, este término generalmente se utiliza para hacer referencia a una interacción con una base de datos. Es la parte de una URL que contiene los datos que deben pasar a aplicaciones web.
29. **DBMS** (sistemas de gestión de bases de datos) o SGBD (en inglés database management system, abreviado DBMS): son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.
30. **GUI** (del inglés *graphical user interface*): interfaz gráfica de usuario, es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.
31. **Debugger** (en español depurador): es un programa que permite depurar o limpiar los errores de otro programa informático.
32. **AJAX**, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML): es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

33. **Plataforma:** en informática, una plataforma es precisamente el principio, en el cual se constituye un hardware, sobre el cual un software puede ejecutarse/desarrollarse. No debe confundirse esto con arquitecturas.
34. **HTTP:** protocolo de transferencia de hipertexto (**HTTP**, *HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web.
35. **CGI:** interfaz de entrada común (en inglés **Common Gateway Interface**, abreviado **CGI**) es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa.
36. **Servidor proxy:** en el contexto de las redes informáticas, el término proxy hace referencia a un programa o dispositivo que realiza una acción en representación de otro. Su finalidad más habitual es la de servidor proxy, que sirve para permitir el acceso a Internet a todos los equipos de una organización, cuando sólo se puede disponer de un único equipo conectado, esto es, una única dirección IP (protocolo de internet).
37. **Renderizado** (*Render* en inglés): es un término usado en jerga informática para referirse al proceso de generar una imagen desde un modelo. Este término técnico es utilizado por los animadores o productores audiovisuales.
38. **EJB (Enterprise JavaBeans):** son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE 5.0) de Oracle Corporation (inicialmente desarrollado por Sun Microsystems).
39. **IDE:** entorno de desarrollo integrado o IDE (acrónimo en inglés de **integrated development environment**), es un programa informático compuesto por un conjunto de herramientas de programación.
40. **Widget:** es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o **Widget Engine**. Entre sus objetivos están los de dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.
41. **Standard Generalized Markup Language:** (**SGML**) o "Lenguaje de Mercado Generalizado". Consiste en un sistema para la organización y etiquetado de documentos. La Organización Internacional de Estándares (ISO) normalizó este lenguaje en 1986. El lenguaje SGML sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial. El lenguaje HTML está definido en términos del SGML. XML es un estándar de creación posterior, que incorpora un subconjunto de la funcionalidad del SGML

(suficiente para las necesidades comunes), y resulta más sencillo de implementar pues evita algunas características avanzadas de SGML.

42. **Windows Shell:** es el aspecto más visible de la línea de Microsoft Windows de los sistemas operativos. Es el contenedor dentro de la que toda la interfaz de usuario se presenta, incluyendo la barra de tareas, el escritorio, el explorador de Windows, así como muchos de los cuadros de diálogo y controles de interfaz, pero también se describe el pasado de conchas, como MS-DOS ejecutivo y gerente de programa.
43. **Movimientos:** son acciones que se realizan sobre los AFT como por ejemplo: altas, bajas y modificaciones de los AFT. En el caso de las modificaciones, movimientos es la asociación entre el documento de tipo modificación ya sea de propiedad o de valor y la modificación del AFT.