



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 3**

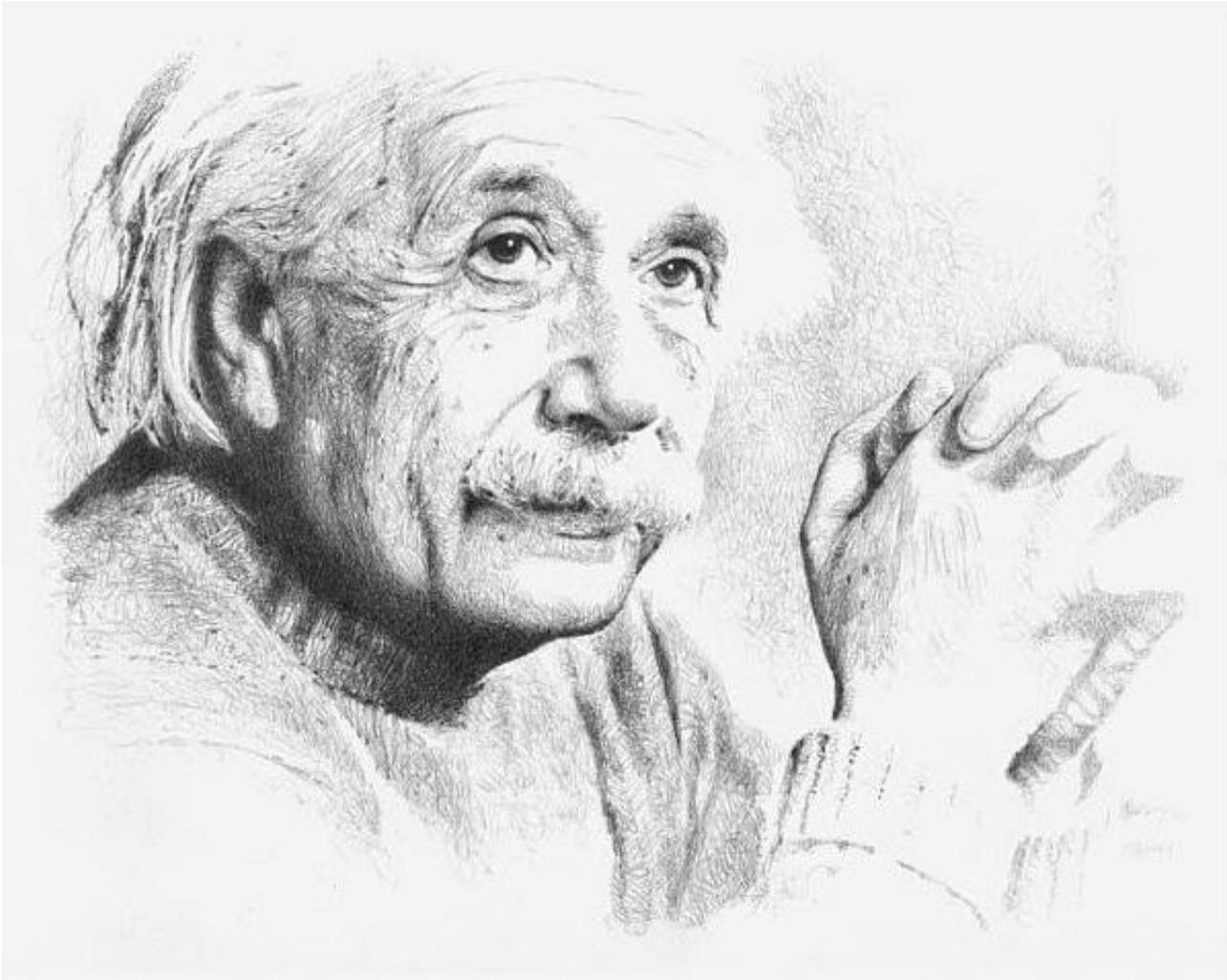
**Título: Desarrollo del componente Depreciación  
del subsistema Activos Fijos Tangibles de Cedrux.**

**Autor:** Ernesto Fraga Quintero.

**Tutor(es):** Ing. Judiht García Rodríguez.  
Ing. Yuriel Martínez Ordaz.

**Ciudad de La Habana, 2013  
“Año 54 de la Revolución.”**

## Pensamiento



**Todos somos muy ignorantes.  
Lo que ocurre es que no todos ignoramos las mismas  
cosas.**

**Albert Einstein**

## DECLARACIÓN DE AUTORÍA

Por este medio declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

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

Ernesto Fraga Quintero

\_\_\_\_\_  
Firma del Autor

Ing. Judiht García Rodríguez

\_\_\_\_\_  
Firma del Tutor

Ing. Yuriel Martínez Ordaz

\_\_\_\_\_  
Firma del Tutor

## **Agradecimientos**

Agradezco de corazón a todos los amigos de alguna forma contribuyeron al resultado que he obtenido, particularmente a mis tutores que con su experiencia colaboraron en la realización de este trabajo. Muchas gracias a mi pukita, a toda mi familia y en especial a mi mamá y mi papá por apoyarme en todo y tener confianza en mí.

Les agradezco a todas aquellas personas que alguna vez preguntaron: ¿Cómo va la tesis? ¿Por fin cuándo expones? A todos, gracias muchas...

## **Dedicatoria**

Este trabajo se lo dedico a mis padres, que con sus consejos y su apoyo me alumbraron el camino a seguir y me dieron la posibilidad de estar hoy donde estoy.

## **Resumen**

El presente trabajo abarca el ciclo completo de desarrollo del componente de depreciación de activos fijos tangibles para el Sistema Integral de Gestión CedruX. Se exponen las razones por las cuales se precisa de un sistema automatizado con la funcionalidad de depreciar los activos fijos tangibles. Con el desarrollo de este componente se espera agilizar y facilitar el proceso de depreciación de activos fijos tangibles. Se realiza un análisis comparativo de las tendencias tecnológicas a utilizar. Se describe la solución propuesta incluyendo temas como el análisis y diseño del componente que se desarrolla. Se implementan las funcionalidades que dan respuesta a los requisitos y se realizan un grupo de pruebas para validar la solución propuesta.

**Palabras Clave:** Activo Fijo Tangible, componente, depreciación, economía, empresa.

## Índice

|  |    |
|--|----|
| Introducción .....   | 10 |
| CAPITULO #1: Fundamentación teórica .....  | 14 |
| 1.1 Introducción .....   | 14 |
| 1.2 Activos Fijos Tangibles .....  | 14 |
| 1.3 Depreciación.....  | 15 |
| 1.4 Sistemas de Planificación de Recursos Empresariales .....                                      | 16 |
| 1.4.1 Sistemas utilizados en Cuba para la planificación de recursos en las<br>entidades .....      | 16 |
| 1.4.2 Sistemas Internacionales para la planificación de recursos en las<br>entidades cubanas ..... | 20 |
| 1.5 Modelo de desarrollo de software .....   | 24 |
| 1.6 Arquitectura adoptada en el componente .....   | 25 |
| 1.7 Lenguajes, herramientas y tecnologías a utilizar para el desarrollo del<br>componente .....    | 26 |
| 1.7.1 Lenguajes de programación del lado del servidor .....  | 26 |
| 1.7.2 Lenguajes de programación del lado del cliente .....   | 27 |
| 1.7.3 Estudio de los lenguajes de modelado utilizados .....  | 28 |
| 1.7.4 Herramientas para el desarrollo de la solución informática .....                             | 29 |
| 1.7.5 Tecnologías utilizadas en la solución del problema .....                                     | 33 |
| 1.8 Conclusiones del Capitulo #1 .....   | 37 |
| CAPITULO #2: Descripción de la solución propuesta .....  | 38 |
| 2.1 Introducción .....   | 38 |
| 2.2 Reglas del negocio .....   | 38 |
| 2.3 Modelado de procesos de negocio .....  | 38 |
| 2.3.1 Descripción del proceso de negocio Depreciación de AFT .....                                 | 39 |
| 2.3.2 Diagrama del proceso de negocio Depreciación de AFT .....                                    | 42 |
| 2.4 Modelo conceptual Depreciación de AFT .....  | 42 |
| 2.4.1 Definición de los conceptos fundamentales .....  | 43 |

|  |           |
|--|-----------|
| 2.5.1 Requisitos funcionales .....   | 44        |
| 2.5.2 Requisitos no funcionales .....                                      | 45        |
| 2.6 Patrones utilizados .....  | 47        |
| 2.7 Diagrama de clases del diseño .....                                    | 49        |
| 2.8 Diagrama de modelo de datos .....                                      | 51        |
| 2.9 Conclusiones de este capítulo .....                                    | 51        |
| <b>CAPITULO #3: Implementación y prueba del componente propuesto .....</b> | <b>52</b> |
| 3.1 Introducción .....   | 52        |
| 3.2 Diagrama de relación entre componentes .....                           | 52        |
| 3.3 Matriz de relación entre componentes .....                             | 53        |
| 3.4 Estándares de codificación .....                                       | 53        |
| 3.5 Pruebas internas .....   | 56        |
| 3.5.1 Pruebas de Caja Negra .....  | 56        |
| 3.5.2 Pruebas de Caja Blanca .....   | 59        |
| 3.6 Validación de las variables declaradas en el problema a resolver ..... | 64        |
| 3.7 Conclusiones de este capítulo .....                                    | 66        |
| Conclusiones generales .....   | 68        |
| Recomendaciones .....  | 69        |
| Bibliografía .....   | 70        |

## Índice de Tablas y Figuras

|   |    |
|---|----|
| Tabla 1. Métodos de depreciación de SISCON 5. ....  | 18 |
| Figura 1. Diagrama de arquitectura de paquetes.....   | 26 |
| Figura 2. Diagrama del proceso de negocio Depreciación de AFT. ....                             | 42 |
| Figura 3. Diagrama del Modelo Conceptual de AFT. ....   | 43 |
| Tabla 2. Especificación de requisitos funcionales. ....   | 45 |
| Figura 4. Diagrama de clases del diseño. ....   | 49 |
| Figura 5. Modelo de datos. ....   | 51 |
| Figura 6. Diagrama de Relación entre componentes. ....  | 52 |
| Tabla 3. Relación del componente depreciación con componentes internos. ....                    | 53 |
| Figura 7. Ejemplo de Identación de código.....  | 54 |
| Figura 8. Ejemplo de cabecera de archivo.....   | 54 |
| Figura 9. Ejemplo de comentario de descripción de funciones. ....                               | 54 |
| Figura 10. Ejemplo de nomenclatura de clases. ....  | 55 |
| Figura 11. Ejemplo de nomenclatura según el tipo de clases. ....                                | 55 |
| Figura 12. Ejemplo de nomenclatura de función.....  | 55 |
| Figura 13. Ejemplo de nomenclatura de variable.....   | 56 |
| Tabla 4: Requisito a probar “Cambiar el Método de Depreciación a un AFT”. ...                   | 58 |
| Tabla 5: Descripción de variable Cambiar el Método de Depreciación a un AFT.<br>.....           | 58 |
| Tabla 6: Juego de datos para el requisito “Cambiar el Método de Depreciación a<br>un AFT”. .... | 59 |
| Figura 14. Método aplicarLaDepreciacion.....  | 61 |
| Figura 15. Grafo del método aplicarLaDepreciacion. ....   | 61 |
| Tabla 7. Caminos Básicos. ....  | 62 |
| Tabla 8. Validación de las variables declaradas en el problema a resolver. ....                 | 66 |

## **Introducción**

La revolución de las tecnologías de la información y las comunicaciones (TIC) ha traído consigo grandes cambios en todas las esferas de la gestión empresarial, Cuba como el resto del mundo muestra gran interés en la evolución de la ciencia, la tecnología y el desarrollo de software. Los sistemas de planificación de recursos empresariales (ERP) como parte importante de estos cambios permiten incorporar de forma creciente las nuevas tecnologías, las cuales se han convertido en un instrumento clave para una gestión eficaz y eficiente, básicamente, por su capacidad de proceso de grandes cantidades de información en muy poco tiempo y con unos costes razonablemente bajos [**Guillermo J. Sierra, Bernabé Escobar Pérez**].

Empresas como Microsoft, Oracle y Google han demostrado lo rentable que puede llegar a ser la industria del software [**Lliulla, Carmen. 2012**] y en aras de explotar este campo, el país ha materializado novedosas ideas e importantes proyectos, un ejemplo de ello es la preparación de profesionales competentes, para esto se han creado instituciones de alto nivel como la Universidad de las Ciencias Informáticas (UCI), que tiene entre sus principales objetivos desarrollar aplicaciones para empresas cubanas y extranjeras. Entre las soluciones informáticas que se desarrollan existe un Sistema Integral de Gestión llamado Cedrux, pensado y ajustado a las necesidades de las entidades cubanas presupuestadas y empresariales, el cual defiende el concepto de independencia tecnológica y pertenece al proyecto ERP.

Cedrux centraliza la información de la contabilidad de la empresa lo que posibilita una mejor administración, gestión, agilización de procesos, optimización del tiempo y mayor control sobre los recursos de una empresa. EL sistema cuenta con varios subsistemas, entre estos se encuentra Activos Fijos Tangibles (AFT) que está conformado por un gran conjunto de funcionalidades que brindan la posibilidad de acceder y controlar los AFT de una entidad, a continuación se mencionan algunas: Configuración de la Entidad, Nomencladores de (Activos Fijos Tangibles y Activos Fijos Tangibles por Entidad), Gestionar (Movimientos y Cierre). Asociados de manera interactiva con esos procesos se manejan temas como la depreciación que actualmente están inconclusos [**autores, Varios. 2010**].

La depreciación influye en la estructura financiera de la empresa, por ejemplo, en

el balance general, puesto que al desgastar un activo, este disminuye su valor hasta desaparecer, por lo que la empresa tendrá que proceder a reponer ese activo depreciado por su desgaste en uso. La existencia en las empresas de soluciones tecnológicas que gestionen la depreciación, permite optimizar la capacidad operativa, contribuye a preservar el patrimonio, evita perder capital por una mala distribución de los recursos, ayuda en la toma de decisiones y de manera general propicia tener una visión más acertada del estado financiero y económico [Tineo, Claudia Giovanna Huaytalla. 2009].

En este momento en las empresas cubanas existen soluciones de software para la gestión de los activos fijos, pero ninguno resulta idóneo pues muchos de ellos son propietarios y para utilizarlos hay que pagar una licencia, esto además de ser un gasto adicional e innecesario, va en contra de las políticas de independencia tecnológica que mantiene el país. Por otra parte algunos sistemas por ser desarrollados inicialmente para economías capitalistas no se acomodan completamente a las empresas cubanas.

Un software que maneje la contabilidad y prescinda de un componente para gestionar la depreciación de los AFT obliga a la empresa a procesar de forma manual grandes cantidades de información lo que cual es muy engorroso, lento y posee el riesgo de perder la información involuntariamente. En los casos de las entidades que no analizan la depreciación o lo hacen de forma errónea se producen alteraciones en cuentas como las de gastos, se dificulta una visión integral de la empresa y se entorpece el proceso de previsión y planificación en el tiempo. La ausencia de esta funcionalidad también impide conocer con certeza el valor real que posee una institución en cuanto a AFT en cualquier momento en que se realice el ejercicio.

La descripción anterior conduce a formular el siguiente **problema de investigación**: ¿Cómo agilizar y facilitar el proceso de depreciación de AFT en las entidades cubanas que utilicen Cedrux para la gestión económica?, donde el **objeto de estudio** es el proceso de depreciación de los Activos Fijos Tangibles y el **campo de acción** es el Proceso de Depreciación de Activos Fijos Tangibles del sistema Cedrux. Para ello se plantea la siguiente **idea a defender**: El desarrollo del Componente de Depreciación del subsistema Activos Fijos Tangibles del sistema Cedrux brindará mayor agilidad y facilidad en el proceso de

depreciación de AFT.

El **Objetivo general** del trabajo es: Desarrollar un componente para agilizar y facilitar el proceso de depreciación de AFT en las entidades cubanas que utilicen Cedrux para la gestión económica.

Los **Objetivos específicos** son:

- Elaborar el marco teórico-referencial de la investigación a partir del estudio de las particularidades de los módulos de depreciación existentes.
- Realizar el modelado del negocio con el fin de sentar las bases para un adecuado diseño de la solución.
- Implementar las funcionalidades necesarias siguiendo buenas prácticas del desarrollo de software para dotar al sistema de Gestión de Activos Fijos de Cedrux con las capacidades para configurar y gestionar la depreciación de activos fijos tangibles.
- Validar las funcionalidades del componente realizado a partir de pruebas internas.

Con el objetivo de proveer las bases necesarias de la investigación se hizo necesaria la utilización de diversos **métodos de investigación**. Estos métodos guardan relación entre ellos y se dividen en dos grupos, los teóricos y los empíricos. Los **métodos teóricos** permiten descubrir en el objeto de investigación las relaciones esenciales y las cualidades fundamentales. Por ello se apoya básicamente en los procesos de abstracción, análisis, síntesis, inducción y deducción [Johanniitha. 2013]. En el desarrollo de esta solución informática se utilizan los siguientes métodos teóricos:

- Histórico-Lógico: Con el uso de este método se avanzó ordenadamente en el estudio de la metodología, herramientas de desarrollo, patrones de diseño y arquitectónicos. También contribuyó al estudio de otros sistemas ERP.
- Modelación: Con este método se logra un mayor entendimiento del problema, a través de artefactos como los diagramas de Procesos del Negocio, de Clases y de Modelo de Datos.
- Analítico-Sintético: Este método permitió a través de un exhaustivo análisis llegar a conclusiones en momentos claves como la descripción de procesos.

La investigación empírica permite al investigador hacer una serie de estudios referente a su problemática, retomando experiencia de otros autores, para de ahí partir con su exploración, también conlleva efectuar el análisis preliminar de la información, así como verificar y comprobar las concepciones teóricas [**autores, grupo de**].

El **método empírico** que se utiliza es:

- **Medición:** Este método se evidenció en la aplicación de métricas de calidad de software, para validar la solución.

La técnica de recopilación de información utilizada es la entrevista. La realización de entrevistas a futuros usuarios ayudó a determinar los requisitos que necesita cumplir la solución informática.

Esta investigación se desglosa en tres capítulos, los cuales de forma simplificada abordan los siguientes contenidos:

**Capítulo #1, Fundamentación teórica:** En este primer capítulo se hace un acercamiento en el tema de la depreciación y la situación en el país de la automatización de este proceso. Se llega a la conclusión sobre la necesidad de desarrollar un componente informático que permita gestionar la depreciación de AFT. Se realiza una propuesta de las herramientas y tecnologías que se utilizan para el desarrollo de la solución.

**Capítulo #2, Descripción de la solución propuesta:** Este capítulo se enmarca principalmente en la descripción de la solución que se propone, detallando los requisitos funcionales y no funcionales, el modelo de dominio para comprender el ámbito donde se desarrolla este componente. Además se modela el componente a través de varios tipos de diagramas, como los de clases del diseño, los diagramas de modelo de datos.

**Capítulo #3, Implementación y prueba del componente propuesto:** Este capítulo abarca todo lo relacionado con el proceso de desarrollo del componente haciendo uso de las herramientas y tecnologías propuestas en el capítulo 1. Además se describen los casos de pruebas y se evalúan los resultados luego de aplicar pruebas internas.

## CAPITULO #1: Fundamentación teórica

### 1.1 Introducción

En consecuencia con la importancia que posee la existencia del componente de depreciación se hace necesario un profundo análisis de los sistemas de gestión empresarial existentes y decidir si dichos sistemas satisfacen las necesidades de las entidades cubanas y si pueden servir de guía en el desarrollo de esta investigación. En el presente capítulo se relacionan los conceptos necesarios afines con la temática tratada, marco en el cual se va a desarrollar el componente de depreciación. Para una mejor comprensión del trabajo, se encuentra anexado a al documento el glosario de términos (ver anexo 1). Se presentarán además los lenguajes, herramientas y tecnologías propuestos para desarrollar la solución informática que ofrezca respuesta al problema planteado inicialmente.

### 1.2 Activos Fijos Tangibles

Representan propiedades físicamente tangibles que han de utilizarse por un período largo en las operaciones regulares de la entidad y que normalmente no se destinan a la venta. Estos activos, con excepción del terreno y los animales productivos de ganado mayor, trasladan su valor paulatinamente, durante su vida útil a la producción de bienes y a la prestación de servicios. En el caso de los animales productivos su valor se traslada de una sola vez al final de su vida útil productiva [Entorno Virtual de Aprendizaje. 2007].

Otra definición de activo fijo tangible es:

Son aquellos bienes tangibles que adquiere una empresa para hacer uso constante de ellos: en la producción comercialización o administración. Son activos que contribuyen en la explotación económica de la empresa, ejemplo de estos bienes son: terrenos, edificaciones, vehículos, maquinarias y muebles. Los activos fijos son bienes que pertenecen a una empresa y se utilizan para el servicio de dicha empresa, no se compran para venderlos, sino para sacarles un rendimiento, estos tienen un precio inicial (precio de compra), un precio final (precio de recuperación), una vida útil en la que son explotados y por la que va perdiendo su valor, correspondiendo el valor máximo con el momento de compra y el valor mínimo con el momento de venta o de jubilación del activo. En última instancia, el empresario

puede deshacerse de los activos fijos vendiéndolos a otra compañía de reciclaje o de aprovechamiento de piezas sueltas [Caro, Ricardo Enriquez. 2012] [Depreciacion.net].

Entre los mayores problemas detectados en las empresas cubanas concernientes a la gestión de los AFT se encuentran:

- No existencia de un modelo estándar de movimientos de los activos, por tanto no es posible agrupar los datos obligatorios.
- Se continúan depreciando los AFT, cuando estos están totalmente depreciados.
- No se desecha el número de inventario cuando se le da baja a algún activo provocando una duplicidad del número de inventario en los activos.
- El control de los AFT en las entidades no automatizadas se lleva de manera manual, provocando que este proceso resulte lento.
- Poco o ningún control de la información procesada manualmente.
- Realización de operaciones sobre los activos sin dejar trazabilidad de los mismos, provocando que el contador no tenga conocimiento del origen o destino de los activos que controla.
- La búsqueda de información relativa a un activo resulta engorrosa debido a los grandes volúmenes de información existentes [Larisa González Alvarez, Tahirí Rivero Álvarez, Osmar Leyet Fernández. 2011].

### **1.3 Depreciación**

La depreciación representa el desgaste que sufren los Activos Fijos Tangibles durante el transcurso de su vida útil. Es uno de los elementos que conforman el costo de un producto o servicio. El método de depreciación aplicado es el que presupone un desgaste de los medios a través del tiempo. Se computa aplicando un porcentaje sobre el valor del Activo Fijo Tangible [Hidalgo, Noidis Barroso. 2012].

Otra definición es:

La depreciación constituye la pérdida progresiva de valor de una máquina, equipo o inmueble por cada año que envejece. Algunos autores señalan la depreciación como un arrendamiento que la empresa se paga así misma por el uso y deterioro de sus instalaciones y equipos [Fernández, Cesar Martínez. 2012].

Actualmente en el país se utiliza el método de Línea Recta para la depreciación de los AFT, pero este método no es el único. El sistema Cedrux debe ser capaz de depreciar los AFT de una empresa por cualquiera de los cuatro principales métodos: Línea Recta, Dígitos Decrecientes, Horas-Máquina y Unidades Producidas. La utilización de varios métodos permite que la depreciación sea más exacta, por ejemplo si el activo es una máquina que produce artículos de tipo (x), lo más factible es que deprecie por el método Unidades Producidas.

#### **1.4 Sistemas de Planificación de Recursos Empresariales**

Los sistemas de Planificación de Recursos Empresariales (*Enterprise Resource Planning*, ERP) son sistemas de gestión de información gerenciales que integran, automatizan y manejan muchas de las acciones asociadas con las operaciones de producción y distribución de una compañía dedicada a la producción de bienes o servicios. Los ERP pueden intervenir en el control de muchas actividades de negocios, tales como ventas, compras, entregas, pagos, producción, contabilidad, administración de inventarios y de recursos humanos. Funcionan en todo tipo de empresas. Para ello, integran todos los departamentos funcionales de la empresa, herramientas de mercadotecnia y administración estratégica en un solo sistema [Gonzalez Brito, Henry Raúl, Lezcano Lozada, Yuniesky. 2005].

Se entiende también por sistemas de planificación de recursos empresariales a sistemas que:

- Administran aspectos de una compañía como producción y distribución.
- Son adaptable, es decir se amoldan a las particularidades de cada institución.
- Dividen sus aspectos en módulos según los diferentes departamentos de la empresa [autores, varios. 2011].

##### **1.4.1 Sistemas utilizados en Cuba para la planificación de recursos en las entidades**

Se debe hacer una adecuada selección del sistema a implantar en la organización teniendo en cuenta factores como la flexibilidad que brinda de forma tal que proporcione un cierto grado de adaptabilidad para con el funcionamiento de la entidad. En general existen muchos sistemas automatizados de contabilidad, que van desde los gratuitos hasta los que cuestan decenas de miles de dólares, en

dependencia de su complejidad y utilidad para los usuarios. A continuación se describen algunos de los sistemas de gestión implantados en el país y en el mundo [Frida Ruso Armada, Lic. Ariamna Rivero Bolaños, Lic. Lisi Campos Chaurero, Lic. Yoenny Leyva Vega].

### **Sistemas Nacionales para la planificación de recursos en las entidades cubanas:**

#### **SISCONT 5:**

El sistema se aviene a las definiciones y conceptos del Ministerio de la Industria Básica aunque por las acciones contables financieras que permite puede ser utilizado en otras entidades nacionales. Está formado por los Módulos:

- Efectivo en Caja y Bancos.
- Inventarios.
- Cobros y Pagos.
- Facturación.
- Activos Fijos Tangibles.
- Nóminas.
- Contabilidad.

Puede ser explotado en régimen mono usuario y multiusuario. Se define para mono entidad y multientidad, en esta última existe el control de su acceso para las entidades en un mismo equipo de cómputo como servidor. El sistema está preparado para controlar el saldo de cada material en dos monedas, a partir de los procedimientos vigentes en el país en cuanto a política monetaria.

Permite el tratamiento de las contabilizaciones de forma transaccional, por resúmenes diarios o de forma personalizada según se definan los parámetros. Este software incluye dentro del módulo de Activos Fijos Tangibles el tratamiento de la depreciación [Ortiz, Wilfredo Rodriguez Rodriguez. 2007].

Según el manual de usuario de SISCON 5 los métodos de depreciación que maneja esta herramienta ERP son los siguientes:

| <b>Método</b>       | <b>Clasificación</b>                       |
|---------------------|--|
| <b>Proporcional</b> | a. Directo o de línea recta o constante    |
|                     | b. Horas máquina o horas de funcionamiento |

|                    |  |
|--------------------|--|
|                    | c. A base de producción o unidades de producción |
| <b>Decreciente</b> | d. A porcentaje fijo o tanto por ciento fijo.    |
|                    | e. En progresión aritmética (decreciente).       |
|                    | f. Americana o de colé: suma de dígitos.         |
|                    | g. Cargo decreciente.                            |
| <b>Creciente</b>   | h. Progresión aritmética                         |
|                    | i. De colé (creciente)                           |
| <b>Interés</b>     | j. Fondo de amortización o de capitalización     |
|                    | k. De interés sobre inversión.                   |
|                    | l. Amortización o anualidades                    |
| <b>Inventarios</b> | m. De inventarios                                |

Tabla 1. Métodos de depreciación de SISCON 5 [Marcelo de la Cruz Heredia, Carlos Palomino Hurtado. 2007].

**Versat-Sarasola 2.0.0:**

Es un producto cubano. Es el primer Sistema Integral de gestión de contabilidad certificado por el Ministerio de Finanzas y Precios, desarrollado para la gestión económica eficaz y fiable. Su primer cliente fue el MINAZ y actualmente es utilizado en Cuba en alrededor de 500 entidades de varias provincias. Con la creación de este sistema se logró un software igual o superior al que se pensaba adquirir y se evitó la importación de un sistema foráneo con lo cual el país se ahorró varios cientos de miles de dólares que costaban las licencias, excluyendo la labor de consultoría, trabajo técnico y atenciones. Este sistema integrado cuenta con un conjunto de 10 módulos entre los que se encuentran:

- Configuración y seguridad.
- Contabilidad general y de gastos.
- Costos y procesos.
- Análisis económico empresarial.
- Control de activos fijos.
- Finanza y caja.
- Planificación y presupuestos.
- Control de inventarios.
- Pago de salario.

- Facturación.

En el módulo de Control de Activos Fijos recoge las operaciones normales que en esta actividad se realizan (altas, bajas, modificaciones de los medios). También se le ha incorporado otras opciones novedosas, dentro de las cuales se encuentran la posibilidad de amortizar los activos por los 4 métodos más conocidos internacionalmente. Posee un asistente (*wizard*) para la configuración del subsistema considerada la tarea más compleja de este módulo [**Marisel Sosa Porteiro, Pedro H. Cobo Morales**].

**Características:**

- Es una aplicación de escritorio.
- Implementado en el lenguaje *Delphi*.
- Trabaja sobre el sistema operativo *Windows* en sus diferentes versiones.
- Soporte para base de datos SQL Server 2000.

**Rodas XXI 3.0:**

Sistema multiempresa y multiusuario creado por CITMATEL para la automatización de la gestión empresarial. Contiene diferentes módulos que pueden usarse integrados o independientes:

- Contabilidad.
- Efectivo caja y banco.
- Nóminas.
- Activos Fijos Tangibles.
- Inventarios.
- Cobros y pagos.
- Facturación.
- Finanzas.
- Tele-cobranza.

Además, cuenta con el módulo Administrador, que brinda mayor integralidad al sistema y garantiza facilidades adicionales durante su instalación y explotación. El módulo de Activos Fijos Tangibles de RODAS XXI le permite realizar ajustes a los activos fijos con facilidad mediante una opción que brinda el sistema, además de la

opción de depreciar de forma automática, 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 [CITMATEL. 2013].

#### **Valoración crítica de las soluciones cubanas:**

El país está librando la batalla por la soberanía tecnológica y las soluciones informáticas SISCONT 5, Versat-Sarasola 2.0.0 y Rodas XXI 3.0 no son ideales ya que corren solo sobre la plataforma de Windows, que como es conocido por todos es un sistema operativo privado. Además, son aplicaciones de escritorio, lo que significa que es necesario tenerlos instalados en cada estación de trabajo trayendo consigo un mayor consumo de recursos del sistema [Marcelo de la Cruz Heredia, Carlos Palomino Hurtado. 2007] [Varios\_autores5] [Varios\_autores3].

#### **1.4.2 Sistemas Internacionales para la planificación de recursos en las entidades cubanas**

##### **Openbravo 3.0:**

El hecho de que la aplicación de Openbravo tenga una interfaz de usuario familiar parecida a las de escritorio conlleva que resulte fácil de aprender y acelera los ratios<sup>1</sup> de adopción en la empresa. Como cualquier aplicación 100% web, ofrece la posibilidad de acceder a Openbravo a través de una red. Lo único que se necesita es un navegador moderno. Permite añadir información o aplicaciones relevantes a cada rol de empleado en un único lugar gracias al espacio de trabajo de Openbravo. Posee facilidades como las de arrastrar y soltar ventanas de otras aplicaciones de utilidad como el email o aplicaciones de terceros. Sus usuarios pueden trabajar de forma eficaz en varias tareas independientes al mismo tiempo gracias al diseño multipestaña.

Openbravo ERP utiliza tecnologías modernas, pero sólidas y suficientemente probadas, para cumplir los requisitos estrictos de rendimiento y escalabilidad de cualquier entorno empresarial:

- Java y Javascript
- SQL y PL/SQL

---

<sup>1</sup> Es la razón o cociente de dos magnitudes relacionadas. Los ratios financieros o contables son los coeficientes que aportan unidades financieras de medida y comparación.

- XML
- HTML

La gestión avanzada de activos le permite a Openbravo de manera rápida y fácil crear nuevos activos desde una factura de compra existente, así como crear y ejecutar planes de depreciación bajo una base lineal y mensual. La lista de movimientos de activos le provee de información detallada y los módulos de documentos de activos especiales, incluidos agilizan el movimiento de activos, las depreciaciones y apreciaciones [**Openbavo**].

### **Exact Globe Next:**

Exact Globe Next le permite ver y analizar la información financiera y logística desde diversos ángulos. Esto se debe a que este ERP es fácil de usar y muy intuitivo, al mismo tiempo que proporciona informes muy completos ya que integra información relativa a todos los procesos de negocio de la organización. Esto permite tener una visión muy detallada de la situación real de la empresa. Brinda alertas automáticas que se pueden monitorear a través de su flujo de trabajo o a través del correo electrónico, esto posibilitará tomar decisiones adecuadas de forma oportuna. La colaboración interdepartamental será más fácil debido a que la información se almacena en una única base de datos central, asegurando que todos los empleados tengan acceso a la misma información actualizada. Mediante el flujo de trabajo, los empleados pueden dividirse las tareas entre sí, además de estar continuamente informados se podrán compartir fácilmente las noticias corporativas a través de la intranet [**Exact**]. Este Software ERP es un sistema integrado de gestión de negocio para toda su organización que permite automatizar sus procesos de negocio en los siguientes ámbitos:

- Gestión Financiera
- Gestión de almacén y de producto
- Gestión de producción
- Gestión de clientes y proveedores
- Gestión de servicios
- Gestión de información
- Gestión de flujos de trabajo
- Gestión de proyectos

- Soluciones móviles [**ERP Globe de Exact**]

Esta solución permite la depreciación de los AFT y lo hace quitándole al valor del AFT, en cada período y de forma uniforme un % previamente definido. El momento de la depreciación es configurable y la vida útil del AFT va a estar dada por las fechas de inicio y de fin que se definan. También es configurable el valor residual.

#### **SAP 6.40:**

La corporación se ha desarrollado hasta convertirse en la quinta más grande compañía mundial de software. Cuenta con el módulo *Controlling* (CO), que permite el control de los gastos generales, costes de producto, consta de resultados y centros de beneficio. El subsistema Inventario de SAP permite reducir los costes de almacenamiento, transporte, cumplimiento de pedidos, manipulación de materiales y a la vez mejorar el servicio al cliente. Puede mejorar significativamente la rotación de inventario, optimizar el flujo de mercancías, acortar las rutas en su almacén o centro de distribución. Entre los beneficios adicionales de la gestión de inventarios se encuentran la mejora del flujo de caja, la visibilidad y la toma de decisiones. Para la gestión de almacenes puede realizar un inventario físico, optimizar los recursos del almacén, realizar un seguimiento de la cantidad y valor de todos sus materiales. Los empleados pueden planificar, introducir y documentar movimientos de almacén interno gestionando tanto las entradas como las salidas de mercancías, el almacenamiento, la recogida, el embalaje y los traslados físicos.

Entre sus características principales se encuentran:

- Implementado en .NET y WebSphere.
- SAP también ofrece una nueva plataforma tecnológica denominada SAP NetWeaver, esta plataforma tecnológica convierte a SAP en un programa *Web-enabled*, lo que significa que estaría totalmente preparado para trabajar con él mediante la web.
- Trabaja sobre el sistema operativo Windows.
- Soporte para bases de datos Oracle [**SAP**].

El sistema distingue entre tres tipos de depreciación:

- **Depreciación**

Correcciones de valor para activos fijos resultantes de la depreciación normal por el uso. Métodos definidos: Línea Recta y Dígitos Decrecientes.

- **Depreciación especial**

Depreciación requerida solo bajo la ley de impuestos. Este tipo de depreciación permite la depreciación en función del porcentaje sin tener en cuenta la depreciación real del activo fijo.

- **Amortización**

Correcciones de valor para activos fijos que se deprecian debido a daños, al progreso técnico, turnos de demanda, reducción de costes de sustitución o a malas inversiones, en contraposición a los que se deprecian por depreciación normal durante el uso del activo fijo [**SAP.**].

### **Valoración crítica de las soluciones extranjeras:**

En los ejemplos anteriores de software extranjeros utilizados en el país se evidencian un grupo de limitantes que frenan la utilización de estas aplicaciones en las empresas, por ejemplo:

Openbravo, en la actualidad solamente se pueden depreciar los AFT por el método de Línea Recta, además no dispone de un gestor de recursos humanos ni uno documental, además el soporte en cuanto a la exportación de datos y la interconexión con herramientas externas no es tan bueno. Además, aun siendo modular, no es posible implementar pequeñas funcionalidades nuevas creando módulos e instalándolos en el sistema [**Openbavo**] [**OpenERPSpain**].

Por otra parte, del SAP y el Exact Globe Next no se pueden utilizar todas sus potencialidades, es decir no se adaptan bien a las características de las empresas cubanas ya que al nivel de la práctica empresarial, la determinación de los parámetros organizativos y de funcionamiento resulta muy débil, por ejemplo; el Exact Globe Next, según lo descrito en su manual de usuario, no puede depreciar por un método tan elemental como lo es Dígitos Decrecientes y el sistema SAP solo puede depreciar por los métodos Línea Recta, Dígitos Decrecientes y mediante una depreciación especial que consiste en un porcentaje fijo que no se analiza el valor real de la depreciación [**Igor Lopes-Martínez, Martha Inés Gómez-Acosta, José Antonio Acevedo-Suárez. 2012**].

## 1.5 Modelo de desarrollo de software

Debido a la magnitud del proyecto CEDRUX se hace necesario que cada uno de los equipos de desarrollo posean un modelo estandarizado, así como una definición clara y precisa de las responsabilidades de cada uno de los roles involucrados en el desarrollo de la solución [**Equipo\_de\_desarrollo. 2009**]. En el Centro de Informatización de Entidades (CEIGE), los proyectos pueden definir su propia metodología basada en un modelo de desarrollo, en este caso la metodología está orientada a componentes y define claramente los artefactos que deben ser generados por cada uno de los roles involucrados en el proceso de desarrollo de software, el flujo de actividades y las responsabilidades de cada uno de los roles involucrados en el desarrollo de la solución.

Este modelo de desarrollo tiene las siguientes características:

**Centrado en la arquitectura:** La arquitectura orienta las prioridades del desarrollo y resuelve las necesidades tecnológicas y de soporte para el desarrollo. La arquitectura determina la línea base, 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.

**Orientado a componentes:** Las iteraciones son orientadas por el nivel de significancia arquitectónica de los componentes, los mismos son abstracciones arquitectónicas de los procesos de negocio y requisitos asociados que modelan, el componente es la unidad de medición y ordenamiento de las iteraciones.

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

**Ágil y adaptable al cambio:** Los clientes y funcionales están involucrados en el proyecto y poseen parte de las responsabilidades del éxito del mismo. Los cambios son conciliados semanalmente, discutidos y aprobados. El desarrollo de las partes formaliza solamente las características principales de la solución, priorizando los talleres y las comunicaciones entre las personas [**Laborí, Lázaro Pérez Bajuelo y Danier Estévez. 2010**].

Los artefactos que deben ser generados al desarrollar este componente según el modelo de desarrollo son:

- Modelo Conceptual
- Mapa de procesos
- Descripción de procesos del negocio
- Diagrama de procesos del negocio
- Reglas de negocio
- Especificación de requisitos
- Descripción de requisitos
- Modelo de datos
- Diagrama de clases de diseño
- Diagrama de componentes
- Diseño de casos de prueba
- Ficheros de código
- Glosario de términos

### **1.6 Arquitectura adoptada en el componente**

El sistema Cedrux utiliza el patrón arquitectónico Modelo Vista Controlador (MVC) el cual separa el modelado del dominio, las presentaciones al usuario y las acciones que se realizan. En el componente de depreciación no se aplica completamente este patrón ya que no se implementan clases controladoras, solo clases de modelo y clases de servicio. Utilizando la arquitectura de paquetes del sistema antes mencionado, para el componente se creó una estructura (Figura 1) que consta de un paquete “depreciacion”, que tiene dentro dos paquetes, “models” y “services”, luego dentro de “models” uno llamado “bussines”. Para una mejor comprensión en la siguiente figura se muestra la arquitectura de paquetes antes menciona.

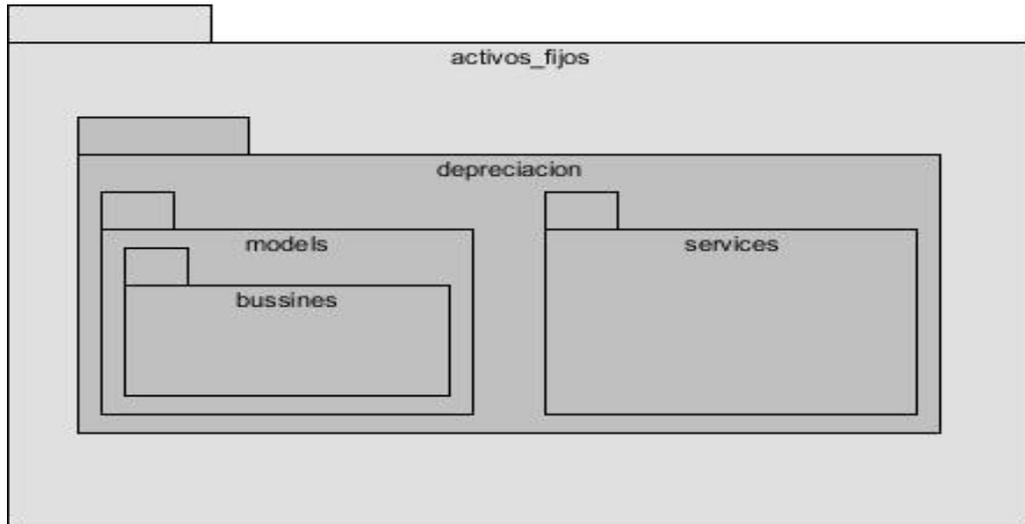


Figura 1. Diagrama de arquitectura de paquetes.

## 1.7 Lenguajes, herramientas y tecnologías a utilizar para el desarrollo del componente

Existe una gran variedad de herramientas y tecnologías disponibles en la actualidad para el desarrollo de software. CEIGE tiene definidos los lenguajes, herramientas y tecnologías que se deben utilizar para el desarrollo de software. Además, el componente que se propone como solución de esta investigación, constituye la continuación y perfeccionamiento del desarrollo de un subsistema del cual ya se hablaba anteriormente, por lo que las tecnologías y herramientas deben estar acordes para que sea posible la integración de esta solución.

### 1.7.1 Lenguajes de programación del lado del servidor

Son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y es capaz de enviar datos a los clientes. Los lenguajes del lado del servidor son independientes del cliente y les es indiferente el navegador o la versión de estos que se use [Torre, Aníbal de la. 2006]. Los lenguajes del lado del servidor pueden hacer consultas a bases de datos, peticiones a otras aplicaciones, conexiones de red, entre otras tareas.

#### PHP 5:

*Hypertext Preprocessor* (PHP) es el lenguaje por excelencia para la creación de páginas web del lado del servidor y aplicaciones web, se ha extendido a

numerosos aspectos, como la interacción con otras tecnologías, la nueva sintaxis de orientación a objetos y otras novedades más.

PHP es rápido, gratuito e independiente de plataforma, está desarrollado bajo las políticas de código abierto y a lo largo de la historia ha tenido muchas contribuciones por lo que posee una amplia librería de funciones y mucha documentación lo que facilita su aprendizaje.

Ventajas:

- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma, puede ser usado tanto en el sistema operativo Linux como en Windows.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modulación.
- Dificulta la organización por capas de la aplicación [**Instituto tecnológico de Veracruz**].

### **1.7.2 Lenguajes de programación del lado del cliente**

Son aquellos lenguajes que son interpretados y procesados por los navegadores o aplicaciones clientes en general, lo cual evita recargar de trabajo al servidor. Los lenguajes que se utilizaron son:

**JavaScript:**

Es compatible con la mayoría de los navegadores y esto lo hace uno de los más usados. Realiza algunos efectos especiales como, cambio de color, crear contenidos dinámicos y movimiento, además puede ejecutar instrucciones como crear páginas interactivas con el usuario, calculadoras, agendas y tablas de cálculo. El mayor recurso con que cuenta este lenguaje es el propio navegador ya sea Mozilla Firefox, Internet Explorer, Opera, Netscape y algunos más [**Bautista, Ana Maria. 2011**].

#### **XML:**

Sus siglas significan *eXtensible Markup Language*. Este lenguaje puede ser leído tanto por máquinas como por personas, utiliza el conjunto de caracteres UNICODE, lo que facilita la internacionalización. Ofrece un formato para la descripción de datos estructurados. Contiene tres características muy importantes que son: extensibilidad, estructura y validación. Con XML gran parte del procesamiento ocurre en la aplicación cliente ayudando a no congestionar la red. Es una herramienta de código abierto, independiente de plataforma y tiene un amplio soporte extendido a un sinnúmero de herramientas y desarrolladores [**L., Juan Cadillo**].

#### **HTML:**

HTML, siglas de *HyperText Markup Language* (lenguaje de marcado de hipertexto). Este lenguaje participa hasta cierto punto en la apariencia de un documento. Es el lenguaje de marcado más conocido y básico de todos para el desarrollo web. HTML es un sistema de etiquetas que no tiene un compilador por lo que los errores de sintaxis no se detectan y se visualizan como el navegador lo entienda. Es menos complejo que lenguajes como C++, Visual Basic o java, para trabajar con HTML solo se necesita un procesador de texto como el Bloc de Notas, el *Notepad++* o cualquier otro [**Web, Tekno**].

### **1.7.3 Estudio de los lenguajes de modelado utilizados**

#### **UML:**

UML, (*Unified Modeling Language*), es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos. UML permite modelar tanto cosas concretas como clases y esquemas de bases de datos como

cosas conceptuales, tales como funciones de sistema y procesos de negocio. Este lenguaje tiene sintaxis, artefactos, la capacidad de extinción y no está orientado a metodologías [Patricio Salinas Caro, Nancy Histchfeld K].

En el presente trabajo se utiliza UML para modelar diagramas como los de componentes, clases del diseño, modelo de datos y modelo conceptual permitiendo reflejar la realidad a través de objetos gráficos.

### **BPMN:**

BPMN o *Business Process Modeling Notation* (del español Notación para el Modelado de Procesos de Negocio), esta notación se utiliza en este trabajo para modelar los procesos de negocios que posee el componente de Depreciación de AFT.

Características:

- Es independiente de cualquier metodología de proceso, de cualquier herramienta y por tanto de cualquier fabricante.
- Esta notación para simplificar los diagramas introduce el concepto de eventos.
- Utiliza flujos como los de secuencia y los de mensaje.
- Crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos.
- BPMN permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento a todas las personas de una organización [Bizagi].

## **1.7.4 Herramientas para el desarrollo de la solución informática**

### **Herramienta de modelado Visual Paradigm 5.0:**

Visual Paradigm es una herramienta para desarrollo de aplicaciones utilizando modelado UML. Esta herramienta es utilizada para la creación de sistemas a gran escala ofreciendo estabilidad y confiabilidad tanto a los clientes como a los ingenieros de software, analistas de sistemas y arquitectos de sistemas que lo utilizan. Visual Paradigm también ofrece:

- Navegación intuitiva entre la escritura del código y su visualización.

- Potente generador de informes en formato PDF/HTML.
- Documentación automática *Ad-hoc*.
- Ambiente visualmente superior de modelado.
- Sofisticado diagramador automático de *layout* (plantilla).
- Sincronización de código fuente en tiempo real [**Software.com.ar**].

### **Navegador web Mozilla Firefox 17.0.1:**

Mozilla Firefox es un navegador web de código abierto y software libre, desarrollado por la Fundación Mozilla. El proyecto cuenta con miles de colaboradores en todo el mundo, lo que permite que cualquier error sea detectado y corregido rápidamente.

Principales características:

- Mejora en la velocidad de carga de las páginas.
- Sincronización entre diferentes dispositivos.
- Amplia personalización.
- Interfaz mejorada [**Varios\_autores2**].

### **IDE de desarrollo NetBeans IDE 6.9:**

Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE. NetBeans es un producto de Software Libre y gratuito sin restricciones de uso.

Esta herramienta es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Apache, control de versiones y *refactoring*. Tiene soporte para PHP [**leonardo.deseta**].

En su versión 6.9 destacan las características de soporte para Zend Framework y un nuevo formateador con muchas más reglas [**leonardo.deseta. 2010**].

### **Servidor web Apache 2.2.4:**

Apache es un servidor web gratuito, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Es indiscutiblemente uno de los mayores logros del Software Libre. Alguna de sus características:

- Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- Muy sencillo de configurar.
- Es *Open-source* (Código abierto).
- Amplias librerías de PHP y Perl a disposición de los programadores.
- Posee diversos módulos que permiten incorporarle nuevas funcionalidades, estos son muy simples de cargar.
- Es capaz de utilizar lenguajes como PHP, TCL y Python [Ing. Oiner Gómez Baryolo, Ing. Yoandry Morejón Borbón, Ing. Darien García Tejo].

### **Sistemas Gestores de Bases de Datos (SGBD):**

Los sistemas de gestión de bases de datos, en inglés (*Database Management System*) son un tipo de software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones. El objetivo principal es gestionar de manera clara, sencilla y ordenada un conjunto de datos. Un sistema gestor de bases de datos se debe caracterizar principalmente por:

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia.** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

- **Manejo de transacciones.** Una transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado.
- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD demora en proporcionar la información solicitada y en almacenar los cambios realizados [**Sánchez, Jorge. 2009**].

### **PostgreSQL 9.0:**

Es un sistema de gestión de bases de datos multiplataforma, es una derivación libre (*OpenSource*) del proyecto POSTGRES, de la universidad de Berkeley. Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos puntos de referencia se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL). Implementa el uso de subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle. Entre sus principales características se encuentran:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP,...) y cadenas de bits. También permite la creación de tipos propios.
- Incorpora una estructura de datos en forma de arreglo.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas y orientadas a operaciones con redes.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.

- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos [Varios\_autores1].

### **Sistema para el control de versiones *Subversion*:**

Sistema de control de versiones centralizado para compartir información que gestiona archivos, directorios y sus cambios a través del tiempo. *Subversion* es un repositorio en forma de árbol con una jerarquía de directorios y archivos además puede recrear un proyecto en cualquier momento así como recordar cada cambio que se haya realizado en el repositorio, en un archivo o en el árbol de directorios. El objetivo de un sistema de control de versiones es el de permitir editar de forma colaborativa y compartir información [Briano, Fernando. 2008].

### **TortoiseSVN 1.7:**

Para el sistema de control de versiones *Subversion* hay un cliente gratuito de código abierto que se llama TortoiseSVN que está desarrollado bajo la Licencia Pública General GNU (GPL). TortoiseSVN permite gestionar cambios en ficheros y directorios y examinar historiales.

Cabe destacar entre las características:

- Integración con el *shell* de Windows.
- Iconos sobre impresionados.
- Fácil acceso a los comandos de *Subversion*.

TortoiseSVN proporciona una serie de herramientas internas:

- TortoiseMerge: Permite ver las diferencias entre ficheros de texto, fusionar esos cambios e incluso revisar y aplicar parches.
- TortoiseBlame: Hace más fácil la lectura de los ficheros de autoría [rantamplan87].

## **1.7.5 Tecnologías utilizadas en la solución del problema**

### **AJAX:**

Ajax, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML). Es una tecnología de desarrollo web para la creación de páginas y aplicaciones interactiva que se ejecutan en el navegador del usuario y mantiene comunicación asíncrona con el servidor en segundo plano. Es patrón de diseño

que propone un nuevo modelo de interacción Web y combina tecnologías ya existentes como:

- El DOM (*Document Object Model*) para visualizar dinámicamente e interactuar con la información presentada,
- XML, XSLT para intercambiar y manipular datos.
- CSS para definir el aspecto del documento.
- JSON y JSON-RPC como alternativas a XML/XSLT.
- XMLHttpRequest para recuperar datos de forma asincrónica.
- Javascript como nexo de unión de todas estas tecnologías.

Se caracteriza por:

- Las aplicaciones son más interactivas, responden a las interacciones del usuario más rápidamente, al estilo aplicaciones de escritorio.
- Estas aplicaciones tienen un aspecto muy similar a las aplicaciones de escritorio tradicionales sin depender de *plugins* o características específicas de los navegadores.
- Se reduce el tamaño de la información intercambiada.
- Muchas micro-peticiones, pero el flujo de datos global es inferior.
- El procesamiento se realiza en la parte cliente, liberando la parte servidora.
- AJAX actualiza porciones de la página en vez de la página completa [JAPAR18, HACKER EL].

### **JSON:**

*JavaScript Object Notation* (JSON) es un formato de intercambio de datos abierto y basado en texto, que ofrece un intercambio estandarizado de datos, más conveniente para aplicaciones web de estilo Ajax aunque se puede usar prácticamente en cualquier escenario en que las aplicaciones necesiten intercambiar o almacenar información estructurada como texto.

Es legible y libre de plataforma además tiene a su disposición una amplia gama de implementaciones. Los datos con formato según el estándar JSON son ligeros y las implementaciones de JavaScript pueden analizarlos sintácticamente con increíble facilidad.

Características de JSON:

- Ofrece tipos de datos escalares y la capacidad de expresar datos estructurados a través de matrices y objetos.
- Compatibilidad con matrices y objetos nativos.
- Reconoce el valor *null* (nulo) de forma nativa.
- Reconoce el valor *null* de forma nativa [Atif Aziz, Scott Mitchell. 2007].

**CSS:**

Actualmente el tipo más común de hoja de estilo es la hoja de estilo en cascada (CSS) o *Cascading Style Sheets*. Existen varias especificaciones y niveles desarrollados por el *W3 Consortium*. CSS no es más que una colección de reglas que afectan a la apariencia de un documento y se pueden escribir dentro del código HTML (o XML) de la página web o en un archivo aparte, en este último caso hay que enlazar la página con ese archivo mediante una declaración en archivos externos.

Estas reglas se refieren al modo en que aparecerá un documento en pantalla cuando el usuario utilice un navegador, controlando por ejemplo el color, el fondo, tipo de fuente, apariencia de los bordes, márgenes, alineación y espacio entre caracteres. Es una tecnología que permite crear páginas web de una manera más precisa y homogénea. Gracias a las CSS se controlan los resultados finales de la página, pudiendo hacer muchas cosas que no se podían hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos y colores [Lapuent, María Jesús Lamarca. 2011].

**Zend Framework 1.7:**

Zend Framework es una librería PHP para la creación de aplicaciones web, está escrito en PHP5, provee un conjunto estandarizado de componentes que incluye un robusto componente MVC (Modelo Vista Controlador) para garantizar que su sitio web está estructurando de acuerdo a las mejores prácticas de programación y otros componentes como la autenticación de usuarios, la búsqueda, localización, creación de PDF, correo electrónico y conexiones a los servicios web. En este *framework* (marco de trabajo) es fácil desarrollar, tiene una vasta documentación, es modular, o sea que se usan componentes a voluntad, esto hace que sea fácil de aprender [Conde, Juan Carlos. 2011].

### ExtJS 2.2:

Es una librería de Javascript que permite desarrollar aplicaciones complejas utilizando componentes predefinidos así como un manejador de *layouts* (el layout, en este sentido, puede ser una especie de plantilla que presenta tablas o espacios en blanco) similar al que provee Java Swing, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno. ExtJS permite:

- Componentes UI del alto rendimiento y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias *Open-Source* y comerciales.

Con ExtJS se pretende lograr una apariencia similar a la de un entorno de escritorio, algo de lo cual siempre ha carecido la web y muy deseado por el usuario. Usar un motor de *render* como ExtJS permite tener además estos beneficios:

- Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo al servidor atender mayor cantidad de solicitudes.
- Comunicación asíncrona. Existe una comunicación con el servidor, la cual es transparente al usuario, despojándose de la obligación de hacer un clic o que el usuario ejecute una acción.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico [Rosas, Juan Eladio Sánchez. 2008].

### Doctrine Framework 1.2.1:

Doctrine es un ORM (*Object Relational Mapping*) de PHP, el más usado en este lenguaje y está inspirado en Hibernate de Java. *Object Relational Mapping*, viene a traducirse como Mapeo de objeto relacional. En concreto es una técnica o patrón arquitectónico, que permite comunicar dos sistemas distintos, típicamente una base de datos relacional con objetos de un lenguaje orientado a objetos, mediante

un sistema que “mapea” (vincula) ambos sistemas. Este ORM tiene sus características particulares:

- Muy sencillo de configurar. De hecho, no es necesario mantener complejos ficheros de configuración XML.
- Tiene su propio dialecto de SQL, llamado DQL (*Doctrine Query Language*).
- Puede generar los modelos (las clases PHP que representan una fila de la BD) a partir de la base de datos. Sólo habría que definir las dependencias entre ellos.
- Del mismo modo, también puede generar la base de datos a partir de los modelos [Jesús. 2011].

Las herramientas, lenguajes y tecnologías anteriormente descritas forman parte de las utilizadas en el marco de trabajo donde se desarrolla el sistema Cedrux, las mismas fueron seleccionadas para lograr una correcta integración entre este sistema y el componente de depreciación.

### **1.8 Conclusiones del Capítulo #1**

A partir de los estudios realizados en este capítulo se desarrolló el diseño teórico de la investigación quedando claramente expuestos los principales conceptos relacionados con Depreciación, Activos Fijos Tangibles y de los sistemas ERP, sus características, definiciones, deficiencias actuales e importancia de su utilización en las entidades cubanas. Se realizó la propuesta de herramientas y tecnologías necesarias para el desarrollo de esta investigación, entre las más útiles se destacan Netbeans para la implementación del código y el Visual Paradigm por la cantidad de artefactos que en él se generan.

## **CAPITULO #2: Descripción de la solución propuesta**

### **2.1 Introducción**

Esta fase se caracteriza por ofrecer un entendimiento preciso del negocio. Se hace necesario realizar un grupo de tareas que contribuirán a preparar el terreno para una eficaz implementación. Las tareas que se realizarán son las de modelar los procesos de negocios, levantar los requisitos funcionales y no funcionales, analizar y diseñar el componente de Depreciación.

### **2.2 Reglas del negocio**

Las reglas del negocio son de vital importancia para los objetivos trazados, estas reglas consisten en definiciones, normas o políticas que deben cumplirse, regulando así al negocio en general. Algunas de las reglas asociadas a los AFT son:

- Los AFT poseen un número de medio básico que es único en toda la entidad.
- Cada AFT pertenece a algún grupo de AFT.
- Para que un a AFT se le pueda dar apertura tiene que tener declarado su método de depreciación.
- A un AFT que ya se le dio apertura en la entidad no se le puede modificar el método de depreciación a menos que no se haya depreciado aun por primera vez.
- No se puede realizar el cierre sin haber realizado la depreciación.
- La depreciación acumulada es la sumatoria de las depreciaciones mensuales realizadas.
- El valor actual de un AFT es el valor inicial menos la depreciación acumulada.

### **2.3 Modelado de procesos de negocio**

Para mejorar la comprensión del sistema que se desea desarrollar se hace necesario el modelado del negocio, en este caso el modelado se realizará por procesos. Los procesos representan actividades que una persona, sistema o empresa realizan para producir una salida determinada. Modelar procesos de

negocio consiste en emular la realidad a esquemas de forma tal que esta pueda ser entendida. En esta solución se utiliza la notación BPMN, la cual ha permitido que el modelado quede claro y muestre las características particulares del negocio, obteniendo una visión acertada del funcionamiento del mismo.

### 2.3.1 Descripción del proceso de negocio Depreciación de AFT

A continuación se presenta la descripción del proceso de negocio del componente de Depreciación de AFT y para un mejor entendimiento el diagrama del proceso de negocio Depreciación de AFT (Figura 2).

|   |  |
|---|--|
| <b>Objetivo</b>                                 | Realizar el proceso de depreciación a los distintos tipos de AFT que se gestionan.   |
| <b>Evento(s) que lo genera(n)</b>               | Cierre contable al final de un período.  |
| <b>Pre condiciones</b>                          | Llegada del final del período a depreciar.<br>Deben estar especificados los métodos de depreciación en los AFT de la entidad.<br>Debe estar predefinido que la entidad va a depreciar.   |
| <b>Marco legal</b>                              | Resolución 10 del 2007 del Ministerio de Finanzas y Precios.<br>Resolución 9 del 2007 del Ministerio de Finanzas y Precios.<br>Resolución 87 del 2007 del Ministerio de Finanzas y Precios.<br>Resolución 379 del 2003 del Ministerio de Finanzas y Precios. |
| <b>Clientes internos</b>                        | Contador   |
| <b>Clientes externos</b>                        | N/A  |
| <b>Entradas</b>                                 | Tasa de depreciación   |
| <b>Flujo de eventos</b>                         |  |
| <b>Flujo básico &lt;Depreciación de AFT&gt;</b> |  |
| 1.  | Si la entidad deprecia se procede a analizar cada AFT de la entidad.   |
| 2.  | Se toma un AFT que no se haya procesado.   |
| 3.  | Se verifica que no se haya depreciado en el período actual.  |
| 4.  | Se verifica que el AFT puede depreciar atendiendo a las operaciones realizadas   |
| 5.  | Se verifica que el valor actual del AFT es mayor que el valor de recuperación.   |
| 6.  | Se calcula la nueva tasa de depreciación para el mes actual por el método que está configurado en el AFT   |
| 7.  | Se verifica que la tasa sea menor que el valor actual.   |

8. Se le suma a la depreciación acumulada el valor que depreció el AFT
9. Se incrementa en 1 la cantidad de períodos depreciados
10. Se guardan los datos modificados
11. Se hace un informe sobre el resultado del proceso
12. Concluye el proceso de depreciación.

**Pos-condiciones**

1. Los AFT de la entidad quedan depreciados.

**Salidas**

- 1 AFT depreciados

**Flujos paralelos**

N/A

**Pos-condiciones**

N/A

**Salidas**

N/A

**Flujos alternos**

**1.a La entidad no deprecia**

1. Concluye el proceso de depreciación

**Pos-condiciones**

1. No se deprecian los AFT.

**Salidas**

N/A

**3.a El AFT ya está depreciado en el período actual**

1. Se regresa al paso 2 del flujo básico

**Pos-condiciones**

No se deprecia ese AFT

**Salidas**

N/A

**4.a El AFT no puede depreciar atendiendo a las operaciones realizadas**

1. Se regresa al paso 2 del flujo básico

**Pos-condiciones**

No se deprecia ese AFT

**Salidas**

No se deprecia ese AFT

**5.a El valor actual del AFT es menor que el valor de recuperación**

1. Se regresa al paso 2 del flujo básico

**Pos-condiciones**

No se deprecia ese AFT

**Salidas**

N/A

**7.a La nueva tasa para mes actual es mayor que el valor actual del AFT**

---

1. Se regresa al paso 2 del flujo básico

---

**Pos-condiciones**

---

No se deprecia ese AFT

---

**Salidas**

---

N/A

---

**Asuntos pendientes**

---

N/A

---

### 2.3.2 Diagrama del proceso de negocio Depreciación de AFT

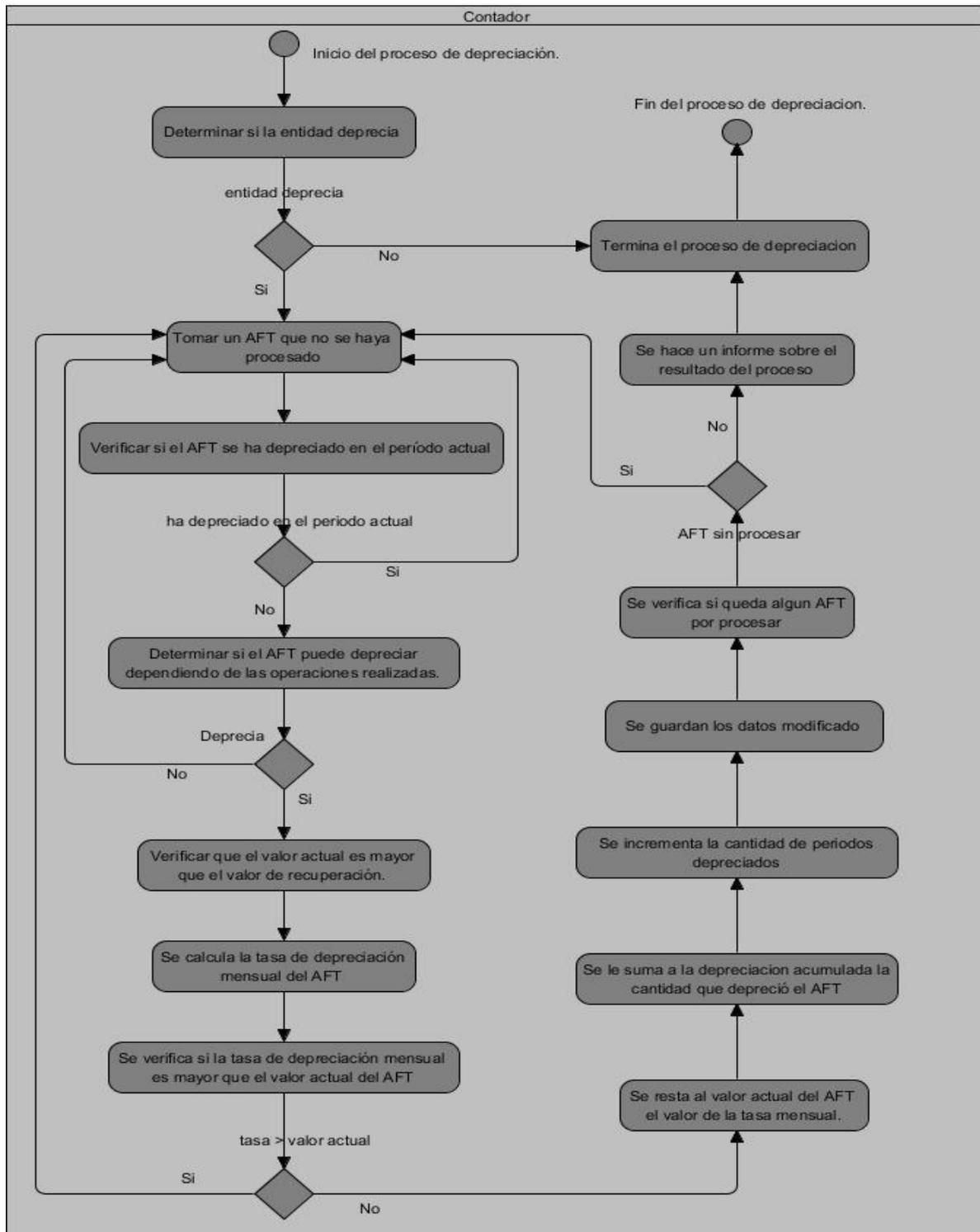


Figura 2. Diagrama del proceso de negocio Depreciación de AFT.

### 2.4 Modelo conceptual Depreciación de AFT

El Modelo Conceptual explica cuales son y cómo se relacionan los conceptos relevantes en la descripción del problema, es también conocido como modelo de

dominio y se presenta como uno o más diagramas de clases que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. Este modelo ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación.

El modelo que a continuación se presenta (Figura 3), constituye una representación gráfica de los conceptos más importantes que forman parte del subsistema AFT, destacando los conceptos necesarios para el desarrollo del componente de depreciación.

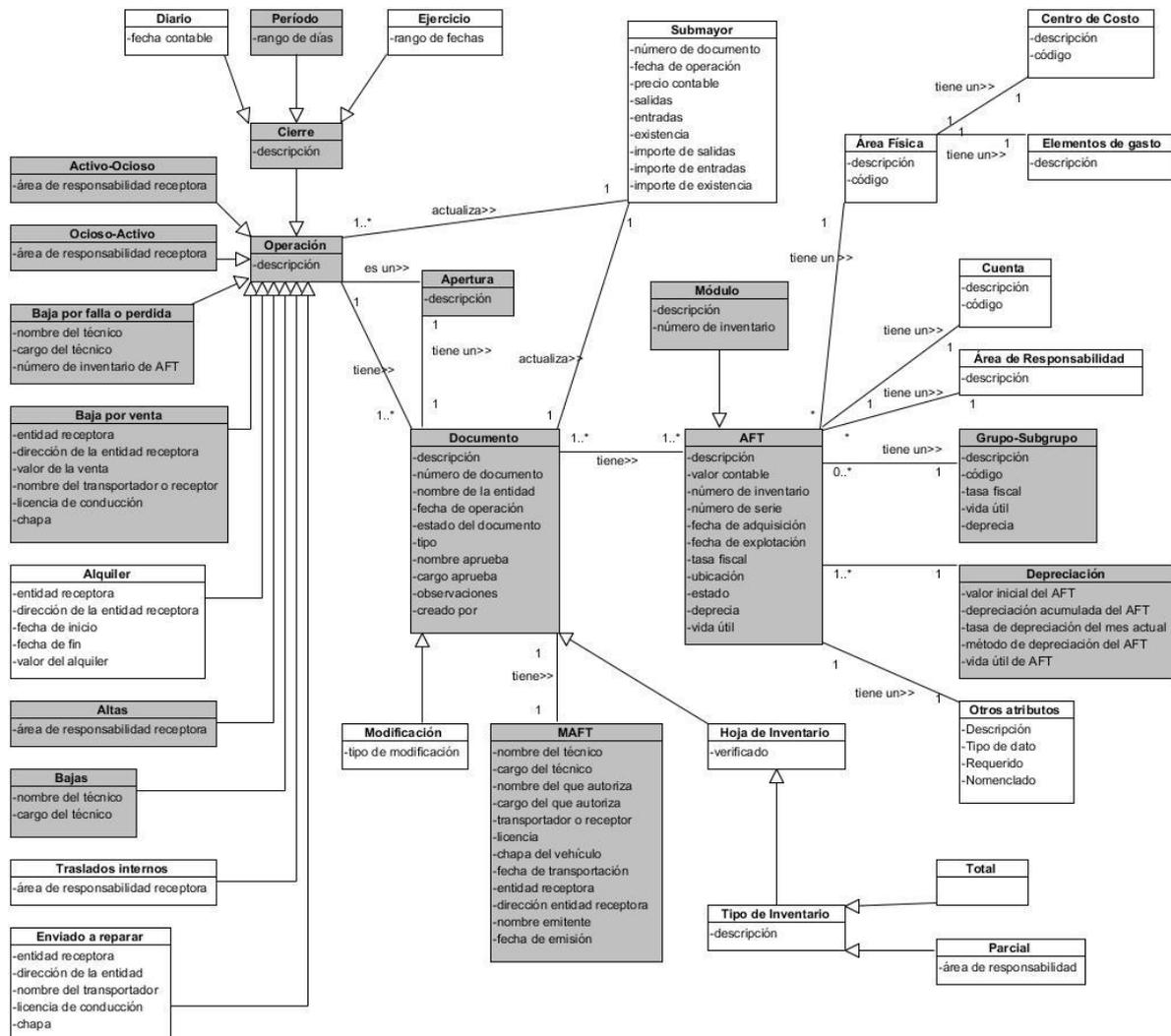


Figura 3. Diagrama del Modelo Conceptual de AFT.

### 2.4.1 Definición de los conceptos fundamentales

- Activo-Ocioso/Ocioso-Activo: Traslado de un AFT que se encuentra en Activo, cuando se declara ocioso o de uno ocioso que se incorpora

nuevamente a la actividad.

- **Baja por venta:** Es la cesión, mediante un precio fijado por acuerdo o por tasación entre vendedor y comprador de la posesión definitiva de un AFT.
- **Alta:** Es la adquisición de AFT nuevos, de uso, contruidos con medios propios y contratados con terceros. Incluye además la conclusión del proceso inversionista, así como los AFT que fueron recibidos en condición de Donaciones.
- **Baja:** Son los AFT que habiendo terminado o no su vida útil programada, el informe técnico determina que no tiene valor de uso.
- **Apertura:** Permite fijar un inventario inicial de los AFT en el subsistema y está disponible hasta que se ejecute el Cierre de Apertura. En este proceso se registran los datos de los AFT existentes en la entidad al momento de comenzarse a trabajar con el Módulo.
- **Depreciación:** La depreciación representa la pérdida del valor que sufren los AFT debido al desgaste ocasionado por el uso normal o extraordinario durante los períodos que presten servicios o participen en la producción.

## **2.5 Definición de requisitos funcionales y no funcionales**

Levantar requisitos es analizar y recopilar las necesidades de un cliente, como resultado de este proceso se obtiene un documento que describe qué es lo que realmente se necesita. La técnica utilizada para la captura de requisitos fue realización de entrevistas, apoyándose en un modelo de estas (ver anexo 2). Estos requisitos ayudarán a entender el problema contribuyendo así a una solución más acertada. Los requisitos se clasifican en dos grupos, funcionales y no funcionales. Posterior al desarrollo de esta solución los requisitos serán validados a través de prototipos de interfaz de usuario (ver anexo 3).

### **2.5.1 Requisitos funcionales**

Son los requisitos que especifican las funcionalidades del sistema (Tabla 2) y lo que este es capaz de hacer. Los requisitos funcionales dan una idea de las operaciones que se ejecutan sobre las entradas, produciendo así las salidas.

A continuación se presentan los requisitos funcionales identificados durante el proceso de levantamiento de requisitos correspondiente a dicha disciplina del modelo de desarrollo empleado.

### **Especificación de requisitos funcionales**

| <b>Nombre</b>  | <b>Complejidad</b> |
|--|--------------------|
| RF1. Calcular tasa de depreciación.  | Alta               |
| RF2. Ejecutar depreciación.  | Alta               |
| RF3. Cambiar el Método de Depreciación a un AFT.                                 | Alta               |
| RF4. Cambiar el Método de Depreciación a un grupo de AFT.                        | Alta               |
| RF5. Cambiar la cantidad de Unidades-Producidas/Horas-Máquina del AFT en el mes. | Alta               |
| RF6. Listar grupos de AFT que puede poseer la entidad.                           | Media              |
| RF7. Listar AFT que se encuentran en la entidad.                                 | Media              |
| RF8. Mostrar estadísticas.   | Alta               |

Tabla 2. Especificación de requisitos funcionales.

La descripción de los requisitos funcionales se anexa al documento principal de este trabajo (ver anexo 4).

### **2.5.2 Requisitos no funcionales**

Sin los requisitos no funcionales se vería afectado el buen funcionamiento del sistema. Estos requisitos junto a los funcionales complementan el total de requisitos que se levantan según las necesidades del cliente,

#### **Apariencia:**

Se concibe una interfaz debe ser amigable, fácil de usar, clara, no debe poseer sobrecargas de imágenes u otros elementos que distraigan al usuario del objetivo principal. Es una interfaz web, con colores suaves a la vista y las funcionalidades bien claras.

#### **Requisitos de usabilidad:**

La aplicación debe posibilitar un rápido aprendizaje y pronta familiarización del usuario con el sistema teniendo en cuenta que el usuario puede o no tener conocimientos informáticos. Este sistema debe ser práctico y fácil de usar.

**Requisitos de rendimiento:**

El sistema debe ser eficiente en cuanto al tiempo de respuesta, esta eficiencia estará determinada en gran medida por la velocidad de las consultas en la base de datos y el aprovechamiento de los recursos del servidor y de la aplicación cliente.

**Requisitos de soporte:**

En aras de garantizar el soporte a los clientes de esta herramienta, se dotará a los clientes con un manual de usuario donde estará documentada todas las funcionalidades, en este manual el usuario podrá encontrar toda la ayuda necesaria desde la instalación del sistema hasta su uso. Además se capacitará al administrador del sistema para realizar el mantenimiento del mismo.

**Requisitos de seguridad:**

El sistema mediante un conjunto de políticas de seguridad permite asignar ciertos privilegios a determinados usuarios y grupos de usuarios, posibilitando que un usuario no pueda hacer más de lo que debe. La información que se maneja en el sistema debe ser protegida rigurosamente contra situaciones inestables y de corrupción. Cada contraseña debe cumplir con un grupo de reglas de fortaleza que contribuyen a la custodia de la información.

**Requisitos de software:**

El sistema se ejecutará sobre cualquier plataforma pero se recomienda para las PC servidoras tener Ubuntu Server. Las herramientas específicas para las PC servidoras y para las PC clientes son las siguientes:

Se recomienda tener 2 PC servidora, una con:

- Servidor web, Apache Web Server v2.0 o superior.
- PHP v5.2.4 o superior.

Y otra con:

- Gestor de bases de datos PostgreSQL v8.2 o superior.

Para las PC clientes:

- Un navegador web, preferentemente el Mozilla Firefox versión 2.0.17 o superior.

### Requisitos de hardware:

Para las PC servidoras:

- Procesador: 1 GHZ
- RAM: 1 GB
- Tarjeta de red: 1
- UPS: 1
- Disco duro: 160 GB
- Lector de CD: 1

Para las PC clientes:

- Procesador: 1.40 GHZ
- Tarjeta de red: 1
- RAM: 256 MB, recomendado 512 MB o mas

### 2.6 Patrones utilizados

En el momento de comenzar el trabajo con patrones se hizo un análisis minucioso de qué patrones pudieran ser útiles en aras de alcanzar un alto grado de organización en el código. Los patrones que se utilizan son un conjunto de buenas prácticas de programación que han dotado al componente que se propone como solución de la robustez y la versatilidad necesaria para integrarse al resto de la aplicación, estos patrones son los siguientes:

#### Patrones de diseño, GRASP:

Son patrones generales de diseño de software para la asignación de responsabilidades, GRASP es el acrónimo de *General Responsibility Assignment Software Patterns*, del español Patrones Generales de Software para la Asignación de Responsabilidades. Se considera que son una serie de "buenas prácticas" recomendable en el diseño de software.

- Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con qué las conoce y con qué recurre a ellas.
- Alta Cohesión: Orienta en la medida de lo posible un alto nivel de comunicación entre las clases buscando obtener acceso a toda la información de manera simple.

- Experto: Con la utilización de este patrón se logra mantener el encapsulamiento ya que los objetos utilizan su propia información para resolver las tareas.

### **Patrones de diseño GoF:**

Los patrones de diseño GoF se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. Dentro de los estructurales se utilizaron el patrón Fachada.

- Fachada: Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema [Sanjuan, Angel Vilorio. 2012].

En el trabajo se aplican los patrones de diseño Alta Cohesión y Bajo Acoplamiento, en la clase DepreciacionService.php ya que esta realiza peticiones y delega funcionalidades a la clase DepreciacionModel.php. Esto ha permitido hacer estas clases más independientes. En la clase DepreciacionModel.php se evidencia el patrón experto ya que ella es la responsable de desarrollar la principal funcionalidad que resuelve el proceso de depreciación. Otro patrón utilizado es el Fachada, el cual se aprecia en la clase DepreciacionService.php debido a que la función principal de esta es mantener la comunicación de las funcionalidades implementadas en el modelo con el resto de los componentes (Figura 4).

## 2.7 Diagrama de clases del diseño

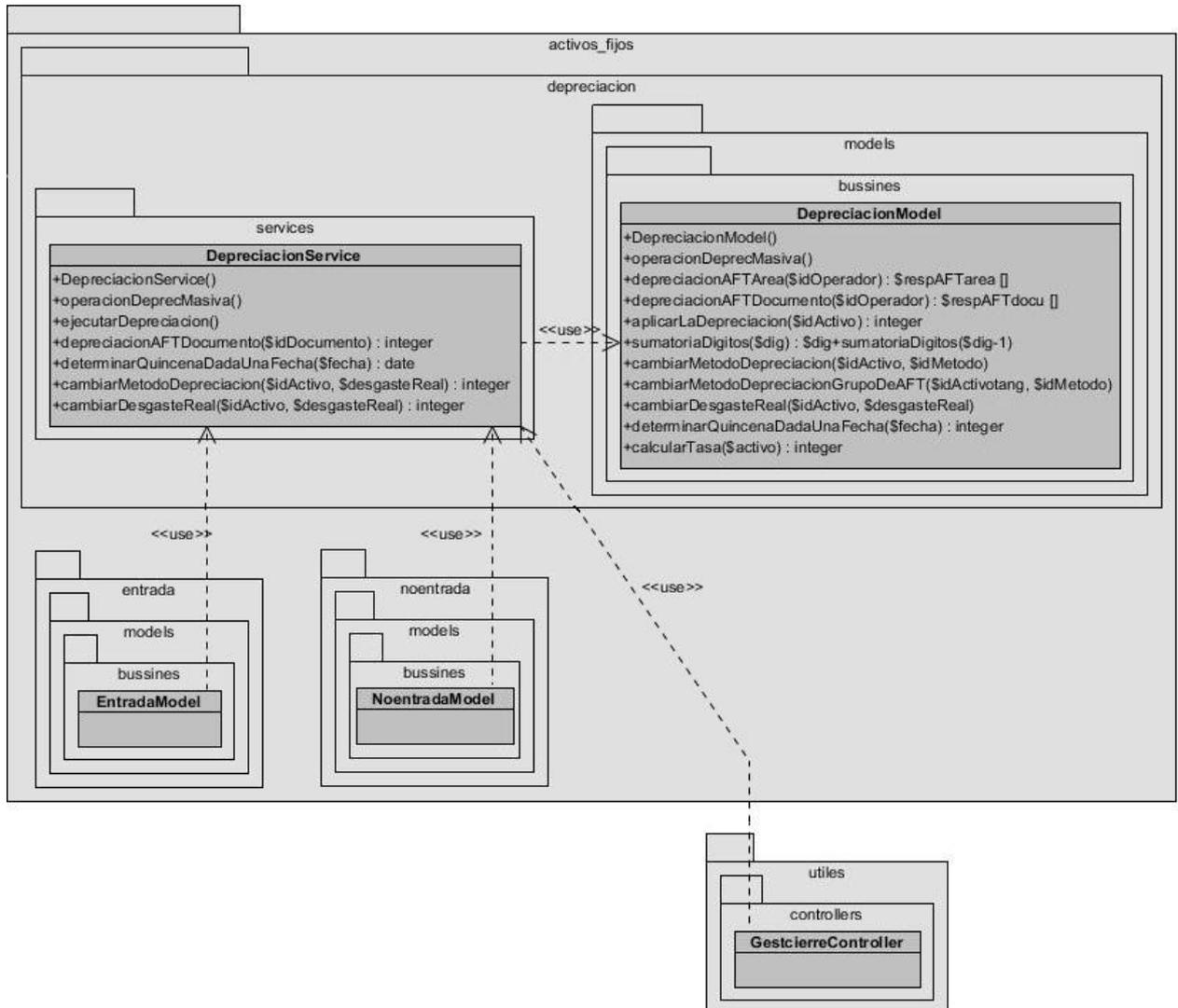


Figura 4. Diagrama de clases del diseño.

El diseño de clases queda como se muestra en la figura anterior, siguiendo una línea de arquitectura definida en el sistema. La razón de ser del Componente de Depreciación no es recibir o mostrar datos a algún usuario sino brindar servicios a otros componentes o módulos que lo soliciten, es por eso que no posee Java Script ni PHTML.

La solicitud del servicio se realiza a través del IoC (Inversión de Control), clase la cual se encarga de recepcionar solicitudes específicas, pedir a otros módulos o componentes que realicen las acciones que sean necesarias para después dar una respuesta adecuada. La clase DepreciacionService.php es la que va a ser llamada en algún momento y va a brindar el servicio de depreciación pero ella no es la que contiene las validaciones necesarias para depreciar los AFT.

DepreciacionService.php tiene como misión llamar a la clase DepreciacionModel.php y con las respuestas que obtenga entonces responder ella. En la clase DepreciacionModel.php es donde se desarrolla la implementación de la lógica del negocio y donde ocurre la depreciación del AFT.

Sobre los documentos de una entidad pueden ocurrir movimientos de entradas (alta, traslados de ocioso a activo) y de no entradas (bajas, traslados de activo a ocioso), en los cuales se necesitaron hacer algunas validaciones. En el momento en que ocurre el movimiento en dependencia de del tipo de movimientos que sea se puede solicitar el servicio de depreciación para depreciar los AFT que se encuentran en el documento al cual se le está haciendo el movimiento. Otro proceso que se necesitó modificar fue el de Cierre de período ya que este proceso es el que debe solicitar el servicio para depreciar masivamente los AFT de la entidad.

Este diagrama no contempla las modificaciones realizadas en las clases loC.php y otras clases que sirven como enlaces para ejecutar el servicio. Tampoco se incluye en el diagrama la interfaz del momento de apertura, que antes permitía asignarle a un AFT el número de serie, número de inventario, entre otros atributos, ahora se le asigna también al AFT el método de depreciación.

Otras clases que no se incluyen en el diagrama y que se implementan para dar una solución integral, son las que posibilitan una interfaz apropiada para listar los grupos de AFT que puede poseer la entidad y cambiar su método de depreciación. También se creó una interfaz encargada de listar los AFT que ya pasaron por el proceso de apertura y se dieron de alta en la entidad, con un número de serie y de inventario, además esta interfaz debe permitir cambiar el método de depreciación de estos AFT siempre y cuando no se hayan depreciado por primera vez y cambiar la cantidad de horas trabajadas y la cantidad de unidades producidas para los AFT que deprecien por los método Horas-Maquina y Unidades-Producidas respectivamente.

## 2.8 Diagrama de modelo de datos



Figura 5. Modelo de datos.

En el diagrama del modelo de datos se muestran las tablas que están directamente relacionadas con el proceso de depreciación, específicamente la tabla **dat\_numeroserieactivo** que es donde se guardan los principales datos del activo. Al incorporar el componente de depreciación al sistema Cedruz no se agregan tablas nuevas a la base de datos de este sistema, solo se añaden nueve campos, los últimos en la tabla antes mencionada (Figura 5).

## 2.9 Conclusiones de este capítulo

En este capítulo se explican las funcionalidades con que contará el sistema, así como las clases y las tablas con las que guarda relación el componente, de lo que se puede concluir que los artefactos generados posibilitan un mejor entendimiento del negocio. Con la realización satisfactoria de tareas como la especificación de los requisitos y la descripción del Proceso de Depreciación quedan sentadas las bases para continuar a la fase de implementación.

## CAPITULO #3: Implementación y prueba del componente propuesto

### 3.1 Introducción

La implementación es la fase donde quedan desarrolladas las clases que dan solución al problema planteado, se modelan los componentes que interactúan con la depreciación y a través de una matriz se muestra la relación entre ellos. Una buena práctica en la programación de cualquier aplicación informática es emplear estándares de codificación los cuales brindan claridad a la hora de leer el código y facilidad en el momento de implementarlo. En la solución que se implementa se utilizan una serie de estándares de codificación y patrones que ayudan a un correcto desarrollo de software, sin embargo, siempre existe la posibilidad de cometer errores, es por eso que una solución informática luego de implementada debe ser sometida a una fase de prueba. El componente que se implementa es sometido a un conjunto de pruebas que comprobarán la eficiencia de las clases y las funcionalidades que responden a los requisitos descritos.

### 3.2 Diagrama de relación entre componentes

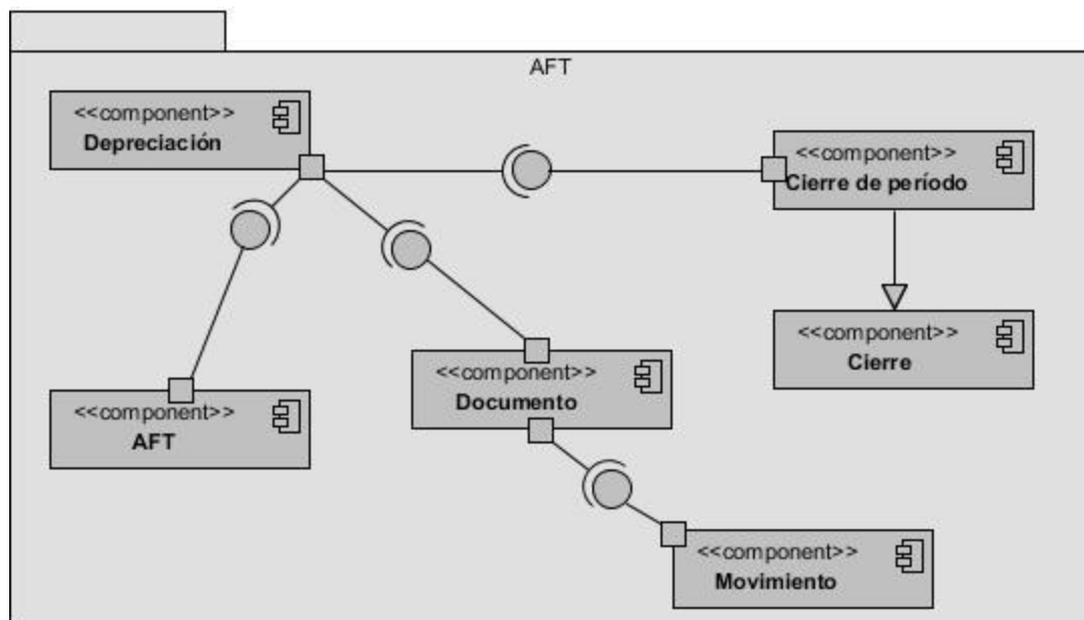


Figura 6. Diagrama de Relación entre componentes.

En el diagrama de componente se muestran las organizaciones y dependencias

lógicas entre el componente Depreciación y el resto de los componentes del sistema Cedrux con los que este se relaciona (Figura 6). El componente Depreciación guarda relación con el componente Cierre de Período ya que la depreciación masiva de los activos se produce en el momento del cierre de período. También guarda relación con el componente AFT debido a que la depreciación ocurre precisamente sobre los activos fijos tangibles y se relaciona con el componente Documento ya que es mediante documentos que se realizan los Movimientos de activos en las entidades.

Los nombres de los servicios que brinda el componente de depreciación fueron creados de la forma más descriptible posible, a continuación se muestra una tabla (Tabla 3) con las funcionalidades que enlazan a los componentes AFT, Movimientos y Cierre de período con el componente Depreciación:

### 3.3 Matriz de relación entre componentes

| Componentes internos                                     |  |   |                       |
|--|--|---|-----------------------|
|  | AFT  | Movimientos                                 | Cierre de período     |
| D<br>e<br>p<br>r<br>e<br>c<br>i<br>a<br>c<br>i<br>ó<br>n | +cambiarDesgasteReal(\$idActivo, \$desgasteReal)           | +depreciacionAFTDocumento(\$idDocumento)    | +depreciacionMasiva() |
|  | +cambiarMetodoDepreciacion(\$idActivo, \$desgasteReal)     | +cambiarAtributoDepreciacion(\$idDocumento) |                       |
|  | +cambiarMetodoDepreciacionGrupoDeAFT(\$codigo, \$idMetodo) |   |                       |
|  | +determinarQuincenaDadaUnaFecha(\$fecha)                   |   |                       |
|  |  |   |                       |

Tabla 3. Relación del componente depreciación con componentes internos.

### 3.4 Estándares de codificación

- **Identación en el código (Tabs o espacios):**

La indentación en el código (Figura 7) son aquellos tabs o espacios que se ponen delante de las líneas de código y se han utilizado para lograr una estructura más organizada y entendible.

```
151 function sumatoriaDigitos($dig){
152     if($dig >= 0 )
154         return $dig + sumatoriaDigitos($dig-1);
155     }
156
```

Figura 7. Ejemplo de Identación de código.

- **Cabecera del archivo:**

Se han insertado cabeceras a los archivos de código (Figura 8) para orientar acerca de qué tipo de operaciones se realizan en la clase, quien la implementó, a que paquete pertenece, la versión y el nombre del archivo.

```
1 <?php
2 /**
3  * Modelo para servicios de operaciones de depreciacion en
4  * Activos Fijos Tangibles (AFT).
5  * @author Ernesto Fraga Quintero
6  * @package logistica
7  * @version 1.1
8  * @filesource DepreciacionModel.php
9  */
```

Figura 8. Ejemplo de cabecera de archivo.

- **Comentario de descripción en las funciones:**

Como se muestra en la figura (Figura 9), los comentarios redactados al inicio de las funcionalidades describen el objetivo de la misma así como los parámetros que consume y el tipo de resultado que arroja.

```
11 /*
12  * Constructor de la clase
13  * @param none
14  * return void
15  */
16 public function DepreciacionModel() {
17     parent::ZendExt_Model();
18 }
```

Figura 9. Ejemplo de comentario de descripción de funciones.

- **Nomenclatura de clases:**

Se utiliza la notación PascalCasing (Figura 10), la cual define que se escribe la primera letra del nombre de la clase con mayúscula y el resto con minúscula, si el nombre es compuesto las demás palabras se escriben juntas y empezando con mayúscula también.

```
1 <?php
2
3 class DepreciacionService extends ZendExt_Model {
4
5     protected $model;
6
7     public function DepreciacionService() {
8         $this->model = new DepreciacionModel();
9     }
}
```

Figura 10. Ejemplo de nomenclatura de clases.

- **Nomenclatura según el tipo de clases:**

La nomenclatura según el tipo de clases (Figura 11) consiste en agregar al final del nombre una palabra que identifica el tipo de la clase, por ejemplo si pertenece al modelo se le agrega “*Model*” y si es una clase de servicio se le adiciona “*Service*”.

```
8 * @filesource DepreciacionModel.php
9 */
10 class DepreciacionModel extends ZendExt_Model {
11     /*
12     * Constructor de la clase
13     * @param none
14     * return void
15     */
}
```

Figura 11. Ejemplo de nomenclatura según el tipo de clases.

- **Nomenclatura de funciones:**

El nombre de las funciones se escribe con minúscula y si la palabra es compuesta se escribe junta y con mayúscula la primera letra de cada palabra menos la primera palabra, como se muestra en la imagen (Figura 12).

```
52
53
54     public function aplicarLaDepreciacion($idActivo) {
55         $activo = $this->integrator->activosFijosT->obtenerAf
56         $valrecup = $activo->valorrecuperacion;
}
```

Figura 12. Ejemplo de nomenclatura de función.

- **Nomenclatura de variable:**

La nomenclatura utilizada en las variables (Figura 13) es similar a la nomenclatura usada en las funciones, solo se diferencia en que para comenzar una variable se le antepone el símbolo “\$” el cual es predefinido por el lenguaje PHP.

```
53  
54 public function aplicarLaDepreciacion ($idActivo) {  
55     $activo = $this->pIntegrator->activosFijos->obten
```

Figura 13. Ejemplo de nomenclatura de variable.

### 3.5 Pruebas internas

La fase de pruebas de software es la encargada de evaluar si la solución que se implementó cumple con los requisitos especificados por el cliente, también se encarga de chequear que los componentes se integren correctamente y de identificar errores y corregirlos antes de entregar el producto al cliente en el menor tiempo posible. La calidad y la usabilidad son cualidades imprescindibles en una aplicación informática es por eso que las pruebas se integran dentro de las diferentes fases de desarrollo por las que transita un software. Las pruebas que se utilizan para validar el Componente de Depreciación son:

- Pruebas de caja blanca, técnica de camino básico.
- Pruebas de caja negra, técnica de la partición de equivalencia. Pruebas de aceptación por parte del equipo de desarrollo.

#### 3.5.1 Pruebas de Caja Negra

##### Pruebas de aceptación:

Las pruebas de aceptación son aquellas realizadas por los usuarios con carácter previo al paso de producción de una nueva versión del producto. Se trata de pruebas de Caja Negra que se realizan en un entorno de real de aplicación del software en el que el usuario final verifica si las funcionalidades pactadas para la entrega y recogidas en catálogos de requisitos, cumplen las expectativas [Jummp. 2011].

Las pruebas de aceptación del componente fueron realizadas por parte del equipo de desarrollo de la línea de logística el cual emitió una carta evaluando la implementación del mismo y plasmando las no conformidades (ver anexo 5).

Las pruebas de Caja Negra o pruebas Funcionales se realizan sobre la interfaz del software, entendiendo por interfaz las entradas y salidas del mismo. No es necesario conocer su lógica de funcionamiento, únicamente la funcionalidad que

debe realizar. Estas pruebas se basan en la especificación del programa o componente el cual se ve como una “Caja Negra” cuyo comportamiento solo puede ser determinado mediante el estudio de las entradas y salidas obtenidas. Estas pruebas permiten encontrar:

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

Entre las técnicas para desarrollar la prueba de Caja Negra se encuentran:

- Técnica de la Partición de Equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

De las técnicas mencionadas se aplicará la de Partición de Equivalencia ya que es una de las más efectivas. Permite definir casos de prueba que declaren clases de errores, reduciendo el número de casos de prueba a desarrollar para demostrar que las funciones del software son operativas [**Varios autores4**].

Se seleccionó un conjunto de entradas y salidas sobre las que se realizan las pruebas y para seleccionarlas se consideró que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en el sistema y como consecuencia producen una o varias salidas que revelan la presencia de defectos o errores. Para verificar que la aplicación se comporta según los requisitos establecidos por el cliente se diseñan casos de pruebas usando la técnica antes mencionada. A continuación se tomaron el requisito “Cambiar Método de Depreciación a un AFT” y se desarrolló un caso de prueba.

**Requisito a probar:** “Cambiar Método de Depreciación a un AFT”.

**Condiciones de ejecución:**

1. Se debe identificar y autenticar ante el sistema y tener los permisos para ejecutar esta acción.
2. Se debe seleccionar dentro de este el módulo “Logística” que está en el menú “Inicio”, “AFT”->”Nomencladores”->”Activo fijo tangible por entidad”.
3. Se debe seleccionar un grupo de AFT para mostrar sus existencias en la entidad.

| Nombre del requisito                        | Descripción general   | Escenarios de pruebas   | Flujo del escenario   |
|---|---|---|---|
| 1: Cambiar Método de Depreciación a un AFT. | Es una funcionalidad que permite al usuario cambiarle el método de depreciación a un AFT siempre y cuando no se halla depreciado por primera vez. | EP 1.1: Se le cambia el método de depreciación a un AFT que no ha depreciado aun por primera vez. | - Se asigna el nuevo método de depreciación al AFT.   |
|   |   | EP 1.2: Se le cambia el método de depreciación a un AFT que ha depreciado al menos en un período. | - Se asigna el nuevo método de depreciación al AFT.<br>- Se muestra el mensaje emitido por el sistema con el texto: “No se puede cambiar el método de depreciación a un AFT que se haya depreciado al menos una vez”. |

Tabla 4: Requisito a probar “Cambiar el Método de Depreciación a un AFT”.

| No | Nombre de campo         | Tipo      | Válido  | Inválido |
|----|-------------------------|-----------|---|----------|
| 1  | Método de depreciación. | Combobox. | Valores predeterminados. Se selecciona una de las opciones mostradas. | No.      |

Tabla 5: Descripción de variable Cambiar el Método de Depreciación a un AFT.

| Id del escenario | Escenario   | Método de Depreciación   | Respuesta del sistema  | Resultado de la prueba |
|------------------|---|--------------------------|--|------------------------|
| EP 1.1           | Se le cambia el método de depreciación a un AFT que no ha depreciado aun por primera vez y tiene predefinido un método de depreciación distinto a "Dígitos Decrecientes". | V(Dígitos Decrecientes). |  |                        |
| EP 1.2           | Se le cambia el método de depreciación a un AFT que ha depreciado al menos en un período y tiene predefinido un método de depreciación distinto a "Dígitos Decrecientes". | I(Dígitos Decrecientes). | El sistema emite un mensaje de notificación: "No se puede cambiar el método de depreciación a un AFT que se haya depreciado al menos una vez." |                        |

Tabla 6: Juego de datos para el requisito "Cambiar el Método de Depreciación a un AFT".

### 3.5.2 Pruebas 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 [Esquivel, Ing. Karina].

Los pasos que se siguieron para la aplicación de esta técnica son:

1. Enumerar las sentencias de código del procedimiento.

```

public function aplicarLaDepreciacion($idActivo) { #1
    $activo1 = $this->pIntegrator->activosFijosT->obtenerAftDadoIdAFTER($idActivo); #1
    $activo = $activo1[0]; #1
    $valrecup = $activo->valorrecuperacion; #1
    $valoractual = $activo->valor; #1
    $deprecia = $activo->deprecia; #1
    $vidautil = $activo->vidautil; #1
    if($valoractual > $valrecup && $vidautil != 0 && ($deprecia == 1 || $deprecia == 3)) { #2
        if($activo->valorinicial == null) #3
            $activo->valorinicial = $activo->valor; #4
        $tasadepreciacion = $this->calcularTasa($activo); #5
        if($tasadepreciacion < $valoractual){ #6
            $activo->tasadepreciacion = $tasadepreciacion; #7
            $activo->depreciacionacum += $tasadepreciacion; #7
            $activo->depreciacionfiscal += $tasadepreciacion; #7
            $activo->cantidadmesesdepreciados += 1; #7
            $activo->valor = $activo->valor - $tasadepreciacion; #7
            switch ($deprecia) { #8
                case 1: #9
                    $this->pIntegrator->activosFT->modificarActivoFT($activo); #9
                    return 1; #17
                case 3: #10
                    $activo->deprecia = 4; #10
                    $this->pIntegrator->activosFT->modificarActivoFT($activo); #10
                    return 3; #17
            }
        }
        else{ #11
            $activo->finaldepreciacion = 1; #11
            $this->pIntegrator->activosFT->modificarActivoFT($activo); #11
            return 0; #17
        }
    }
    else{ #12
        if($valoractual < $valrecup || $vidautil == 0){ #12
            $activo->finaldepreciacion = 1; #13
            $this->pIntegrator->activosFT->modificarActivoFT($activo); #13
            return 0; #17
        }
        switch($deprecia) { #14
            case 2: #15
                $activo->deprecia = 1; #15
                $this->pIntegrator->activosFT->modificarActivoFT($activo); #15
                return 2; #17
            case 4: #16
                return 4; #17
            }
        }
    }
}

```

Figura 14. Método aplicarLaDepreciacion().

La Figura 14 muestra una funcionalidad de la clase DepreciacionModel.php. A partir de la estructura de este fragmento de código se enumeran las sentencias, contribuyendo así a determinar la cantidad de nodos que tendrá el grafo asociado a este método.

2. A partir del código fuente se dibuja un grafo de flujo asociado.

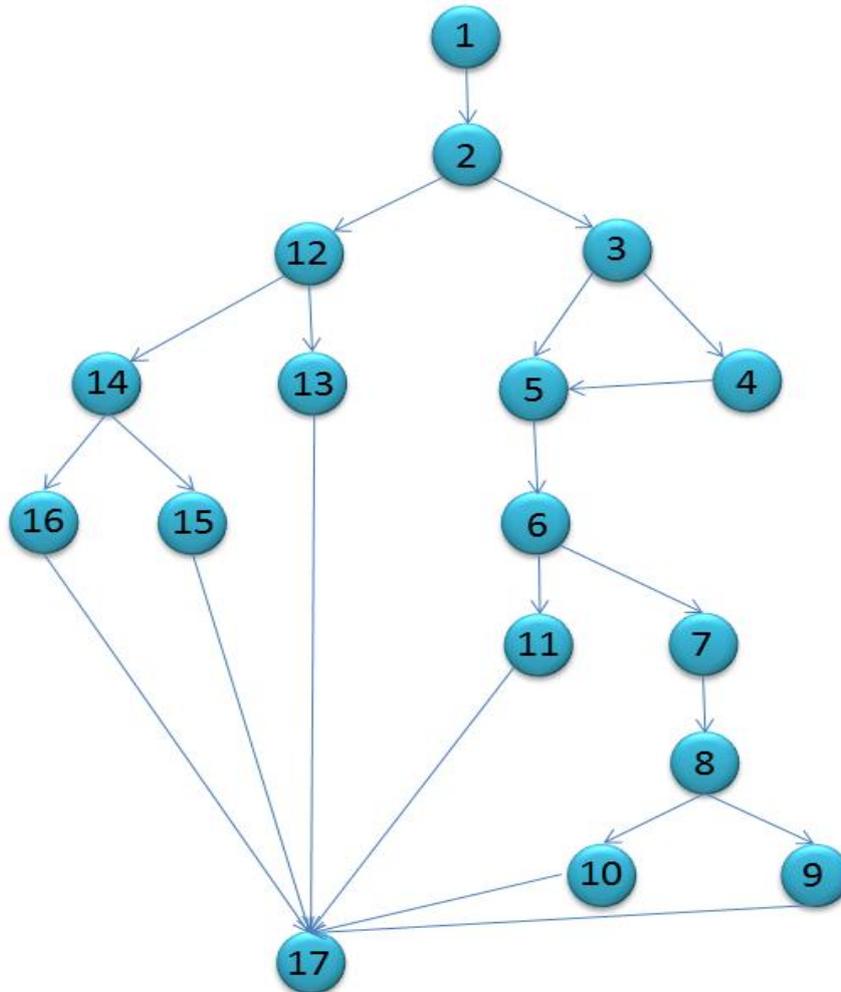


Figura 15. Grafo del método aplicarLaDepreciacion().

El grafo representado (Figura 15) permite observar fácilmente la diversidad de caminos que se pueden recorrer al ejecutar el código al cual pertenece este grafo. La representación gráfica del código a través del grafo ayuda a determinar la cantidad de aristas, la cantidad de regiones y la cantidad de nodos y nodos predicados del grafo.

3. Se calcula la complejidad ciclomática del grafo.

Fórmulas para calcular complejidad ciclomática:

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

$$V(G) = (22 - 17) + 2$$

$V(G) = 7$  Siendo "A" la cantidad total de aristas y "N" la cantidad total de nodos.

Se puede usar también:

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

$$V(G) = 6 + 1$$

$$V(G) = 7$$

Siendo "P" la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = R$$

$$V(G) = 7$$

Siendo "R" la cantidad total de regiones, para cada fórmula " $V(G)$ " representa el valor del cálculo.

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es de 7, lo que significa que existen a lo sumo siete posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado. Seguidamente es necesario representar los caminos básicos (Tabla 7) por los que puede recorrer el flujo.

| Número. | Camino Básico.        |
|---------|-----------------------|
| 1       | 1-2-3-4-5-6-7-8-9-17  |
| 2       | 1-2-3-5-6-7-8-9-17    |
| 3       | 1-2-3-4-5-6-7-8-10-17 |
| 4       | 1-2-3-4-5-6-11-17     |
| 5       | 1-2-3-5-6-11-17       |
| 6       | 1-2-12-13-17          |
| 7       | 1-2-12-14-15-17       |

Tabla 7. Caminos Básicos.

Luego de haber extraído siete caminos básicos se ejecutan los casos correspondientes a cada uno de estos caminos. Para realizar los casos de pruebas es necesario cumplir con un grupo de normas:

**Descripción:** se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.

**Condición de ejecución:** se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.

**Entrada:** se muestran los parámetros que entran al procedimiento

**Resultados Esperados:** se expone el resultado que se espera que devuelva el procedimiento.

**Caso de prueba para el camino básico 1 [1-2-3-4-5-6-7-8-9-17]:**

Descripción: Los datos de entrada son atributos de un activo fijo tangible.

Condición de ejecución: [\$valoractual > \$valrecup], [\$vidauttil != 0], [\$deprecia = 1], [\$valinicial = null], [\$tasadepreciacion < \$valoractual]

Entrada: [\$idActivo = 40], [valor = 300], [\$valoractual = 300], [\$valrecup = 0], [\$vidauttil = 4], [\$valinicial = null], [(\$tasadepreciacion = 20), [\$deprecia = 1]

Resultados Esperados: El método retorna 1, lo cual indica que el AFT se deprecia correctamente y se debe seguir depreciando el próximo período.

**Caso de prueba para el camino básico 2 [1-2-3-5-6-7-8-9-17]:**

Descripción: Los datos de entrada son atributos de un activo fijo tangible.

Condición de ejecución: [\$valoractual > \$valrecup], [\$vidauttil != 0], [\$deprecia = 1], [\$valinicial != null], [\$tasadepreciacion < \$valoractual]

Entrada: [\$idActivo = 40], [valor = 300], [\$valoractual = 300], [\$valrecup = 0], [\$vidauttil = 4], [\$valinicial = 300], [(\$tasadepreciacion = 20), [\$deprecia = 1]

Resultados Esperados: El método retorna 1, lo cual indica que el AFT se deprecia correctamente y se debe seguir depreciando el próximo período.

**Caso de prueba para el camino básico 3 [1-2-3-4-5-6-7-8-10-17]:**

Descripción: Los datos de entrada son atributos de un activo fijo tangible.

Condición de ejecución: [\$valoractual > \$valrecup], [\$vidauttil != 0], [\$deprecia = 3], [\$valinicial = null], [\$tasadepreciacion < \$valoractual]

Entrada: [\$idActivo = 40], [valor = 300], [\$valoractual = 300], [\$valrecup = 0], [\$vidauttil = 4], [\$valinicial = null], [(\$tasadepreciacion = 20), [\$deprecia = 3]

Resultados Esperados: El método retorna 4, lo cual indica que el AFT se deprecia correctamente y se debe dejar de depreciar a partir del próximo período.

**Caso de prueba para el camino básico 4 [1-2-3-4-5-6-11-17]:**

Descripción: Los datos de entrada son atributos de un activo fijo tangible.

Condición de ejecución: [\$valoractual > \$valrecup], [\$vidauttil != 0], [\$deprecia = 1 o [\$deprecia = 3], [\$valinicial = null], [\$tasadepreciacion > \$valoractual]

Entrada: [\$idActivo = 40], [valor = 300], [\$valoractual = 15], [\$valrecup = 0], [\$vidauttil = 4], [\$valinicial = null], [(\$tasadepreciacion = 20), [\$deprecia = 1]

Resultados Esperados: El método retorna 0, lo cual indica que el AFT no se pudo depreciar debido a que llego al final de su vida útil.

#### **Caso de prueba para el camino básico 5 [1-2-3-5-6-11-17]:**

Descripción: Los datos de entrada son atributos de un activo fijo tangible.

Condición de ejecución: [\$valoractual > \$valrecup], [\$vidauttil != 0], [\$deprecia = 1 o [\$deprecia = 3], [\$valinicial != null], [\$tasadepreciacion > \$valoractual]

Entrada: [\$idActivo = 40], [valor = 300], [\$valoractual = 15], [\$valrecup = 0], [\$vidauttil = 4], [\$valinicial = 300], [(\$tasadepreciacion = 20), [\$deprecia = 1]

Resultados Esperados: El método retorna 0, lo cual indica que el AFT no se pudo depreciar debido a que llego al final de su vida útil.

#### **Caso de prueba para el camino básico 6 [1-2-12-13-17]:**

Descripción: Los datos de entrada son atributos de un activo fijo tangible.

Condición de ejecución: [\$valoractual < \$valrecup o \$vidauttil = 0]

Entrada: [\$idActivo = 40], [valor = 300], [\$valoractual = 300], [\$valrecup = 0], [\$vidauttil = 0], [\$valinicial = 300], [(\$tasadepreciacion = 20), [\$deprecia = 1]

Resultados Esperados: El método retorna 0, lo cual indica que el AFT no se pudo depreciar debido a que llego al final de su vida útil.

#### **Caso de prueba para el camino básico 7 [1-2-12-14-15-17]:**

Descripción: Los datos de entrada son atributos de un activo fijo tangible.

Condición de ejecución: [\$valoractual > \$valrecup], [\$vidauttil != 0], [\$deprecia = 2]

Entrada: [\$idActivo = 40], [valor = 300], [\$valoractual = 300], [\$valrecup = 0], [\$vidauttil = 4], [\$valinicial = 300], [(\$tasadepreciacion = 20), [\$deprecia = 2]

Resultados Esperados: El método retorna 2, lo cual indica que el AFT no debía depreciarse y no se depreció pero a partir del próximo período comenzara a depreciar.

### **3.6 Validación de las variables declaradas en el problema a resolver**

**Agilidad:**

La agilidad es una variable en función del tiempo y a través de ella se demostró que con la implementación de este componente el proceso de depreciación se efectúa de forma más rápida (Tabla 8).

**Facilidad:**

La facilidad con que se realiza un proceso está en función del esfuerzo necesario para realizar la acción. La facilidad de un proceso depende de la facilidad y la cantidad de subprocesos que tenga este proceso. A través de esta variable se demuestra cuán fácil puede llegar a ser el proceso de depreciación al ser automatizado (Tabla 8).

A continuación se refleja cómo se realiza la depreciación en una entidad que no tenga automatizada la gestión de la depreciación de AFT y como se realizaría si este proceso estuviese automatizado.

|  | <b>Agilidad:</b>   | <b>Facilidad:</b>  |
|--|--|--|
| <b>N<br/>O<br/>A<br/>U<br/>T<br/>O<br/>M<br/>A<br/>T<br/>I<br/>Z<br/>A<br/>D<br/>O</b> | Para depreciar los AFT, el usuario tiene que hacer de forma manual el cálculo de la tasa de depreciación y luego evaluar un grupo de condiciones para determinar si el AFT debe depreciar o no. Luego si el AFT deprecia debe restarle al valor del AFT la tasa calculada y sumarle esa tasa a la depreciación que va acumulando este AFT durante el tiempo. Un usuario no puede realizar este proceso a 10 AFT en menos de 5 min. | <p>Sin contar que este proceso esta propenso a errores por el hecho de realizarse manual, el usuario tiene que:</p> <ol style="list-style-type: none"> <li>1- Establecer en un documento la vida útil y el método por el que va a depreciar el AFT.</li> </ol> <p>Esta acción se realiza una sola vez.</p> <p>A demás tiene que:</p> <ol style="list-style-type: none"> <li>1- Consultar datos en varios documentos a la vez.</li> <li>2- Tomar decisiones y realizar los cálculos necesarios para llevar a cabo el proceso.</li> <li>3- Actualizar los datos del AFT por los nuevos que se calcularon.</li> </ol> <p>Este grupo de acciones se realizan al final de cada período.</p> |

|  |  |  |
|--|--|--|
| A<br>U<br>T<br>O<br>M<br>A<br>T<br>I<br>Z<br>A<br>D<br>O | <p>En el sistema Cedrux, con este componente, ejecuta la depreciación de forma automática, lo cual permite depreciar 10 AFT en menos de un cuarto de minuto.</p> | <p>Debido a que el proceso de depreciación se ejecuta de manera automática y transparente al usuario en el momento en que se realiza el cierre de período o algún movimiento que solicite el servicio, el usuario lo único que necesita es:</p> <p>1- Configura en el momento en que se le da de Alta al AFT, el método por el cual va a depreciar y la vida útil. Esta acción solo requiere de dos clics y de teclear un número.</p> <p>Esta acción se realiza una sola vez.</p> <p>Para los AFT que deprecien por el método de “Horas-Maquina” y “Unidades-Producidas”, el usuario antes de que se ejecute la depreciación tiene que:</p> <p>1- Introducir en el sistema la cantidad de horas que realmente trabajó o la cantidad de productos que realmente produjo el AFT.</p> <p>Esta acción se realiza al final de cada período puesto que es imposible saber con exactitud el tiempo de uso o la productividad real que tendrá en el futuro un AFT.</p> |
|--|--|--|

Tabla 8. Validación de las variables declaradas en el problema a resolver.

### 3.7 Conclusiones de este capítulo

Utilizando buenas prácticas de programación se dotó al sistema Cedrux con funcionalidades para la gestión y la configuración de la depreciación. El código resultante de la implementación ha quedado organizado y legible gracias a la utilización de patrones y estándares de codificación. Luego de aplicar los métodos

de Caja Negra y Caja Blanca, para validar tanto la interfaz como el correcto funcionamiento interno del software, se alcanzaron resultados satisfactorios desde el punto de vista interno y funcional.

## **Conclusiones generales**

El proceso de depreciación en la mayoría de entidades cubanas es lento o ineficaz debido a que el proceso ocurre de forma manual o utilizando soluciones informáticas poco eficientes. Este componente toma ideas positivas de algunos sistemas existentes e intenta corregir un grupo de deficiencias que estos presentan. Con el componente desarrollado, las empresas que utilicen Cedrux tendrán asegurada la gestión y la configuración de la depreciación de los AFT. El valor fundamental de este componente se expresa en la contribución a facilitar y agilizar el proceso de depreciación de AFT en las entidades cubanas.

## **Recomendaciones**

Se recomienda que en posteriores versiones del componente se logre:

- Facilitar aún más las tareas de configuración y ejecución de la depreciación.
- Optimizar el tiempo de ejecución del proceso de depreciación.
- Integrar el componente en nueva versiones que se desarrollen de Cedrux.

## Bibliografía

- Atif Aziz, Scott Mitchell. 2007.** Msdn Microsoft. [Online] Junio 26, 2007. [Cited: 20, 2012.] <http://msdn.microsoft.com/es-es/library/bb299886.aspx>.
- autores, varios. 2011.** Definición abc. [Online] 2011. [Cited: 2 12, 2012.] <http://www.definicionabc.com/tecnologia/erp.php>.
- autores, Varios. 2010.** Manual de Usuario del Modulo Ativos Fijos Tanjibles. 2010.
- autores, grupo de. EcuRed. [Online] [Cited: 4 16, 3013.]** [http://www.ecured.cu/index.php/M%C3%A9todos\\_Cient%C3%ADficos\\_de\\_Investigaci%C3%B3n](http://www.ecured.cu/index.php/M%C3%A9todos_Cient%C3%ADficos_de_Investigaci%C3%B3n).
- Bautista, Ana Maria. 2011.** Progra web. [Online] 9 25, 2011. [Cited: 12 7, 2012.] <http://ingsistanabautista.blogspot.com/2011/09/lenguajes-de-programacion-del-lado-del.html>.
- Bizagi. BizAgi. [Online] [Cited: 12 7, 2012.]** <http://www.bizagi.com/docs/BPMNbyExampleSPA.pdf>.
- Briano, Fernando. 2008.** Picandocodigo. [Online] 2008. [Cited: 2 20, 2012.] <http://picandocodigo.net/downloads/docs/subversion-presentacion-01.pdf>.
- Caro, Ricardo Enriquez. 2012.** Administración moderna. [Online] 2 11, 2012. [Cited: 6 23, 2012.] <http://thesmadruga2.blogspot.com/2012/06/activo-fijo-tangible.html>.
- CITMATEL. 2013.** RODAS XXI. Activos Fijos. [Online] 2 5, 2013. [Cited: 12 6, 2013.] <http://www.rodasxxi.cu/activos%20fijos.php>.
- Conde, Juan Carlos. 2011.** Libera tu Código. [Online] 3 11, 2011. [Cited: 2 14, 2012.] <http://www.liberatucodigo.com/2011/03/introduccion-zend-framework/>.
- Depreciacion.net. Depreciacion.net. [Online] [Cited: 4 9, 2013.]** <http://www.depreciacion.net/activos-fijos.html>.
- Entorno Virtual de Aprendizaje. 2007.** Entorno Virtual de Aprendizaje. EVA. [Online] 2007. [Cited: 12 5, 2012.] [http://eva.uci.cu/file.php/249/Contabilidad\\_y\\_Finanzas/Cuentas.pdf](http://eva.uci.cu/file.php/249/Contabilidad_y_Finanzas/Cuentas.pdf).
- Equipo\_de\_desarrollo. 2009.** Modelo de Desarrollo orientado a componentes del proyecto ERP-CUBA. La Habana : s.n., 2009.
- ERP Globe de Exact. Next Time Software. [Online] [Cited: 12 6, 2012.]** <http://www.nexttimesoftware.com/index.php/soluciones/erp-globe-de-exact>.

**Esquivel, Ing. Karina.** UNAN LEÓN / DOCENTE DPTO. COMPUTACIÓN. [Online] [Cited: 5 15, 2013.]

[http://kesquivel.files.wordpress.com/2009/08/tema15\\_estrategiaspruebasw.ppt](http://kesquivel.files.wordpress.com/2009/08/tema15_estrategiaspruebasw.ppt).

**Exact.** Exact. [Online] [Cited: 12 6, 2012.] <http://www.exact.es/software-y-servicios/software/sistemas-erp/algunos-beneficios>.

**16-Fernández, Cesar Martínez. 2012.** bibliomaster. [Online] 2012. [Cited: 4 8, 2013.] <http://www.bibliomaster.com/pdf/3173.pdf>.

**Frida Ruso Armada, Lic. Ariamna Rivero Bolaños, Lic. Lisi Campos Chaurero, Lic. Yoenny Leyva Vega.** COFIN HABANA. [En línea] [Citado el: 12 de 2 de 2012.] [http://cofinhabana.fcf.uh.cu/index.php?option=com\\_docman&task=doc\\_download&gid=320&Itemid=](http://cofinhabana.fcf.uh.cu/index.php?option=com_docman&task=doc_download&gid=320&Itemid=). ISSN 2073-6061.

**Gonzalez Brito, Henry Raúl, Lezcano Lozada, Yuniesky. 2005.** Sistema de Inventario Participativo de la UCI. La Habana : s.n., 2005.

**Guillermo J. Sierra, Bernabé Escobar Pérez.** SISTEMAS DE INFORMACIÓN INTEGRADOS (ERP). [Gráficas ORMAG] Sevilla : Asociación Española de Contabilidad y Administración de Empresas.

**Hidalgo, Noidis Barroso. 2012.** CIG-LOG-N-AFT-i2602. La Habana : s.n., 2012.

**Igor Lopes-Martínez, Martha Inés Gómez-Acosta, José Antonio Acevedo-Suárez. 2012.** Scielo. [Online] 12 11, 2012. [Cited: 3 6, 2013.]

[http://scielo.sld.cu/scielo.php?pid=S1815-59362012000300011&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S1815-59362012000300011&script=sci_arttext).

**22-Ing. Oiner Gómez Baryolo, Ing. Yoandry Morejón Borbón, Ing. Darien García Tejo.** GESTEC. [Online] [Cited: 2 20, 2012.]

[http://www.gestec.disaic.cu/ponencias%202010/CUBA/\(55\)Oiner%20GomezTRABAJO%20ARQUITECTURA%20TECNOL%C3%93GICA%20PARA%20EL%20DESARROLLO%20DE%20SOFTWARE.doc](http://www.gestec.disaic.cu/ponencias%202010/CUBA/(55)Oiner%20GomezTRABAJO%20ARQUITECTURA%20TECNOL%C3%93GICA%20PARA%20EL%20DESARROLLO%20DE%20SOFTWARE.doc).

**Instituto tecnológico de Veracruz.** Instituto tecnológico de Veracruz. [Online] [Cited: 12 7, 2012.] <http://www.prograweb.com.mx/pweb/0203ladoServidor.html>.

**JAPAR18, HACKER EL.** Elmasterdelaweb. [Online] [Cited: 2 20, 2012.]

<http://elmasterdelaweb.wikispaces.com/file/view/QUE+ES+AJAX.pdf>.

**Jesús. 2011.** Genoma. [Online] 2011. [Cited: ] <http://gnoma.es/blog/introduccion-a-doctrine-2/>.

**johanniitha. 2013.** Investigación científica. [Online] 04 08, 2013. [Cited: 11 16, 2011.] <http://investiguemosconjohanniitha.blogspot.com/2011/11/metodos-de-la->

investigacion.html.

**Jummp. 2011.** Jummp. [Online] 8 14, 2011. [Cited: 5 19, 2013.]

<http://jummp.wordpress.com/2011/08/14/desarrollo-de-software-pruebas-de-aceptacion/>.

**L., Juan Cadillo.** Educación, Sistemas, Redes y TIC. [Online] [Cited: 12 7, 2012.]

<http://conocimientoysistemas.wordpress.com/tag/caracteristicas-xml/>.

**Laborí, Lázaro Pérez Bajuelo y Danier Estévez. 2010.** Diseño e Implementación del Componente Nómina del subsistema Capital Humano del Sistema Integral de Gestión CEDRUX. La Habana : s.n., 2010.

**Lapuent, María Jesús Lamarca. 2011.** hipertexto. [Online] 12 5, 2011. [Cited: 2 20, 2012.] <http://www.hipertexto.info/documentos/css.htm>.

**Larisa González Alvarez, Tahirí Rivero Álvarez, Osmar Leyet Fernández. 2011.**

Serie Científica. [Online] 02 12, 2011. [Cited: 04 08, 2013.]

<http://publicaciones.uci.cu/index.php/SC/article/download/162/358>.

**leonardo.deseta.** Dos Ideas. [Online] [Cited: 2 19, 2012.]

<http://www.dosideas.com/wiki/NetBeans>.

**leonardo.deseta. 2010.** Dos Ideas. [Online] 6 17, 2010. [Cited: 2 19, 2012.]

<http://www.dosideas.com/noticias/java/915-netbeans-ide-69-ya-esta-disponible.html>.

**Lliulla, Carmen. 2012.** Webspacio. [Online] 5 22, 2012. [Cited: 1 5, 2012.]

<http://myspace.wihe.net/top-10-empresas-tecnologia/>.

**Marcelo de la Cruz Heredia, Carlos Palomino Hurtado. 2007.** Siscont Software contable-Financiero Profesional, Guía Práctica de auto aprendizaje. Perú : s.n., 2007.

**Marcelo de la Cruz Heredia, Carlos Palomino. 2007.** Scribd. [Online] 2007.

[Cited: 3 6, 2013.] <http://es.scribd.com/doc/61333202/LIBRO-SISCONT-2007>.

**Marisel Sosa Porteiro, Pedro H. Cobo Morales.** DISAIC Casa Consultora.

[Online] [Cited: 2 12, 2012.]

<http://www.disaic.cu/modules.php?name=content&pa=showpage&pid=818>.

**Openbavo.** Openbavo. [Online] [Cited: 12 6, 2012.]

<http://www.openbravo.com/es/why-openbravo-50-reasons>.

**OpenERPEspain.** OpenERPEspain. [Online] [Cited: 3 2013, 6.]

<http://www.openerspain.com/hay-otros-erps-libres>.

**Ortiz, Wilfredo Rodriguez Rodriguez. 2007.** Portal del Minbas. [Online] 2007.

[Cited: 12 12, 2012.]

<http://siscont.tm.minbas.cu/Docs/Documentos/Forms/AllItems.aspx?RootFolder=%2FDocs%2FDocumentos%2FManuales>.

**Patricio Salinas Caro, Nancy Histchfeld K.** Universidad de Chile. Facultad de Ciencias Físicas y Matemáticas, Departamento de Ciencias de la Computación.

[Online] [Cited: 12 8, 2012.]

<http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>.

**rantamplan87.** Que duro es ser Informático. [Online] [Cited: 4 21, 2010.]

<http://nosoyfriky.wordpress.com/2010/04/21/como-usar-tortoisenv-manual-bsico/>.

**Rosas, Juan Eladio Sánchez. 2008.** Desarrollo en Web. [Online] 10 22, 2008.

[Cited: 2 20, 2012.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.

**Sánchez, Jorge. 2009.** OPENLIBRA. [Online] 2009. [Cited: 2 20, 2012.]

<http://www.etnassoft.com/biblioteca/sistemas-gestores-de-bases-de-datos/>.

**Sanjuan, Angel Viloría. 2012.** Angel Viloría. [Online] marzo 21, 2012. [Cited: 2 20,

2012.] <http://aviloria.blogspot.com/2012/03/patrones.html>.

**SAP.** SAP. [Online] [Cited: 12 6, 2012.] [www.sap.com/spain/solutions/business-suite/erp/index.epx](http://www.sap.com/spain/solutions/business-suite/erp/index.epx).

**SAP.** SAP Help Portal. Depreciación automática de activos fijos. [Online] [Cited: 04 09, 2013.]

[http://help.sap.com/saphelp\\_financialsondemand\\_1302/es/KTP/Products/BYD\\_ODF/WEKTRA\\_for\\_Work\\_Centers/FIN/Ess/ESS\\_FA/ESS\\_PRO\\_Automatic\\_Depreciation.html](http://help.sap.com/saphelp_financialsondemand_1302/es/KTP/Products/BYD_ODF/WEKTRA_for_Work_Centers/FIN/Ess/ESS_FA/ESS_PRO_Automatic_Depreciation.html).

**Software.com.ar.** Software.com.ar. [Online] [Cited: 2 18, 2012.]

<http://www.software.com.ar/visual-paradigm-para-uml.html>.

**Tineo, Claudia Giovanna Huaytalla. 2009.** monografias.com. [Online] 2009.

[Cited: 6 12, 2013.] <http://www.monografias.com/trabajos69/depreciacion-perdida-valor-bien/depreciacion-perdida-valor-bien.shtml>.

**Torre, Aníbal de la. 2006.** Lenguajes del lado servidor o cliente. [Online] 2006.

[Cited: 2 18, 2012.]

[http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html).

**Varios\_autores1.** EcuRed. [Online] [Cited: 2 20, 2012.]

<http://www.ecured.cu/index.php/PostGreSQL>.

**Varios\_autores2.** EcuRed. [Online] [Cited: 2 19, 2012.]

[http://www.ecured.cu/index.php/Mozilla\\_Firefox](http://www.ecured.cu/index.php/Mozilla_Firefox).

**Varios\_autores3.** EcuRed. [Online] [Cited: 3 2013, 6.]

[http://www.ecured.cu/index.php/RODAS\\_XXI](http://www.ecured.cu/index.php/RODAS_XXI).

**Varios\_autores4.** EcuRed. [Online] [Cited: 5 19, 2013.]

[http://www.ecured.cu/index.php/Pruebas\\_de\\_caja\\_negra](http://www.ecured.cu/index.php/Pruebas_de_caja_negra).

**Varios\_autores5.** EcuRed. [Online] [Cited: 3 2013, 6.]

[http://www.ecured.cu/index.php/Versat\\_Sarasola](http://www.ecured.cu/index.php/Versat_Sarasola).

**Web, Tekno.** Tekno Web. [Online] [Cited: 12 7, 2012.]

[http://www.teknoweb.co/index.php?option=com\\_content&view=article&id=61:html&catid=36:plataformas-web](http://www.teknoweb.co/index.php?option=com_content&view=article&id=61:html&catid=36:plataformas-web).