

Universidad de las Ciencias Informáticas

Facultad 15



**Análisis, diseño e implementación de los componentes
para la Gestión de Factores que Influyen en el Plan y
Configuración de Perfil para el Sub-sistema Dirección por
Objetivo.**

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

Autor(es): Lidiarys Hernández Ponce

Arnoldo Montes de Oca Montero

Tutor(es): Ing. Néstor Bernal Vidal

Ciudad de La Habana, Junio de 2010

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Informatización para la Gestión de Entidades (CEIGE) de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 18 días del mes de junio del año 2010.

Lidiarys Hernández Ponce

Arnoldo Montes de Oca Montero

Firma del Autor

Firma del Autor

Néstor Bernal Vidal

Firma del Tutor

No solo sabremos resistir cualquier agresión sino que sabremos vencer a cualquier agresión y nuevamente no tendríamos otra disyuntiva que aquella con que iniciamos la lucha revolucionaria, la de la libertad o la muerte, solo que ahora libertad quiere decir patria y la disyuntiva nuestra sería patria o muerte.

Fidel Castro Ruz

DATOS DE CONTACTO

Nombre y Apellidos: Ing. Néstor Bernal Vidal.

Correo: ubernal@uci.cu

Situación laboral: 2 años de adiestrado.

Institución: Universidad de las Ciencias Informáticas (UCI).

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba. Código postal 19370.

AGRADECIMIENTOS

Lidianys:

Quisiera agradecerle de forma muy especial a mi mamá por ser una madre excepcional. Por estar siempre a mi lado y darme lo mejor de ella. Por educarme y guiarme por la vida con paciencia y amor infinito. Porque además de ser mi madre siempre ha sido mi mejor amiga. Gracias porque sin ella no hubiera podido llegar donde he llegado. Le agradezco a mi papá por sus consejos a tiempo. Por su ayuda y apoyo incondicional. Por creer en mí siempre y darme la confianza para crecer en la vida. Le agradezco a Eliza por su cariño y apoyo todos estos años. Le agradezco a mi abuela que ve por mis ojos y que no se cansa de luchar por mí, por ser mi motivo de inspiración y mi fuerza para mirar siempre adelante y porque ella es el regalo más preciado que me ha dado Dios. Agradezco a mis hermanas, mis primos y mis tíos por estar a mi lado siempre que los he necesitado y porque me han impulsado a continuar aun cuando no he tenido fuerzas. Le agradezco a mi novio y compañero de tesis por haberme soportado todos estos años con amor y paciencia. Y por compartir conmigo los buenos y malos momentos de mi vida. Le agradezco a Néstor, mi tutor, por su ayuda durante toda la tesis, por defenderla como si fuera la de él. Le agradezco a Elena por ser mi amiga en todo momento, aunque la haya dejado durmiendo sola muchas veces. A Rayza porque supo llorar y reír conmigo mis alegrías y tristezas. A Guille por su amistad y ayuda. A mis amigos y compañeros que se han ganado mi más sincero respeto, admiración, cariño y amistad.

A todos mis más sinceros agradecimientos.

Arnoldo:

Quiero dedicar este trabajo a mi familia, por acompañarme en cada una de las locuras que he emprendido y ser siempre mis más fervientes compañeros. "No hay palabras que puedan describir mi profundo agradecimiento hacia mis Padres Arnoldo y Nuria, quienes durante todos estos años confiaron en mí; comprendiendo mis ideales, por sus sabios consejos, por todo lo que me han dado en esta vida y por permitirme llegar hasta este momento tan importante logrando otra meta sacrificando el tiempo que no estuve con ellos". A mis hermanitas Amnielis y Arletis por siempre alegrarme la vida. A mi tutor que más que mi tutor Néstor ha sido mi compañero y poniendo su esfuerzo en cada una de las etapas de Trabajo. Benito y Eliza, quienes se convirtieron en mis segundos padres brindándome su ayuda incondicional. Danier (Manchita) por escucharme, soportarme y convertirse en mis mejores amigos, poniendo a prueba su capacidad y desempeño en el desarrollo de la solución. A Yordis y Oelys. A mis amigos y amigas de la UCI quienes se convirtieron en mi familia Universitaria Nelson, Edgar, Carlos R, Daniel, Alexander, Castillo, Carlos, Denis, Leo, Lamothe, Leosdeny, Javier, Natacha, Elena, Dainelis y a Rayza. La gente del grupo de 1ro, 2do, 3ro, 5to, de ERQ, los Búfalos de Sgto., de los diferentes por los que pasado Aptos, por ayudarme en cada momento y hacerme sentir como en casa cuando lo necesité. A mi Cielita Lidi, simplemente por ser como es y a toda su familia. Con todas sus manías y defectos, con todas sus virtudes y bellezas. Gracias por inspirar mi poesía, caminar a mi lado durante todo este tiempo y mostrarme con una sonrisa, que el AMOR de verdad puede existir.

...quienes indirectamente, o a propósito, trataron de impedir esta tesis y me hicieron redoblar esfuerzos.

DEDICATORIA

A black and white photograph of a fountain pen resting in an inkwell. The pen is positioned diagonally, with its nib pointing towards the upper right. The inkwell is a simple, rounded glass container with a lid that is slightly ajar. The background is a plain, light-colored surface, and the lighting creates soft shadows, highlighting the textures of the pen and the glass.

A Uds. que en sus hombros nos han llevado y han sido parte de cada uno de nuestros

logros y sacrificios, a Uds...

Nuestros Padres.

RESUMEN

Las tendencias actuales en el ambiente empresarial han supuesto para Cuba la necesidad de integrar la información y automatizar los procesos en las entidades. La Universidad de las Ciencias Informáticas ha asumido la construcción de un sistema de Planificación de Recursos Empresariales (ERP), que permita gestionar las operaciones de las empresas cubanas dedicadas a la producción de bienes o servicios. Este sistema tiene entre sus principales objetivos automatizar todo el proceso de planificación en las entidades cubanas.

En Cuba la base de toda planificación es la Dirección por Objetivo, siendo éste un proceso de planificación que se enfatiza en lo que se quiere lograr (objetivos) y en lo que se quiere hacer (estrategias). Posibilita el diseño de medidas de seguimiento y control que permiten la evaluación de los avances. El proceso centra su importancia en asegurar mediante su ejecución los objetivos principales y específicos trazados por las entidades del país.

En la actualidad no es posible lograr una optimización del proceso de gestión de planes estratégicos y por objetivo, imposibilitando la creación de indicaciones específicas para conformar los planes del año actual, siguiente y mensual. Tampoco existe la forma de controlar las órdenes, directivas y manuales, emitidas desde diferentes niveles en la empresa.

Es por ello que el presente trabajo está orientado a desarrollar módulos que satisfagan las necesidades que existen en las empresas cubanas para gestionar Factores que Influyen en el Plan y Configurar Perfil, todo ello dentro del proceso de Dirección por Objetivo. El desarrollo de este sistema se sostiene de la aplicación de metodologías y el uso de herramientas actuales que aseguran que su resultado sea un producto de software confiable, con un alto grado de calidad.

PALABRAS CLAVES

Automatizar, Planificación de Recursos Empresariales, Planificación, Dirección por Objetivo, planes estratégicos, Factores que Influyen en el Plan, Configurar un Perfil, software.

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 INTRODUCCIÓN	5
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	5
1.2.1 ¿Qué es un ERP?.....	5
1.2.2 ¿Qué significa Planificación?.....	5
1.2.3 ¿Qué es la Dirección por Objetivo?.....	6
1.3 TÉCNICAS Y TECNOLOGÍAS	7
1.3.1 Técnicas y tecnologías del lado del cliente	8
1.3.2 Técnicas y tecnologías del lado del servidor	9
1.3.3 Plataforma de desarrollo.....	10
1.4 HERRAMIENTAS.....	11
1.4.1 Herramientas CASE.....	11
1.4.2 Mozilla Firefox 3.5.....	12
1.4.3 Herramienta para programar en Java Script.....	12
1.4.4 Herramienta para programar en PHP	12
1.4.5 TortoiseSVN.....	12
1.5 MODELO DE DESARROLLO DE SOFTWARE	13
1.6 LENGUAJE DE MODELADO.....	14
1.6.1 UML.....	14
1.7 BPMN	14
1.8 SOLUCIÓN PROPUESTA	14
1.7 CONCLUSIONES	15
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	16
2.1 INTRODUCCIÓN	16
2.2 OBJETO DE ESTUDIO	16
2.2.1 Objeto de automatización	16
2.2.2 Propuesta de solución	16
2.3 MODELACIÓN DEL NEGOCIO	17
2.3.1 Diagrama de proceso de negocio.....	17
2.3.2 Descripción del proceso de negocio.....	18
2.3.3 Validación del proceso de negocio.....	19
2.3.4 Modelo conceptual.....	19
2.4 REQUISITOS	19
2.4.1 Lista de requisitos funcionales identificados	19
2.4.2 Especificación de Requisitos Funcionales	20
2.4.3 Validación de los Requisitos Funcionales.....	23
2.4.4 Requisitos No Funcionales.....	23

2.5	CONCLUSIONES	25
CAPÍTULO 3: MODELACIÓN DEL SISTEMA.....		26
3.1	INTRODUCCIÓN	26
3.2	ARQUITECTURA.....	26
3.2.1	<i>Arquitectura utilizada</i>	26
3.3	DISEÑO.....	31
3.3.1	<i>Diagrama de clases del diseño</i>	32
3.3.2	<i>Diagramas de secuencia</i>	32
3.3.3	<i>Métricas para validar el diseño</i>	33
3.4	CONCLUSIONES	35
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA DE LOS COMPONENTES GESTIONAR FIP Y CONFIGURACIÓN DE PERFIL.....		36
4.1	INTRODUCCIÓN	36
4.2	IMPLEMENTACIÓN	36
4.2.1	<i>Diagrama de componentes</i>	36
4.2.2	<i>Estrategias de integración</i>	37
4.2.3	<i>Estándares utilizados</i>	38
4.3	PRUEBAS DE SOFTWARE	38
4.3.1	<i>Tipos de pruebas</i>	39
4.4	CONCLUSIONES	44
CONCLUSIONES.....		45
RECOMENDACIONES.....		46
BIBLIOGRAFÍA.....		47
GLOSARIO DE TÉRMINOS.....		50
ANEXOS		53
	<i>Anexo 1</i>	53
	<i>Anexo 2</i>	54
	<i>Anexo 3</i>	55
	<i>Anexo 4</i>	63
	<i>Anexo 5</i>	64
	<i>Anexo 6</i>	65
	<i>Anexo 7</i>	68
	<i>Anexo 8</i>	70

INTRODUCCIÓN

La llegada del siglo XXI, ha propiciado el avance tecnológico en muchos países del mundo, entre ellos Cuba, país que se encuentra enfrascado en el perfeccionamiento del sector empresarial. Este sector se ve afectado por diversos factores que limitan su desarrollo, entre ellos está la planificación, pilar fundamental para el crecimiento empresarial. La planificación se refiere a las acciones llevadas a cabo para realizar planes y proyectos de diferentes índoles.

En el proceso de planificación se definen Estrategias, Políticas y Objetivos de la entidad, estos últimos quedan establecidos en el plan junto a los procedimientos adecuados para alcanzarlos. Este proceso proporciona que los miembros de la entidad desempeñen actividades y tomen decisiones congruentes con los objetivos y procedimientos escogidos. De manera general la planificación es la parte que opera la ejecución directa de los planes, los que pueden ser a corto, mediano y largo plazo, según la amplitud y magnitud del proceso que se va a desarrollar.

En Cuba la base de toda planificación es la Dirección por Objetivo, siendo éste un proceso de planificación que se enfatiza en lo que se quiere lograr (objetivos) y en lo que se quiere hacer (estrategias). Es decir, busca centrarse en aquellos objetivos factibles que permitan obtener el éxito. Define para ello las acciones a realizar y los problemas a resolver, priorizando soluciones, estableciendo recursos y responsabilidades. Posibilita el diseño de medidas de seguimiento y control que permiten la evaluación de los avances. El proceso centra su importancia en asegurar mediante su ejecución los objetivos principales y específicos trazados por las entidades del país.

Para gestionar todo este proceso de Dirección por Objetivo, el país cuenta con un sistema de gran atraso en el mundo tecnológico, como es el caso del PETRAB (sistema MS DOS). El uso de este sistema propicia la insatisfacción por parte de las entidades cubanas, pues provoca que la gestión de la información correspondiente a estos procesos no sea la mejor. El PETRAB no permite que se lleve a cabo un buen proceso de toma de decisiones. Todo ello influye negativamente en el proceso de Dirección por Objetivo.

A raíz de esta situación y la necesidad de planificar, puntualizar y controlar, así como recuperar información, Cuba decidió la creación de un Sistema de Planificación de Recursos Empresariales (ERP). El ERP es un sistema de información gerencial que integra y maneja muchos de los negocios asociados con las operaciones de producción. Este sistema está basado en el ofrecimiento de una solución completa que permite a las empresas evaluar, implementar y administrar con mayor facilidad su negocio. Esta notable tarea tiene lugar en la Universidad de la Ciencias Informáticas, constituyendo un reto para la naciente industria cubana del software. Al sistema lo integran entre otros Sub-sistemas el de Planificación y dentro de él está el Sub-sistema de Dirección por Objetivo, con el que se pretende perfeccionar todo el proceso de Dirección por Objetivo en las entidades del país. En la actualidad no es posible lograr una optimización del proceso de gestión de planes estratégicos y por objetivo, imposibilitando la creación de indicaciones específicas para conformar los planes del año actual, siguiente y mensual. Tampoco existe la forma de controlar las órdenes, directivas y manuales, emitidas desde diferentes niveles en la empresa.

A estos problemas se suma el descontrol sobre la información generada por Factores que Influyen en el Plan y la necesidad de personalización de funcionalidades y de usabilidad dentro del proceso de Dirección por Objetivo. Ante esta situación el **problema a resolver** radica en: ¿Cómo controlar y darle seguimiento a factores que influyen en el plan, lograr la personalización de funcionalidades y una mayor usabilidad en el Sub-sistema de Dirección por Objetivo? Tomando como **objeto de estudio** el proceso de desarrollo de Dirección por Objetivo en las entidades del país y los sistemas que lo implementan. Definiendo como **campo de acción** la gestión de los procesos para controlar y darle seguimiento a los factores que influyen en el plan y la personalización de funcionalidades en el Sub-sistema de Dirección por Objetivo. Teniendo en cuenta lo anterior se define como **objetivo general**:

- ✓ Desarrollar los componentes para el Sub-Sistema Dirección por Objetivo que faciliten el flujo de la información, el control y seguimiento de la misma así como la configuración y personalización de funcionalidades según la necesidad del usuario.

Para dar cumplimiento a este objetivo se han trazado los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico de la investigación.
- ✓ Realizar el modelado del negocio para identificar las actividades que requieren informatización.
- ✓ Obtener los requisitos necesarios para el funcionamiento del sistema.
- ✓ Obtener el componente FIP y el componente Configurar Perfil.

- ✓ Realizar pruebas a los dos componentes desarrollados, garantizando su calidad.

Teniendo en cuenta lo anteriormente planteado, se define como **idea a defender**: El desarrollo de los componentes FIP y Configurar Perfil, posibilita controlar y darle seguimiento a factores que influyen en el plan dentro del proceso de Dirección por Objetivo en las entidades del país, así como lograr la personalización de funcionalidades según la necesidad del usuario y una mayor usabilidad en el Sub-Sistema Dirección por Objetivo.

Para desarrollar cada uno de estos objetivos específicos se planificaron las siguientes **tareas que tributan a la investigación**:

- ✓ Consulta de fuentes bibliográficas para comprender los conceptos relacionados con el tema a tratar.
- ✓ Investigación sobre los sistemas utilizados en Cuba y el mundo que implementan la Dirección por Objetivo.
- ✓ Selección de las herramientas que se utilizarán en la implementación del sistema informático.
- ✓ Investigación sobre cómo ocurre y en qué consiste el proceso de Dirección por Objetivo.
- ✓ Investigación sobre cómo ocurre y en qué consiste el proceso de gestión, control y seguimiento de los factores que influyen en el plan así como la personalización de funcionalidades.
- ✓ Descripción de todo el proceso de negocio de gestión, seguimiento y control de los factores que influyen en el plan y de la personalización de funcionalidades, este último dentro del proceso de configuración para la elaboración de planes.
- ✓ Identificación y especificación de requisitos funcionales para el proceso de negocio de gestión, seguimiento y control de los factores que influyen en el plan y de la personalización de funcionalidades.
- ✓ Análisis y diseño de la Gestión de Factores que Influyen en el Plan.
- ✓ Análisis y diseño de la Configuración del Perfil.
- ✓ Implementación de los componentes para la Gestión de Factores que Influyen en el Plan y Configuración del Perfil.
- ✓ Realización de pruebas de validación a los dos componentes desarrollados.
- ✓ Documentación del trabajo completo, propiciando el desarrollo de nuevas versiones.

Valorando como **posibles resultados**:

- ✓ Componentes del Sub-Sistema Dirección por Objetivo para gestionar y personalizar funcionalidades, así como para controlar el seguimiento y cumplimiento de órdenes, indicaciones, resoluciones y acuerdos.

El presente trabajo está estructurado por cuatro capítulos.

Capítulo #1: Fundamentación Teórica, tecnologías, técnicas, metodologías y herramientas utilizadas:

En este capítulo se presentan temas que integran la fundamentación teórica de la investigación a realizar, o sea, el estado del arte, para lo se hace necesario el análisis y revisión de las fuentes bibliográficas relacionadas con el tema en cuestión. Se hace un análisis sobre las principales tecnologías, metodologías y herramientas que se utilizan en la actualidad para este tipo de aplicación.

Capítulo #2: Características del sistema:

En este capítulo se describen las principales características que debe tener el sistema, mediante el modelamiento de los procesos negocio y la especificación de los requisitos de software. Se desarrollan parte de los artefactos propuestos en la fase de Modelación definida en la metodología de software aplicada.

Capítulo #3: Modelación del sistema:

En el presente capítulo se da continuación a la fase de Modelación del sistema. Se realiza el diseño de los componentes a implementar y se hace referencia a la arquitectura que sustenta los módulos de software. Se da culminación a la fase de Modelación, obteniendo los artefactos restantes de la fase.

Capítulo #4: Implementación y prueba de los componentes FIP y Configurar de Perfil:

En este capítulo se abarca todo lo concerniente con la implementación del sistema. Se realizan además pruebas a la aplicación, en aras de garantizar su calidad.

CAPÍTULO 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se exponen los conceptos vinculados al problema tratado, y se ofrece información acerca del proceso de Dirección por Objetivo. Se hace referencia a las principales características de algunas soluciones de software relacionadas con el proceso de Dirección por Objetivo en las entidades cubanas. Por último, se ofrece una visión de la metodología, tecnologías y las herramientas que se utilizarán durante el trabajo, lo que conlleva a una propuesta de la solución.

1.2 Conceptos asociados al dominio del problema

A continuación se tratarán conceptos relacionados con el problema en cuestión, que resultan de gran importancia para lograr la comprensión del mismo.

1.2.1 ¿Qué es un ERP?

La traducción literal del término ERP es Planificador de Recursos Empresariales. Se trata de un paquete software o aplicación, generalmente multi-modular, que ofrece soluciones integradas diseñadas para dar soporte a todos los procesos de negocio de la empresa. Los sistemas ERP típicamente manejan la producción, logística, distribución, inventario, envíos, facturas y contabilidad de la entidad. Estos sistemas son una solución robusta para aquellas empresas que buscan una solución universal a la centralización de su información.

1.2.2 ¿Qué significa Planificación?

La planificación determina por anticipado cuáles son los objetivos que deben cumplirse y qué debe hacerse para alcanzarlos; por tanto, es un modelo teórico para actuar en el futuro. Comienza por establecer los objetivos y detallar los planes necesarios para alcanzarlos de la mejor manera posible.

De manera general el fin de planificar es eliminar al máximo los imprevistos, es decir, lograr los objetivos y metas de la empresa con el máximo de ventajas, el mínimo de desventajas, el mínimo de riesgos. Posibilitando optimizar al máximo el uso de los recursos humanos, materiales y financieros con que cuenta la empresa.

1.2.2.1 Tipos de planificación

Existen diversas clasificaciones acerca de la planificación. Entre las más usadas están: la planificación estratégica y la planificación operativa.

La planificación estratégica es un proceso de evaluación sistemática de la naturaleza de un negocio. Define los objetivos a largo plazo, identificando metas y objetivos cuantitativos. Desarrolla estrategias para alcanzar dichos objetivos y permite localizar recursos para llevar a cabo dichas estrategias. Mientras que la planificación operativa, consiste en formular planes a corto plazo que pongan de relieve las diversas partes de la organización. Muestra cómo se pueden aplicar los planes estratégicos en el quehacer diario. Los planes estratégicos y los planes operativos están vinculados a la definición de la misión de una organización, la meta general que justifica la existencia de una organización. Los planes estratégicos difieren de los planes operativos en cuanto a su horizonte de tiempo, alcance y grado de detalles.

1.2.3 ¿Qué es la Dirección por Objetivo?

El término **objetivo** proviene de dos raíces: “jactum” que significa lanzado y “ob” hacia. De aquí que, en una primera aproximación, podamos decir que es aquello que se *lanza hacia el logro de un fin concreto*. Un propósito o meta que se propone a cumplir en tiempo definido.

La definición de objetivos es uno de los pilares en los que se apoya la estrategia empresarial. Son las metas intentadas que prescriben o establecen un determinado criterio y señalan dirección a los esfuerzos del administrador. Los objetivos dentro del proceso de planificación sirven de enlace o vínculo entre planificación y ejecución, concretando las categorías básicas en resultados específicos a alcanzar por las organizaciones.

La **Dirección por Objetivo** es un enfoque gerencial, mediante el cual el sujeto y el objeto de dirección, conjuntamente, definen y estructuran los objetivos del sistema. Establecen además las áreas de responsabilidad de cada cual, como base para la medición y evaluación de los resultados a alcanzar. Es una forma particular de funcionamiento en torno a las tareas directivas. Es un sistema de dirección cuyo propósito básico es lograr una adecuada coordinación de personas, recursos y mecanismos de una organización para obtener resultados relevantes sobre la base de la amplia y efectiva participación.

1.2.3.1 Factores que Influyen en el Plan (FIP)

En la concepción o ejecución de un plan existen eventos que influyen notablemente, pues tributan a generar actividades que van contempladas dentro del plan. A estos eventos se les conoce como FIP: Factores Internos que influyen en el Plan. Están categorizados en: órdenes, indicaciones, resoluciones y acuerdos que son generadas desde un nivel superior a sus subordinados o en el mismo nivel en dependencia de la necesidad de dicho nivel. Cada FIP tiene asociado actividades y/u objetivos y a su vez deben tributar a un determinado elemento de la planificación, de ahí depende su estado de cumplimiento. El control y seguimiento de los FIP dentro de la planificación tiene gran importancia, pues pueden provocar un desorden o caos total en dependencia de los elementos de la planificación que se vean afectados por el surgimiento de estos si no se sabe qué generan y a qué tributan.

1.2.3.2 Sistema que implementa la Dirección por Objetivo

En Cuba el sistema que se utiliza para gestionar el proceso de Dirección por Objetivo en las entidades cubanas es el PETRAB. Este sistema es arcaico, corre sobre el sistema operativo MSDOS, además no permite controlar y darle seguimiento a elementos de la planificación como por ejemplo: actividades, objetivos, órdenes, indicaciones y resoluciones que se generan durante el proceso de planificación y que resulta necesario que se contemplen en cada plan de la entidad. Es imprescindible la creación de una aplicación que a diferencia del PETRAB presente una interfaz amigable para el usuario, permitiéndole al mismo configurar su perfil, y darle la posibilidad de que establezca sus propios objetivos o actividades trazadas para cumplir con la planificación.

A nivel internacional no se conoce ningún sistema que implemente el proceso de Dirección por Objetivo. En las empresas se conoce como un proceso de planificación o sea una mejor manera de dirigir el control de los recursos materiales y financieros de la entidad; una manera de establecer un control sobre el proceso de dirección.

1.3 Técnicas y tecnologías

Cada una de las técnicas y tecnologías que se mencionan a continuación, fueron definidas por la dirección del proyecto ERP.

1.3.1 Técnicas y tecnologías del lado del cliente

1.3.1.1 Java Script

Java Script es un lenguaje basado en objetos, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento.

Un **script** (cuya traducción literal es *guión*) o archivo de órdenes o archivo de procesamiento por lotes es un programa usualmente simple, que generalmente se almacena en un archivo de texto plano.

1.3.1.2 AJAX

Ajax, acrónimo de *Asynchronous Java Script And XML* (Java Script asíncrono y XML). Técnica de desarrollo web para crear aplicaciones interactivas mediante la combinación de tres tecnologías ya existentes (Factoría de Internet S.L, 2010):

- ✓ HTML (o XHTML) y Hojas de Estilo en Cascada (CSS) para presentar la información;
- ✓ Document Object Model (DOM) y Java Script, para interactuar dinámicamente con los datos, y
- ✓ XML y XSLT, para intercambiar y manipular datos de manera desincronizada con un servidor web.

La ayuda más potente que ofrece AJAX es el poder hacer consultas asíncronas al servidor sin necesidad de recargar la página.

1.3.1.3 ExtJS 2.2

ExtJS 2.2 es un framework completo y extremadamente avanzado. Este framework está basado completamente en la programación Orientada a Objeto. Cada objeto contiene lo típico: propiedades, métodos, eventos, etc. ExtJS 2.2 basa toda su funcionalidad en Java Script a través de librerías. Así, en tiempo de ejecución carga y crea todos los objetos HTML a través del uso intenso de DOM. Los datos son obtenidos con mucho AJAX a través de XML.

Este es el framework que se utilizará para gestionar la presentación de la aplicación.

1.3.1.4 Subversion

Subversión es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos (Attribution-NonCommercial-ShareAlike 2.5). Es un sistema de control de versiones que permite administrar las versiones (valga la redundancia) de los archivos de un proyecto. Promueve con esto la colaboración entre los miembros de un equipo de trabajo y, de una manera muy cómoda, coordina las tareas entre ellos.

1.3.2 Técnicas y tecnologías del lado del servidor

1.3.2.1 PHP v5.2 o superior

PHP (Hipertext Preprocesores) un lenguaje de programación interpretado, completamente orientado al desarrollo de aplicaciones webs dinámicas con acceso a información almacenada en una base de datos. Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma. El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

1.3.2.2 Servidor web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo con los comandos solicitados. En este punto es necesario aclarar lo siguiente: mientras que comúnmente se utiliza la palabra servidor para referirse a una computadora con un software servidor instalado, en estricto rigor un servidor es el software que permite la realización de las funciones descritas. (Misrespuestas)

1.3.2.2.1 Servidor web Apache 2.0 o superior

Es un servidor web que corre en una multitud de sistemas operativos, lo que lo hace prácticamente universal. Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es

importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si se quiere ver que es lo que se está instalando como servidor, se puede saber. Tiene todo el soporte que se necesita para tener páginas dinámicas. Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Apache admite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

1.3.2.3 Sistema Gestor de Base de Datos

Los sistemas de gestión de bases de datos o SGBD (en inglés database management system, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se trata de un software de propósito general que ayuda en los procesos de definición (especificación de los tipos de datos, las estructuras y restricciones), construcción (almacenar los datos concretos en dispositivos físicos) y manipulación (funciones como consultas, actualizaciones, inserciones, borrados y modificaciones de los datos) de una base de datos. (Alvarez)

1.3.2.3.1 PostgreSQL v8.3 o superior

PostgreSQL es un sistema de gestión de bases de datos relacional orientada a objetos, libre y multiplataforma, publicado bajo la licencia BSD (*Berkeley Software Distribution*). **PostgreSQL** es un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. PostgreSQL tiene la extraordinaria potencialidad de permitir que mientras un proceso escribe en una tabla, otros accedan a la misma sin necesidad de bloqueos esto es posible gracias a un sistema denominado Acceso Concurrente Multiversión.

1.3.3 Plataforma de desarrollo

1.3.3.1 Zend Framework

Zend Framework (ZF) es un framework (marco de trabajo) de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. Este framework basa su funcionamiento en el patrón MVC

(explicado posteriormente). Específicamente de este framework se utiliza el Zend View (una clase del Zend framework) para cargar las páginas phtml y para la inyección de dependencias en la controladora se utiliza la clase ZendExt_Controller_Secure.

1.3.3.2 Framework Doctrine

Es un framework utilizado para el mapeo de objetos relacionales. Crea las clases bases y las relaciones entre las tablas. Es un mapeador objeto relacional (ORM) para PHP que se encuentra en la parte superior de una capa de abstracción de base de datos de gran alcance. Una de sus principales características es la posibilidad de escribir consultas de base de datos en un objeto de propiedad orientada al dialecto SQL llamado Doctrine (Lenguaje de Consulta Doctrine DQL).

1.3.3.3 Sistema Operativo Linux

GNU/Linux (Linux) es uno de los términos empleados para referirse al sistema operativo libre similar a Unix que usualmente utiliza herramientas de sistema GNU. Es un sistema operativo gratuito, multitarea, multiusuario, multiplataforma y multiprocesador. Como sistema operativo, Linux es muy eficiente y tiene un excelente diseño. El sistema lo forman el núcleo del sistema (kernel) más un gran número de programas / librerías que hacen posible su utilización.

1.4 Herramientas

1.4.1 Herramientas CASE

CASE es una sigla, que corresponde a las iniciales de: **C**omputer **A**ided **S**oftware **E**ngineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación. Es un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Proporcionan un conjunto de herramientas semiautomatizadas y automatizadas para apoyar una o más fases del ciclo de vida del desarrollo de software.

1.4.1.1 Visual Paradigm 6.0

Es una herramienta CASE que utiliza “UML”: como lenguaje de modelado. Es considerada como muy completa y fácil de usar, con soporte multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. El software que modela del UML le ayuda a construir aplicaciones

con calidad más rápidas, mejor y en más abajo costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

1.4.2 Mozilla Firefox 3.5

Mozilla Firefox es un navegador web libre, multiplataforma y está disponible en varias versiones de Microsoft Windows, GNU/Linux y algunos sistemas basados en Unix. Incluye navegación por pestañas, corrector ortográfico, búsqueda progresiva, marcadores dinámicos, un administrador de descargas y un sistema de búsqueda integrado.

1.4.3 Herramienta para programar en Java Script

1.4.3.1 Aptana

Es un entorno de desarrollo dirigido hacia las aplicaciones web escritas en Ajax/Java Script, permite editar fácilmente HTML, CSS y Java Script. Aptana está basado en el conocido entorno de desarrollo Eclipse (IDE = Integrated Development Environment), también Open Source. Pero mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo web.

1.4.4 Herramienta para programar en PHP

1.4.4.1 Zend Studio Neon

Zend Studio Neon es para desarrollar aplicaciones web y servicios web con PHP, presenta un completo entorno de desarrollo integrado para dicho lenguaje de programación. Está escrito en Java, y está disponible para plataformas como Microsoft Windows y GNU/Linux. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

1.4.5 TortoiseSVN

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. TortoiseSVN maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo

que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. (Historia de TortoiseSVN)

1.5 Modelo de Desarrollo de Software

Por las necesidades que tiene el proyecto ERP de ser un software robusto, predecible y de fácil mantenimiento, la dirección del proyecto constituyó un nuevo modelo de desarrollo de software.

Se compone de los modelos basado en Componentes, el Iterativo y el Incremental. El modelo de desarrollo de software basado en componentes, proporciona la reutilización de software y como su nombre lo indica propicia la calidad de la aplicación basada en componentes. Lo que permite que sean probados de forma independiente y que sean gestionados por el desarrollador sin afectar otras partes del sistema. Mientras que en el Modelo Iterativo e Incremental el ciclo de vida se compone de iteraciones. En cada iteración es integrado hasta obtener el producto completo. Obteniéndose como resultado en cada iteración un incremento. Sus características principales son:

Centrado en la arquitectura: pues 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. La arquitectura orienta las prioridades del desarrollo y resuelve las necesidades tecnológicas y de soporte para el desarrollo.

Orientado a componentes: ya que las iteraciones son orientadas por el nivel de significancia arquitectónicas 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: las iteraciones son planificadas y coordinadas con el equipo de arquitectura, los clientes y la alta gerencia. Cada iteración constituye el desarrollo de componentes, los cuales son integrados al término de la integración, permitiendo de esta manera la evolución incremental del producto.

Ágil y adaptable al cambio: pues el desarrollo de las partes formaliza solamente las características principales de la solución, priorizando los talleres y las comunicaciones entre las personas. Los clientes y funcionales están involucrados en el proyecto y poseen parte de las responsabilidades del éxito del

mismo. Los cambios son conciliados semanalmente, discutidos y aprobados. (CENTRO DE SOLUCIONES DE GESTIÓN)

1.6 Lenguaje de modelado

1.6.1 UML

El Lenguaje de Modelado Unificado (Unified Modeling Language) es una especificación de notación orientada a objetos. Es utilizado con el fin de especificar y documentar un sistema de software, de un modo estándar. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto.

1.7 BPMN

BPMN es el acrónimo de **B**usiness **P**rocess **M**odeling **N**otation y no es otra cosa que una notación gráfica estandarizada para el modelado de los procesos de negocio. Su principal objetivo es: Resolver las dificultades de comunicación que tiene el lenguaje común. (Madrid, 2010)

- ✓ Proporciona un método normalizado para representar procesos de negocio
- ✓ Facilita su entendimiento debido a la poca complejidad de su notación
- ✓ Proporciona un lenguaje común entre los usuarios de negocio y los técnicos
- ✓ Facilita la diagramación de los procesos de negocio.
- ✓ Proporciona una mayor visibilidad de los procesos de las empresas.
- ✓ Mejora la flexibilidad y agilidad para la adaptación al cambio y posibilita integrar la información del negocio dispersa en diferentes sistemas.

1.8 Solución propuesta

La solución que se propone estará dada por:

- ✓ Navegador: Mozilla Firefox
- ✓ Lenguaje de programación del lado del servidor: PHP v5.2 o superior
- ✓ Lenguaje de programación del lado del cliente: XML, Java Script
- ✓ Gestor de base datos: PostgreSQL v8.3 o superior

- ✓ Tipo de servidor: Apache 2.0
- ✓ Modelo de Desarrollo de Software: Modelo de Desarrollo de Software UCID-ERP.
- ✓ Herramienta CASE para el modelado: Visual Paradigm 6.0
- ✓ Lenguaje de modelado: UML, BPM.
- ✓ Sistema Operativo: Linux

1.7 Conclusiones

Durante el desarrollo de todo el capítulo, se abordaron conceptos relacionados con el proceso de Dirección por Objetivo, en aras de lograr una comprensión óptima del objeto de estudio definido al inicio. Se realizó un estudio de las técnicas, tecnologías y herramientas indispensables para desarrollar el software y quedó plasmada con claridad la metodología a utilizar en todo el proceso de software definido por la dirección del proyecto.

CAPÍTULO 2: Características del Sistema

2.1 Introducción

En este capítulo se describen las características que debe tener el sistema, a través de los requisitos funcionales y no funcionales. Se realizará el modelamiento del negocio, mediante la metodología y el lenguaje de modelado propuesto. Además, se presentan los artefactos que se obtienen durante el flujo de trabajo.

2.2 Objeto de estudio

2.2.1 Objeto de automatización

El proceso que será objeto de automatización dentro del proceso de Dirección por Objetivo es la Gestión de FIP.

Gestionar FIP: Durante el proceso de Dirección por Objetivo se hace necesario que se contemplen en el plan, factores que lleven asociados actividades u objetivos por los distintos niveles de cada organización. Los mismos pueden referirse a la emisión de una orden, indicación, resolución o acuerdo, por lo que se hace necesario establecer un control del cumplimiento sobre cada uno de estos elementos de la planificación. Emitir un FIP puede estar dado por el cumplimiento de otro emitido previamente por el organismo superior o por una necesidad interna del organismo. En caso de que sea creado para darle cumplimiento a la emisión de otro FIP, se recibe un documento de FIP del nivel superior que contiene uno o varios FIP, dicho documento debe ser estudiado. Para dar cumplimiento al FIP emitido se formulan nuevos objetivos y/o nuevas actividades los cuales deben ser incluidos en el plan, acción mediante la que se establece un plan. Posteriormente se envía el documento FIP al nivel inmediato inferior. Este documento es enviado a los subordinados con los nuevos cambios para que actualicen los planes correspondientes. En caso de que el FIP sea emitido por una necesidad interna del organismo, el documento es elaborado en el propio nivel y luego sigue el mismo proceso anterior.

2.2.2 Propuesta de solución

Asumiendo que no existe un sistema capaz de gestionar todos los factores que intervienen en la planificación dentro del proceso de Dirección por Objetivo, se pretende desarrollar una aplicación

encaminada a resolver satisfactoriamente este problema. Dicha aplicación debe posibilitar el control y seguimiento de órdenes, indicaciones, acuerdos y resoluciones emitidas por los niveles superiores de cada entidad. El sistema brindará la posibilidad de que los usuarios puedan crear sus propios objetivos y/o actividades, a partir de la emisión de un FIP, plasmándolas posteriormente en el plan. Además, contará con una interfaz amigable para el usuario, de manera que le sea fácil y seguro el manejo de su información personal. Se persigue con la misma proporcionar al usuario la posibilidad de configurar su entorno de trabajo, y de gestionar su propio calendario, el que le ayudará a cumplir en tiempo con su planificación. La aplicación constará de un sistema de seguridad fiable, capaz de gestionar los permisos necesarios a las personas autorizadas para interactuar con la información disponible. Esto garantiza la integridad de los datos.

2.3 Modelación del negocio

En la modelación del negocio se describen los procesos de negocio, mediante el artefacto Descripción de Procesos de Negocio. El mismo tiene como propósito lograr una mejor comprensión del problema que el software tiene que resolver. En este capítulo se expone además un Glosario de términos, donde se tratan los conceptos relacionados con el proceso de negocio desarrollado.

2.3.1 Diagrama de proceso de negocio

Un proceso de negocio es un conjunto de actividades relacionadas que permiten crear un producto o servicio final a través de la transformación de uno o varios productos o servicios iniciales. (Real)

Un diagrama de proceso de negocio es una representación gráfica del proceso de negocio, en el caso del ERP, utilizando la Notación para la Gestión de Procesos de Negocio (Business Process Management Notation o BPMN)

A continuación se muestra el diagrama de proceso de negocio para el proceso Gestionar FIP. El mismo muestra cómo surge un FIP, quién lo crea y se muestra todo el proceso de creación hasta que finalmente quede establecido dentro de un plan.

2.3.1.1 Gestionar FIP

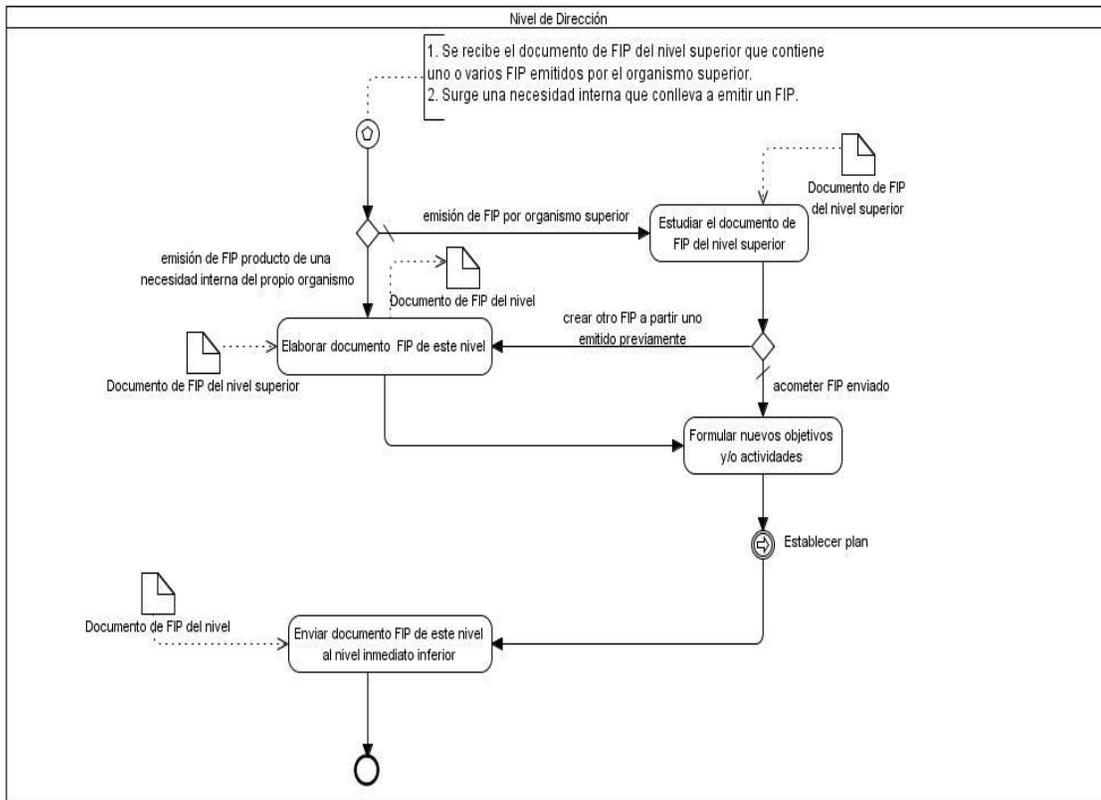


Figura 1 Diagrama del proceso de negocio Gestionar FIP

2.3.2 Descripción del proceso de negocio

Durante la Descripción del proceso de negocio se documenta todo el proceso de negocio del módulo correspondiente. Esta descripción permite una mejor comprensión de todo el proceso, pues describe el objetivo que persigue el proceso, los eventos que lo generan, las condiciones que deben estar creadas para que se genere y describe detalladamente el flujo de eventos que dan lugar a todo el proceso. Todos estos aspectos van recogidos en una plantilla de Descripción del proceso de negocio.

Para ver la plantilla de Descripción del proceso de negocio correspondiente a Gestionar FIP referirse a Anexo 1.

2.3.3 Validación del proceso de negocio

El proceso de negocio desarrollado queda validado por la dirección del proyecto. Como constancia queda emitido un documento donde se valida el proceso de negocio descrito. (Centro de Soluciones de Gestión)

2.3.4 Modelo conceptual

El modelo conceptual, permite dominar los principales conceptos con los que se relaciona el módulo a desarrollar y establece las relaciones que existen entre cada uno de ellos. Todos estos conceptos son modelados mediante tablas. Para elaborar el modelo conceptual se tiene como entrada la Descripción del proceso de negocio y el Glosario de términos. El modelo conceptual incluye un Diccionario de Datos, en él se describen cada uno de los atributos de cada tabla modelada. (Gestión, 2009)

Para ver el modelo conceptual concerniente a Gestionar FIP y Configurar Perfil ir a [Anexo 2](#).

2.4 Requisitos

Condición o capacidad que debe cumplir o poseer un sistema o componente del sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto. Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. Es de gran importancia para los desarrolladores y clientes, pues es una descripción completa del comportamiento del sistema a desarrollar.

En el presente trabajo los requisitos identificados se agrupan en paquetes de requisitos.

2.4.1 Lista de requisitos funcionales identificados

Gestionar FIP

- ✓ Adicionar FIP
- ✓ Modificar FIP
- ✓ Eliminar FIP
- ✓ Listar FIP
- ✓ Buscar FIP
- ✓ Realizar búsqueda avanzada de FIP

✓ Desactivar FIP

✓ Activar FIP

Gestionar atributos dinámicos

✓ Adicionar atributo dinámico

✓ Modificar atributo dinámico

✓ Eliminar atributos dinámicos

✓ Desactivar atributo dinámico

✓ Activar atributos dinámicos

Configurar calendario

✓ Adicionar calendario

✓ Modificar calendario

✓ Eliminar calendario

✓ Desactivar calendario

✓ Activar calendario

2.4.2 Especificación de Requisitos Funcionales

Para la especificación de requisitos se utiliza una plantilla. Dicha plantilla cuenta con secciones para describir claramente cada requisito, lo que debe suceder para que se ejecute la funcionalidad en el sistema. Describe además el flujo de eventos que da lugar a todo el proceso de ejecución del requisito.

A continuación se exponen la Especificación de requisitos correspondiente a cada requisito identificado.

2.4.2.1 Especificaciones de los Requisitos Funcionales identificados

Precondiciones	N/A.
Flujo de eventos	
Flujo básico	
1.	Se introducen los datos del FIP: Denominación Descripción Fecha de inicio

	Fecha de fin	
	Número del FIP	
	Emita	
	Categoría del FIP	
	Nivel de acceso	
	Origen	
2.	El sistema permite adicionar los involucrados y el responsable que pertenecerán al FIP.	
3.	El sistema permite seleccionar la opción de llevar el historial del FIP.	
4.	El sistema permite adicionar los indicadores que posibilitarán determinar el estado de cumplimiento del FIP.	
5.	El sistema permite seleccionar los elementos de la planificación con los cuales estará relacionado el FIP.	
6.	El sistema permite adicionar observaciones acerca del plan.	
7.	El sistema valida (ver validación 1) los datos introducidos.	
8.	Si los datos son correctos el sistema los registra.	
9.	El sistema confirma el registro de los datos.	
10.	Concluye el requisito.	
Pos-condiciones		
1.	Se registró en el sistema un nuevo FIP.	
Flujos alternativos		
Flujo alternativo 10.a Información errónea		
1	El sistema señala los datos erróneos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 9 del flujo básico.	
Pos-condiciones		
1	N/A.	
Flujo alternativo 10.b Información incompleta		
1	El sistema señala los datos vacíos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 9 del flujo básico.	
Pos-condiciones		
1	N/A.	
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se registran los datos.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual CSG-ERP-N- DPO-i2201.	
Relaciones	Requisitos Incluidos	<p>El sistema permite adicionar los indicadores que posibilitarán determinar el estado de cumplimiento del FIP: Adicionar indicador, en la agrupación Gestionar indicadores.</p> <p>El sistema permite adicionar observaciones acerca del FIP: Adicionar observaciones, en la agrupación Gestionar observaciones.</p> <p>El sistema permite adicionar documentos anexos al FIP: Adicionar anexos, en la agrupación Gestionar anexos.</p> <p>El sistema permite seleccionar los elementos de la planificación con los cuales estará relacionado el FIP: Adicionar relaciones entre elementos de la planificación, en la agrupación Gestionar relaciones entre elementos de la planificación.</p> <p>El sistema permite seleccionar los elementos de la planificación con los cuales estará relacionado el FIP: Eliminar relaciones entre elementos de la planificación, en la agrupación Gestionar relaciones entre elementos de la planificación.</p>

	Extensiones	N/A.
Conceptos	FIP	Visibles en la interfaz: Denominación Descripción Fecha de inicio Fecha de fin Fecha en que se cumplió Número del FIP Emite Por ciento de cumplimiento Activo Privado Utilizados internamente: N/A.
	Elemento de la planificación	Visibles en la interfaz: Denominación Descripción Fecha de inicio Fecha de fin Por ciento de cumplimiento Estado de cumplimiento Activo Utilizados internamente: N/A.
	Observación	Visibles en la interfaz: Autor Fecha Asunto Texto Utilizados internamente: N/A.
	Involucrado	Visibles en la interfaz: N/A. Utilizados internamente: Tipo Nivel
	Anexo	Visibles en la interfaz: Denominación Descripción Peso Tipo de archivo Utilizados internamente: N/A
	Indicador	Visibles en la interfaz: Estado de cumplimiento Utilizados internamente: N/A.
	Rango	Visibles en la interfaz: Límite inferior Límite superior Utilizados internamente: N/A.

Historial	Visibles en la interfaz: Acción Fecha Autor Utilizados internamente: N/A.
Requisitos especiales	N/A.
Asuntos pendientes	N/A.

Figura 2 Especificación de requisito Adicionar FIP

Las tablas que describen la especificación de cada uno de los requisitos restantes se encuentran en el Anexo 3.

2.4.3 Validación de los Requisitos Funcionales

Los requisitos funcionales descritos quedan aceptados y validados por la dirección del proyecto. Como constancia queda emitido un documento de Aceptación de Requisitos. (Centro de Soluciones de Gestión)

2.4.4 Requisitos No Funcionales

Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. Si se conocen las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable puede ser el software, un producto será bien aceptado.

Los requisitos no funcionales con los que cuenta la aplicación son los mismos establecidos para el producto CEDRUX. (UCID)

2.4.4.1 Prototipo de interfaz de usuario para Gestionar FIP

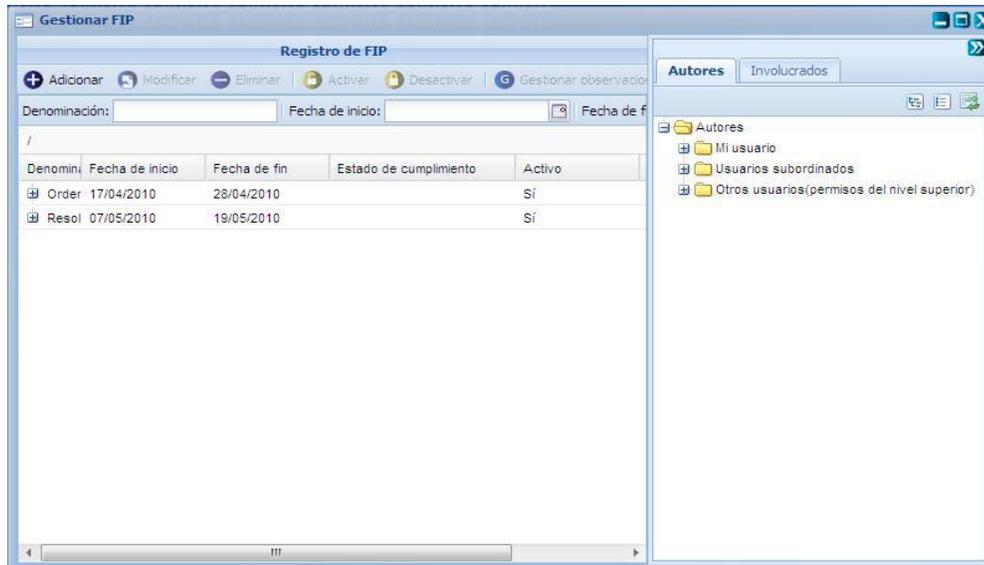


Figura 3 Prototipo de interfaz de usuario para Gestionar FIP

2.4.4.2 Prototipo de interfaz de usuario para Configurar Perfil

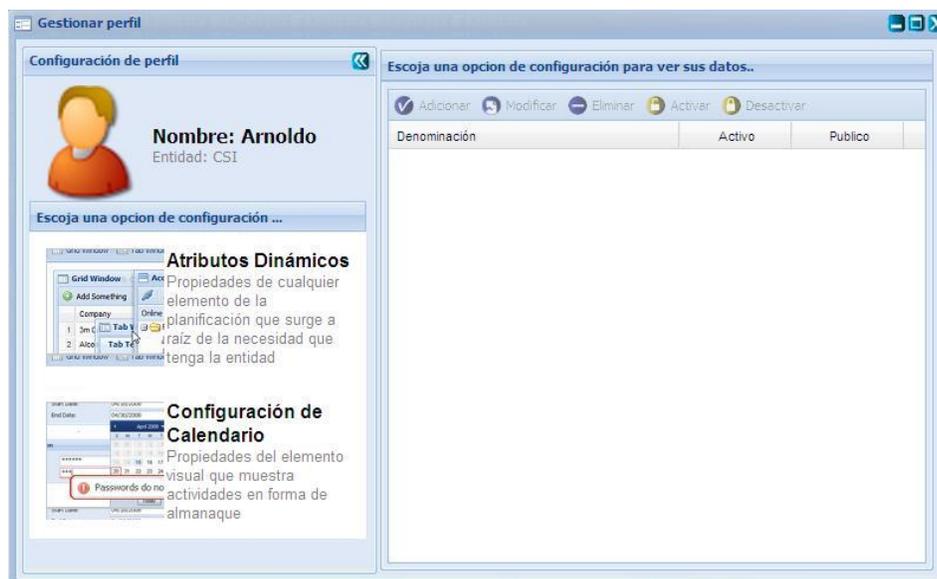


Figura 4 Prototipo de interfaz de usuario para Configurar Perfil

2.5 Conclusiones

A modo de conclusión quedan elaborados parte de los artefactos establecidos durante la fase de Modelación como: Descripción del proceso de negocio, Modelo conceptual, Glosario de términos, Lista de los requisitos funcionales y Especificación de los requisitos de software y hardware. Mediante estos artefactos se realizó la descripción de todo el proceso de negocio, así como la especificación de todos los requisitos identificados de acuerdo a las funcionalidades que debe cumplir el sistema. Quedó claro cómo se desarrolla el proceso de Gestionar FIP. De manera resumida se establece la propuesta de solución que se desea desarrollar.

CAPÍTULO 3: Modelación del sistema

3.1 *Introducción*

Este capítulo dará continuación al desarrollo de los artefactos obtenidos en la fase de Modelación del sistema como: Modelo conceptual, Diagramas de clases del diseño, Diagrama de componentes, Modelo de datos. Se realizará el diseño de los componentes a desarrollar, persiguiendo como fin entender los requisitos funcionales del software. Se especificarán los patrones utilizados para la elaboración de los diagramas de clases. Se define además la arquitectura que va a tener el sistema.

3.2 *Arquitectura*

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema. Una Arquitectura de Software o Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.

3.2.1 *Arquitectura utilizada*

Se utilizó la arquitectura Multi-capa, pues permite el aislamiento de la lógica de aplicaciones en componentes independientes susceptibles de reutilizarse después en otros sistemas. Hace una distribución de las capas en varios nodos físicos y en varios procesos. Cada capa puede ser modificada tanto como sea necesario sin provocar cambios en las demás. Una capa no tiene dependencias con la capa superior, cada capa depende solamente de la fachada que le permite comunicarse con la capa inmediata inferior. Esta dependencia entre capas es normalmente a través de fachadas, asegurando que el acoplamiento sea el más bajo posible y la abstracción del funcionamiento de la capa inferior, sea casi total. Todo ello mejora el desempeño, la coordinación y el compartir la información.

La arquitectura utilizada presenta 2 características principales: Basada en Capas y Orientada a Componentes, donde cada componente implementa el patrón Modelo Vista Controlador, el cual se explica posteriormente en el documento.

3.2.1.1 Diseño de la arquitectura de sistema

La Arquitectura de Sistema es una de las disciplinas más complejas dentro de la Arquitectura de Software, responsable de definir correctamente cohesionados, acoplados e interrelacionados los elementos computacionales del producto. Aquí están presentes los niveles de empaquetamiento del diseño arquitectónico, estos tienen un alto impacto en el diseño de la solución.

Nivel Sistema. El nivel de sistema es el TODO. Está compuesto por todos los subsistemas, y es la vista global del producto.

Nivel Subsistema. Corresponde con la abstracción de las áreas de proceso del negocio, que posee objetos propios y operaciones asociadas a esos objetos. Tiene implicación en el diseño de instalación de la solución, está compuesto por módulos u otros subsistemas que a su vez están compuestos igualmente por módulos. La colaboración entre subsistemas será gestionada por el nivel más alto, nivel de sistema.

Nivel Módulo. Este nivel surge para empaquetar conjunto de componentes con funcionalidades similares. Puede contener además de los componentes, otros módulos, según el nivel de complejidad del negocio. La colaboración entre módulos será gestionada por el nivel de subsistema.

Nivel Componente. Corresponde con la abstracción de los procesos concretos contenidos en los procesos generales de las áreas de proceso que se modelan. Está compuesto por los elementos estructurales del diseño y componentes.

3.2.1.1.1 Descripción de los elementos de la arquitectura

La actividad tiene como responsabilidad identificar y diseñar la configuración de los elementos de mayor abstracción de la arquitectura de sistema (Subsistemas, Módulos y componentes más generales). Se identifican los estilos y patrones arquitectónicos a utilizar según el problema que se modela. Se realiza una propuesta candidata de los patrones de diseño a utilizar en los principales elementos (componentes) identificados, se revisa el cumplimiento de los patrones GRAPS en el diseño arquitectónico propuesto.

✓ ¿Qué es un patrón?

En términos generales, un patrón es un conjunto de información que proporciona respuesta a un conjunto de problemas similares. Es una solución a un problema en un contexto, donde (Bustamante):

- ✓ El Contexto son las situaciones recurrentes a las que es posible aplicar el patrón.
- ✓ El Problema es el conjunto de metas y restricciones que se dan en ese contexto.
- ✓ La Solución es el diseño a aplicar para conseguir las metas dentro de las restricciones.

3.2.1.1.1 Patrón arquitectónico utilizado

Los patrones arquitectónicos son los que definen la estructura de un sistema software. Se componen de subsistemas con sus responsabilidades. Tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema. (PBWORKS)

Como patrón arquitectónico se utilizó el Modelo Vista Controlador (MVC). El patrón MVC es un patrón que ayuda a darle cierta estructura lógica a las aplicaciones. Su principal objetivo es separar la lógica del negocio de la lógica de presentación o interfaz. Posibilita organizar el flujo de datos en las aplicaciones permitiendo construir sistemas más robustos y fáciles de mantener y extender. En este patrón se identifican tres componentes fundamentales que se relacionan entre sí:

Modelo: Representa a toda la información con la que opera la aplicación. Gestiona el comportamiento y los datos del dominio. Responde a las peticiones de información sobre el estado, que vienen de la Vista.

Vista: La Vista gestiona la presentación de la información en la aplicación. Todo lo relativo a la interfaz de usuario, los datos de que dispone para seguir interactuando con la aplicación.

Controlador: Gestiona las entradas del usuario. Se encarga de dar respuesta a eventos invocados desde la Vista. Llama a la lógica de negocio para procesar y producir una respuesta. Interpreta las entradas del usuario, informando al modelo y/o a la vista de los cambios que supongan esas entradas.

Ejemplo de cómo se utiliza el patrón: Un usuario realiza una petición a la clase controladora GestionarFipController.php la cual procesa la información y solicita los datos en la clase modelo, ejemplo DatFipModel.php que se encarga de realizar las operaciones pertinentes en la base de datos y enviar los datos pedidos a la clase controladora donde son capturados mediante una URL en la vista gestionarfip.js donde se le da la respuesta al usuario.

3.2.1.1.2 Patrones de diseño utilizados

Los patrones de diseño son formas de solucionar muchos problemas parecidos de un mismo modo. De hecho, se centran en la parte usual del componente a diseñar, y de esa forma son más específicos. Ayudan a evitar que los cambios en el sistema se realicen de una forma concreta de manera que se afecte lo menos posible. Los siguientes patrones fueron los utilizados para realizar el diseño de la aplicación:

✓ **Patrones Grasp** (Patrones generales de software para asignar responsabilidades)

Los patrones Grasp describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Existen nueve patrones Grasp los cuales son: Experto, Creador, Alta Cohesión, Bajo Acoplamiento, Controlador, Polimorfismo, Fabricación Pura, Induración y No Hables con Extraños. De ellos se utilizaron los siguientes:

Experto: Está enfocado en resolver el problema de ¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos? La solución se basa en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

En la aplicación se definieron clases como la clase `DatFipModel`, para la Gestión de FIP y para la Configuración de Perfil, está la clase `DatPerfilModel`. Estas clases efectuarán los procedimientos para: insertar, actualizar y eliminar cada elemento.

Creador: Asignar responsabilidades para la creación de objetos. El patrón trata de resolver el problema de diseño de quién debería ser responsable de crear una nueva instancia de alguna clase.

Este patrón se evidencia en las clases del paquete `Domain`, las que crean los objetos de tipo `Doctrine_Query`, posibilitando el acceso a la información almacenada a nivel de datos.

Bajo acoplamiento: Es la meta principal que es preciso tener presente durante el diseño. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento.

Al definir las clases persistentes correspondientes al Modelo de Datos, se estuvo aplicando este patrón.

Alta cohesión: Plantea que una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. O sea, indica la coherencia que debe existir en la información almacenada en una clase.

Un ejemplo de cómo se utilizó este patrón es el diseño del subsistema de Dirección por Objetivo. El subsistema está agrupado en componentes según los procesos que se realizan. Por lo que cada una de las clases que implementa dicho componente está relacionada sobre la misma área funcional.

Controlador: Asigna la responsabilidad a una clase de manejar mensajes correspondientes a eventos en un sistema. Dicha clase es la encargada de recibir los datos del usuario y enviarlos a las distintas clases. La aplicación del patrón conlleva a separar la lógica de negocios de la capa de presentación.

El patrón se ejemplifica a través de las clases GestionarFipController, ConfigurarPerfilController, las que tendrán a cargo la responsabilidad de manejar los eventos dentro del componente

✓ **Patrones Gof** (Gang of Four)

Los patrones Gof son 23 patrones que describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Están clasificados en patrones creacionales, estructurales y de comportamiento.

Los patrones creacionales se encargan de la creación de los objetos ayudando a que el sistema sea independiente de la creación, composición y representación de los objetos. Los patrones estructurales son los encargados de definir cómo las clases y objetos están compuestos para formar estructuras más grandes. Usan la herencia para componer interfaces u objetos en tiempo de ejecución. Los patrones de comportamiento plantean algoritmos y la asignación de responsabilidades entre objetos. En la aplicación se utilizaron los siguientes patrones:

Fachada: Patrón estructural que provee una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema. Es utilizado en varios de los diagramas de clases realizados en el diseño. Un ejemplo de su uso es en los servicios, donde la relación existente entre las clases controladoras y los servicios permite acceder a métodos que no están implementados en el componente Gestionar FIP y que se encuentran, tanto en otros componentes pertenecientes al subsistema Dirección por Objetivo, como en otros subsistemas externos. La interfaz utilizada como fachada sería el IoC (en inglés: Inversion of Control) que se explicará posteriormente.

Cadena de responsabilidad: La cadena de responsabilidad se encarga de evitar el acoplamiento del remitente de una petición a su receptor, dando a más de un objeto la posibilidad de manejar la petición. Es utilizado en la mayoría de los diagramas de clases siendo aplicado en el tratamiento de excepciones. Un ejemplo de su uso es cuando se produce un error al insertar en la base de datos, el cual es captado por las capas superiores, reenviando la excepción hasta la capa de aplicación donde traduce al lenguaje del usuario.

3.2.1.2 Diseño de la arquitectura de datos

La Arquitectura de Datos, identifica y precisa las mejores clases de datos que apoyan las funciones del negocio definidas en el modelo de negocios. Es la primera de las arquitecturas a ser concretadas porque la calidad de los datos es el producto básico de la función de la Ingeniería de Software. La arquitectura de datos tiene como objetivo puntualizar los principales tipos y fuentes de datos necesarios para dar soporte a las actividades de la entidad, de manera que sean entendibles por los participantes, completas y consistentes y estables. (Unidad de Compatibilización Integración y Desarrollo De Software para la Defensa, 2009)

3.2.1.2.1 Modelo de datos

El modelo de datos surge a través de la normalización del Modelo conceptual. El mismo presenta las tablas persistentes que intervienen durante el desarrollo de los dos componentes. Muestra además la relación entre dichas tablas y los atributos que van a poseer cada una.

Muchas veces es necesario diseñar un modelo de datos donde se repitan campos que ya existen en otras tablas dentro o fuera del mismo esquema ya que se necesita almacenar y recuperar información de forma rápida con la menor cantidad de código y accesos a tablas mediante sentencias JOIN. Esto no descarta que una buena práctica del modelado de datos también implique no repetir campos para llevar el modelo de datos a la 3ra Forma Normal.

Para ver el modelo de datos ir a [Anexo 4](#).

3.3 Diseño

El diseño es el proceso de aplicar distintas técnicas y principios con el propósito de definir un producto con los suficientes detalles como para permitir su realización física. Con el diseño se pretende construir un

sistema que respete los requisitos funcionales y no funcionales identificados y traducirlos de manera que represente al software. Permite producir un modelo o representación técnica del software que se va a desarrollar. Es una guía que pueden leer y entender los que construyan el código y los que prueban y mantienen el software.

3.3.1 Diagrama de clases del diseño

Un **diagrama de clases** es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Mediante los diagramas de clases se especifican y describen las clases del software y de las interfaces en una aplicación. Los diagramas de clases de diseño junto a los demás artefactos que se generan en este flujo de trabajo son el centro de atención al final de la fase de Elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida y crea un plano del modelo de implementación.

Para realizar el diseño de la solución a desarrollar se identificaron cada una de las clases y funcionalidades a implementar de acuerdo a las necesidades de cada componente. Para ver los diagramas de clases correspondientes a los componentes implementados ver [Anexo 5](#).

3.3.2 Diagramas de secuencia

A continuación se presentan los diagramas de secuencia correspondientes a los requisitos funcionales identificados. Como su nombre lo indica, cada diagrama muestra la secuencia de operaciones que ocurren entre las clases implementadas ante una petición del usuario. O sea, dichos diagramas describen la manera de interactuar del usuario con la aplicación.

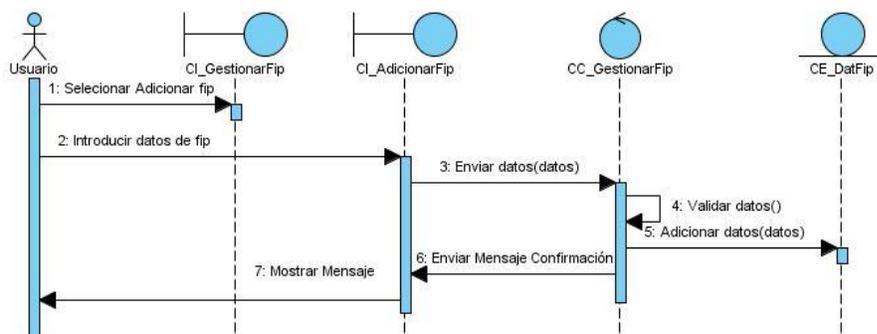


Figura 5 Diagrama de secuencia para el requisito Adicionar FIP

Se desarrolló un diagrama de secuencia para cada requisito funcional. Ver el [Anexo 6](#).

3.3.3 Métricas para validar el diseño

Actualmente uno de los desafíos técnicos suele ser las métricas de calidad del diseño, pues ayudan a entender al proceso que se utiliza para desarrollar un producto, así como a cuantificar y gestionar formas más efectivas de desarrollo. Para la evaluación de la calidad del modelo de diseño propuesto se hizo un estudio de la creación de métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objeto referenciadas por Pressman. La métrica concebida como instrumento para evaluar la calidad del diseño del componente Perfil del subsistema Dirección por Objetivo y su relación con los atributos de calidad definidos es la siguiente:

✓ Tamaño operacional de clase

Atributos de calidad que se abarca:

1. Responsabilidad. Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
2. Complejidad de implementación. Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
3. Reutilización. Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.

3.3.3.1 Tamaño Operacional de Clase (TOC)

El tamaño operacional de clase (TOC), está dado por el número de métodos asignados una clase.

Teniendo en cuenta los siguientes atributos:

Atributo que Afecta	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución en el grado de reutilización de la clase.

Figura 6 Tamaño Operacional de Clase (TOC)

Resultados del instrumento de evaluación de la métrica Tamaño Operacional de Clase (TOC)

	Categoría	Criterio
Responsabilidad	Baja	<=Prom.
	Media	Entre Prom. Y 2* Pom.
	Alta	> 2* Prom.
Complejidad implementación	Baja	<=Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	<= Prom.

Figura 7 Rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC

Para ver los resultados de la evaluación de la métrica TOC para FIP y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) referirse al Anexo 7

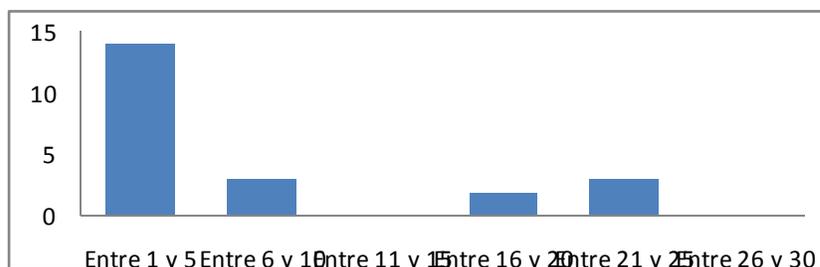


Figura 8 Representación de los resultados obtenidos (agrupados por de intervalos)

La representación de la incidencia de los resultados de la evaluación de la métrica TOC por cada atributo aplicado al componente FIP y las tablas correspondientes a la métrica aplicada al componente Configurar Perfil se muestra a partir del Anexo 8.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica TOC, se puede concluir que el diseño de los componentes FIP y Configurar Perfil tiene una calidad buena. Se puede observar que más de 60% de las clases posee menos cantidad de operaciones que la media registrada en las mediciones. Además, más del 50% de las clases posee evaluaciones

positivas en los atributos Responsabilidad y Complejidad de Implementación y en el caso del atributo Reutilización, posee un alto índice, lo que garantiza la reutilización de los componentes.

3.4 Conclusiones

En el capítulo se generaron todos los artefactos establecidos en la fase de Modelación, sirviendo como entrada a la fase de Construcción. Se realizó el diseño de la solución propuesta, el cual se validó mediante métricas de calidad, obteniéndose resultados aceptables. Se puede decir que con todo lo realizado en el capítulo, se cumplieron los objetivos propuestos en el presente trabajo, dándole solidez a la solución. Se establecieron las bases por las cuales se regirán los desarrolladores para poder pasar a la implementación de la solución. Quedando así concluida la fase de Modelación.

CAPÍTULO 4: Implementación y Prueba de los componentes Gestionar FIP y Configuración de Perfil

4.1 Introducción

En el presente capítulo se le estará dando terminación a la fase de Construcción que propone la metodología de desarrollo de software utilizada. La misma implica la implementación de los dos componentes a desarrollar y las pruebas realizadas a cada uno garantizando su funcionamiento y calidad.

4.2 Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, se implementan en términos de componentes, ya sean ficheros de código fuente y ejecutables. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados. Se evidencia la dependencia entre los componentes.

4.2.1 Diagrama de componentes

Un componente es un grupo de objetos o componentes más pequeños que interactúan entre ellos y se combinan para dar un servicio. Un diagrama de componente es, como su nombre lo indica, un esquema o diagrama que muestra las interacciones y relaciones de los componentes de un modelo.

Para lograr una implementación eficiente, se necesita contar con un diagrama que represente todos los componentes relacionados con el módulo a implementar. El mismo facilitará al programador la implementación de los servicios que utilizará el módulo y los que brindará a otros módulos dentro del subsistema.

El siguiente diagrama corresponde a los componentes FIP y Configurar Perfil. Muestra los servicios que consumen y brindan cada uno de estos componentes dentro del Sub-sistema de Dirección por Objetivo.

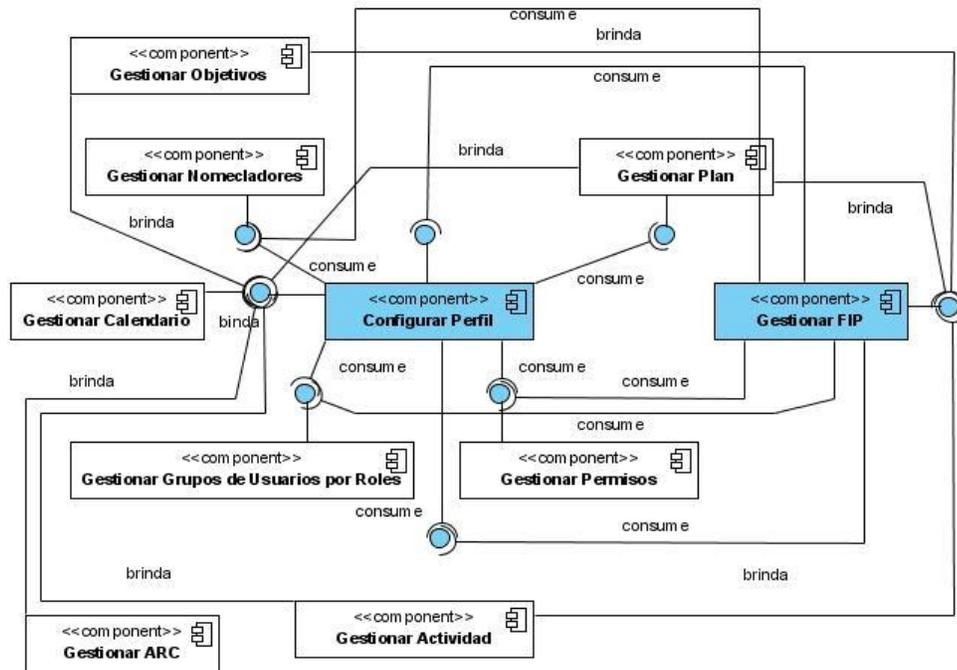


Figura 9 Diagrama de Componentes

4.2.2 Estrategias de integración

En los componentes se aplica la integración vertical que consiste en el flujo de los datos desde la vista hacia la capa de datos y viceversa, pasando por los diferentes elementos que componen la arquitectura. Esta consta de cuatro nodos de integración, el que se encuentra entre la vista y el controlador, el que está entre la del controlador y el modelo, el que vincula el modelo con el framework Doctrine y el que se encuentra entre el doctrine y la base de datos. Todo el código dentro un mismo componente utiliza llamadas a métodos o eventos de forma directa. La comunicación entre diferentes módulos y componentes se realiza mediante llamadas a la inversión de control (Inversion of Control en inglés, IoC). El IoC especifica respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que otro módulo o componente lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir.

Cada componente tiene su registro de los datos de los módulos en un fichero xml que será mapeado por el framework para el funcionamiento del mismo, dicho fichero tiene por nombre IoC y registra las funcionalidades que ofrecen los métodos de las clases control de los componentes del sistema. La base

de datos es accedida de forma directa mediante controladoras y los componentes rehusados son integrados mediante interfaces sencillas. (Silveira, 2009)

4.2.3 Estándares utilizados

Los estándares de diseño son ciertas pautas que se establecen para conseguir uniformidad en el diseño de una aplicación web. Logrando que la misma satisfaga las necesidades y exigencias de los clientes, sea compatible con las tecnologías utilizadas, cumpla con todos los objetivos perseguidos y que represente al centro a manera de sello y presentación de su trabajo.

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código.

La aplicación debe ser implementada siguiendo los estándares de documentación, implementación de interfaz de usuario e implementación de lógica de negocio diseñado para el proceso de desarrollo de software del ERP y la UCID.

- ERP-ARQ Estándar para el diseño de interfaces v1.1.
- Estándar para la documentación.
- Estándar de codificación.

4.3 Pruebas de software

Las pruebas de software son un conjunto de herramientas, técnicas y métodos que evalúan el desempeño de un programa. Las pruebas constituyen una etapa imprescindible durante el proceso de desarrollo del software, pues permiten detectar y corregir el máximo de errores posibles antes de la entrega al cliente del software desarrollado. Por lo que el éxito de las mismas puede mejorar la percepción de calidad del usuario final y lograr su satisfacción. El objetivo principal de las pruebas es asegurar que el software cumpla con las especificaciones requeridas y eliminar los posibles defectos que éste pudiera tener. Es importante considerar que las pruebas de software no garantizan que un sistema esté libre de errores, sino que se detecten la mayor cantidad de defectos posibles para su debida corrección. (Unidad de Compatibilización Integración y Desarrollo De Software para la Defensa, 2009)

4.3.1 Tipos de pruebas

Existen diversos tipos de prueba que se le realizan a la aplicación durante su desarrollo y una vez terminado el software.

Dentro de las pruebas en tiempo de desarrollo están las pruebas de unidades. Estas son pruebas de menor escala y consisten en probar cada uno de los módulos que conforman el programa. Cuando estos módulos son extensos o complejos se dividen para probar objetivamente partes más pequeñas, este tipo de pruebas es la más común.

Las pruebas de integración tienen por objetivo verificar el correcto funcionamiento de dos o más módulos, si bien se deben poner en práctica desde la creación de dos módulos que interactúen entre sí.

Cuando se considera que un módulo está terminado se realizan las pruebas sistemáticas. El objetivo de estas es buscar fallos a través de un criterio específico, estos criterios se denominan "pruebas de caja negra y de caja blanca".

Las *pruebas de caja negra* son aquellas que se enfocan directamente en el exterior del módulo, sin importar el código.

Las *pruebas de caja blanca* son mucho más amplias; normalmente se denominan pruebas de cobertura o pruebas de caja transparente. Básicamente la idea de pruebas de cobertura consiste en diseñar un plan de pruebas en las que se vaya ejecutando sistemáticamente el código hasta que haya corrido todo o la gran mayoría de él. (D'Onofrio)

4.3.1.1 Prueba de Caja Blanca aplicada a los componentes implementados

Como prueba de caja blanca aplicada se realizó la prueba del Camino básico. Esta prueba permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. También garantiza que durante la prueba en los casos de prueba obtenidos a través del camino básico se ejecute cada sentencia del programa por lo menos una vez. Para aplicar la técnica del camino básico se debe introducir la notación para la representación del flujo de control, este puede representarse por un Grafo de Flujo en el cual:

- ✓ Cada nodo del grafo corresponde a una o más sentencias de código fuente.
- ✓ Todo segmento de código de cualquier programa se puede traducir a un Grafo de Flujo.
- ✓ Se calcula la complejidad ciclomática del grafo.

Para construir el grafo se debe tener en cuenta la notación para las instrucciones

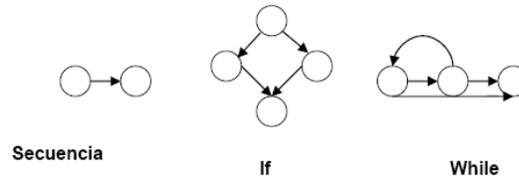


Figura 10 Notación de grafos de flujo para las instrucciones: Secuenciales, If, While

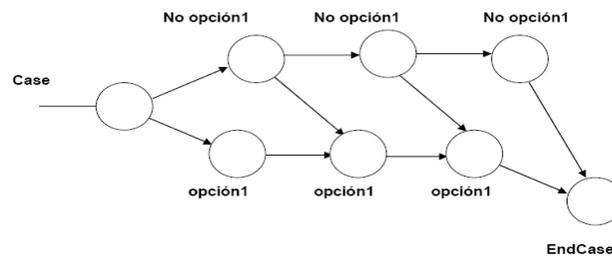


Figura 11 Notación de grafos de flujo para la instrucción Case

Un grafo de flujo está formado por 3 componentes fundamentales que ayudan a su elaboración y comprensión, estos brindan información para confirmar que el trabajo se está haciendo adecuadamente.

Componentes del grafo de flujo:

Nodo: son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

Aristas: son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.

Regiones: son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento. (López, 2009)

Para realizar la prueba del camino básico es preciso calcular la complejidad ciclomática del algoritmo a analizar. A continuación se enumera las sentencias de código del procedimiento realizado sobre el método `buscardenominacionnomencladorAction ()` el cual se encarga de buscar el tipo nomenclador asociado a un `idnomenclador` dado.

```
function buscardenominacionnomenclador($idnomenclador, $tipo) {

    $idusuario = $this->global->Perfil->idusuario; 1
    $estados = $this->pIntegrator->configuracionpdo->GetTodosNomencladoresPorTipo ( $tipo, $idusuario ); 1
    foreach ( $estados as $estado ) { 2
        if ( $estado ['idnomenclador'] == $idnomenclador) 3
            return $estado ['denom']; 4
    } 5
    return ''; 6
}
```

Figura 12 Algoritmo buscardenominacionnomencladorAction ()

Grafo de flujo asociado al código anterior

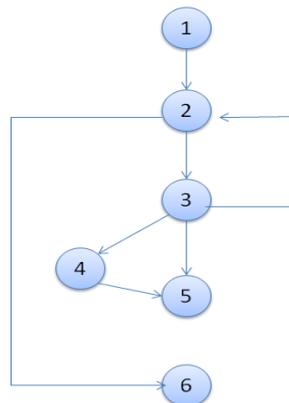


Figura 13 Grafo de flujo asociado al algoritmo buscardenominacionnomencladorAction ()

Cálculo de la complejidad ciclomática a partir de un segmento de código

La complejidad ciclomática es una métrica de software extremadamente útil pues proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. (López, 2009)

Para desarrollar el procedimiento se debe conocer la cantidad total de aristas del grafo y la cantidad total de nodos parámetros para conformar para las siguientes fórmulas:

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

$$V(G) = (7 - 6) + 2$$

$$V(G) = 3$$

Siendo $|A|$ la cantidad total de aristas y $|N|$ la cantidad total de nodos.

Se puede usar también:

$$V(G) = P + 1 \quad V(G) = 2 + 1 \quad V(G) = 3$$

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

$$V(G) = R \quad V(G) = 3$$

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

Luego de haber realizado el cálculo por las 3 vías planteadas se llega a la conclusión que el algoritmo presentado anteriormente tiene una complejidad ciclomática de 3 que lo que significa que existen tres posibles caminos lógicos por donde recorrer el algoritmo.

Caminos básicos:

- 1) 1-2-3-4-5
- 2) 1-2-3-2-3-4-5
- 3) 1-2-6

A continuación se ejecutan los casos de pruebas para este procedimiento, se debe realizar al menos un caso de prueba por cada camino básico. En este caso se diseñaron tres casos de prueba. Para su correcta ejecución se deben tener en cuenta los siguientes parámetros:

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 resultado que se espera que devuelva el procedimiento.

Descripción	Los datos de entrada cumplirán con los siguientes requisitos: El \$idnomenclador, y \$tipo serán números enteros.
Condición de ejecución	El \$tipo será igual a 4 y el \$idnomenclador será igual a 0001245
Entrada	\$tipo = 4 \$idnomenclador=000245
Resultados Esperados	El método debe devolver la denominación del nomenclador. Se alcanzaron los resultados esperados.

Figura 14 Caso de prueba para el camino básico # 1

Descripción	Los datos de entrada cumplirán con los siguientes requisitos: El \$idnomenclador, y \$tipo serán números enteros.
Condición de ejecución	El \$tipo será igual a 7 y el \$idnomenclador será igual a 0001245
Entrada	\$tipo =4 \$idnomenclador=000245
Resultados Esperados	El método debe devolver la denominación del nomenclador. Al comparar el \$idnomenclador que traía el arreglo es distinto al que se paso por parámetro.

Figura 15 Caso de prueba para el camino básico # 2

Descripción	Los datos de entrada cumplirán con los siguientes requisitos: El \$idnomenclador, y \$tipo serán números enteros.
Condición de ejecución	El \$tipo será igual null a y el \$idnomenclador será igual a null
Entrada	\$tipo =8 \$idnomenclador=000451
Resultados Esperados	El método debe devolver la denominación del nomenclador. No devuelve nada porque el arreglo que devuelve el servicio viene vacío.

Figura 16 Caso de prueba para el camino básico # 3

Luego de haber realizado las pruebas de caja blanca al método propuesto, se puede afirmar que dicho método cumple el propósito para el que fue creado, pues en cada diseño de casos de prueba se obtuvieron los resultados esperados.

4.3.1.2 Prueba de Caja Negra aplicada a los componentes implementados

Se realizó un caso de prueba para cada requisito implementado. A continuación se muestra el caso de prueba realizado al requisito Adicionar FIP que describe cada uno de los escenarios que pueden existir ante las posibles acciones realizadas por el usuario. De manera similar se realizaron los casos de prueba a los requisitos restantes. Para ver los requisitos restantes dirigirse a (Scull).

Nombre de la Sección	Escenarios de la Sección	Descripción de la Funcionalidad	Flujo Central
Adicionar FIP	Adicionar FIP	El sistema permite adicionar un nuevo FIP	El sistema almacena los datos creando un nuevo FIP
	Adicionar FIP de forma incorrecta	El sistema permite adicionar un nuevo FIP	El usuario introduce los datos dejando algunos campos vacíos. El sistema señala el error en el formulario y no permite adicionar un FIP.
	Cancelar la Acción	El sistema permite adicionar un nuevo FIP	El sistema cierra la ventana cancelando la acción.

Figura 17 Prueba de Caja Negra realizada al requisito Adicionar FIP

La prueba de Caja Negra realizada a los componentes arrojó 5 no conformidades a las que se le dieron solución en un período de 2 días.

4.4 Conclusiones

A través del desarrollo de este capítulo se le dio cumplimiento a la fase construcción generando un modelo de implementación. Este propicia un refinamiento de la vista de la arquitectura y sirve para mostrar parte de la vista estática del sistema a partir del diagrama de componentes, en el que se pueden observar las relaciones entre componentes que muestran la organización de dependencias lógicas entre los mismos. Se pudo apreciar como el modelo de implementación constituye una entrada fundamental para el modelo de prueba, del cual se obtuvieron los casos de prueba que ayudan a probar cada parte el producto realizado, el que ya se encuentra terminado.

Conclusiones

Se puede afirmar a modo de conclusión que se cumplieron todos los objetivos planteados a lo largo de la investigación. Se alcanzó el resultado propuesto, los componentes capaces de cubrir los procesos de Gestión de FIP y Configuración de Perfil de forma correcta. Dichos componentes permiten una gestión integral con funcionalidades generales y las particularidades del proceso de Dirección por Objetivo, ratificando así la utilidad y la validez del empleo de las tecnologías informáticas para apoyar las labores en cualquier esfera de la vida. Lo anterior se ve demostrado a través de lo siguiente:

- ✓ Se analizaron los fundamentos teóricos y las principales aplicaciones vinculadas al campo de acción, tanto a nivel nacional como internacional, demostrando la necesidad de automatizar los procesos de Gestión de Factores que Influyen en el Plan y Configuración de Perfil.
- ✓ Se realizó un estudio sobre las técnicas, tecnologías y herramientas definidas para conocer las bondades que brindan las mismas, llegando a la conclusión de que cumplen con los requisitos para proyectar soluciones en el campo de la soberanía tecnológica.
- ✓ Se obtuvo una visión mejor y profunda de los procesos a automatizar, que contribuyó a la obtención de una solución lo suficientemente robusta la cual mejoró el proceso de Dirección por Objetivo en las entidades del país.

Recomendaciones

Se recomienda:

- ✓ Agregar funcionalidades al componente FIP en versiones posteriores que se hagan del mismo, como por ejemplo brindarle la opción al usuario de realizar una búsqueda avanzada por cada atributo que posee un FIP y posibilitar la anexión de archivos a un FIP deseado.
- ✓ Realizar el despliegue de los dos componentes implementados en las entidades del país, para lograr el seguimiento y control de los FIP y la personalización de funcionalidades y de usabilidad en el proceso de Dirección por Objetivo.
- ✓ Realizar un manual de usuario para facilitarle al usuario interactuar con la aplicación.
- ✓ Utilizar XML para realizar las validaciones de códigos.

Bibliografía

Alvarez, Miguel Angel. desarrolloweb. [En línea] <http://www.desarrolloweb.com/articulos/que-es-html.html>.

Attribution-NonCommercial-ShareAlike 2.5 . Subversion-Guía Ubuntu. [En línea] <http://www.guia-ubuntu.org/index.php?title=Subversion>.

Brito, Henry Raúl González. 2007. ERP cubano, un paso estratégico para la consolidación del Software Libre en Cuba. [En línea] 2007. http://www.informaticahabana.com/evento_virtual/?q=node/286&ev=III%20Taller%20Internacional%20de%20Software%20Libre..

Bustamante, Diana Paola Hurtado. *Patrones GRASP*.

Careaga, Adriana. 2005. *¿POR QUE ES IMPORTANTE ESTABLECER OBJETIVOS EN LA PLANIFICACIÓN DE UN CURSO? 2005.*

Carnota, O. *Teoría y practica de la dirección socialista*. La Habana : Pueblo y Educación.

Centro de Soluciones de Gestión. *Acta de Aceptación de Requisitos de DPO*.

CENTRO DE SOLUCIONES DE GESTION. *CICLO DE VIDA DEL PROYECTO*.

Centro de Soluciones de Gestión. *Validación de procesos de negocios-DPO*.

Ciberaula Madrid. Ciberaula Linux. [En línea] http://linux.ciberaula.com/articulo/linux_apache_intro/.

Cornejo, José Enrique González. *¿Qué es UML?*

Diseño web España. 2010. Masadelante.com. [En línea] 2010. <http://www.masadelante.com/faqs/url>.

D'Onofrio, Diego Lucio. el Guille. [En línea] <http://www.elguille.info/Clipper/probando.htm>.

Factoría de Internet S.L. 2010. WebTallercom. [En línea] 2010. <http://www.webtaller.com/maletin/articulos/que-es-ajax.php>.

FREE DOWNLOAD MANAGER. Sitio de descarga de software. [En línea] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).

Gestión, Centro de Soluciones de. 2009. *Modelo Conceptual del subsistema de Dirección por Objetivo*. La Habana : s.n., 2009.

Gracia, Joaquín. 2006. WebEstilo. [En línea] 2006. <http://www.webestilo.com/javascript/>.

Grau, Xavier Ferré. 2004. Desarrollo Orientado a Objetos con UML. [En línea] 2004. <http://www.clikear.com/manuales/uml/>.

Gutierrez, Jorge A. Saavedra. 2009. *PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad)*. 2009.

Gutierrez, Juan. 2004. *Eclipse (2.1) y Java*. Valencia : s.n., 2004.

Haro, Miguel Ángel Alemany. *PLAN DE MEJORA*.

- HERRAMIENTAS CASE. [En línea]
<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm..>
- Historia de TortoiseSVN.** Historia de TortoiseSVN. [En línea]
http://tortoissvn.net/docs/nightly/TortoiseSVN_es/tsvn-introduction.html.
- ismaeledios.** debugmodeon. [En línea] <http://es.debugmodeon.com/usuario/ismaeledios>.
- Katy Caridad Herrera Lemus, Carlos Cristóbal Martínez Martínez. 2005.** *Desarrollo de la Dirección por objetivos.* 2005.
- Lago, Ramiro. 2009.** *Patrones de diseño software.* 2009.
- Laura Bermejo Sanz, Enrique Gómez Monreal.** *Eclipse como IDE.*
- López, Carlos. 2008.** GestioPolis. [En línea] 2008. [Citado el: 15 de 12 de 2009.]
<http://www.gestiopolis.com/lidera/revistas/edic7/ERP.htm>.
- López, Jesus. 2008.** ApuntesGestión. [En línea] 13 de 8 de 2008. [Citado el: 15 de 12 de 2009.]
<http://www.apuntesgestion.com/2008/08/13/definicion-planificacion/>.
- López, Oilede Pérez Ramírez y Yonisley García. 2009.** *Desarrollo del Componente Trabajador del Subsistema Capital Humano del sistema Cedrux.* 2009.
- Madrid, Víctor Javier. 2010.** Introducción a BPMN. [En línea] 2010.
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=bpmn>.
- Maestros del Web. 2007.** ¿Qué son las bases de datos? [En línea] 2007.
[http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/..](http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/)
- Melton Technologies. 2010.** deGerencia. [En línea] 2010.
http://www.degerencia.com/articulo/rescatando_la_dpo_en_un_nuevo_contexto.
- Misrespuestas.** MisRespuestas. [En línea] <http://www.misrespuestas.com/que-es-un-servidor-web.html>.
- Open E-learning.** mailxmail. [En línea] [Citado el: 13 de 12 de 2009.] <http://www.mailxmail.com/curso-administracion-empresas/que-es-planificacion>.
- PBWORKS.** isg3. [En línea] <http://isg3.pbworks.com/Patrones-Arquitect%C3%B3nicos>.
- Pressman, Roger S.** *Ingeniería de Software* .
- Prieto, Félix. 2009.** *Patrones de diseño.* 2009.
- Real, Red de Conocimiento en Informática Industrial y Aplicaciones de Gestión en Tiempo.** redindustria. [En línea]
- Scull, Sandy Machado.** *Diseño de Casos de Prueba.*
- Silveira, José Antonio Linares y Torres Jorge Núñez. 2009.** *Implementación del Componente Nomencladores y Configuración del Sistema Integral de Gestión Cedrux.* 2009.
- Toledo, Suiberto Hechavarría. 2005.** *La Dirección por Objetivos y la planeación estratégica.* 2005.

Tutorial de Postgre SQL. [En línea] <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>.

UCID. *Especificaciones complementarias.*

UCID, ERP-Cuba UCI. *Manual del marco de trabajo.*

UCID, ERP-Cuba UCI. 2009. *Manual del marco de trabajo.* La Habana : s.n., 2009.

Unidad de Compatibilización Integración y Desarrollo De Software para la Defensa. 2009. *Proceso de Desarrollo y Gestión de Proyectos de Software.* 2009.

Yuliet Galán Ramírez, Néstor Bernal Vidal. 2008. *Proceso de Planificación por Objetivos en las entidades de las FAR.* 2008.

Glosario de Términos

Acceso Concurrente Multiversión: permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Apache/BSD: (Apache License o Apache Software License) es una licencia de software libre creada por la Apache Software Foundation (ASF)

Aplicaciones interactivas: Aplicación que permite dibujar y crear una animación con los pasos dados.

Artefactos: productos tangibles del proyecto que son producidos, modificados y usados por las actividades.

Asincrónica: la comunicación que se establece entre el servidor y el cliente, cuando el cliente no espera ningún mensaje del servidor para continuar su labor.

BSD: es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution* o Distribución de Software Berkeley).

Código fuente abierto: (en inglés *open source*) es el término con el que se conoce al software distribuido y desarrollado libremente.

CSS: CSS (Cascading Style Sheets, u Hojas de Estilo en Cascada) es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación.

CVS: (Concurrent Versions System) es una aplicación informática que implementa un sistema de control de versiones.

Depuración de código: acción que hacemos para optimizar el código.

DHTML: HTML Dinámico o DHTML (del inglés *Dynamic HTML*). Se utiliza para hacer páginas web que respondan dinámicamente a eventos producidos en el navegador, por los cuales se pueden mostrar y ocultar elementos de la página, se puede modificar su posición, dimensiones, color.

DOM: (Document Object Model o Modelo de Objeto de Documento) es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Java Script para cambiar dinámicamente los contenidos y aspecto de la página.

DPO: Dirección por Objetivo, es un sistema de dirección cuyo propósito básico es lograr una adecuada coordinación de personas, recursos y mecanismos de una organización para obtener resultados relevantes sobre la base de la amplia y efectiva participación.

FIP: Factores Internos que influyen en el Plan.

FlashPlayer: es una aplicación en forma de reproductor multimedia.

Flujos de trabajo: fases por las que transita el software.

Framework:(marco de trabajo) un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

GNU: un sistema operativo completamente libre.

Hardware: corresponde a todas las partes físicas y tangibles de una computadora.

Herramientas: es un objeto elaborado a fin de facilitar la realización de una tarea mecánica.

HTML: HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

Internet Explorer (IE): navegador web.

Interoperabilidad: es la condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos.

Iteraciones: son pequeños procesos compuestos de varias actividades cuyo objetivo es entregar una parte del sistema parcialmente completo, probado, integrado y estable

MSDOS: Microsoft Disk Operating System, sistema operativo de disco de Microsoft.

Multiplataforma: es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Páginas interactivas: Páginas web, que permiten mediante sus características comunicarse con el usuario.

Plan: proyecto que elabora toda entidad para ejecutar su contabilidad en un período de tiempo.

Plataforma .NET: plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

Plugins: es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Repositorio: es un sitio centralizado donde se almacena y mantiene información, habitualmente bases de datos o archivos informáticos.

Sistema operativo: es un programa informático que actúa de interfaz entre los dispositivos de hardware y el usuario.

SOAP: (Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Software: se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica.

Técnicas: es un procedimiento o conjunto de reglas, normas o protocolos, que tienen como objetivo obtener un resultado determinado.

Unix: es un sistema operativo portable, multitarea y multiusuario.

URL: son las siglas de **Localizador de Recurso Uniforme** (en inglés *Uniform Resource Locator*), la dirección global de documentos y de otros recursos.

W3C: El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

XHTML: (eXtensible Hypertext Markup Language o lenguaje extensible de marcado de hipertexto) es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.

XSLT: (Extensible Stylesheet Language Transformations o lenguaje de hojas extensibles de transformación), que permite convertir documentos XML de una sintaxis a otra (por ejemplo, de un XML a otro o a un documento HTML).

Anexos

Anexo 1

Objetivo	Emitir un FIP que puede estar dado por el cumplimiento de otro emitido previamente por el organismo superior o por una necesidad interna del organismo.
Evento(s) que lo genera(n)	Se recibe el Documento de FIP del nivel superior que contiene uno o varios FIP emitidos por el organismo superior. Surge una necesidad interna que conlleva a emitir un FIP.
Pre condiciones	Debe existir un Documento de FIP del nivel superior o la necesidad interna de emitir un FIP.
Marco legal	N/A.
Clientes internos	Establecer plan. Puntualizar plan. Establecer sistema de control.
Clientes externos	N/A.
Entradas	Documento de FIP del nivel superior.
Flujo de eventos	
Flujo básico	
	<ol style="list-style-type: none"> Si se recibe un FIP emitido por el nivel superior, se realiza la acción: Estudiar el Documento de FIP del nivel superior. Este documento contiene uno o varios FIP que son emitidos por el organismo superior. Una vez recibido el documento, se hace necesario estudiarlo para dar cumplimiento al (los) FIP recibido(s). En caso de que se emita un FIP por una necesidad interna del organismo se ejecuta el flujo alterno 1.a Crear nuevo FIP. Formular nuevos objetivos y/o actividades. Para dar cumplimiento al FIP emitido se formulan nuevos objetivos y/o nuevas actividades los cuales deben ser incluidos en el plan. En caso de que se sea necesario emitir un nuevo FIP se ejecuta el flujo alterno 1.a Crear nuevo FIP. Se ejecuta el proceso Establecer plan. Enviar documento FIP al nivel inmediato inferior. Este documento es enviado a los subordinados con los nuevos cambios para que actualicen los planes correspondientes. Concluye el proceso.
Pos-condiciones	
	<ol style="list-style-type: none"> Se inserta en el documento los nuevos objetivos y/o actividades. Se acomete FIP emitido. Se ha actualizado el plan.
Salidas	
	<ol style="list-style-type: none"> Documento de FIP del nivel (Documento Word).
Flujos paralelos	
	N/A.
Pos-condiciones	
	N/A.
Salidas	
	N/A.
Flujo alterno	
Flujo alternativo 1. a Crear nuevo FIP.	
	<ol style="list-style-type: none"> Elaborar documento FIP. El jefe del nivel de dirección elabora un documento que contiene uno o varios FIP. Continúa en el paso 2 del flujo básico: Formular nuevos objetivos y/o actividades.

Pos-condiciones

1. Se ha elaborado un nuevo documento FIP.

Salidas

1. Documento de FIP del nivel (Documento Word).

Flujo alternativo

Flujo alternativo 2. a Crear otro FIP a partir de uno recibido.

1. Se ejecuta el flujo alternativo 1.a Crear nuevo FIP.

Pos-condiciones

N/A.

Salidas

N/A.

Asuntos pendientes

N/A.

Figura 18 Descripción del proceso de negocio Gestionar FIP

Anexo 2

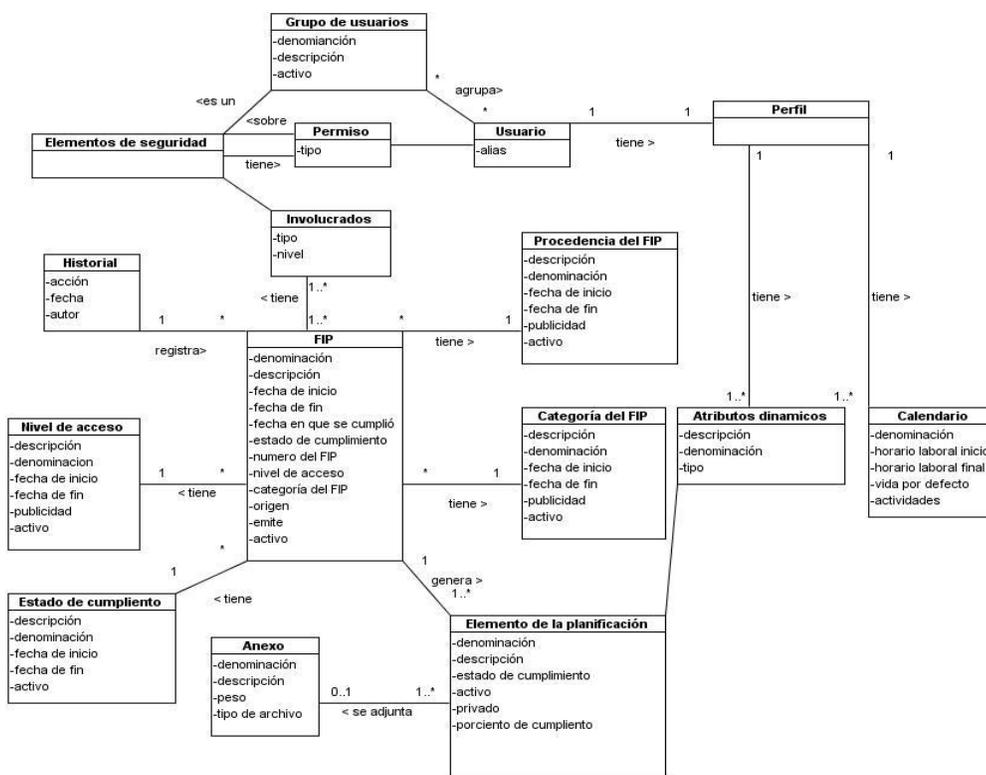


Figura 19 Modelo Conceptual

Anexo 3

	Se ha registrado al menos un FIP en el sistema.
Precondiciones	
Flujo de eventos	
Flujo básico	
1	Se selecciona el FIP a modificar.
2	El sistema muestra y permite editar los datos del FIP.
3	Se introducen los datos del FIP: Denominación Descripción Categoría Fecha de emisión Fecha de cumplimiento Número Nivel de acceso Procedencia del FIP Documento origen Emite Porcentaje de cumplimiento
4	El sistema permite adicionar los involucrados y el responsable que pertenecerán al FIP.
5	El sistema permite seleccionar la opción de llevar el historial del FIP.
6	El sistema permite gestionar los indicadores que posibilitarán determinar el estado de cumplimiento del FIP.
7	El sistema determina el estado de cumplimiento basado en el porcentaje de cumplimiento registrado y los indicadores definidos.
8	El sistema permite seleccionar los elementos de la planificación con los cuales estará relacionado el FIP.
9	El sistema permite gestionar observaciones acerca del plan.
10	El sistema permite gestionar documentos anexos al plan.
11	El sistema valida (ver validación 1) los datos introducidos.
12	Si los datos son correctos el sistema los registra.
13	El sistema confirma el registro de los datos.
14	Concluye el requisito.
Pos-condiciones	
1	Se modificaron los datos del FIP.
Flujos alternativos	
Flujo alternativo 5.a Información errónea	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.
Pos-condiciones	
1	N/A.
Flujo alternativo 5.b Información incompleta	
1	El sistema señala los datos vacíos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.
Pos-condiciones	
1	N/A.
Flujo alternativo *.a El usuario cancela la acción	
1	Concluye el requisito.
Pos-condiciones	
1	No se registran las modificaciones realizadas.
Validaciones	

1	Se validan los datos según lo establecido en el Modelo conceptual CSG-ERP-N-DPO-i2201.	
Relaciones	Requisitos Incluidos	<p>El sistema permite gestionar los indicadores que posibilitarán determinar el estado de cumplimiento del FIP: Adicionar indicador, en la agrupación Gestionar indicadores.</p> <p>El sistema permite gestionar los indicadores que posibilitarán determinar el estado de cumplimiento del FIP: Modificar indicador, en la agrupación Gestionar indicadores.</p> <p>El sistema permite gestionar los indicadores que posibilitarán determinar el estado de cumplimiento del FIP: Eliminar indicador, en la agrupación Gestionar indicadores.</p> <p>El sistema permite gestionar observaciones acerca del FIP: Adicionar observaciones, en la agrupación Gestionar observaciones.</p> <p>El sistema permite adicionar observaciones acerca del FIP: Modificar observaciones, en la agrupación Gestionar observaciones.</p> <p>El sistema permite adicionar observaciones acerca del FIP: Eliminar observaciones, en la agrupación Gestionar observaciones.</p> <p>El sistema permite gestionar documentos anexos al FIP: Adicionar anexos, en la agrupación Gestionar anexos.</p> <p>El sistema permite gestionar documentos anexos al FIP: Modificar anexos, en la agrupación Gestionar anexos.</p> <p>El sistema permite gestionar documentos anexos al FIP: Eliminar anexos, en la agrupación Gestionar anexos.</p> <p>El sistema permite seleccionar los elementos de la planificación con los cuales estará relacionado el FIP: Adicionar relaciones entre elementos de la planificación, en la agrupación Gestionar relaciones entre elementos de la planificación.</p> <p>El sistema permite seleccionar los elementos de la planificación con los cuales estará relacionado el FIP: Eliminar relaciones entre elementos de la planificación, en la agrupación Gestionar relaciones entre elementos de la planificación.</p>
	Extensiones	N/A.
Conceptos	FIP	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Descripción</p> <p>Categoría</p> <p>Fecha de emisión</p> <p>Fecha de cumplimiento</p> <p>Número</p> <p>Nivel de acceso</p> <p>Procedencia del FIP</p> <p>Emite</p> <p>Fecha en que se cumplió</p> <p>Documento origen</p> <p>Estado de cumplimiento</p> <p>Porcentaje de cumplimiento</p> <p>Utilizados internamente:</p> <p>N/A.</p>
	Elemento de la planificación	<p>Visibles en la interfaz:</p> <p>Denominación</p> <p>Descripción</p> <p>Fecha de inicio</p> <p>Fecha de fin</p> <p>Porcentaje de cumplimiento</p> <p>Estado de cumplimiento</p> <p>Utilizados internamente:</p>

	N/A.
Observación	Visibles en la interfaz: Autor Fecha Asunto Texto Utilizados internamente: N/A.
Involucrado	Visibles en la interfaz: N/A. Utilizados internamente: Tipo Nivel
Anexo	Visibles en la interfaz: Denominación Descripción Fecha de inicio Fecha de fin Porcentaje de cumplimiento Utilizados internamente: N/A.
Indicador	Visibles en la interfaz: Estado de cumplimiento Utilizados internamente: N/A.
Rango	Visibles en la interfaz: Límite inferior Límite superior Utilizados internamente: N/A.
Historial	Visibles en la interfaz: Acción Fecha Autor Utilizados internamente: N/A.
Requisitos especiales	N/A.
Asuntos pendientes	N/A.

Figura 20 Especificación de requisito Modificar FIP

Precondiciones	Se ha registrado al menos un FIP en el sistema.
Flujo de eventos	
Flujo básico	
1	Se selecciona el FIP a eliminar.
2	El sistema verifica (ver validación 1) que se pueda eliminar el FIP.
3	Se solicita confirmación para eliminar el FIP.
4	Si el usuario confirma se elimina el FIP.
5	El sistema confirma la eliminación.
6	Concluye el requisito.
Pos-condiciones	
1	Se eliminó el FIP.

Flujos alternativos		
Flujo alternativo 3.a El FIP está en uso		
1		El sistema notifica por qué no puede eliminarse el FIP.
2		Se ejecuta el requisito Desactivar plan.
3		Concluye el requisito.
Pos-condiciones		
1		N/A.
Flujo alternativo *.a El usuario cancela la acción		
1		Concluye el requisito.
Pos-condiciones		
1		No se elimina el FIP.
Validaciones		
1		No se puede eliminar un FIP si tiene otros elementos de la planificación asociados.
Relaciones	Requisitos Incluidos	Se ejecuta el requisito Desactivar FIP: Desactivar FIP, en la agrupación Gestionar FIP.
	Extensiones	N/A.
Conceptos	FIP	Visibles en la interfaz: Denominación Descripción Categoría Fecha de emisión Fecha de cumplimiento Número Nivel de acceso Procedencia del FIP Emite Fecha en que se cumplió Documento origen Estado de cumplimiento Porcentaje de cumplimiento Utilizados internamente: N/A.
	Elemento de la planificación	Visibles en la interfaz: Denominación Descripción Fecha de inicio Fecha de fin Porcentaje de cumplimiento Estado de cumplimiento Utilizados internamente: N/A.
Requisitos especiales		N/A.
Asuntos pendientes		N/A.

Figura 21 Especificación de requisito Eliminar FIP

Precondiciones	Se ha registrado al menos un FIP en el sistema.
Flujo de eventos	
Flujo básico	
1	El sistema muestra un listado de los FIP. De los FIP se muestran: la denominación, la fecha de inicio, la fecha de fin y el estado de cumplimiento.

2	Concluye el requisito.	
Pos-condiciones		
1	N/A.	
Flujos alternativos		
Flujo alternativo		
1	N/A.	
Pos-condiciones		
1	N/A.	
Validaciones		
1	N/A.	
Relaciones	Requisitos Incluidos	N/A.
	Extensiones	N/A.
Conceptos	FIP	Visibles en la interfaz: Denominación Descripción Categoría Fecha de emisión Fecha de cumplimiento Número Nivel de acceso Procedencia del FIP Emite Fecha en que se cumplió Documento origen Estado de cumplimiento Porcentaje de cumplimiento Utilizados internamente: N/A.
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

Figura 22 Especificación de requisito Listar FIP

Se ha registrado al menos un FIP en el sistema.	
Precondiciones	
Flujo de eventos	
Flujo básico	
1	Se insertan los criterios de búsqueda: Denominación Fecha de inicio Fecha de fin.
2	El sistema muestra un listado de los FIP que cumplen los criterios de búsqueda especificados.
3	Concluye el requisito.
Pos-condiciones	
1	N/A.
Flujos alternativos	
Flujo alternativo 2.a No existen datos que cumplan con los criterios especificados	
1	El sistema notifica que no existen datos que cumplan con los criterios especificados.
Pos-condiciones	
1	N/A.
Validaciones	

1	N/A.	
Relaciones	Requisitos Incluidos	N/A.
	Extensiones	N/A.
Conceptos	FIP	Visibles en la interfaz: Denominación Descripción Categoría Fecha de emisión Fecha de cumplimiento Número Nivel de acceso Procedencia del FIP Emite Fecha en que se cumplió Documento origen Estado de cumplimiento Porcentaje de cumplimiento Utilizados internamente: N/A.
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

Figura 23 Especificación de requisito Buscar FIP

	Se ha registrado al menos un FIP en el sistema
Precondiciones	
Flujo de eventos	
Flujo básico	
1	Se insertan los criterios de búsqueda avanzada: Denominación Emite Origen Fecha de inicio No del FIP Nivel de acceso Fecha de fin Categoría del FIP Estado de cumplimiento Fecha en que se cumplió Privado Grado de cumplimiento (agregar) Descripción
2	El sistema muestra un listado de los FIP que cumplen los criterios de búsqueda especificados. Se muestran denominación, categoría y responsable.
3	Concluye el requisito.
Pos-condiciones	
1	N/A.
Flujos alternativos	
Flujo alternativo 2.a No existen datos que cumplan con los criterios especificados	
1	El sistema notifica que no existen datos que cumplan con los criterios especificados.
Flujo alternativo	

1	N/A.	
Pos-condiciones		
1	N/A.	
Validaciones		
1	N/A.	
Relaciones	Requisitos Incluidos	N/A.
	Extensiones	N/A.
Conceptos	FIP	Visibles en la interfaz: Denominación Descripción Categoría Fecha de emisión Fecha de cumplimiento Número Nivel de acceso Procedencia del FIP Emite Fecha en que se cumplió Documento origen Estado de cumplimiento Porcentaje de cumplimiento Utilizados internamente: N/A.
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

Figura 24 Especificación de requisito Realizar búsqueda avanzada de FIP

	El usuario ha activado al menos un FIP en el sistema.
Precondiciones	
Flujo de eventos	
Flujo básico	
1	Se selecciona el FIP a desactivar.
2	Se solicita confirmación para desactivar el FIP.
3	Si el usuario confirma se desactiva el FIP.
4	El sistema confirma la desactivación.
5	Concluye el requisito.
Pos-condiciones	
1	Se ha desactivado el FIP.
Flujos alternativos	
Flujos alternativos	
Flujo alternativo 3.a El FIP está en uso	
1	El sistema notifica por qué no puede desactivar el FIP.
2	Concluye el requisito.
Pos-condiciones	
1	N/A
Flujo alternativo 4.a El usuario cancela la acción	
1	Concluye el requisito.
Pos-condiciones	
1	No se desactiva el FIP.
Validaciones	

1	Se validan los datos según lo establecido en el Modelo conceptual CSG-ERP-N- DPO-i2201.	
2	Se puede desactivar un FIP si no tiene otros elementos de planificación asociados.	
Relaciones	Requisitos Incluidos	N/A.
	Extensiones	N/A.
Conceptos	FIP	Visibles en la interfaz: Denominación Descripción Categoría Fecha de emisión Fecha de cumplimiento Número Nivel de acceso Procedencia del FIP Emite Fecha en que se cumplió Documento origen Estado de cumplimiento Porcentaje de cumplimiento Utilizados internamente: N/A.
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

Figura 25 Especificación de requisito De activar FIP

		El usuario ha desactivado un FIP en el sistema.
Precondiciones		
Flujo de eventos		
Flujo básico		
1	Se selecciona el FIP a activar.	
2	Se solicita confirmación para activar el FIP.	
3	Si el usuario confirma se activa el FIP.	
4	El sistema confirma la activación.	
5	Concluye el requisito.	
Pos-condiciones		
1	Se ha activado el FIP.	
Flujos alternativos		
Flujo alternativo 4.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se activa el FIP.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual CSG-ERP-N- DPO-i2201.	
Relaciones	Requisitos Incluidos	N/A.
	Extensiones	N/A.
Conceptos	FIP	Visibles en la interfaz: Denominación Descripción Categoría Fecha de emisión

	Fecha de cumplimiento Número Nivel de acceso Procedencia del FIP Emite Fecha en que se cumplió Documento origen Estado de cumplimiento Porcentaje de cumplimiento Utilizados internamente: N/A.
Requisitos especiales	N/A.
Asuntos pendientes	N/A.

Figura 26 Especificación de requisito Activar FIP

Anexo 4

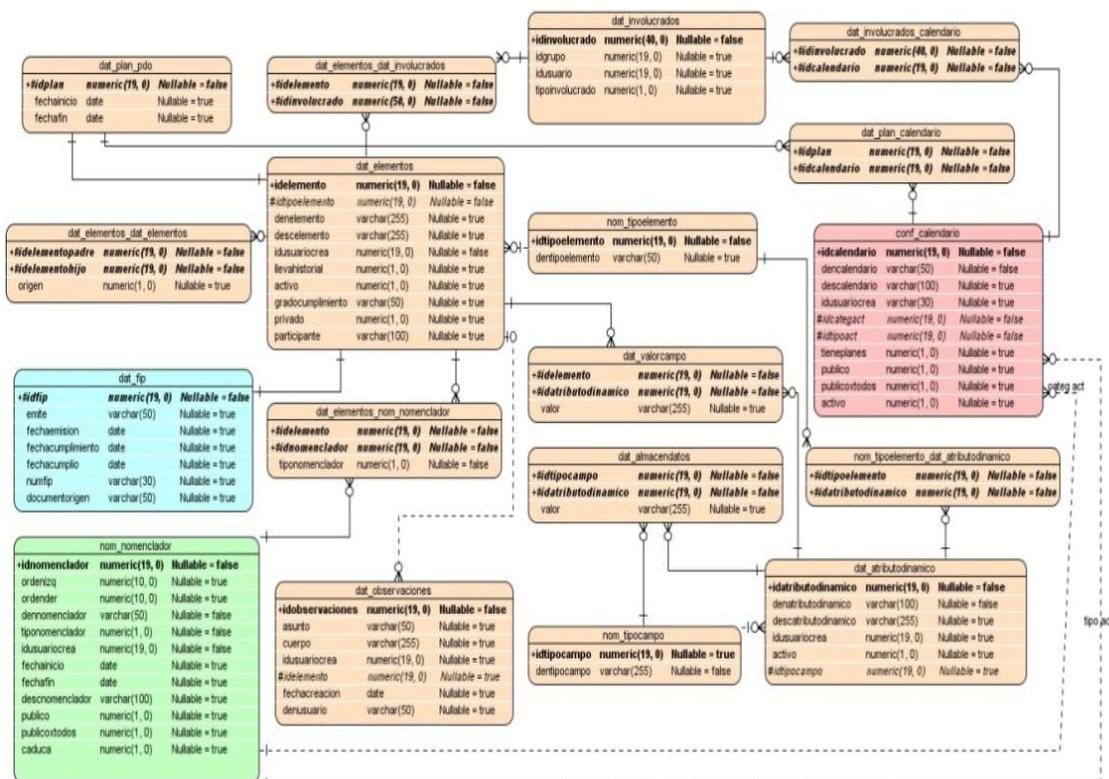


Figura 27 Modelo de Datos

Anexo 6

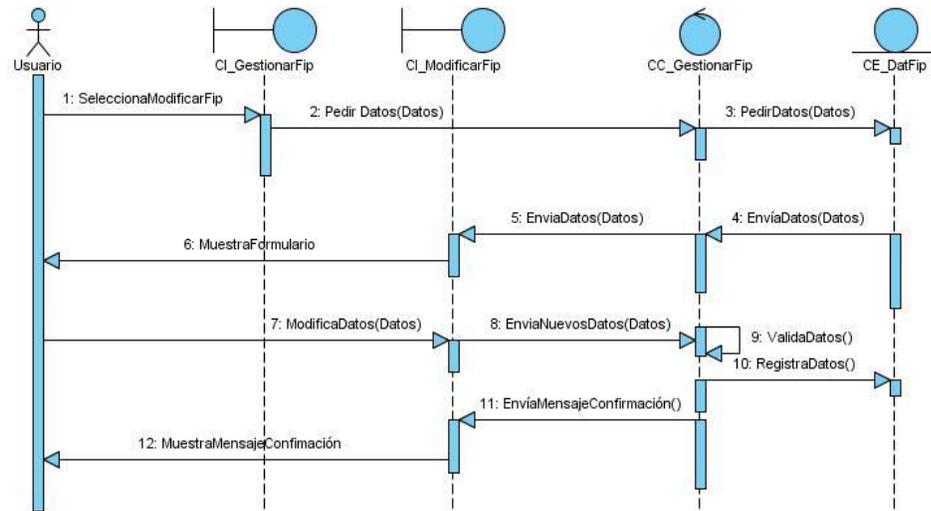


Figura 30 Diagrama de Secuencia para el requisito Modificar FIP

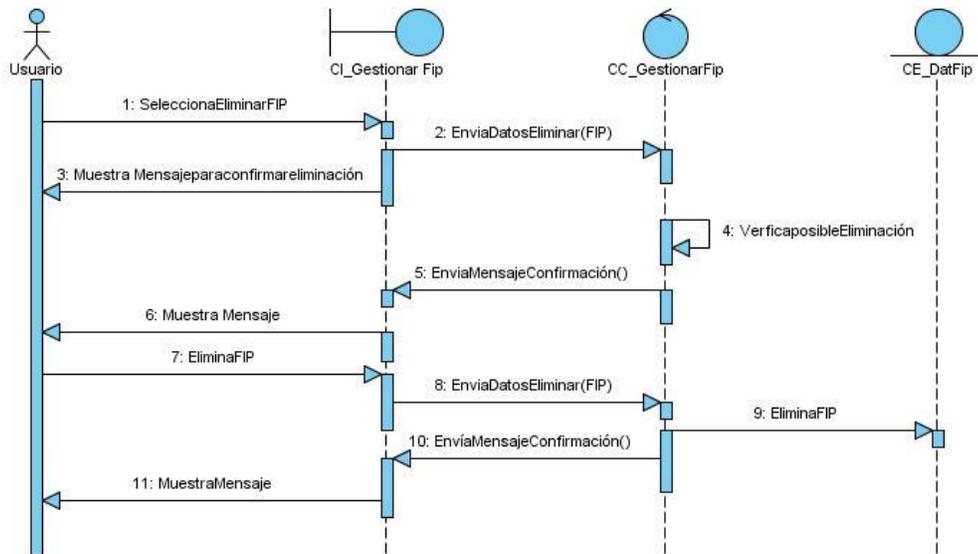


Figura 31 Diagrama de secuencia para el requisito Eliminar FIP

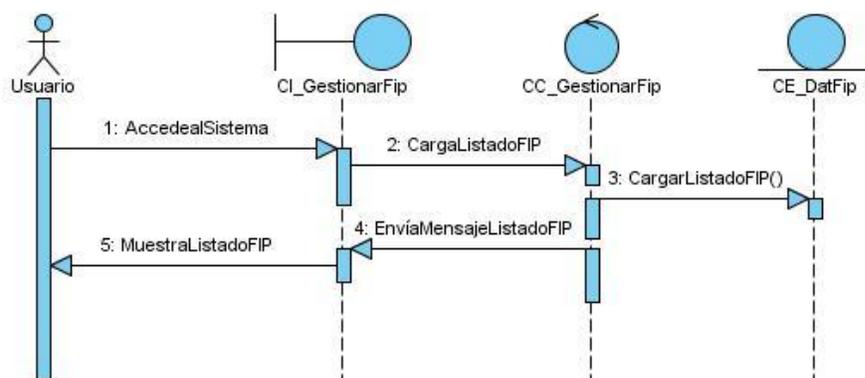


Figura 32 Diagrama de secuencia para el requisito Listar FIP

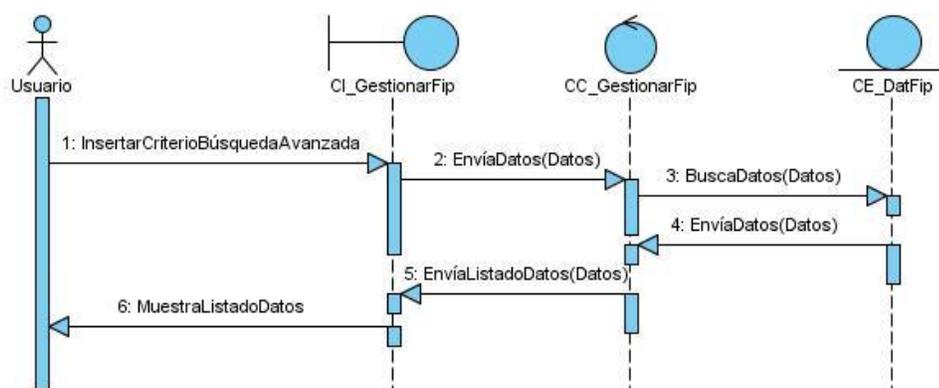


Figura 33 Diagrama de secuencia para el requisito Realizar búsqueda avanzada de FIP

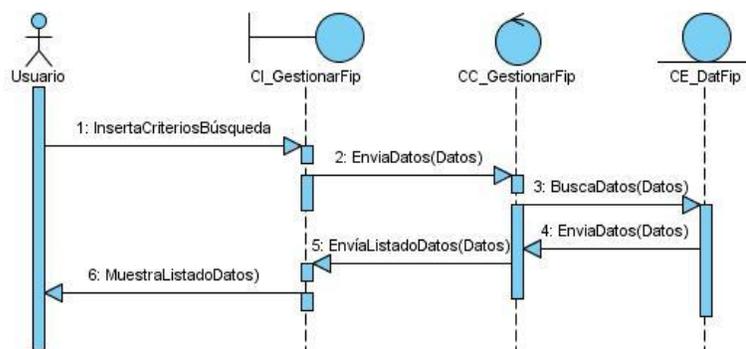


Figura 34 Diagrama de secuencia para el requisito Buscar FIP

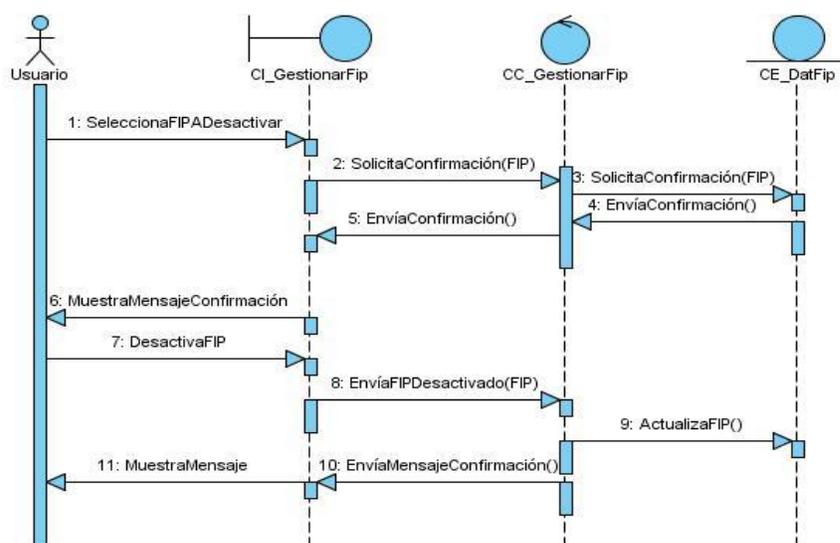


Figura 35 Diagrama de secuencia para el requisito De activar FIP

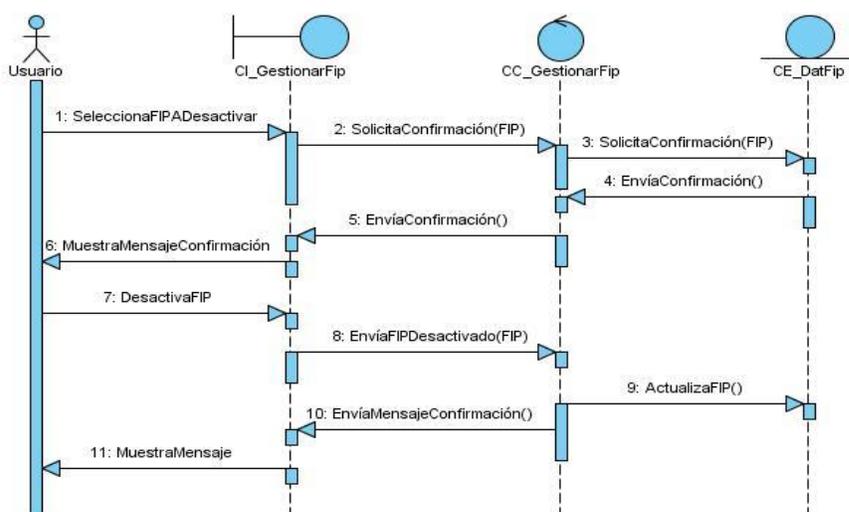


Figura 36 Diagrama de secuencia para el requisito Activar FIP

Anexo 7

No	Componente	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
1	Gestionar FIP	GestionarFipController	25	Alta	Alta	Baja
2	Gestionar FIP	gestionarfip	16	Baja	Baja	Alta
3	Gestionar FIP	DatFip	7	Baja	Baja	Alta
4	Gestionar FIP	DatElementos	25	Alta	Alta	Baja
5	Gestionar FIP	DatInvolucrados	16	Baja	Baja	Alta
6	Gestionar FIP	DatElementosDatInvolucrados	10	Media	Media	Media
7	Gestionar FIP	NomInvolucradosModel	4	Baja	Baja	Alta
8	Gestionar FIP	DatElementosModel	24	Alta	Alta	Baja
9	Gestionar FIP	DatFipModel	4	Baja	Baja	Alta
10	Gestionar FIP	DatElementosDatHistorialModel	4	Baja	Baja	Alta
11	Gestionar FIP	DatInvolucradosModel	4	Baja	Baja	Alta
12	Gestionar FIP	DatElementosHisHistorial	4	Baja	Baja	Alta
13	Gestionar FIP	DatAnexos	5	Baja	Baja	Alta
14	Gestionar FIP	NomIndicadores	4	Baja	Baja	Alta
15	Gestionar FIP	DatAnexosModel	4	Baja	Baja	Alta
16	Gestionar FIP	DatElementosDatInvolucradosModel	9	Media	Media	Media
17	Gestionar FIP	BaseDatAnexos	2	Baja	Baja	Alta
18	Gestionar FIP	BaseDatElementos	2	Baja	Baja	Alta
19	Gestionar FIP	BaseDatElementosHisHistorial	2	Baja	Baja	Alta
20	Gestionar FIP	BaseDatElementosDatInvolucrados	4	Baja	Baja	Alta
21	Gestionar FIP	BaseDatFip	2	Baja	Baja	Alta
22	Gestionar FIP	BaseNomIndicadores	2	Baja	Baja	Alta

Figura 37 Resultados de la evaluación de la métrica TOC para Gestionar FIP y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización)

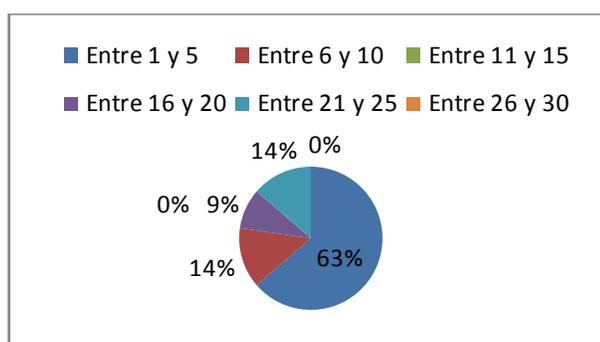


Figura 38 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos del componente Gestionar FIP

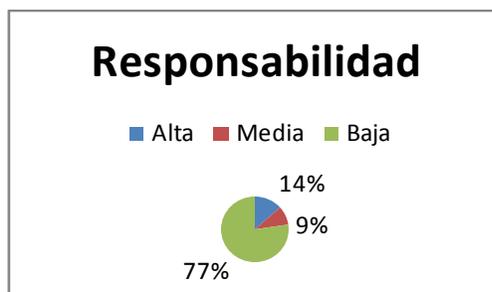


Figura 39 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad del componente Gestionar FIP

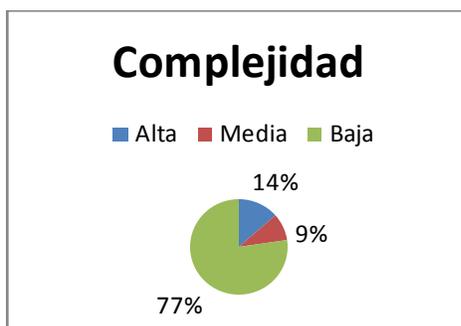


Figura 40 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación del componente Gestionar FIP



Figura 41 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización del componente Gestionar FIP

Anexo 8

No	Componente	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
1	Configurar Perfil	ConfigurarPerfil Controller	15	Alta	Alta	Baja
2	Configurar Perfil	gestionarperfil	5	Media	Media	Media
3	Configurar Perfil	gestionaratributos dinamicos	7	Media	Media	Media
4	Configurar Perfil	gestionarcalendarario	7	Media	Media	Media
5	Configurar Perfil	DatAtributos DinamicosModel	4	Baja	Baja	Alta
6	Configurar Perfil	DatCalendario Model	4	Baja	Baja	Alta
7	Configurar Perfil	DatPerfilModel	4	Baja	Baja	Alta
8	Configurar Perfil	BaseDatAtributos Dinamicos	2	Baja	Baja	Alta
9	Configurar Perfil	BaseDatCalendario	2	Baja	Baja	Alta
10	Configurar Perfil	DatAtributos Dinamicos	4	Media	Media	Media
11	Configurar Perfil	DatCalendario	4	Media	Media	Media
12	Configurar Perfil	DatPerfil	4	Baja	Baja	Alta
13	Configurar Perfil	BaseDatPerfil	2	Baja	Baja	Alta

Figura 42 Resultados de la evaluación de la métrica TOC para Configurar Perfil y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización)

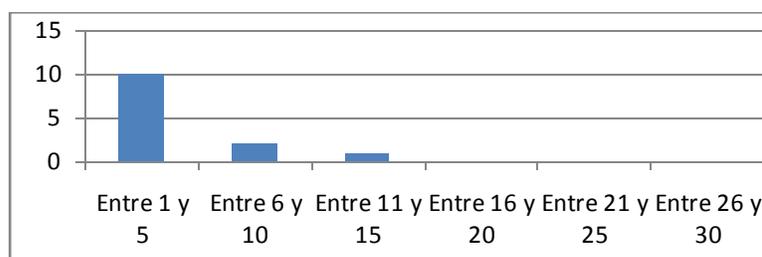


Figura 43 Representación de los resultados obtenidos (agrupados por de intervalos)

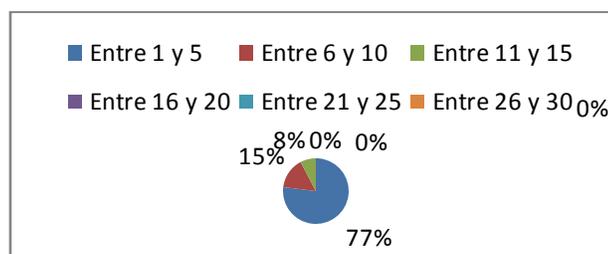


Figura 44 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos del componente Configurar Perfil

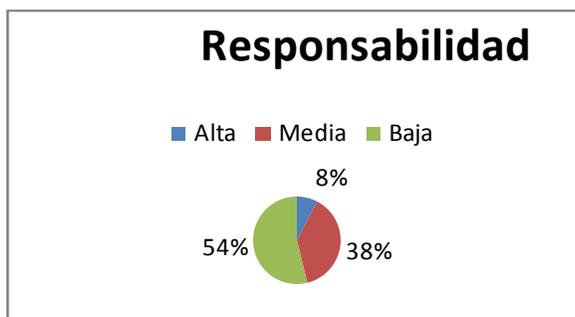


Figura 45 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad del componente Configurar Perfil

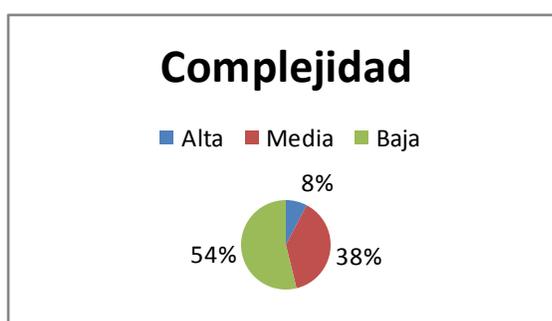


Figura 46 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación del componente Configurar Perfil

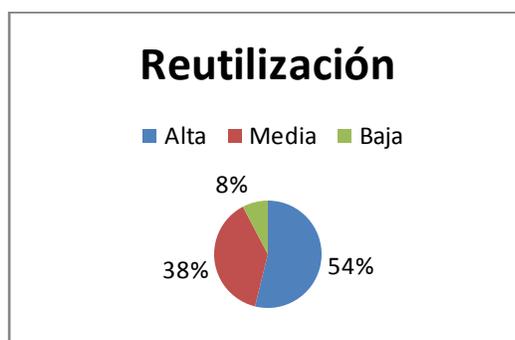


Figura 47 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización del componente Configurar Perfil